TGI-EB: A New Framework for Edge Bundling Integrating Topology, Geometry and Importance

Quan Nguyen, Seok-Hee Hong, and Peter Eades

School of Information Technologies, University of Sydney, Australia
Capital Markets CRC, Sydney, Australia
{qnguyen,shhong,peter}@it.usyd.edu.au

Abstract. Edge bundling methods became popular for visualising large dense networks; however, most of previous work mainly relies on *geometry* to define *compatibility* between the edges.

In this paper, we present a new framework for edge bundling, which tightly integrates topology, geometry and importance. In particular, we introduce new edge compatibility measures, namely *importance compatibility* and *topology compatibility*. More specifically, we present four variations of force directed edge bundling method based on the framework: Centrality-based bundling, Radial bundling, Topology-based bundling, and Orthogonal bundling.

Our experimental results with social networks, biological networks, geographic networks and clustered graphs indicate that our new framework can be very useful to highlight the most *important topological skeletal structures* of the input networks.

1 Introduction

Overviews of large and complex networks are useful for conveying information and commonly used for extracting global patterns, such as clusters and outliers in a data set. However, visualising large and complex networks is very challenging, especially, for large dense graphs due to visual clutters which hinder human understanding and analytic tasks.

Recently, edge bundling methods became popular for visualising large dense networks, and have received much attention by the Graph Drawing community and Information Visualisation community [8, 11, 13–15]. Most of the methods are based on *geometry*, i.e., a given drawing of graphs, to define *geometry compatibility* between the edges (i.e., edges are typically polylines or splines that are bundled together if they are compatible). While those edge bundling methods reduce visual clutters and show some high level edge patterns, they may not necessarily highlight the important skeletal structure of the network.

In this paper, we present a new framework for edge bundling, which tightly integrates *topology*, *geometry* and *importance*. In particular, we introduce new measures of edge compatibility based on network analysis and topology, namely *importance compatibility* and *topology compatibility*, which are independent from the geometry of the given input drawing.

M. van Kreveld and B. Speckmann (Eds.): GD 2011, LNCS 7034, pp. 123–135, 2012.

As an example to define importance compatibility, we use social network analysis methods [20]. For example, *centrality* analysis determines the relative importance of vertices and edges in a network. The *k-core* decomposition can be used to identify cohesive groups of actors within a network. As an example to define topology compatibility, we use clustered graph model.

More specifically, we present four variations of force directed edge bundling method, based on the framework:

- CenEB (Centrality-based edge bundling): tightly integrates edge centrality analysis with edge bundling.
- TopoEB (Topology-based edge bundling): tightly integrates clustered graph topology with edge bundling.
- RadEB (Radial edge bundling): tightly integrates k-core analysis with edge bundling.
- OrthoEB (Orthogonal edge bundling): uses orthogonal-like edge representation to produce orthogonal-like crossings.

We implemented our new framework and conducted experiments with social networks, biological networks, geographic networks and clustered graphs. Our experimental results show that our new framework can be useful to highlight the most important topological skeletal structures of the input network, and significantly improve visual analysis.

The new approach has proved very useful for the analysis on the integrated NF-κB protein-protein interaction and signalling transduction networks, clearly showing a number of significant functional groups. In fact, our visualisation guided biologists to derive new biological hypothesis, and currently laboratory experiments are being conducted.

2 Related Work

The use of attractions on control points for curved edges was first introduced by Brandes and Wagner [6] and later Finkel et al. [9], though the term "edge bundling" was coined several years later by others.

Holten [13] presented Hierarchical Edge Bundling method for hierarchical graphs using B-splines. Balzer et al. [4] proposed a multi-level compound visualisation using transparent surfaces and edge bundling for a hierarchical 3D visualisation.

Zhou et al. [21] presented a hierarchical edge clustering using Delaunay triangulation, where control points are hierarchically clustered by energy-based optimisation. Geometry Based Edge Bundling by Cui et al. [8] uses a control mesh for edge clustering, where edge bundles share the same control points on the mesh. Lambert et al. [15] generalised a control mesh to route graph edges using a shortest path algorithm and mesh edge weights are updated to encourage graph edges to share mesh edges.

Gansner et al. [11] improved circular layouts by merging splines of edges to minimise the total amount of ink needed to draw the edges. Cornelissen et al. [7]

presented a circular bundle view of the hierarchical graphs to study in software engineering, such as, the program execution traces.

Holten and van Wijk introduced a Force-Directed Edge Bundling (FDEB) algorithm [14], which models edges intuitively as flexible springs that can attract each other. The attractive force depends on the distance of the springs and the compatibility of the edges. The method achieves smoother bundles that are easy to read, although it incurs high computational complexity.

Telea et al. [19] proposed an Image-Based Edge Bundles that aims for coarse-grained edge shapes of bundled edges to further simplify visual representation of the network structure. Nachmanson et al. [16] consider edge bundling in layered drawings in which edges already routed as polylines or splines; the method preserves the topology of the original drawing and disambiguates edges.

Recently, Gansner et al. [10] introduced a multi-level method which approximates k-neighbor edge proximity graphs using kd-tree as input for their agglomerative bundling algorithm. They reported experiments on the approach up to one million edges in a few minutes.

This previous work on edge bundling reduces visual clutter and displays some high-level patterns. Yet the "bundles" are mainly based on geometry in disregard of the importance and the topology of the network. This motivates our new framework for edge bundling which integrates topology, geometry and importance, to highlight important skeletal structures of the networks.

3 Integrated Framework for Edge Bundling

This section presents our new generic framework for edge bundling which tightly integrates topology, geometry and importance. Our framework is *flexible*: one can use other measures for importance, geometry, and topology.

For our specific framework, we first use a force-directed edge bundling method as a basis, and then integrate geometry with *importance*, defined by centrality and k-core analysis. Finally, we further integrate *topology* into the model, defined by a clustered graph model.

3.1 New Edge Compatibility Measures

Existing edge bundling methods mainly use geometry to define *geometry compatibility* $G(e, e')$. For instance, several metrics are proposed in FDEB (force-directed edge bundling) method [14] to define geometry compatibility (in their paper $C(e, e')$ is used). "Angle" metric is designed to avoid bundling edges that are almost perpendicular. "Scale" metric ensures edges that differ considerably in length should not be bundled together. "Position" metric aims to avoid bundling edges that are very far apart. "Visibility" metric avoids bundling edges that are parallel and equal in length.

Importance Compatibility. Here, we introduce a new measure "*importance compatibility*" to integrate importance into geometry for edge bundling. Importance compatibility is conceptually to guide the bundling with respect to important edges and thus is independent from geometry, i.e., the given input drawing of a graph. Importance can be defined from application domain or specific analysis in analytic task.

Topology Compatibility. We now introduce another new notion of compatibility, called "*topology compatibility*". The topology compatibility can be defined from topological structure or combinatorial structure of given graph model. The topology compatibility, like importance compatibility, is independent from geometry.

3.2 The Framework

As an example of the integrated framework, we integrate our new edge compatibility measures into FDEB [14].

More specifically, the FDEB algorithm first inserts control points in each edge, and then uses a force-directed method to compute the position of the control points. Their forces depend on the "geometry compatibility" $G(e, e')$.

For a subdivision point e_i on edge e, the total force F_{e_i} exerted on e_i is a sum of the two spring forces exerted by two neighbors e_{i-1} and e_{i+1}, and the total of electrostatic forces F_s:

$$F_{e_i} = k_e(|\mathsf{p}_{e_{i-1}} - \mathsf{p}_{e_i}| + |\mathsf{p}_{e_i} - \mathsf{p}_{e_{i+1}}|) + F_s, \tag{1}$$

where k_e is the stiffness of edge e, and $\mathsf{p}(x)$ is the location of x.

In FDEB, electrostatic force model is

$$F_s = \sum_{e' \in \mathcal{E}} G(e, e') * |\mathsf{p}_{e_i} - \mathsf{p}_{e'_i}|^{-d}, \tag{2}$$

TGI-EB, our new general framework for edge bundling integrating topology, geometry and importance, can be described as follows. In its most general form, our electrostatic force model is

$$F_s = \sum_{e' \in \mathcal{E}} G(e, e') * I(e, e') * T(e, e') * g(|\mathsf{p}_{e_i} - \mathsf{p}_{e'_i}|), \tag{3}$$

where $\mathcal{E}$ is the set of compatible edges of e; $G(e, e')$, $I(e, e')$ and $T(e, e')$ are geometry compatibility, importance compatibility, and topology compatibility measures for a pair of edges e and e'; and g is a function of $|\mathsf{p}_{e_i} - \mathsf{p}_{e'_i}|$, e.g., $g{=}|\mathsf{p}_{e_i} - \mathsf{p}_{e'_i}|^{-d}$ where d is a numeric constant.

Note that our new framework TGI-EB is very general and flexible. For example, one can derive various models by controlling the weight parameters between $G(e, e')$, $I(e, e')$ and $T(e, e')$. Furthermore, one can define different metric to define geometry compatibility, importance compatibility and topology compatibility.

3.3 Centrality Based Edge Bundling (CenEB)

As an example to define importance compatibility, we use edge centrality. Centrality is the most well-known network analysis method, which determines the relative prominence of vertices and edges in a network [5, 20]. For instance, edge centrality analysis, which finds the important edges, has been used for mesh coarsening, analyzing biological networks and community detection.

CenEB is a special case of the general model TGI-EB described in Equation 3, which integrates importance compatibility and geometry compatibility, and $T(e,e')$ is absent. We use the edge centrality metric to highlight important edges and bundle high centrality edges together.

The most general form of our electrostatic force model for CenEB is

$$F_s = \sum_{e' \in \mathcal{E}} G(e,e') * I(e,e') * g(|\mathsf{p}_{e_i} - \mathsf{p}_{e'_i}|), \tag{4}$$

where $\mathcal{E}$ is the set of compatible edges of e, $I(e,e')$ is calculated based on the centrality values of the edges e and e'. For example, g=$|\mathsf{p}_{e_i} - \mathsf{p}_{e'_i}|^{-d}$, where d is a numeric constant, and $I(e,e')$ is defined from centrality values of e and e'.

3.4 Topology Based Edge Bundling (TopoEB)

As an example of topology compatibility, here we use a clustered graph model. A clustered graph $\mathtt{G}$=(V,E) consists of a number of clusters $\mathtt{G}_i$=(V_i,E_i). An edge that connects two nodes in the same cluster is called an *intra-cluster edge*, while an edge connecting two nodes from different clusters is called an *inter-cluster edge*.

Using the topology of the clustered graphs, we can define topology compatibility as follows:

- Two intra-cluster edges are not topology-compatible unless they belong to the same cluster;
- All inter-cluster edges are topology-compatible; in fact, they all belong to the *root* cluster $\mathtt{G}$;
- A pair of an intra-cluster edge and an inter-cluster edge is not topology-compatible.

In fact, the benefits of topology compatibility in clustered graph model are two-fold.

- First, by using topology compatibility, the number of compatible edges $\mathcal{E}$ of an edge e can be significantly reduced, which results in faster bundling iterations.
- Second, for better flexibility, one can define a topology compatibility metric $T(e,e')$, which may allow bundling intra- and inter-cluster edges together. The metric is defined in three cases depending on whether the edges e and e' are intra-cluster edges in the same cluster (intra-intra), inter-cluster edges (inter-inter), or one inter-cluster edge and one intra-cluster edge (inter-intra).

As an example of integration for TopoEB, we now integrate topology compatbility, with the model defined above for CenEB, which combines importance compatibility and geometry compatibility. TopoEB is the special case of the general model TGI-EB in Equation 3, and can be described as follows:

$$F_s = \sum_{e' \in \mathcal{E}} G(e,e') * I(e,e') * T(e,e') * g(|\mathsf{p}_{e_i} - \mathsf{p}_{e'_i}|), \tag{5}$$

where $I(e, e')$ is defined based on the centrality values of e and e', and $T(e, e')$ is defined from the clustered graph model. Note that, inter-cluster edges often have higher edge centralities than intra-cluster edges.

For example, the metric $T(e, e')$ can be simply defined as :

- c_{intra}: if e and e' are intra-cluster edges in the same cluster
- c_{inter}: if e and e' are inter-cluster edges
- c_{mix}: if e and e' is a pair of an intra-cluster edge and an inter-cluster edge;

where the contants c_{intra}, c_{inter} and c_{mix} are chosen from 0 to 1. It is worth-noting that when these constants are chosen equal, the value $T(e, e')$ is the same for every pair of edges and thus the method is said *topology-insensitive*. When c_{mix} is zero, there is no bundling between intra-cluster edge and inter-cluster edge. Generally, one may choose a value close to 1 for c_{intra} and c_{inter} and a small value for c_{mix}.

3.5 Radial Bundling (RadEB)

We now present another variation of edge bundling, called Radial bundling (RadEB), which uses a radial layout consisting of concentric circles for the input of edge bundling. The radial layout can be used to display hierarchy or k-core analysis of graphs. As a specific example in this paper, we used k-core analysis to define a radial layout.

An important group-level network analysis is to identify cohesive subgroups of actors with strong ties [5, 20]. A well-known example is the *k-cores* of a graph, each of which is a maximal-connected subgraph whose nodes have the induced degree at least k [5]. The k-core analysis has been used in social networks such as collaboration networks, and biological networks for analyzing PPI networks.

For a radial layout, we use forces to constrain the vertices u in each k-core to a circle of radius $r_u{=}f(k)$. The forces place vertices from the same k-core along the same circle.

We integrate the standard force-directed layout method with a new *radial force* for each vertex u: $F_{rad} = c_{rad}(|\mathsf{p}_u - \mathsf{p}_o| - r_u)$, where o is the center of the circles. Typically, f is a linear function, although we have also used logarithmic functions.

Clustering Constraints. We further extended RadEB to handle clustering constraints. We introduce a *similarity clustering force*, which attracts vertices of close similarity indices together. Thus, a new attraction force $f_a(u, v)$ is applied between every pair of vertices u and v:

$$F_{clus} = \sum_{(u,v)\in E} f_a(u, v) * exp(-|\mathsf{i}_u - \mathsf{i}_v|), \tag{6}$$

where i_u and i_v is clustering indices of u and v, respectively. The clustering index can defined based on the application: for example, functional-similarity for biological networks, and group membership for social networks.

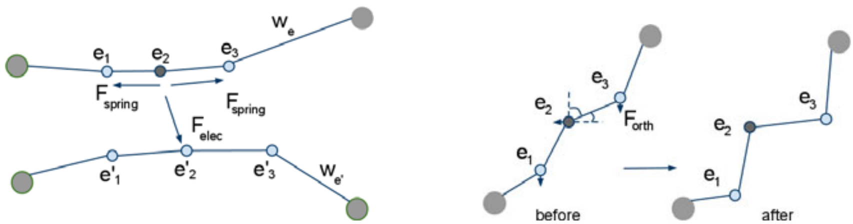
Note that our radial layout is different from the k-core visualisation by Alvarez et al. [3], which produces a radial layout using the polar coordinates. In fact, our model is more flexible, since we can further combine clustering constraints.

After producing the radial layout for visualising k-core analysis, we apply our CenEB to the resulting layout for radial bundling.

3.6 Orthogonal Edge Bundling (OrthEB)

We also present a new variation of edge representation for edge bundling, called OrthEB, which produces orthogonal-like edge bundles. Orthogonal edge bundling can be effective to produce a bundles with right angle crossings.

More specifically, we adapt forces in CenEB using magnetic field forces [18], to produce orthogonal-like bundled edges. Figure 1a and Figure 1b show example of forces in CenEB and OrthoEB in each iteration.



(a) Two interacting edges e and e'. The spring and electrostatic forces on a control point e_2

(b) Orthogonal forces on edge e

Fig. 1. Examples of forces in CenEB and OrthEB

The *orthogonal forces* are applied on the control points of each edge. The orthogonal force on point e_i is based on the tangent of the subsegment $e_{i-1}e_i$ of the edge, and e_i is sequentially moved towards the axis (either x-axis or y-axis) that forms smaller angle. Consequently, sub-segments are placed almost horizontally or vertically. In the final drawing, splines are used to connect the control points in each edge to achieve aesthetically pleasing bundling effects.

3.7 Time Complexity and Implementation

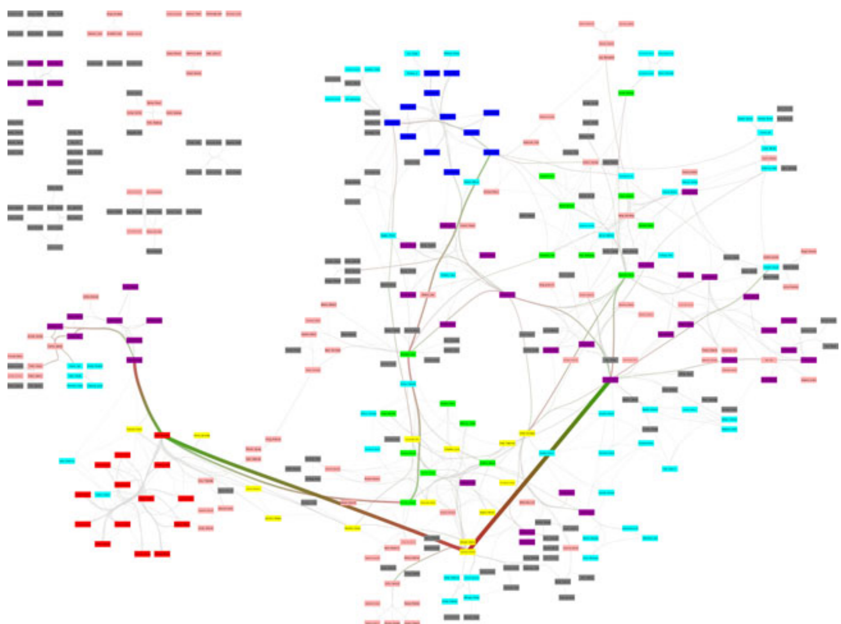
Like force-directed edge bundling (FDEB), our TGI-EB traverses every pair of edges to determine compatible edges, thus it takes $O(|E|^2)$ time for an iteration. Our force-directed radial layout with clustering constraints takes $O(|V|^2)$ time. Yet our experimental results show that our methods are quite fast for graphs with up to a few hundred nodes and two thousand edges. It took a few seconds to produce a nicely bundled layouts.

We have implemented our new edge bundling methods using our own implementation in Java for k-core radial layout, a prototype implementation of FDEB from the jFlowMap project [1], and various clustered graph layouts [12] implemented in GEOMI [2].

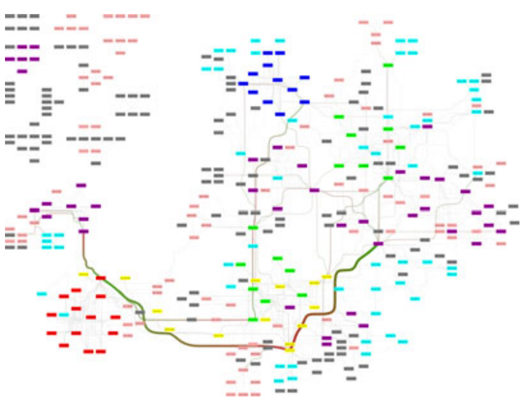
4 Experimental Results

4.1 Social Networks

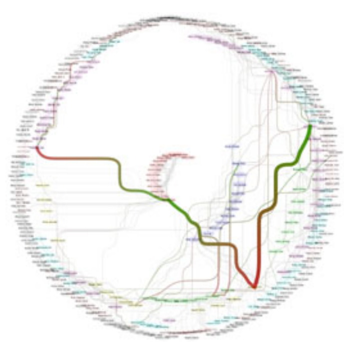
As an example of a social network, we use the 2010 Graph Drawing competition data set consisting of research collaborations in Graph Drawing research papers from 2004-2010. The data set is a graph with 362 nodes and 942 edges. Our case



(a) Using CenEB



(b) Using OrthEB



(c) Using RadEB and CenEB

Fig. 2. Collaboration network using CenEB, OrthEB and RadEB

study on collaboration networks aims to identify important researchers, research groups and collaboration patterns.

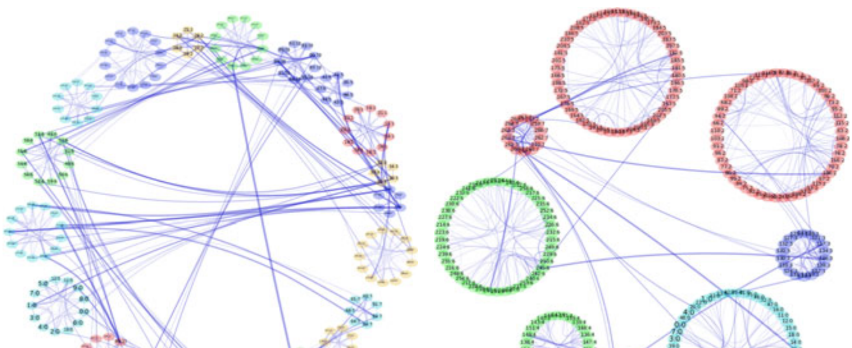
Figure 2a and Figure 2b show visualisations using CenEB and OrthEB, respectively, with edge betweenness centrality. The figures enable the following visual analyses.

First, one can easily identify the major research groups and research collaborations between the groups. The largest group is a 13-core (red) of Spanish and German researchers; the second largest group is an 11-core (blue) of an Australian clique. Second, the drawing clearly highlights researchers with high betweenness centrality: Brandes, Brandenburg, Kaumann, Kobourov, Kratochvil, Liotta, Mutzel and Wolff. Third, one can also identify several important edges with high centrality values; for example, the collaborations between Kaufmann and Kobourov; between Kauffman, Wolff and Symvonis; between Kratochvil and Wolff; between Brandes and Dwyer; between Brandes and Symvonis; between Kobourov and Sander. Finally, one can find a clique of four people with high betweenness centrality values: Brandenburg, Kobourov, Liotta and Mutzel.

Figure 2c shows a drawing of the collaboration network produced by the integration of RadEB and OrthoEB. It shows a clearer structure of the groups within different k-core circles. The inner most circle contains the 13-core group of researchers. The next circle contains the 11-core group of researchers. The drawing also highlights the important collaboration paths between researchers. Strong collaboration paths are more visible from the orthogonal-like bundled edges.

4.2 Clustered Graphs

We have experimented with randomly generated clustered graphs with different inter-cluster edge densities: sparse and dense. We use clustered graph layouts of Ho and Hong [12] implemented in GEOMI [2]. This case study uses two clustered



(a) 20 cluster clustered graph (b) 8 cluster clustered graph

Fig. 3. Dense clustered graphs in Circular-Circular layouts using TopoEB

graph layouts: circular-circular layouts and circular-force directed layouts, which draw each cluster on a circle and each cluster is, respectively, drawn using a circular layout or a spring algorithm.

We found that clustered graphs with sparse inter-cluster edges have less edge bundling effects, compared to the dense inter-cluster edge instances. Thus, we present two examples with dense inter-cluster edges. Two instances were selected from randomly generated clustered graphs: one has 20 clusters consisting of 191 nodes and 2165 edges; and the other has 8 clusters consisting of 272 nodes and 2407 edges.

Figure 3a and Figure 3b show our TopoEB results on the two clustered graphs using circular-circular layout. The figures clearly show important inter-cluster and important intra-cluster edges, and the clusters from intra-cluster edge bundles. Inter-cluster edge bundling has been shown to be effective for dense clustered graphs.

4.3 Biological Networks

This case study aims to identify new important regulatory elements and structures in a protein-protein interaction (PPI) network. We use a NF-κB PPI network consisting of 778 nodes and 1868 edges, and 14 levels of coreness.

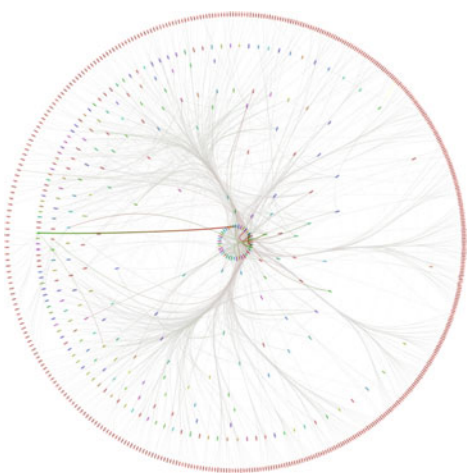


Fig. 4. NF-κB network using our RadEB and CenEB

Figure 4 shows a visualisation produced by RadEB and CenEB. It clearly shows six significant paths indicating different cell functionality. The drawing also clearly depicts several protein groups with specific functionality.

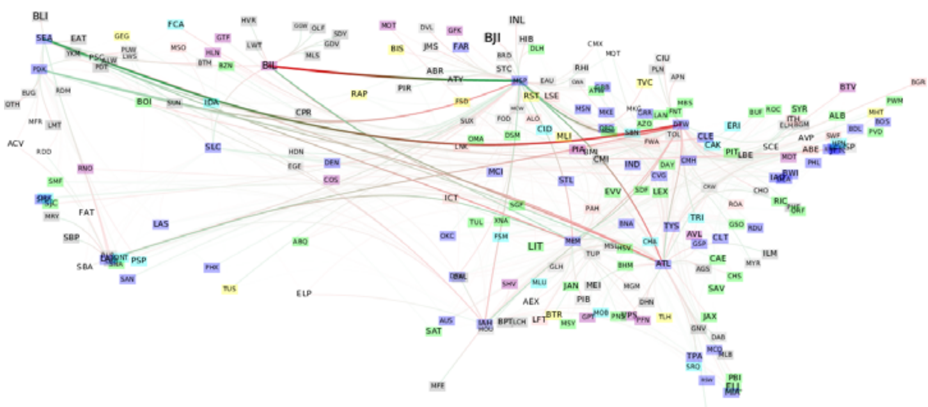
The edge centralities reflect the importance of elements. One can identify the important proteins that directly influence the translocation of the NF-κB transcription factor. Proteins that act in similar biological processes are grouped together and form network structures and motifs. In fact, our new visualisation has inspired biologists to generate a new hypothesis, based on the newly identified six important paths and important proteins around the paths. Some lab experiments are being conducted to verify the hypothesis.

4.4 Geographic Networks

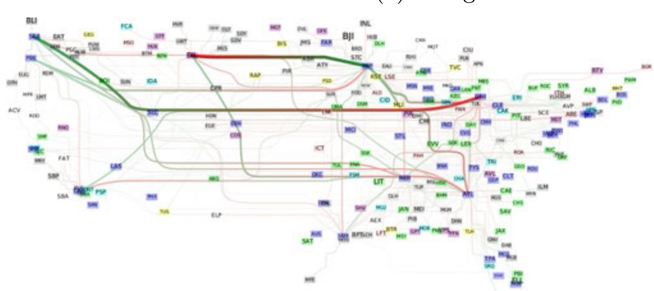
For geographic networks, critical analysis include tasks, such as flight scheduling and facility allocation. We extend our case study on airlines networks to identify important airports and flights. We use the US airlines network, which contains 235 nodes and 2101 directed edges (1297 undirected edges).

Figure 5a and Figure 5b show the airlines network using CenEB and OrthEB, respectively. Airport nodes are colored with k-core values. One can identify several important flights: between SEA and DTW, between BIL and MSP, between LAX and SEA, and between BIL and ATL.

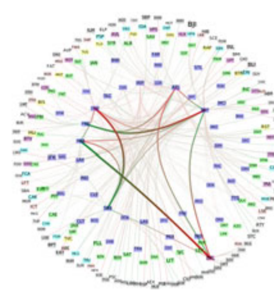
The airlines network using RadEB is shown in Figure 5c. The figure shows the most highly connected group consists of 23 airports of the 13-core around the inner most circle, including important airports, e.g., SEA, DTW, MSP, ATL, MEM and TAH. One can also identify all the important flights connecting the



(a) Using CenEB



(b) Using OrthEB



(c) Using RadEB

Fig. 5. The US Airline network using CenEB, OrthEB and RadEB

13-core airports. Interestingly, an outlier was identified as depicted in the figure: BIL airport in a low core has several "important" flights (those connected to MEM, ATL, MSP and SEA airports). This is possibly because BIL is geographically located in the middle between MEM, ATL, MSP airports (the east) and SEA airport (the west), as shown in Figure 5a and Figure 5b.

5 Future Work

Our future work is to improve the running time to address the scalability problem for huge network instances; for example, adapting the agglomative edge bundling algorithm of Gansner et al. [10].

We also plan to design new criteria or metric to evaluate the performance of edge bundling methods. We plan to generalise the magnetic field in our orthogonal edge bundling method to handle any arbitrary angles rather than just 90 degree, similar to gradient computation in Strzodka et al. [17].

Acknowledgements. Thanks to S. J. Janowski, J. Stoye and C. Kaltschmidt from Faculty of Technology, University of Bielefeld, Germany for the PPI networks, and valuable discussions on the biological hypotheses and lab experiments.

References

1. jFlowMap (2010), http://code.google.com/p/jflowmap/
2. Ahmed, A., Dwyer, T., Forster, M., Fu, X., Ho, J., Hong, S.-H., Koschützki, D., Murray, C., Nikolov, N.S., Taib, R., Tarassov, A., Xu, K.: GEOMI: Geometry for Maximum Insight. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 468–479. Springer, Heidelberg (2006), http://dblp.uni-trier.de/db/conf/gd/gd2005.html#AhmedDFFHHKMNTTX05
3. Alvarez, H.J.I., Dall, A.L., Barrat, A., Vespignani, A.: Large scale networks fingerprinting and visualization using the k-core decomposition. In: Advances in Neural Information Processing Systems, vol. 18, p. 41 (2006)
4. Balzer, M., Deussen, O.: Level-of-detail visualization of clustered graph layouts. In: APVIS, pp. 133–140 (2007)
5. Brandes, U., Erlebach, T.: Network analysis: methodological foundations. Springer, Heidelberg (2005)
6. Brandes, U., Wagner, D.: Using Graph Layout to Visualize Train Interconnection Data. In: Whitesides, S.H. (ed.) GD 1998. LNCS, vol. 1547, pp. 44–56. Springer, Heidelberg (1999)
7. Cornelissen, B., Zaidman, A., Holten, D., Moonen, L., van Deursen, A., van Wijk, J.J.: Execution trace analysis through massive sequence and circular bundle views. Journal of Systems and Software 81(12), 2252–2268 (2008)
8. Cui, W., Zhou, H., Qu, H., Wong, P.C., Li, X.: Geometry-based edge clustering for graph visualization. IEEE Transactions on Visualization and Computer Graphics, 1277–1284 (2008)
9. Finkel, B., Tamassia, R.: Curvilinear Graph Drawing using the Force-Directed Method. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 448–453. Springer, Heidelberg (2005)

10. Gansner, E., Hu, Y., North, S., Scheidegger, C.: Multilevel agglomerative edge bundling for visualizing large graphs. In: IEEE PacificVis, pp. 187–194 (2011)
11. Gansner, E., Koren, Y.: Improved Circular Layouts. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 386–398. Springer, Heidelberg (2007)
12. Ho, J., Hong, S.-H.: Drawing Clustered Graphs in Three Dimensions. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 492–502. Springer, Heidelberg (2006)
13. Holten, D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. IEEE Transactions on Visualization and Computer Graphics, 741–748 (2006)
14. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. Computer Graphics Forum 28(3), 983–990 (2009)
15. Lambert, A., Bourqui, R., Auber, D.: Winding roads: Routing edges into bundles. Computer Graphics Forum 29, 853–862 (2010)
16. Pupyrev, S., Nachmanson, L., Kaufmann, M.: Improving Layered Graph Layouts with Edge Bundling. In: Brandes, U., Cornelsen, S. (eds.) GD 2010. LNCS, vol. 6502, pp. 329–340. Springer, Heidelberg (2011)
17. Strzodka, R., Telea, A.: Generalized distance transforms and skeletons in graphics hardware. In: VisSym, pp. 221–230 (2004)
18. Sugiyama, K., Misue, K.: Graph drawing by the magnetic spring model. Journal of Visual Languages and Computing 6(3), 217–231 (1995)
19. Telea, A., Ersoy, O.: Image-Based Edge Bundles: Simplified Visualization of Large Graphs. Computer Graphics Forum 29, 843–852 (2010)
20. Wasserman, S., Faust, K.: Social network analysis: Methods and applications, 1st edn. Cambridge University Press (1994)
21. Zhou, H., Yuan, X., Cui, W., Qu, H., Chen, B.: Energy-based hierarchical edge clustering of graphs. In: IEEE PacificVis, pp. 55–61 (2008)