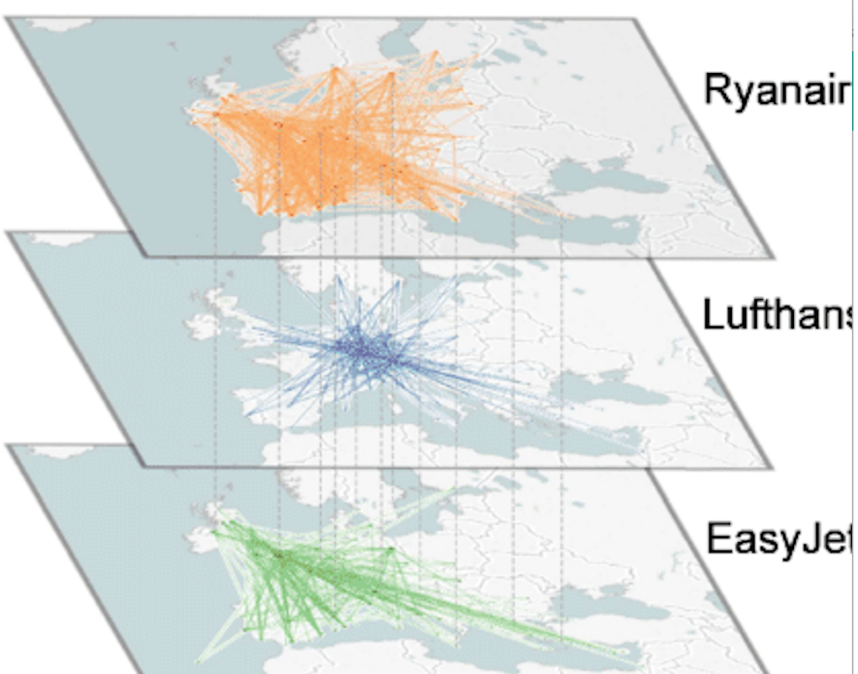
Multilayer Network Visualization

Course : Data Visualization
Lecturer : Tamara Mchedlidze
Utrecht University, Dept. of Information and Computing Sciences



Wu et al, 2017


Telea et al, 2011

Lecture Overview

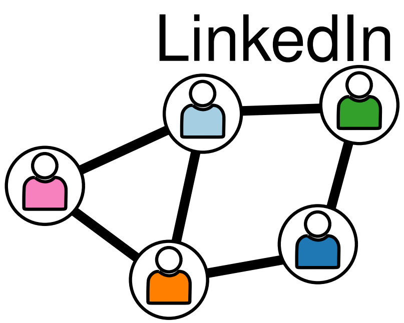
- Multilayer network
- Visualization types for multilayer networks
- Algorithm for visualization in 2.5D
- Edge simplification - bundling
- An algorithm for edge bundling
- Proposed technique for the implementation

Adding complexity

- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex

Adding complexity

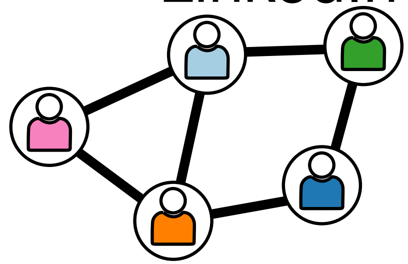
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex

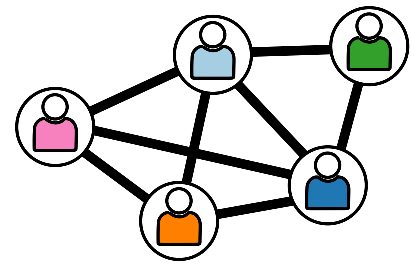


Adding complexity

- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex

LinkedIn

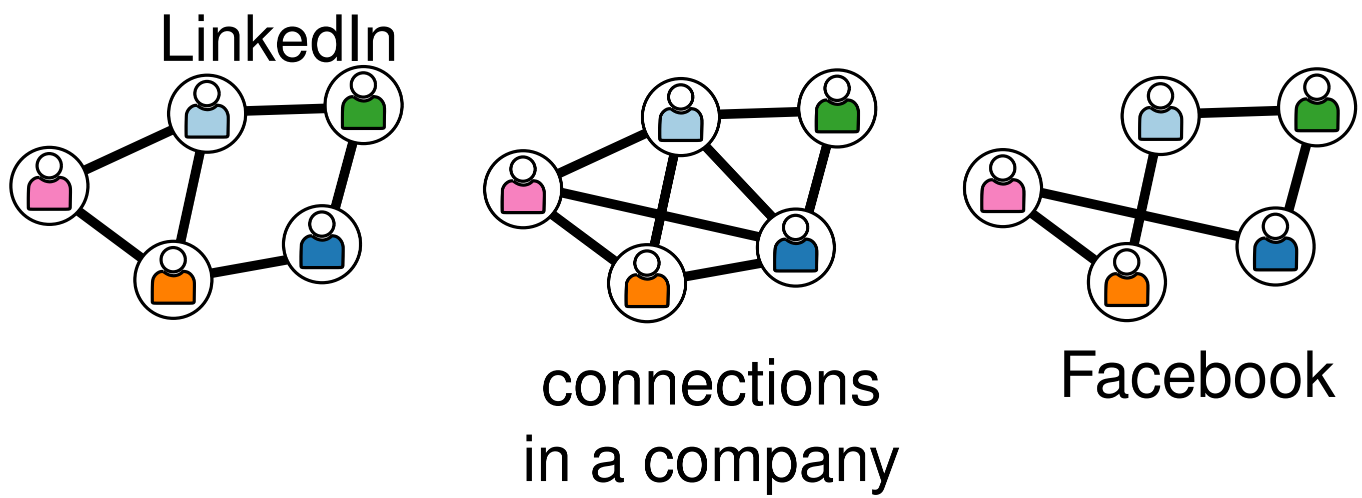




connections
in a company

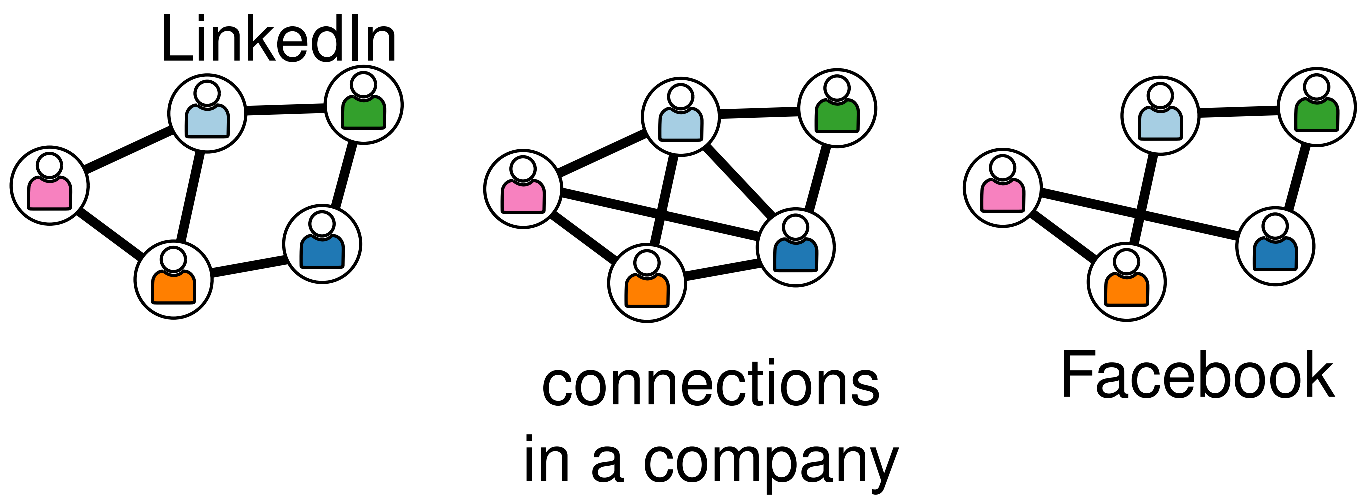
Adding complexity

- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



Adding complexity

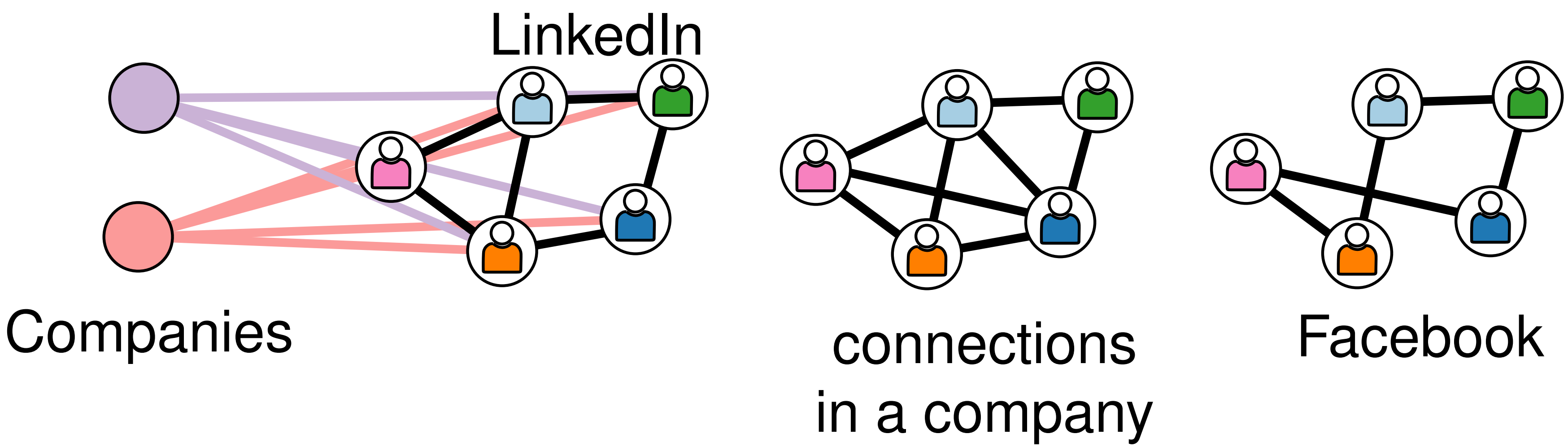
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other

Adding complexity

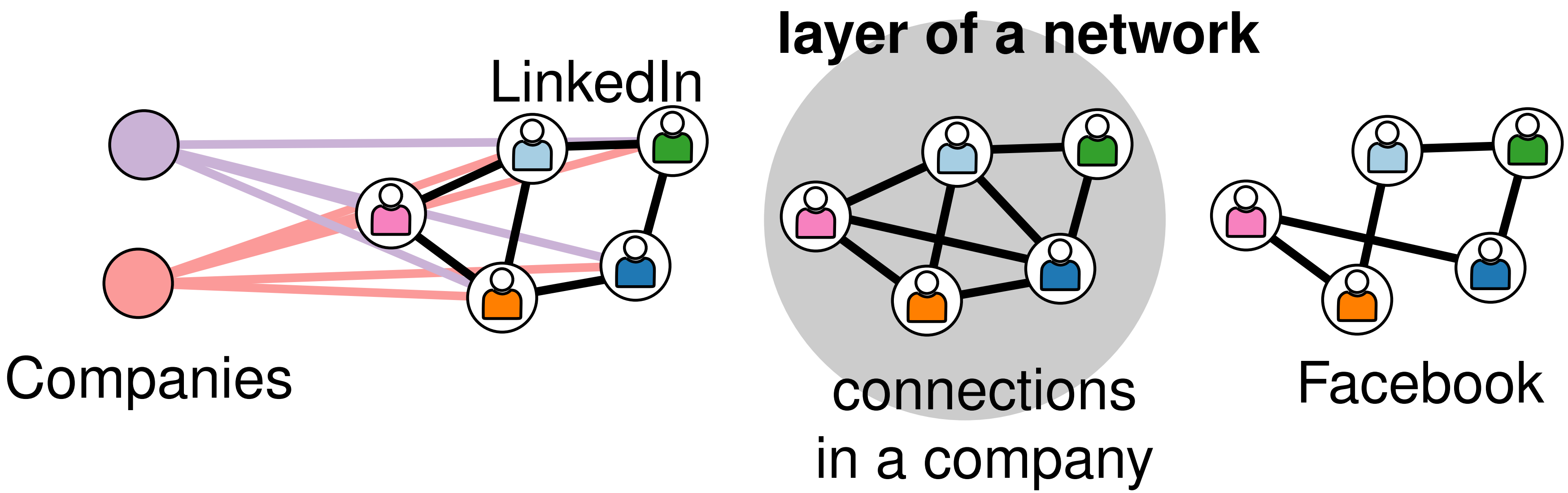
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other
- nodes of other types

Adding complexity

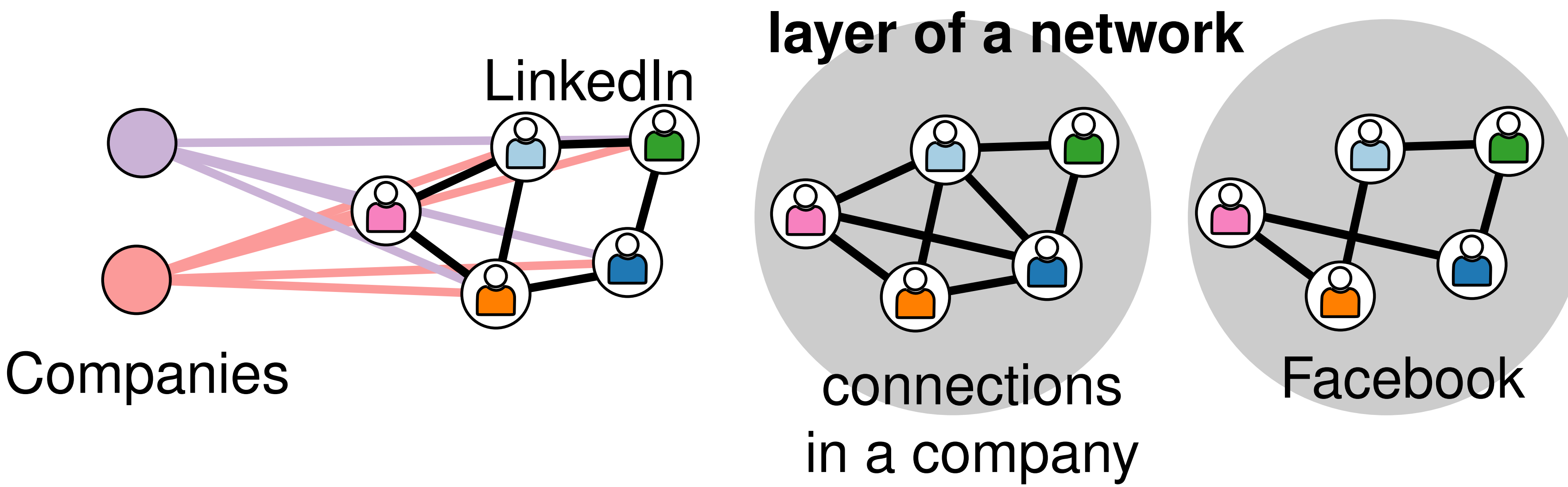
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other
- nodes of other types

Adding complexity

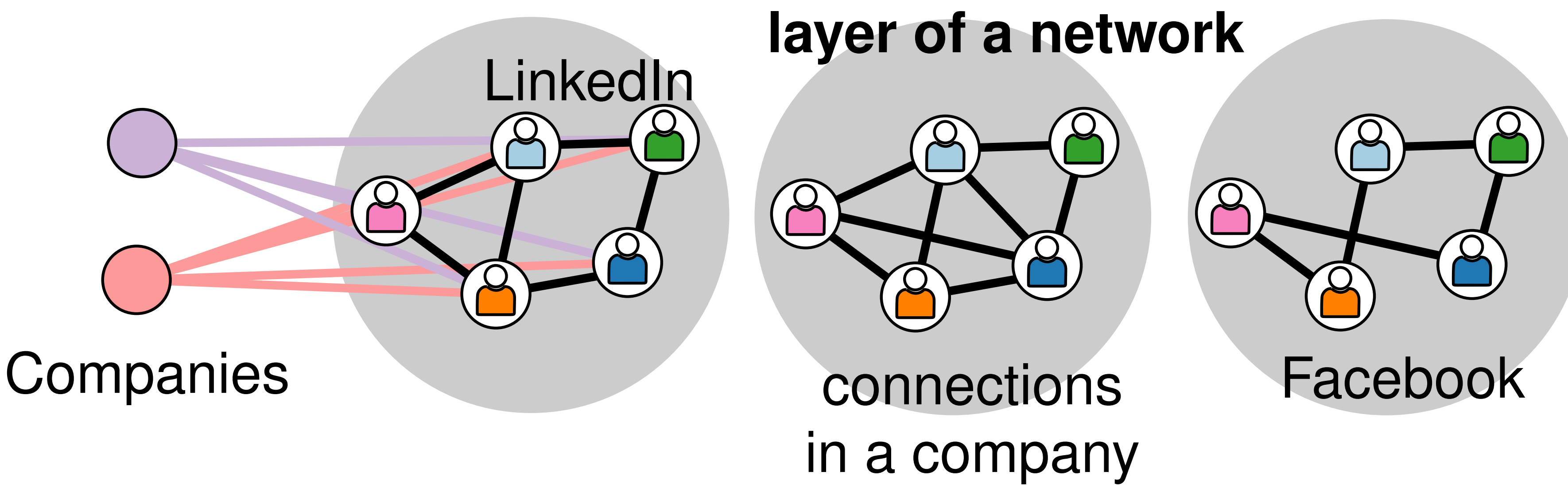
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other
- nodes of other types

Adding complexity

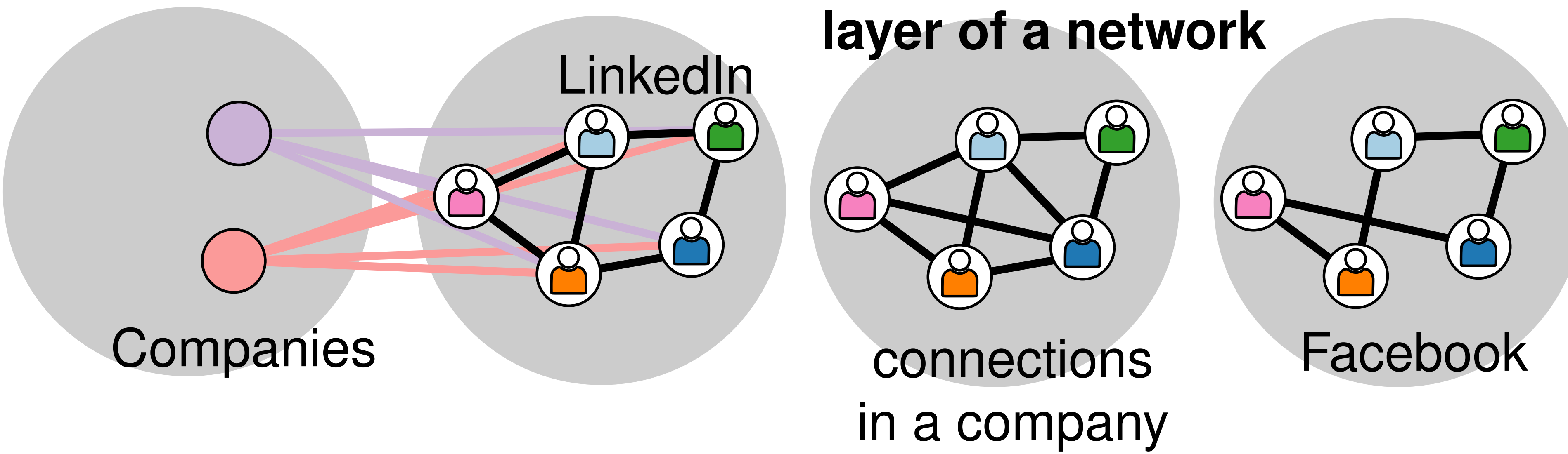
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other
- nodes of other types

Adding complexity

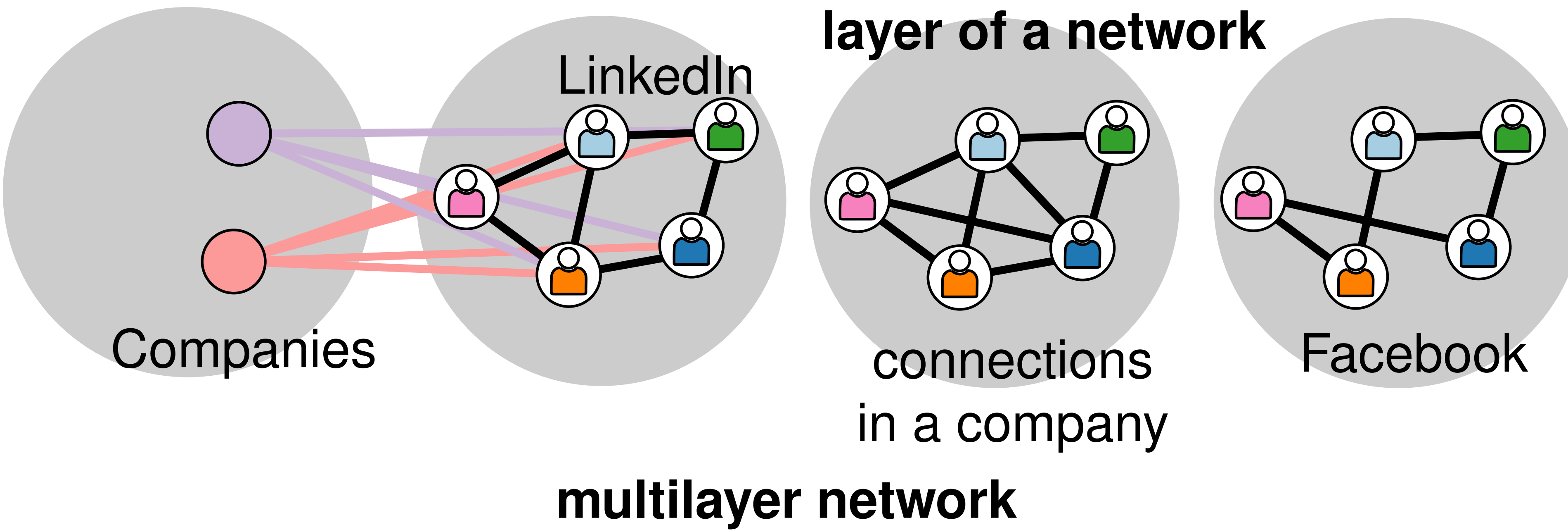
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other
- nodes of other types

Adding complexity

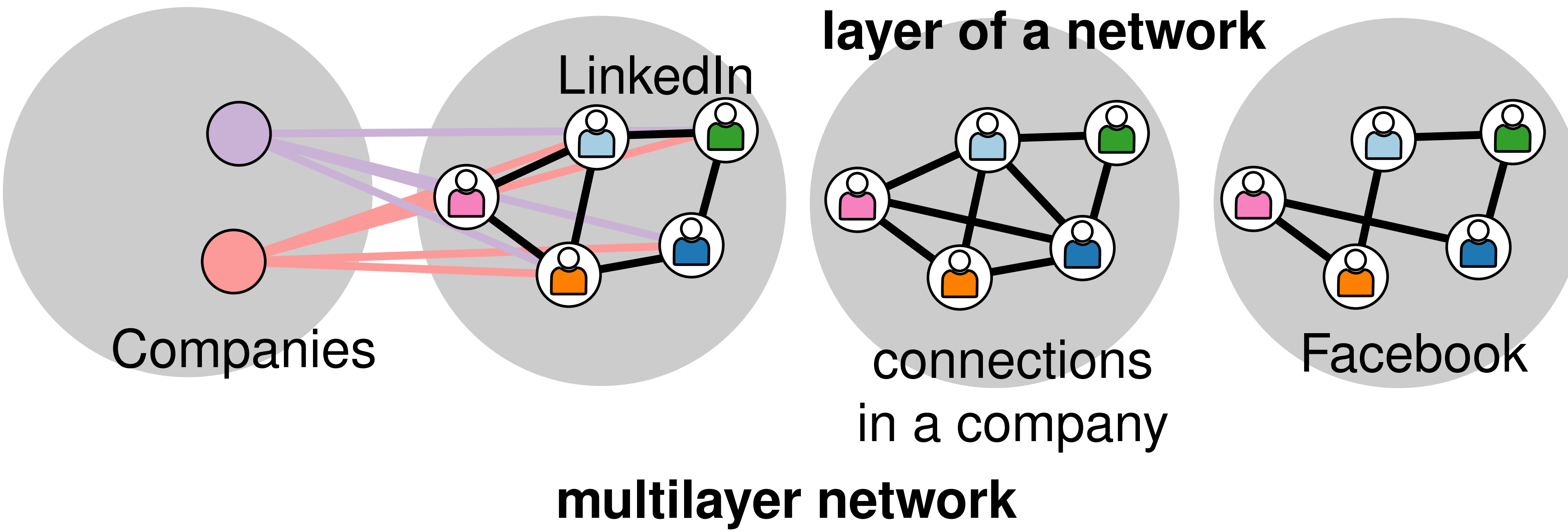
- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other
- nodes of other types

Adding complexity

- definition of network/graph we used till now (nodes, edges and perhaps labels) is a simplification of reality, where the network are often way more complex



- changes in one network effect changes in the other
- nodes of other types
- analysis of graph patters across layers reveal complex facts about the data

Multilayer Network

- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$

Multilayer Network

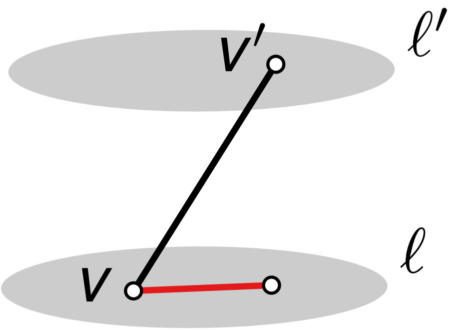
- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers

Multilayer Network

- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers
- V – set of all distinct vertices;
  L – set of layers;
  $V_m \subseteq V \times L$ – vertices of a multilayered graph

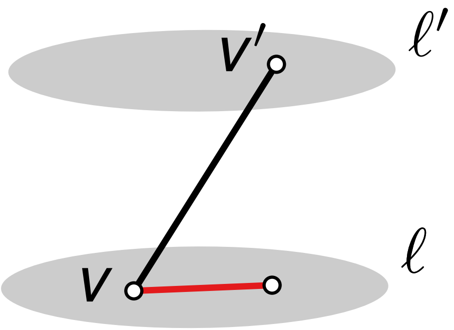
Multilayer Network

- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers
- V – set of all distinct vertices;
  L – set of layers;
  $V_m \subseteq V \times L$ – vertices of a multilayered graph
- $E_m \subseteq V_m \times V_m$ – edges of a multilayer graph connect pairs (v, ℓ) and (v', ℓ')



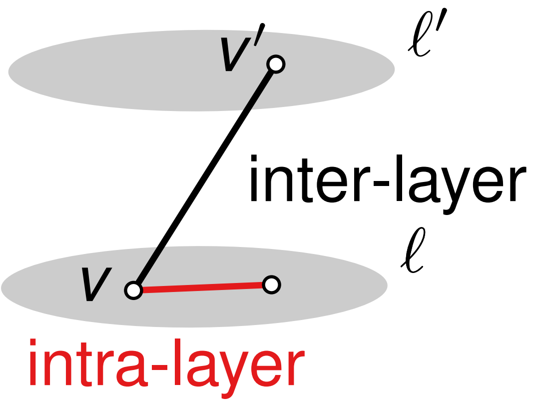
Multilayer Network

- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers
- V – set of all distinct vertices;
  L – set of layers;
  $V_m \subseteq V \times L$ – vertices of a multilayered graph
- $E_m \subseteq V_m \times V_m$ – edges of a multilayer graph connect pairs (v, ℓ) and (v', ℓ')
- edge $\{(v, \ell), (v', \ell')\}$ is *inter-layer* if $\ell \neq \ell'$ and is *intra-layer* if $\ell = \ell'$ (edge living in a layer)



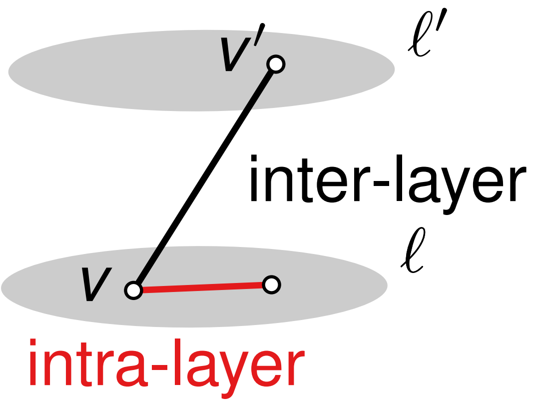
Multilayer Network

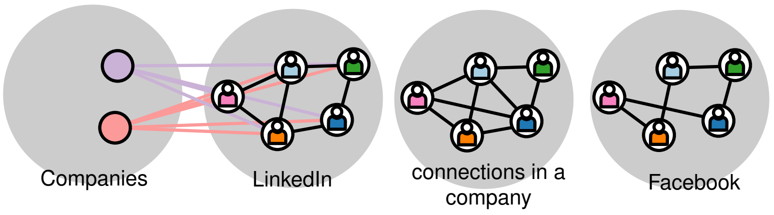
- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers
- V – set of all distinct vertices;
  L – set of layers;
  $V_m \subseteq V \times L$ – vertices of a multilayered graph
- $E_m \subseteq V_m \times V_m$ – edges of a multilayer graph connect pairs (v, ℓ) and (v', ℓ')
- edge $\{(v, \ell), (v', \ell')\}$ is *inter-layer* if $\ell \neq \ell'$ and is *intra-layer* if $\ell = \ell'$ (edge living in a layer)



Multilayer Network

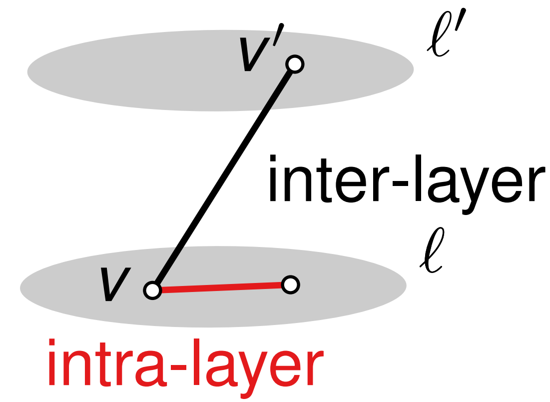
- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers
- V – set of all distinct vertices;
  L – set of layers;
  $V_m \subseteq V \times L$ – vertices of a multilayered graph
- $E_m \subseteq V_m \times V_m$ – edges of a multilayer graph connect pairs (v, ℓ) and (v', ℓ')
- edge $\{(v, \ell), (v', \ell')\}$ is *inter-layer* if $\ell \neq \ell'$ and is *intra-layer* if $\ell = \ell'$ (edge living in a layer)



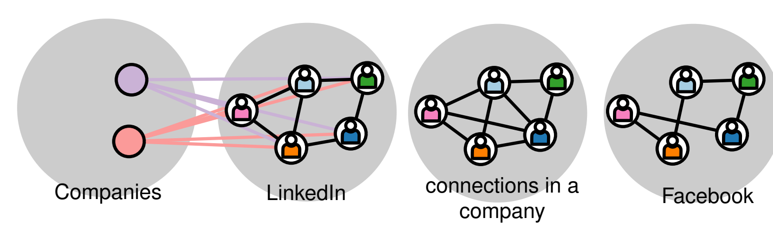


Multilayer Network

- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers
- V – set of all distinct vertices;
  L – set of layers;
  $V_m \subseteq V \times L$ – vertices of a multilayered graph

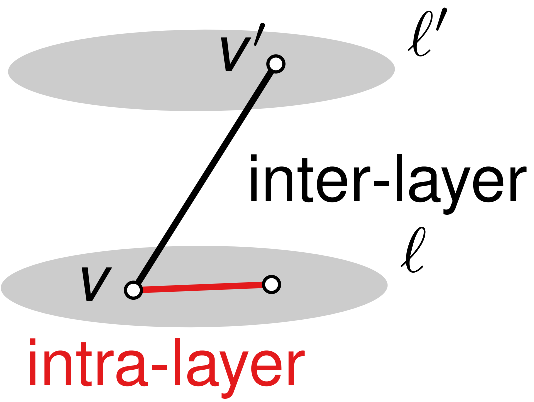


- $E_m \subseteq V_m \times V_m$ – edges of a multilayer graph connect pairs (v, ℓ) and (v', ℓ')
- edge $\{(v, \ell), (v', \ell')\}$ is *inter-layer* if $\ell \neq \ell'$ and is *intra-layer* if $\ell = \ell'$ (edge living in a layer)
- here layers are $\{\ell_1, \ell_2, \ell_3, \ell_4\}$,
  ℓ_1 =LinkedIn,
  ℓ_2 =connections in a company,
  ℓ_3 =Facebook,
  ℓ_4 =companies

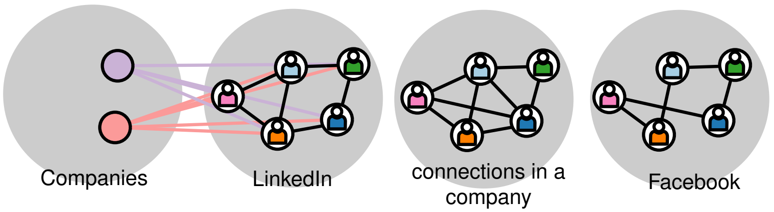


Multilayer Network

- Standard graph definition $G = (V, E)$, $E \subseteq V \times V$
- For multilayer network we need to specify which layer a node belongs to – can belong to multiple layers
- V – set of all distinct vertices;
  L – set of layers;
  $V_m \subseteq V \times L$ – vertices of a multilayered graph



- $E_m \subseteq V_m \times V_m$ – edges of a multilayer graph connect pairs (v, ℓ) and (v', ℓ')
- edge $\{(v, \ell), (v', \ell')\}$ is *inter-layer* if $\ell \neq \ell'$ and is *intra-layer* if $\ell = \ell'$ (edge living in a layer)
- here layers are $\{\ell_1, \ell_2, \ell_3, \ell_4\}$,
  ℓ_1 =LinkedIn,
  ℓ_2 =connections in a company,
  ℓ_3 =Facebook,
  ℓ_4 =companies



- node v = appears as (v, ℓ_1), (v, ℓ_2), (v, ℓ_3) in V_m

Multilayer Network

multilayer networks appear as models in

- biology: genomic, proteomic and metabolomic data to model intricate biological processes

Multilayer Network

multilayer networks appear as models in

- biology: genomic, proteomic and metabolomic data to model intricate biological processes
- civil infrastructure: urban planning taking into account the interplay between multiple networks such as transportation networks, energy networks, telecommunication networks and water/wastewater networks

Multilayer Network

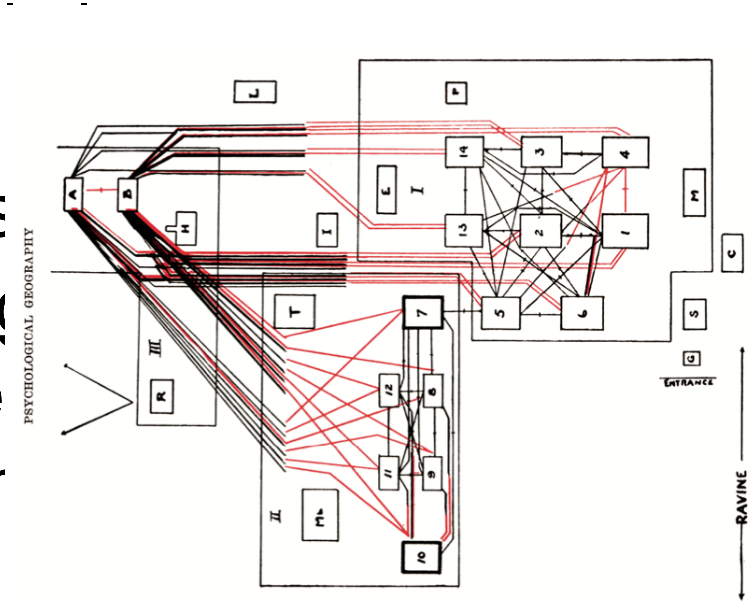
multilayer networks appear as models in

- biology: genomic, proteomic and metabolomic data to model intricate biological processes
- civil infrastructure: urban planning taking into account the interplay between multiple networks such as transportation networks, energy networks, telecommunication networks and water/wastewater networks
- epidemiology, sociology (including criminology), digital humanities

Multilayer Network

multilayer networks appear as mode

- biology: genomic, proteomic and
  model intricate biological process
- civil infrastructure: urban plannin
  the interplay between multiple ne
  transportation networks, energy r
  telecommunication networks and
  networks

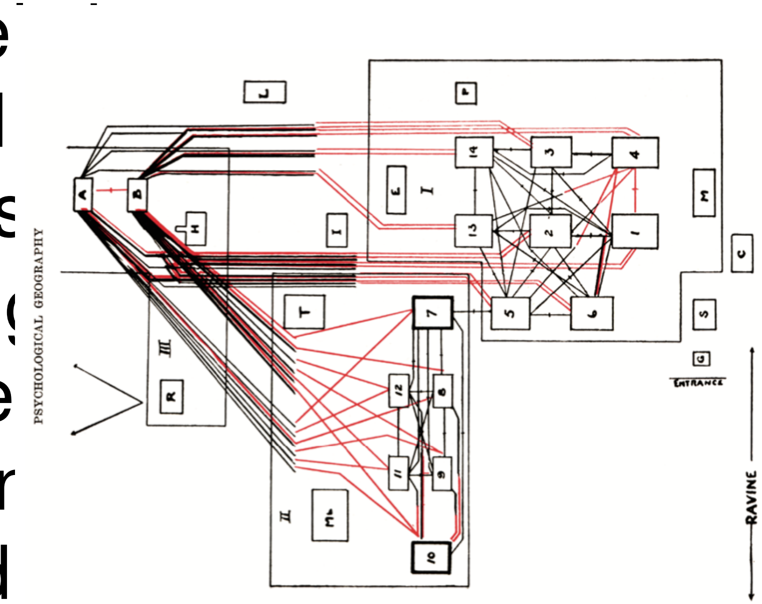


date back to 50s – notion of many relationships between individuals in the sociograms introduced by Moreno

Multilayer Network

multilayer networks appear as mode

- biology: genomic, proteomic and model intricate biological process
- civil infrastructure: urban plannin the interplay between multiple ne transportation networks, energy r telecommunication networks and networks



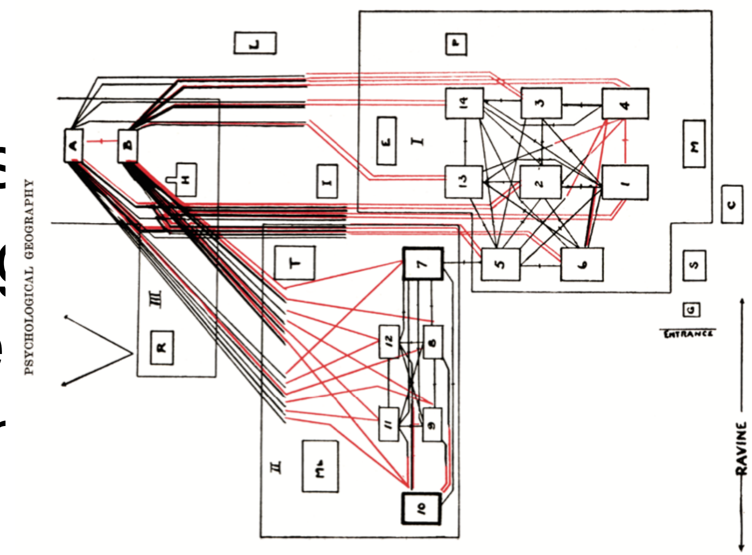
date back to 50s – notion of many relationships between individuals in the sociograms introduced by Moreno

many names: multi-label, multi-edge, multirelational, multiplex, heterogeneous, multimodal, multiple edge set networks, interdependent networks, interconnected networks, networks of networks, ... – unified under a single framework by Kivelä et al. 2014

Multilayer Network

multilayer networks appear as mode

- biology: genomic, proteomic and model intricate biological process
- civil infrastructure: urban plannin the interplay between multiple ne transportation networks, energy r telecommunication networks and networks

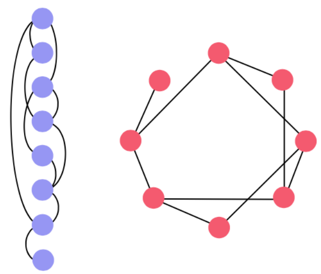


Have you seen relational data that need to be modeled as multilevel networks?

networks, networks of networks, ... – unified under a single framework by Kivelä et al. 2014

Multilayer Network Visualizations

Types of visualizations of multilayer networks

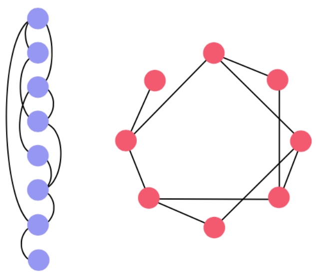


1-dimensional: circular, linear

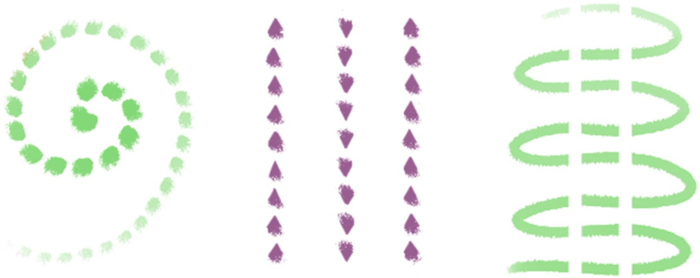
- 1-dimensional representations rely on Gestalt principle of continuation to perceptually group the layers

Multilayer Network Visualizations

Types of visualizations of multilayer networks



1-dimensional: circular, linear

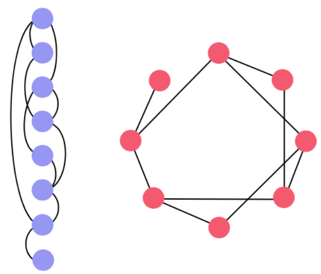


Gestalt principle: continuation, continuity

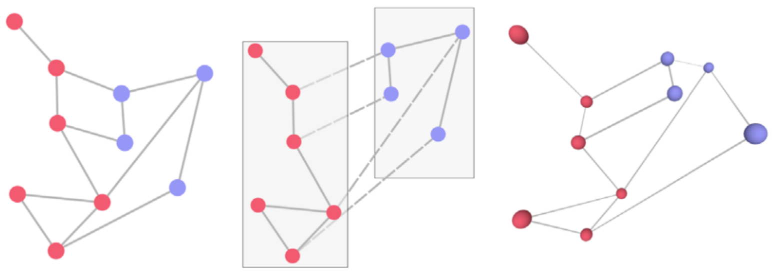
- 1-dimensional representations rely on Gestalt principle of continuation to perceptually group the layers

Multilayer Network Visualizations

Types of visualizations of multilayer networks



1-dimensional: circular, linear

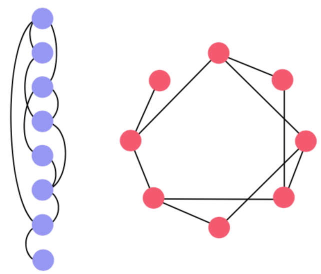


2D, 2.5D, 3D representations

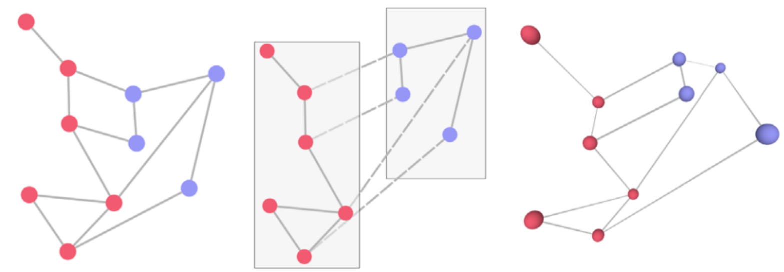
- 1-dimensional representations rely on Gestalt principle of continuation to perceptually group the layers

Multilayer Network Visualizations

Types of visualizations of multilayer networks



1-dimensional: circular, linear

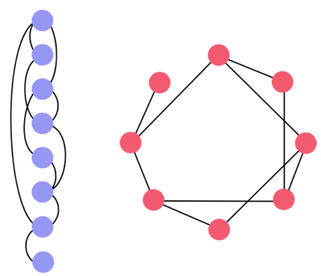


2D, 2.5D, 3D representations

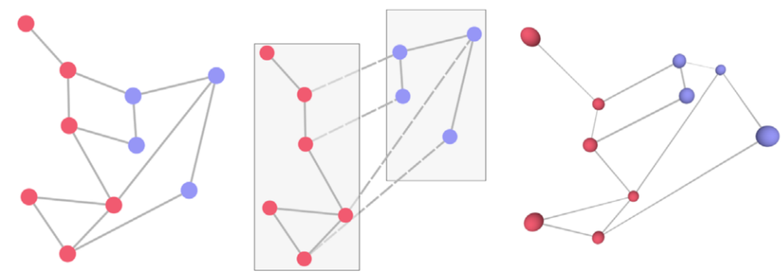
- 1-dimensional representations rely on Gestalt principle of continuation to perceptually group the layers
- 2D - layers indicated by separation (proximity Gestalt principle), color (similarity)

Multilayer Network Visualizations

Types of visualizations of multilayer networks



1-dimensional: circular, linear

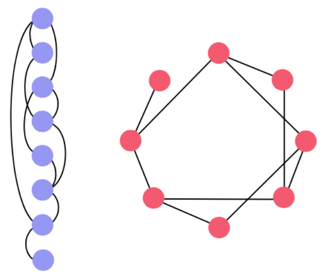


2D, 2.5D, 3D representations

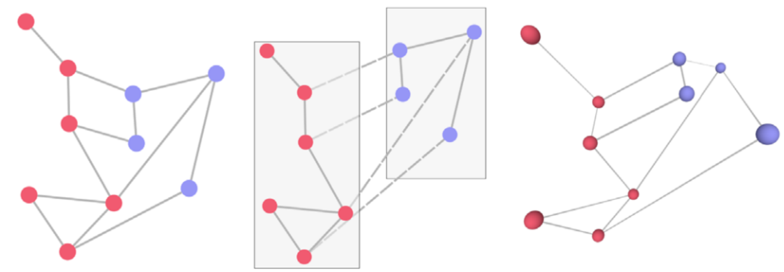
- 1-dimensional representations rely on Gestalt principle of continuation to perceptually group the layers
- 2D - layers indicated by separation (proximity Gestalt principle), color (similarity)
- 2.5D - layers are are different planes stacked next to each other

Multilayer Network Visualizations

Types of visualizations of multilayer networks



1-dimensional: circular, linear

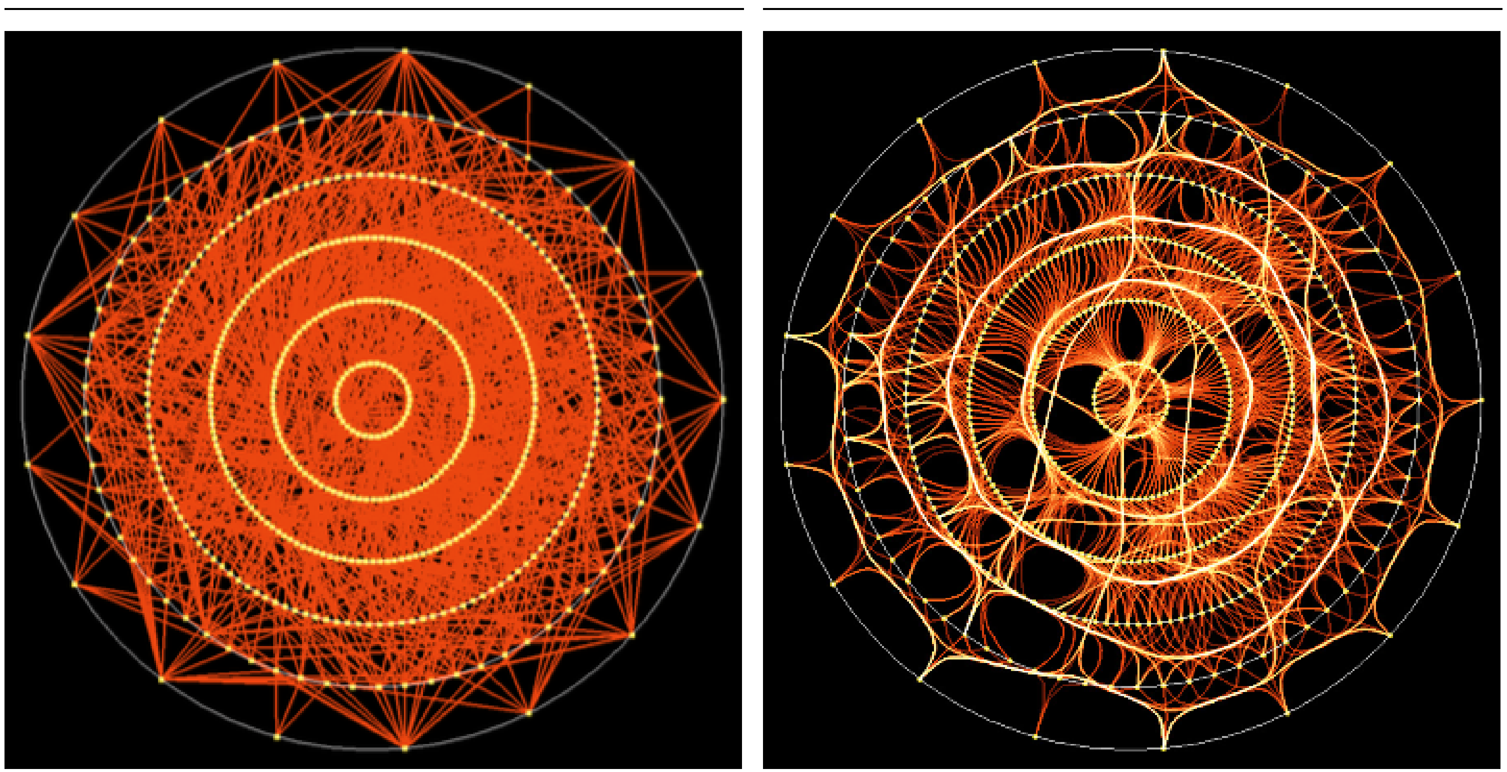


2D, 2.5D, 3D representations

- 1-dimensional representations rely on Gestalt principle of continuation to perceptually group the layers
- 2D - layers indicated by separation (proximity Gestalt principle), color (similarity)
- 2.5D - layers are are different planes stacked next to each other
- 3D – depth is indicating the layer, camera movement is necessary

1-dimensional representaiton: circular

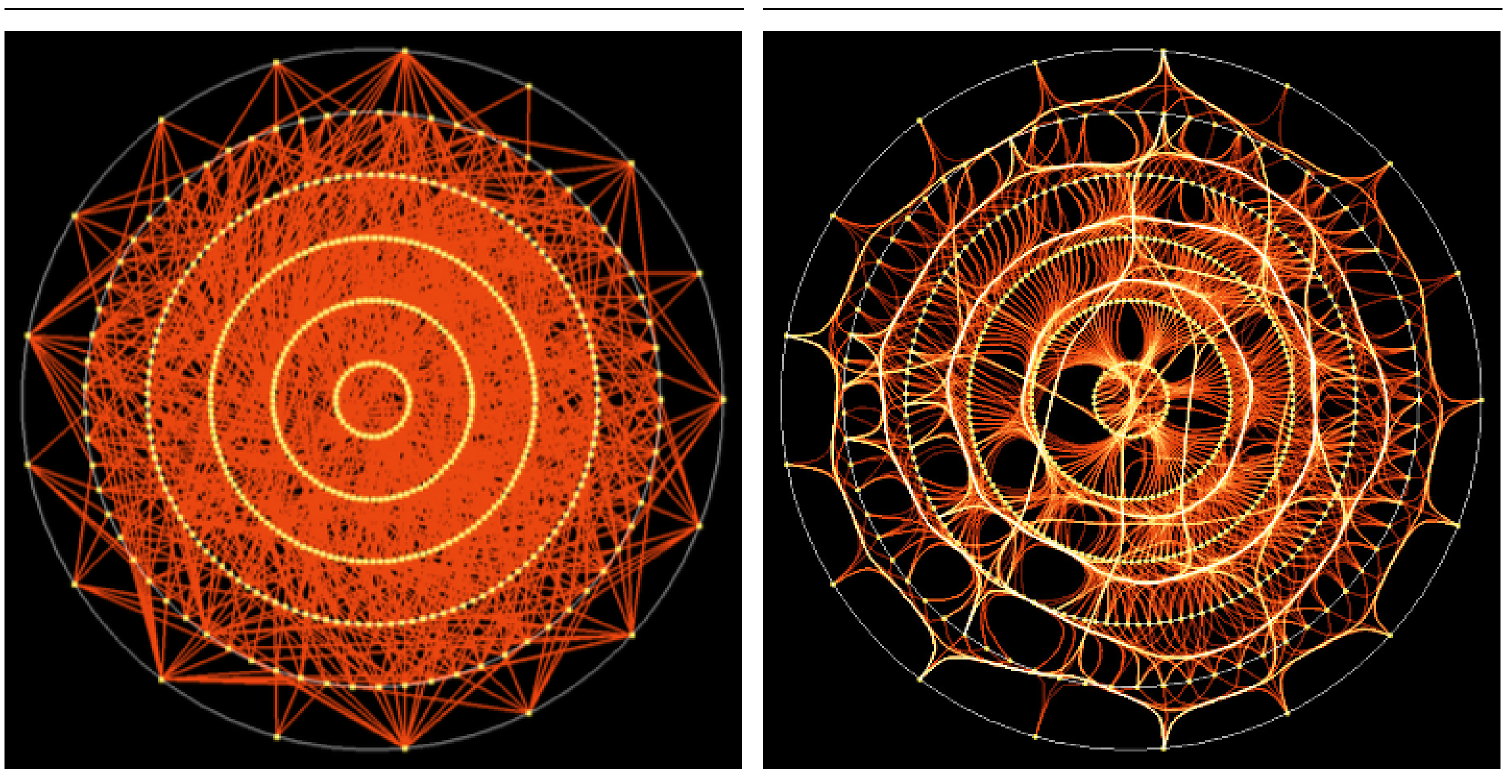
- Mushroom data set from the UC Irvine Machine Learning Repository



Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm, Bothorel et al 2013

1-dimensional representaiton: circular

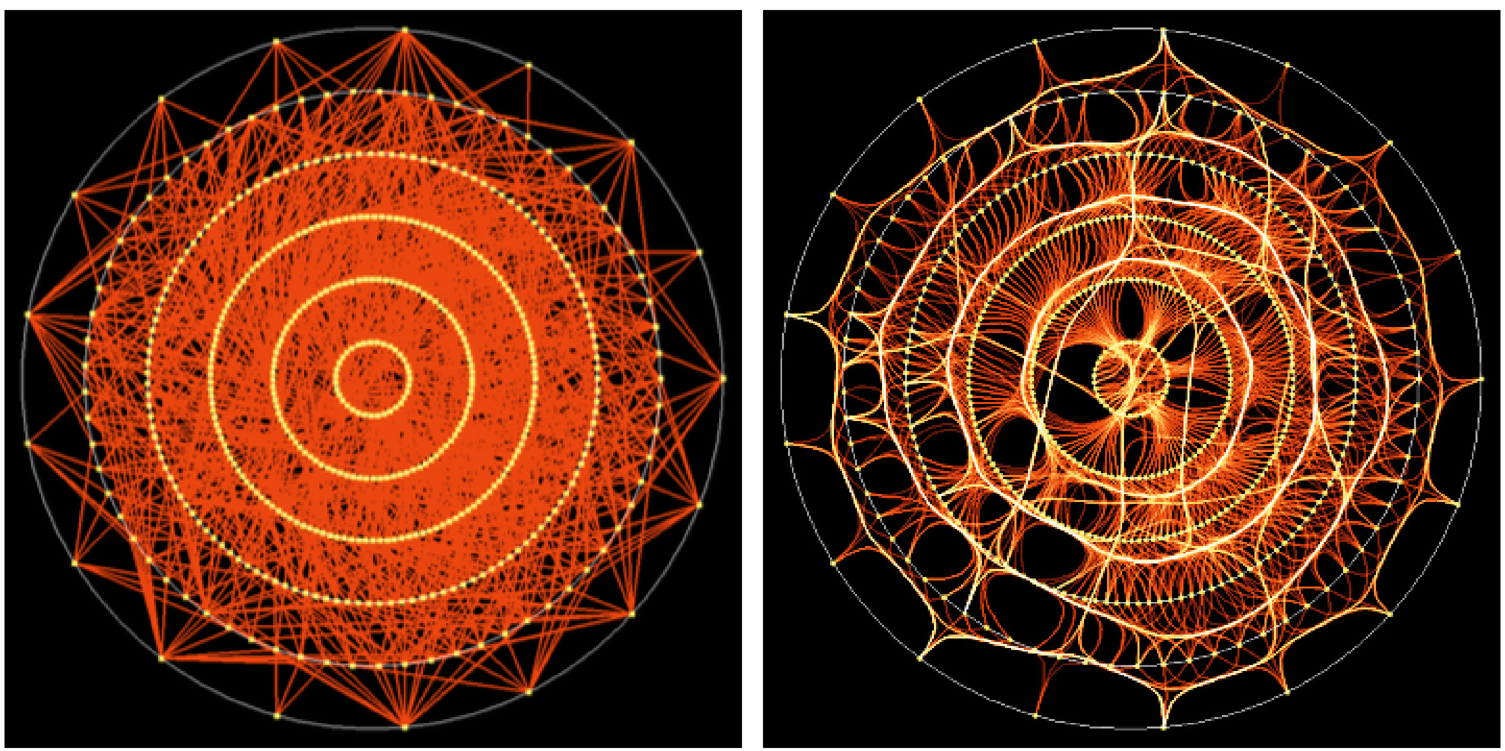
- Mushroom data set from the UC Irvine Machine Learning Repository
- nodes – mushrooms



Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm, Bothorel et al 2013

1-dimensional representaiton: circular

- Mushroom data set from the UC Irvine Machine Learning Repository
- nodes – mushrooms
- layers – attributes: the edibility, the cap shape, the odor, the ring, etc.

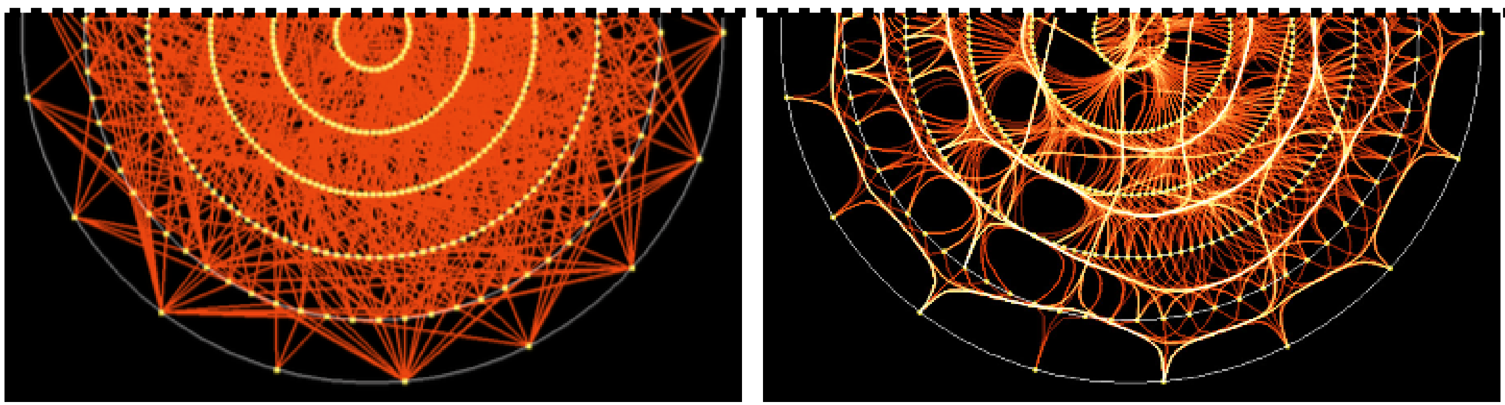


Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm, Bothorel et al 2013

1-dimensional representaiton: circular

- Mushroom data set from the UC Irvine Machine Learning Repository
- nodes – mushrooms
- layers – attributes: the edibility, the cap shape, the odor, the ring,

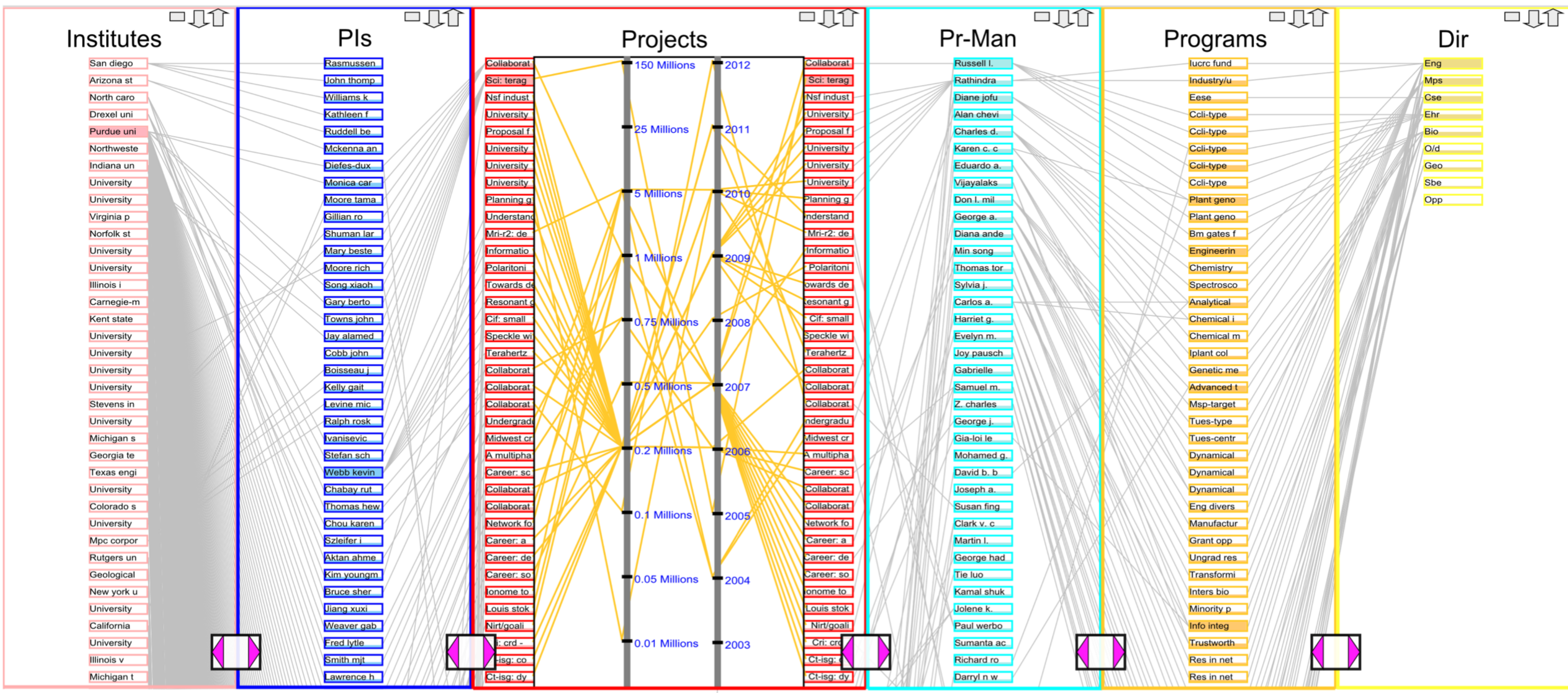
Which of the techniques you know can you use to construct this layout?



Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm, Bothorel et al 2013

1-dimensional representaiton: linear

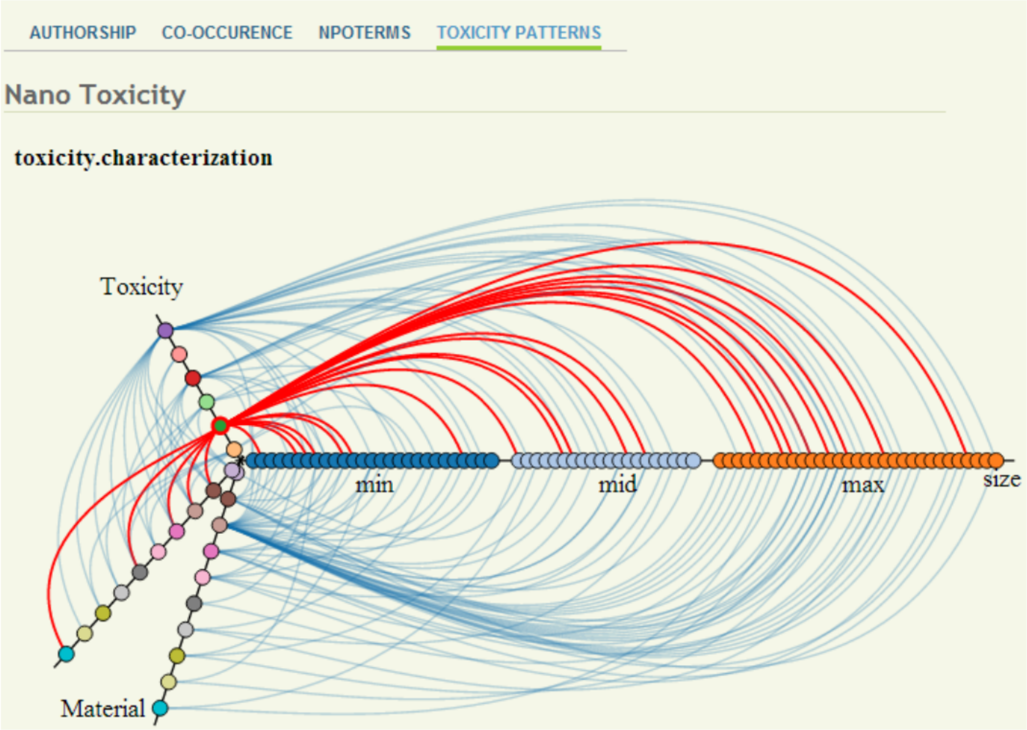
- multimodal NSF funding data consisting of Institutions, PIs (and Co-PIs), Projects, program managers (Pr-Man), NSF programs (Programs), and NSF directorates (Dir)
- remind parallel coordinate plots



Visual Analytics for Multimodal Social Network Analysis: A Design Study with Social Scientists, Ghani et al, 2013

1-dimensional representaiton: linear

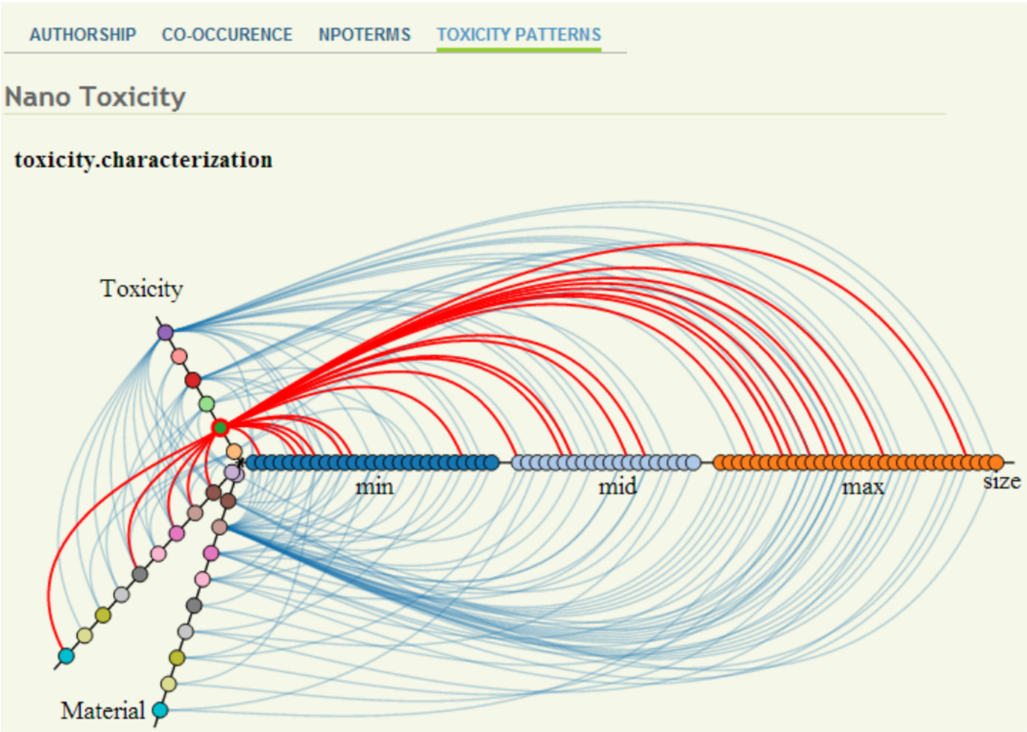
- Hive plot: axes are arranged radially
- investigation among nano-toxicity type, nanomaterial and particle size



A user-centred approach to information visualisation in nano-health, Yang et al, 2016

1-dimensional representaiton: linear

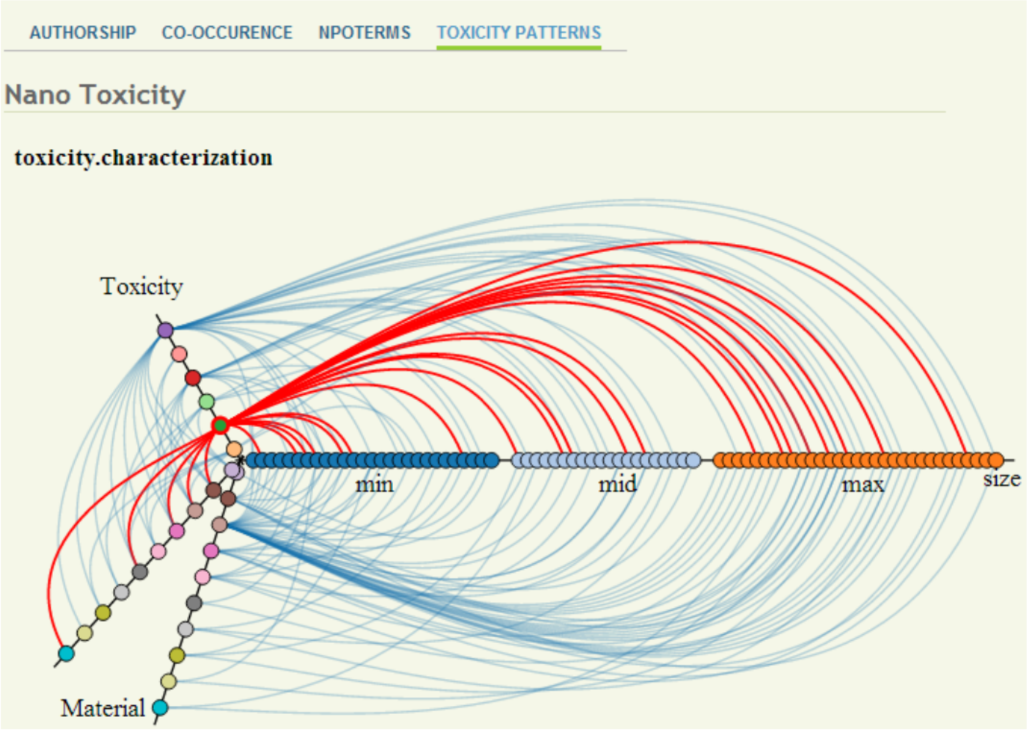
- Hive plot: axes are arranged radially
- investigation among nano-toxicity type, nanomaterial and particle size
- edges are displayed only between adjacent layers



A user-centred approach to information visualisation in nano-health, Yang et al, 2016

1-dimensional representaiton: linear

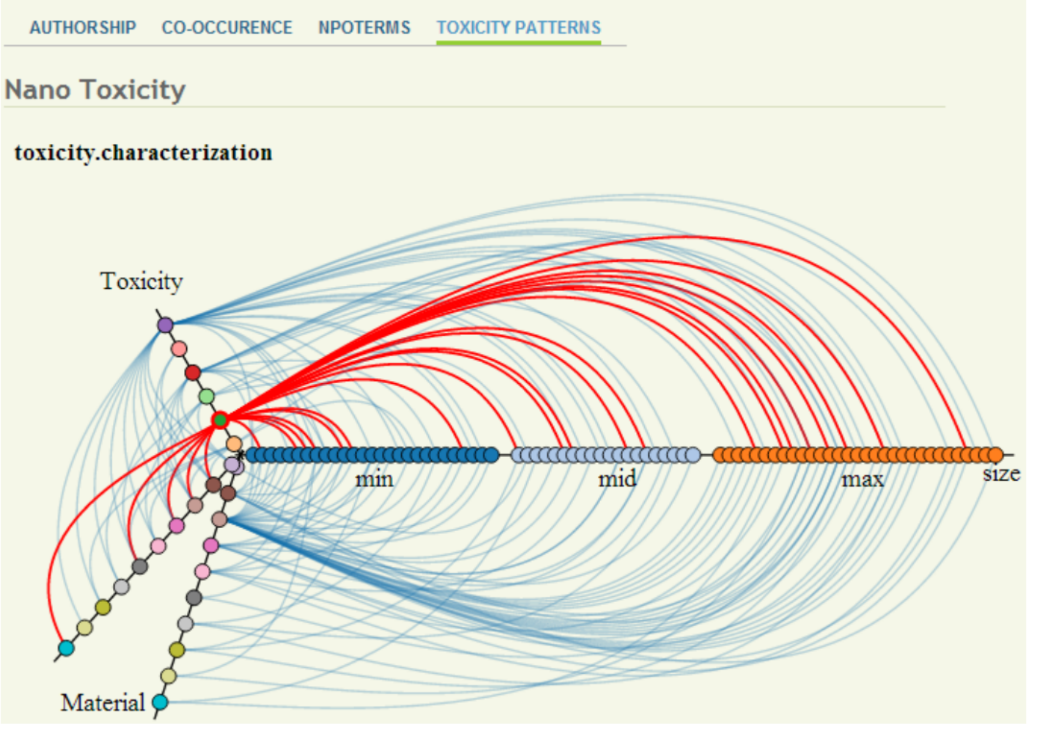
- Hive plot: axes are arranged radially
- investigation among nano-toxicity type, nanomaterial and particle size
- edges are displayed only between adjacent layers
- nodes are arranged based on graph metric (e.g. degree)



A user-centred approach to information visualisation in nano-health, Yang et al, 2016

1-dimensional representaiton: linear

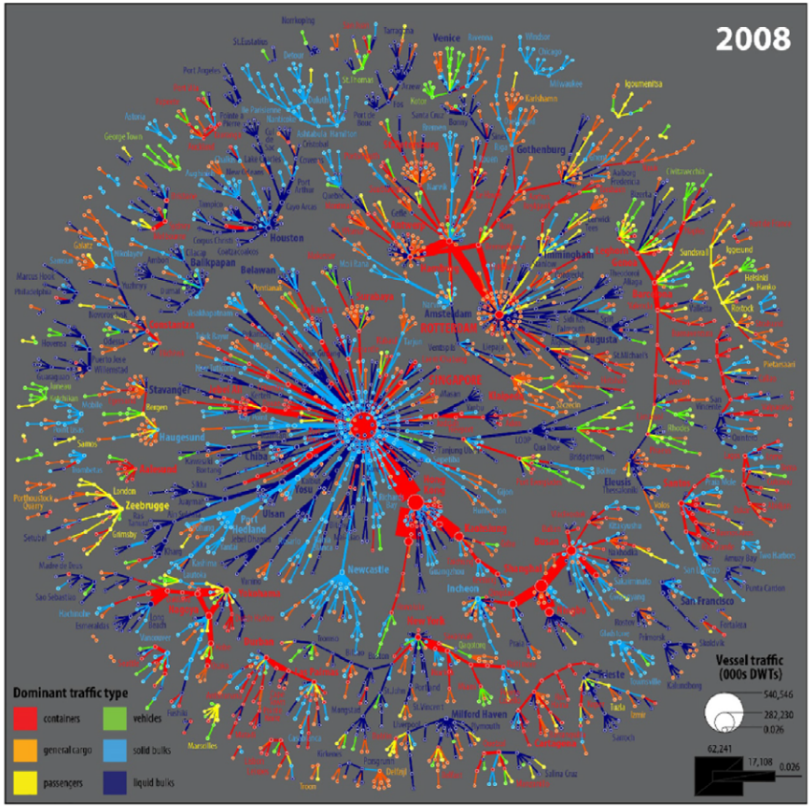
- Hive plot: axes are arranged radially
- investigation among nano-toxicity type, nanomaterial and particle size
- edges are displayed only between adjacent layers
- nodes are arranged based on graph metric (e.g. degree)
- reduce clutter using layer dublication



A user-centred approach to information visualisation in nano-health, Yang et al, 2016

2-dimensional representaiton: color

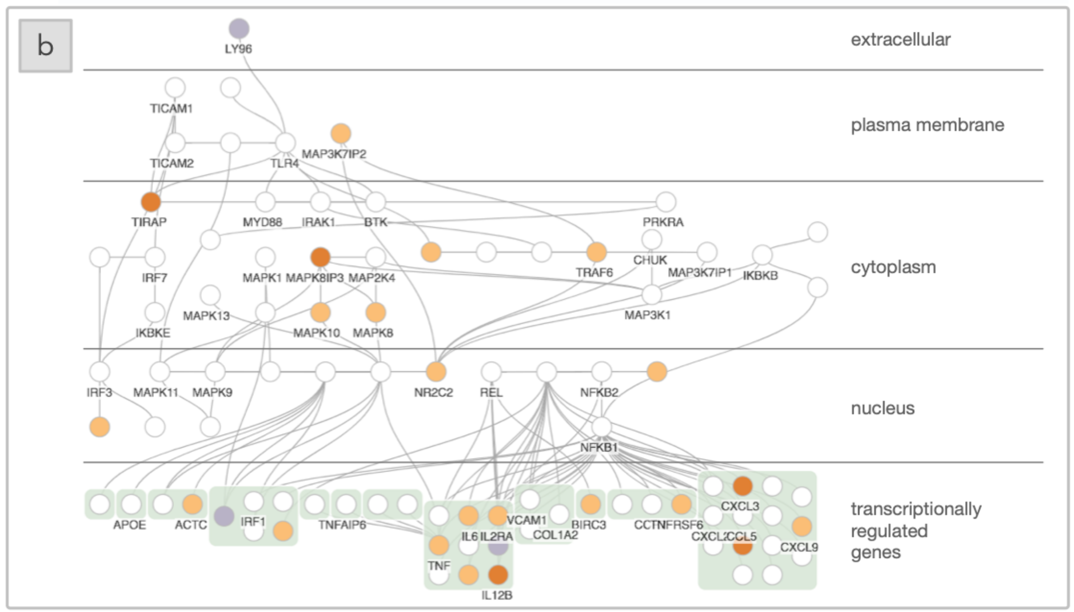
- flow of maritime traffic: nodes represent ports and different edge colours represent different modes of shipping



Multilayer dynamics of complex spatial networks: The case of global maritime flows, Ducuet, 2017

2-dimensional representaiton: separation

- use constrained layouts to separate the nodes of different layers spatially

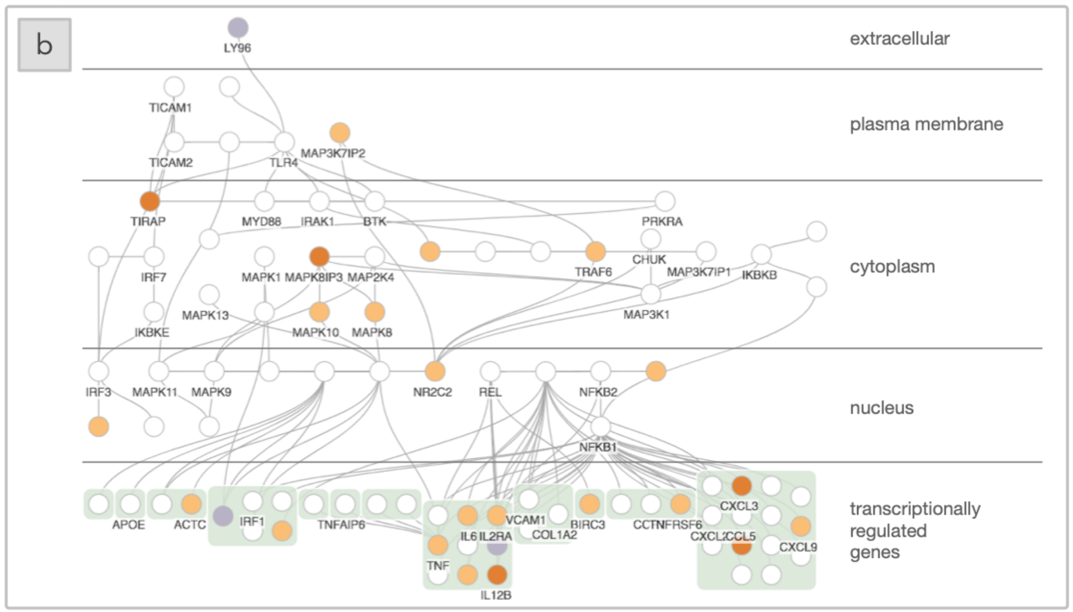


nodes – physical compounds in a cell; that are separated by physical membranes, creating compartments defining their subcellular location – layeredge; edges interactions among nodes

SetCoLa: High-Level Constraints for Graph Layout, Hoffswell et al, 2018

2-dimensional representaiton: separation

- use constrained layouts to separate the nodes of different layers spatially
- not intended for multilayer networks but can be used for them

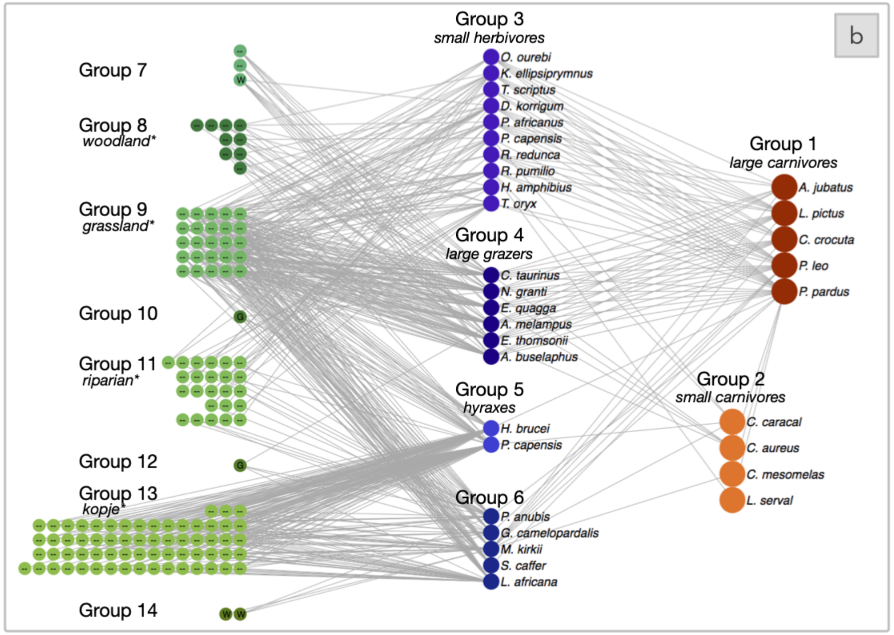


nodes – physical compounds in a cell; that are separated by physical membranes, creating compartments defining their subcellular location – layeredge; edges interactions among nodes

SetCoLa: High-Level Constraints for Graph Layout, Hoffswell et al, 2018

2-dimensional representaiton: separation

- use constrained layouts to separate the nodes of different layers spatially
- not intended for multilayer networks but can be used for them

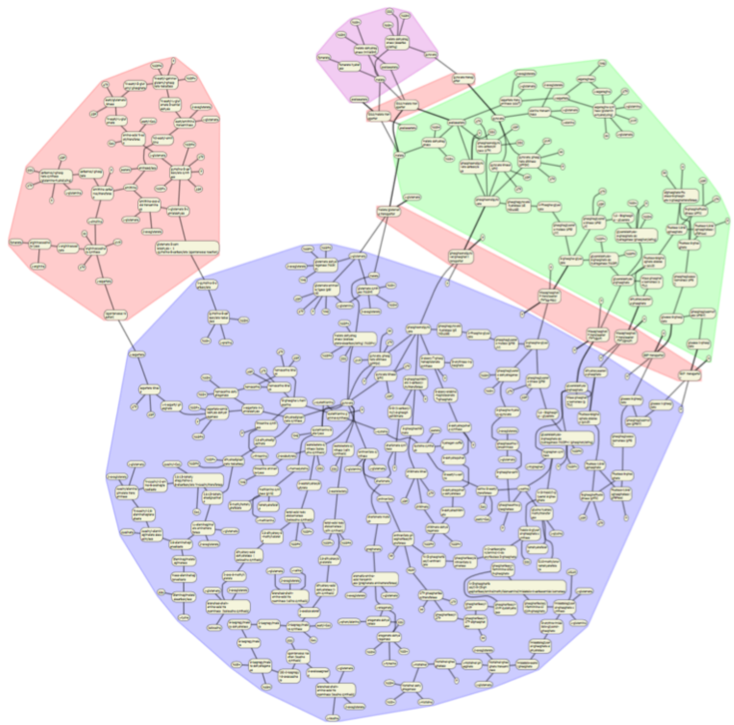


161 plants, herbivores, and carnivores with 592 links between entities – feeding links, groups – clustering, layers – trophic hierarchy

SetCoLa: High-Level Constraints for Graph Layout, Hoffswell et al, 2018

2-dimensional representaiton: separation

- use constrained layouts to separate the nodes of different layers spatially
- not intended for multilayer networks but can be used for them

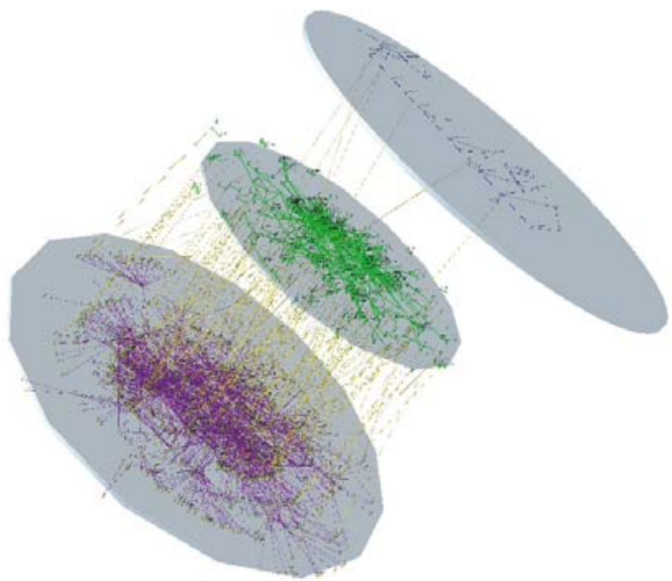


Biological pathways: nodes – proteins, edges–interactions. Rather visualization of clusters, but can be used to show layers too.

Scalable, Versatile and Simple Constrained Graph Layout, Dwyer 2009

2.5-dimensional representaition

- each layer is drawn on a plane and planes are stacked in 3D parallel to each other
- use 2D layout algorithms for a single layer
- same node can appear on many layers – similar positions are desired. Same for reducing edge clutter.

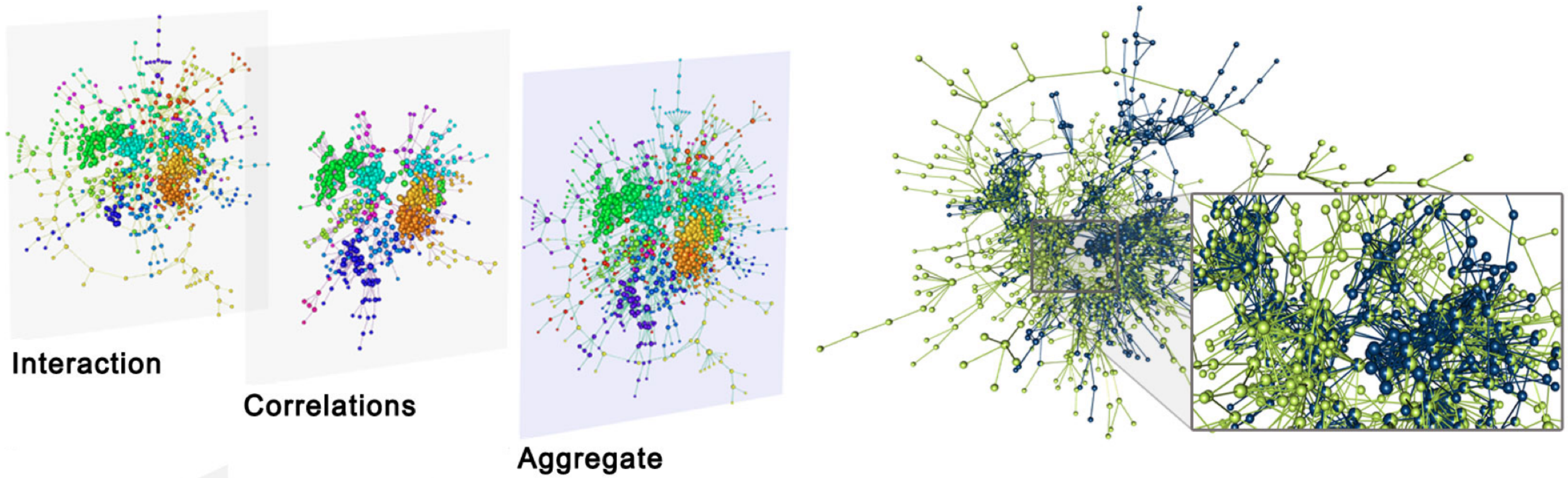


metabolic network, protein interaction networks and gene regulatory network; inter-layer edges: proteins are the result of gene expression, special proteins known as enzymes help transforming metabolites to another.

Visual Analysis of Overlapping Biological Networks, Fung et al, 2009

2.5-dimensional representaition

- same node can appear across layers and lie at the same position
- aggregated layer is possible

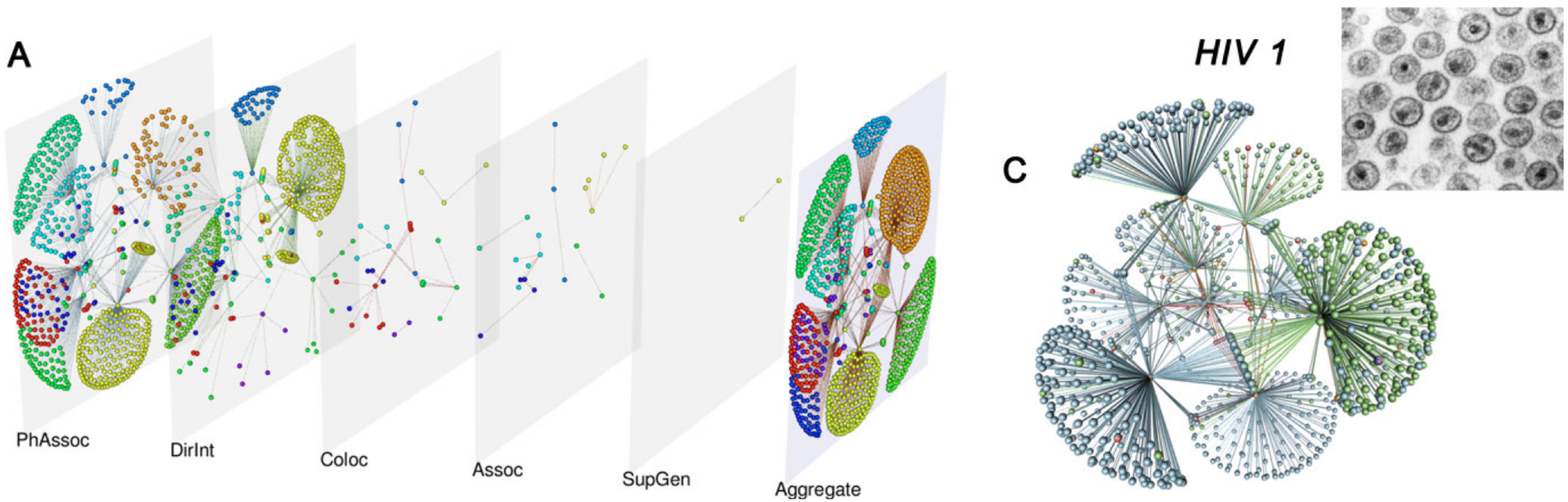


- first layer: interaction of genes in Saccharomyces cerevisiae; second layer: genes with similar interaction profiles are connected to each other; third layer: aggregated network
- right - edge colors represent layers

MuxViz: A Tool for Multilayer Analysis and Visualization of Networks, De Do,enico et al, 2015.

2.5-dimensional representaition

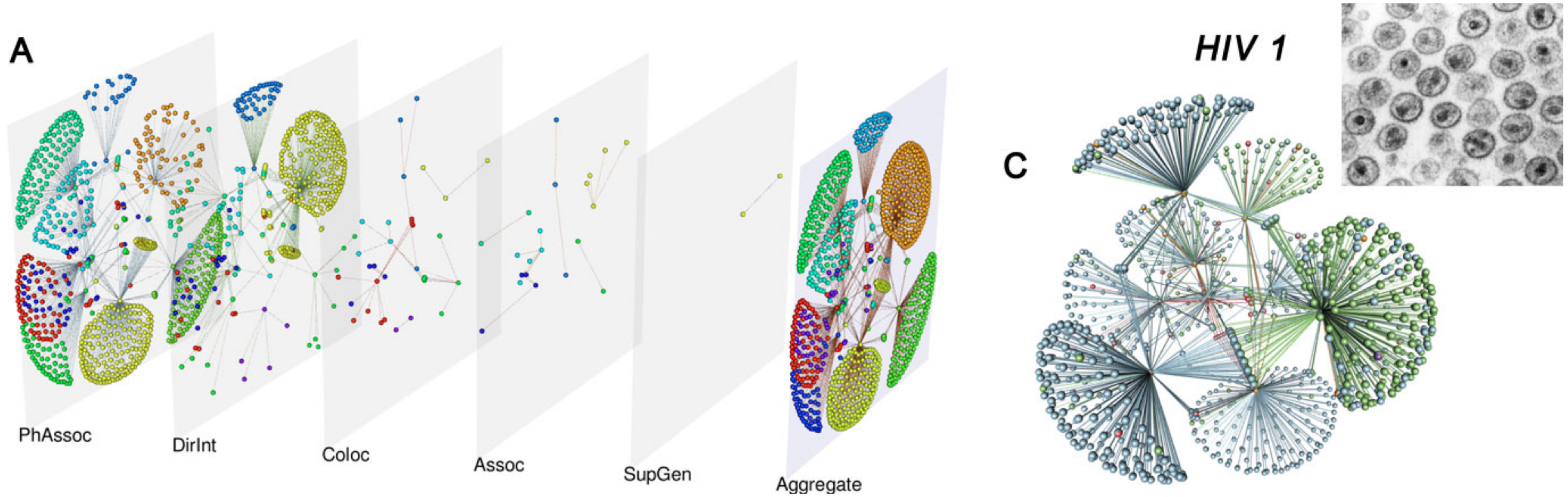
- same node can appear across layers and lie at the same position
- aggregated layer is possible



- Multilayer analysis of HIV-1 genetic interaction network

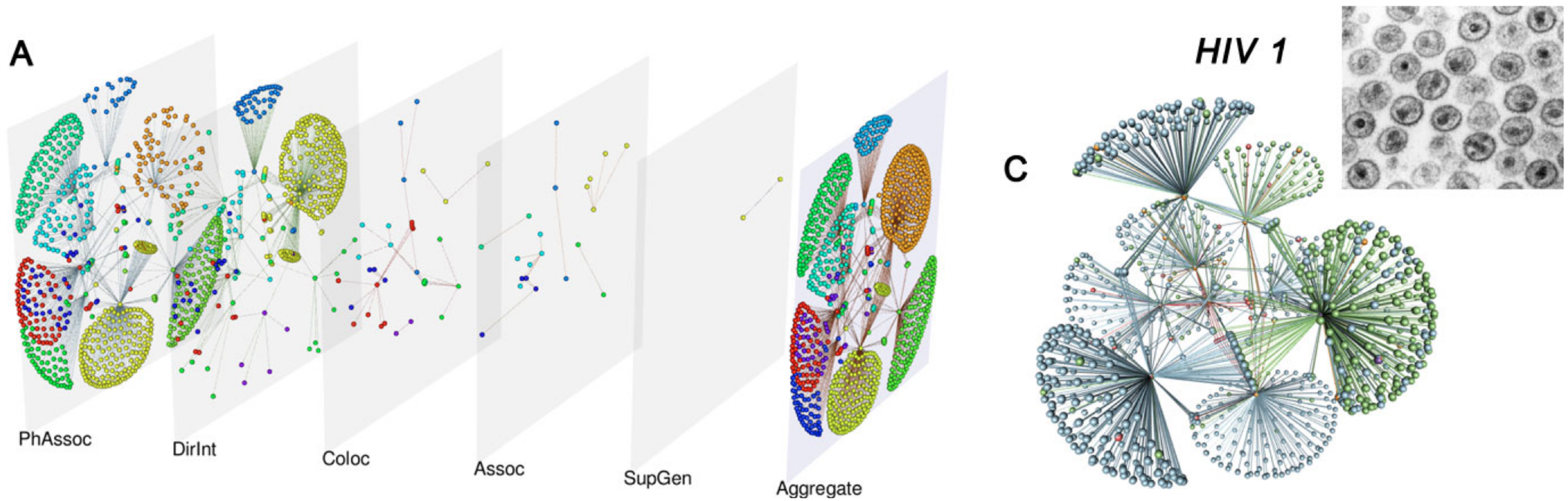
MuxViz: A Tool for Multilayer Analysis and Visualization of Networks, De Do,enico et al, 2015.

Generating 2.5D representations



- If it is essential that same node has exactly the same position over the layers

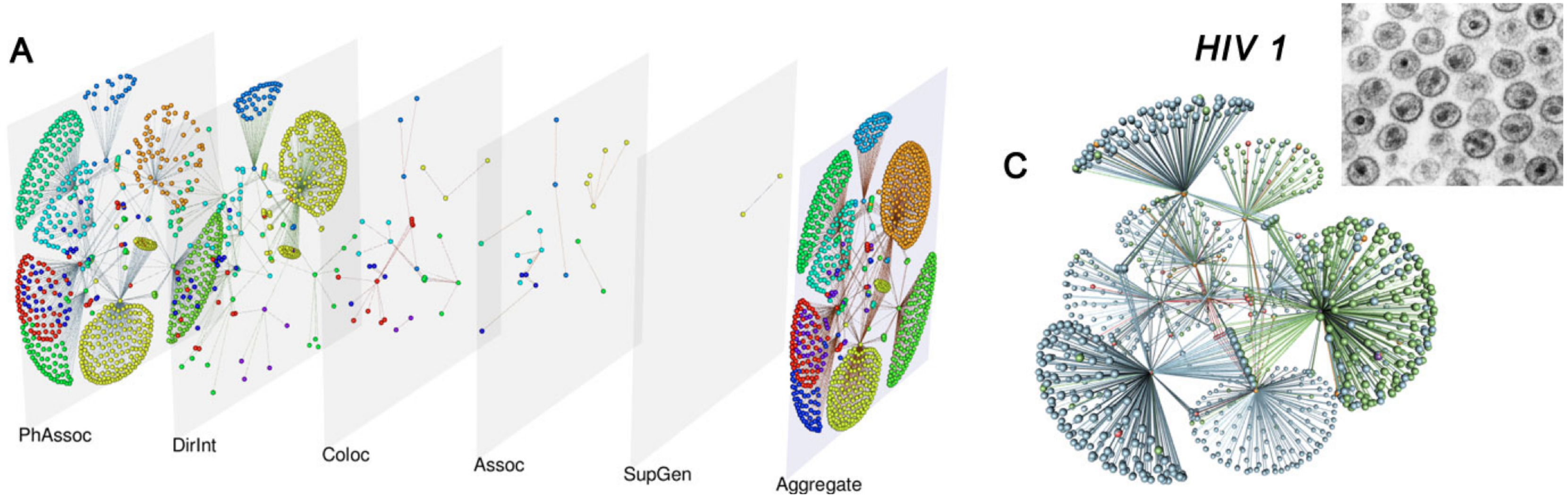
Generating 2.5D representations



- If it is essential that same node has exactly the same position over the layers

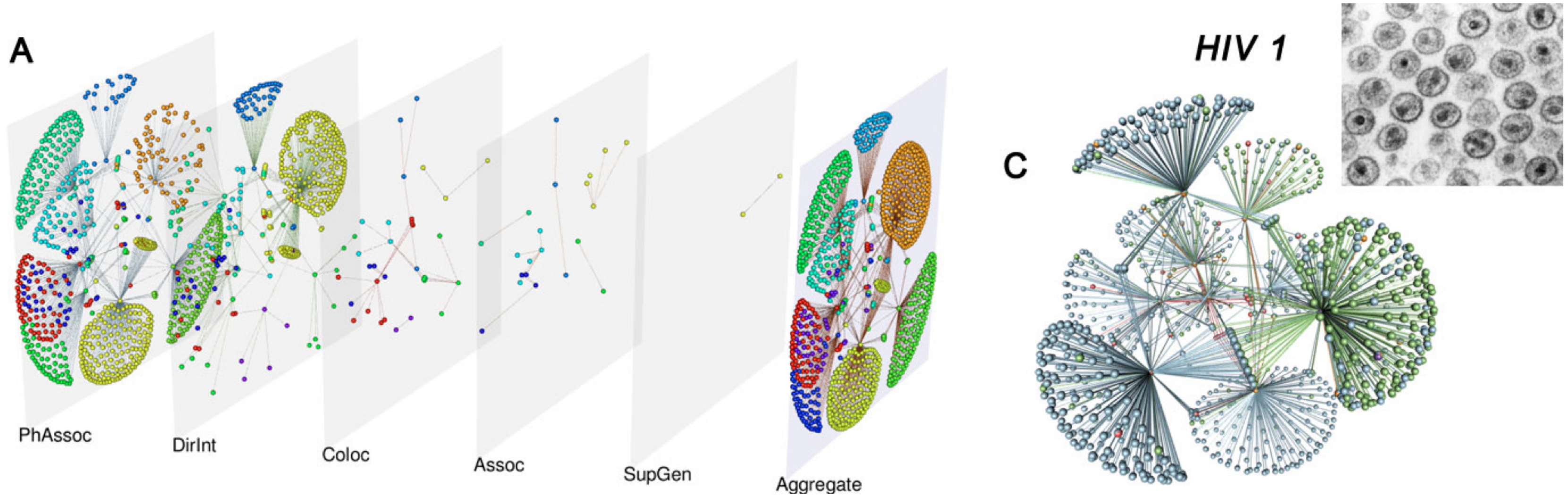
How to construct this representation?

Generating 2.5D representations



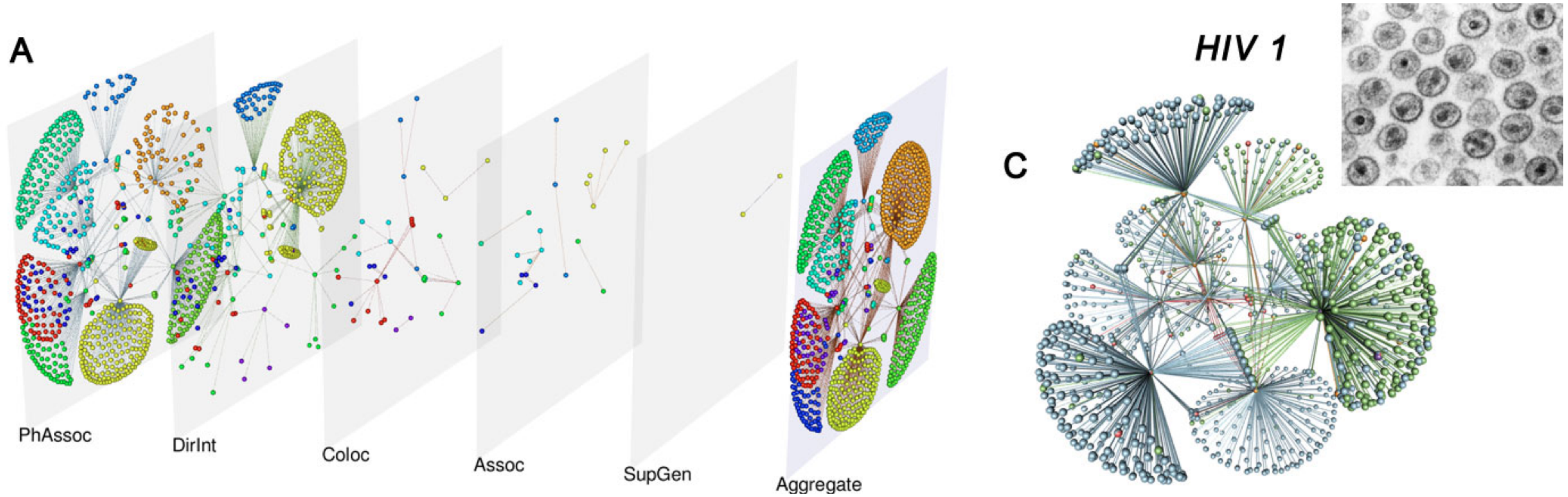
- If it is essential that same node has exactly the same position over the layers
- Aggregate the graphs over the layers into a single graph $G = (V, E)$

Generating 2.5D representations



- If it is essential that same node has exactly the same position over the layers
- Aggregate the graphs over the layers into a single graph $G = (V, E)$
- Layout G with a favorite layout method – aggregated layer

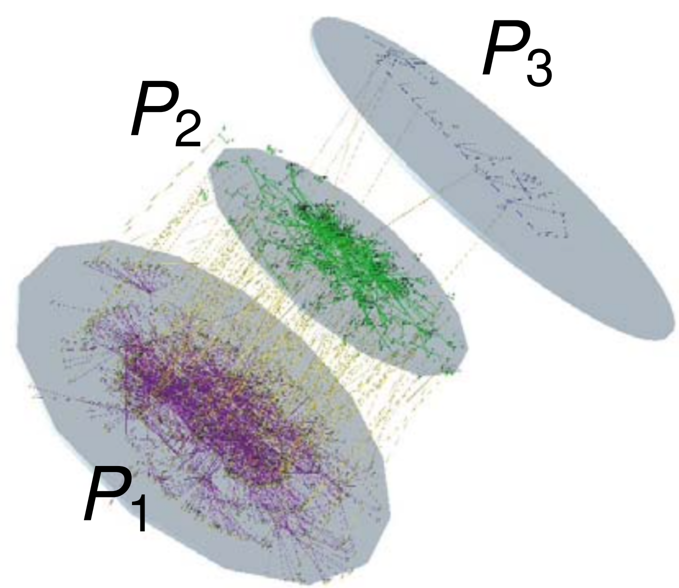
Generating 2.5D representations



- If it is essential that same node has exactly the same position over the layers
- Aggregate the graphs over the layers into a single graph $G = (V, E)$
- Layout G with a favorite layout method – aggregated layer
- Use coordinates of the nodes of G to construct the layouts of the rest layers

Generating 2.5D representations Fung et al, 2009

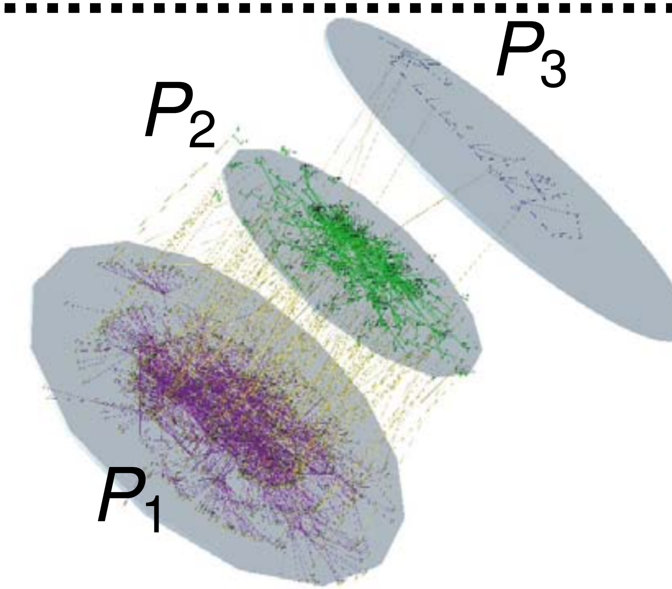
- If it is not essential that same node has exactly the same position over the layers, or there are not many identical nodes over the layers



Generating 2.5D representations Fung et al, 2009

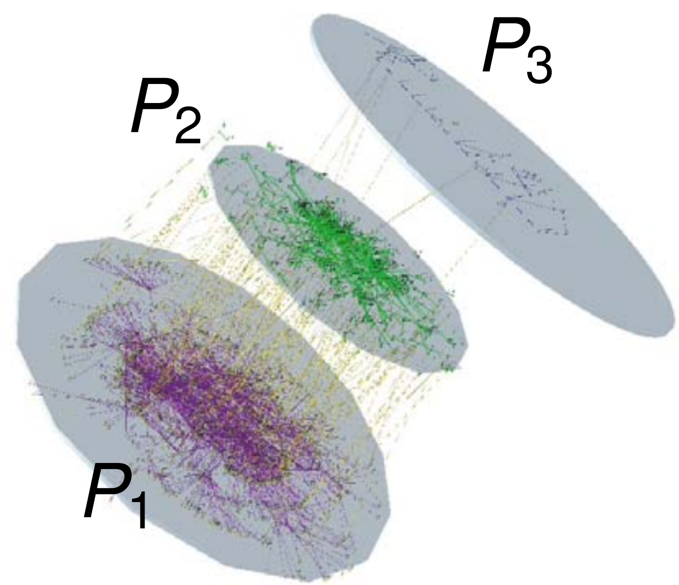
- If it is not essential that same node has exactly the same position over the layers, or there are not many identical nodes over the layers

Same method as before is possible, but how to use the flexibility in node position in order to construct better layout?



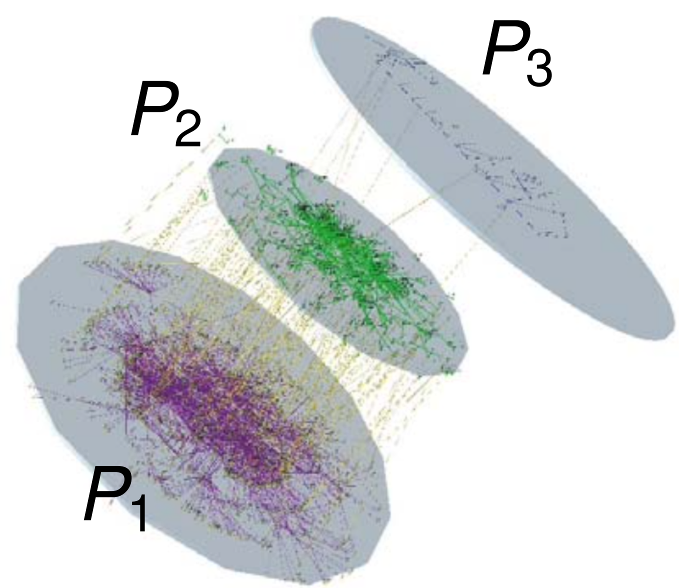
Generating 2.5D representations Fung et al, 2009

- If it is not essential that same node has exactly the same position over the layers, or there are not many identical nodes over the layers
- Assume we have 3 layers ℓ_1, ℓ_2, ℓ_3, let G_i be graph induced by $\{(v, \ell_i) \in V_m : v \in V\}$, $i = 1, 2, 3$



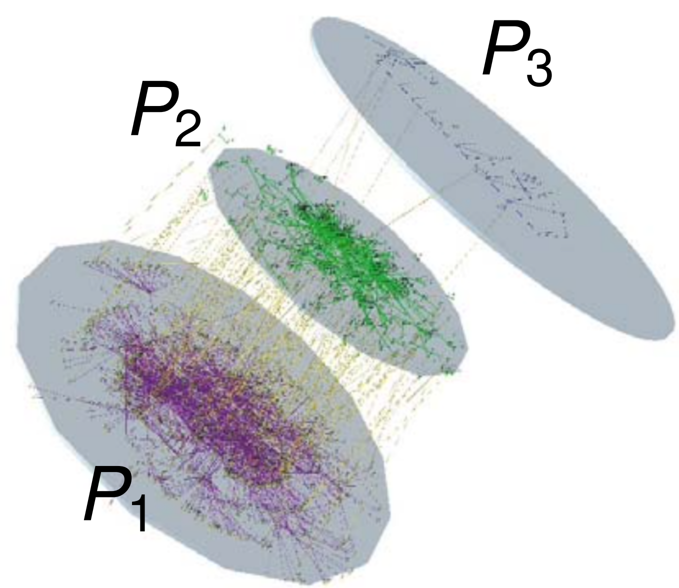
Generating 2.5D representations Fung et al, 2009

- If it is not essential that same node has exactly the same position over the layers, or there are not many identical nodes over the layers
- Assume we have 3 layers ℓ_1, ℓ_2, ℓ_3, let G_i be graph induced by $\{(v, \ell_i) \in V_m : v \in V\}$, $i = 1, 2, 3$
- Draw G_1 and G_3 on planes P_1 and P_3



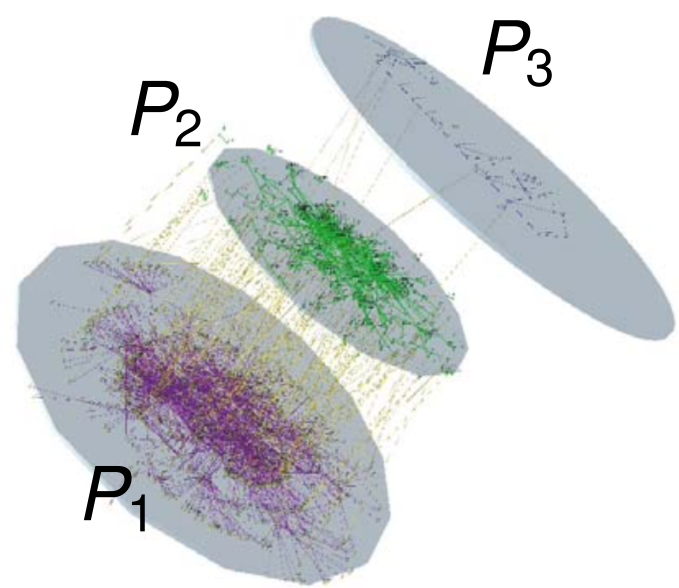
Generating 2.5D representations Fung et al, 2009

- If it is not essential that same node has exactly the same position over the layers, or there are not many identical nodes over the layers
- Assume we have 3 layers ℓ_1, ℓ_2, ℓ_3, let G_i be graph induced by $\{(v, \ell_i) \in V_m : v \in V\}$, $i = 1, 2, 3$
- Draw G_1 and G_3 on planes P_1 and P_3
- Assign initial position to each node $(v, \ell_2) \in G_2$ using barycenter of (v, ℓ_1) and (v, ℓ_3)



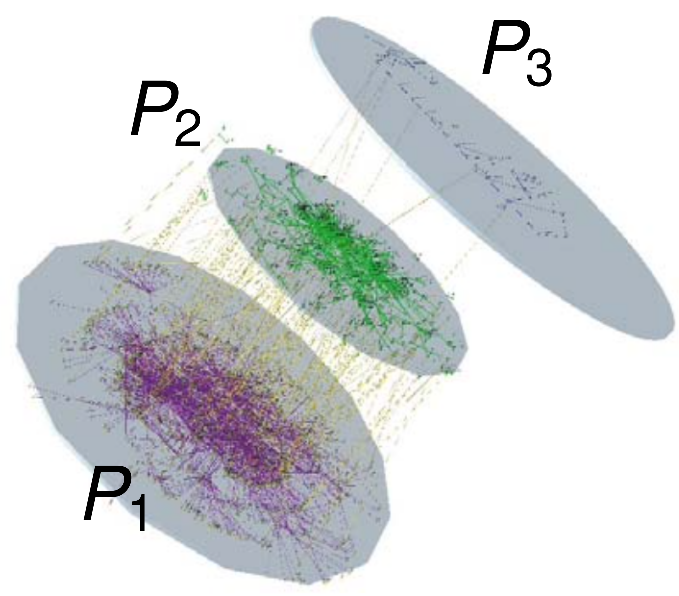
Generating 2.5D representations Fung et al, 2009

- If it is not essential that same node has exactly the same position over the layers, or there are not many identical nodes over the layers
- Assume we have 3 layers ℓ_1, ℓ_2, ℓ_3, let G_i be graph induced by $\{(v, \ell_i) \in V_m : v \in V\}$, $i = 1, 2, 3$
- Draw G_1 and G_3 on planes P_1 and P_3
- Assign initial position to each node $(v, \ell_2) \in G_2$ using barycenter of (v, ℓ_1) and (v, ℓ_3)
- Model inter-layer edges as zero-length spring (attraction only)

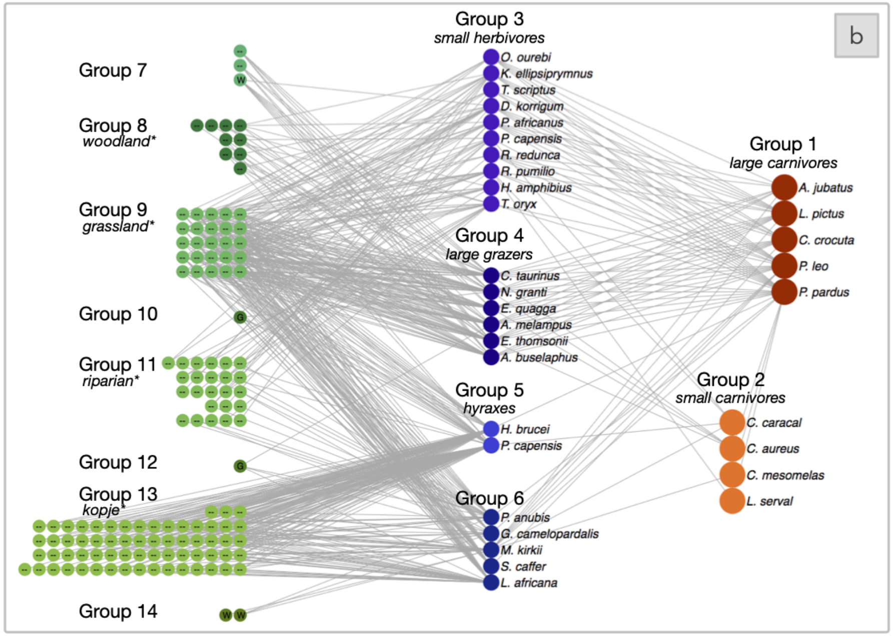


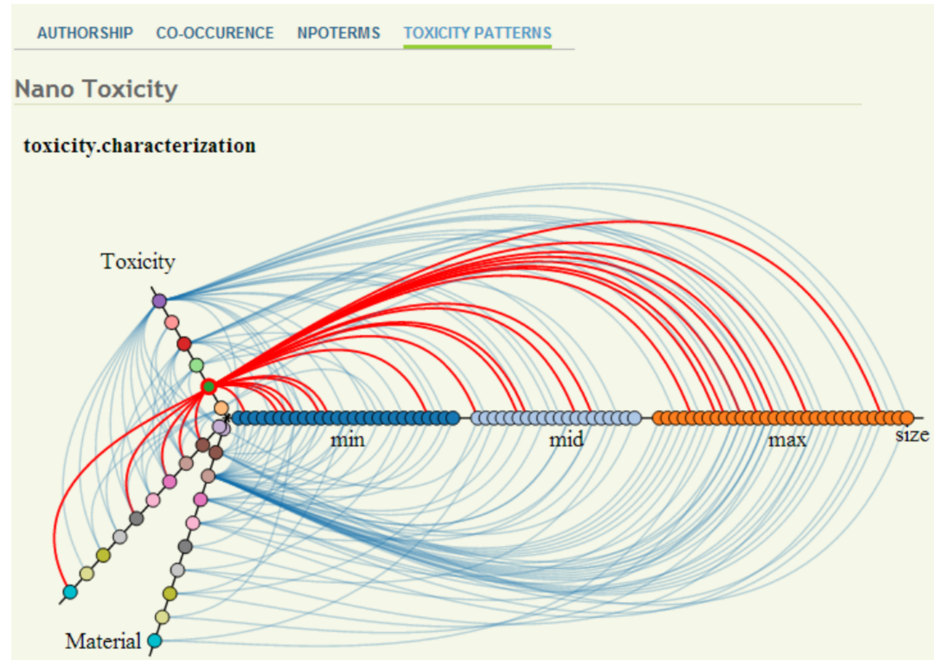
Generating 2.5D representations Fung et al, 2009

- If it is not essential that same node has exactly the same position over the layers, or there are not many identical nodes over the layers
- Assume we have 3 layers ℓ_1, ℓ_2, ℓ_3, let G_i be graph induced by $\{(v, \ell_i) \in V_m : v \in V\}$, $i = 1, 2, 3$
- Draw G_1 and G_3 on planes P_1 and P_3
- Assign initial position to each node $(v, \ell_2) \in G_2$ using barycenter of (v, ℓ_1) and (v, ℓ_3)
- Model inter-layer edges as zero-length spring (attraction only)
- Draw G_2 and the inter-layer edges using a force directed layout

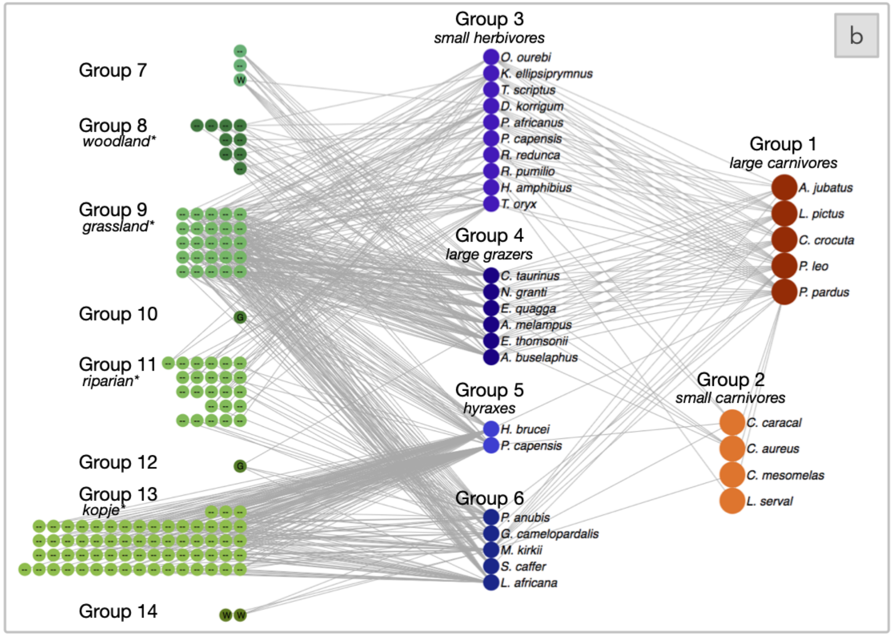


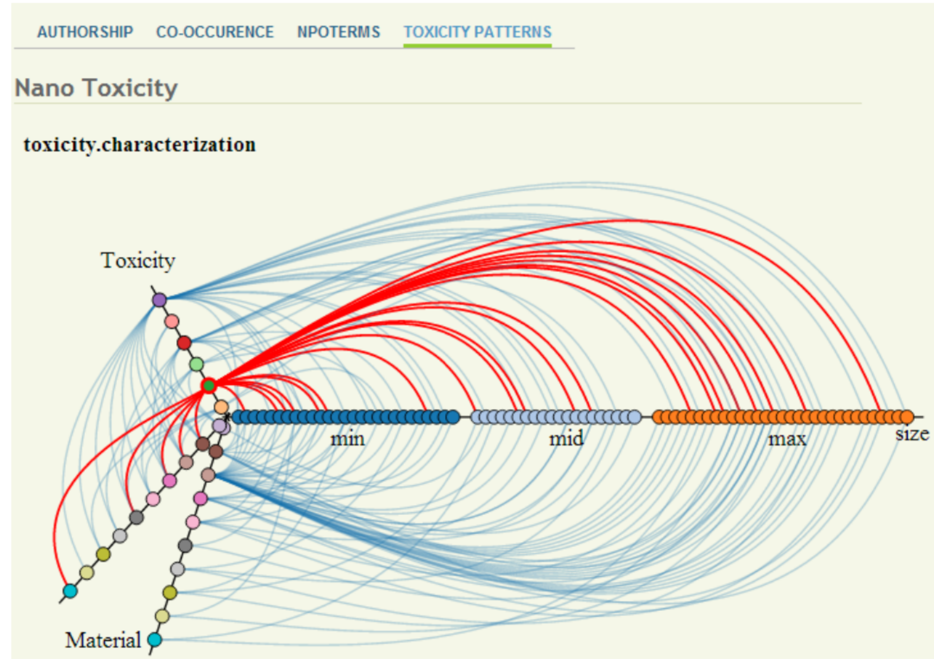
Edge clutter in multilayer visualizations



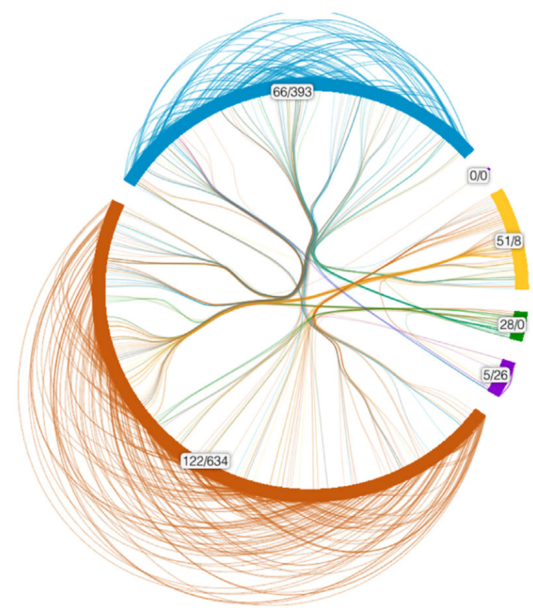


Edge clutter in multilayer visualizations



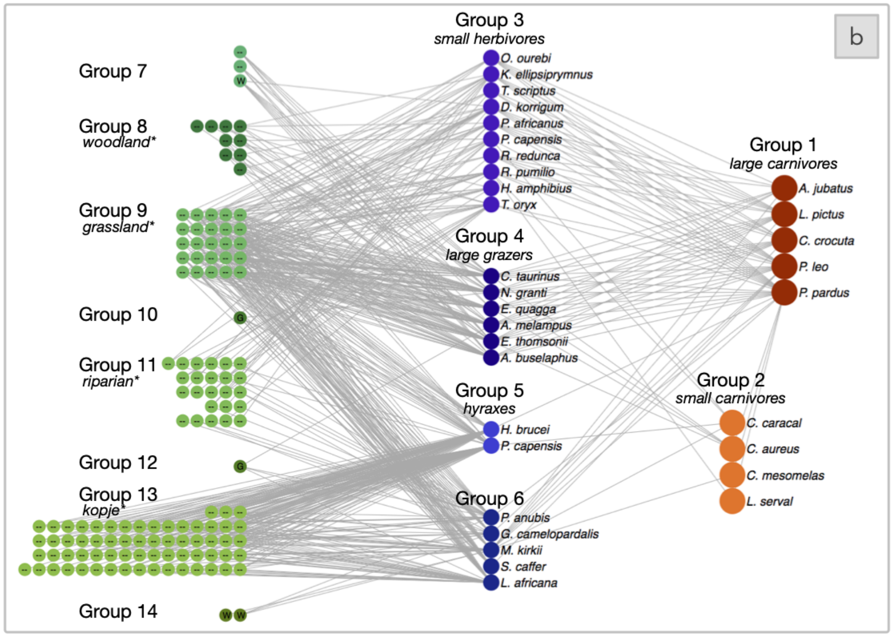


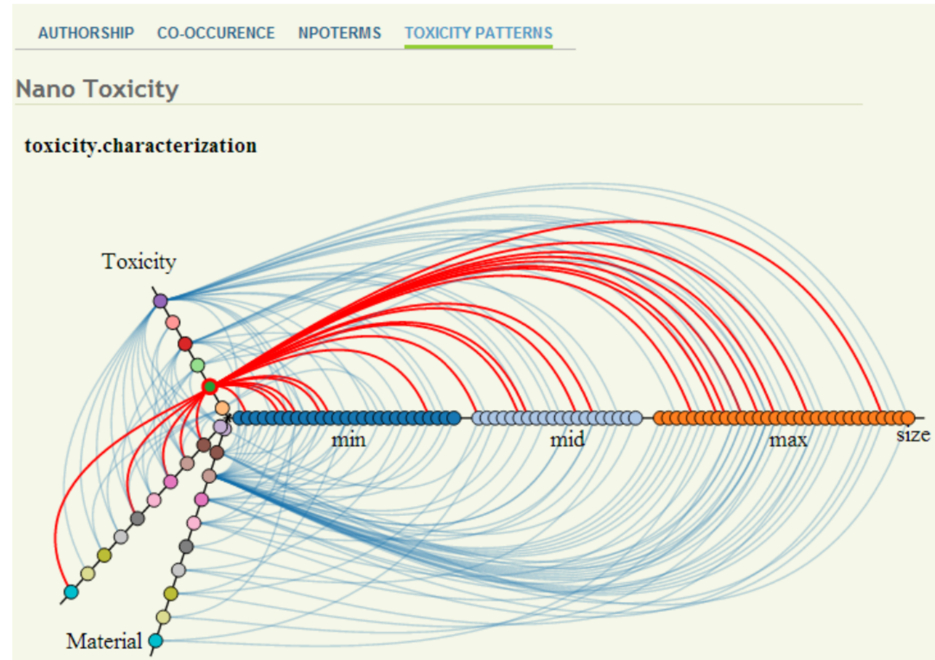
- edge bundling as a method to layout edges in multilayer network visualizations



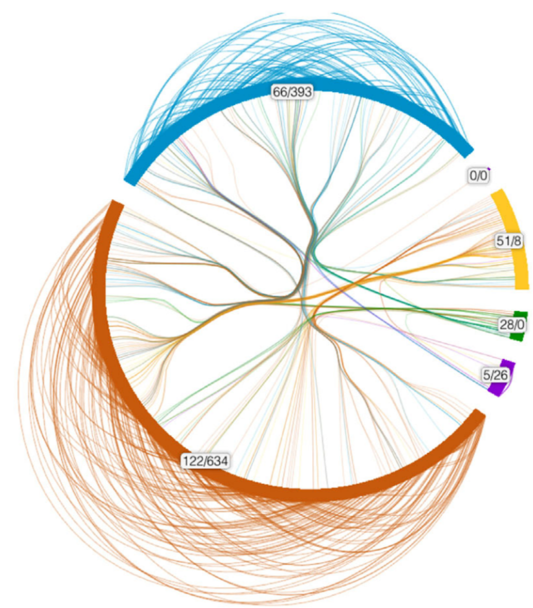
Visualization techniques for categorical analysis of social networks with multiple edge sets, Crnovrsanin et al, 2014

Edge clutter in multilayer visualizations



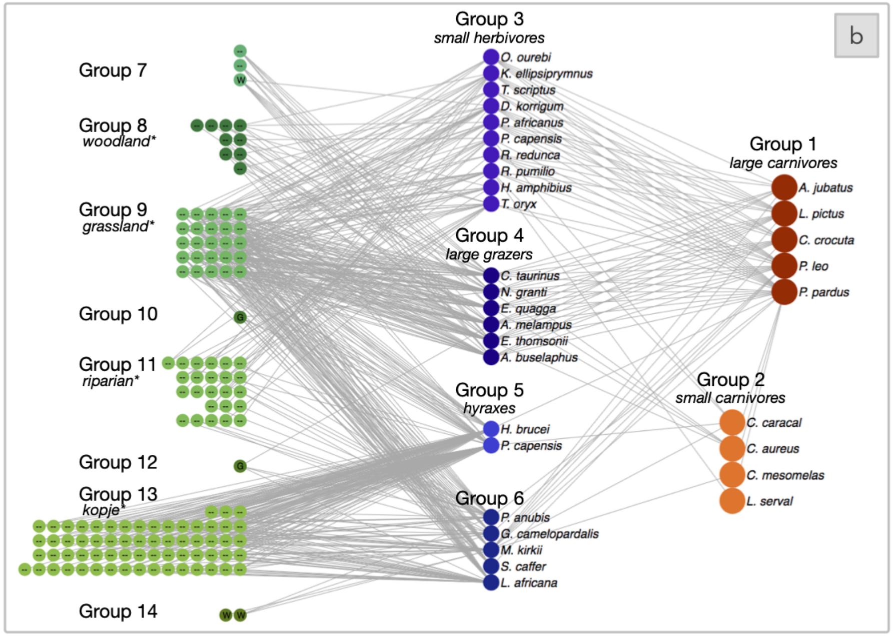


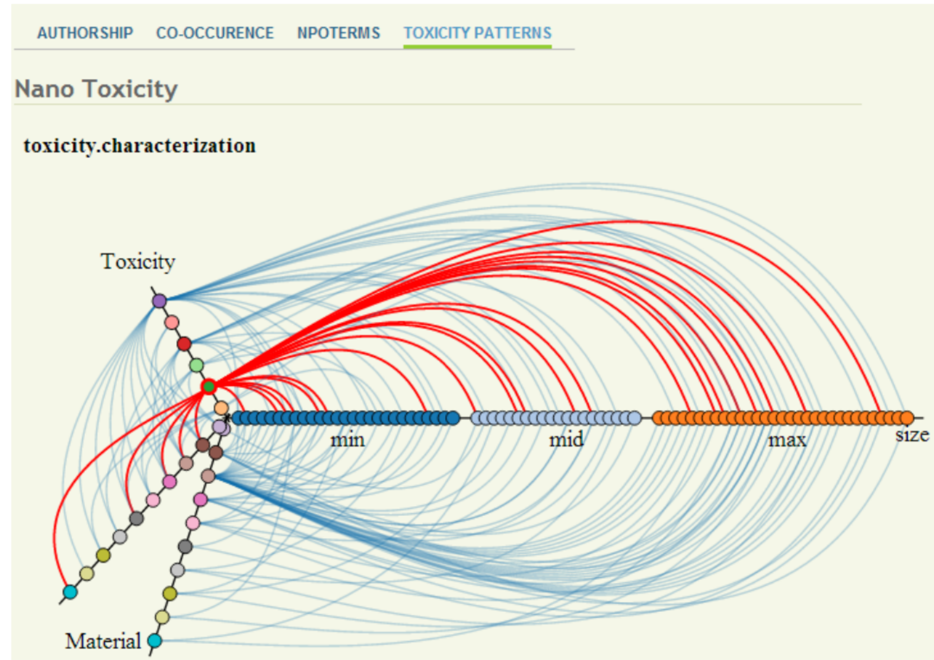
- edge bundling as a method to layout edges in multilayer network visualizations
- bundle only the inter-layer (or intra-layer) edges



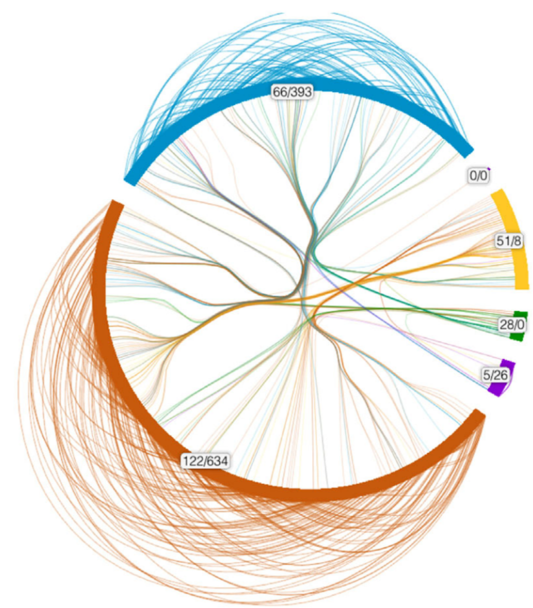
Visualization techniques for categorical analysis of social networks with multiple edge sets, Crnovrsanin et al, 2014

Edge clutter in multilayer visualizations





- edge bundling as a method to layout edges in multilayer network visualizations
- bundle only the inter-layer (or intra-layer) edges
- edge bundling is not specific for multilayer network visualizations

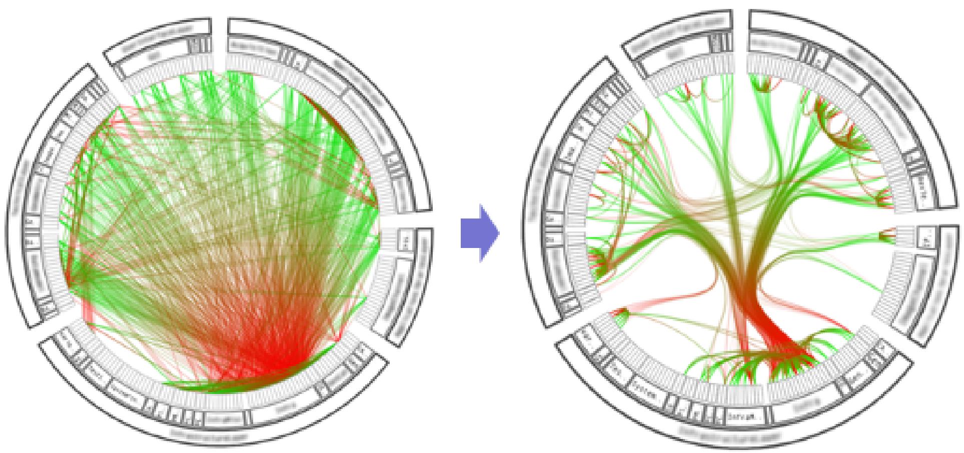


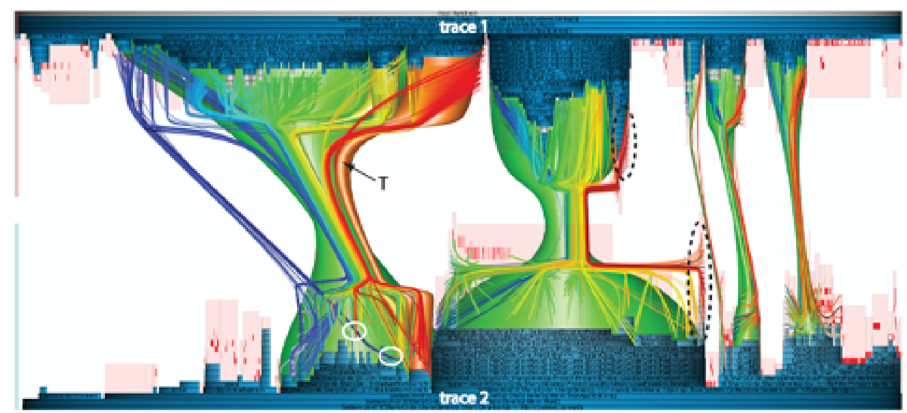
Visualization techniques for categorical analysis of social networks with multiple edge sets, Crnovrsanin et al, 2014

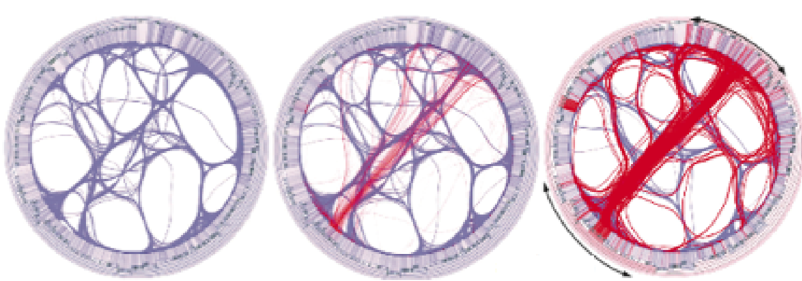
Edge bundling

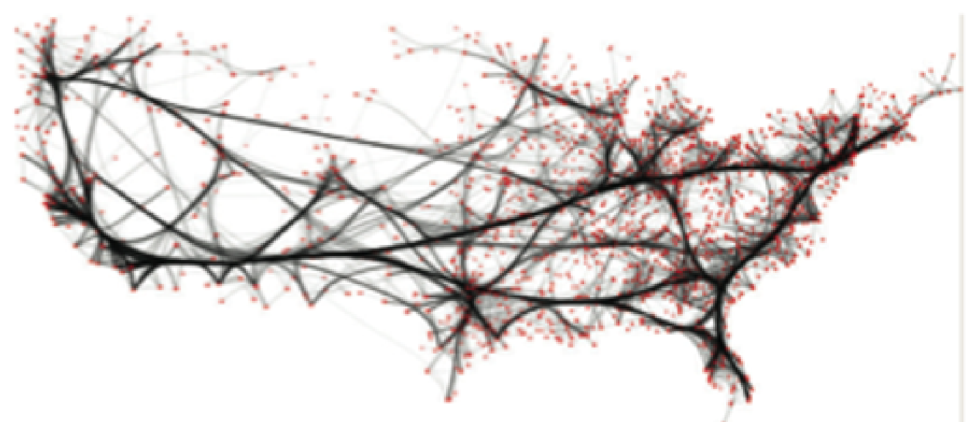
Method for reduction of clutter in a graph layout

“Change the shape of edges by visually bundling them together analogous to the way electrical wires and network cables are merged into bundles...” [Holten, van Wijk, 09]

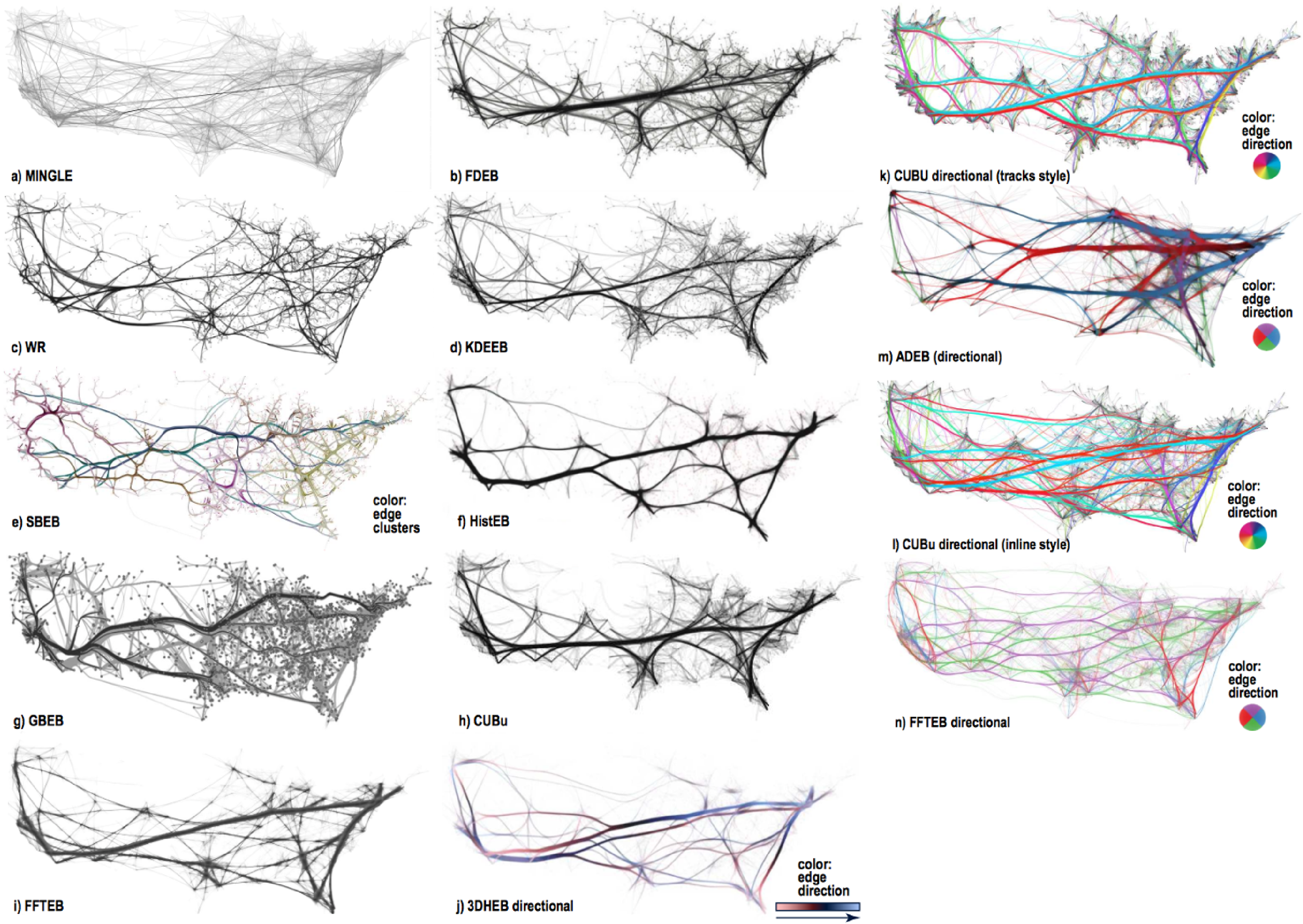




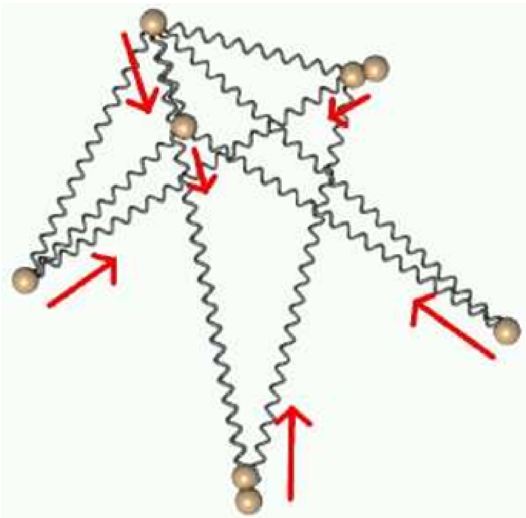




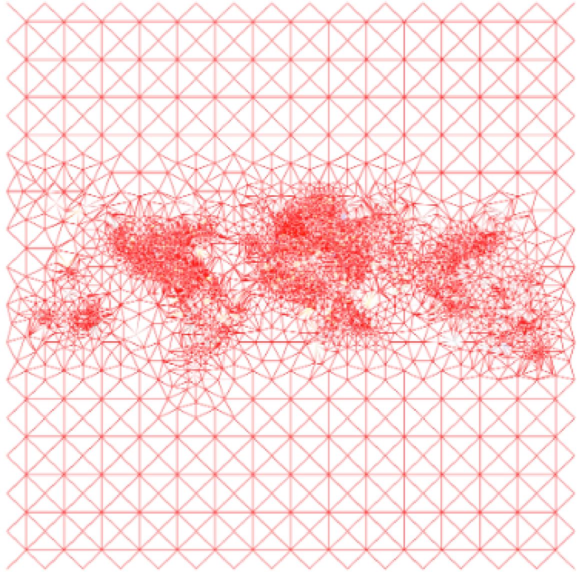
Many methods



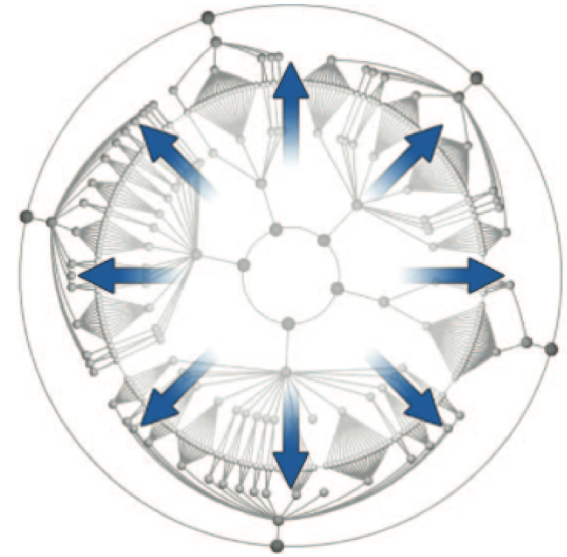
Multiple techniques



spring embedders (FDEB)

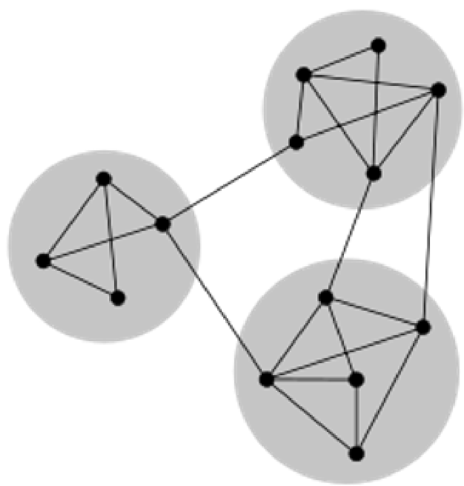


Voronoi/Delaunay diagrams

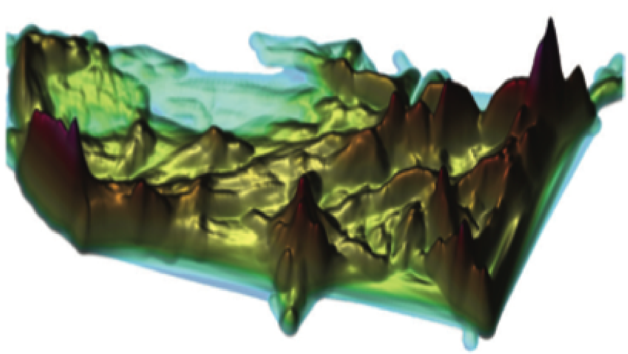


tree layouts & splines

medial axes

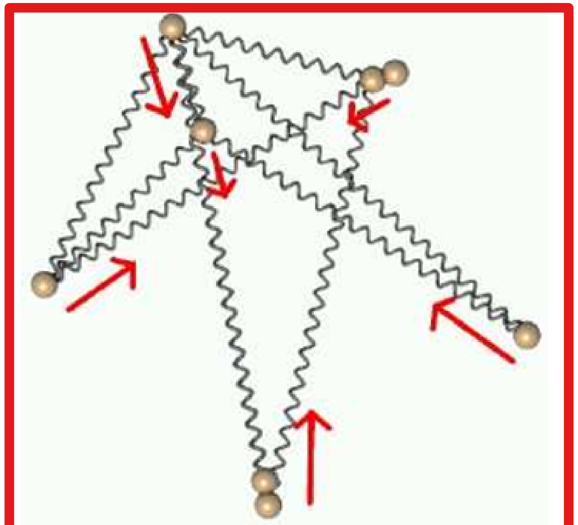


graph clustering

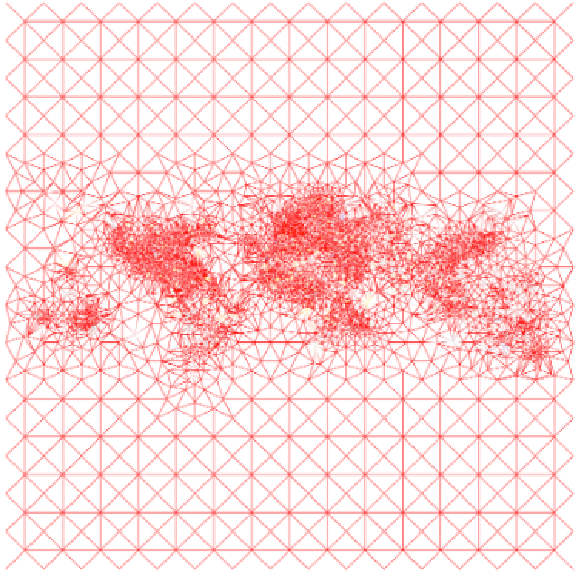


kernel density estimation

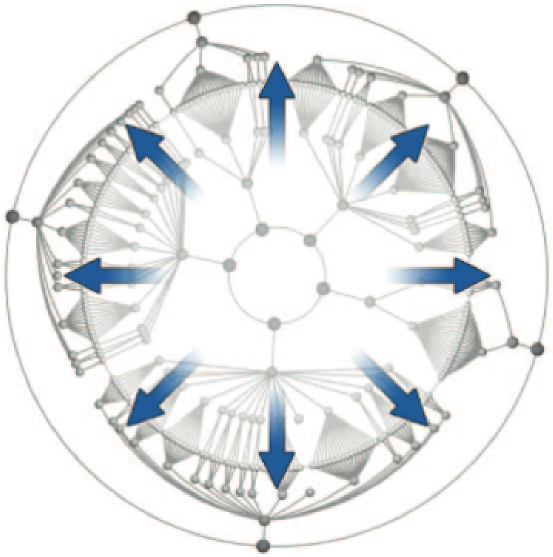
Multiple techniques



spring embedders (FDEB)

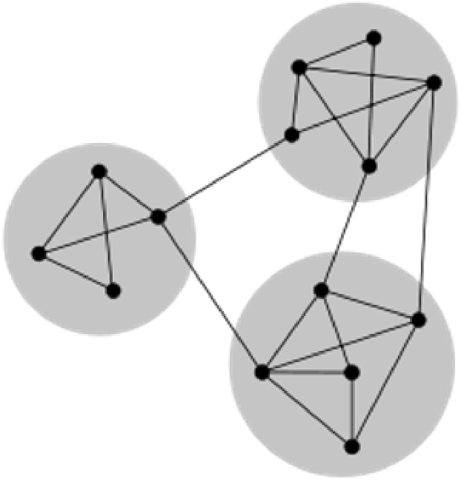


Voronoi/Delaunay diagrams

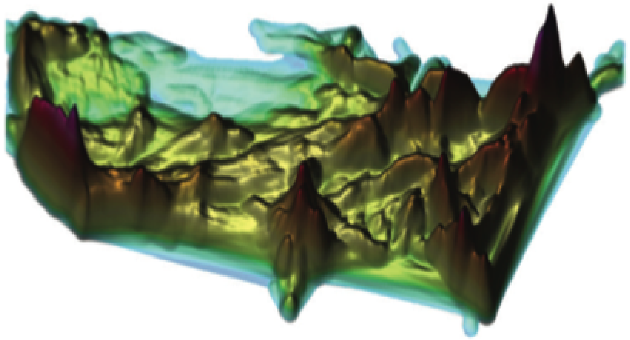


tree layouts & splines

medial axes

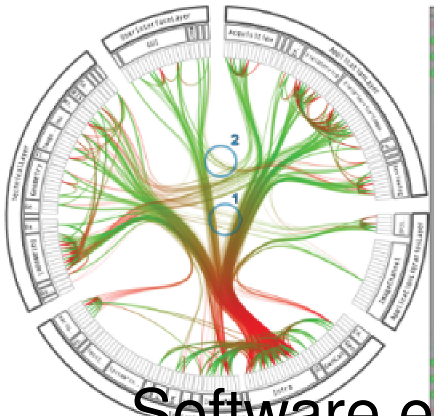
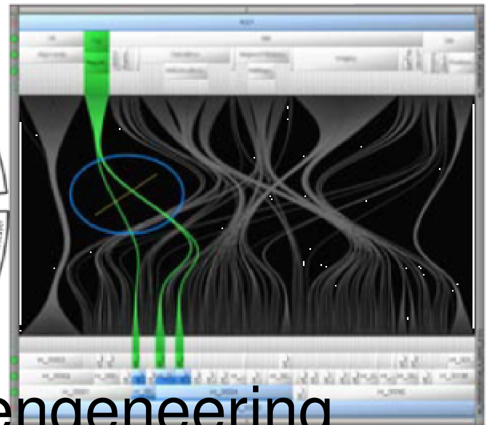


graph clustering



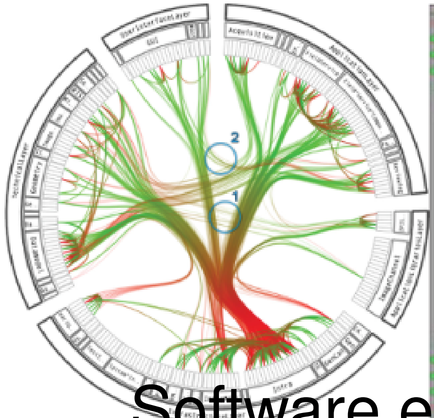
kernel density estimation

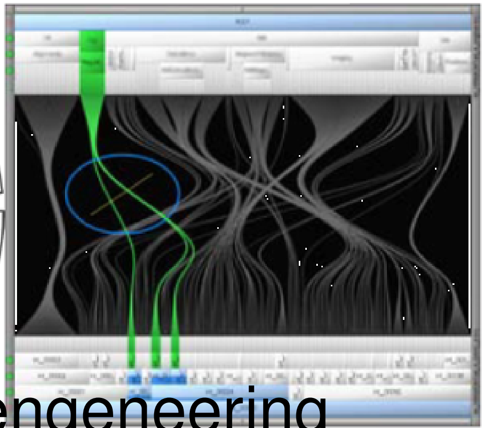
Edge bundling: applications

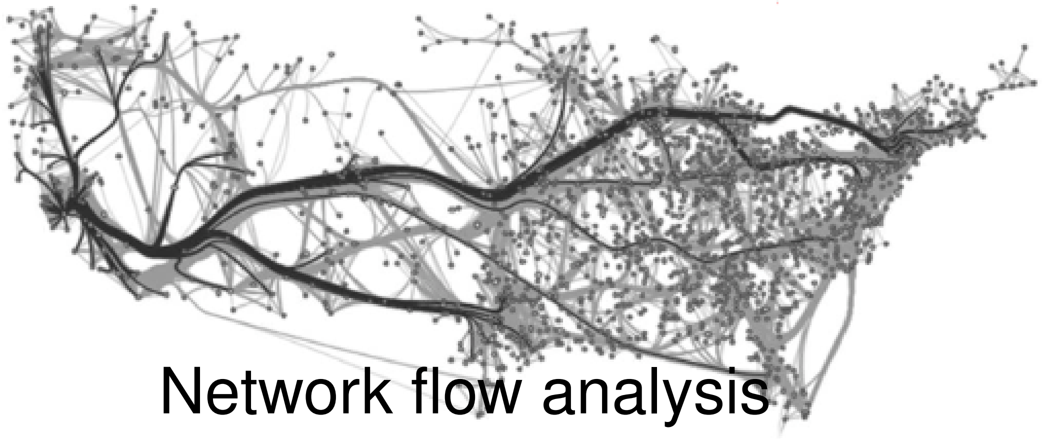
Software engeneering

Edge bundling: applications



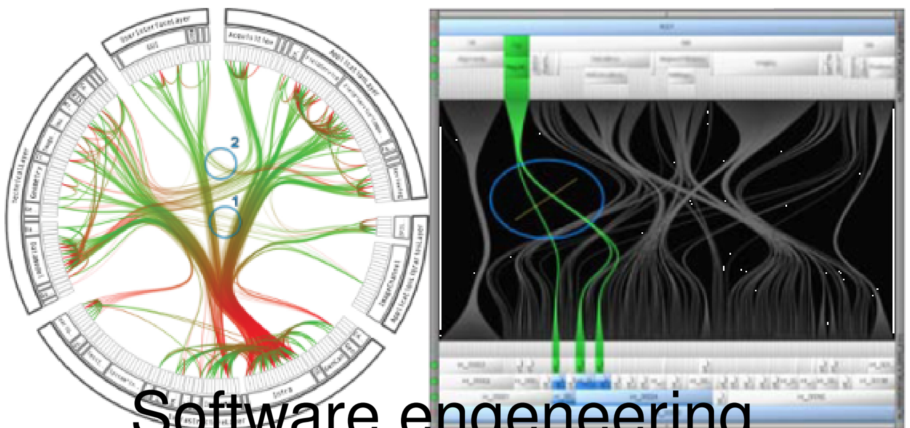


Software engeneering

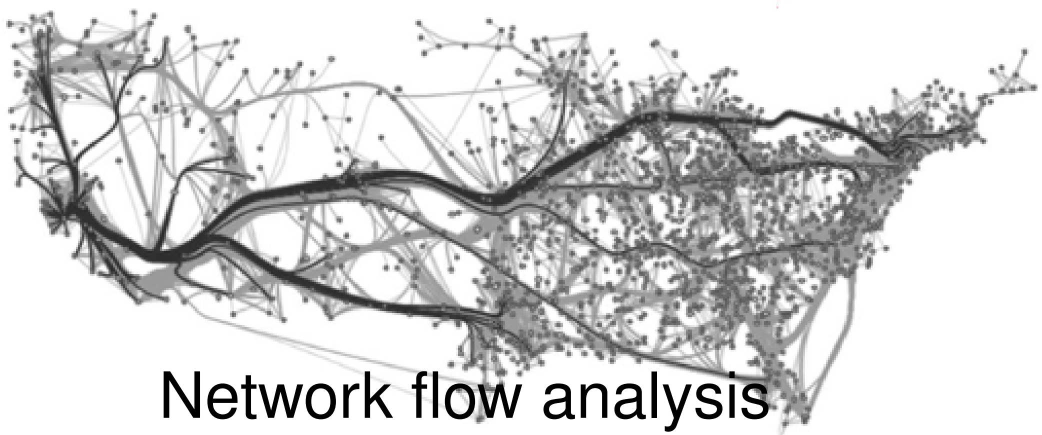


Network flow analysis

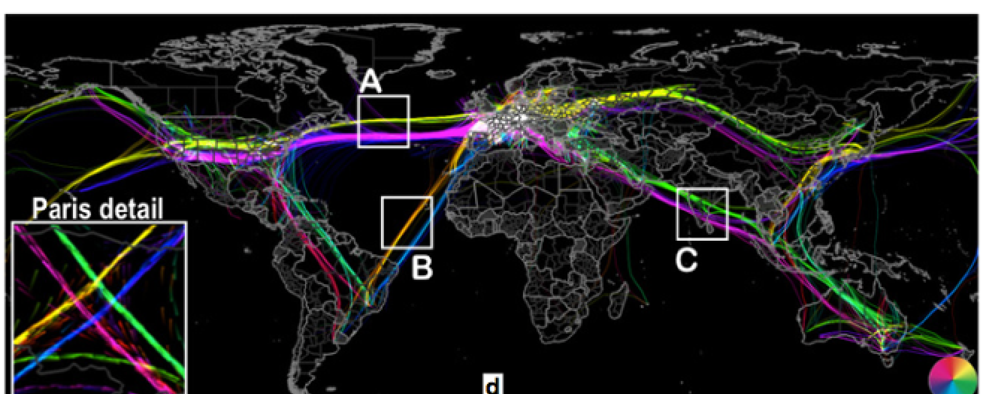
Edge bundling: applications



Software engeneering

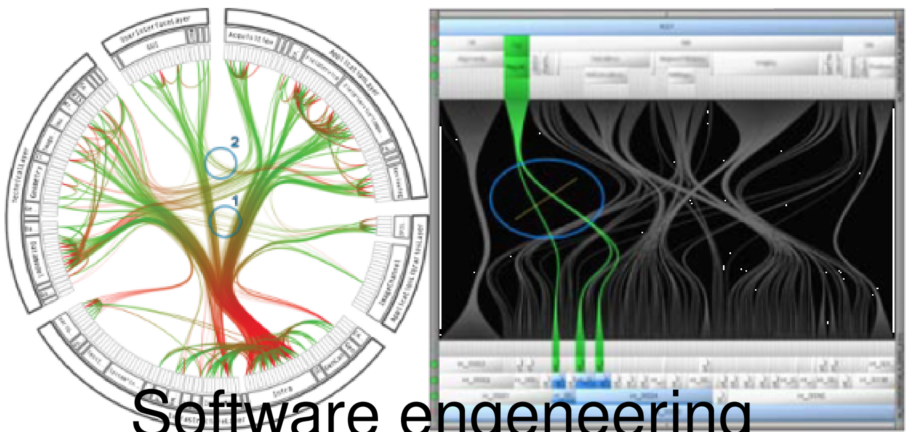


Network flow analysis



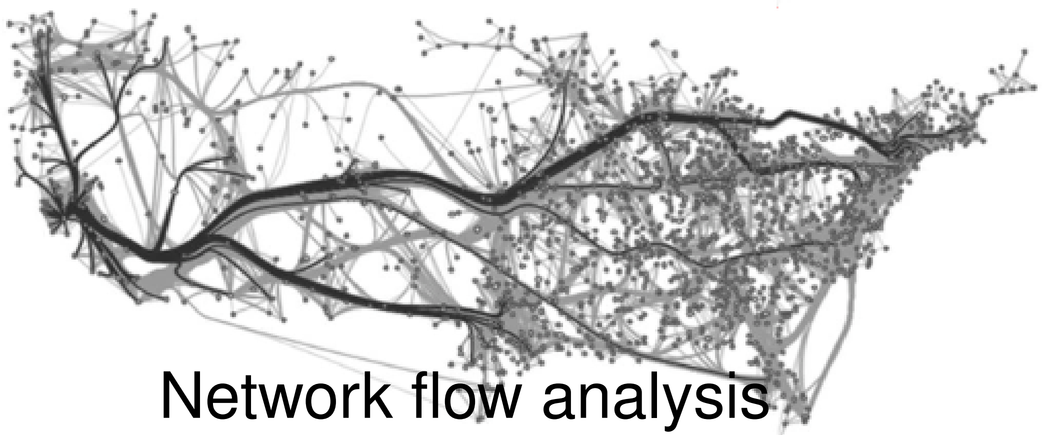
Air traffic control

Edge bundling: applications

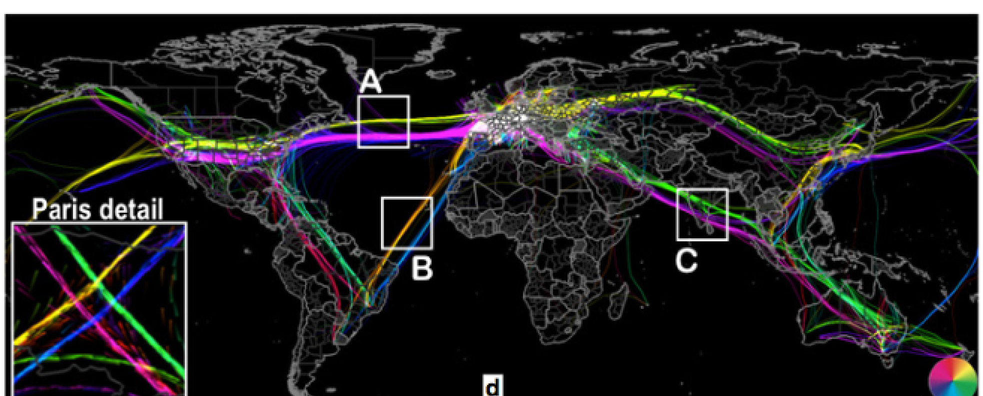


Software engeneering

Bioinformatics

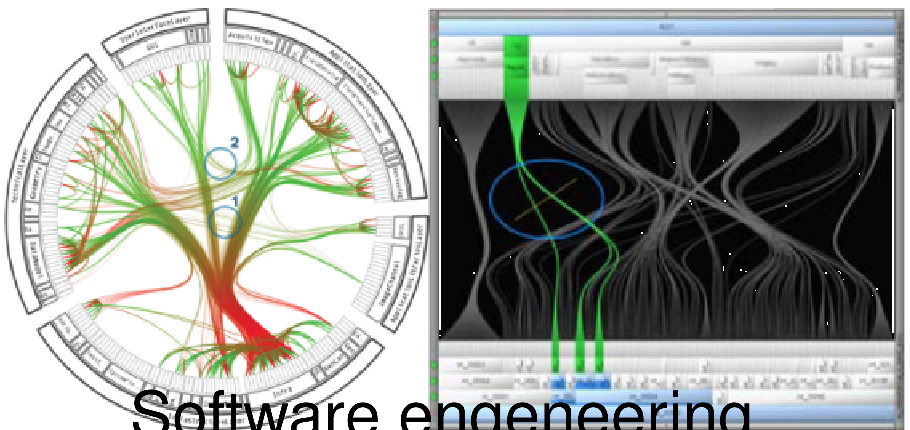


Network flow analysis



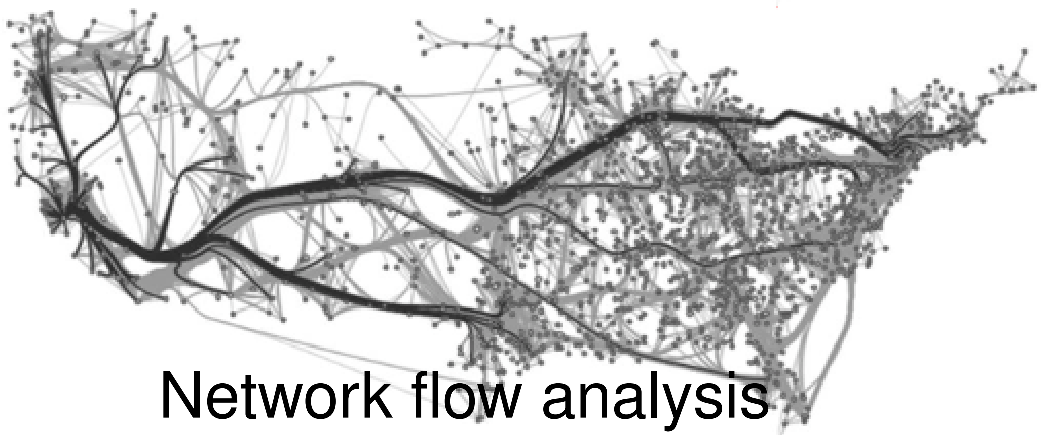
Air traffic control

Edge bundling: applications

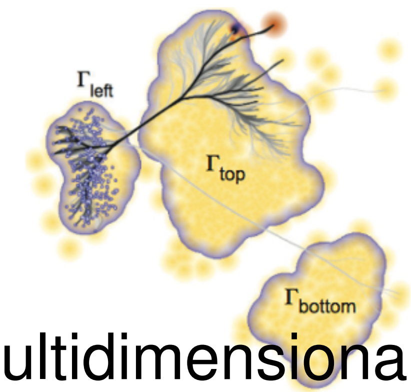


Software engeneering

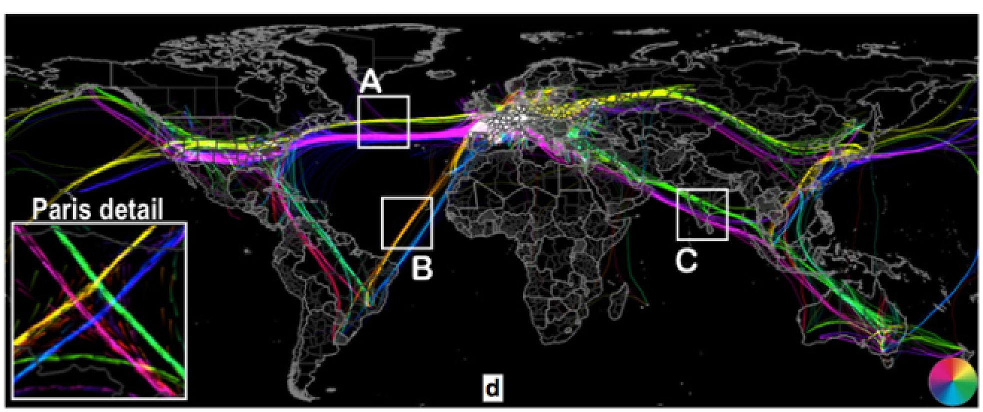
Bioinformatics



Network flow analysis

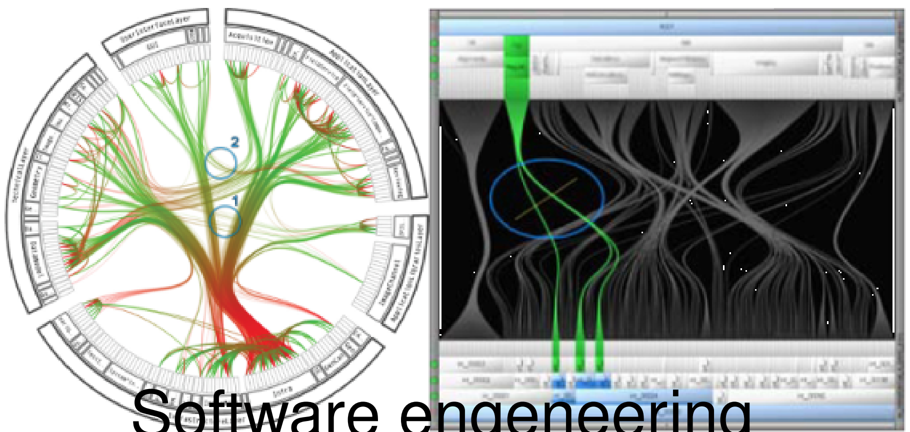


Multidimensional data



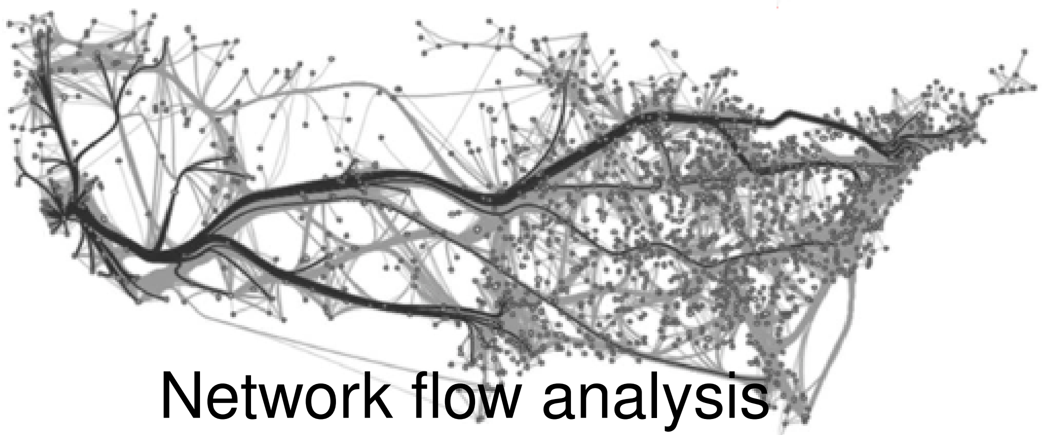
Air traffic control

Edge bundling: applications

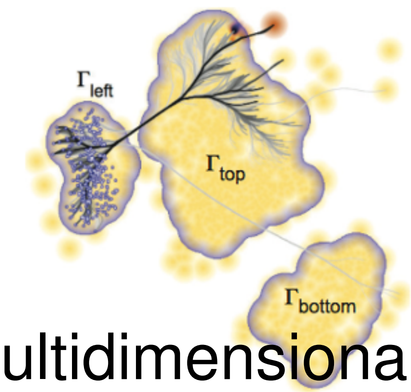


Software engeneering

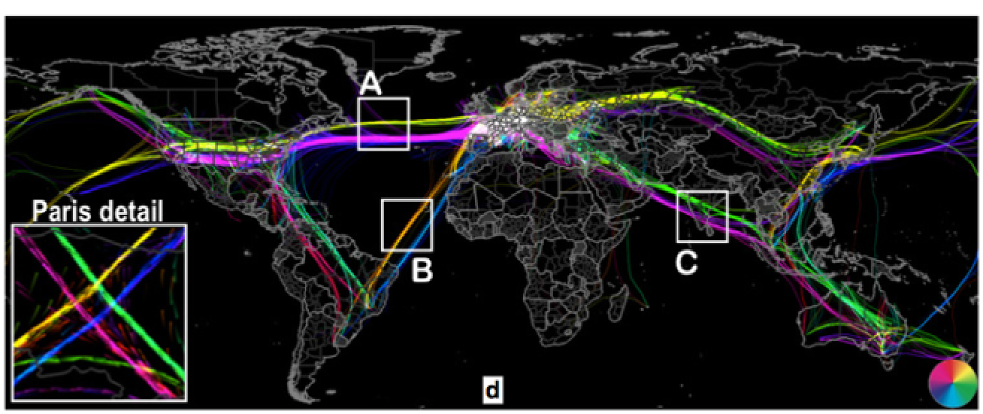
Bioinformatics



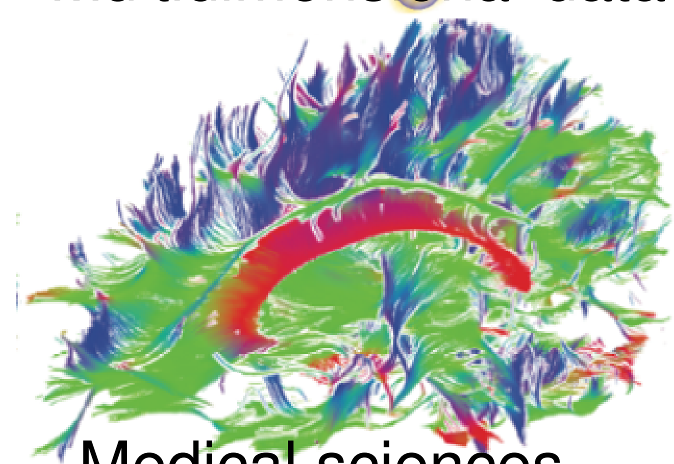
Network flow analysis



Multidimensional data

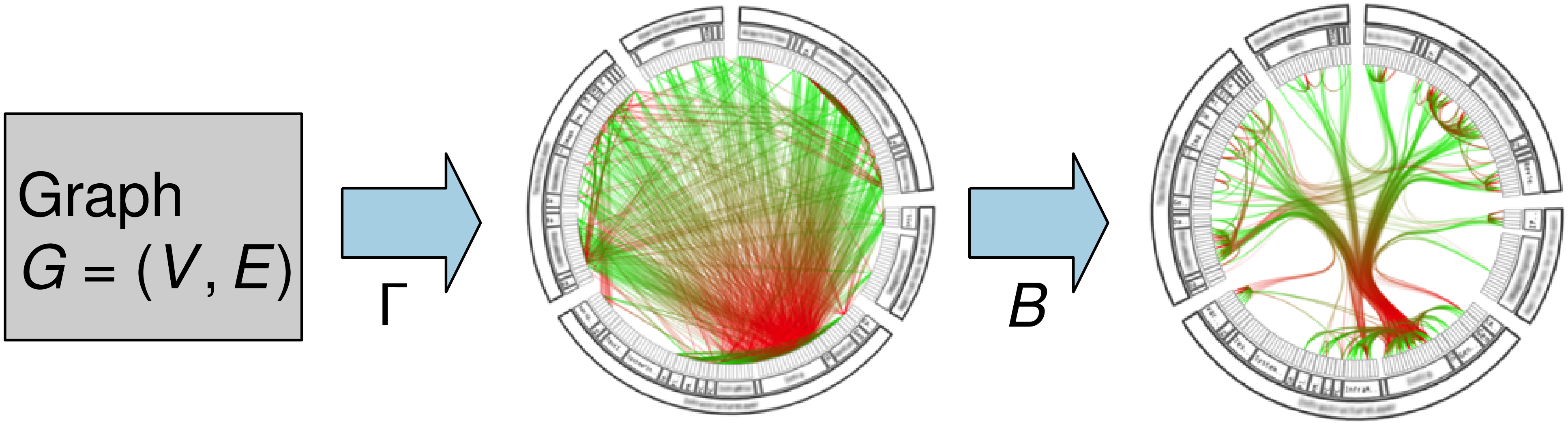


Air traffic control



Medical sciences

Edge bundling: definition



Γ – drawing/layout function

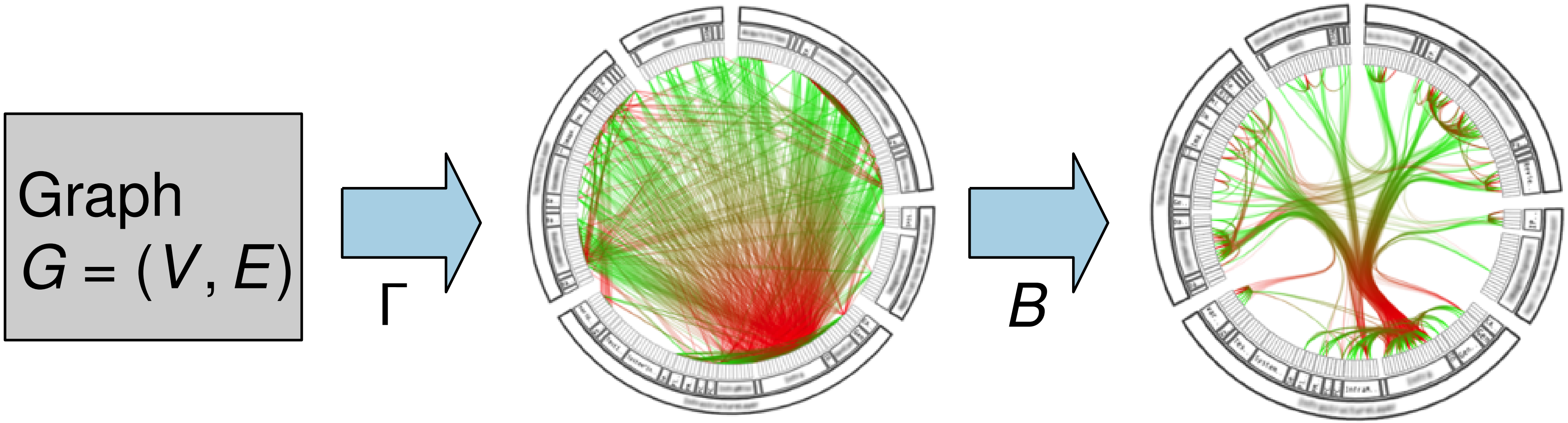
B – bundling function

$$\forall (e_i, e_j) \in E \times E \text{ such that } e_i \neq e_j \wedge k(e_i, e_j) < k_{\max} \rightarrow$$
$$\delta(B(\Gamma(e_i)), B(\Gamma(e_j))) \ll \delta(\Gamma(e_i), \Gamma(e_j))$$

$k_{\max}$ – maximum similarity of the edges that still need to be bundled

k–similarity of two edges; δ – similarity of two curves

Edge bundling: definition



Γ – drawing/layout function

B – bundling function

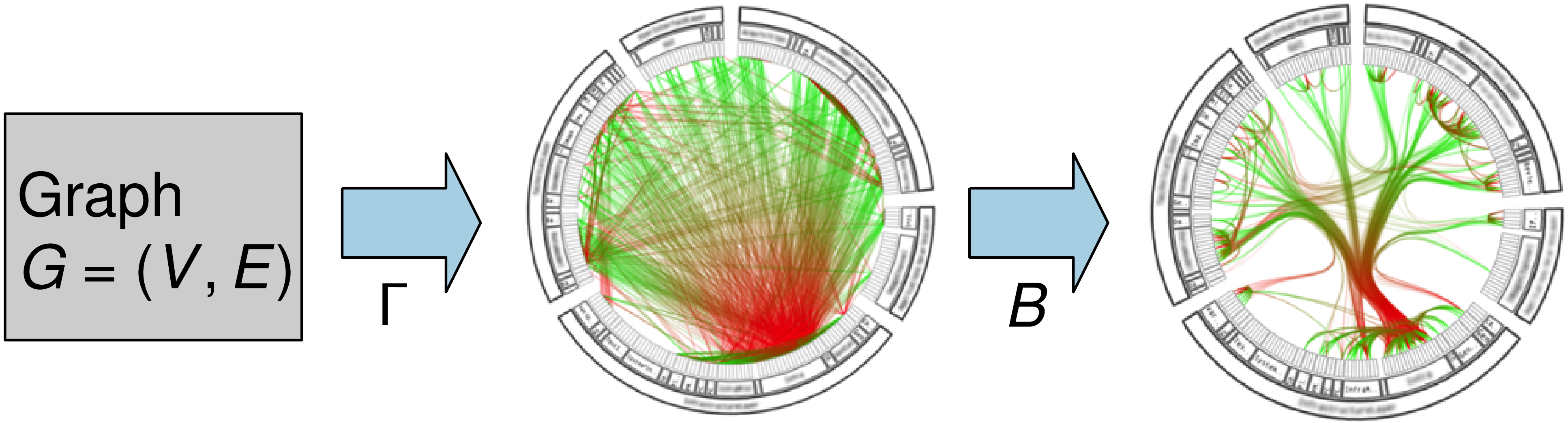
two edges are similar

$$\forall (e_i, e_j) \in E \times E \text{ such that } e_i \neq e_j \wedge k(e_i, e_j) < k_{\max} \rightarrow$$
$$\delta(B(\Gamma(e_i)), B(\Gamma(e_j))) \ll \delta(\Gamma(e_i), \Gamma(e_j))$$

$k_{\max}$ – maximum similarity of the edges that still need to be bundled

k–similarity of two edges; δ – similarity of two curves

Edge bundling: definition



Γ – drawing/layout function

B – bundling function

two edges are similar

$$\forall (e_i, e_j) \in E \times E \text{ such that } e_i \neq e_j \wedge k(e_i, e_j) < k_{\max} \rightarrow$$

$$\delta(B(\Gamma(e_i)), B(\Gamma(e_j))) \ll \delta(\Gamma(e_i), \Gamma(e_j))$$

the distance between curves after bundling is small

$k_{\max}$ – maximum similarity of the edges that still need to be bundled

k–similarity of two edges; δ – similarity of two curves

Edge bundling: definition

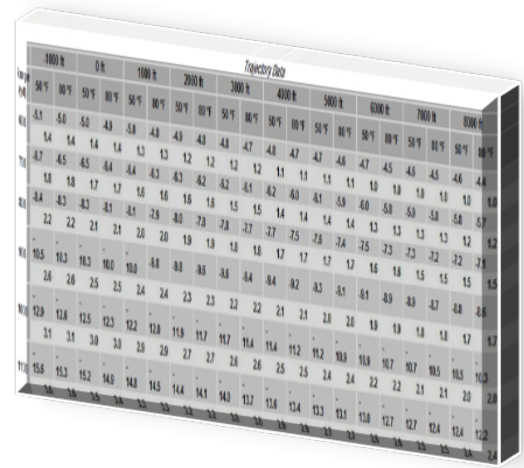
$$\forall(e_i, e_j) \in E \times E \text{ such that } e_i \neq e_j \wedge k(e_i, e_j) < k_{\text{max}} \rightarrow$$
$$\delta(B(\Gamma(e_i)), B(\Gamma(e_j))) \ll \delta(\Gamma(e_i), \Gamma(e_j))$$

k–similarity of two edges; δ – similarity of two curves

Edge bundling: definition

$$\forall(e_i, e_j) \in E \times E \text{ such that } e_i \neq e_j \wedge k(e_i, e_j) < k_{\text{max}} \rightarrow$$
$$\delta(B(\Gamma(e_i)), B(\Gamma(e_j))) \ll \delta(\Gamma(e_i), \Gamma(e_j))$$

k–similarity of two edges; δ – similarity of two curves



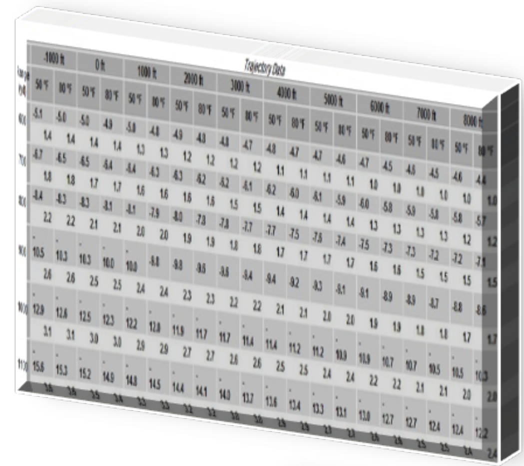
Data-based similarities

- Structured-based
- Attribute-based

Edge bundling: definition

$$\forall (e_i, e_j) \in E \times E \text{ such that } e_i \neq e_j \wedge k(e_i, e_j) < k_{\max} \rightarrow$$
$$\delta(B(\Gamma(e_i)), B(\Gamma(e_j))) \ll \delta(\Gamma(e_i), \Gamma(e_j))$$

k–similarity of two edges; δ – similarity of two curves

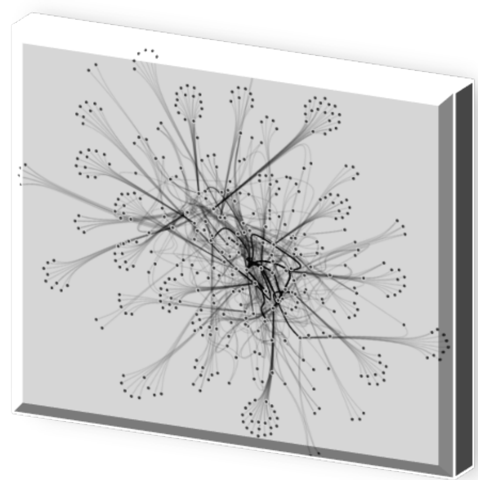


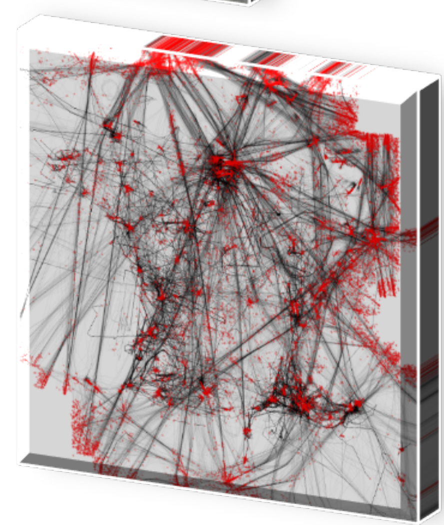
Data-based similarities

- Structured-based
- Attribute-based

Drawing-based similarities

- Geometric-based
- Image-based



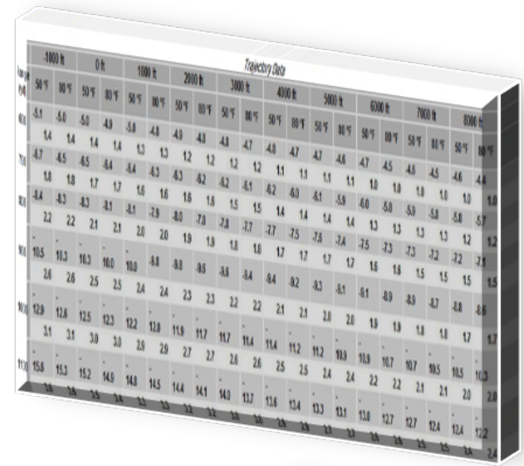


Edge bundling: definition

$$\forall (e_i, e_j) \in E \times E \text{ such that } e_i \neq e_j \wedge k(e_i, e_j) < k_{\max} \rightarrow$$

$$\delta(B(\Gamma(e_i)), B(\Gamma(e_j))) \ll \delta(\Gamma(e_i), \Gamma(e_j))$$

k–similarity of two edges; δ – similarity of two curves

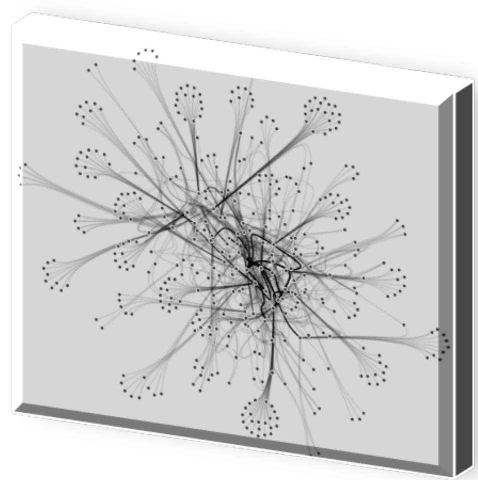


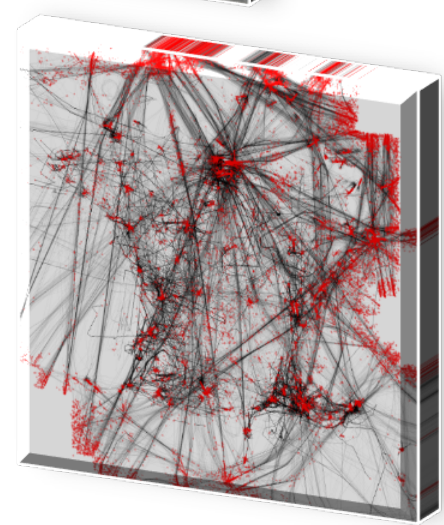
Data-based similarities

- Structured-based
- Attribute-based

Drawing-based similarities

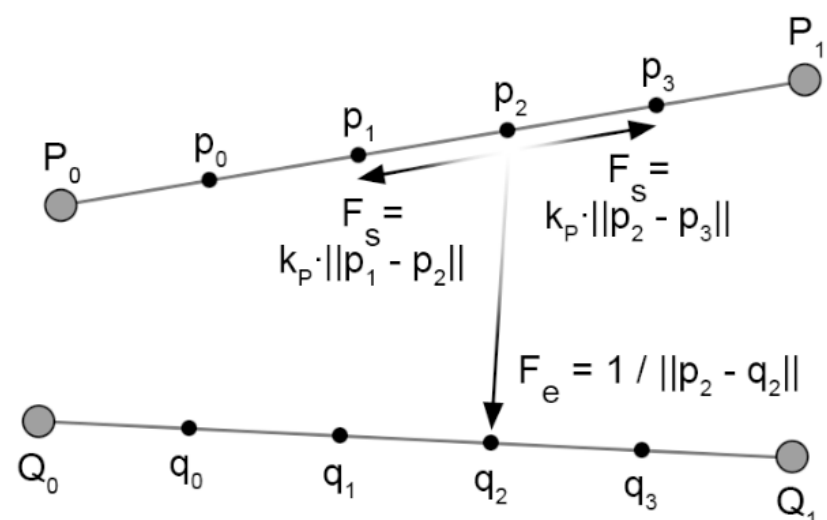
- Geometric-based
- Image-based





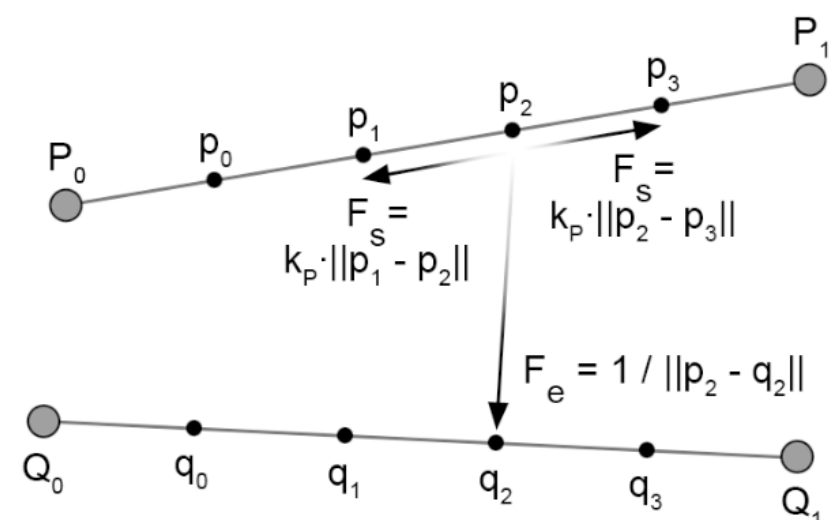
Edge bundling Holten and van Wijk, 2009

- Assume two edges *P* and *Q* need to be bundled (which – later). We say they are *interacting*.



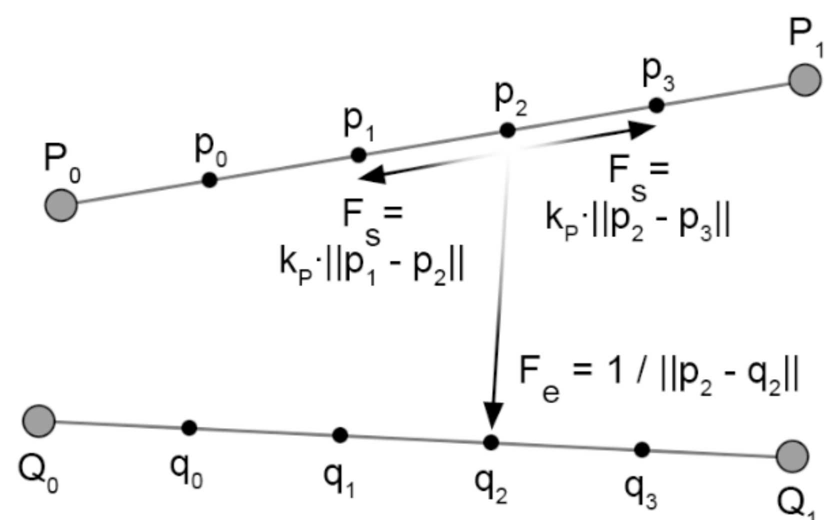
Edge bundling Holten and van Wijk, 2009

- Assume two edges *P* and *Q* need to be bundled (which – later). We say they are *interacting*.
- *P* and *Q* are subdivided using a few subdivision points per edge (how many – later)



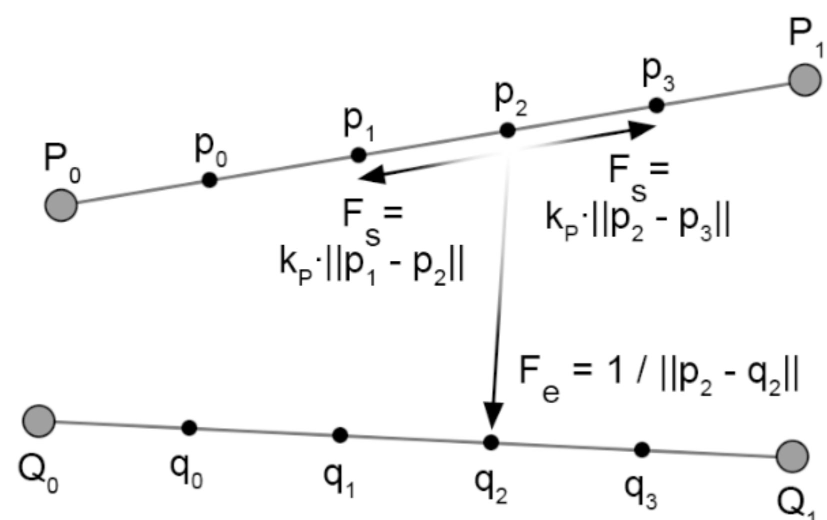
Edge bundling Holten and van Wijk, 2009

- Assume two edges *P* and *Q* need to be bundled (which – later). We say they are *interacting*.
- *P* and *Q* are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed



Edge bundling **Holten and van Wijk, 2009**

- Assume two edges *P* and *Q* need to be bundled (which – later). We say they are *interacting*.
- *P* and *Q* are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed
- Spring (attraction) force F_s between the consecutive vertices of each edge - tries to keep the edges straight

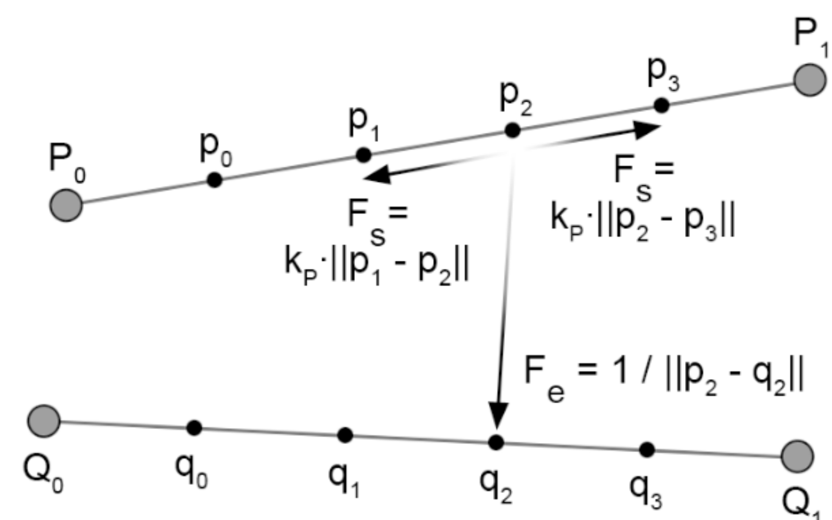


Edge bundling Holten and van Wijk, 2009

- Assume two edges *P* and *Q* need to be bundled (which – later). We say they are *interacting*.
- *P* and *Q* are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed
- Spring (attraction) force F_s between the consecutive vertices of each edge - tries to keep the edges straight

Spring force

$F_s(\{p_i, p_{i+1}\}) = k_P||p_{i-1} - p_i||$



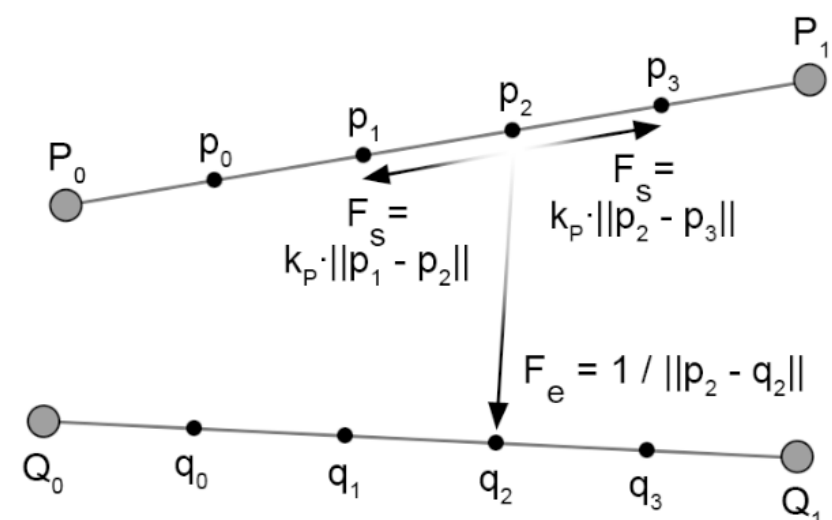
Edge bundling **Holten and van Wijk, 2009**

- Assume two edges *P* and *Q* need to be bundled (which – later). We say they are *interacting*.
- *P* and *Q* are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed
- Spring (attraction) force F_s between the consecutive vertices of each edge - tries to keep the edges straight

Spring force

$F_s(\{p_i, p_{i+1}\}) = k_P||p_{i-1} - p_i||$

n_P – number of segments on *P*,



Edge bundling Holten and van Wijk, 2009

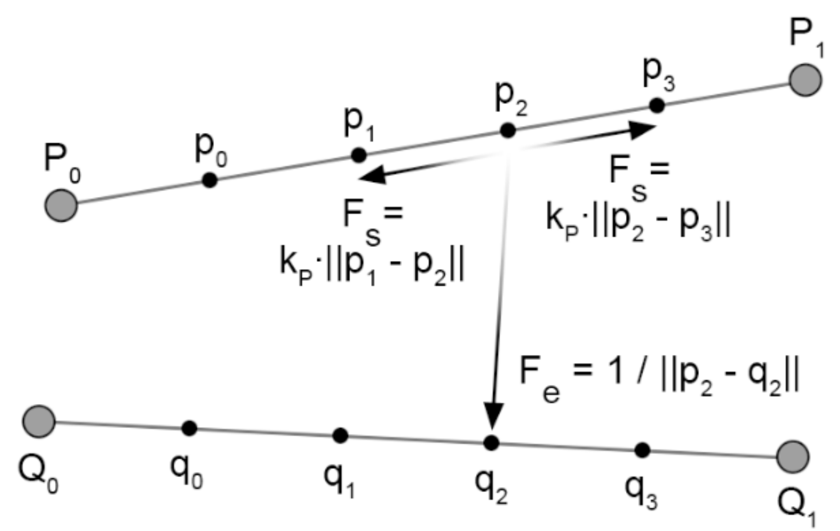
- Assume two edges P and Q need to be bundled (which – later). We say they are *interacting*.
- P and Q are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed
- Spring (attraction) force F_s between the consecutive vertices of each edge - tries to keep the edges straight

Spring force

$F_s(\{p_i, p_{i+1}\}) = k_P||p_{i-1} - p_i||$

n_P – number of segments on P,

$\frac{|P|}{n_P}$ – initial length of a segment of P



Edge bundling Holten and van Wijk, 2009

- Assume two edges P and Q need to be bundled (which – later). We say they are *interacting*.
- P and Q are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed
- Spring (attraction) force F_s between the consecutive vertices of each edge - tries to keep the edges straight

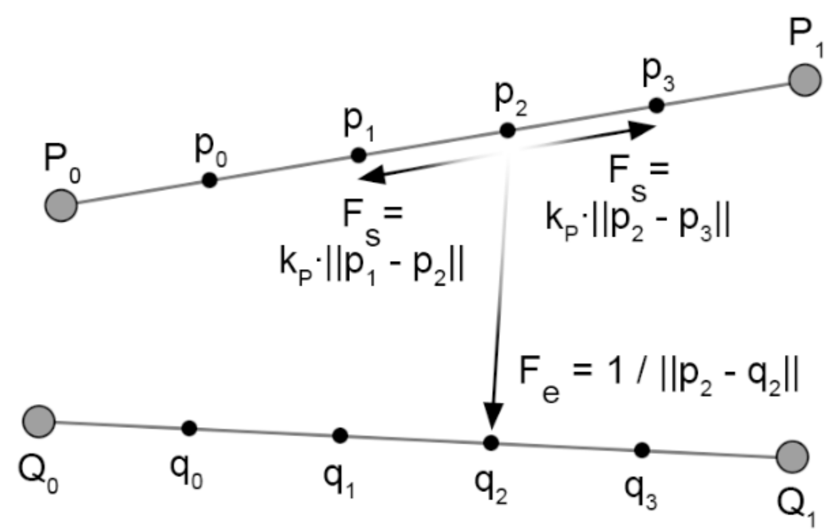
Spring force

$$F_s(\{p_i, p_{i+1}\}) = k_P||p_{i-1} - p_i||$$

n_P – number of segments on P,

$\frac{|P|}{n_P}$ – initial length of a segment of P

$$k_P = K / \frac{|P|}{n_P} = \frac{K}{|P|} n_P$$



Edge bundling **Holten and van Wijk, 2009**

- Assume two edges *P* and *Q* need to be bundled (which – later). We say they are *interacting*.
- *P* and *Q* are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed
- Spring (attraction) force F_s between the consecutive vertices of each edge - tries to keep the edges straight

Spring force

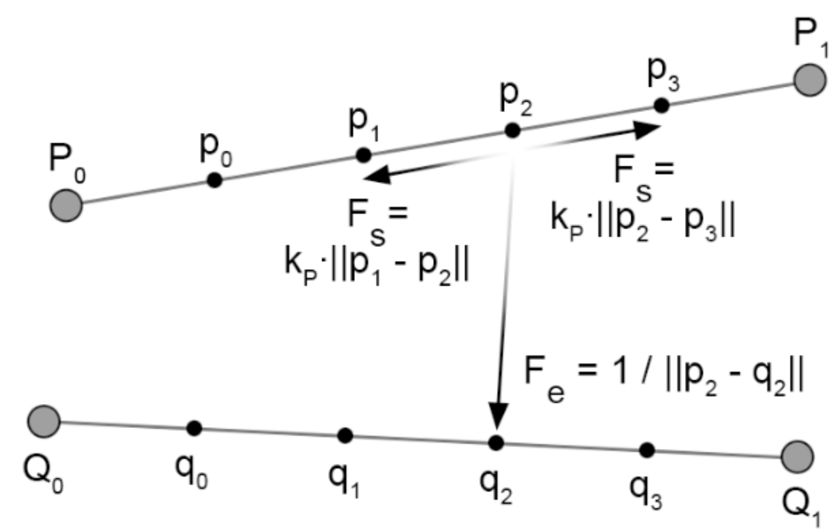
$$F_s(\{p_i, p_{i+1}\}) = k_P||p_{i-1} - p_i||$$

n_P – number of segments on *P*,

$\frac{|P|}{n_P}$ – initial length of a segment of *P*

$$k_P = K/\frac{|P|}{n_P} = \frac{K}{|P|}n_P$$

here *K* – global stiffness constant



Edge bundling Holten and van Wijk, 2009

- Assume two edges P and Q need to be bundled (which – later). We say they are *interacting*.
- P and Q are subdivided using a few subdivision points per edge (how many – later)
- The position of edge end-points P_0, P_1, Q_0, and Q_1 remain fixed
- Spring (attraction) force F_s between the consecutive vertices of each edge - tries to keep the edges straight

Spring force

$F_s(\{p_i, p_{i+1}\}) = k_P||p_{i-1} - p_i||$

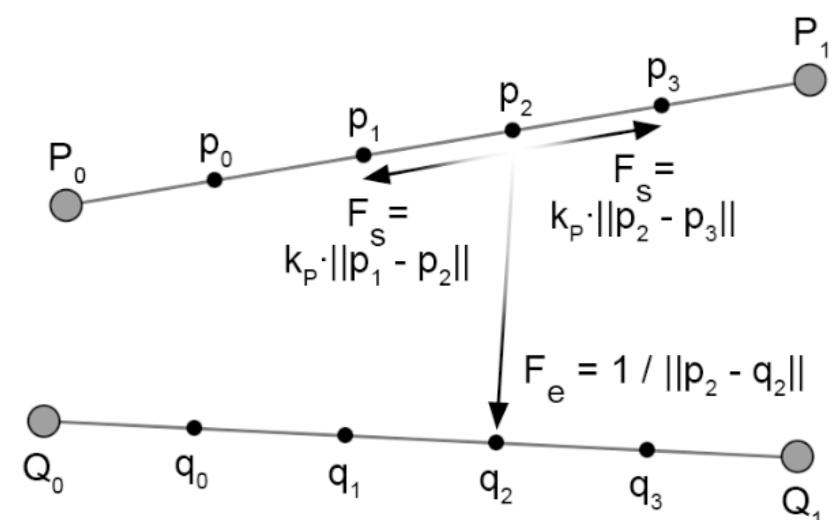
n_P – number of segments on P,

$\frac{|P|}{n_P}$ – initial length of a segment of P

$k_P = K / \frac{|P|}{n_P} = \frac{K}{|P|} n_P$

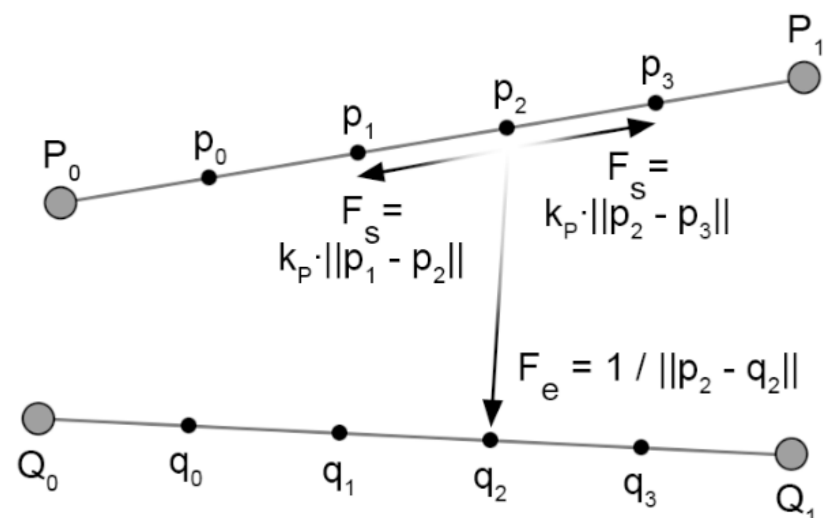
here K – global stiffness constant

Large values of K make system very stiff



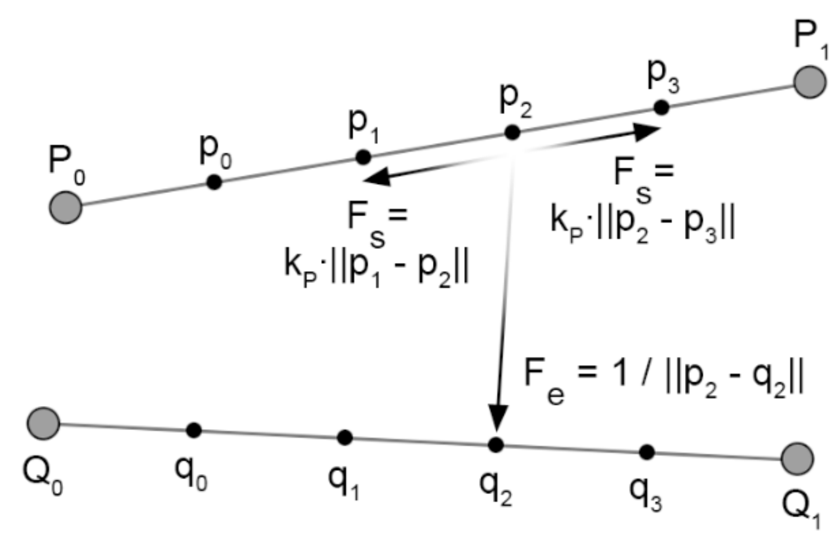
Edge bundling Holten and van Wijk, 2009

- An attraction electrostatic force $F_e(\{p_i, q_i\}) = \frac{1}{||p_i - q_i||}$ is used between each pair of corresponding subdivision points of P and Q, thus between p_0 and q_0, p_1 and q_1, ...



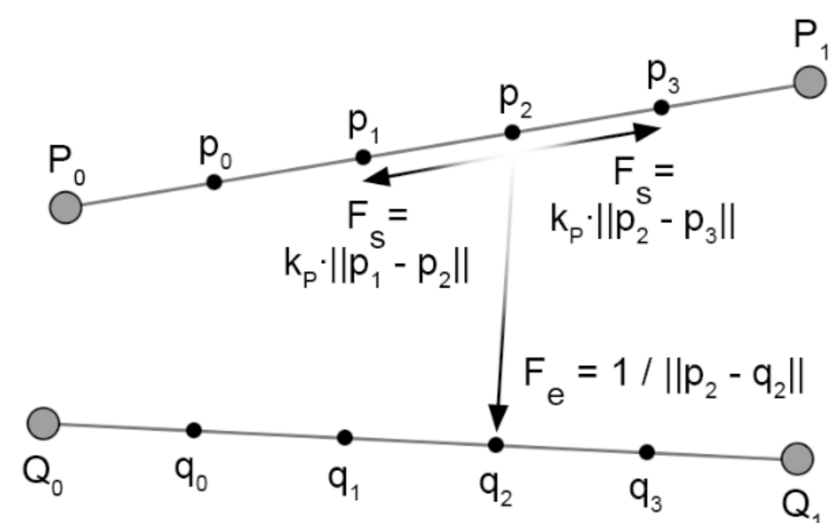
Edge bundling Holten and van Wijk, 2009

- An attraction electrostatic force $F_e(\{p_i, q_i\}) = \frac{1}{||p_i - q_i||}$ is used between each pair of corresponding subdivision points of P and Q, thus between p_0 and q_0, p_1 and q_1, ...
- F_e tries to bundle the edges



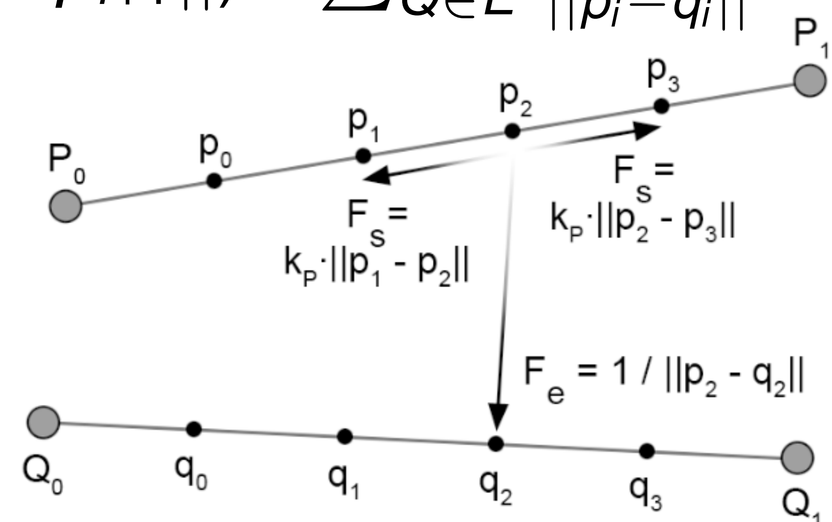
Edge bundling Holten and van Wijk, 2009

- An attraction electrostatic force $F_e(\{p_i, q_i\}) = \frac{1}{||p_i - q_i||}$ is used between each pair of corresponding subdivision points of P and Q, thus between p_0 and q_0, p_1 and q_1, ...
- F_e tries to bundle the edges
- F_e can be calculated for all pairs (p_i, q_j) but this slows down the performance and does not increase the performance significantly (Experiment!)



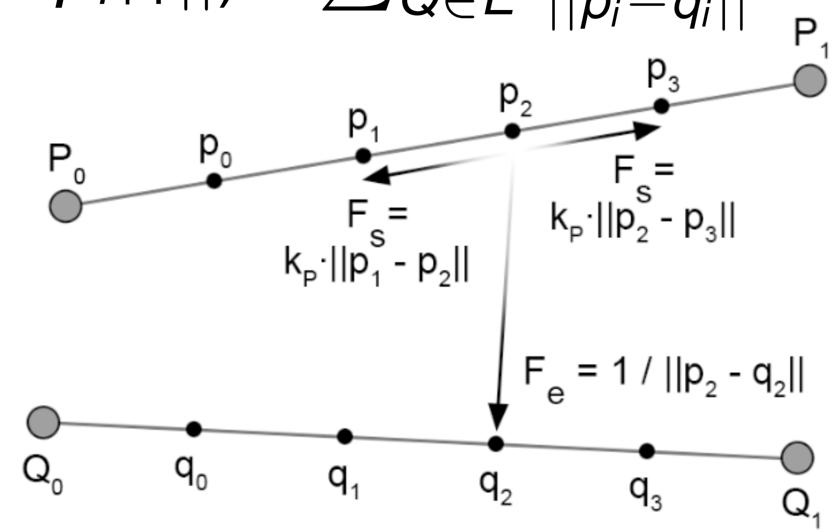
Edge bundling Holten and van Wijk, 2009

- An attraction electrostatic force $F_e(\{p_i, q_i\}) = \frac{1}{||p_i - q_i||}$ is used between each pair of corresponding subdivision points of P and Q, thus between p_0 and q_0, p_1 and q_1, ...
- F_e tries to bundle the edges
- F_e can be calculated for all pairs (p_i, q_j) but this slows down the performance and does not increase the performance significantly (Experiment!)
- The overall force on point p_i is
  $F_{p_i} = k_P \cdot (||p_{i-1} - p_i)|| + ||p_i - p_{i+1}||) + \sum_{Q \in E} \frac{1}{||p_i - q_i||}$



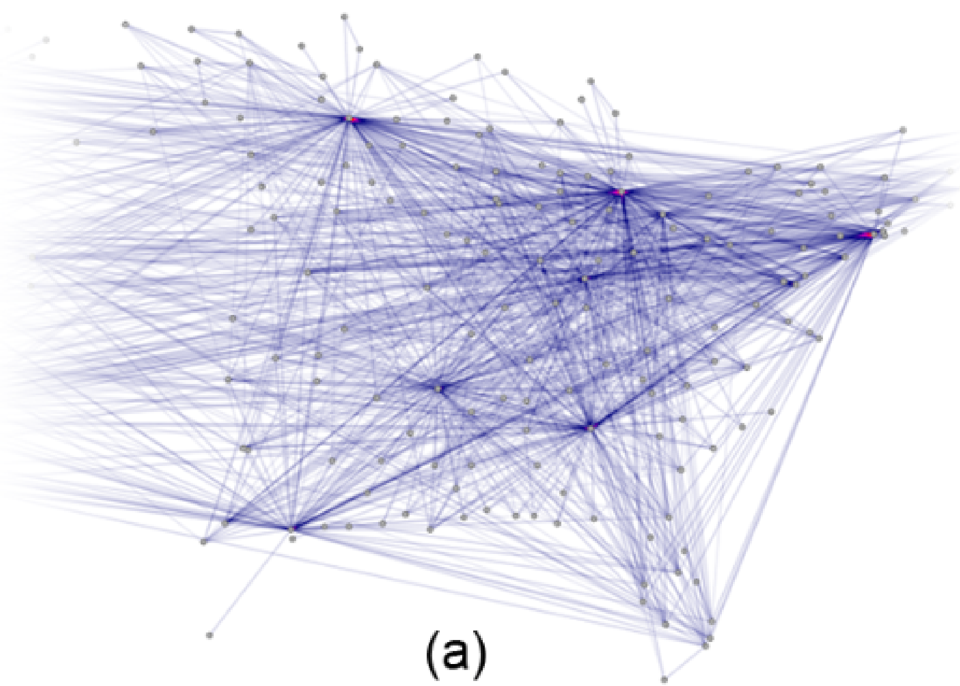
Edge bundling Holten and van Wijk, 2009

- An attraction electrostatic force $F_e(\{p_i, q_i\}) = \frac{1}{||p_i - q_i||}$ is used between each pair of corresponding subdivision points of P and Q, thus between p_0 and q_0, p_1 and q_1, ...
- F_e tries to bundle the edges
- F_e can be calculated for all pairs (p_i, q_j) but this slows down the performance and does not increase the performance significantly (Experiment!)
- The overall force on point p_i is
  $F_{p_i} = k_P \cdot (||p_{i-1} - p_i)|| + ||p_i - p_{i+1}||) + \sum_{Q \in E} \frac{1}{||p_i - q_i||}$
  k_P – constant for edge P

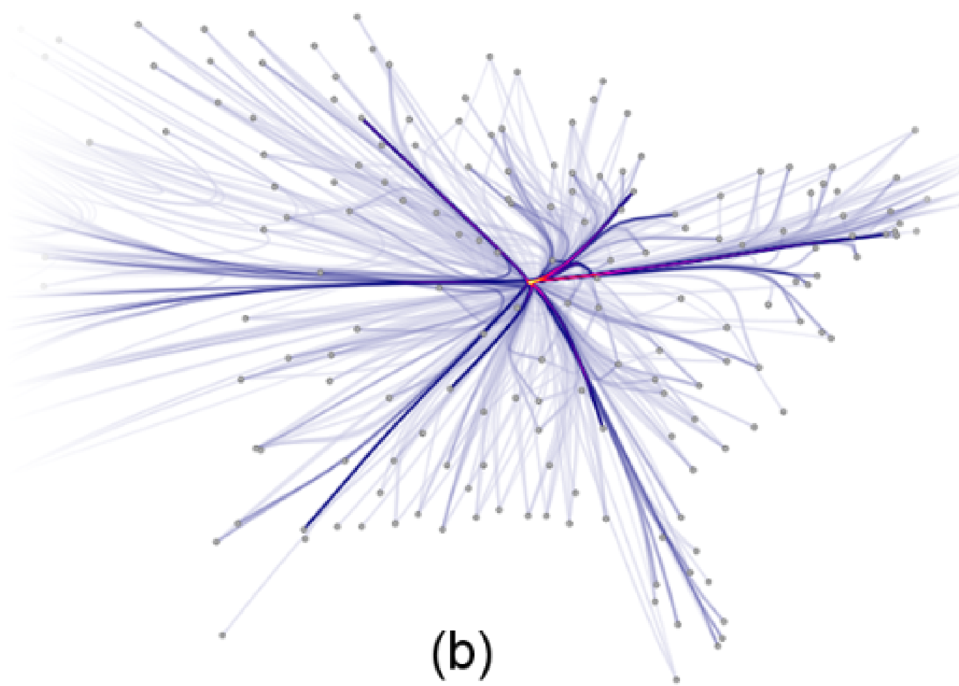


Edge bundling: performance

- Fig.b – performance of the model given up to now. Here all edges interact with all.



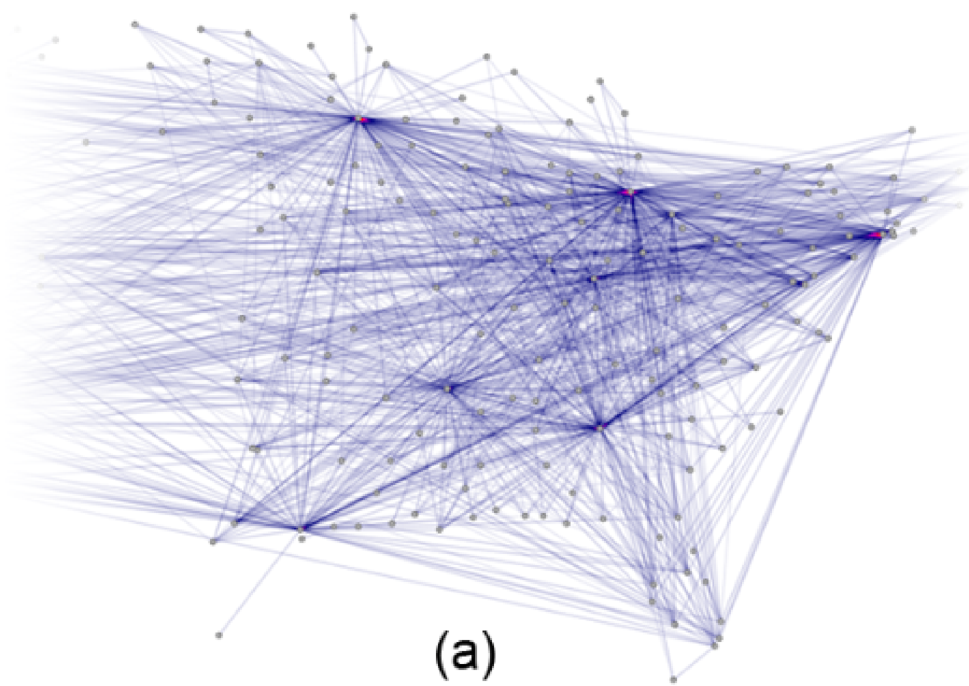
(a)



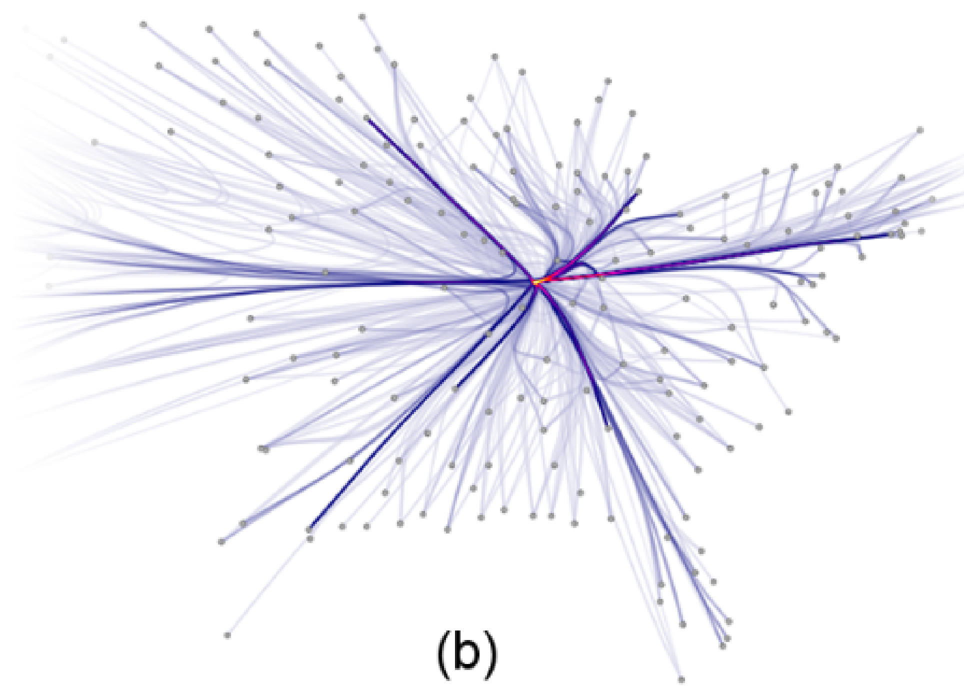
(b)

Edge bundling: performance

- Fig.b – performance of the model given up to now. Here all edges interact with all.
- Increasing the value of *K* gives less bundling overall and therefore in parts of the graph where a high amount of bundling is still desirable



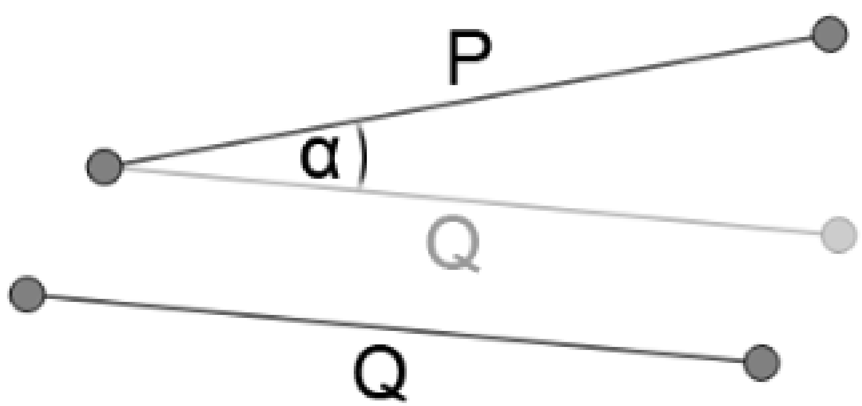
(a)



(b)

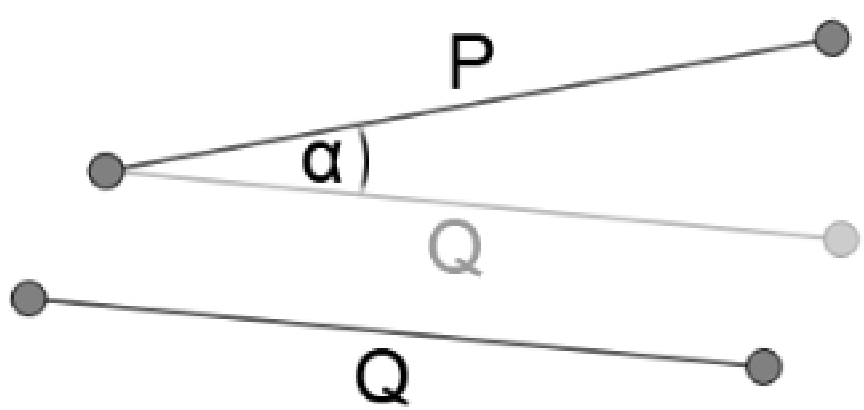
Edge compatibility measures: angle

- edges that are almost perpendicular should not be bundled together, i.e. should not interact



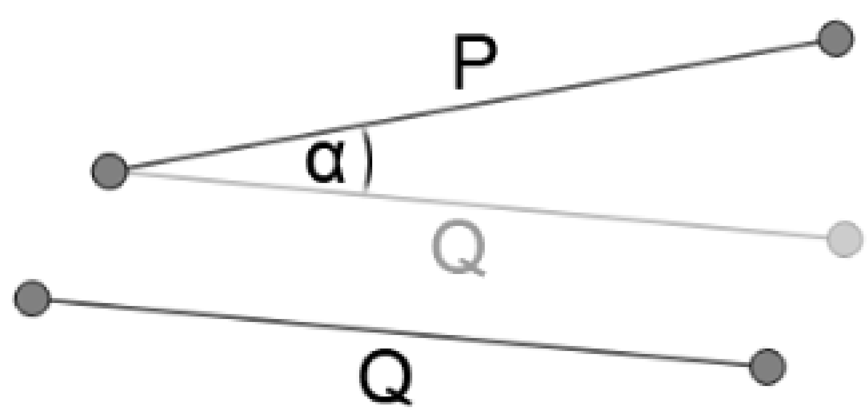
Edge compatibility measures: angle

- edges that are almost perpendicular should not be bundled together, i.e. should not interact
- *angle compatibility* of P and Q is defined as $C_a(P, Q) = |\cos\alpha|$, α is angle between P and Q



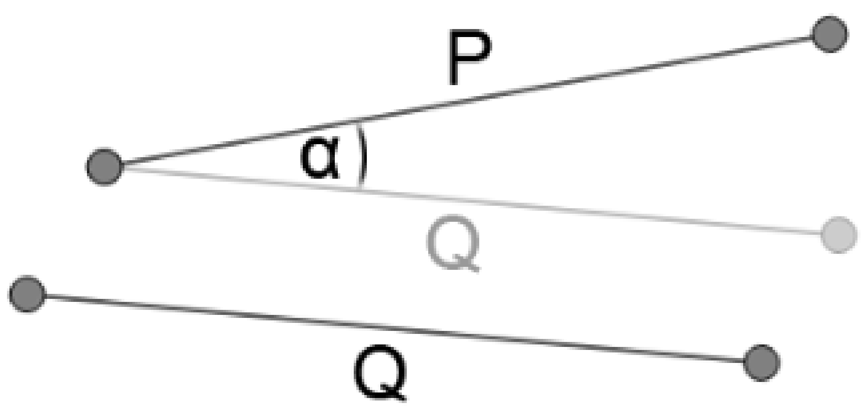
Edge compatibility measures: angle

- edges that are almost perpendicular should not be bundled together, i.e. should not interact
- *angle compatibility* of P and Q is defined as $C_a(P, Q) = |\cos\alpha|$, α is angle between P and Q
- calculate through dot product $\cos\alpha = \frac{P \cdot Q}{||P||||Q||}$



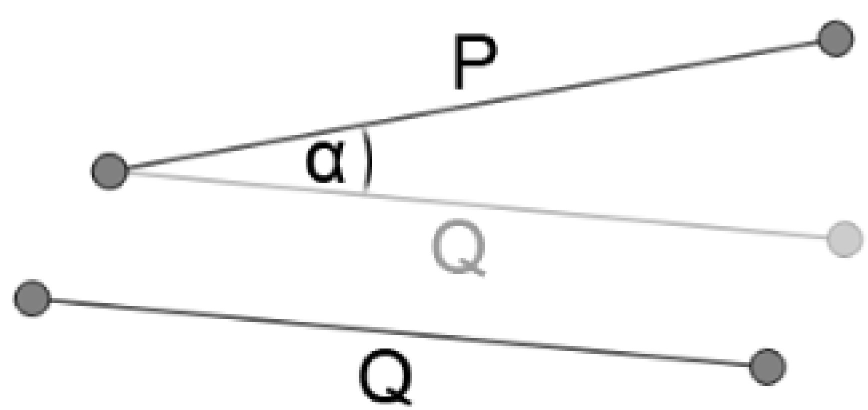
Edge compatibility measures: angle

- edges that are almost perpendicular should not be bundled together, i.e. should not interact
- *angle compatibility* of P and Q is defined as $C_a(P, Q) = |\cos \alpha|$, α is angle between P and Q
- calculate through dot product $\cos \alpha = \frac{P \cdot Q}{||P||||Q||}$
- the larger α, the smaller $C_a(P, Q)$



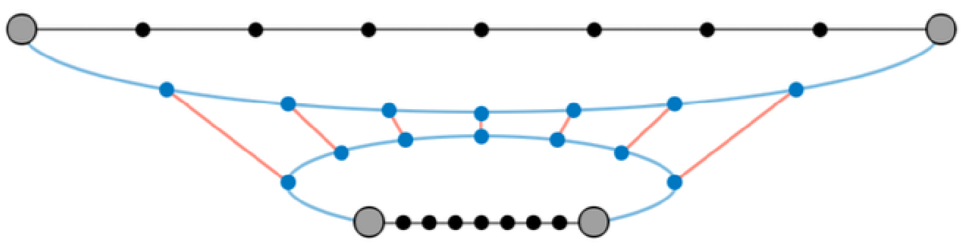
Edge compatibility measures: angle

- edges that are almost perpendicular should not be bundled together, i.e. should not interact
- *angle compatibility* of P and Q is defined as $C_a(P, Q) = |\cos\alpha|$, α is angle between P and Q
- calculate through dot product $\cos\alpha = \frac{P \cdot Q}{||P||\,||Q||}$
- the larger α, the smaller $C_a(P, Q)$
- $C_a(P, Q) = 0$ if $\alpha = 90^\circ$ and $C_a(P, Q) = 1$ if $\alpha = 0$, i.e. P and Q are parallel



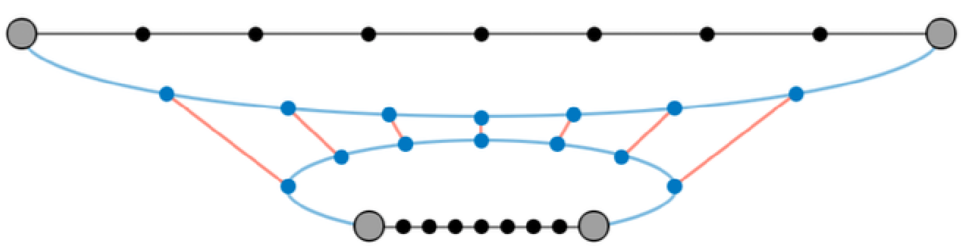
Edge compatibility measures: scale

- edges that differ a lot in length should not be bundled together



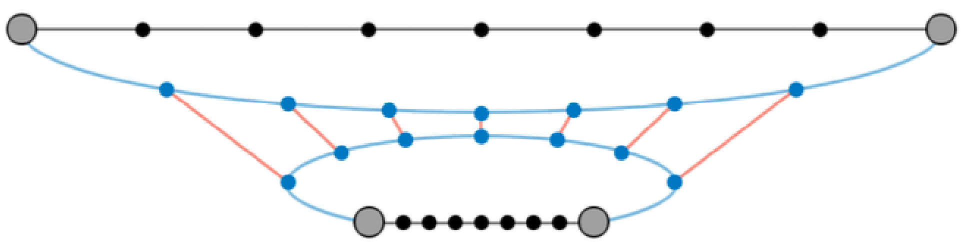
Edge compatibility measures: scale

- edges that differ a lot in length should not be bundled together
- doing so might result in stretching and curving of short edges



Edge compatibility measures: scale

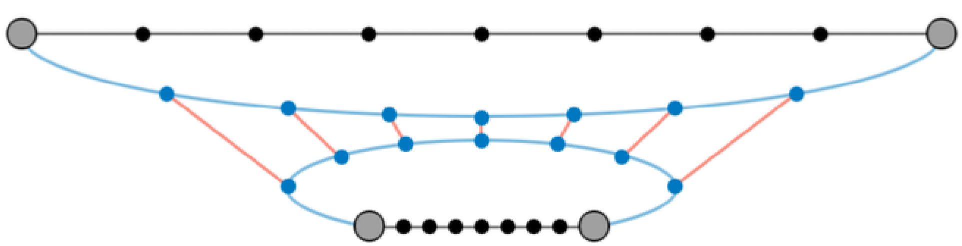
- edges that differ a lot in length should not be bundled together
- doing so might result in stretching and curving of short edges
- *scale compartibility* is defined as
  $C_s(P, Q) = \frac{2}{\ell_{avg} \cdot \min(|P|,|Q|) + \max(|P|,|Q|)/\ell_{avg}}$, where $\ell_{avg} = \frac{|P|+|Q|}{2}$



Edge compatibility measures: scale

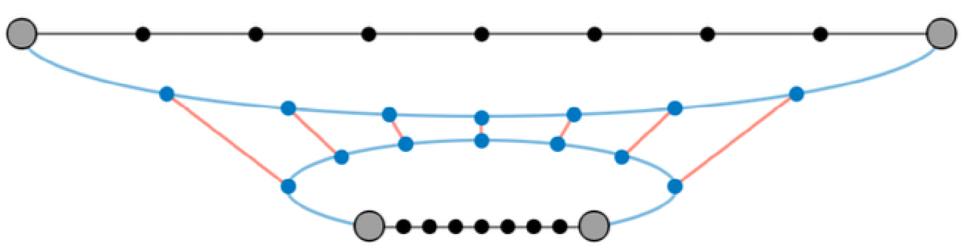
- edges that differ a lot in length should not be bundled together
- doing so might result in stretching and curving of short edges
- *scale compartibility* is defined as
  $C_s(P, Q) = \frac{2}{\ell_{avg} \cdot \min(|P|,|Q|) + \max(|P|,|Q|)/\ell_{avg}}$, where $\ell_{avg} = \frac{|P|+|Q|}{2}$

- $C_s(P, Q) = 1$ if $|P| = |Q|$



Edge compatibility measures: scale

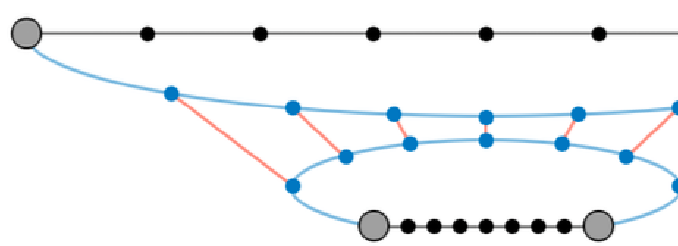
- edges that differ a lot in length should not be bundled together
- doing so might result in stretching and curving of short edges
- *scale compartibility* is defined as
  $C_s(P, Q) = \frac{2}{\ell_{avg} \cdot \min(|P|,|Q|) + \max(|P|,|Q|)/\ell_{avg}}$, where $\ell_{avg} = \frac{|P|+|Q|}{2}$
- $C_s(P, Q) = 1$ if $|P| = |Q|$
  $C_s(P, Q) \to 0$ if $||P| - |Q|| \to \infty$



Edge compatibility measures: scale

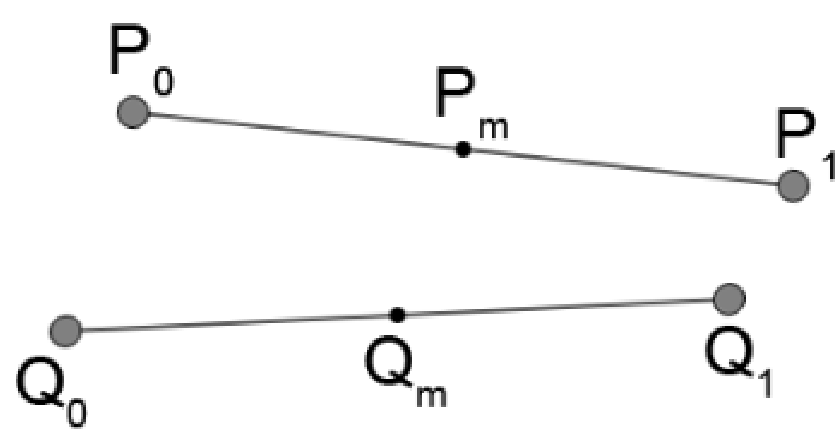
- edges that differ a lot in length should not be bundled together
- doing so might result in stretching and curving of short edges
- *scale compartibility* is defined as
  $C_s(P,Q) = \frac{2}{\ell_{avg} \cdot \min(|P|,|Q|) + \max(|P|,|Q|)/\ell_{avg}}$, where $\ell_{avg} = \frac{|P|+|Q|}{2}$
- $C_s(P,Q) = 1$ if $|P| = |Q|$
  $C_s(P,Q) \to 0$ if $||P| - |Q|| \to \infty$

!!!Correction:
Set $|P| = |P| / \min(|P|,|Q|)$ and $|Q| = |Q| / \min(|P|,|Q|)$ - normalize so that shortest has length one, otherwise $C_s(P,Q) > 1$



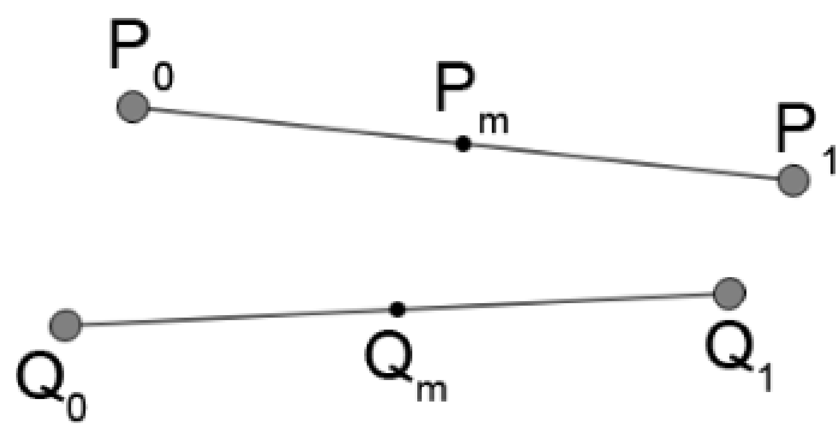
Edge compatibility measures: distance

- edges that are far apart should not be bundled together



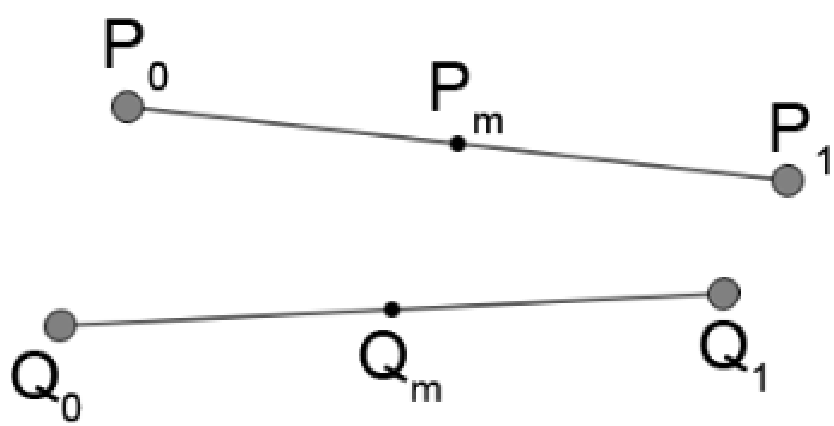
Edge compatibility measures: distance

- edges that are far apart should not be bundled together
- *position compatibility* is defines as
  $C_p(P, Q) = \ell_{\text{avg}} / (\ell_{\text{avg}} + ||P_m - Q_m||)$,
  with P_m and Q_m – midpoints of P and Q, respectively



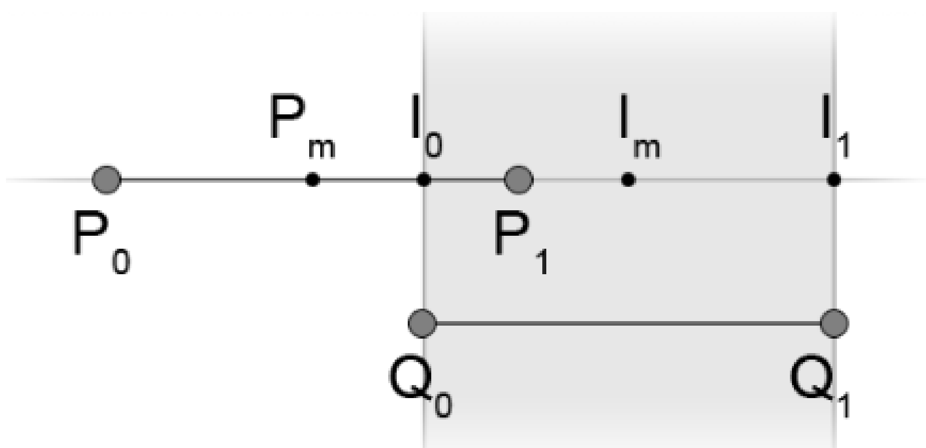
Edge compatibility measures: distance

- edges that are far apart should not be bundled together
- *position compatibility* is defines as
  $C_p(P, Q) = \ell_{\text{avg}}/(\ell_{\text{avg}} + ||P_m - Q_m||)$,
  with P_m and Q_m – midpoints of P and Q, respectively
- $C_p(P, Q) = 1$ if $P_m = Q_m$
  $C_p(P, Q) \to 0$ if $||P_m - Q_m|| \to \infty$



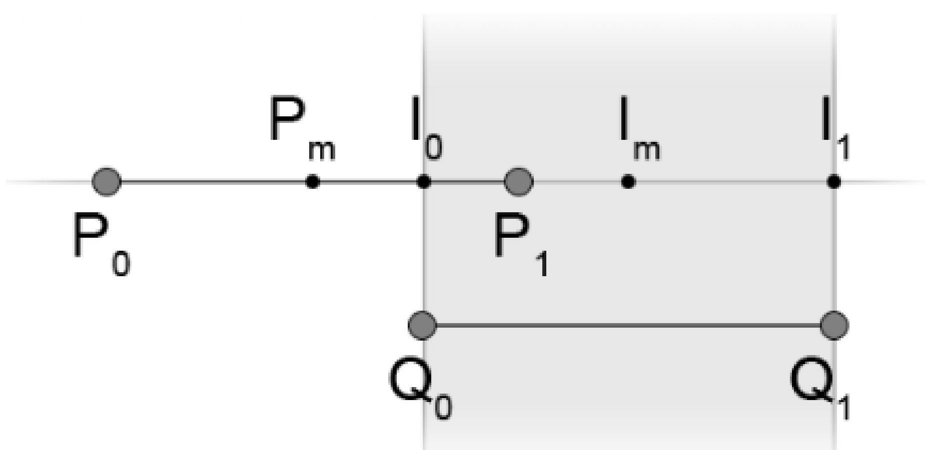
Edge compatibility measures: visibility

- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling



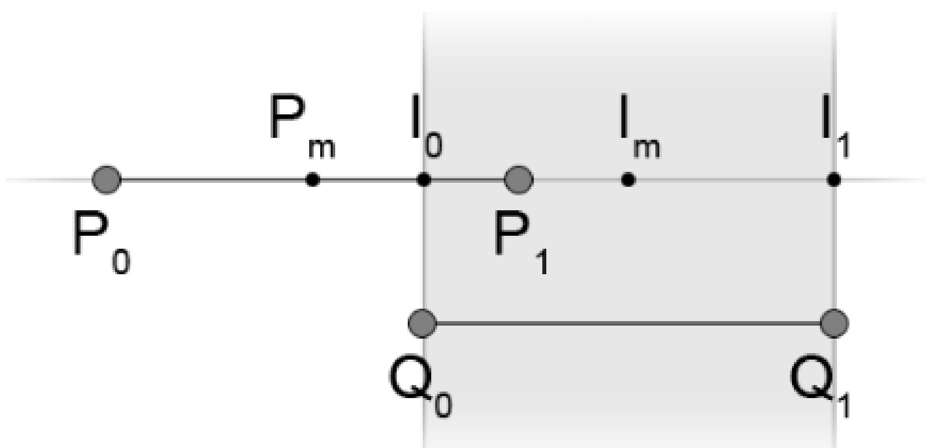
Edge compatibility measures: visibility

- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling
- these are the opposite edges of a skewed parallelogram



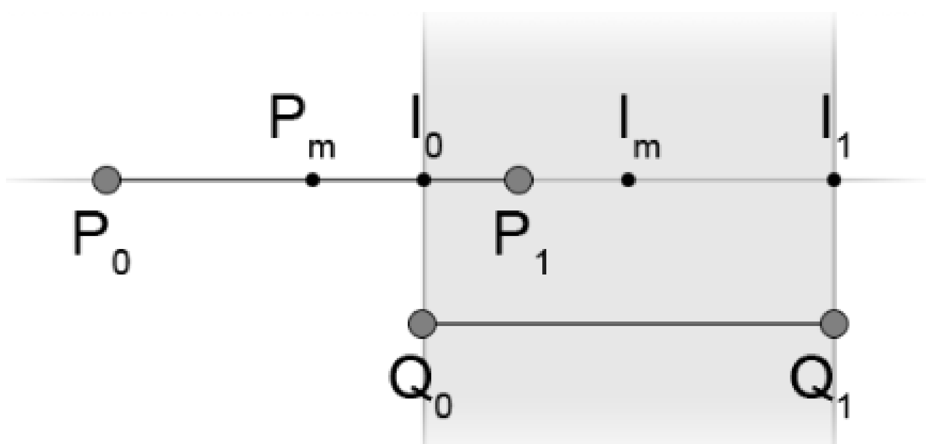
Edge compatibility measures: visibility

- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling
- these are the opposite edges of a skewed parallelogram
- *visibility compatibility*
  $C_v(P, Q) = \min(\text{vis}(P, Q), \text{vis}(Q, P))$, with



Edge compatibility measures: visibility

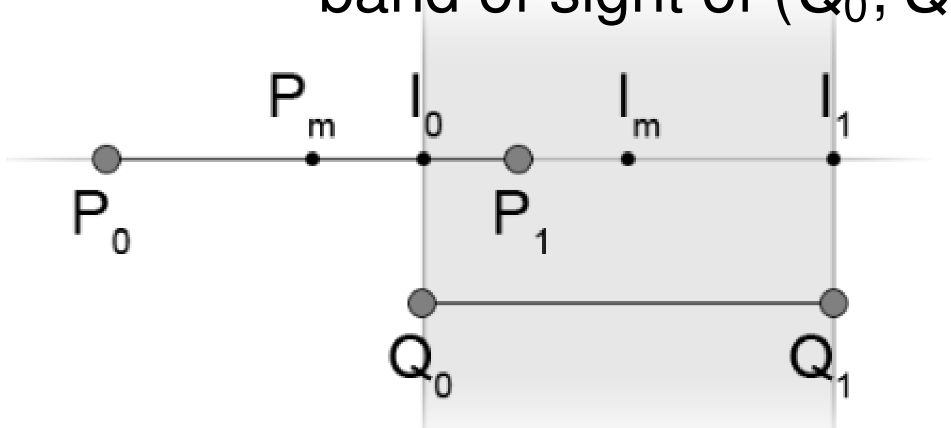
- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling
- these are the opposite edges of a skewed parallelogram
- *visibility compatibility*
  $C_v(P, Q) = \min(\text{vis}(P, Q), \text{vis}(Q, P))$, with
  $\text{vis}(P, Q) = \max(1 - \frac{2||P_m - I_m||}{||I_0 - I_1||}, 0)$, I_m – mid point of I_0 and I_1.



Edge compatibility measures: visibility

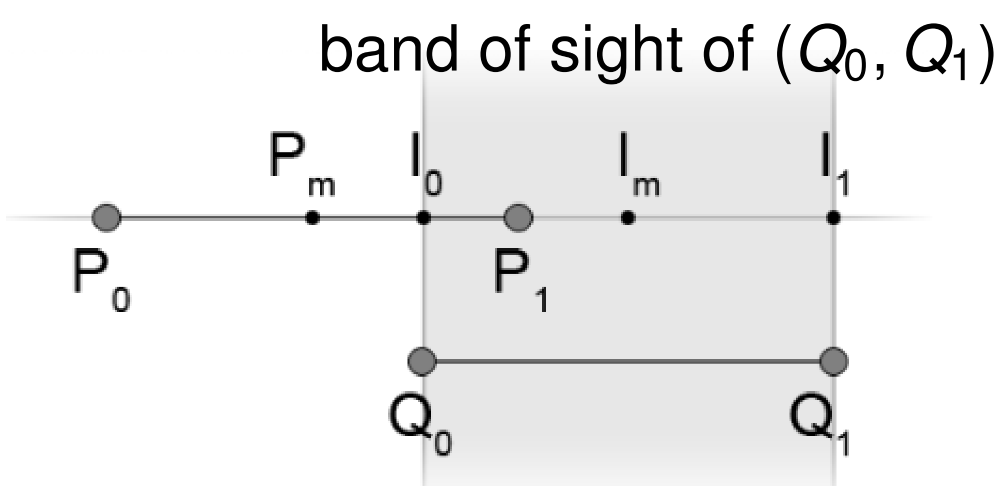
- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling
- these are the opposite edges of a skewed parallelogram
- *visibility compatibility*
  $C_v(P, Q) = \min(\mathrm{vis}(P, Q), \mathrm{vis}(Q, P))$, with
  $\mathrm{vis}(P, Q) = \max\left(1 - \frac{2||P_m - I_m||}{||I_0 - I_1||}, 0\right)$, I_m – mid point of I_0 and I_1.

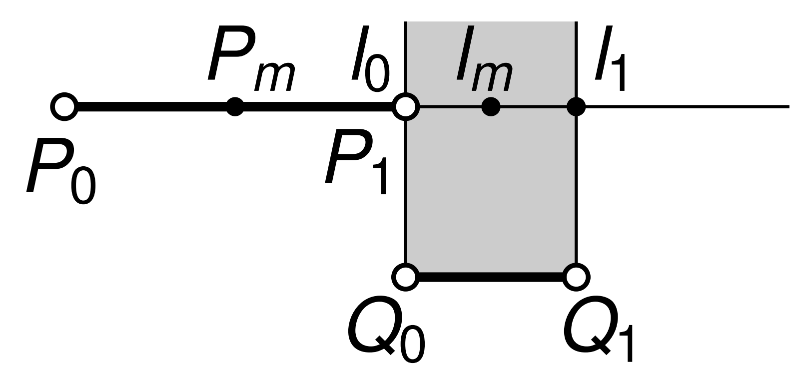
band of sight of (Q_0, Q_1)



Edge compatibility measures: visibility

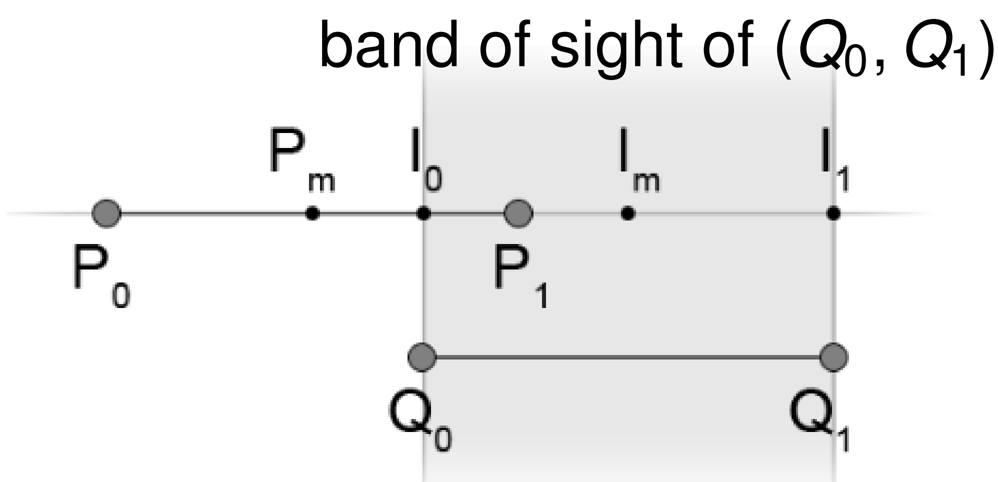
- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling
- these are the opposite edges of a skewed parallelogram
- *visibility compatibility* $C_v(P, Q) = \min(\mathrm{vis}(P, Q), \mathrm{vis}(Q, P))$, with $\mathrm{vis}(P, Q) = \max(1 - \frac{2||P_m - I_m||}{||I_0 - I_1||}, 0)$, I_m – mid point of I_0 and I_1.

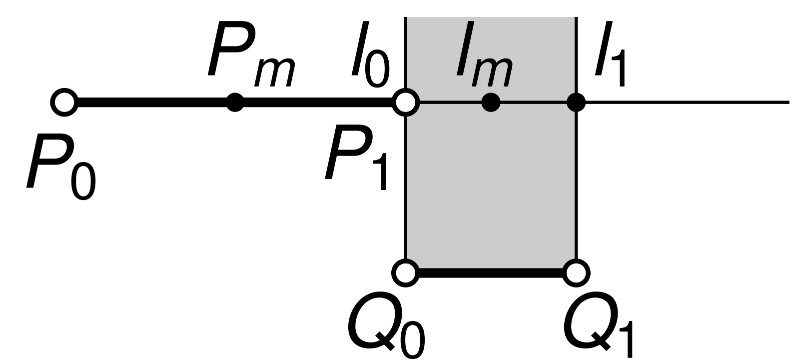




Edge compatibility measures: visibility

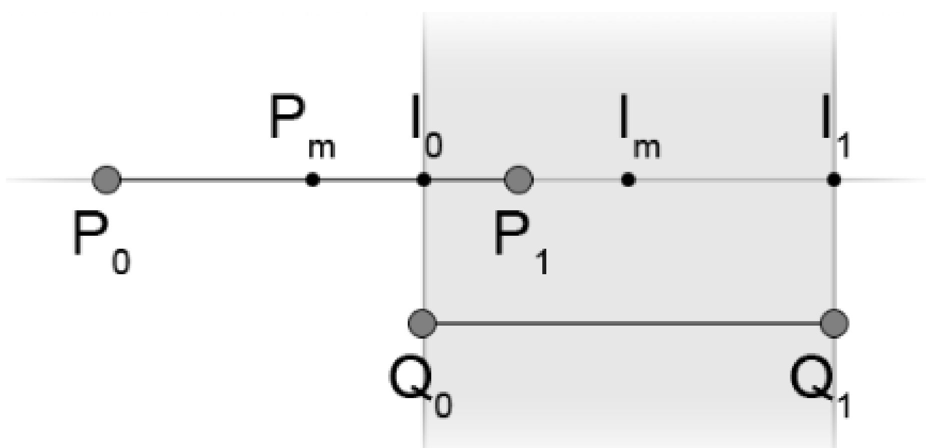
- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling
- these are the opposite edges of a skewed parallelogram
- *visibility compatibility*
  $C_v(P, Q) = \min(\text{vis}(P, Q), \text{vis}(Q, P))$, with
  $\text{vis}(P, Q) = \max(1 - \frac{2||P_m - I_m||}{||I_0 - I_1||}, 0)$, I_m – mid point of I_0 and I_1.
- $C_v(P, Q) = 1$ if P_m coincides with I_m (ideal position),

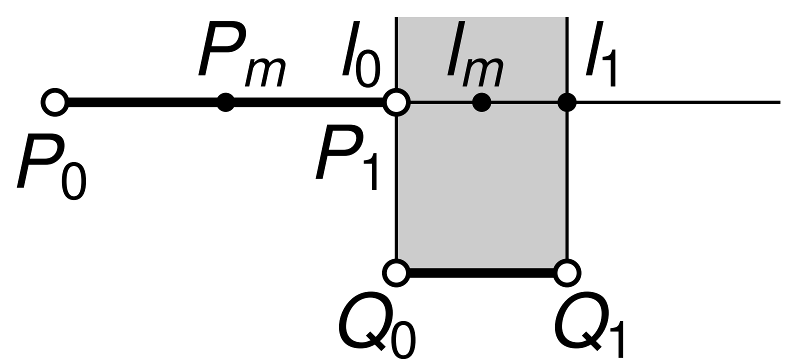




Edge compatibility measures: visibility

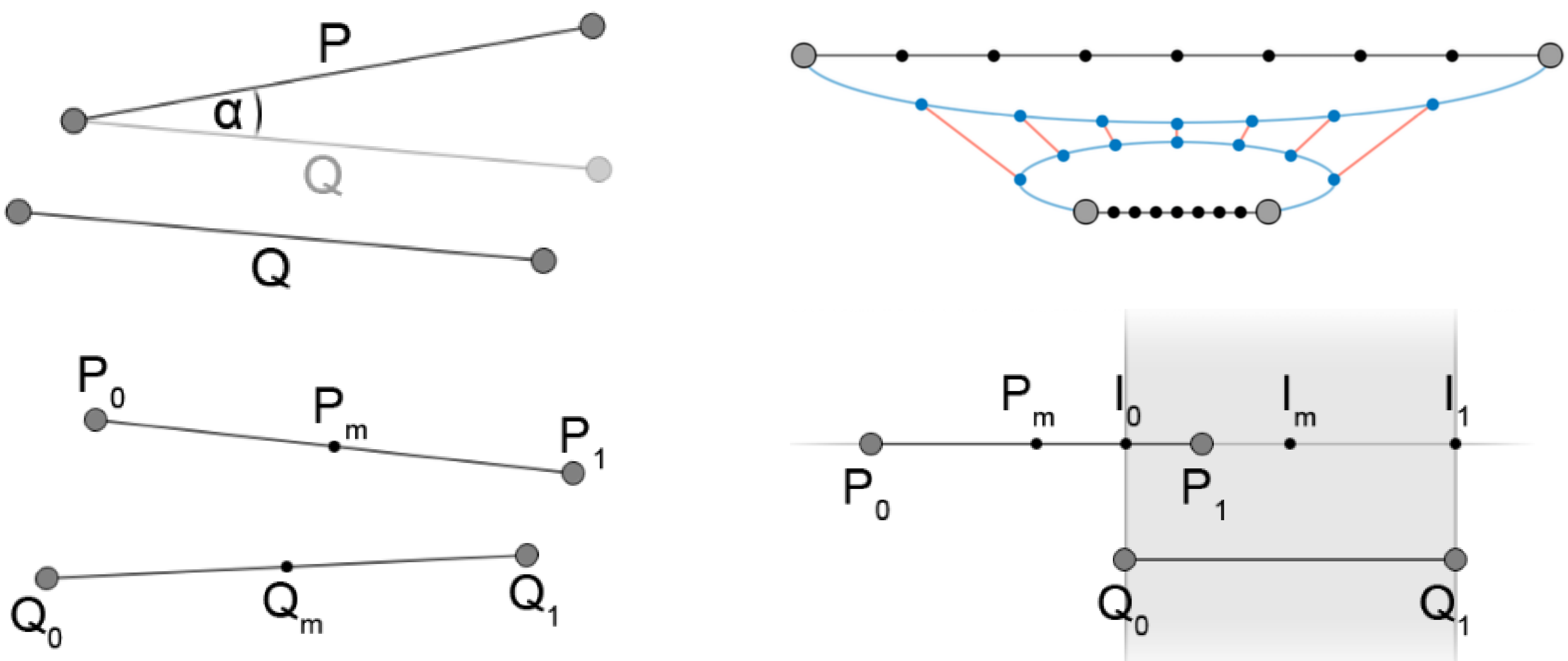
- it is possible that edges are parallel, equal in length, and close together, but should nevertheless have a low bundling
- these are the opposite edges of a skewed parallelogram
- *visibility compatibility*
  $C_v(P, Q) = \min(\text{vis}(P, Q), \text{vis}(Q, P))$, with
  $\text{vis}(P, Q) = \max(1 - \frac{2||P_m - I_m||}{||I_0 - I_1||}, 0)$, I_m – mid point of I_0 and I_1.
- $C_v(P, Q) = 1$ if P_m coincides with I_m (ideal position),
  $C_v(P, Q) = 0$ if P is outside the band of sight of Q





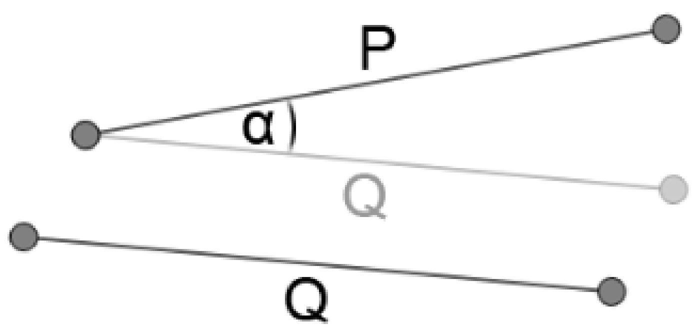
Edge compatibility measures: combined

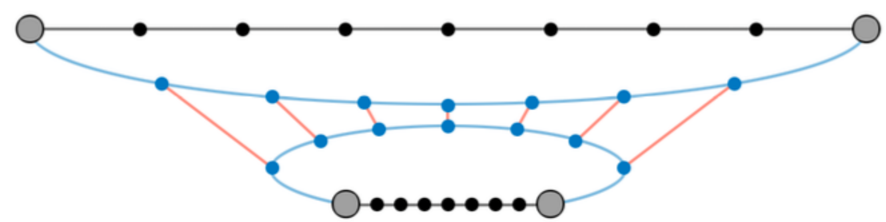
- The overall compatibility is defined as
  $C_e(P, Q) = C_a(P, Q) \cdot C_s(P, Q) \cdot C_p(P, Q) \cdot C_v(P, Q)$

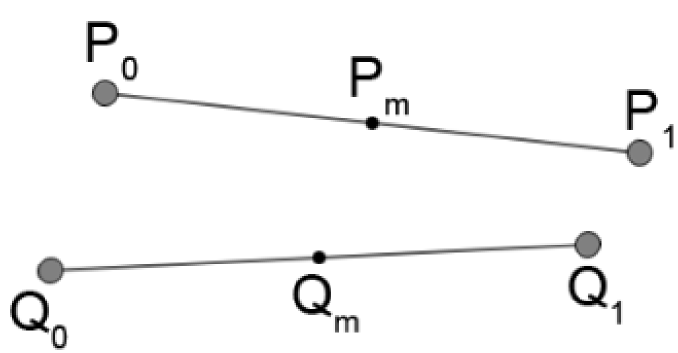


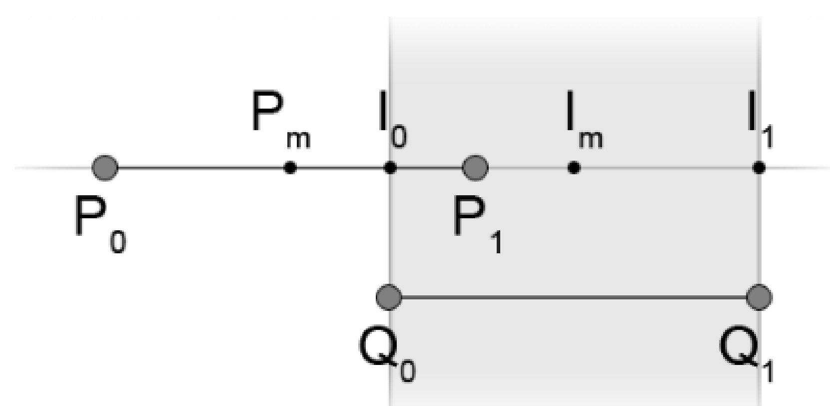
Edge compatibility measures: combined

- The overall compatibility is defined as
  $C_e(P, Q) = C_a(P, Q) \cdot C_s(P, Q) \cdot C_p(P, Q) \cdot C_v(P, Q)$
- The overall force on point p_i is then redefined as
  $F_{p_i} = k_P \cdot (||p_{i-1} - p_i)|| + ||p_i - p_{i+1}||) + \sum_{Q \in E} \frac{C_e(P,Q)}{||p_i - q_i||}$
  k_P – constant for edge P









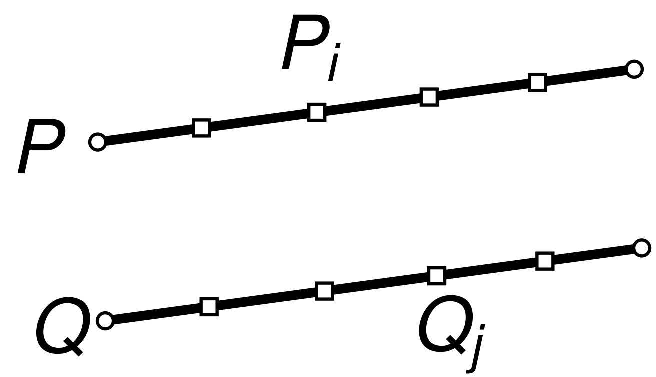
Edge bundling summary

Input: $G = (V, E)$ undirected graph with vertex placement,
number of cycles $C \in \mathbb{N}$,
number of iterations in the first cycle $I_0 \in \mathbb{N}$,
step size $s_0 \in \mathbb{N}$,
number of subdivision points in the first cycle n_0
interaction function $C_e : E \times E \to \mathbb{R}$
Output: Layout with bundled edges

$n \leftarrow n_0$ initial number of subdivisions
$t \leftarrow 1$ iteration counter
$I \leftarrow I_0$ number of iterations in the first cycle
$c \leftarrow 1$ cycle counter
$s \leftarrow s_0$ step size

Edge bundling summary

while $c < C$ **do**
- **foreach** $P \in E$ **do**
  - subdivide P by n points $P_1 \ldots P_n$; $B \leftarrow B \cup \bigcup_{P \in E} \{P_1 \ldots P_n\}$
- **foreach** $P \in E$ **do**
  - **foreach** $0 < i < n$ **do**
    - $F_{P_i} = k_P \cdot (||P_{i-1} - P_i)|| + ||P_i - P_{i+1}||)$
    - **foreach** $Q \neq P \in E$ **do**
      - **foreach** $0 < j < n$ **do**
        - $F_{P_i} = F_{P_i} + \frac{C_e(P,Q)}{||P_i - Q_j||}$;
- **foreach** $p \in B$ **do**
  - $p \leftarrow p + s \cdot F_p$
- $t \leftarrow t + 1$
- **if** $t == I$ **then**
  - $t \leftarrow 1$; $c \leftarrow c + 1$; $n \leftarrow 2n$;
  - $s \leftarrow s/2$; decrease(I);



Edge bundling: experiments

- experimentally the following values are suggested $n_0 = 1$, $s_0 = 0.04$, $C = 6$, and $l_0 = 50$.

Edge bundling: experiments

- experimentally the following values are suggested $n_0 = 1$, $s_0 = 0.04$, $C = 6$, and $I_0 = 50$.
- I is decreased by factor $\frac{2}{3}$

Edge bundling: experiments

- experimentally the following values are suggested $n_0 = 1$, $s_0 = 0.04$, $C = 6$, and $I_0 = 50$.
- I is decreased by factor $\frac{2}{3}$
- this gives the following

cycle	0	1	2	3	4	5
n	1	2	4	8	16	32
s	.04	.02	.01	.005	.0025	.00125
I	50	33	22	15	9	7

Edge bundling: experiments

- experimentally the following values are suggested $n_0 = 1$, $s_0 = 0.04$, $C = 6$, and $I_0 = 50$.
- I is decreased by factor $\frac{2}{3}$
- this gives the following

cycle	0	1	2	3	4	5
n	1	2	4	8	16	32
s	.04	.02	.01	.005	.0025	.00125
I	50	33	22	15	9	7

- performance can be significantly increased without compromising the bundling result by using a threshold for the total edge compatibility $C_e(P, Q)$

Edge bundling: experiments

- experimentally the following values are suggested $n_0 = 1$, $s_0 = 0.04$, $C = 6$, and $I_0 = 50$.
- I is decreased by factor $\frac{2}{3}$
- this gives the following

cycle	0	1	2	3	4	5
n	1	2	4	8	16	32
s	.04	.02	.01	.005	.0025	.00125
I	50	33	22	15	9	7

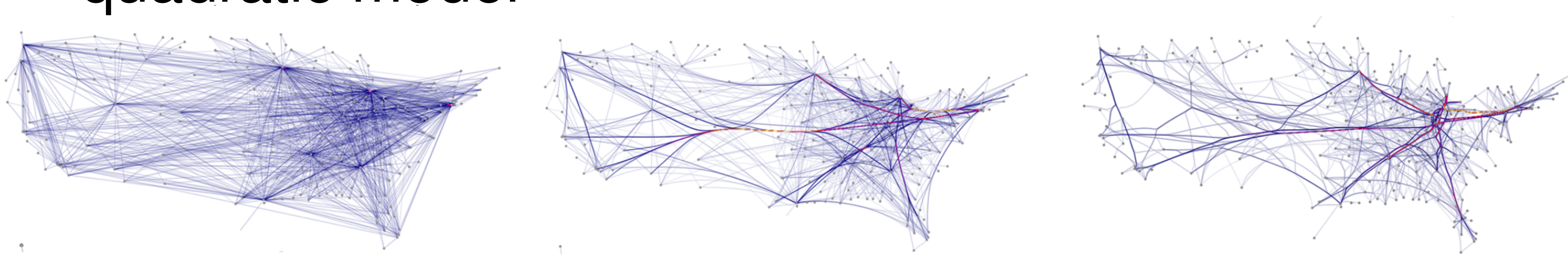
- performance can be significantly increased without compromising the bundling result by using a threshold for the total edge compatibility $C_e(P, Q)$
- edges P and Q are considered for calculation if $C_e(P, Q) > \lambda$, e.g. $\lambda = 0.05$

Edge bundling: experiments

- experimentally the following values are suggested $n_0 = 1$, $s_0 = 0.04$, $C = 6$, and $I_0 = 50$.
- I is decreased by factor $\frac{2}{3}$
- this gives the following

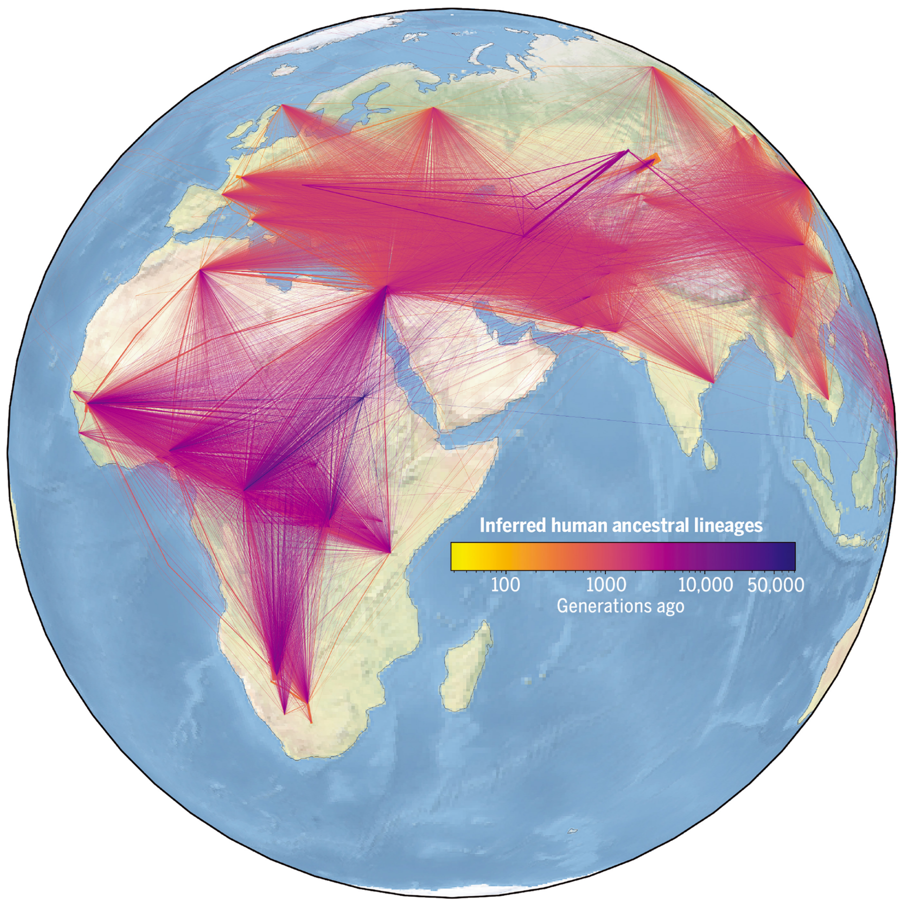
cycle	0	1	2	3	4	5
n	1	2	4	8	16	32
S	.04	.02	.01	.005	.0025	.00125
I	50	33	22	15	9	7

- performance can be significantly increased without compromising the bundling result by using a threshold for the total edge compatibility $C_e(P, Q)$
- edges P and Q are considered for calculation if $C_e(P, Q) > \lambda$, e.g. $\lambda = 0.05$
- US airlines graph with inverse linear and inverse quadratic model



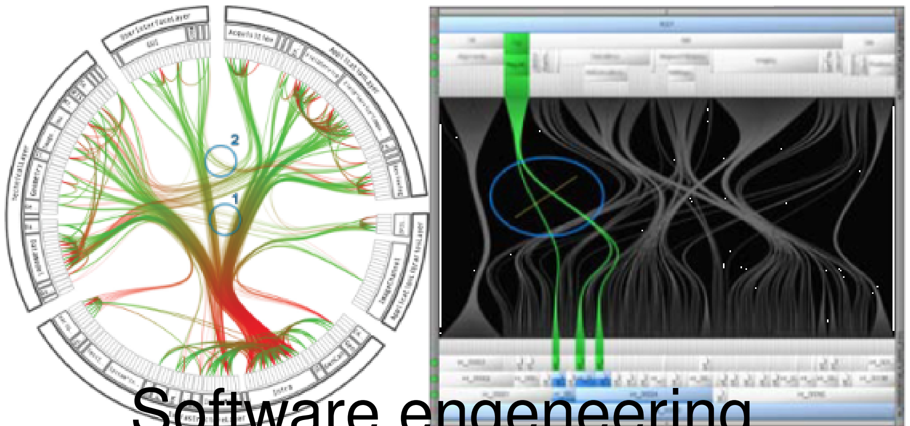
Edge bundling: inspiration

Inspiration: edges are ancestor-descendant relationship in the genealogy of modern and ancient genomes. Edge width – how many times the relationship is observed, color – age of the ancestor



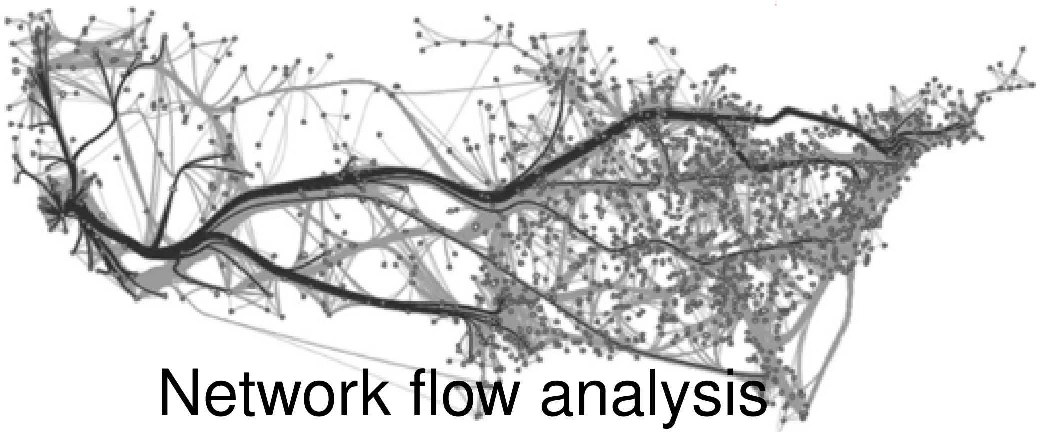
A unified genealogy of modern and ancient genomes, Wohns et al. Nature 2022

Edge bundling: discussion

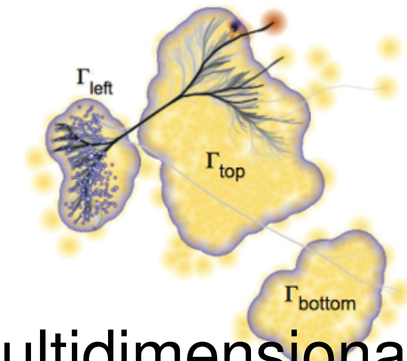


Software engeneering

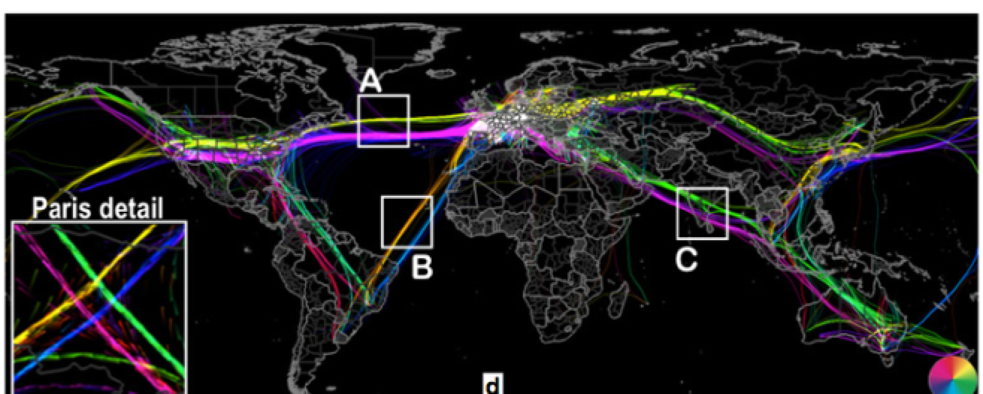
Bioinformatics



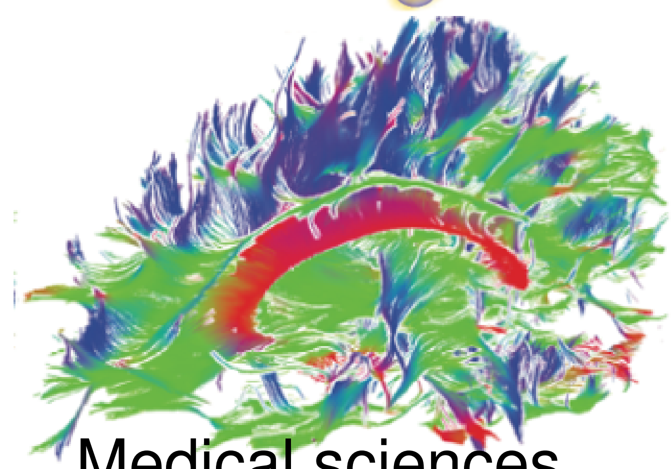
Network flow analysis



Multidimensional data

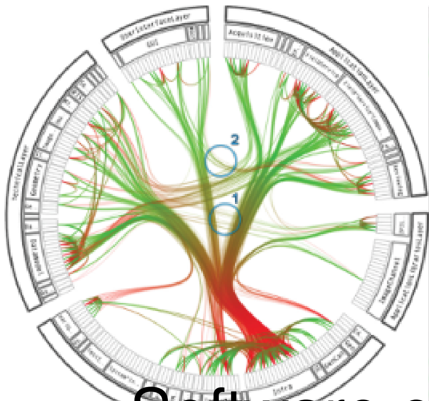


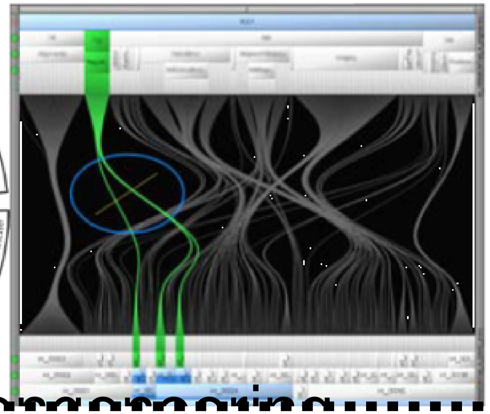
Air traffic control

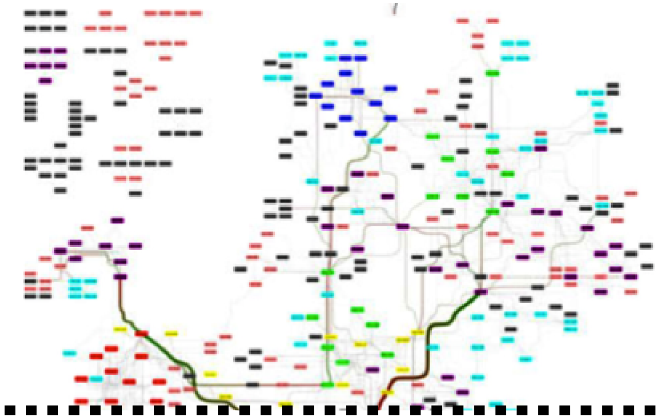


Medical sciences

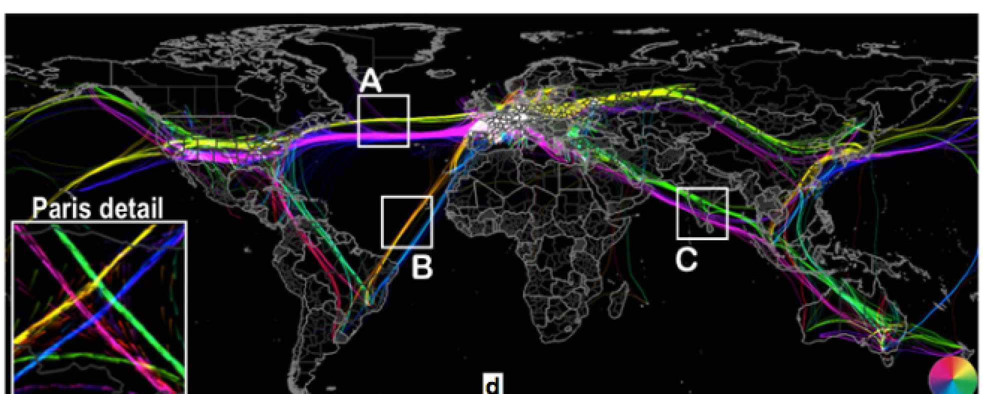
Edge bundling: discussion



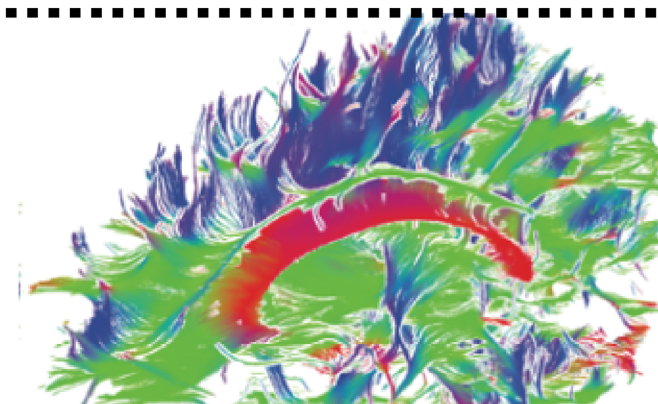




- What are the benefits and the drawbacks of the bundled layouts?
- When are the edge bundling techniques appropriate to use?



Air traffic control



Medical sciences

Tutorial task

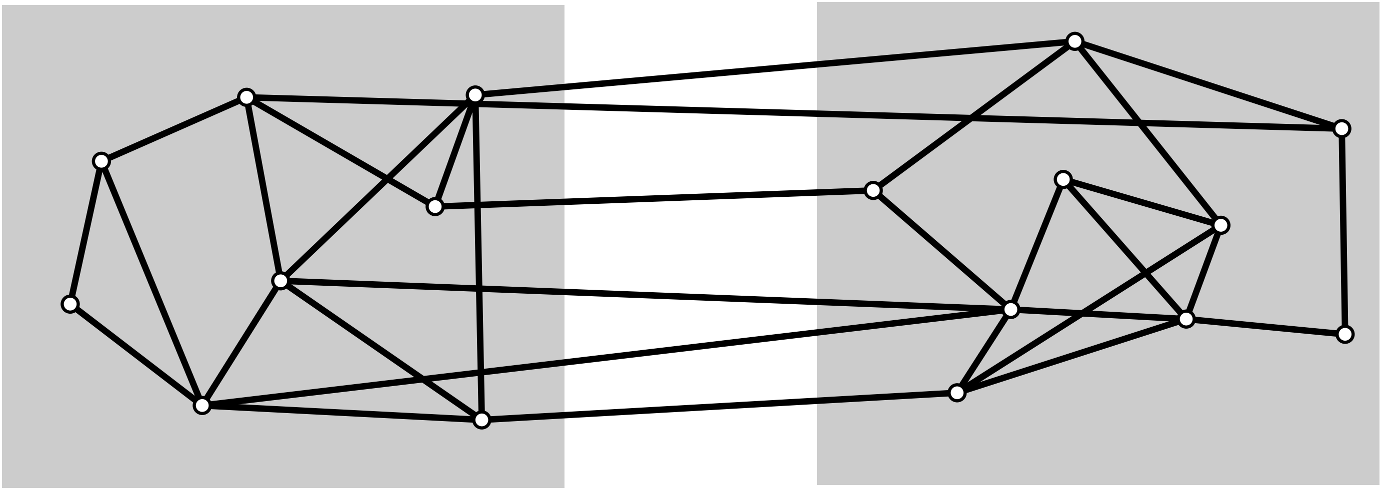
- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type

Tutorial task

- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type
- task: visualize a clustered graph – should work for two clusters

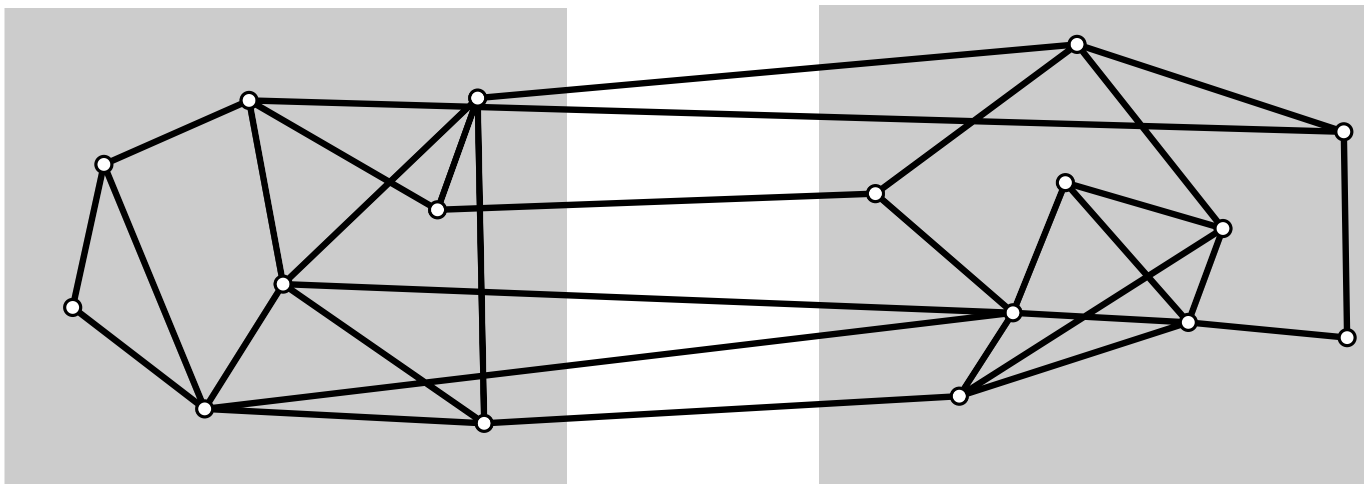
Tutorial task

- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type
- task: visualize a clustered graph – should work for two clusters



Tutorial task

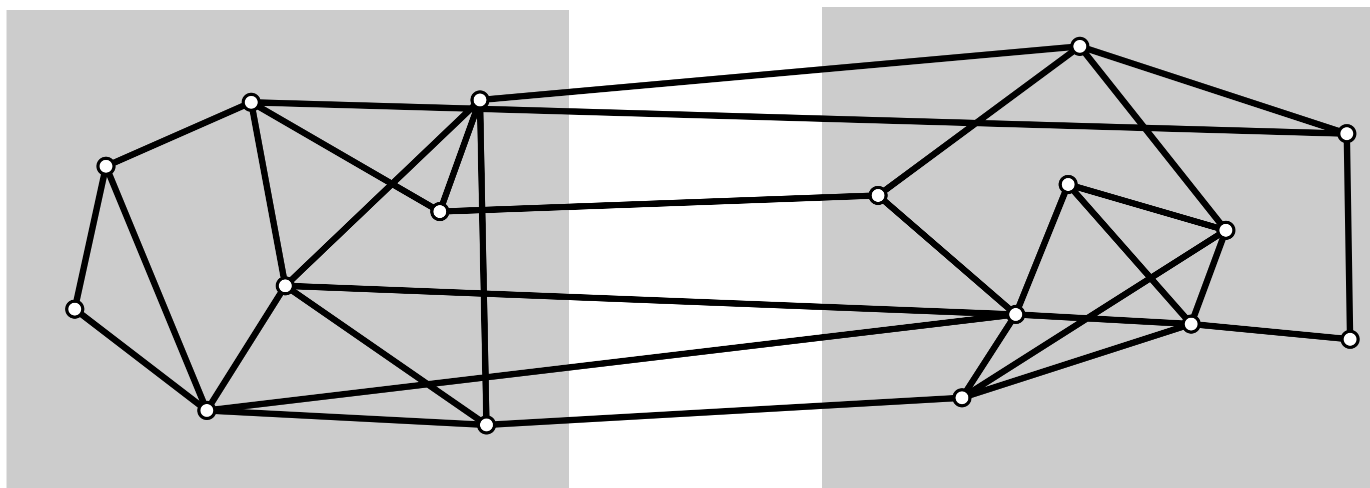
- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type
- task: visualize a clustered graph – should work for two clusters



- draw each cluster (layer) independently applying a force-directed implementation within a rectangle

Tutorial task

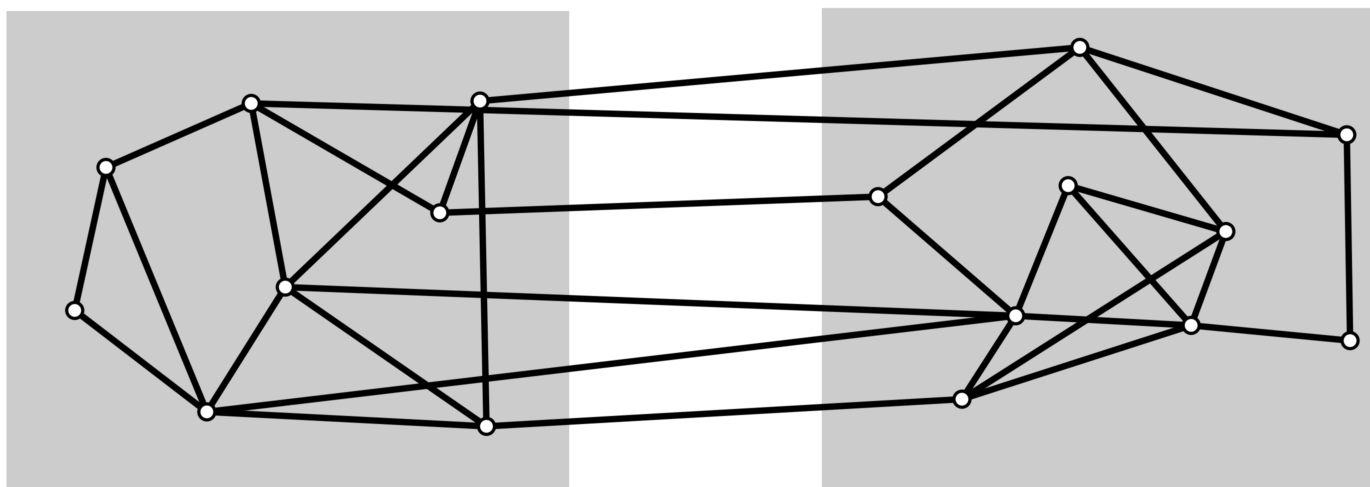
- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type
- task: visualize a clustered graph – should work for two clusters



- draw each cluster (layer) independently applying a force-directed implementation within a rectangle
- place the two rectangles at some distance (parameter)

Tutorial task

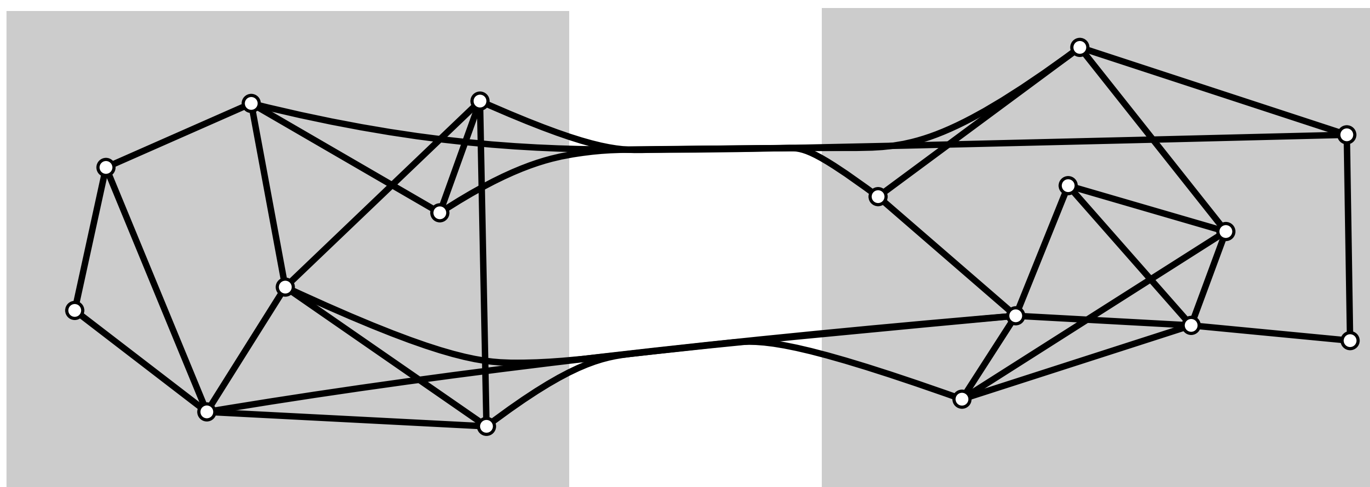
- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type
- task: visualize a clustered graph – should work for two clusters



- draw each cluster (layer) independently applying a force-directed implementation within a rectangle
- place the two rectangles at some distance (parameter)
- implement the bundling algorithm and apply it to inter-cluster (inter-layer) edges

Tutorial task

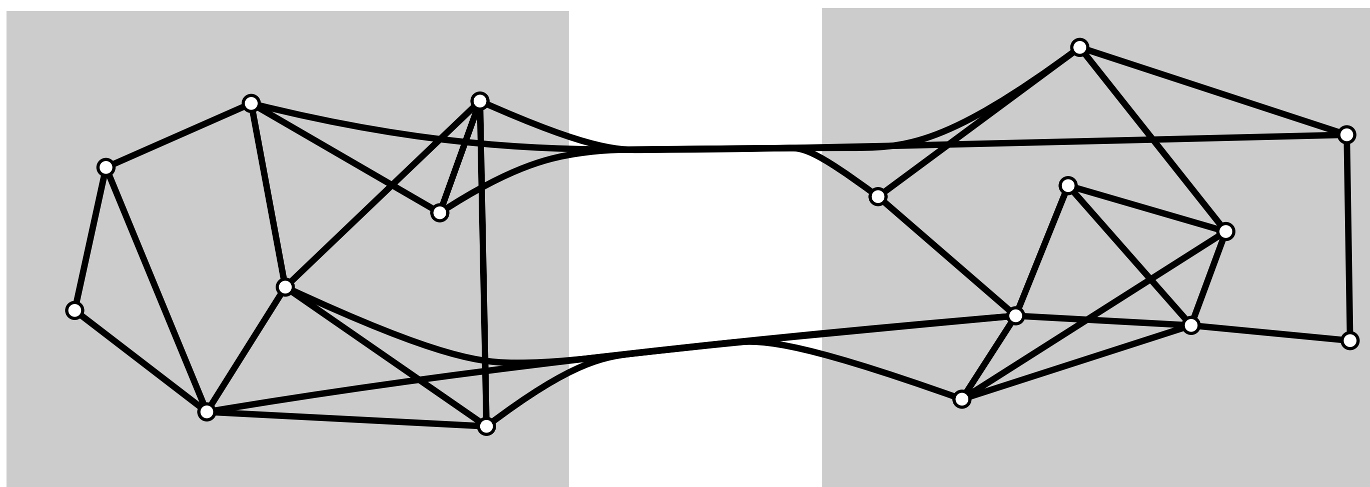
- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type
- task: visualize a clustered graph – should work for two clusters



- draw each cluster (layer) independently applying a force-directed implementation within a rectangle
- place the two rectangles at some distance (parameter)
- implement the bundling algorithm and apply it to inter-cluster (inter-layer) edges

Tutorial task

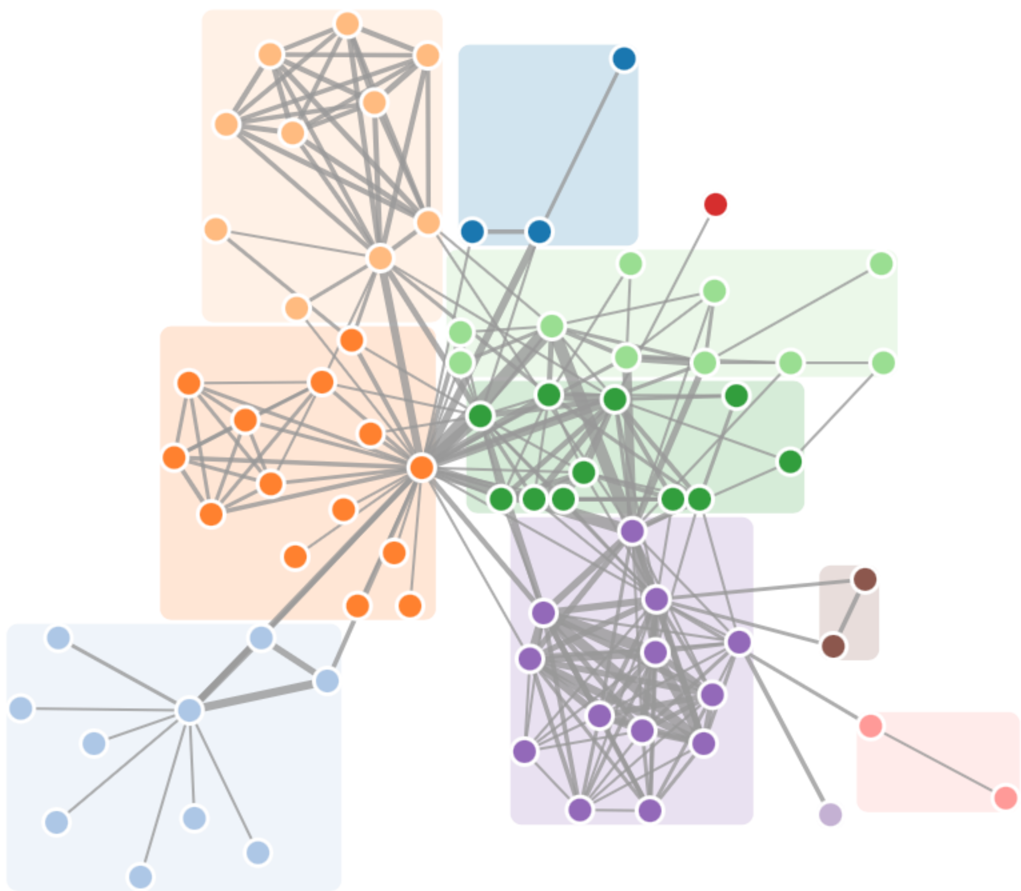
- *clustered graphs* : each node appears in a single layer (cluster) and relationships between nodes are of the same type
- task: visualize a clustered graph – should work for two clusters



- draw each cluster (layer) independently applying a force-directed implementation within a rectangle
- place the two rectangles at some distance (parameter)
- implement the bundling algorithm and apply it to inter-cluster (inter-layer) edges
- experiment with political blogosphere, argument network (besides the two clusters, nodes and edges have different types)

Tutorial task (bonus)

- expand your method to work for many layers/clusters
- you need to find a way to arrange an arbitrary number of boxes – inspiration `cola.js`, `yEd`



Reading and Next

Additional Reading

Paper “The State of the Art in Multilayer Network Visualization” (F. McGee, M. Ghoniem, G. Melancon, B. Otjacques and B. Pinaud)

Paper “Force Directed Edge Bundling” (D. Holten, J. J. van Wijk)

Reading and Next

Additional Reading

Paper “The State of the Art in Multilayer Network Visualization” (F. McGee, M. Ghoniem, G. Melancon, B. Otjacques and B. Pinaud)

Paper “Force Directed Edge Bundling” (D. Holten, J. J. van Wijk)

Next

	March 6	-	Visualization of multilevel networks	Tamara	
11	March 11	-	Tutorial: Step 5	Alister	Step 4
	March 13	-	High-dimensional data visualization	Alex	
12	March 18	-	Tutorial: Step 6	Alister	Step 5
	March 20	-	High-dimensional data visualization: advanced	Alex	
13	March 25	-	Tutorial: Step 7	Alister	Step 6
	March 27	-	—		
14	April 2	17:15-19:00	Final Presentations	Students	Step 7
	April 3	9:00-12:45	Final Presentations	Students	
	April 8				Final deliverables

Final Remarks

- For all organizational questions please ask me (course coordinator)

Final Remarks

- For all organizational questions please ask me (course coordinator)
- Please give us feedback (official, unofficial) - give details and be constructive so that we can improve!

Final Remarks

- For all organizational questions please ask me (course coordinator)
- Please give us feedback (official, unofficial) - give details and be constructive so that we can improve!
- If you like network visualization (algorithms, computational experiments, user studies) – ask me for a topic of master’s thesis!

Final Remarks

- For all organizational questions please ask me (course coordinator)
- Please give us feedback (official, unofficial) - give details and be constructive so that we can improve!
- If you like network visualization (algorithms, computational experiments, user studies) – ask me for a topic of master’s thesis!
- Possible direction: use of the machine learning methods for generation of better graph layouts.

Final Remarks

- For all organizational questions please ask me (course coordinator)
- Please give us feedback (official, unofficial) - give details and be constructive so that we can improve!
- If you like network visualization (algorithms, computational experiments, user studies) – ask me for a topic of master's thesis!
- Possible direction: use of the machine learning methods for generation of better graph layouts.
- Possible direction: just check `https://evm.science.uu.nl` !

Final Remarks

- For all organizational questions please ask me (course coordinator)
- Please give us feedback (official, unofficial) - give details and be constructive so that we can improve!
- If you like network visualization (algorithms, computational experiments, user studies) – ask me for a topic of master's thesis!
- Possible direction: use of the machine learning methods for generation of better graph layouts.
- Possible direction: just check `https://evm.science.uu.nl` !

koeln-marathon.de