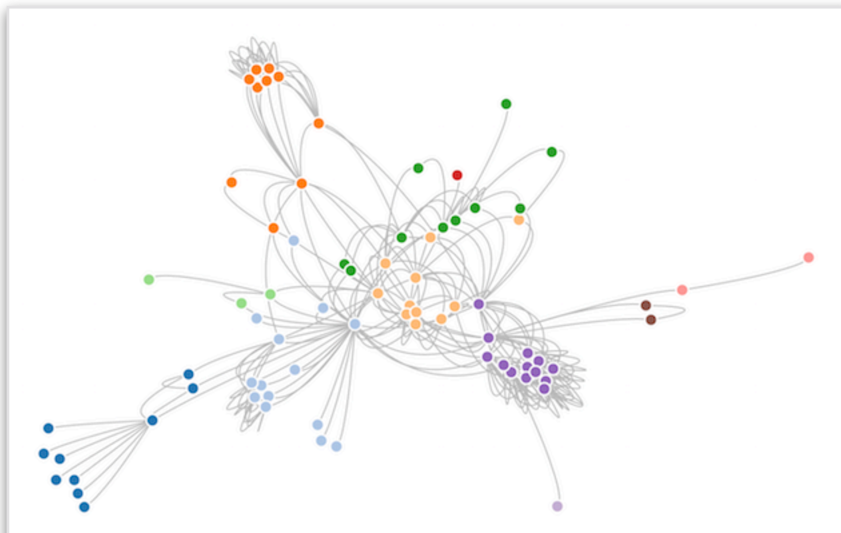
Data Visualization

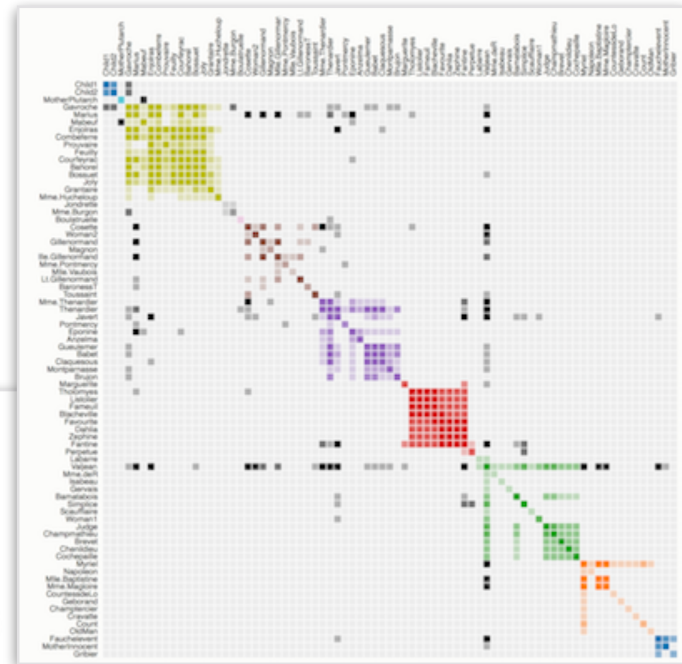
INF552 (2023-2024)

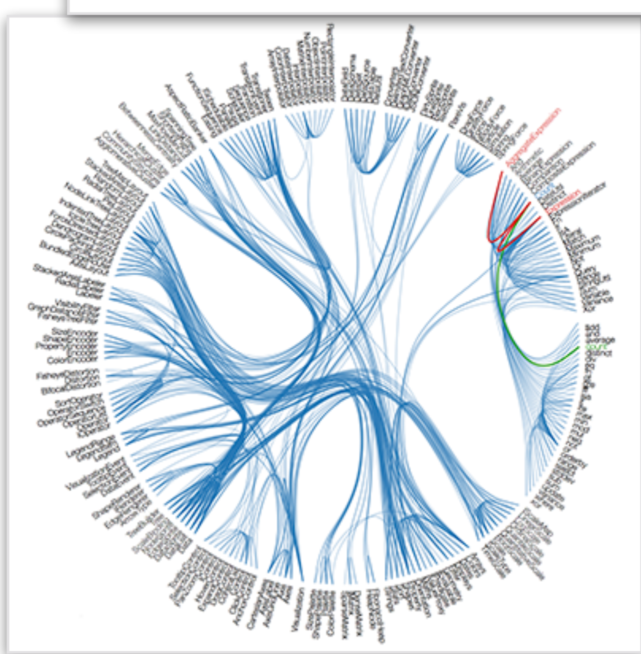
Session 07
Visualization of Hierarchical Structures and Graph Structures (Part I)

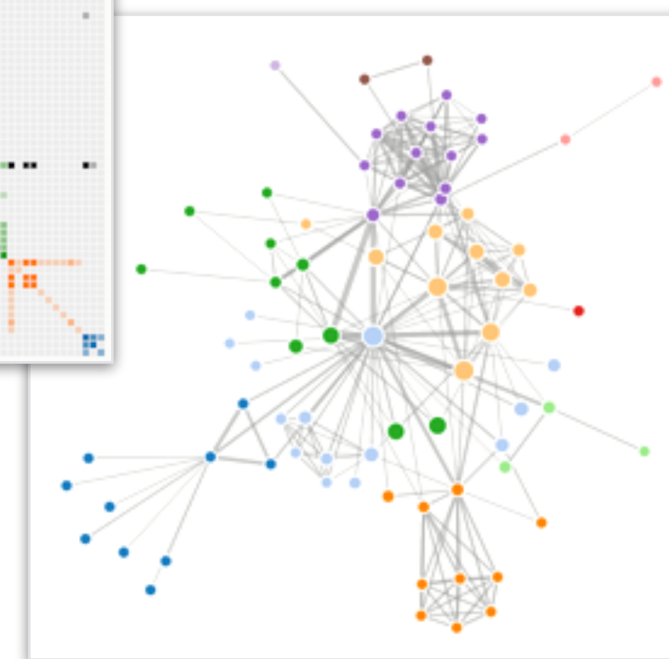
emmanuel.pietriga@inria.fr

Networks





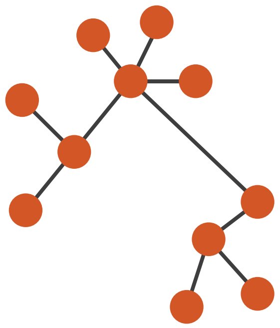




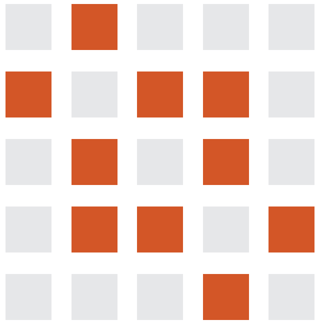
[Image source: d3js.org]

Visual representation

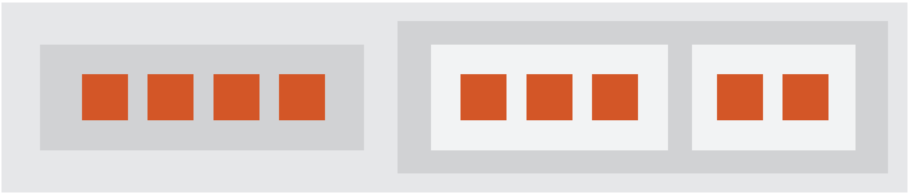
Node–Link Diagrams
Connection Marks
✔ NETWORKS ✔ TREES



Adjacency Matrix
Derived Table
✔ NETWORKS ✔ TREES

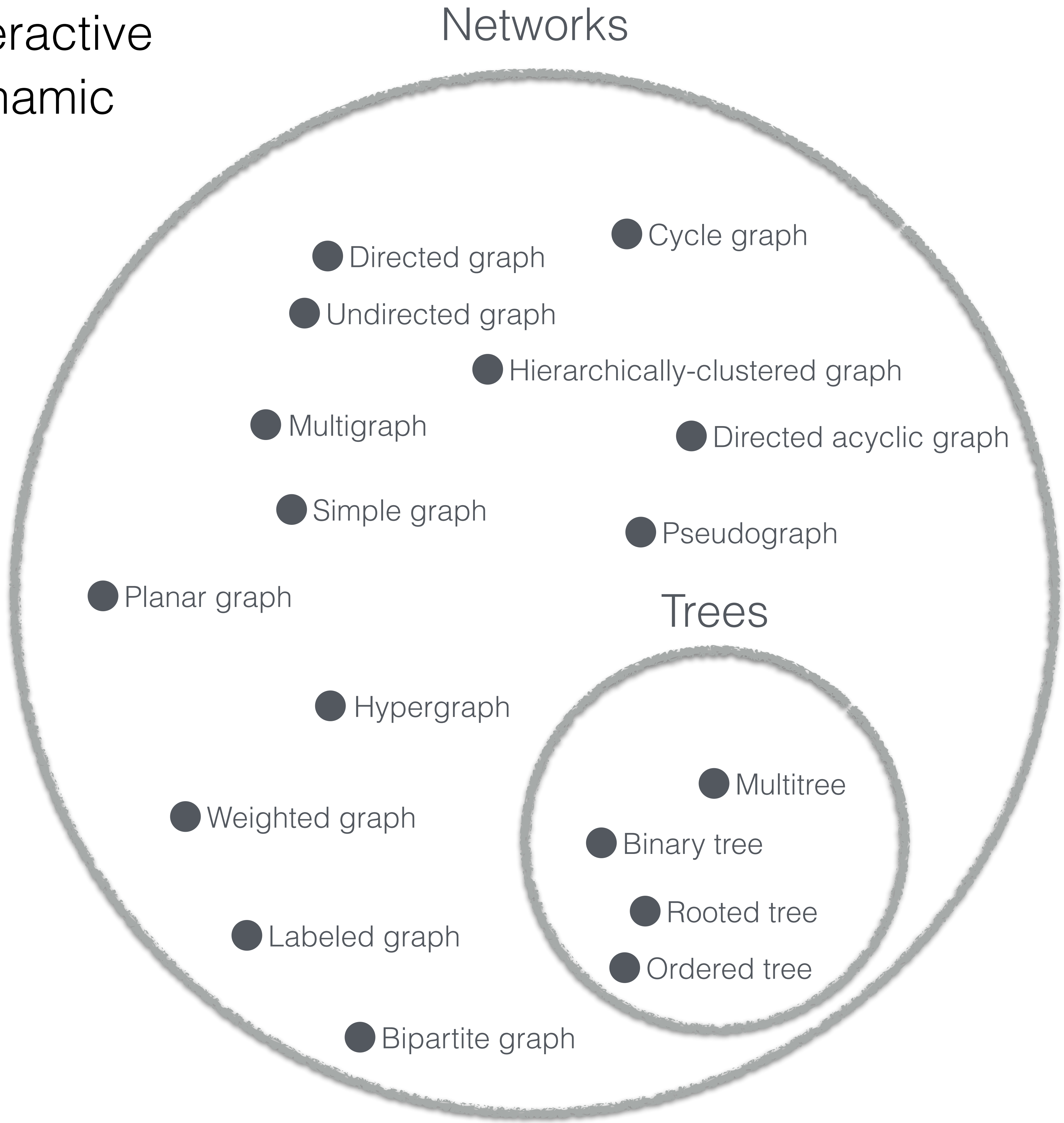


Enclosure
Containment Marks
✖ NETWORKS ✔ TREES

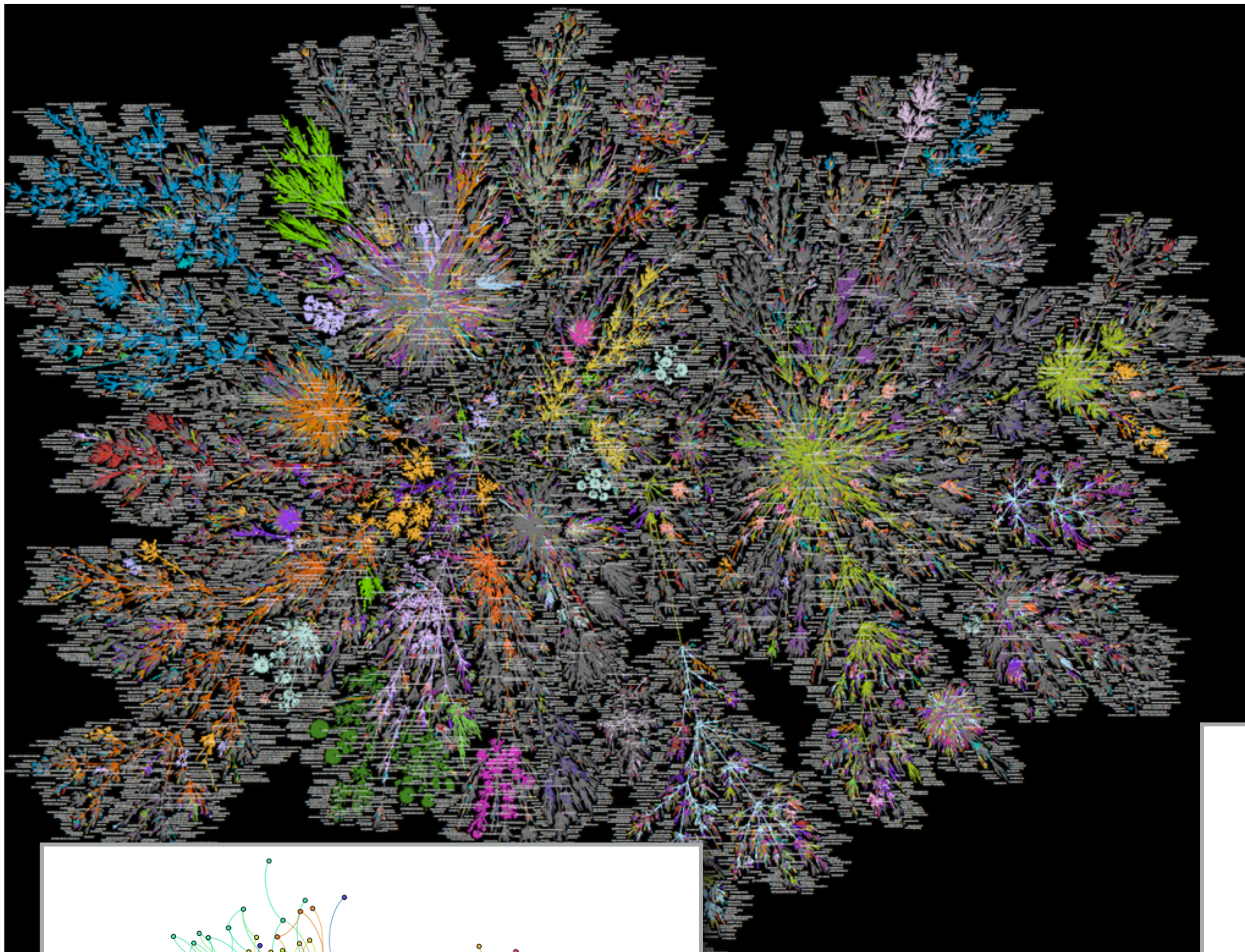


[Image source: Visualization Analysis and Design. Tamara Munzner, with illustrations by Eamonn Maguire. A K Peters Visualization Series, CRC Press, 2014]

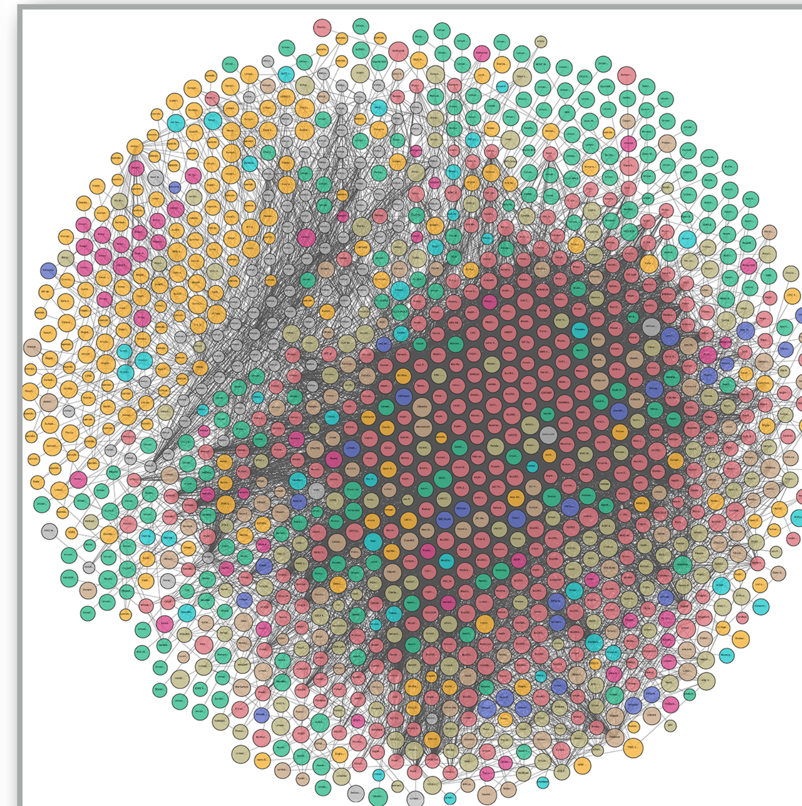
Static *vs.* interactive
Static *vs.* dynamic

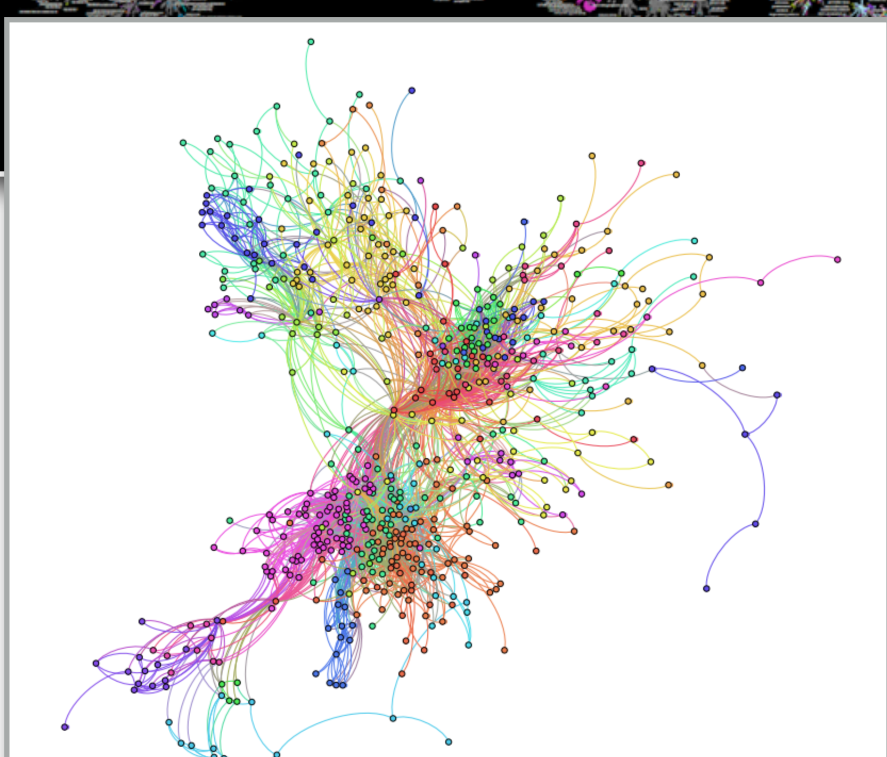


Communication networks, Power grid, Software engineering, KR,…

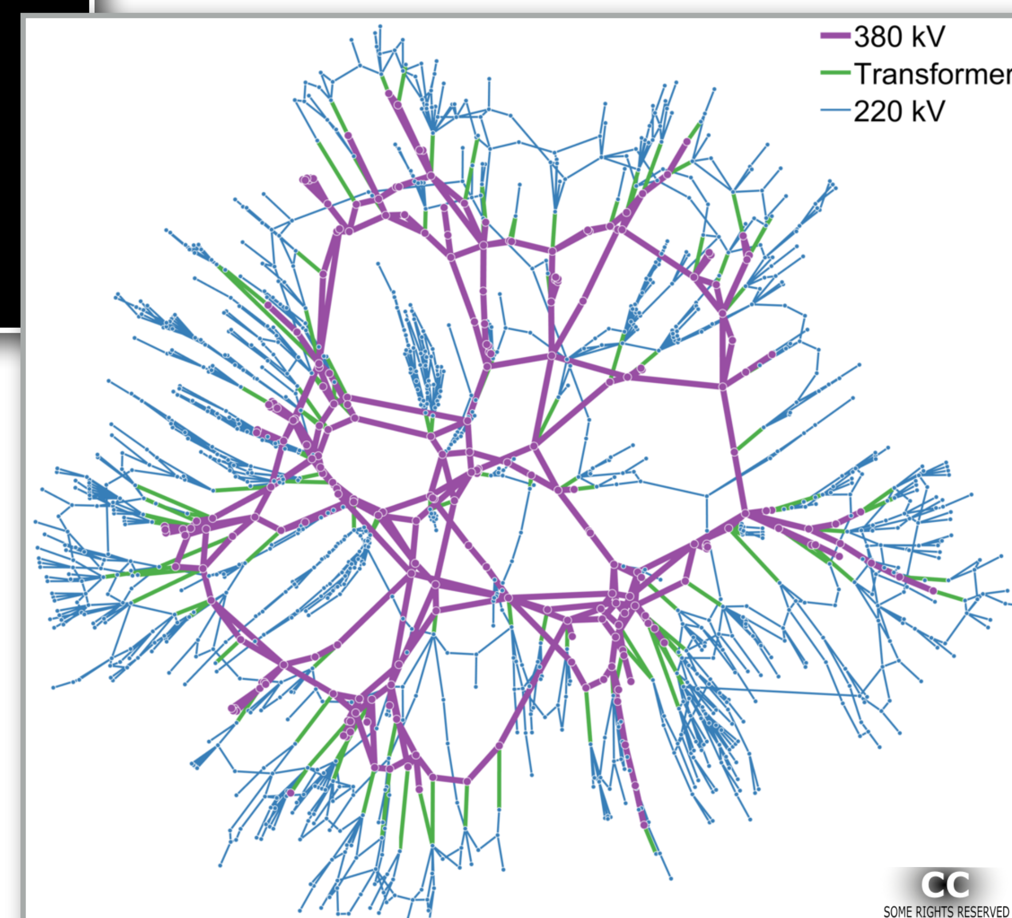


From lod-cloud.net





[Zanetti, Marcelo Serrano, and Frank Schweitzer. "A network perspective on software modularity." ARCS Workshops. IEEE, 2012.]



[Cuffe, P. and Keane, A., 2017. Visualizing the electrical structure of power systems. IEEE Systems Journal, 11(3), pp.1810-1821. https://doi.org/10.1109/JSYST.2015.2427994]

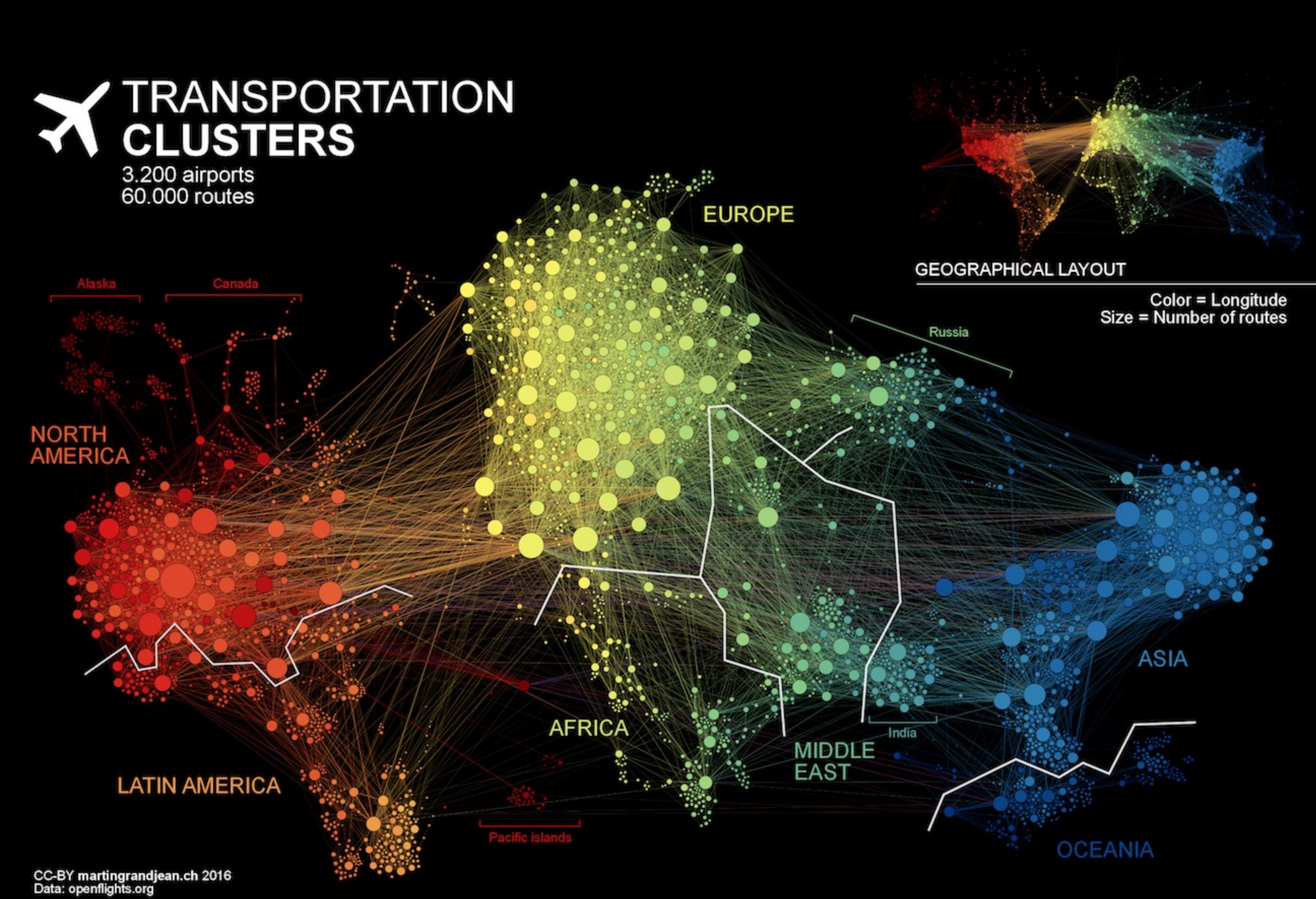
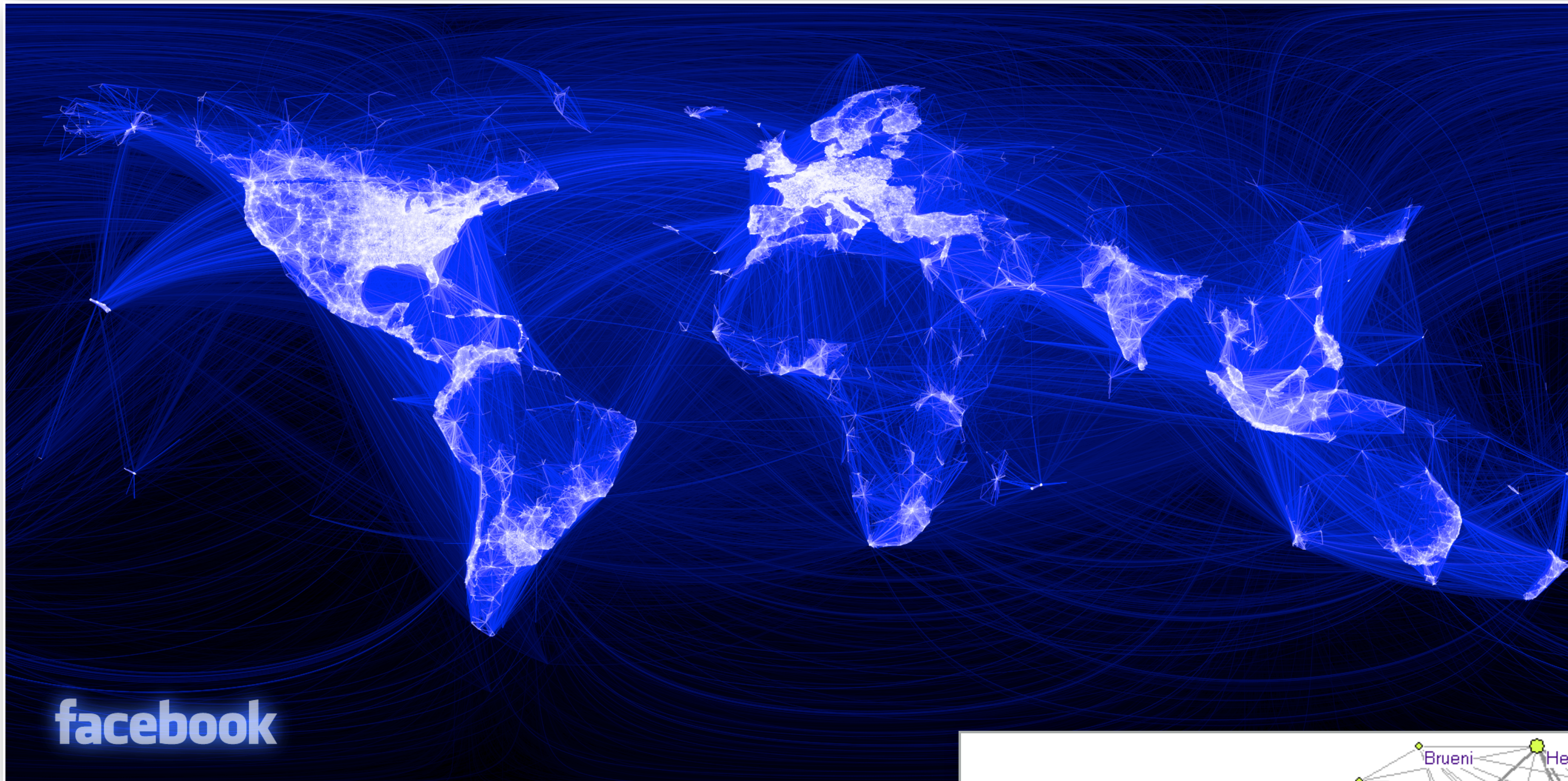
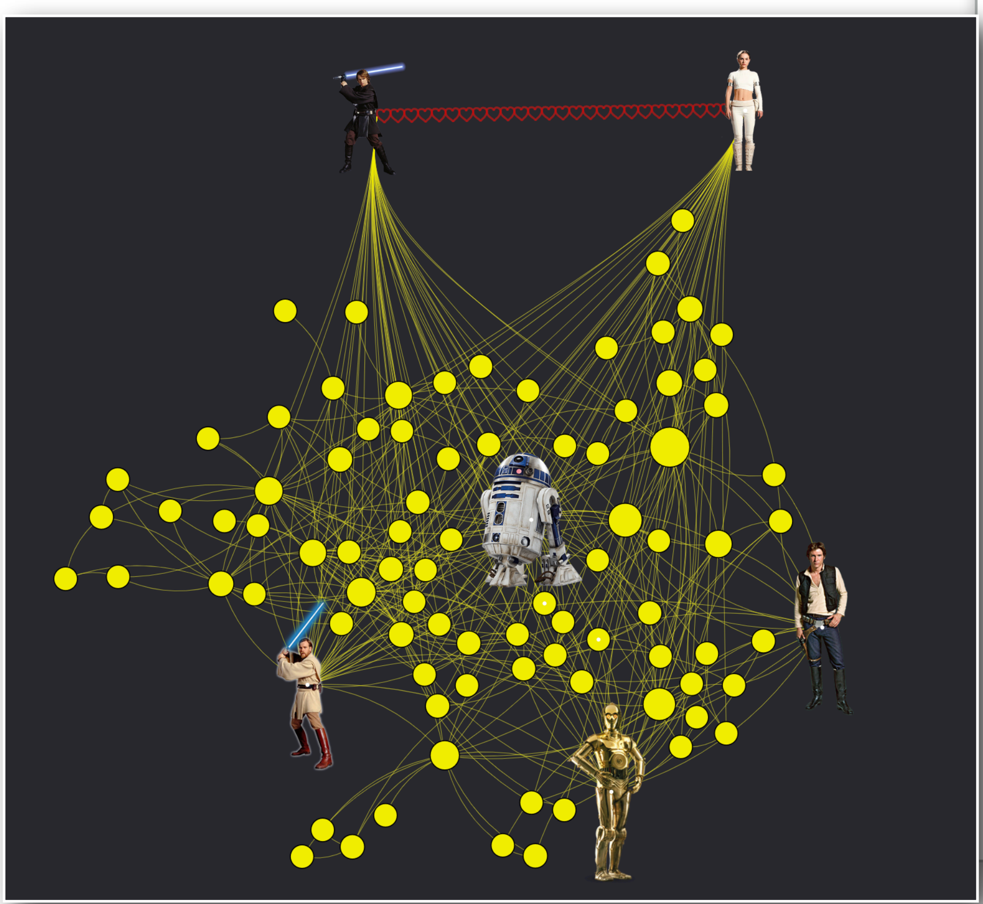

TRANSPORTATION
CLUSTERS
3.200 airports
60.000 routes
EUROPE
GEOGRAPHICAL LAYOUT
Color = Longitude
Size = Number of routes
Alaska
Canada
Russia
NORTH
AMERICA
AFRICA
MIDDLE
EAST
India
ASIA
LATIN AMERICA
Pacific islands
OCEANIA
CC-BY martingrandjean.ch 2016
Data: openflights.org


Схема метро
в 2100 году

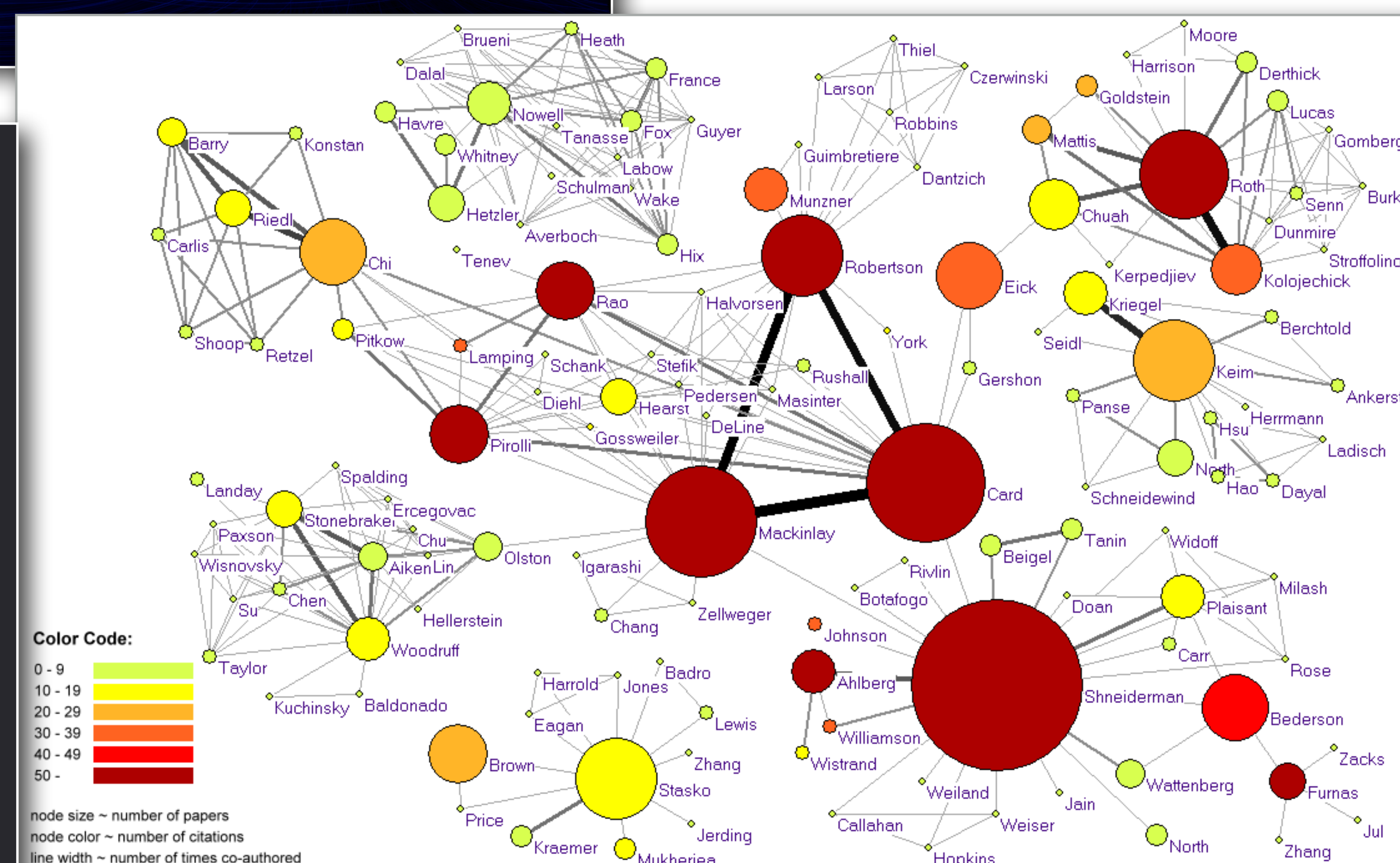
Social networks, trade/financial networks



[Visualizing friendships, by Paul Butler @ facebook, 2010
https://www.facebook.com/note.php?note_id=469716398919]

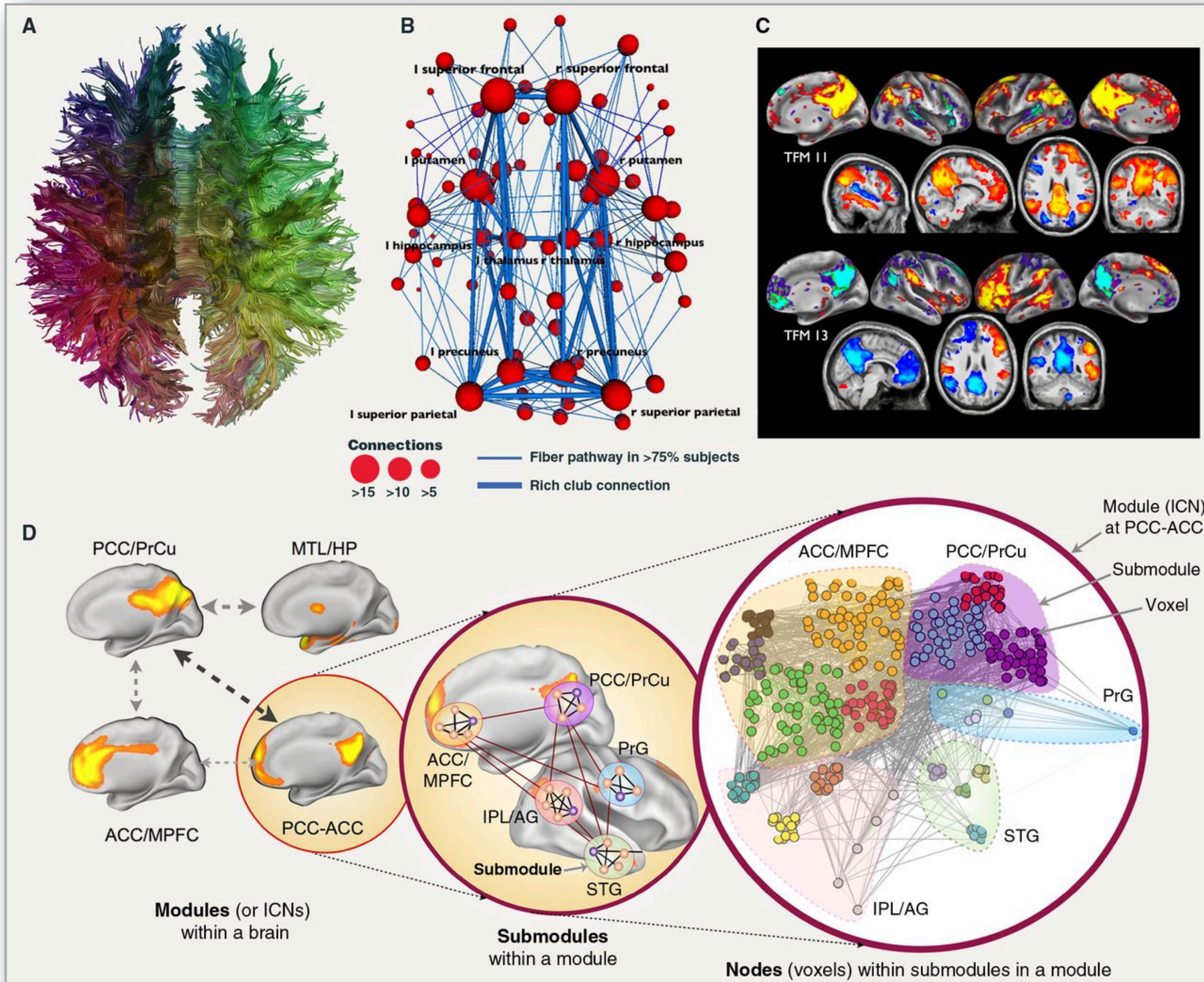


[H. Romat et al., Expressive Authoring of Node-Link Diagrams with Graphies, IEEE Transactions on Visualization and Computer Graphics (TVCG), 2019]

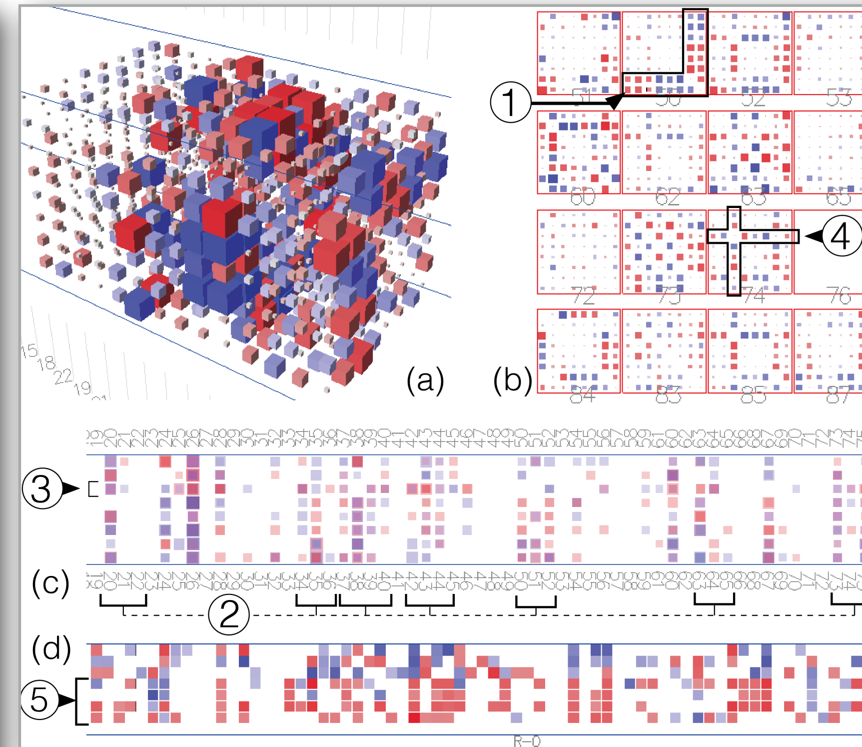


[Weimao Ke, Katy Borner and Lalitha Viswanath, Analysis and Visualization of the IV 2004 Contest Dataset]

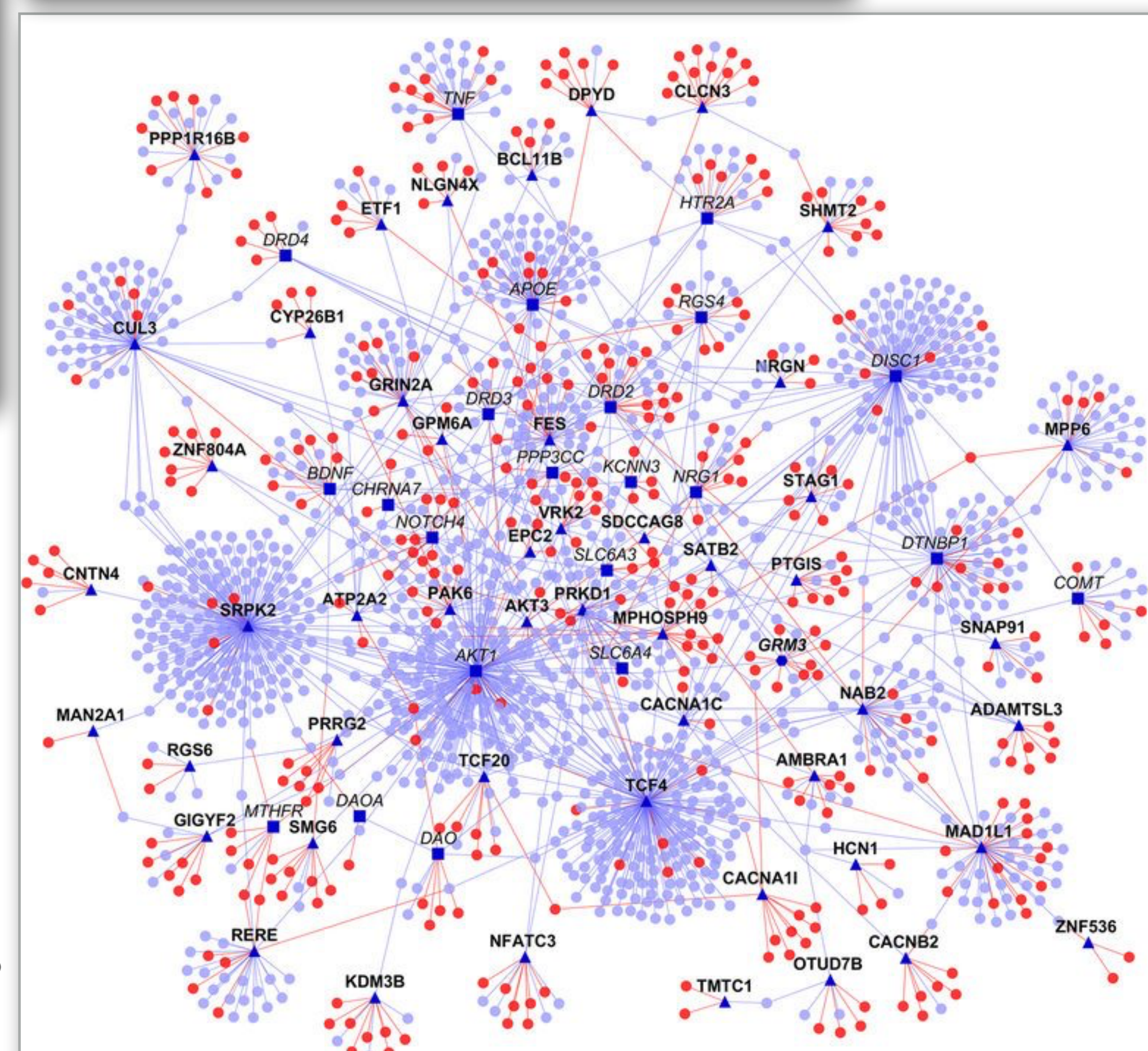
Life sciences



[Park, H. J., & Friston, K. (2013). Structural and functional brain networks: from connections to cognition. Science, 342(6158), 1238411.]



[B. Bach et al.. 2014. Visualizing dynamic networks with matrix cubes. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM 877–886. DOI:https://doi.org/10.1145/2556288.2557010]



[Ganapathiraju MK et al. (2016). "Schizophrenia interactome with 504 novel protein-protein interactions". NPJ Schizophrenia. 2: 16012. doi:10.1038/npjschz.2016.12]

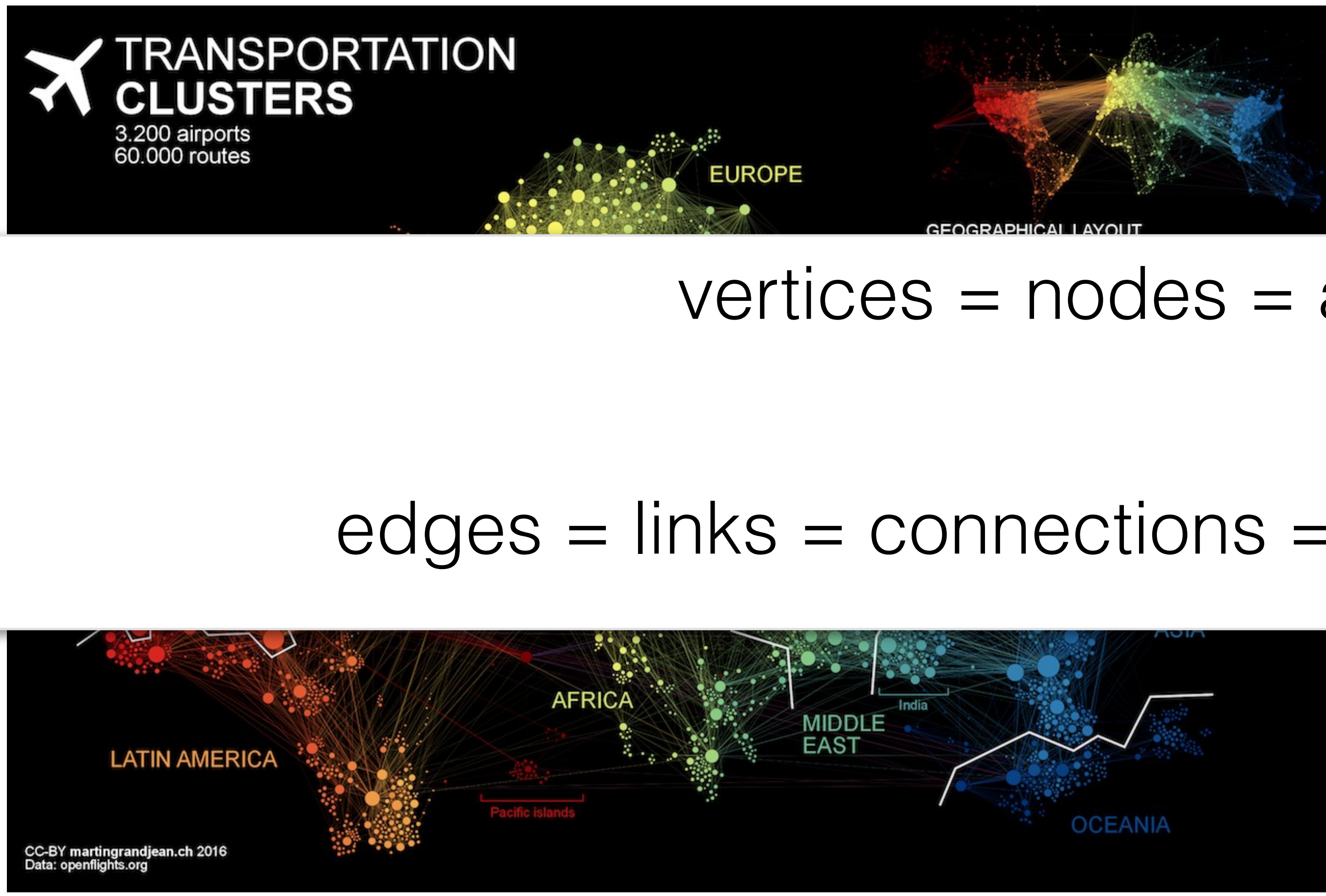
Definitions

- Informal: a network is made of things that are related.
- (more formal): *"a system of vertices connected in pairs by edges"*
- Formal:
  - a graph $G = < V, E >$
  - where $V = v_i$ is a set of vertices
  - and $E = e_{i,j}$ is a (multi-)set of edges with $e_{i,j} = \{v_i, v_j\}$

Definitions

- *Graph*: topology

- *Network*: *Graph* + vertex & edge attributes (multi-variate data)



Vertices (airports):
- number of passengers
- number of airlines

vertices = nodes = actors

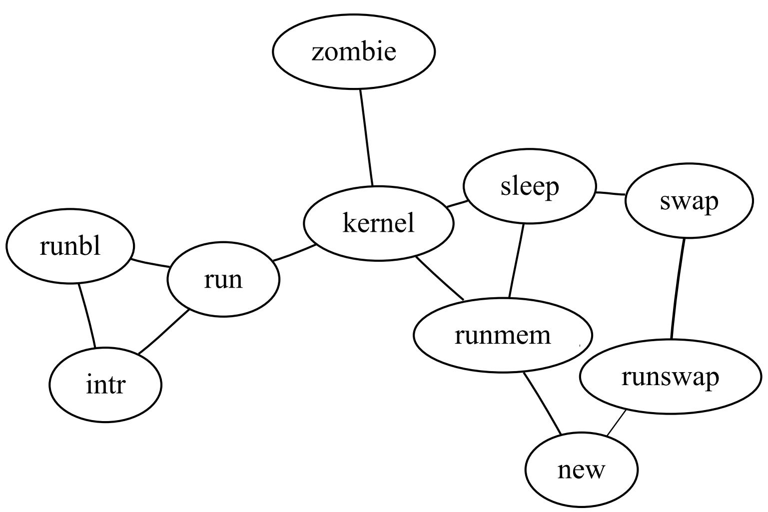
edges = links = connections = relations = arcs

- Vertices & edges *vs.* nodes & links

Graph types: directed vs. undirected graph, mixed

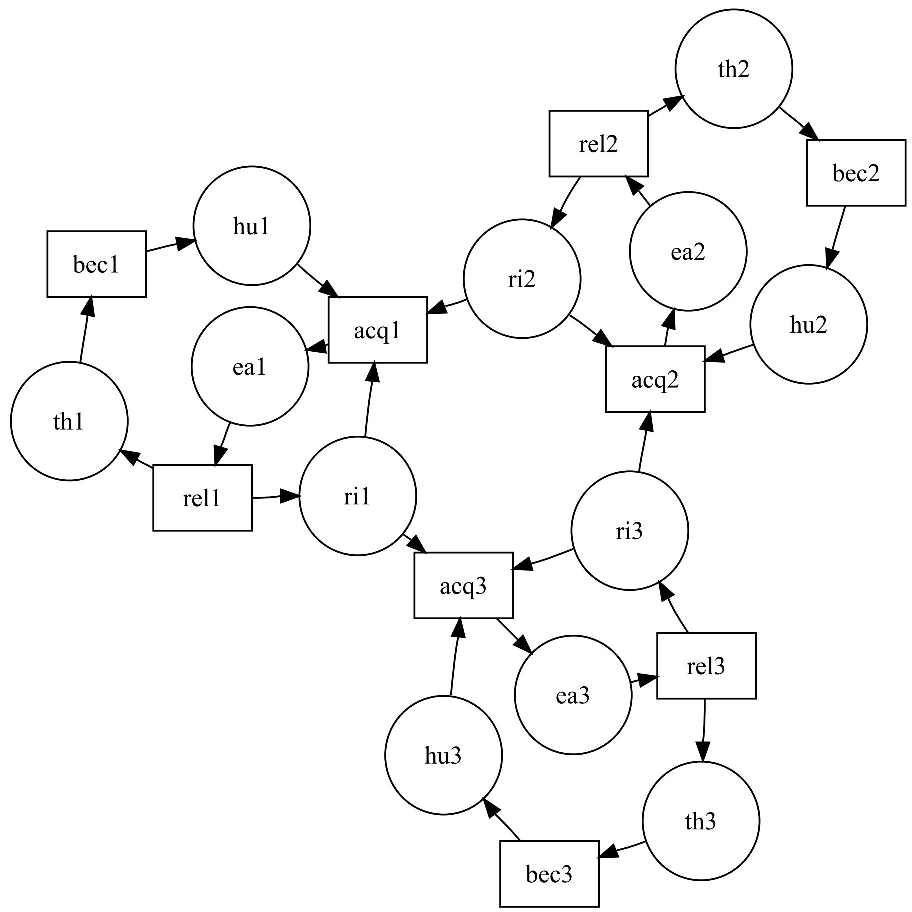
$v_i v_j$

$[v_i, v_j]$



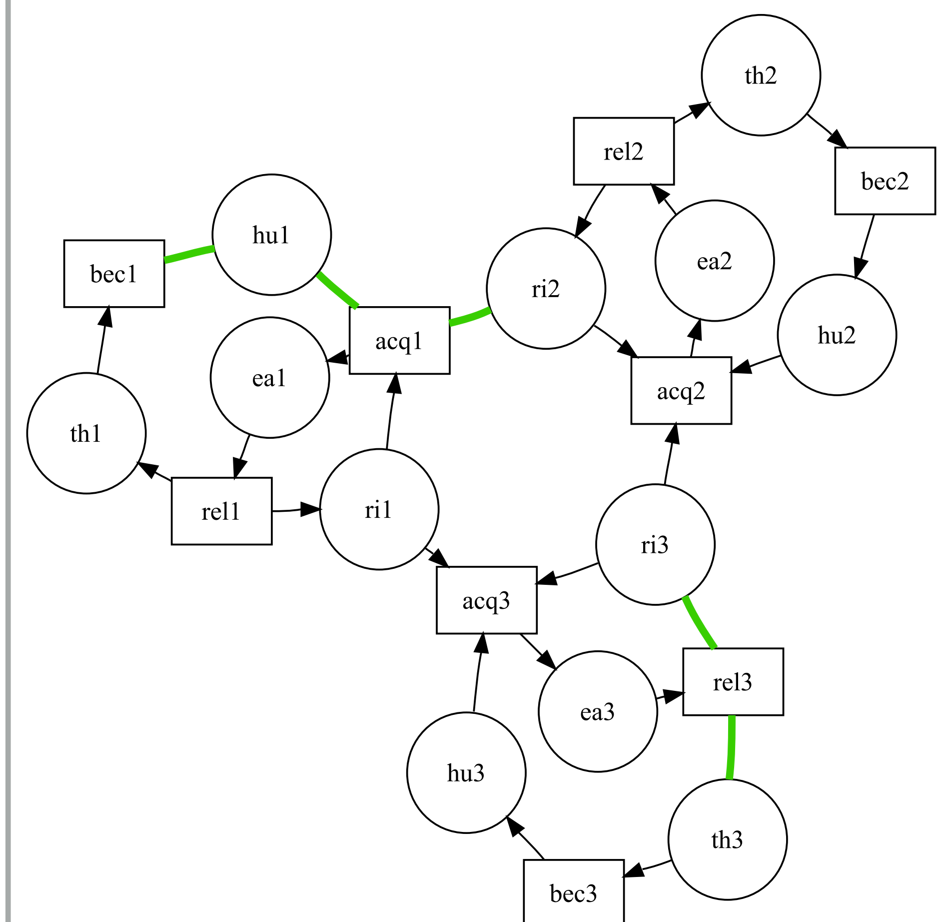
$\overrightarrow{v_i v_j}$

(v_i, v_j)



$G = < V, E, A >$

$e_i e_j$ $\overrightarrow{a_k a_l}$



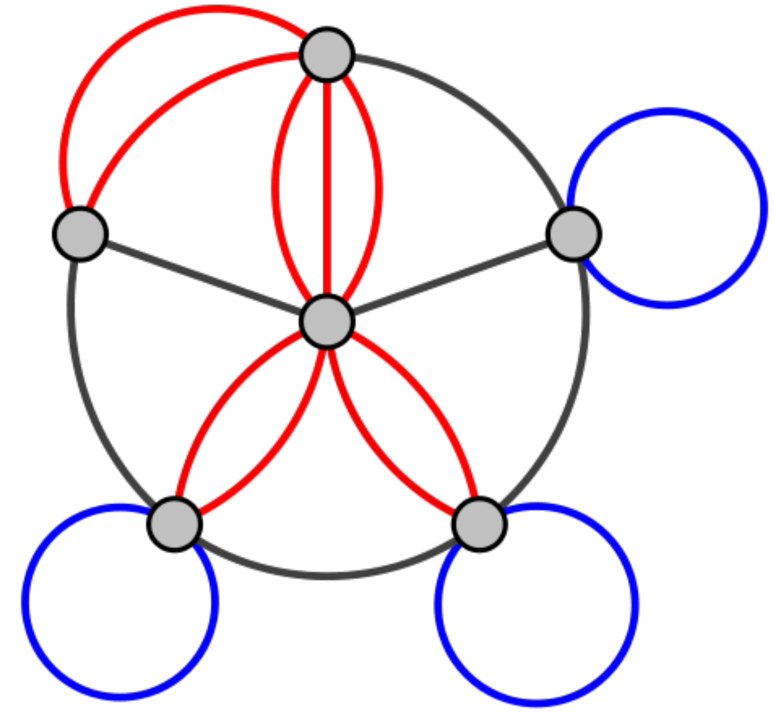
[Examples from http://graphviz.org/gallery/]

Graph types: multigraph, pseudograph, weighted graph

- Two or more edges incident to the same vertices:
  multi-edges, parallel edges

- Multigraph $G = < V, E >$
  multiset

- Pseudograph = multigraph allowing loops



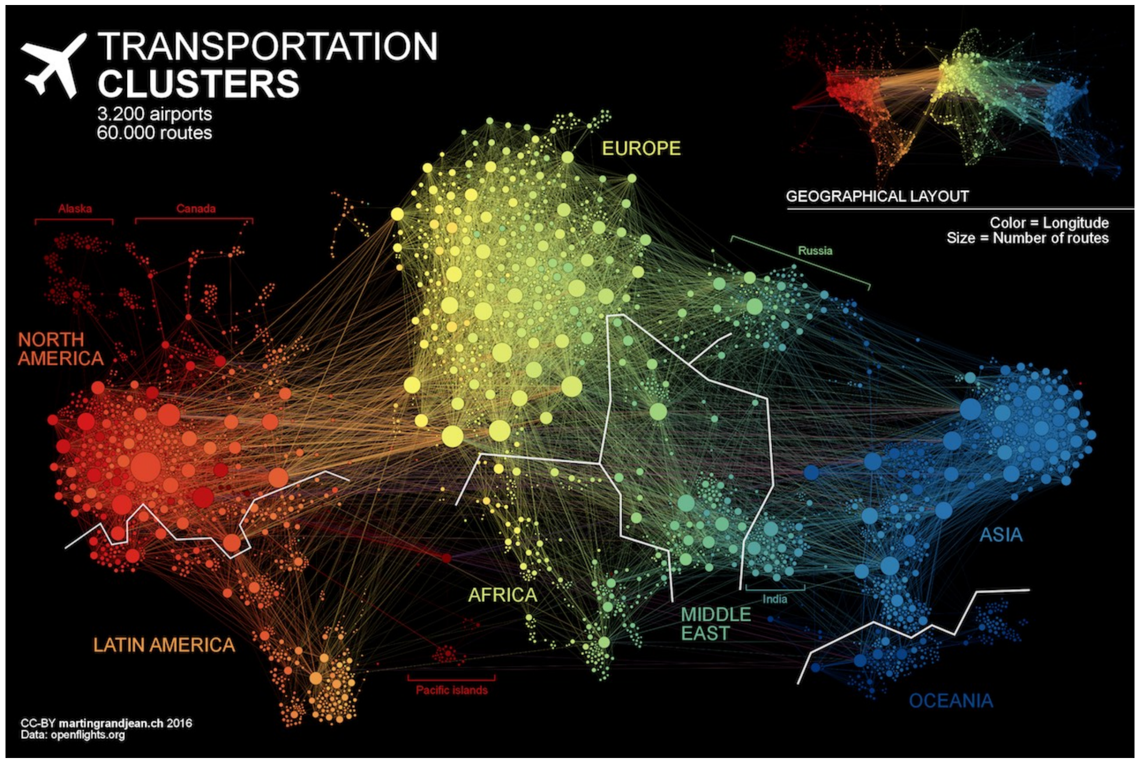
[Image Source: https://commons.wikimedia.org/wiki/File:Multi-pseudograph.svg]

- Graph labelling: assignment of labels to the vertices or edges of the graph

- Case of edge labels in an ordered set: weighted graph

Vertices (airports):
- number of passengers
- number of airlines
- number of runways
- …

Edges (routes):
- number of passengers
- number of flights
- …



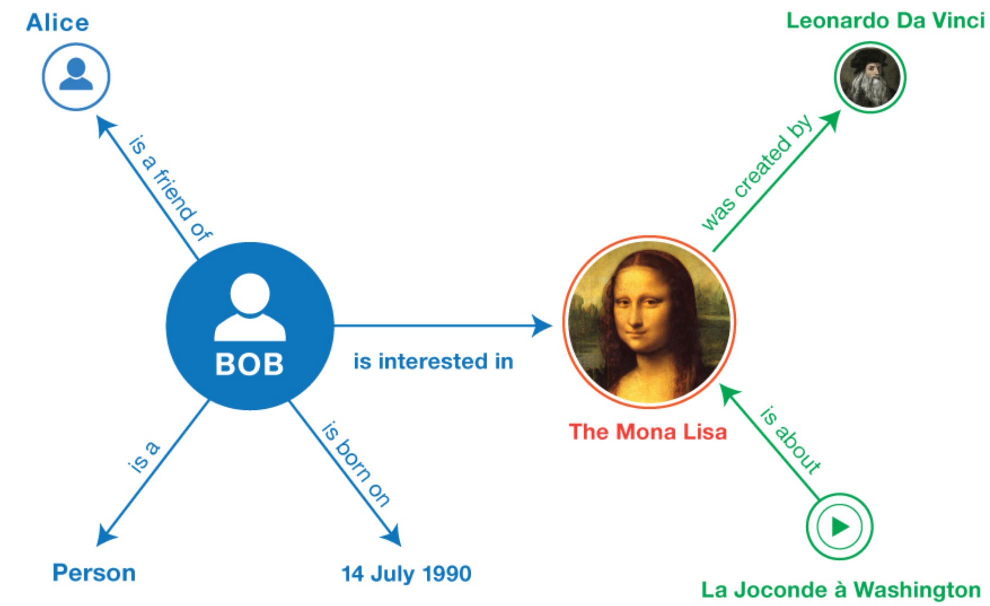
Graph types: labeled graph

- Edge labels in a categorical set often define the nature of the relation => multiple types of links

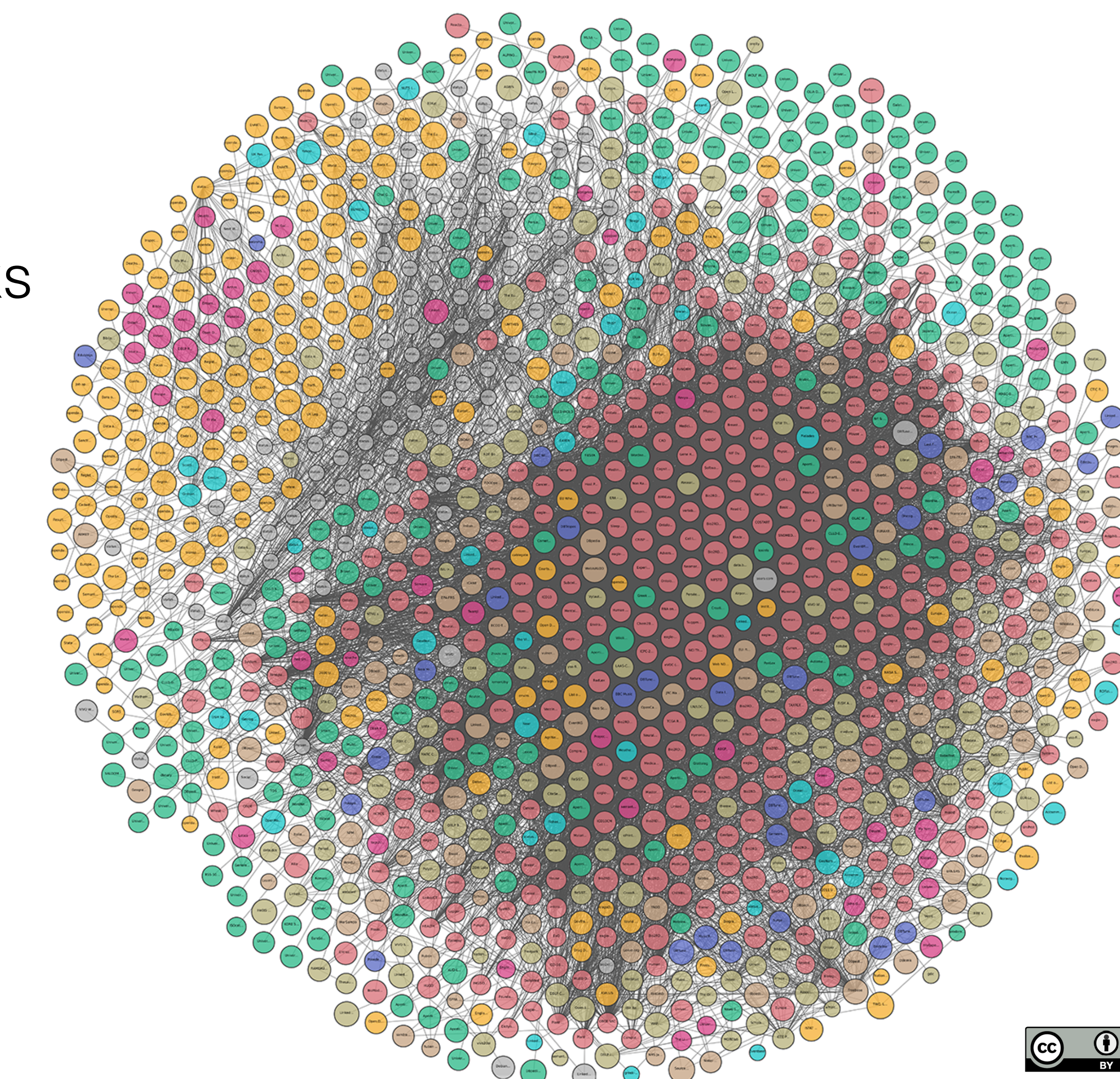
$$G = \langle V, E, D \rangle$$

$$e_{i,j} = (v_i, v_j, d) \quad v_i, v_j \in V, d \in D$$

- Typical of knowledge representation frameworks such as ontology-backed RDF datasets



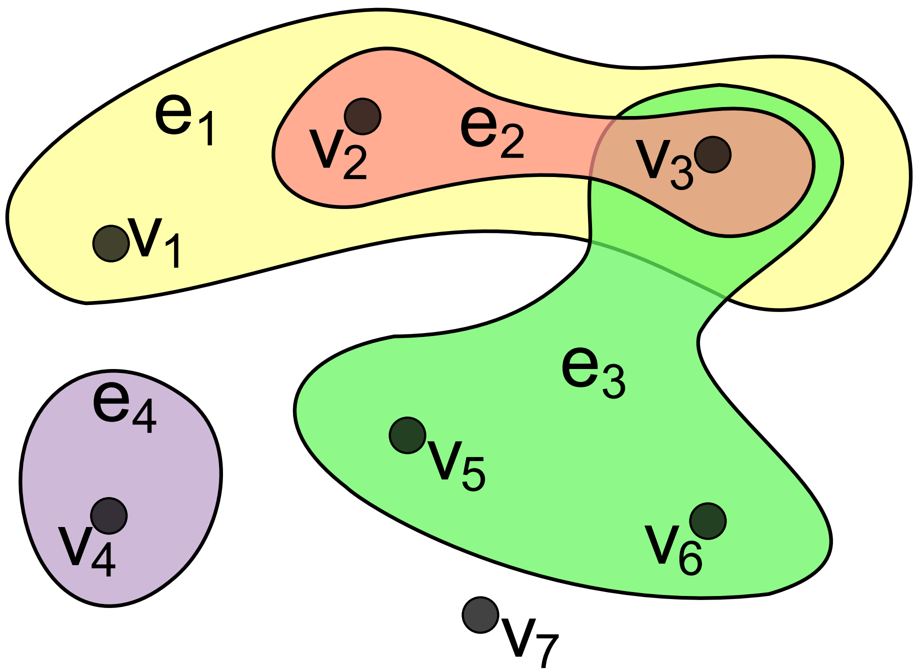
[Source: https://www.w3.org/TR/rdf11-primer/]



From lod-cloud.net

Graph types: hypergraph

- hyperedges join any number of vertices

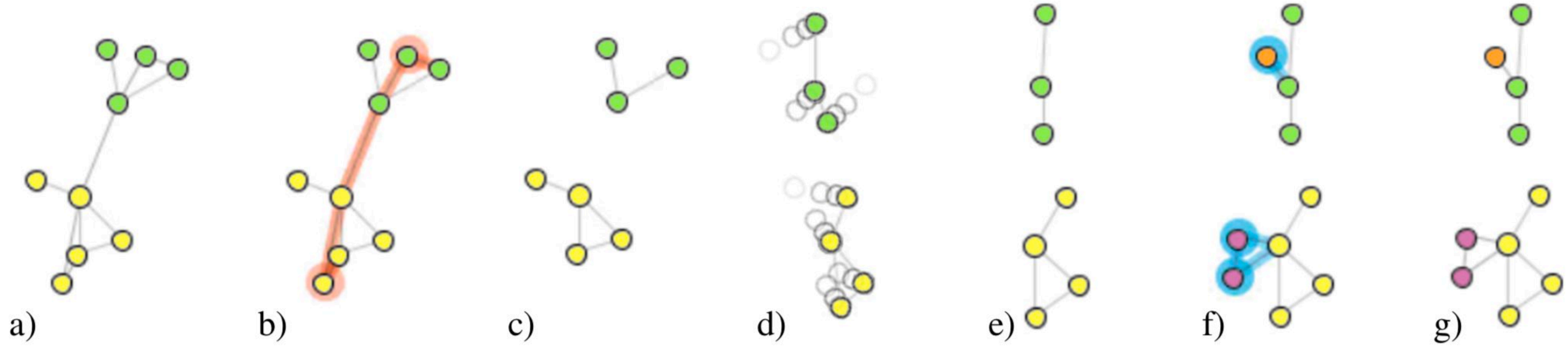


$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$

$$E = \{e_1, e_2, e_3, e_4\} = \{\{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_3, v_5, v_6\}, \{v_4\}\}$$

[Source: https://commons.wikimedia.org/wiki/File:Hypergraph-wikipedia.svg]

Dynamic network

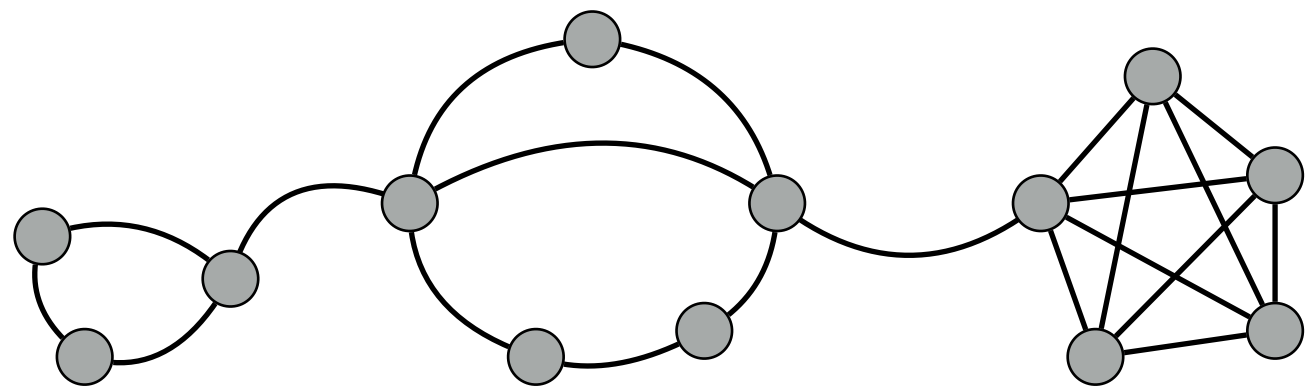


- a.k.a. time-varying networks, temporal networks
- informally: individual vertices and edges only exist at certain points in time
- formally:
  - as contact sequences: $T_{e_{i,j}} = \{t_1, \ldots, t_n\}$
  - as interval graphs: $T_{e_{i,j}} = \{(t_1, t_1'), \ldots, (t_n, t_n')\}$
  - as snapshots: sequence of static networks, one for each time step

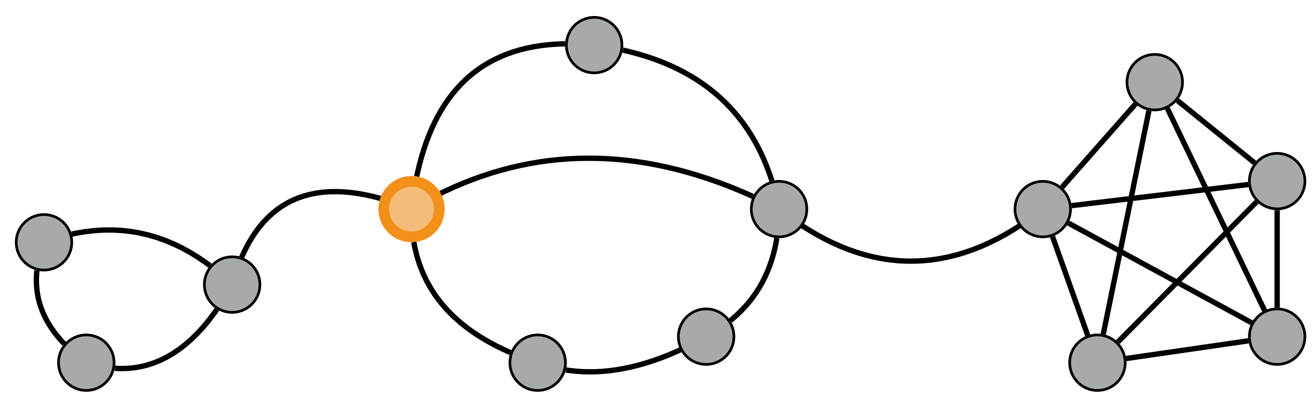
Concepts, metrics and topological structures relevant to analysts' tasks

- Order = number of vertices V
- Size = number of edges E
- Density = size / order, typically $\sqrt{\frac{e}{v^2}}$
- Connected components:

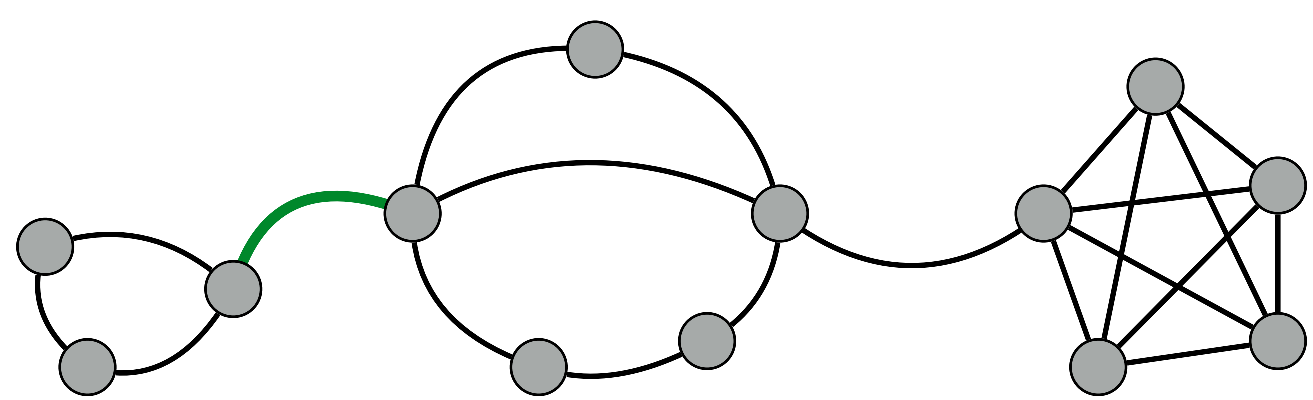
Connected *vs.* disconnected graph



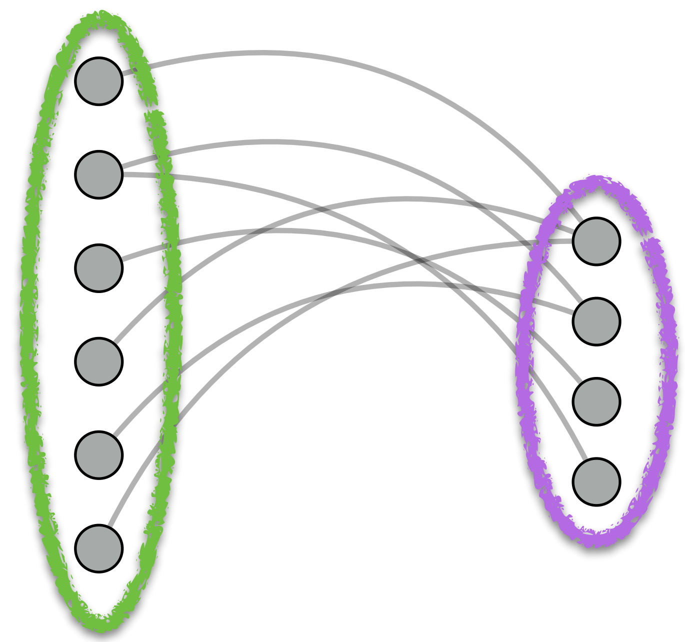
Articulation point



Bridge



Concepts, metrics and topological structures relevant to analysts' tasks



- K-Partite graph.
  Example: bipartite graph

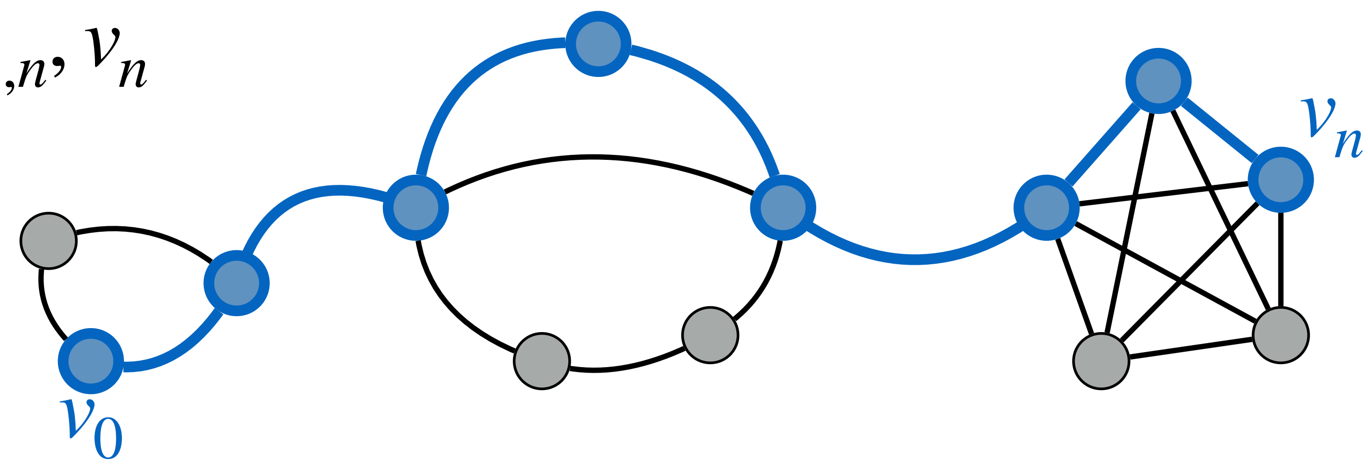
$$G = < U, V, E >$$

- Degree (valency) of a vertex = number of incident edges
  (in-degree, out-degree)

- Neighborhood of a vertex = all adjacent vertices

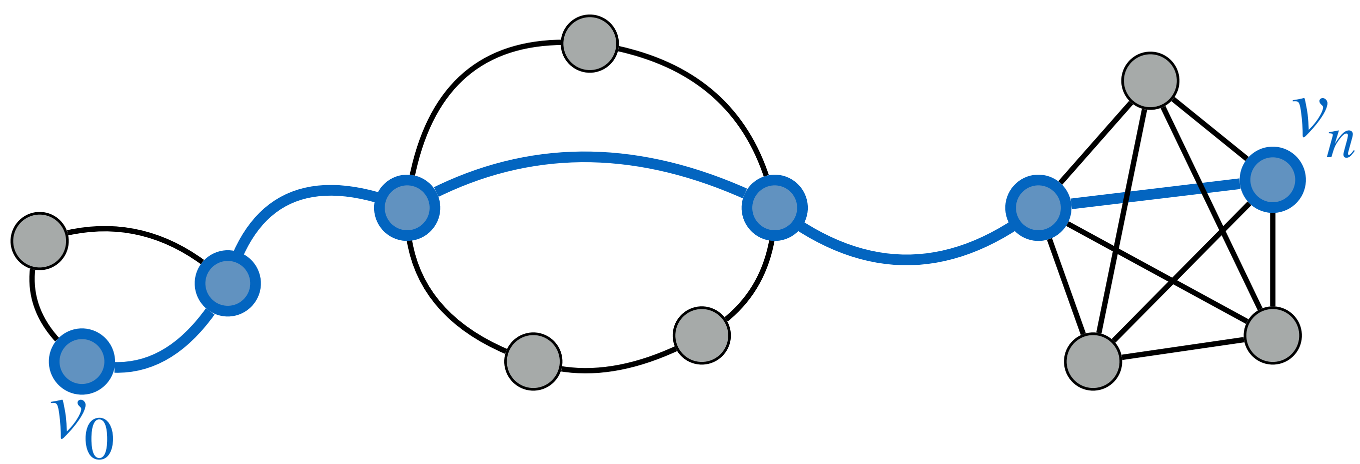
- Apex = vertex adjacent to all other vertices

- Path $= v_0, e_{0,1}, v_1, e_{1,2}, \ldots, v_{n-1}, e_{n-1,n}, v_n$



- Shortest path:

minimize $$\sum_{i=1}^{n-1} f(e_{i,i+1})$$



weights vs. no weights

$f : e \rightarrow \mathbb{R}$ $\quad f : e \rightarrow \{1\}$

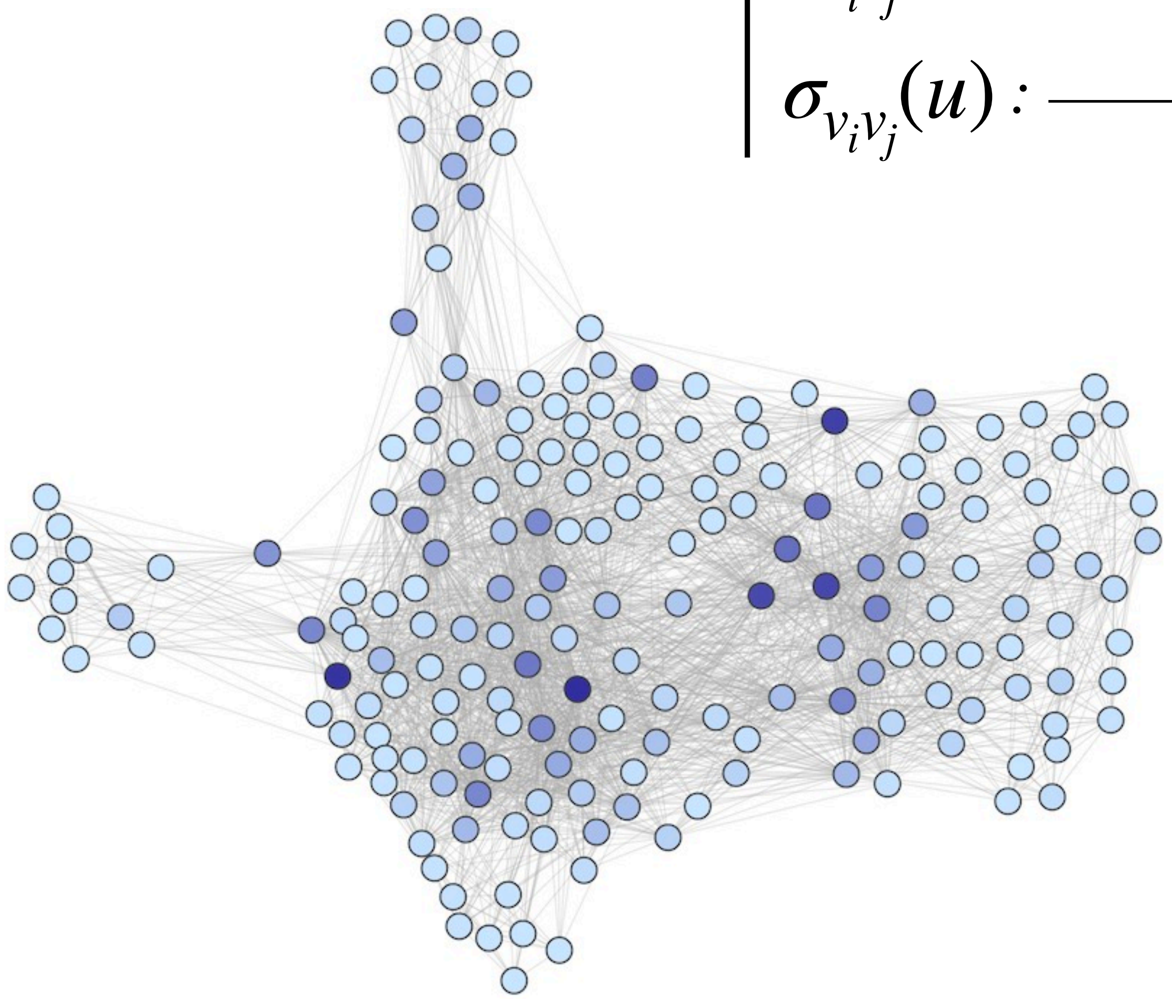
- (weighted) Diameter = longest (shortest path) in G
- Distance between v_i and v_j = number of edges between the two nodes
- Betweenness centrality

$$g(u) = \sum_{v_i \neq u \neq v_j} \frac{\sigma_{v_i v_j}(u)}{\sigma_{v_i v_j}}$$

$u, v_i, v_j \in V$

$\sigma_{v_i v_j}$ *: total number of shortest paths from* v_i *to* v_j

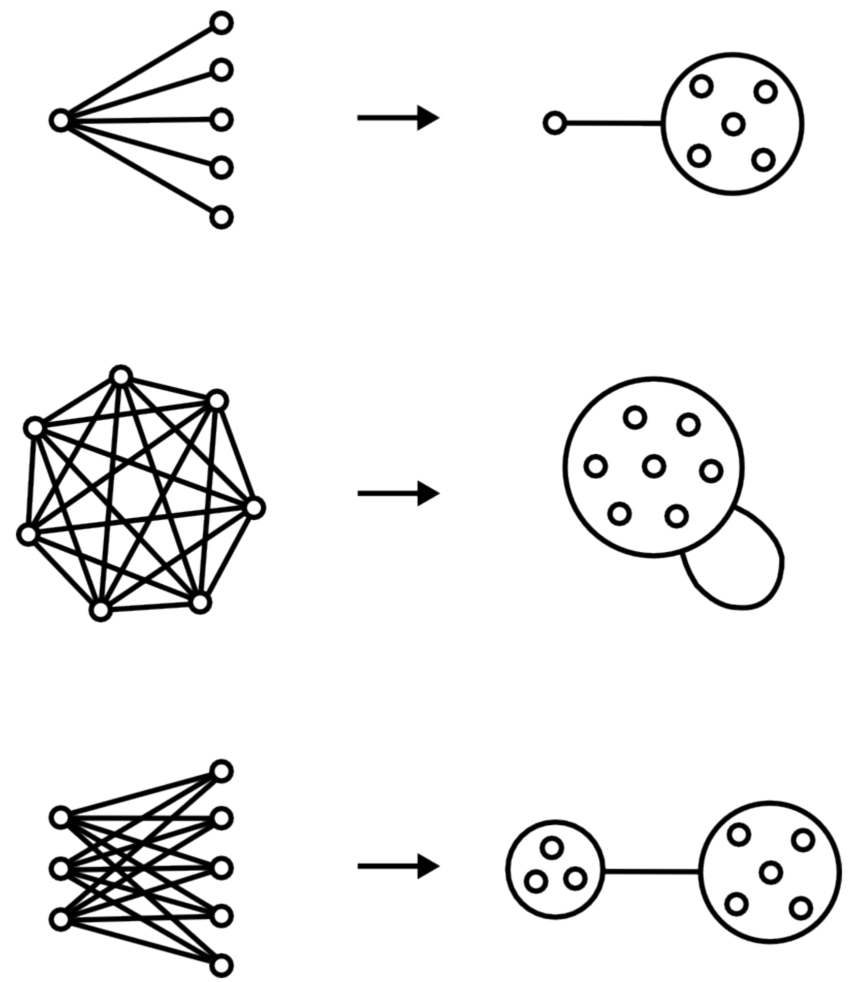
$\sigma_{v_i v_j}(u)$: ———————— *going through* u



Other measures of centrality:
- degree centrality
- closeness centrality
- eigenvector centrality
- …

Motif = recurrent structural patterns

- Star
- Clique = complete subgraph
- Bi-clique
- Cluster / community

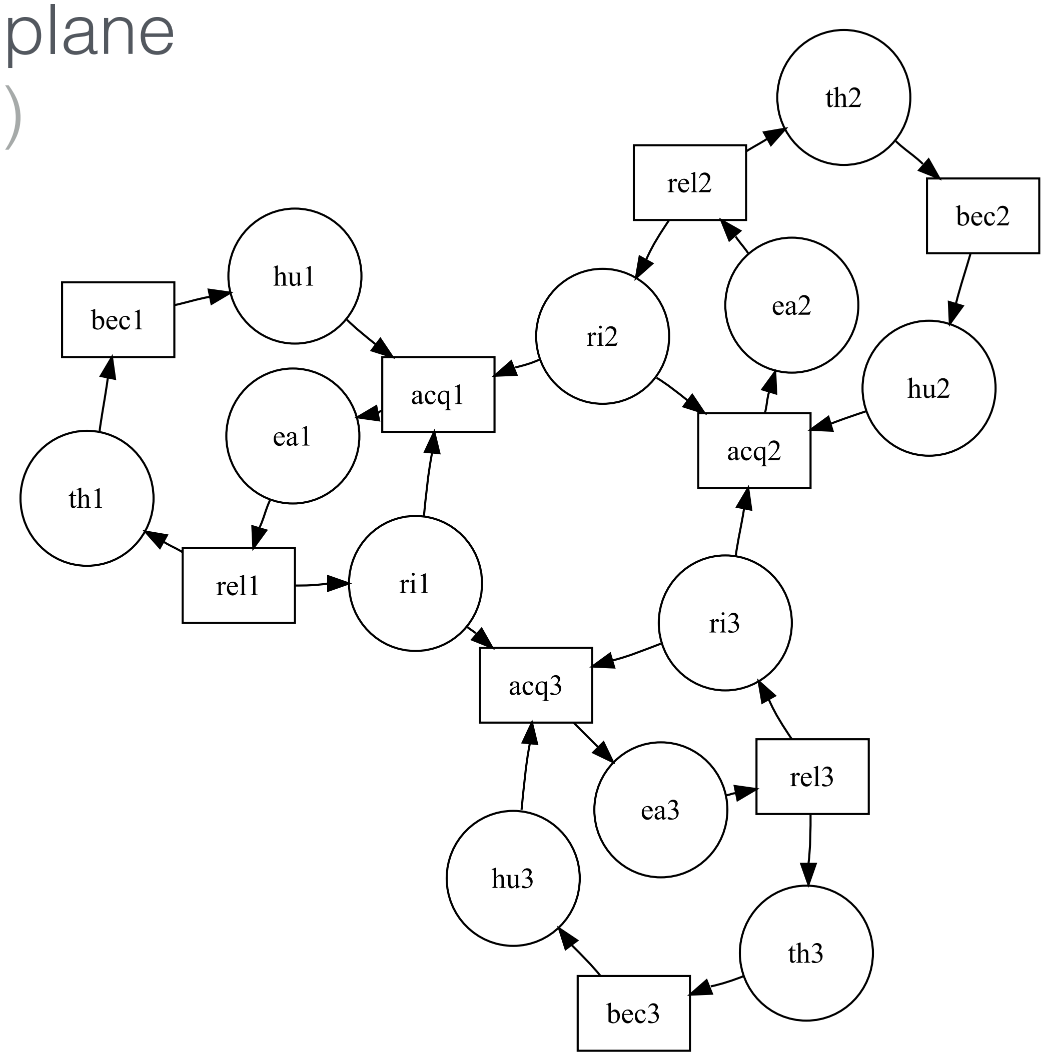


powergraph notation

[Image source: https://en.wikipedia.org/wiki/File:PowerGraphs.png]

Graph Visualization

- A network is made of things that are related
- Visualization helps understand how these things are related:
  - makes use of the freedom afforded by the 2D plane (compared to unidimensional textual syntaxes)
  - to graphically represent all entities involved.



Graph Visualization - Tasks

Low-level tasks

tend to involve few elements;
tend to involve little human judgment.

Higher-level tasks

tend to involve more elements;
tend to involve more human judgment.

Examples

Low-level tasks:

How many neighbors does v_i have?

Are v_i and v_j connected?

How are they connected?
(type, strength, distance)

How dense is the graph?

Higher-level tasks:

Which nodes are important?

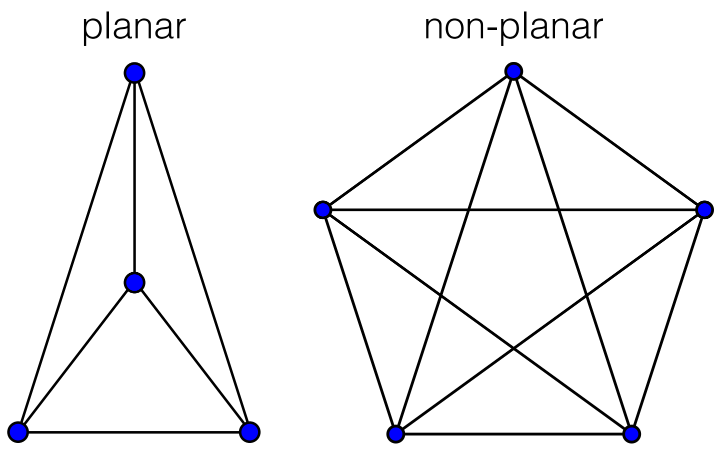
Are there clusters? Where are they?
Other interesting patterns?

How are they connected?
(correlate attributes and topology)

How does the network evolve over time?

Definitions: graph embedding, graph drawing

- Embedding of a graph $G = < V, E >$ on a surface Σ:
  - representation of G on Σ in which $\left| \begin{array}{r} \text{points on } \Sigma \leftrightarrow V \\ \text{arcs on } \Sigma \leftrightarrow E \end{array} \right.$
  - arc endpoints of $e \in E$ are the points on Σ corresponding to the associated vertices $v_i, v_j \in V$
  - two arcs never intersect
  - planar graphs can be embedded in $\mathbb{R}^2$

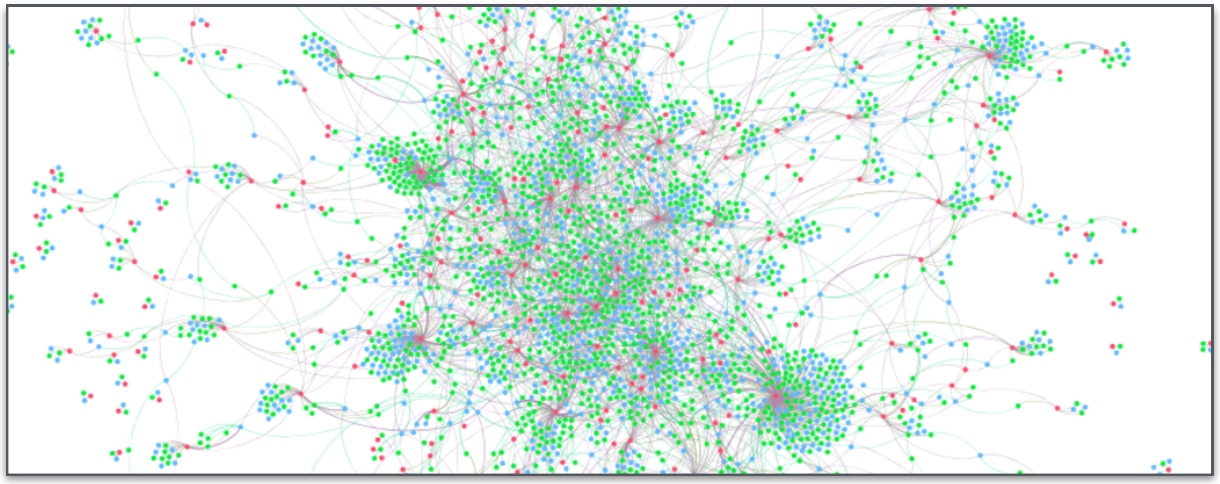


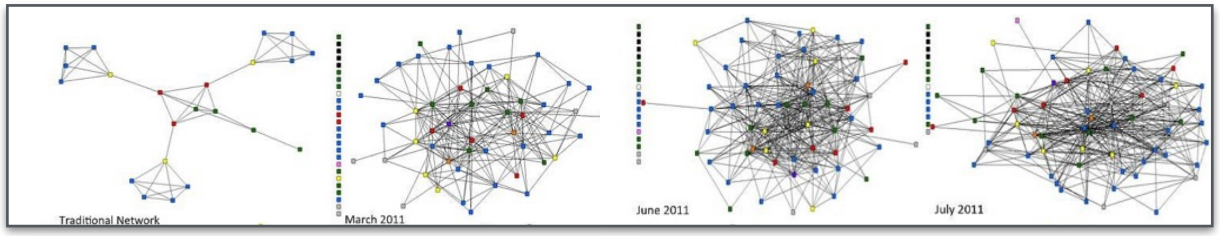
- Drawing a graph: derive a 2-dimensional depiction, often omitting the non-intersection condition on edges

[Source: https://commons.wikimedia.org/wiki/File:Multi-pseudograph.svg]

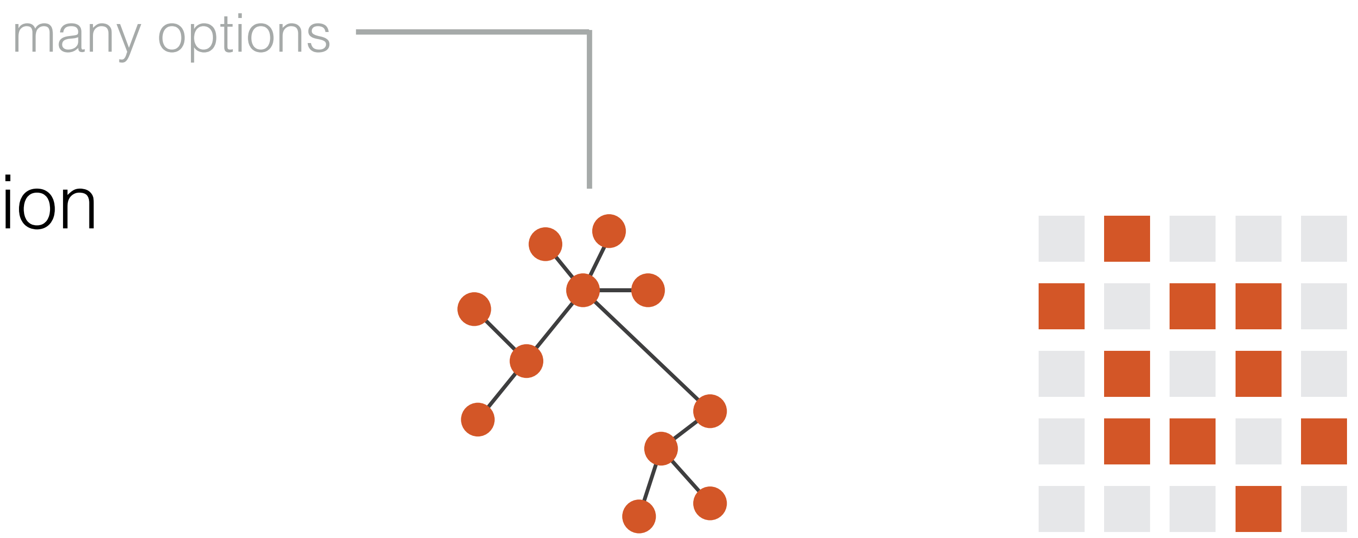
Graph Visualization - Challenges

- Graph order and size
- Graph density
- Visual mappings in multi-variate graphs
- Fixed location for (some) nodes
- Dynamic networks (change over time)





Graph Visualization - Challenges

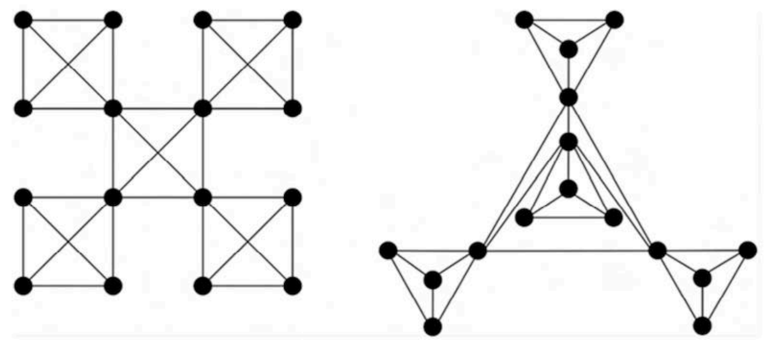


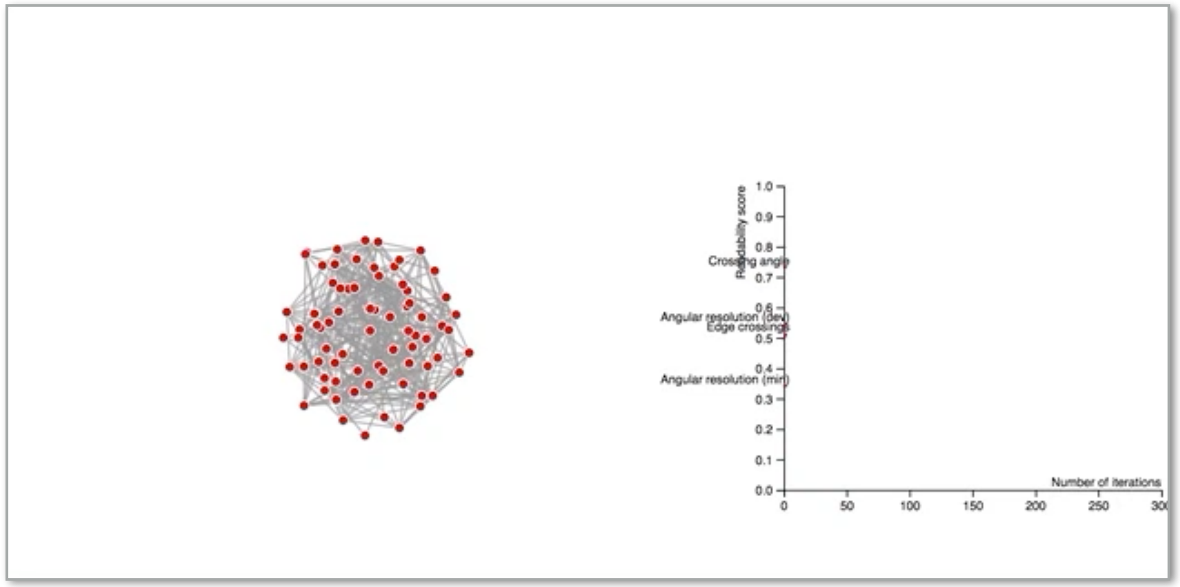
- Choosing the right representation
- Choosing the right layout algorithm
  - free *vs.* styled *vs.* fixed layout
  - topology- *vs.* attribute-driven layout
- Supporting effective navigation in the structure

[Image source: Visualization Analysis and Design. Tamara Munzner, with illustrations by Eamonn Maguire. A K Peters Visualization Series, CRC Press, 2014]

Graph Visualization / Node-link Diagrams

- Optimize legibility
  - Prevent node overlap
  - Minimize edge crossings
- Taking into account aesthetics criteria
  - Aspect ratio
  - Foster symmetry
  - Uniform edge length
  - Equal angles





https://github.com/rpgove/greadability

- Graph Drawing is a field of research of its own

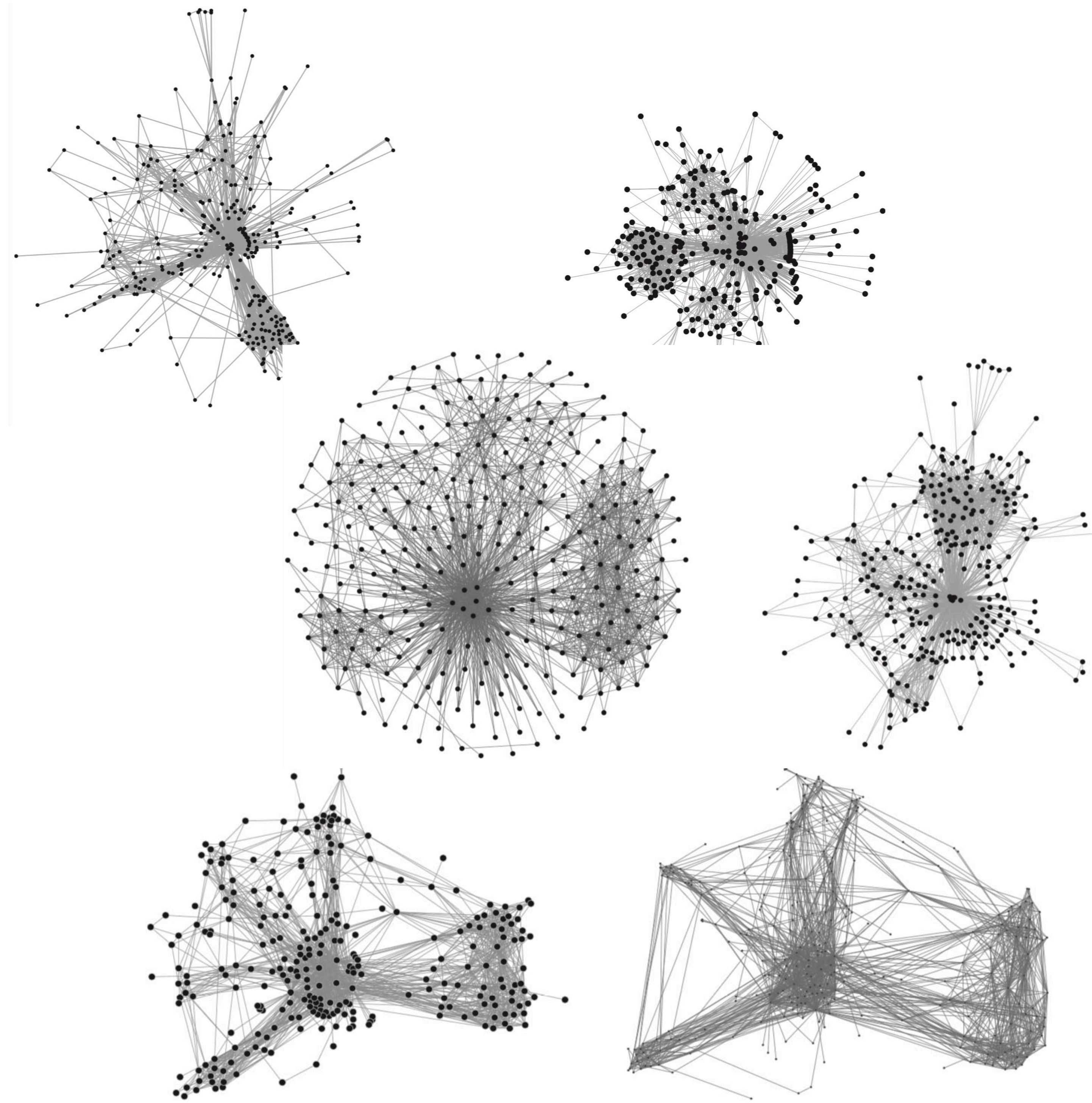
Node-Link Diagrams / Force-directed Layout

- Metaphor of physical forces: attraction (springs, gravity) + repulsion (charged particles)

- Physical simulation
  - iterative process
  - run until an equilibrium is reached

- Algorithms
  - Kamada & Kawai
  - Fruchterman & Reingold
  - GEM
  - ForceAtlas
  - LinLog

- Complexity
  - basic approaches are $O(V^2)$
  - Barnes & Hut's hierarchical force-calculation algorithm is $O(E + V \times log(V))$



[Figures from Gibson et al., A survey of two-dimensional graph layout techniques for information visualisation. Information visualization 12.3-4 (2013): 324-357.]

Node-Link Diagrams / Force-directed Layout

Table 1. A summary of how the graph drawing algorithms in this article compare. A checkmark against a graph drawing principle indicates that it was considered in the design of the algorithm.

Algorithm	Performance	Graph drawing principles				Size	Notes
		Edge crossings	Symmetry	Even node distribution	Uniform edge		
Spring Embedder	Good for graphs of up to 50 nodes		✓Achieves this for small graphs		✓Small graphs have even edge lengths	50 nodes and should not be dense	Best for fewer than 30 nodes
Fruchterman and Reingold (FR)	Aim was for speed and simplicity but slower than GEM	✓Performs worse than Tunkelang's algorithm on edge crossings	✓Shows symmetry	✓	✓This was one of their aims but they did not think they achieved it; but it is better than SA and Tu for dense graphs	Able to draw graphs with thousands of nodes but slowly, and does not often result in a good layout	Does not require parameters to be optimised
GEM	Much faster than FR and KK	✓Although they say they did not explicitly minimise this	✓Shows symmetry well	✓Similar performance to FR and KK	✓Similar to FR and KK	Handles large (>128 nodes) better than FR and KK	Produces similar results to FR and KK when compared using the graph drawing principles
Kamada and Kawai (KK)	Much slower than GEM for graphs of >30 nodes		✓Main aim and is shown in layouts			Up to 30 nodes, better results than GEM on small sparse graphs for graph drawing principles	Euclidean distance should approximate graph theoretic distance
Simulated Annealing (SA)	Considered to be too slow to be used practically	✓Performs worse than FR and Tu	Can show symmetry	✓Performs worse than FR and Tu	✓Performs worse than Fr and Tu	Can be used for graphs of up to 60 nodes with similar performance to FR, KK and GEM	Also tries to prevent nodes coming too close to edges. Adaptable
Tunkelang (Tu)	Similar to FR; 100 nodes in less than 3 s	✓Performs better than FR and SA on this metric	Does not show symmetry at all	For dense graphs FR performs better	✓For dense graphs FR performs better	Tested on graphs up to 60 nodes, sparse and dense. Results are better on the sparser graphs	Aims to place non-adjacent nodes further from each other. If other force-directed layouts do not produce a good result then try this algorithm.
LinLog				Goes against this criterion to show clustering	Violates uniform edge lengths in order to emphasise clustering	Able to draw graphs with many thousands of nodes	Results in graphs with more obvious clusters than KK and FR
Force Atlas	Speed can be sacrificed for greater precision or vice versa. In terms of number of iterations required to lay out the graph it takes fewer than FR			Also tries to optimise clustering but to a lesser extent than LinLog	Edges should be as short as possible	Tested on graphs with more than 20,000 nodes	

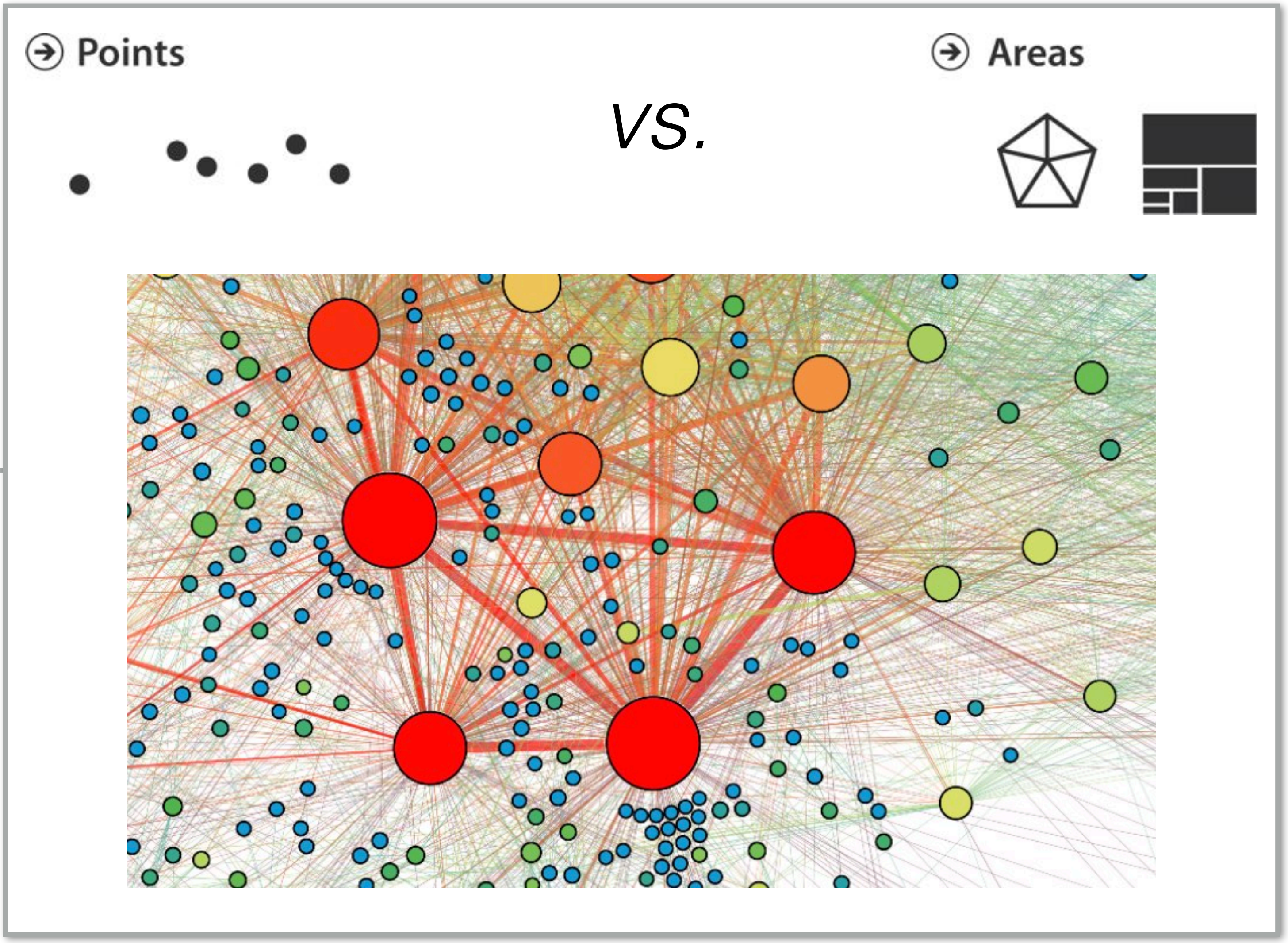
(continued)

Node-Link Diagrams / Force-directed Layout

- Metaphor of physical forces: attraction (springs, gravity) + repulsion (charged particles)

- Physical simulation
  - iterative process
  - run until an equilibrium is reached

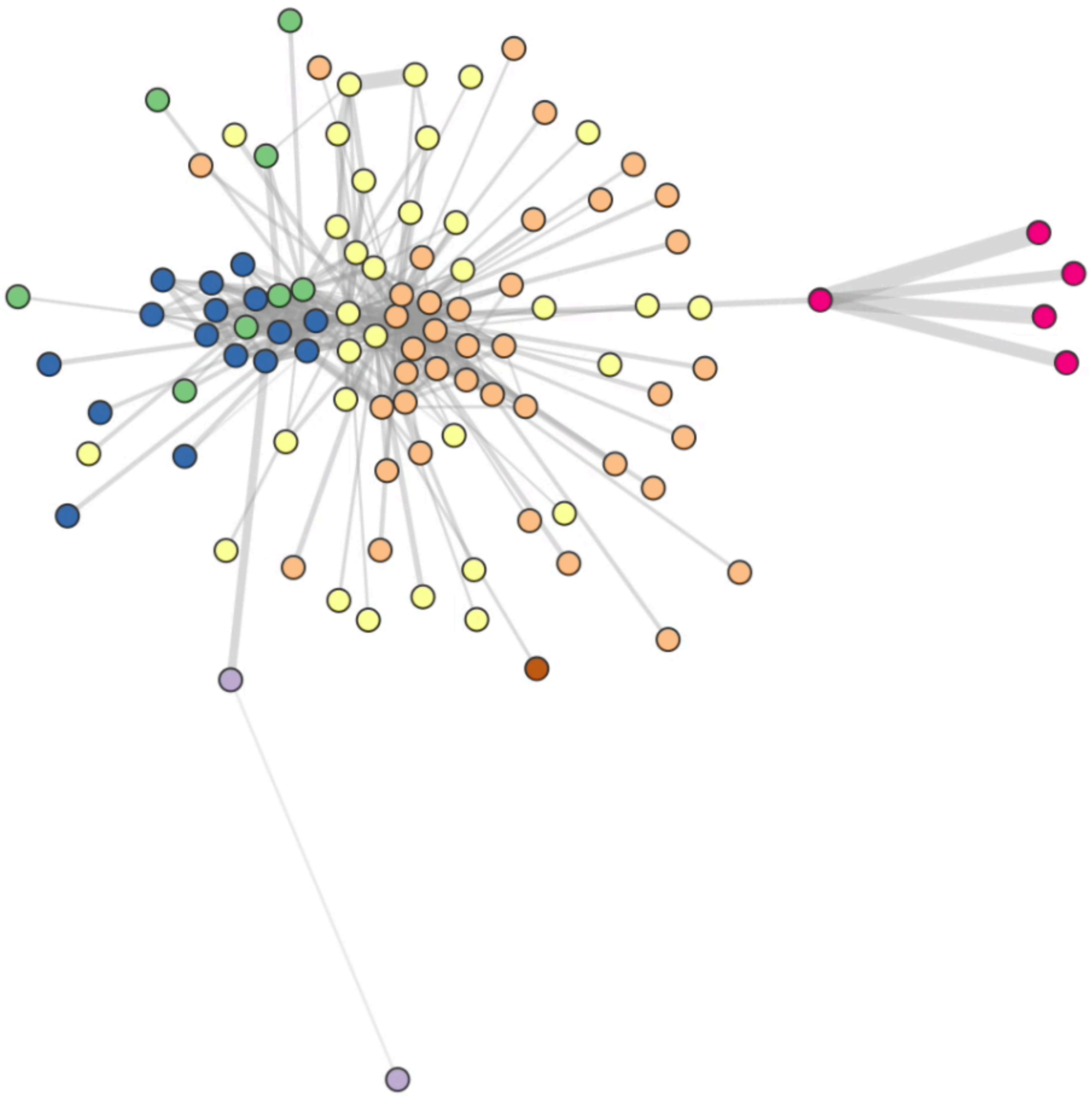
- Algorithms
  - Kamada & Kawai
  - Fruchterman & Reingold
  - GEM
  - ForceAtlas
  - LinLog



- Complexity
  - basic approaches are $O(V^2)$
  - Barnes & Hut's hierarchical force-calculation algorithm is $O(E + V \times log(V))$

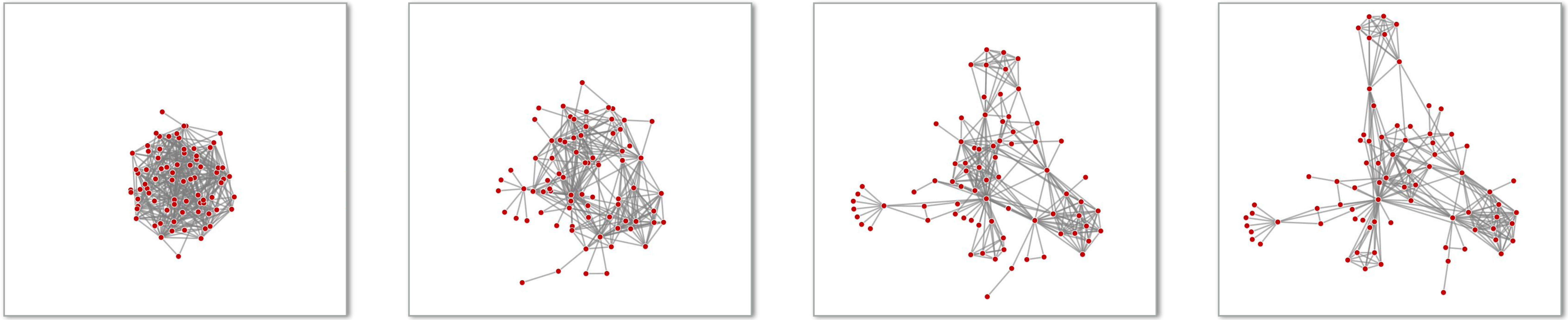
Node-Link Diagrams / Force-directed Layout

- Metaphor of physical forces: attraction (springs, gravity) + repulsion (charged particles)
- Physical simulation
  - iterative process
  - run until an equilibrium is reached
- Algorithms
  - Kamada & Kawai
  - Fruchterman & Reingold
  - GEM
  - Davidson & Harel
  - LinLog
- Complexity
  - basic approaches are $O(V^2)$
  - Barnes & Hut's hierarchical force-calculation algorithm is $O(E + V \times log(V))$



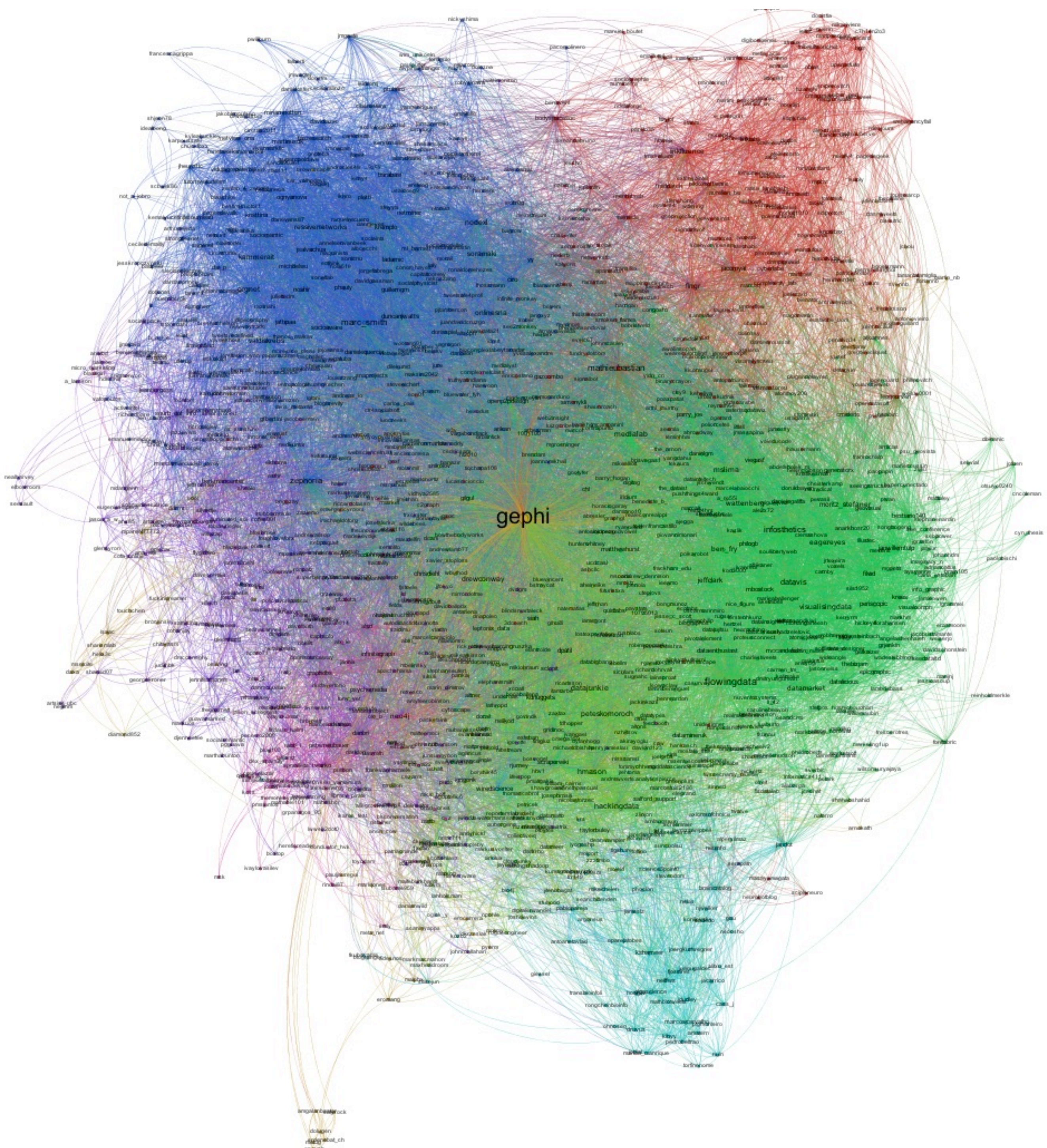
Force-directed Layout / Iterative & non-deterministic

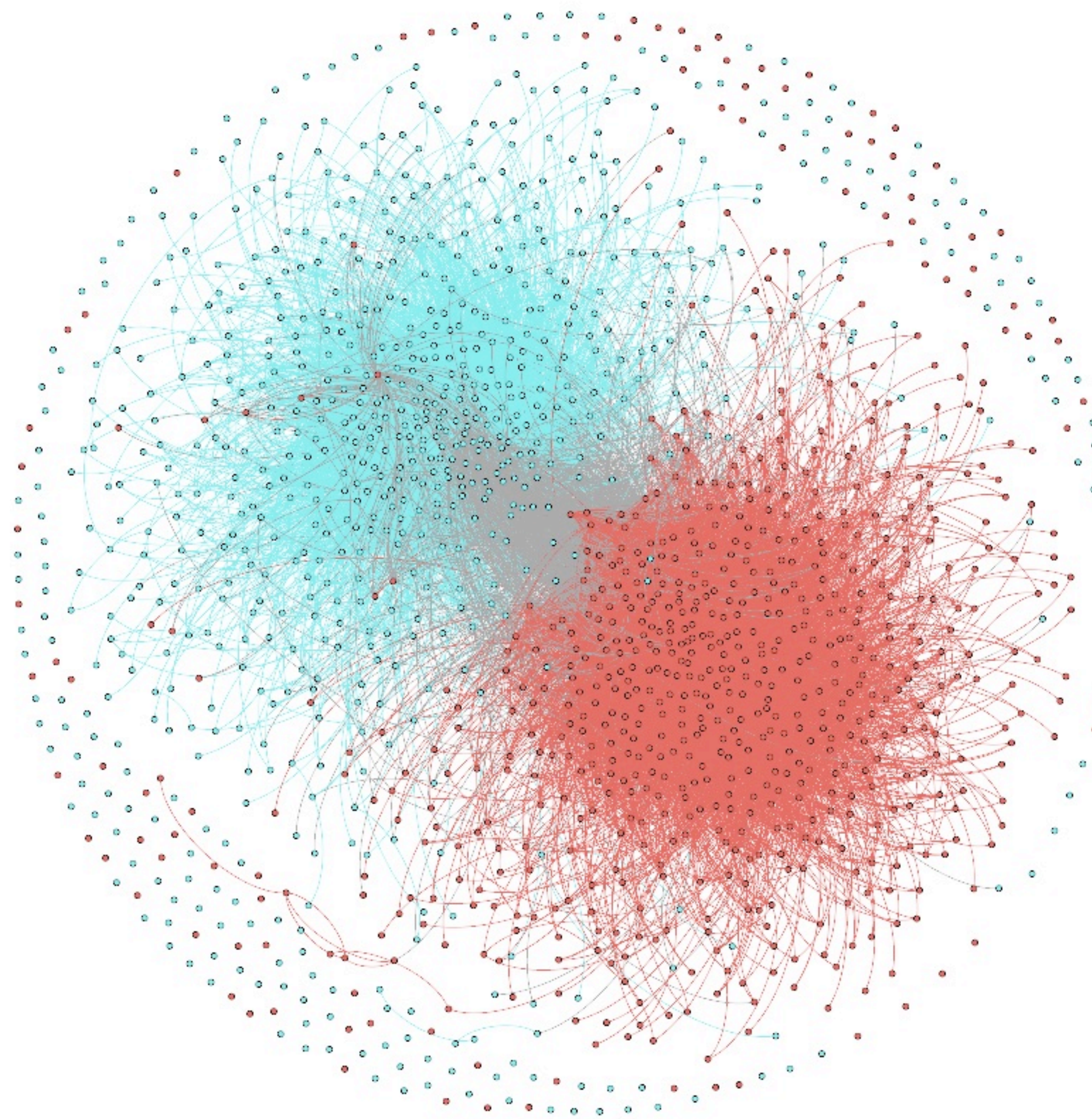
- Physical simulation
  - iterative process
  - run until an equilibrium is reached (can be a local minimum)
  - process can be interrupted



- Depends on initial conditions
- Nondeterministic
- Lack of spatial stability => fail to preserve users' spatial memory

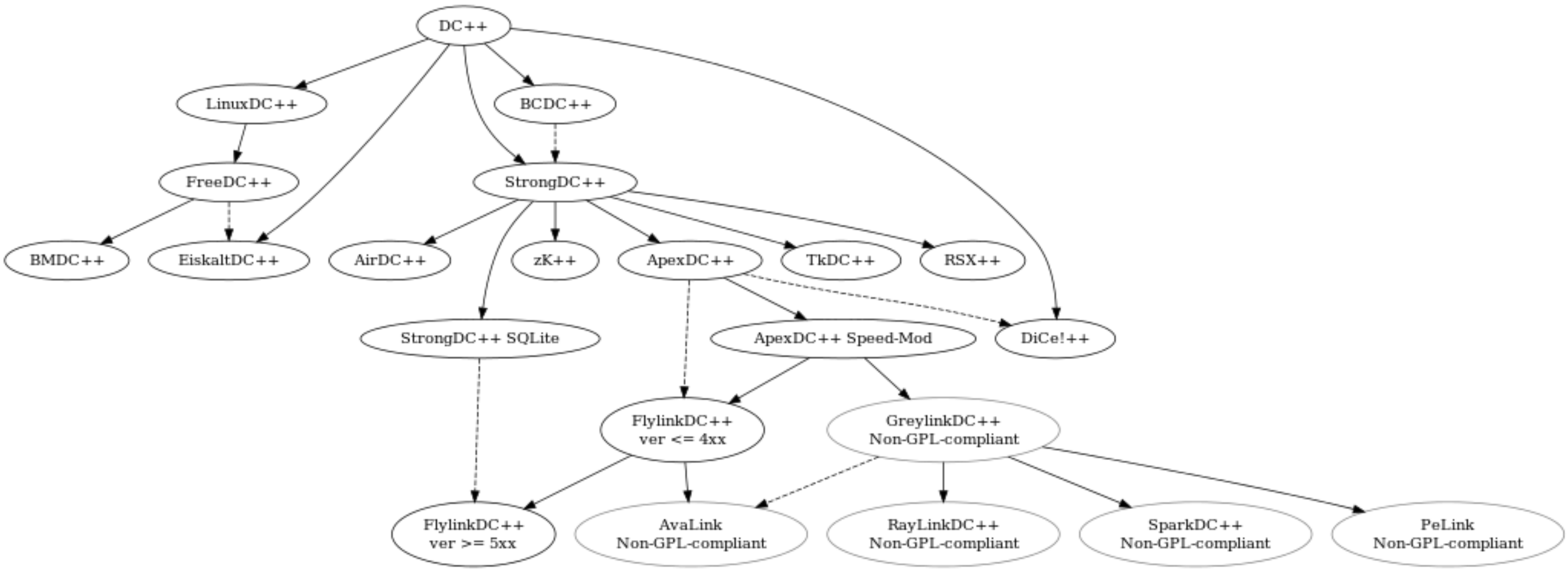
The Hairball





Layered Graph

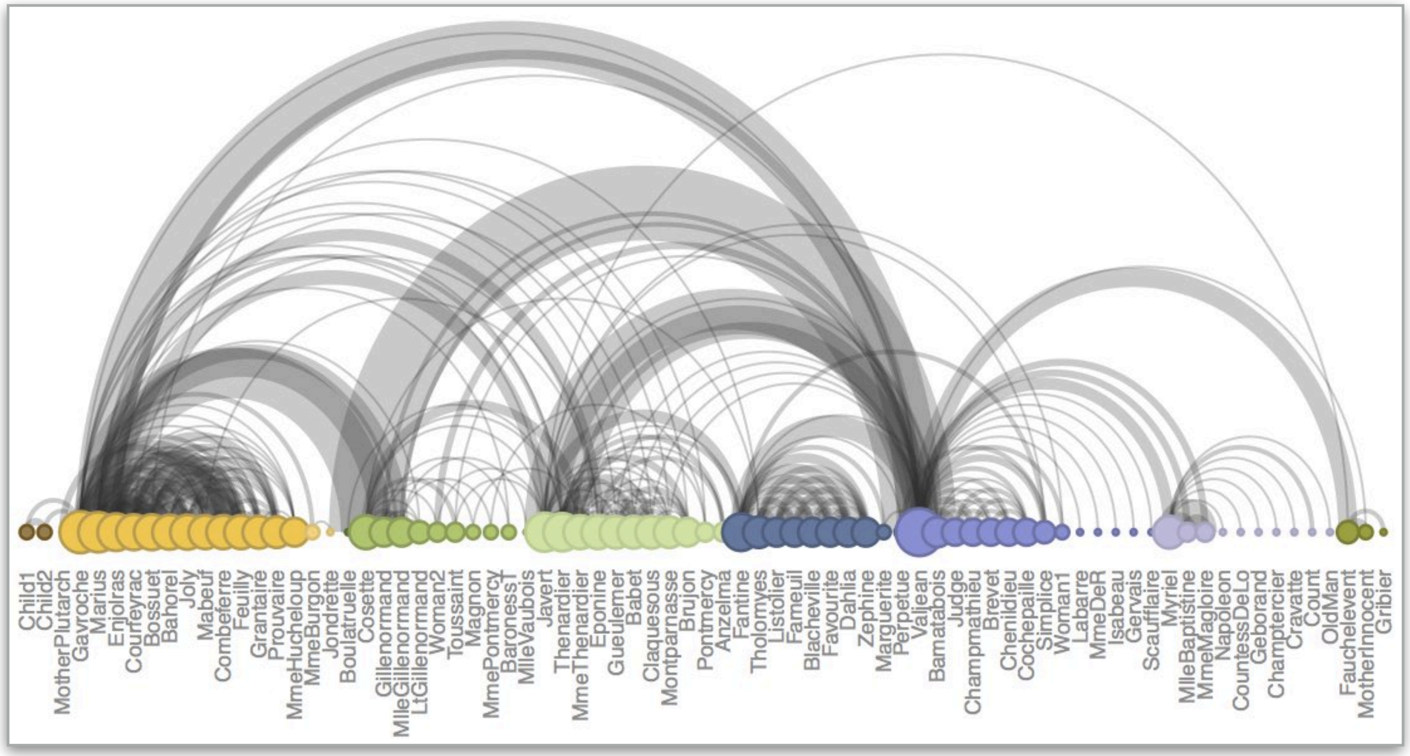
- Layout method for directed acyclic graphs
- Nodes put on different layers depending on their depth in the hierarchy



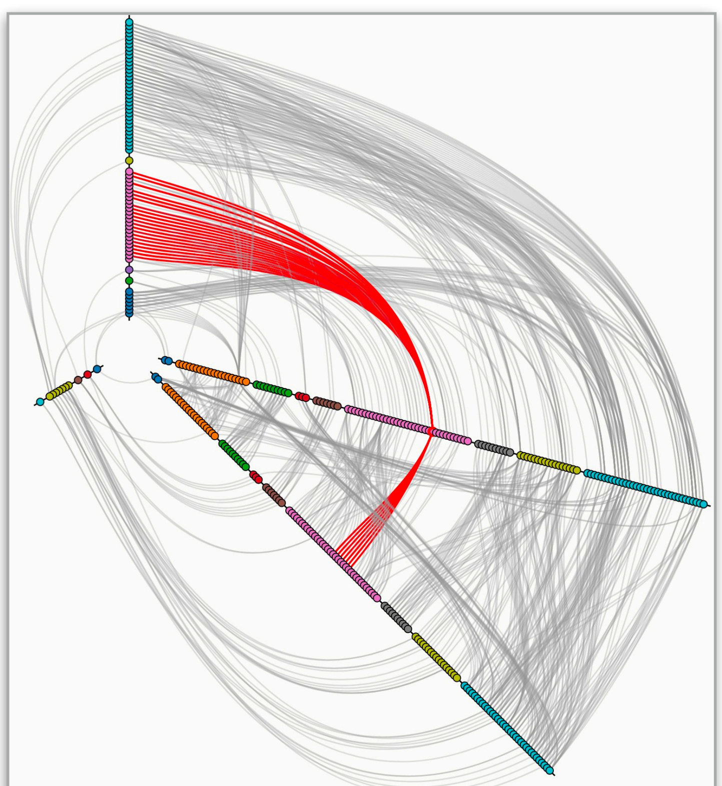
[Sugiyama et al., Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics 11.2 (1981): 109-125.]

Other layouts: radial, linear, …

- Node grouping and ordering

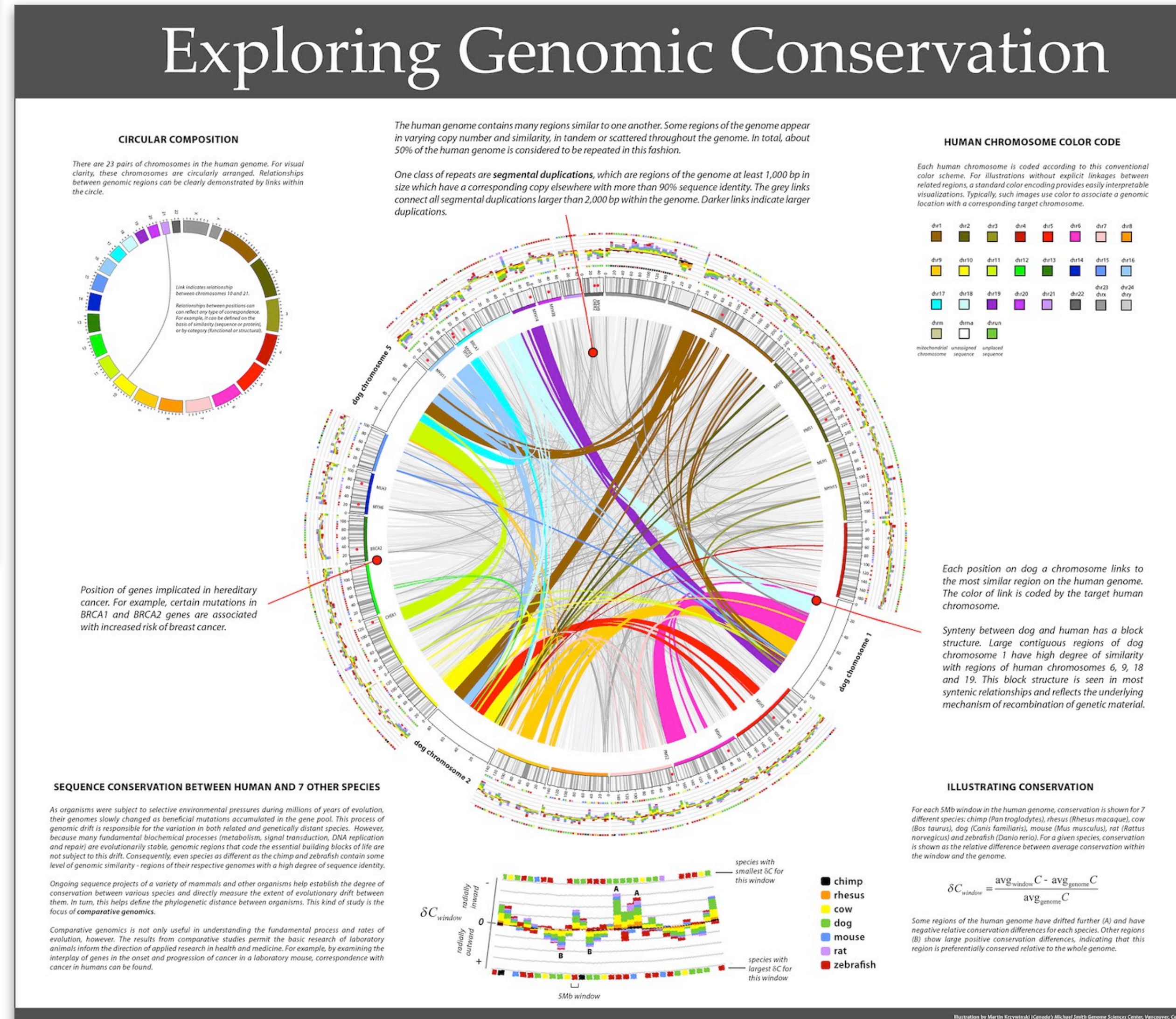


[Wattenberg, M. (2002), "Arc diagrams: visualizing structure in strings", Proc. IEEE Symposium on Information Visualization (INFOVIS 2002), pp. 110–116, doi:10.1109/INFVIS.2002.1173155,]

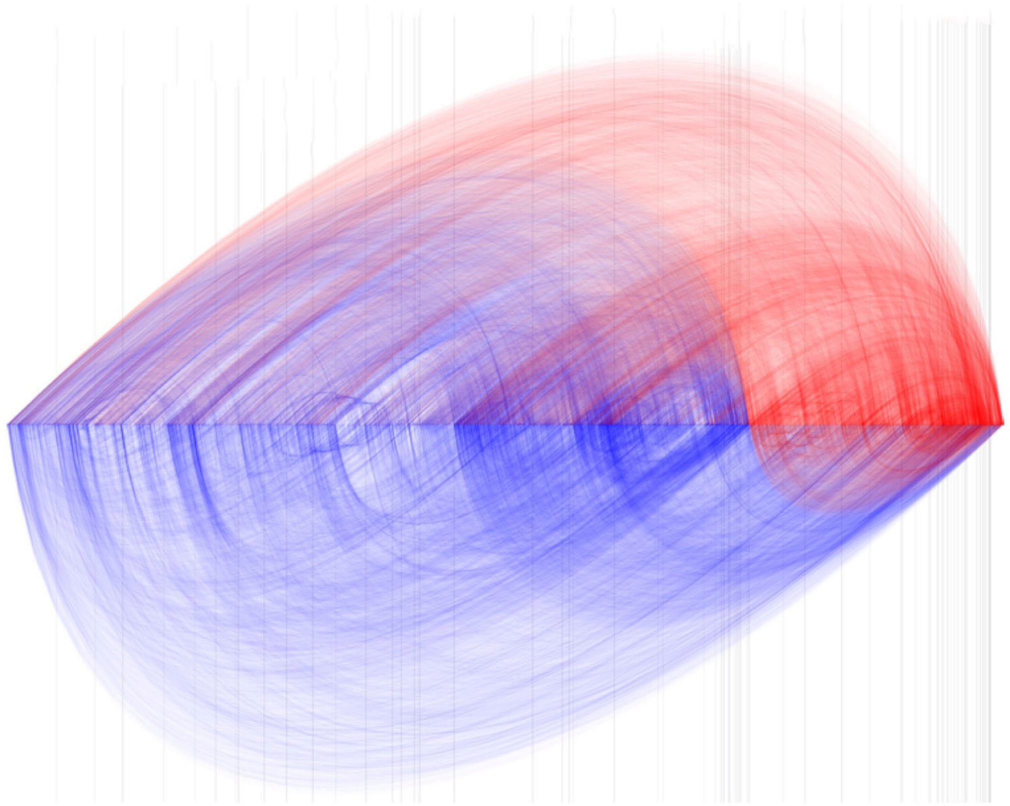


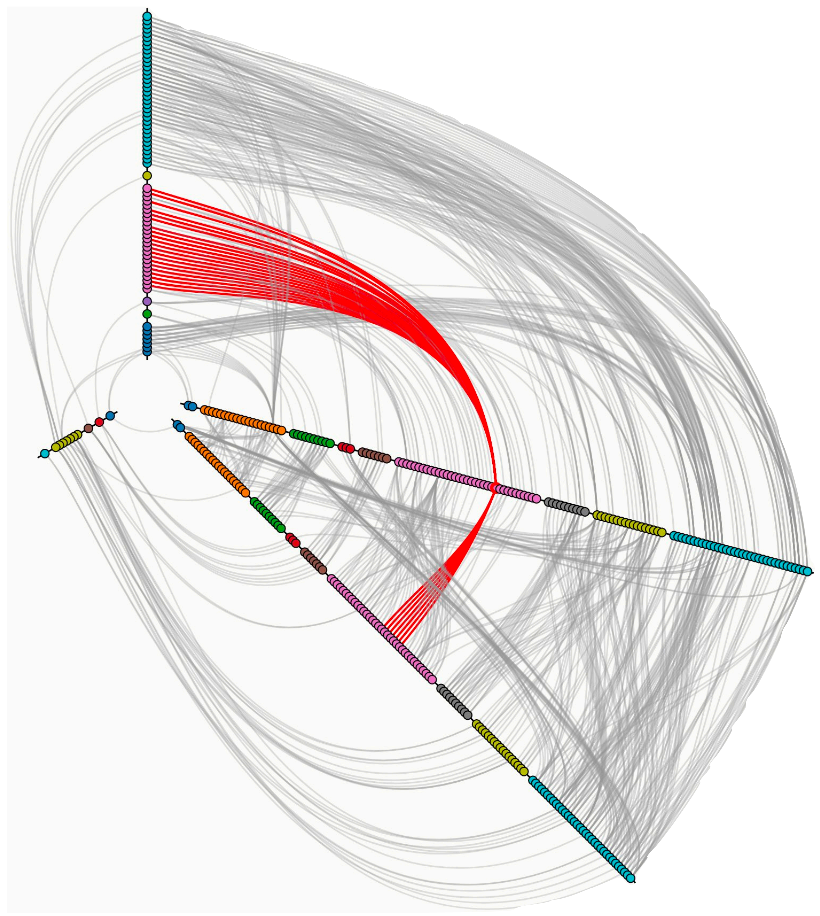
[Krzywinski M, Birol I, Jones S, Marra M (2011). Hive Plots — Rational Approach to Visualizing Networks. Briefings in Bioinformatics, doi: 10.1093/bib/bbr069)]

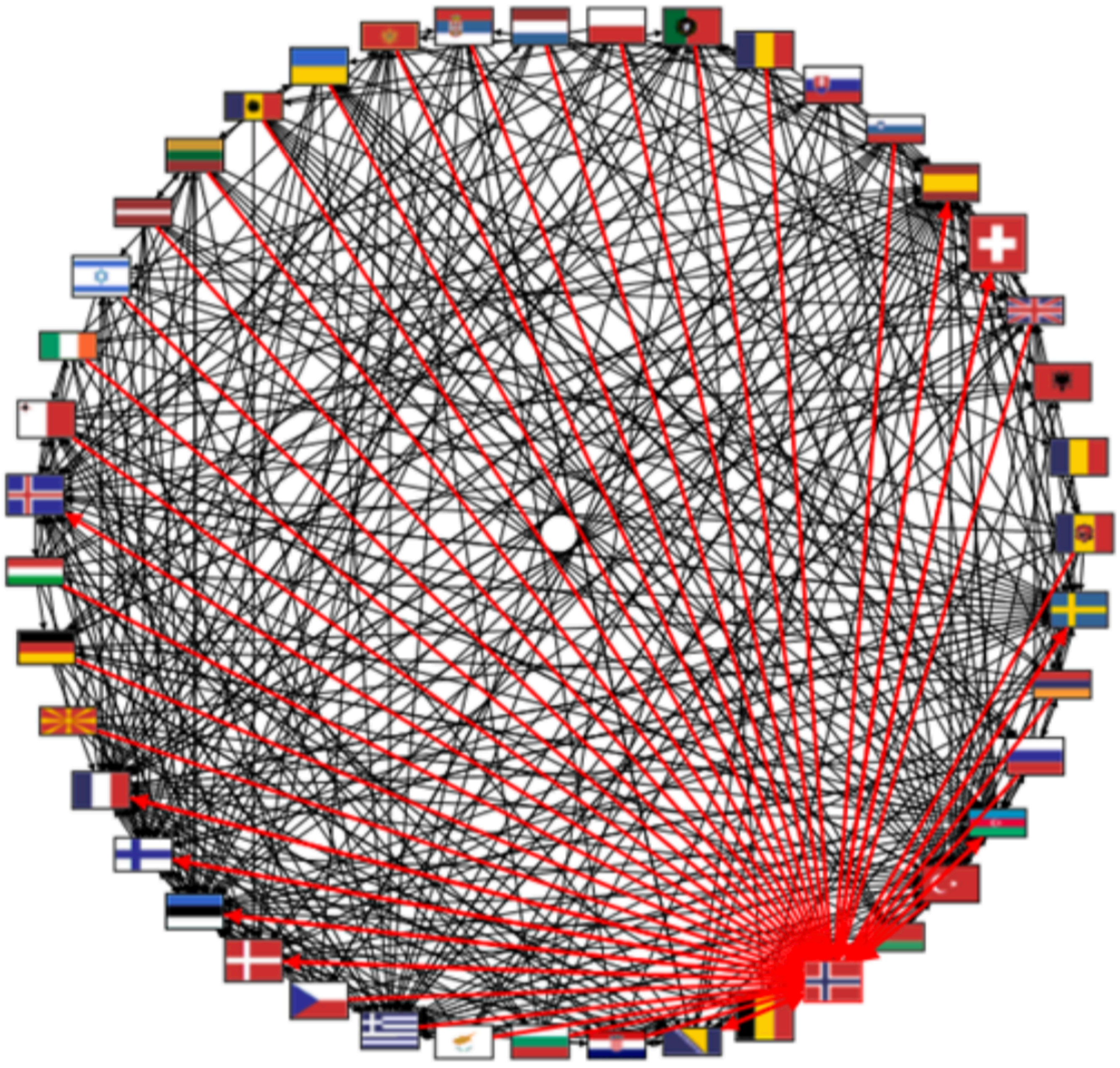
- Visual encoding of node attributes on the periphery
- Node grouping and ordering



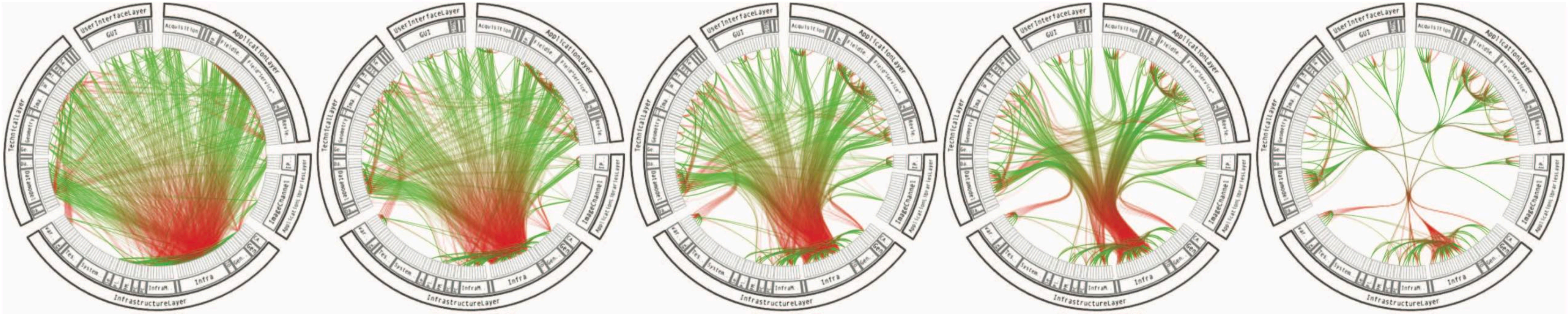
Other layouts: radial, linear, … can be clutter-prone

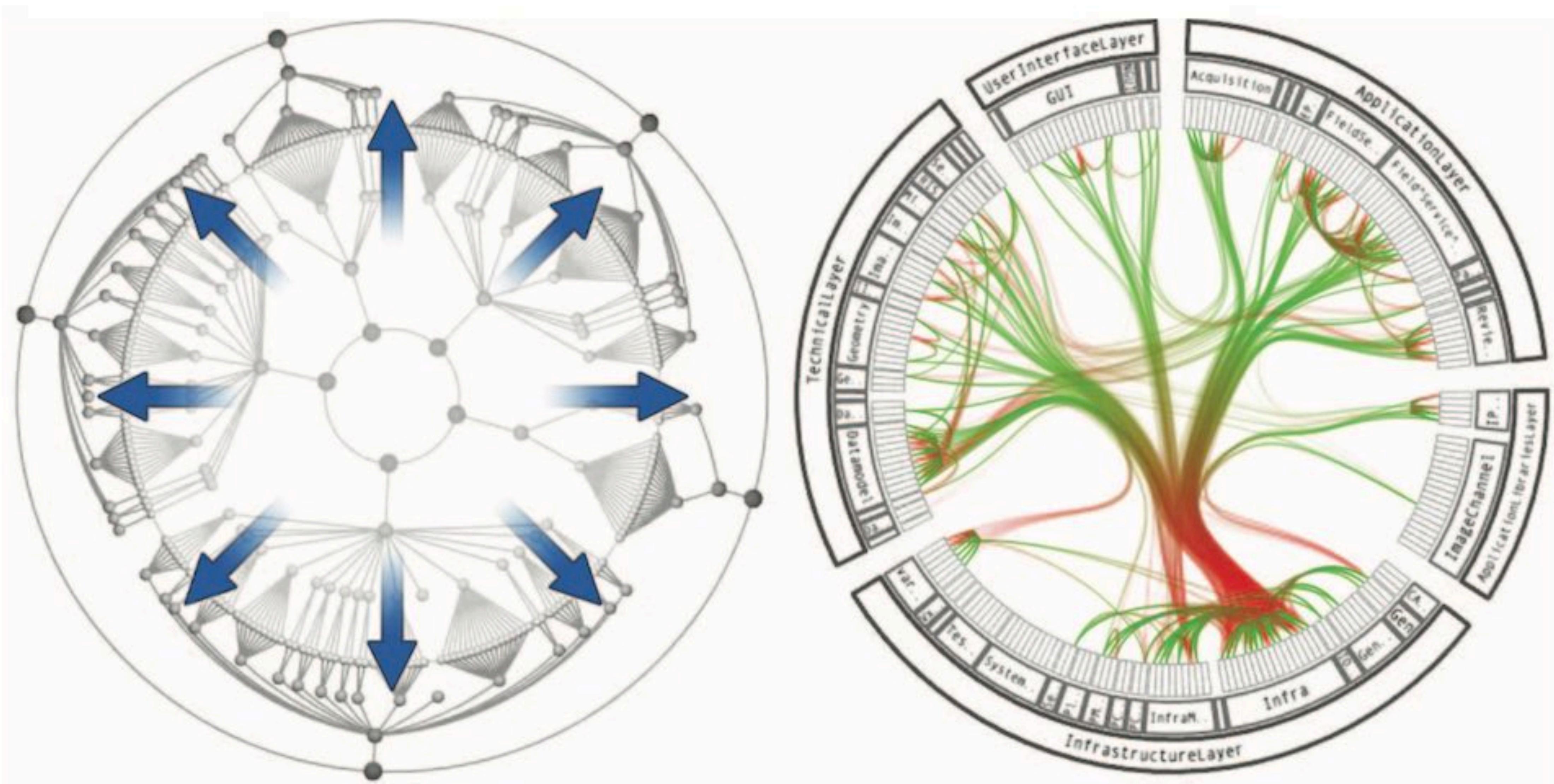






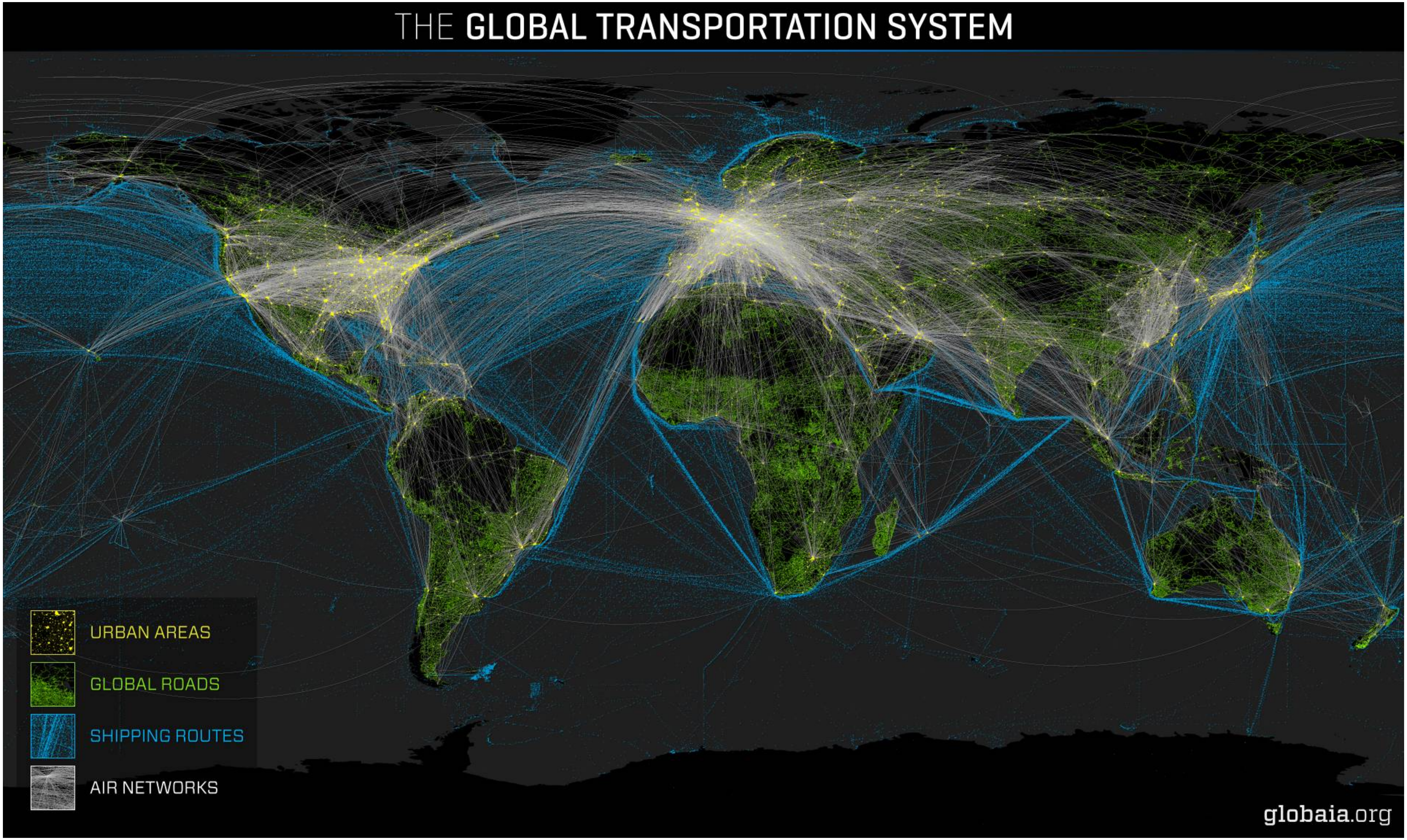
Solution: Hierarchical Edge Bundling



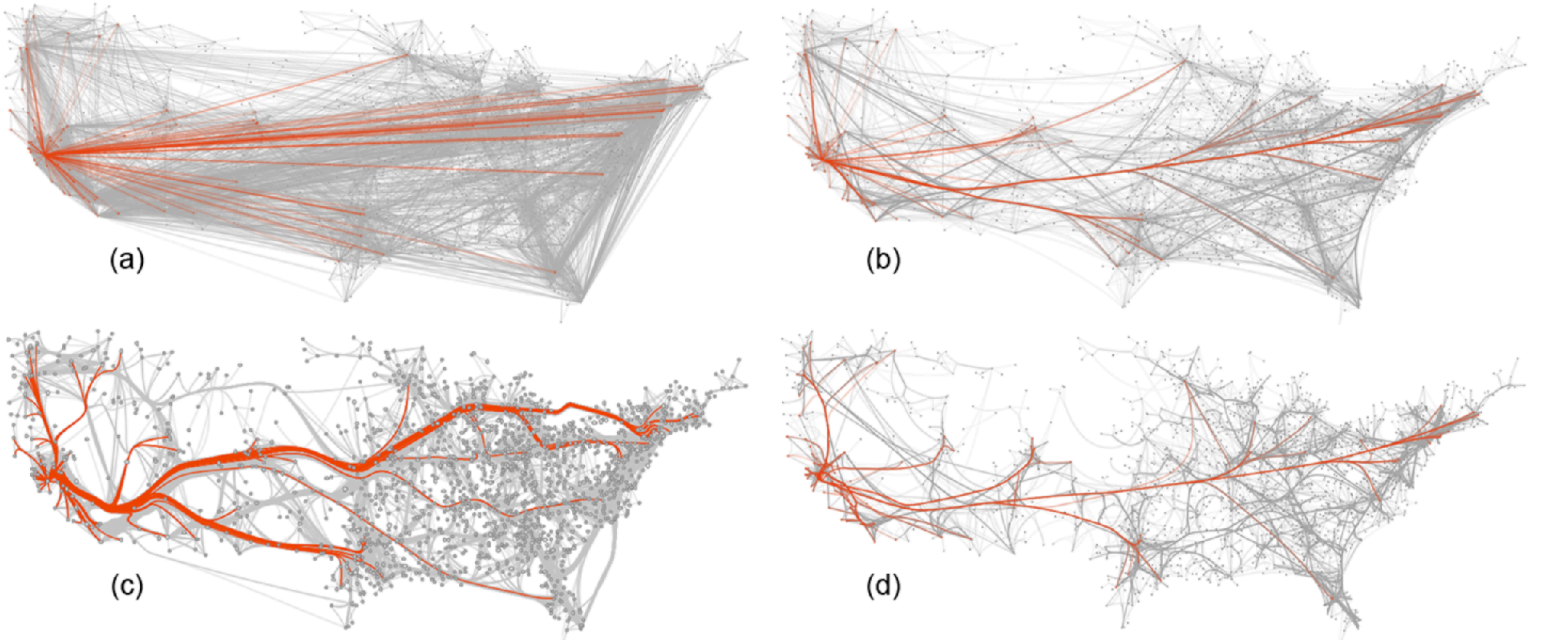


[Danny Holten. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. IEEE Transactions on Visualization and Computer Graphics 12, 5 (September 2006), 741-748. DOI: https://doi.org/10.1109/TVCG.2006.147]

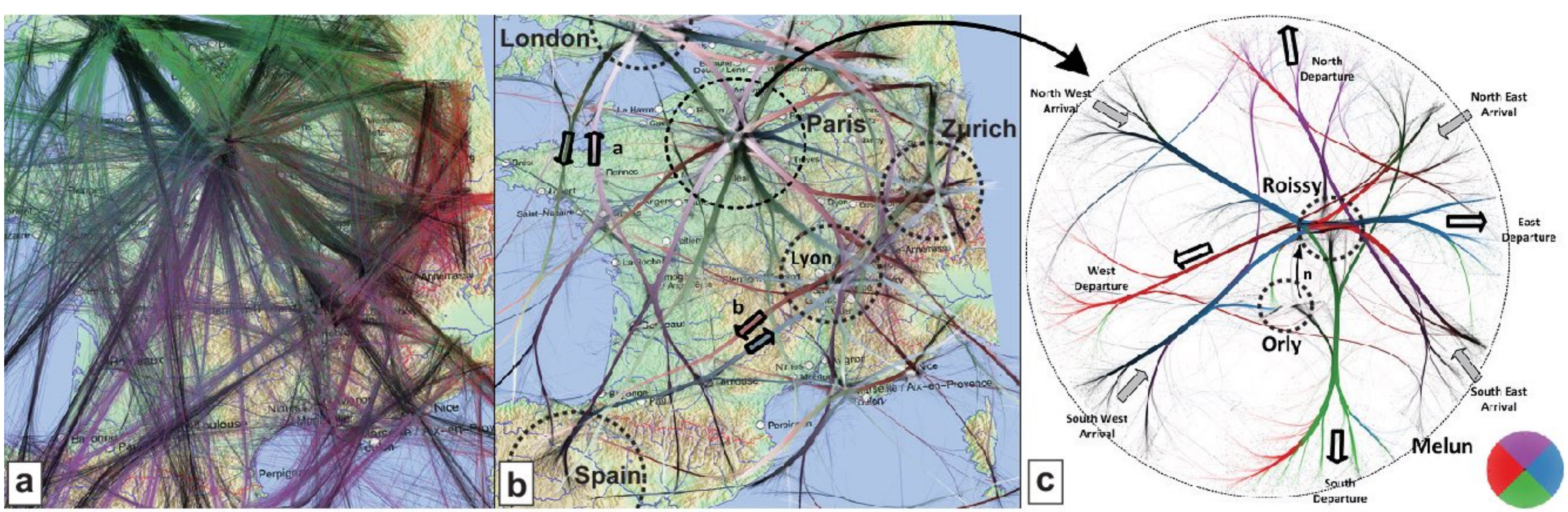
Geospatial network



Edge bundling in geospatial network



[D. Holten and J. J. van Wijk. 2009. Force-directed edge bundling for graph visualization. In Proceedings of the 11th Eurographics / IEEE - VGTC conference on Visualization (EuroVis'09), The Eurographs Association & John Wiley & Sons, Ltd., 983-998]



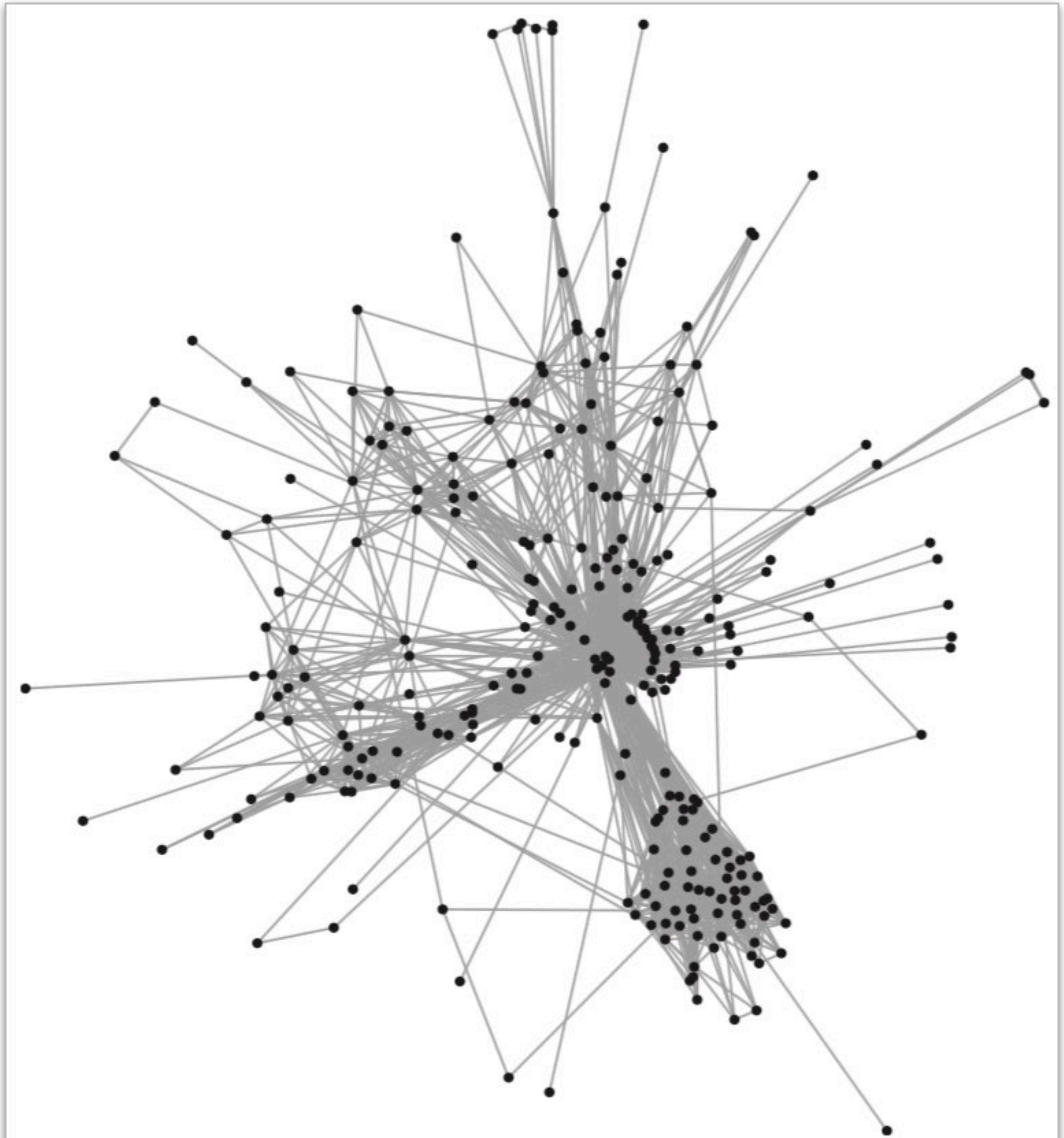
[Peysakhovich, Vsevolod et al. Attribute-driven edge bundling for general graphs with applications in trail analysis. Visualization Symposium (PacificVis), 2015 IEEE Pacific. IEEE, 2015.]

Visualizing
Multivariate Networks

Problem: represent not only the topology,
but attributes of the vertices and edges as well.

Those attributes can be nominal, ordered, quantitative, they can vary over time…

Multi-variate Networks / Available Encoding Channels



Vertex —— **Attributes**

Attribute Types

- Categorical
- Ordered
  - *Ordinal*
  - *Quantitative*

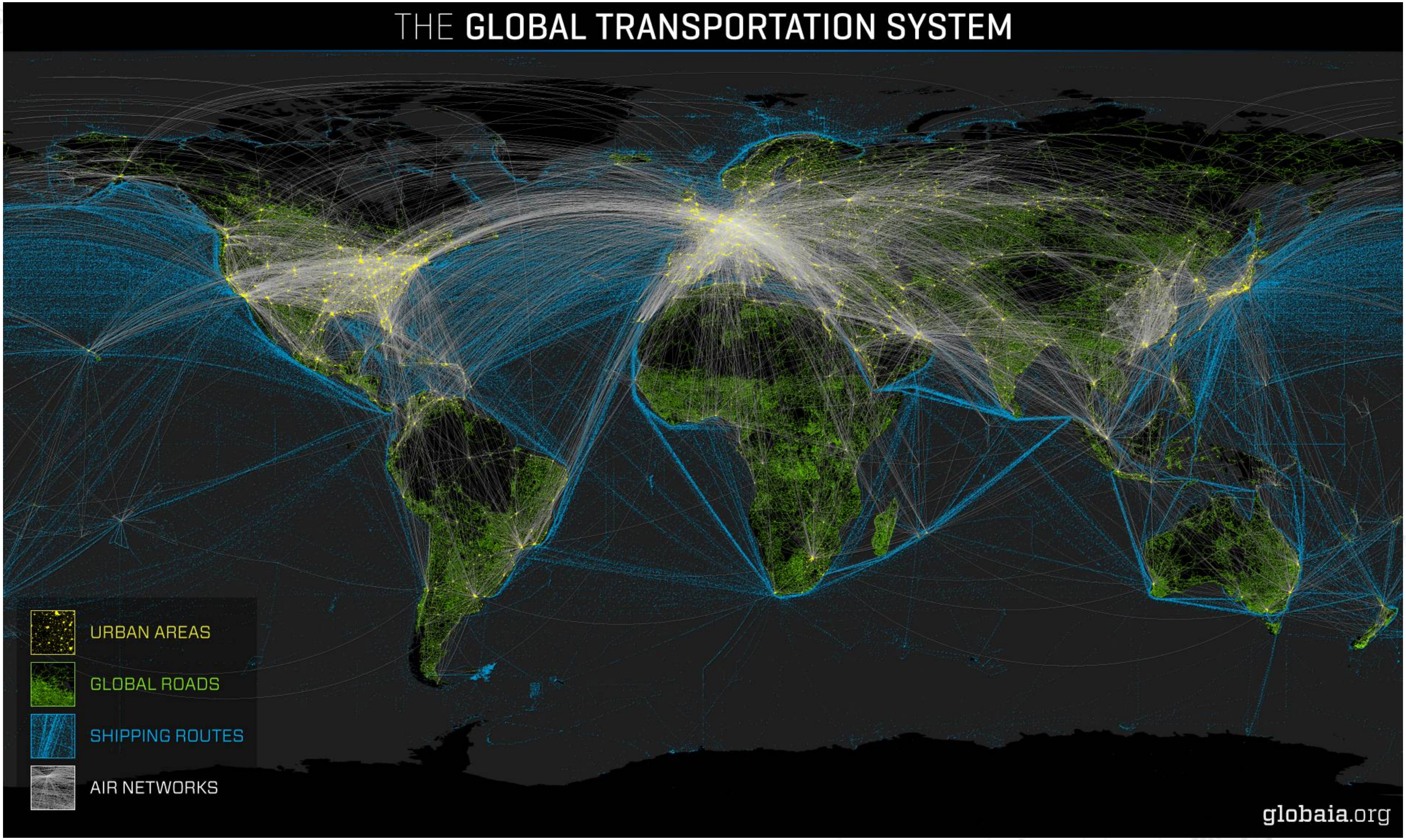
Edge —— **Attributes**

Attribute Types

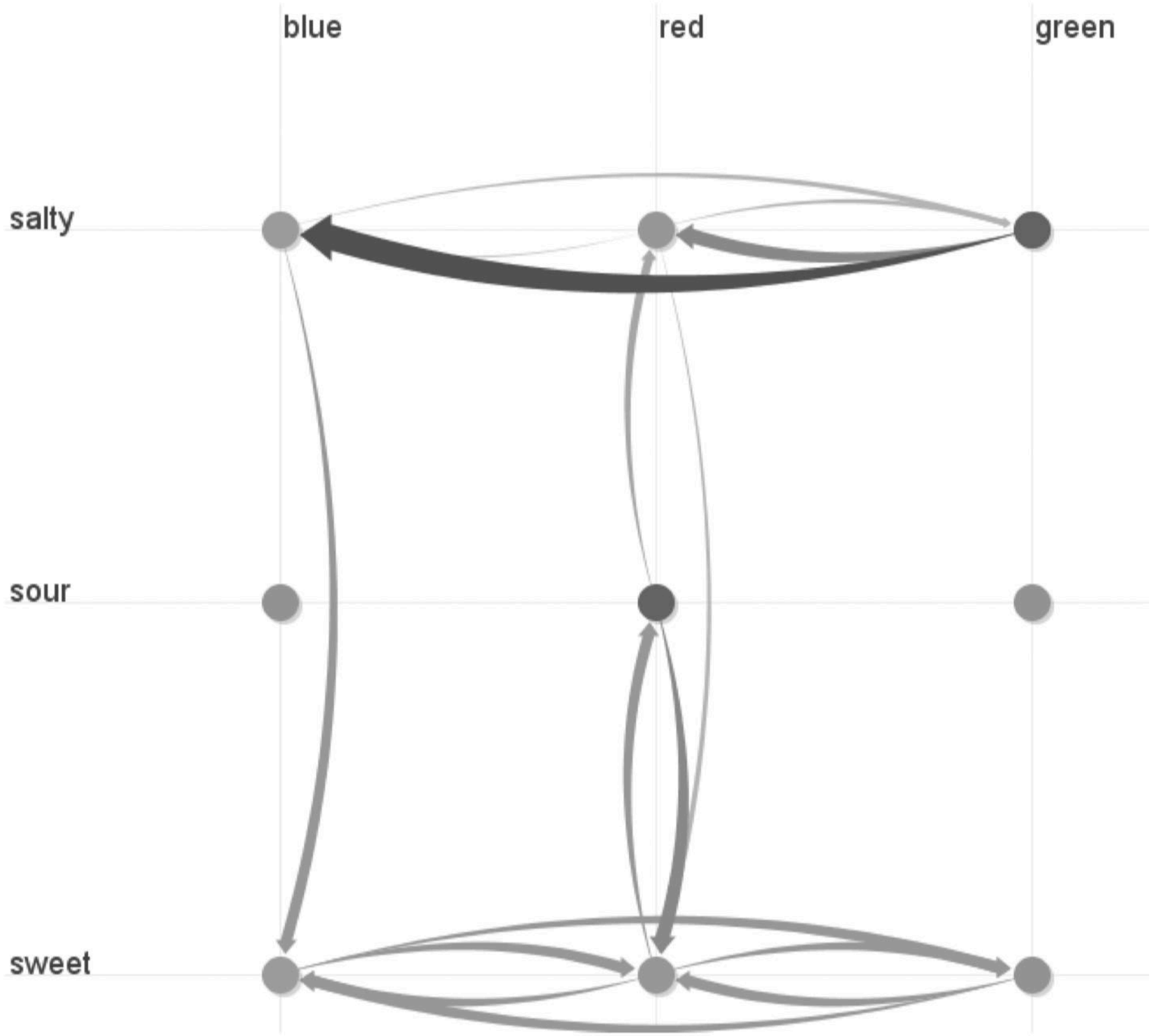
- Categorical
- Ordered
  - *Ordinal*
  - *Quantitative*

Multi-variate Networks / Available Encoding Channels

Channels: Expressiveness Types and Effectiveness Ranks

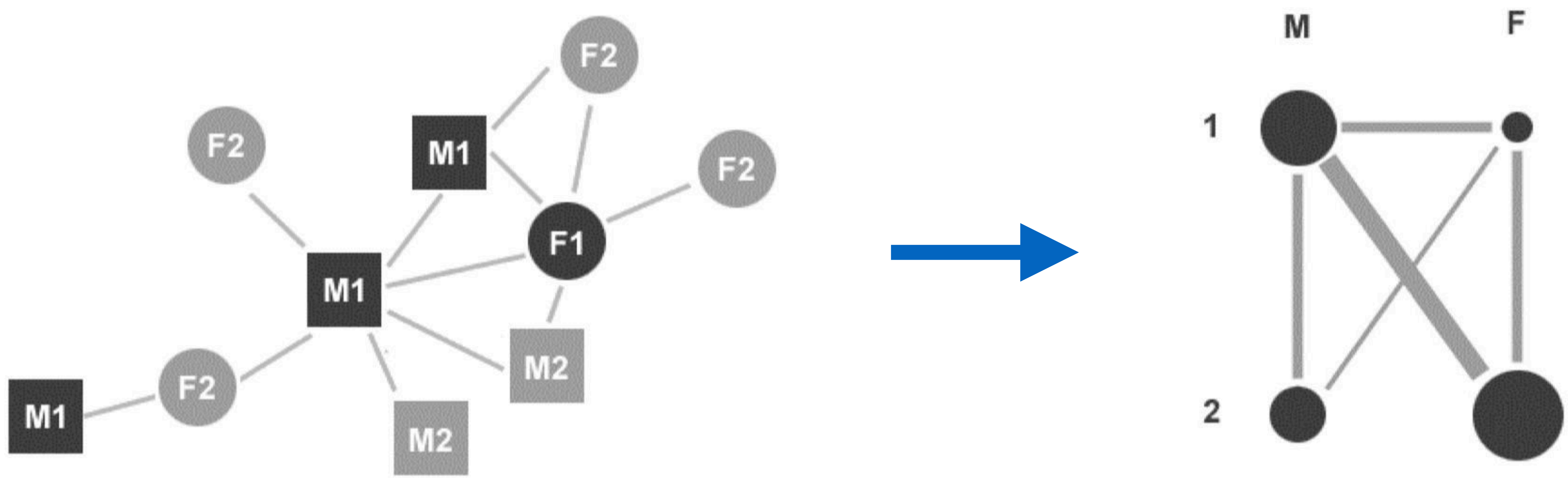


Multi-variate Networks / Aggregation using PivotGraphs



[Martin Wattenberg. 2006. Visual exploration of multivariate graphs. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06), ACM, New York, NY, USA, 811-819.]

Multi-variate Networks / Aggregation using PivotGraphs

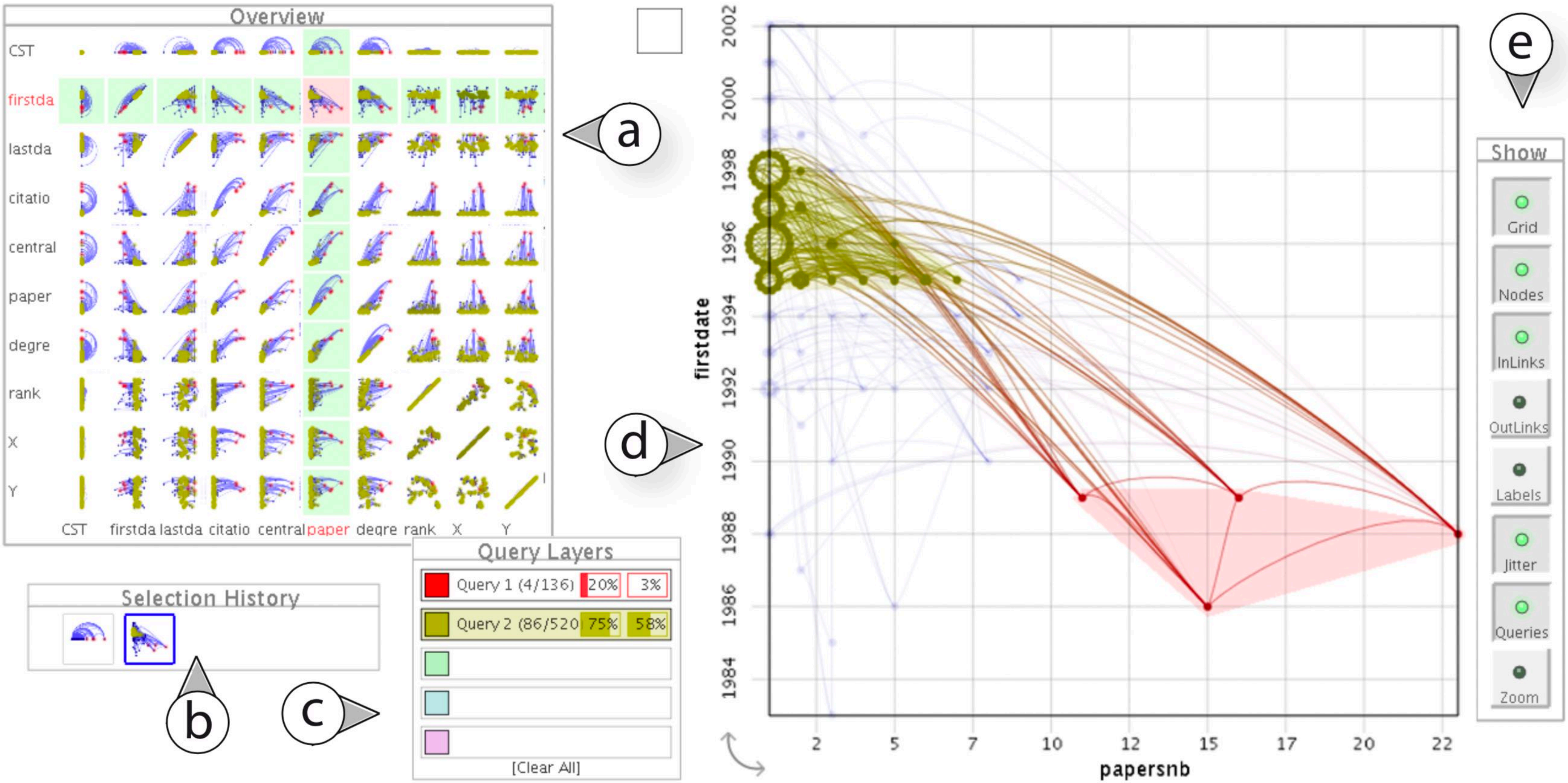


[Martin Wattenberg. 2006. Visual exploration of multivariate graphs. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06), ACM, New York, NY, USA, 811-819.]

Multi-variate Networks / GraphDice

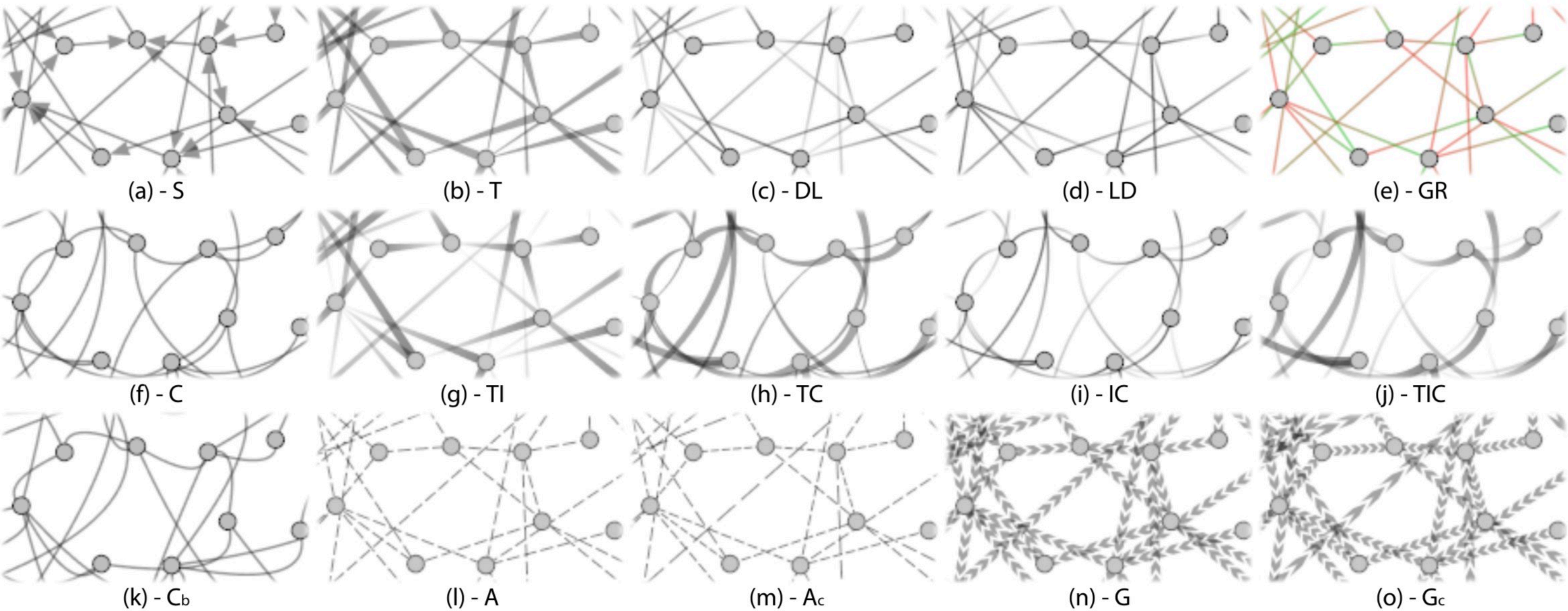
Inspired by “Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation” @ InfoVis 2008

graphdice.mp4



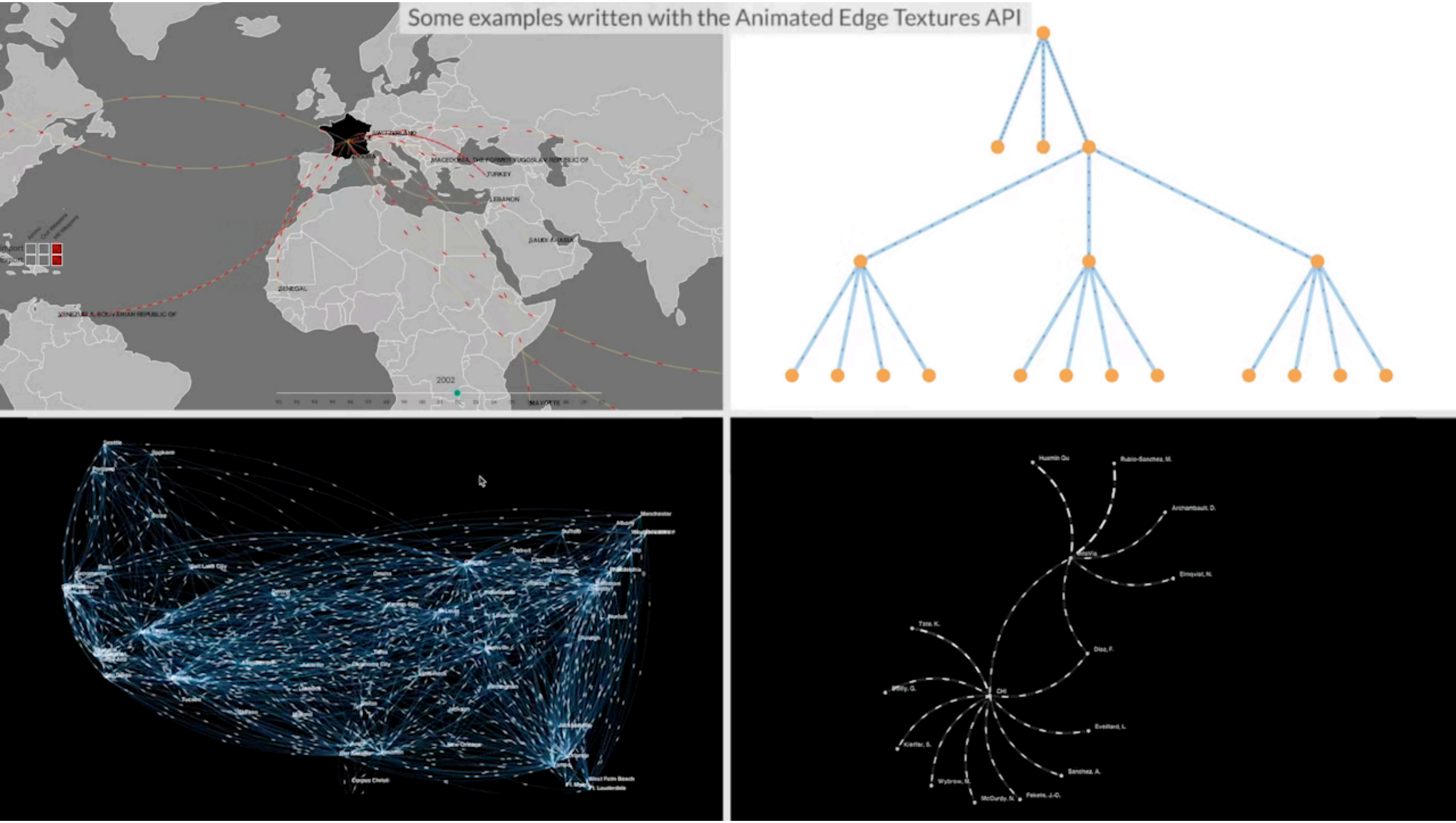
[A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist and J.D. Fekete
GraphDice: A System for Exploring Multivariate Social Networks (Eurovis 2010)]

Edge Directionality

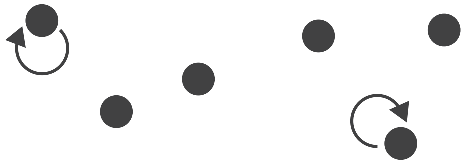


[Danny Holten et al. 2011. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. PacificVis '11. IEEE Computer Society, 195-202.]

Motion-based Encoding with Animated Edge Textures



Motion



Pattern Frequency

Particle Speed

Particle Pattern

Frequency-Adjusted Speed
(Follow Up)

[H. Romat et al., Animated Edge Textures in Node-Link Diagrams: a Design Space and Initial Evaluation, CHI '18: Proceedings of the 36th SIGCHI conference on Human Factors in computing systems, pages 187:1--187:13, April 2018]