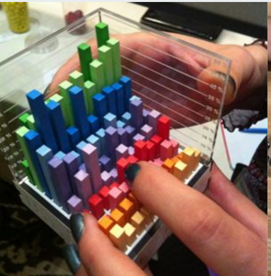
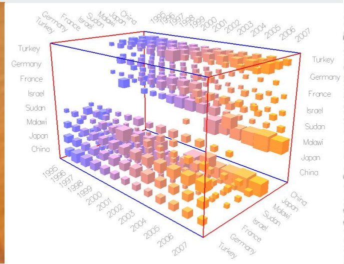
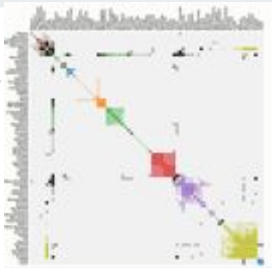
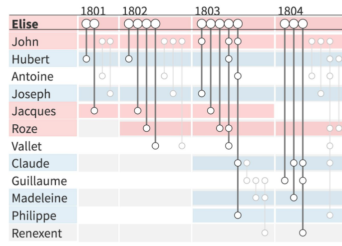
Visualization of Graphs

Jean-Daniel Fekete, Inria
http://www.aviz.fr/~fekete

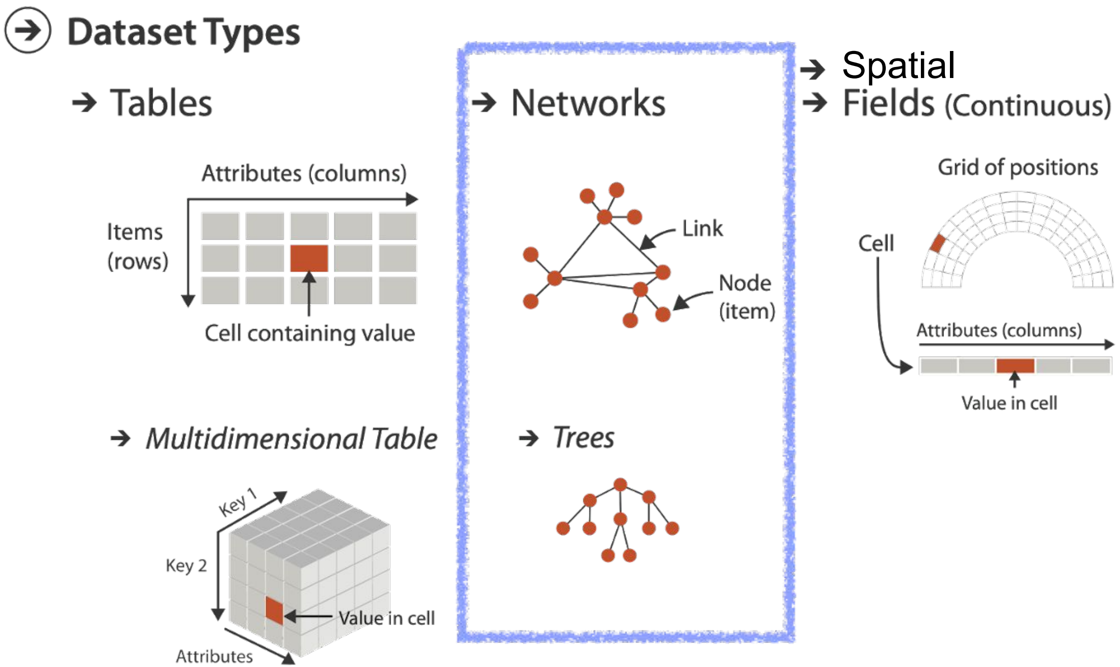


Network data

- networks
  - model relationships between things
    - aka graphs
  - two kinds of items, both can have attributes
    - nodes
    - links
- tree
  - special case
  - no cycles
    - one parent per node

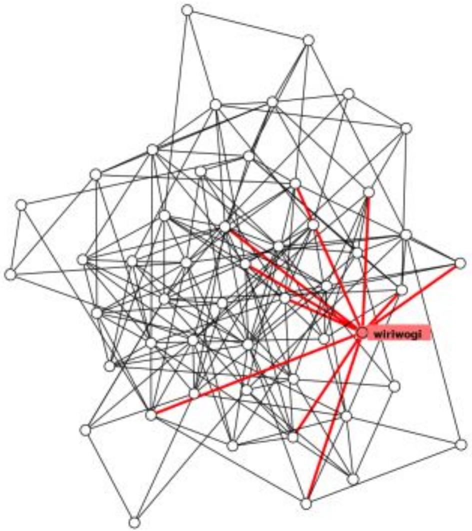


Formal definition

- A graph is:
  - A set of Vertices V={v_i}
  - A set of edges E={e_j} with e=(v_s,v_d) ∈ VxV
  - When the order of the couple in E is meaningful, the graph is **directed**, otherwise, it is **undirected**
  - A graph is a mapping of V into V
  - From there, we can define several measures or intrinsic properties on a graph

Network tasks: topology-based and attribute-based

- topology based tasks
  - find paths
  - find (topological) neighbors
  - compare centrality/importance measures
  - identify clusters / communities
- attribute based tasks (similar to table data)
  - find distributions, ...
- combination tasks, incorporating both
  - example: find friends-of-friends who like cats
    - topology: find all adjacent nodes of given node
    - attributes: check if has-pet (node attribute) == cat



Issues

- Graph Layout
- Scalability
- Navigation

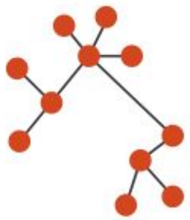
Beware of the hairball!

Arrange Networks and Trees

➔ Node–Link Diagrams

Connection Marks

✔ NETWORKS ✔ TREES



➔ Adjacency Matrix

Derived Table

✔ NETWORKS ✔ TREES

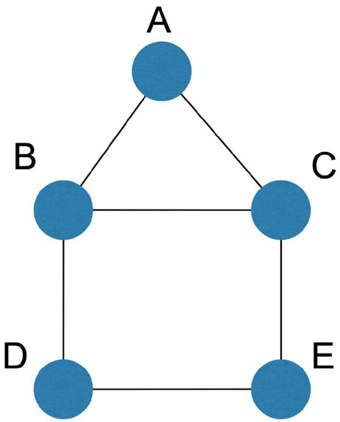
➔ Enclosure

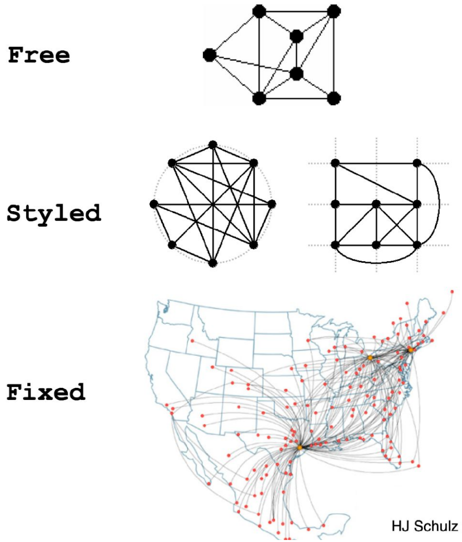
Containment Marks

✖ NETWORKS ✔ TREES

Node-link diagrams

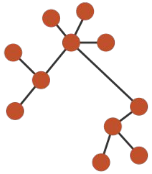
- nodes: point marks
- links: line marks
  - straight lines or arcs
  - connections between nodes
- intuitive & familiar
  - most common
  - many, many variants





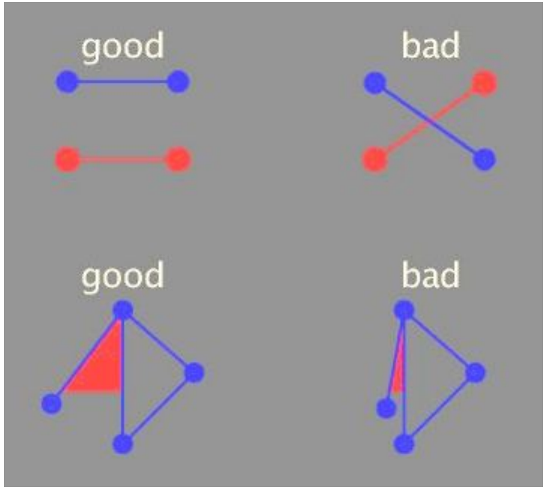
HJ Schulz 2006

➔ **Node–Link Diagrams**
Connection Marks
✔ NETWORKS ✔ TREES



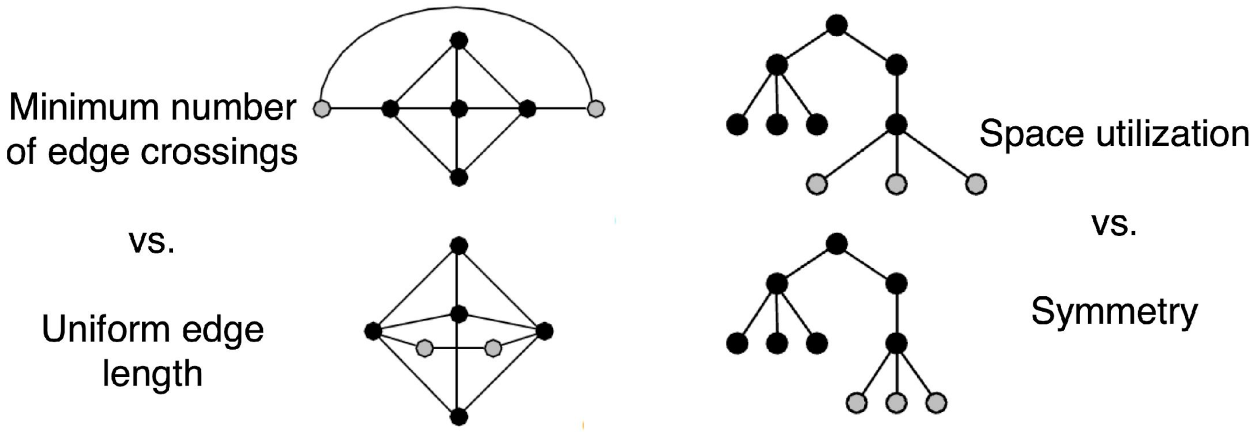
Criteria for good node-link layouts

- minimize
  - edge crossings, node overlaps
  - distances between topological neighbor nodes
  - total drawing area
  - edge bends
- maximize
  - angular distance between different edges
  - aspect ratio disparities
- emphasize symmetry
  - similar graph structures should look similar in layout

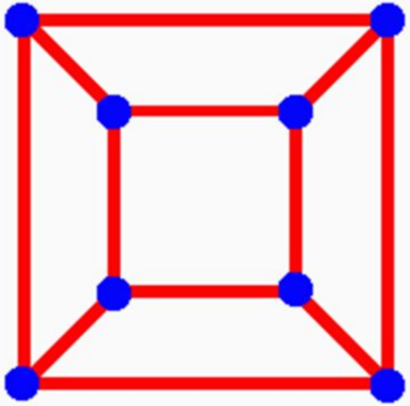
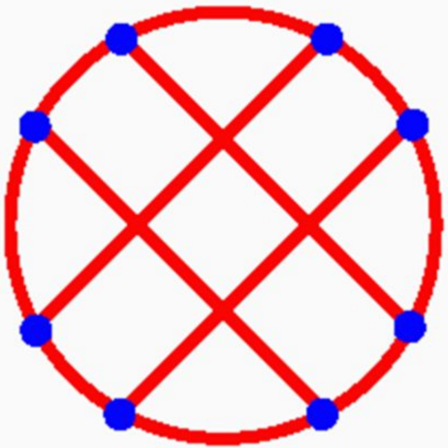
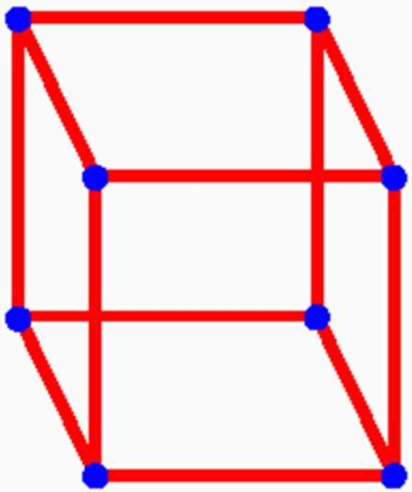
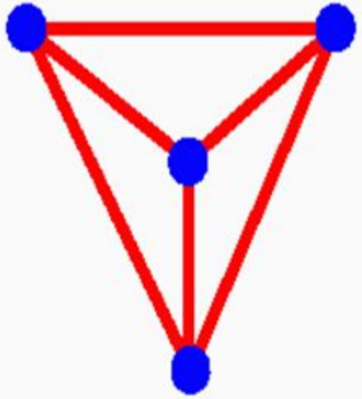
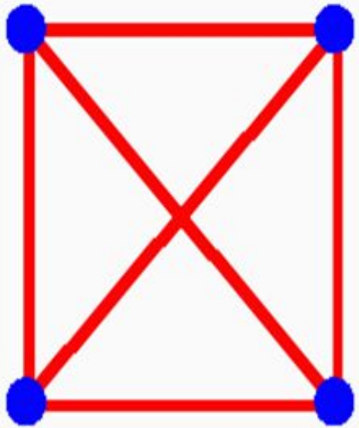


Criteria conflict

- most criteria NP-hard individually
- many criteria directly conflict with each other



Schulz 2004

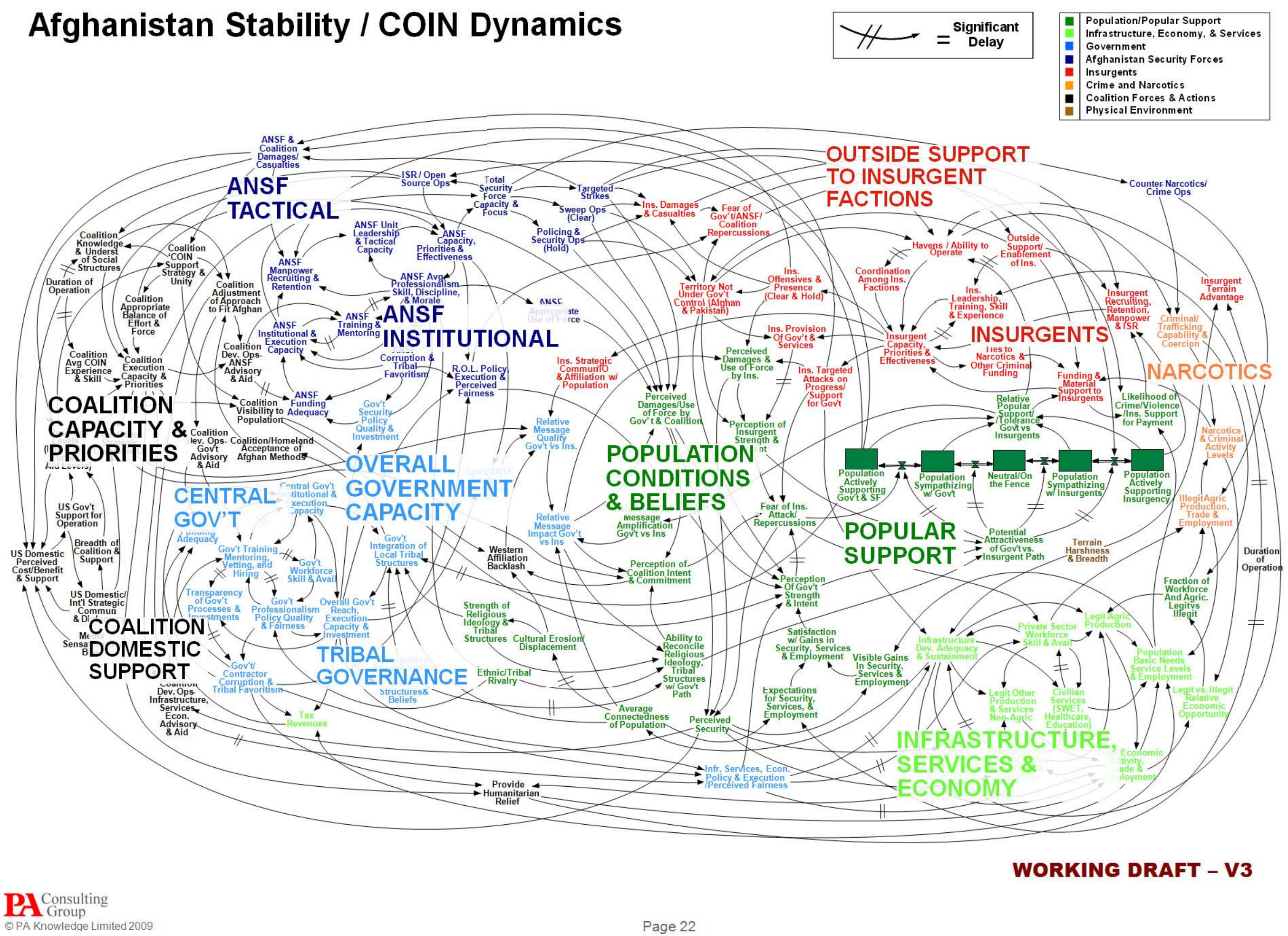






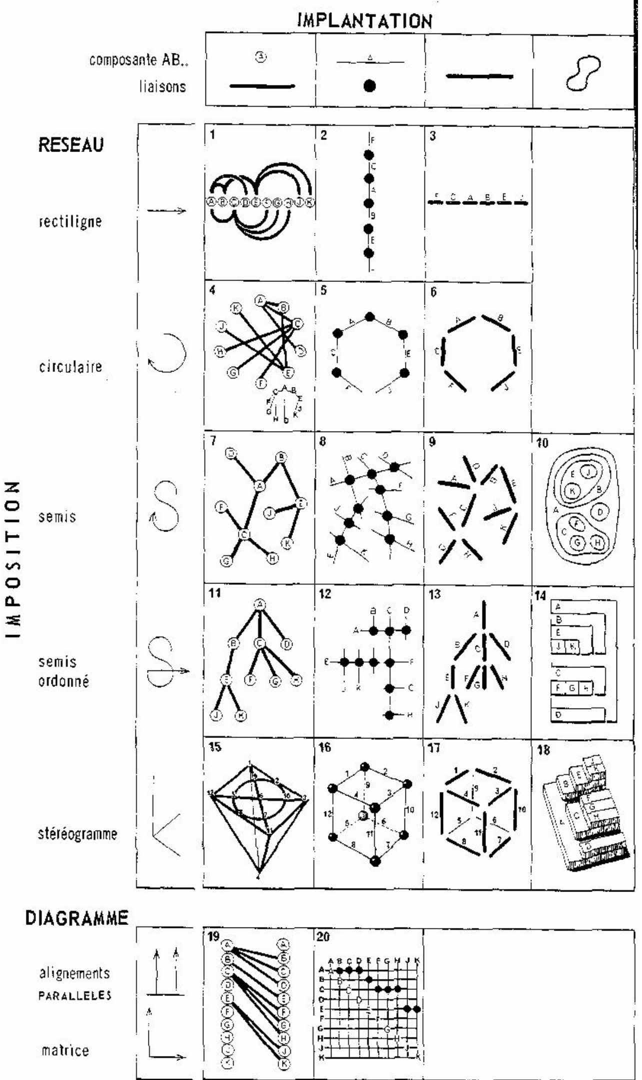
Optimization-based layouts

- formulate layout problem as optimization problem
- convert criteria into weighted cost function
  - F(layout) = a*[crossing counts] + b*[drawing space used]+...
- use known optimization techniques to find layout at minimal cost
  - energy-based physics models
  - force-directed placement
  - spring embedders


facebook
December 2010

Bummiler, 2010

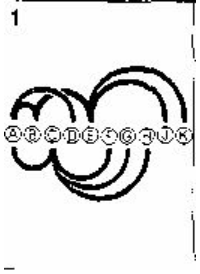


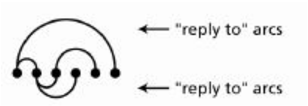

IMPLANTATION
composante AB..
liaisons
RESEAU
rectiligne
circulaire
semis
semis ordonné
stéréogramme
IMPOSITION
DIAGRAMME
alignements
PARALLELES
matrice
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

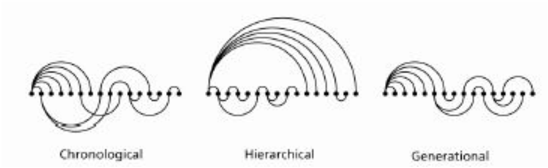
Bertin
67

Linear layout

- **THREAD ARCS: An Email Thread Visualization** (Bernard Kerr, IBM)



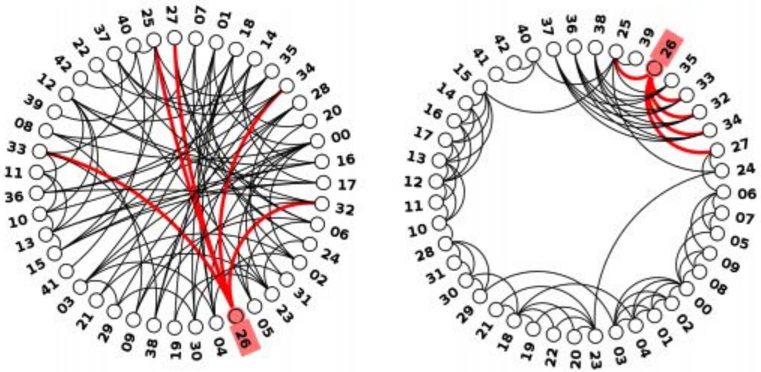


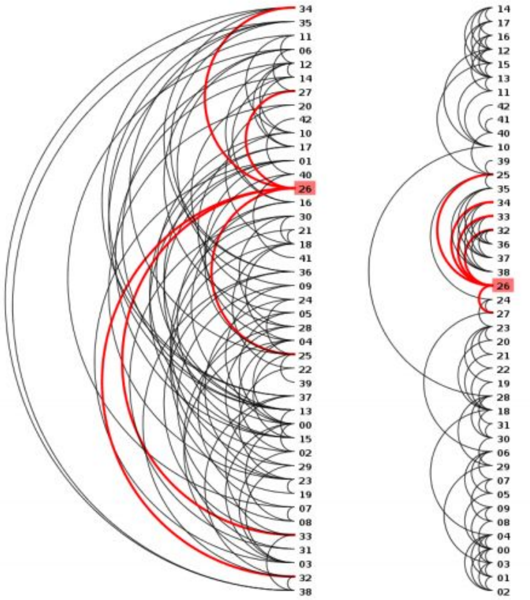


- Arc Diagrams: The Shape of Song (Martin Wattenberg, IBM)
- http://www.bewitched.com/

Idiom: **circular layouts / arc diagrams (node-link)**

- restricted node-link layouts: lay out nodes around circle or along line
- data
  - original: network
  - derived: node ordering attribute (global computation)
- considerations: node ordering crucial to avoid excessive clutter from edge crossings
  - examples: before & after barycentric ordering





fs.etsmtl.ca/mmcguffin/research/2012-mcguffin-simpleNetVis/mcguffin-2012-simpleNetVis.pdf

Linear Layout Theory

- Order vertices so that a structure appears
  - Layout φ
- Some well-known objective functions (Diaz et al. 02):
  - Bandwidth
  - MinLA
  - Cut Width
  - Min Cut
  - Sum Cut

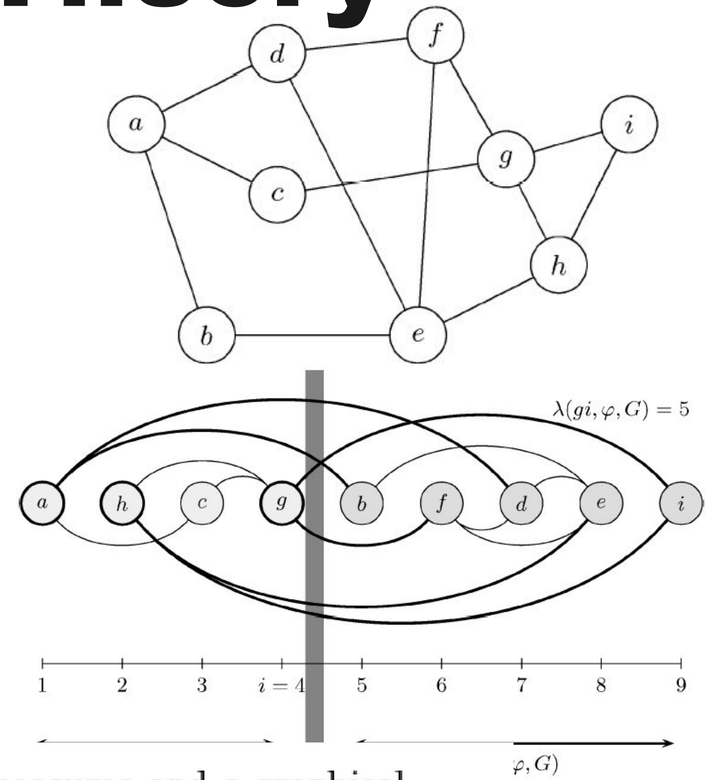
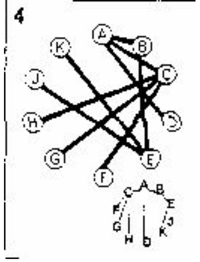
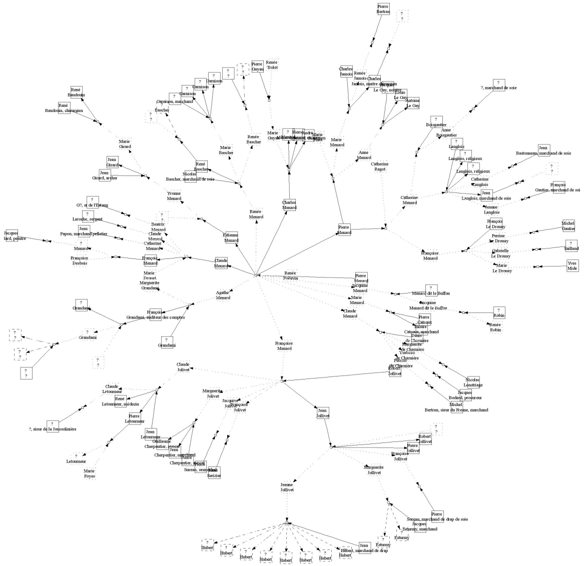


Fig. 1. A graph G together with some layout measures and a graphical representation of the layout $\varphi = \{(a, 1), (b, 5), (c, 3), (d, 7), (e, 8), (f, 6), (g, 4), (j, 9), (h, 2)\}$.

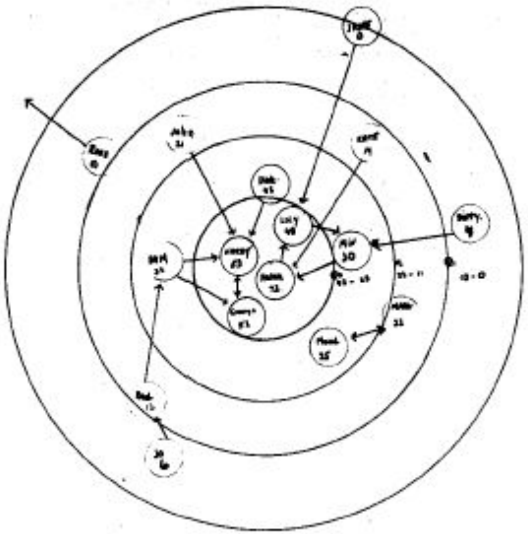
Circular Layout

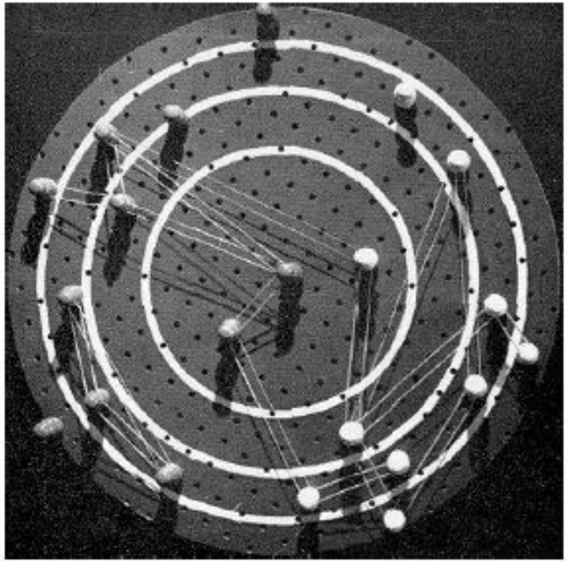
- Egocentric Social Networks
- Can be ordered too





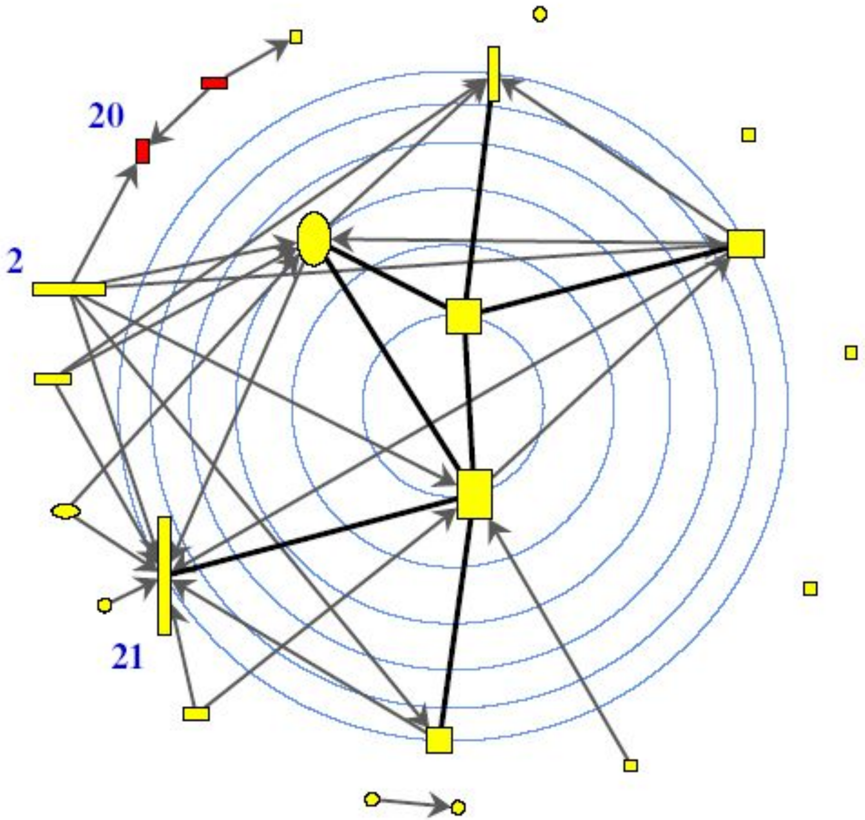
centrality visualization
Northway (1940)





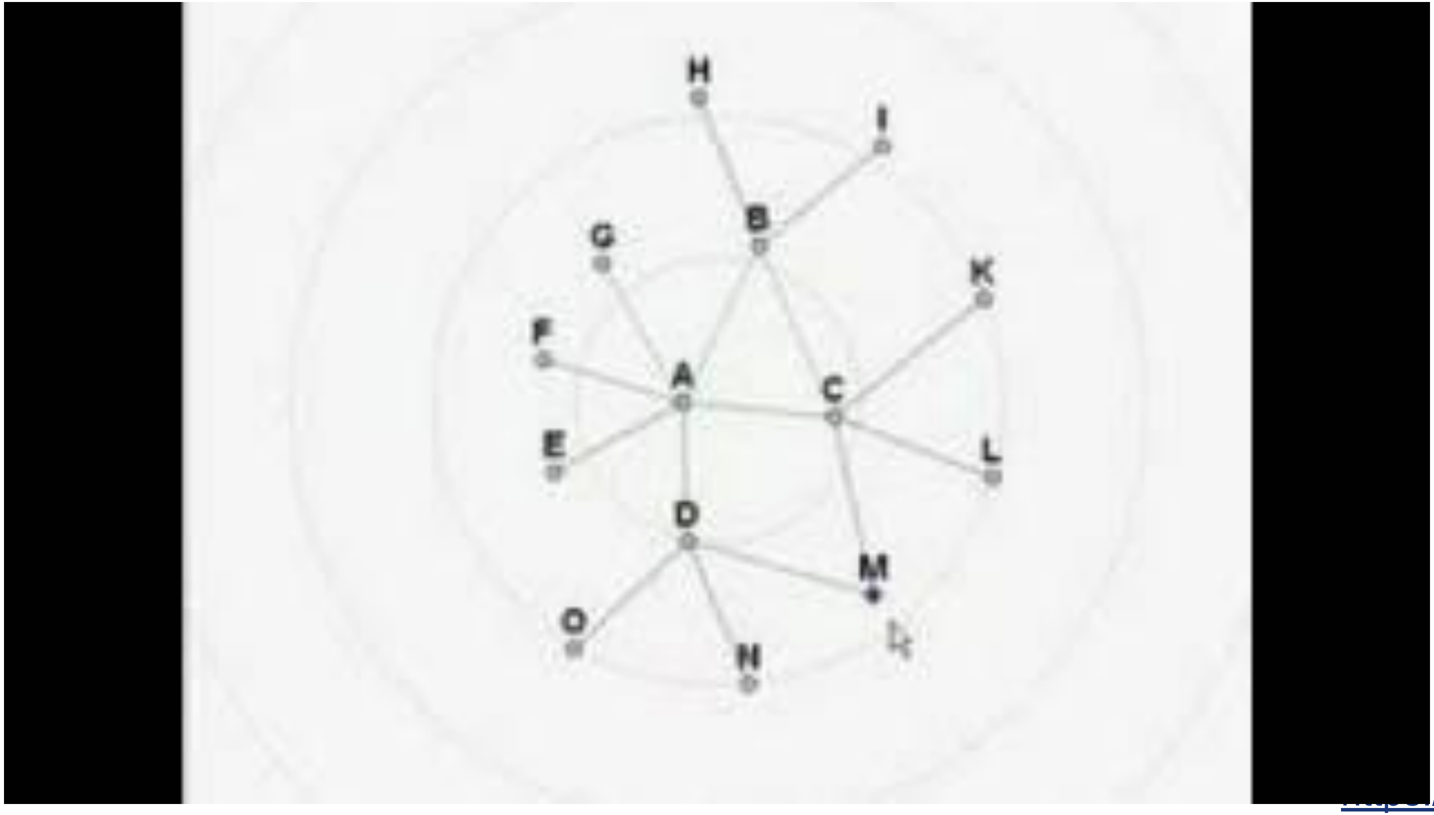
centrality visualization

network of organizations involved in policy making



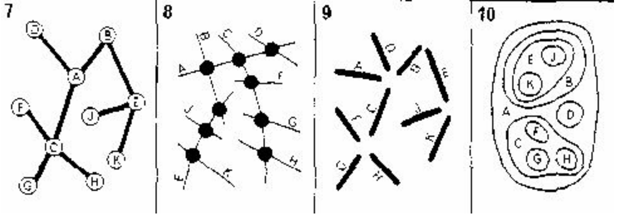
Animation of a circular network

[Animated Exploration of Graphs with Radial Layout, InfoVis'01]



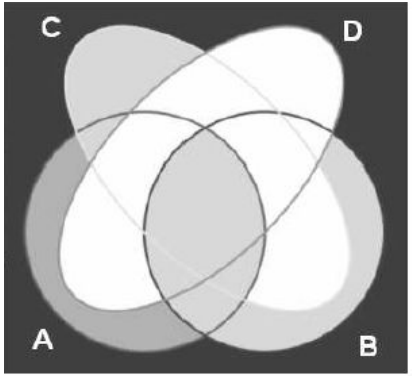
https://youtu.be/OPX5iGro_IA

Free Layout

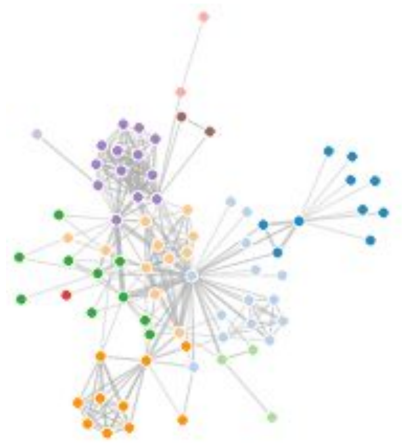


- Order is not meaningful, only proximity is
- Force-directed methods
- Treemaps and Venn Diagrams

for $t \leftarrow 1$ **to** *ITERATIONS* **do**
  for $v \in V$ **do**
    $D \leftarrow \sum_{u:\{u,v\} \notin E} f_{\text{rep}}(p_u, p_v) + \sum_{u:\{u,v\} \in E} f_{\text{spring}}(p_u, p_v);$
    $p_v \leftarrow p_v + \delta \cdot D$



Idiom: **force-directed placement**



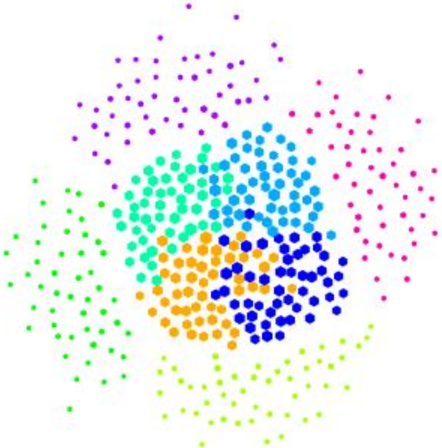
- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length
      - long edges more visually salient than short
- tasks
  - explore topology; locate paths, clusters
- scalability
  - node/edge density E < 4N

http://mbostock.github.com/d3/ex/force.html

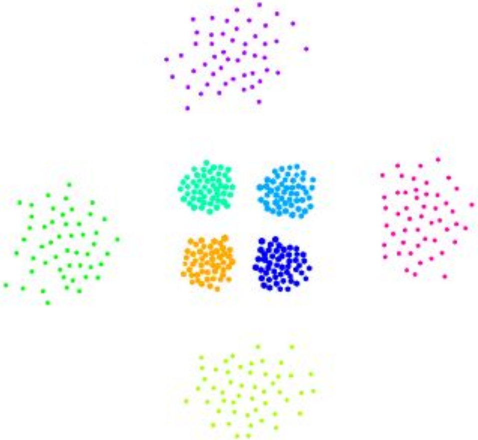
Force-directed Layout

- Spring Model: f_{rep} linear, f_{att} linear
  - Good space filling when the network is sparse and homogeneous
- Kamada&Kawai : f_{rep} linar, f_{att} function of the Graph-theoretical distrance
  - Shows the shortest paths as quasi straight lines
- Fruchterman&Reingold : f_{att} quadratic
  - Fills the space better for varying degrees
- Davidson&Harel : f_{rep} linear, f_{att} linear*degree
  - Better space filling for heterogenous graphs
- Hall/Spectral
  - Very fast but works best with grids
- LinLog : f_{rep} linear, f_{att} linear+log
  - Show clusters
- New algorithms all the time
- Now, merging with multidimensional projections

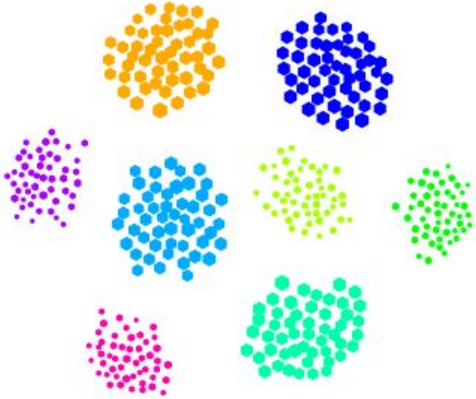
Examples



Fruchterman-Reingold model

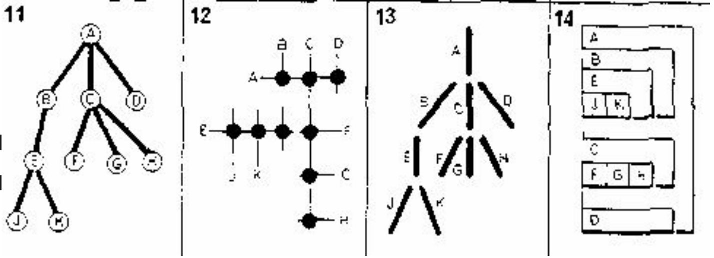


Node-repulsion LinLog

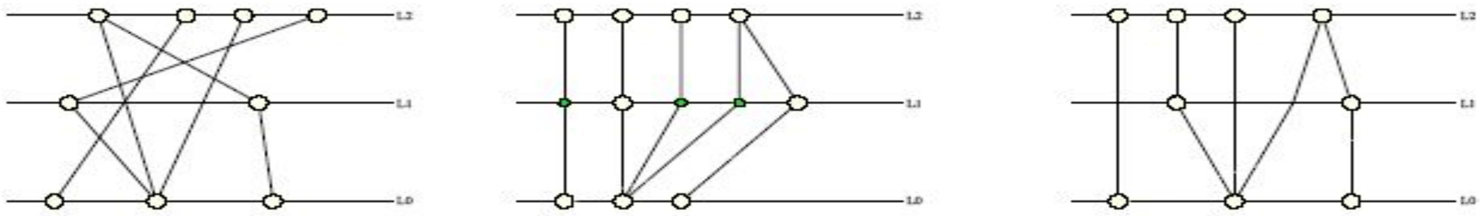


Edge-repulsion LinLog model

Ordered layout



- One dimension is orderable (e.g. genealogical trees)
  - The other can be partially ordered
- Sugiyama algorithm
- Improved by C. North [A Technique for Drawing Directed Graphs.]

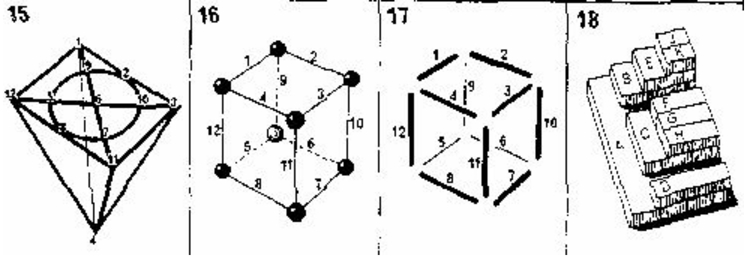


vertical ordering
layer assignment

horizontal ordering
crossing reduction

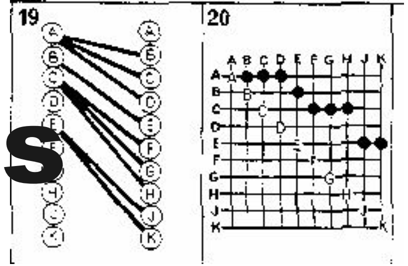
coordinate assignment
spacing and straightness

3D Networks



- 3D methods usually extend to 3D naturally
- Navigation is then required

Diagrams and Matrices



Directed network

$$A \rightarrow B \rightarrow C \rightarrow D$$

Adjacency matrix

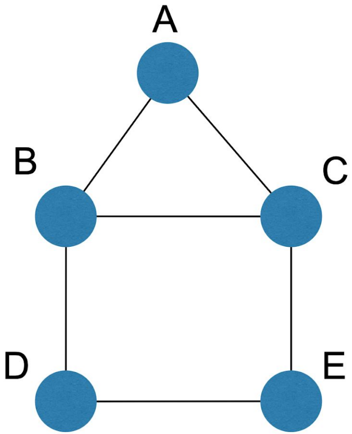
Destination/target

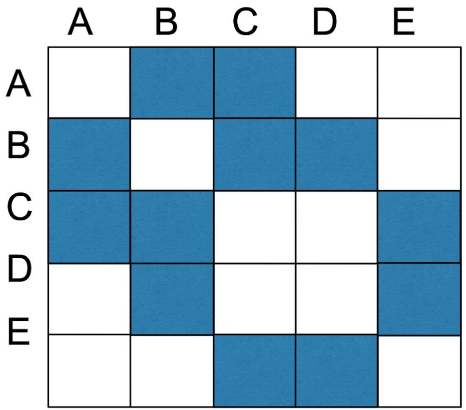
$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 0 \\ C & 0 & 0 & 0 & 1 \\ D & 0 & 0 & 0 & 0 \end{array}$$

Source

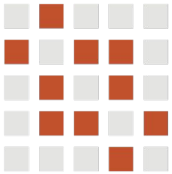
Adjacency matrix representations

- derive adjacency matrix from network

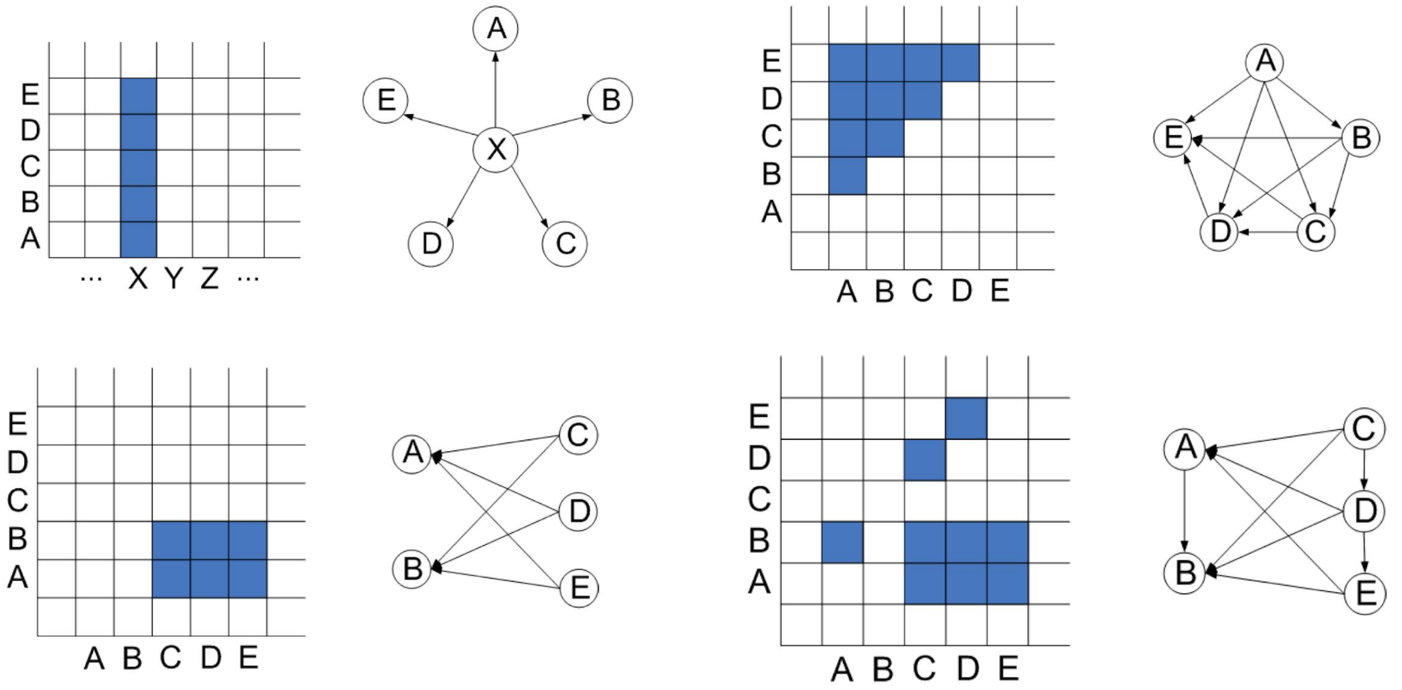




Adjacency Matrix
Derived Table
✔ NETWORKS ✔ TREES

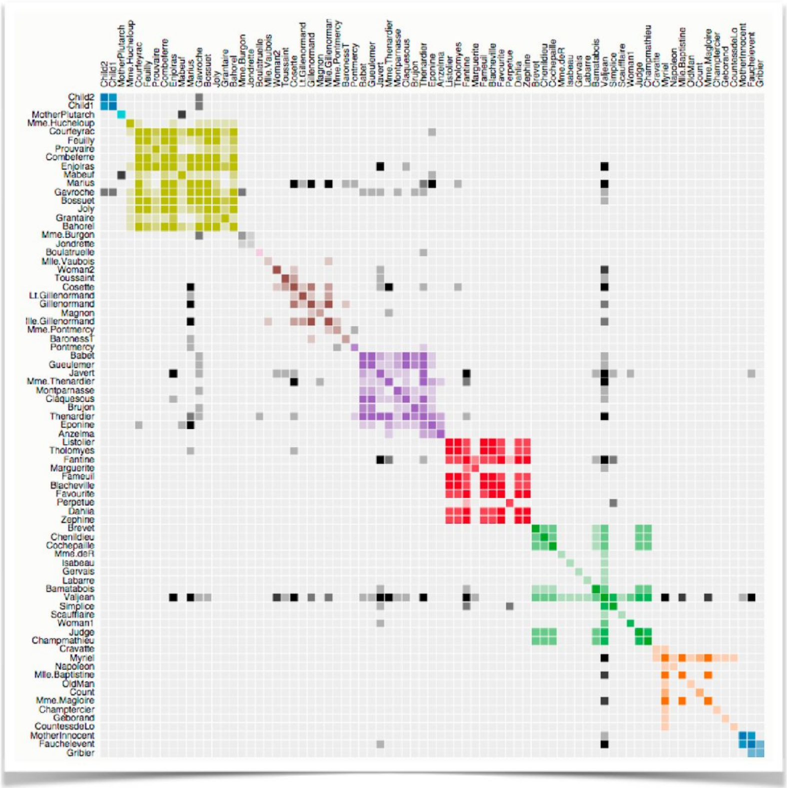


Adjacency matrix examples



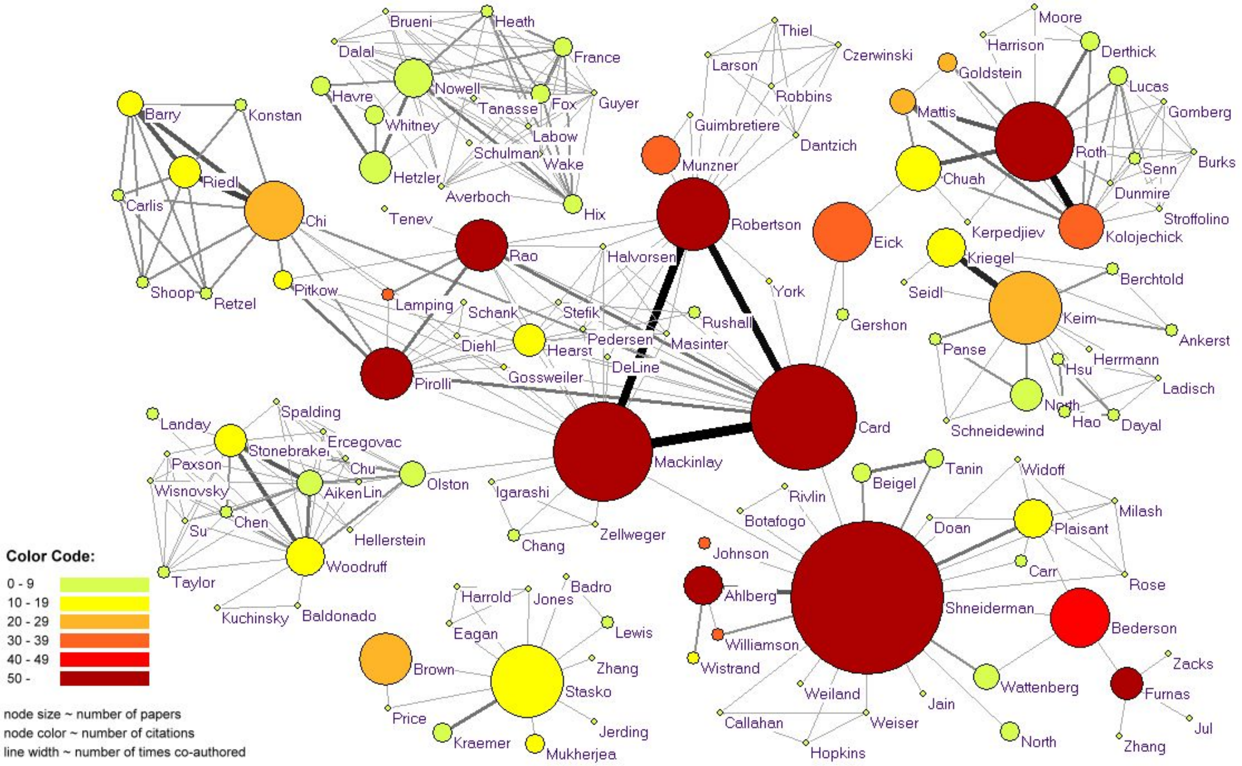
HJ Schulz 2007

Node order is crucial: Reordering

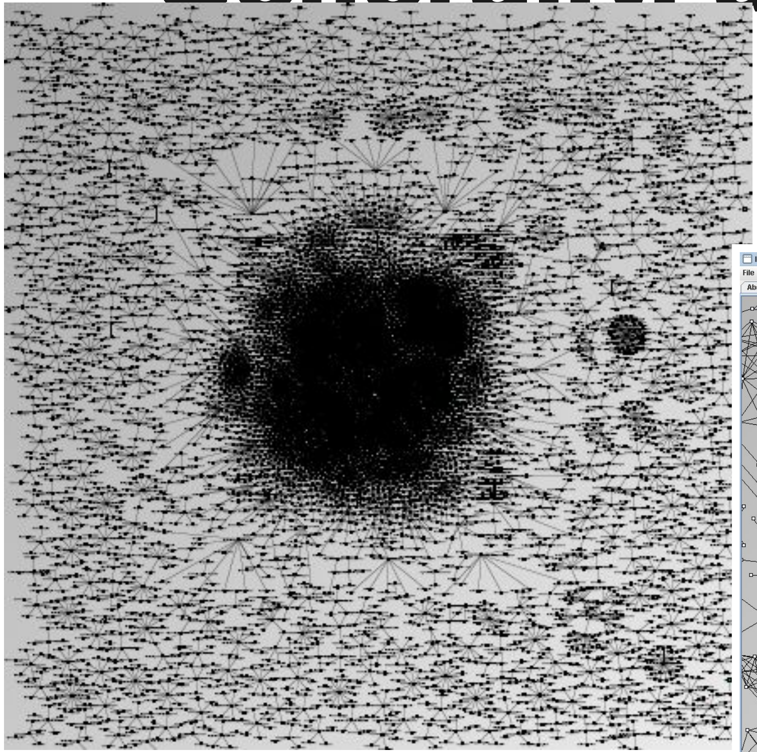


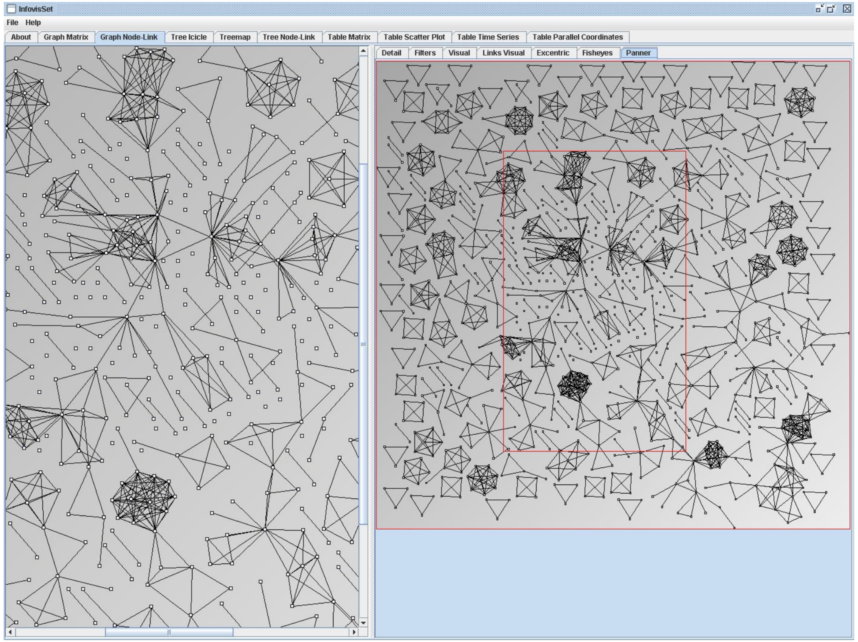
https://bost.ocks.org/mike/miserables/

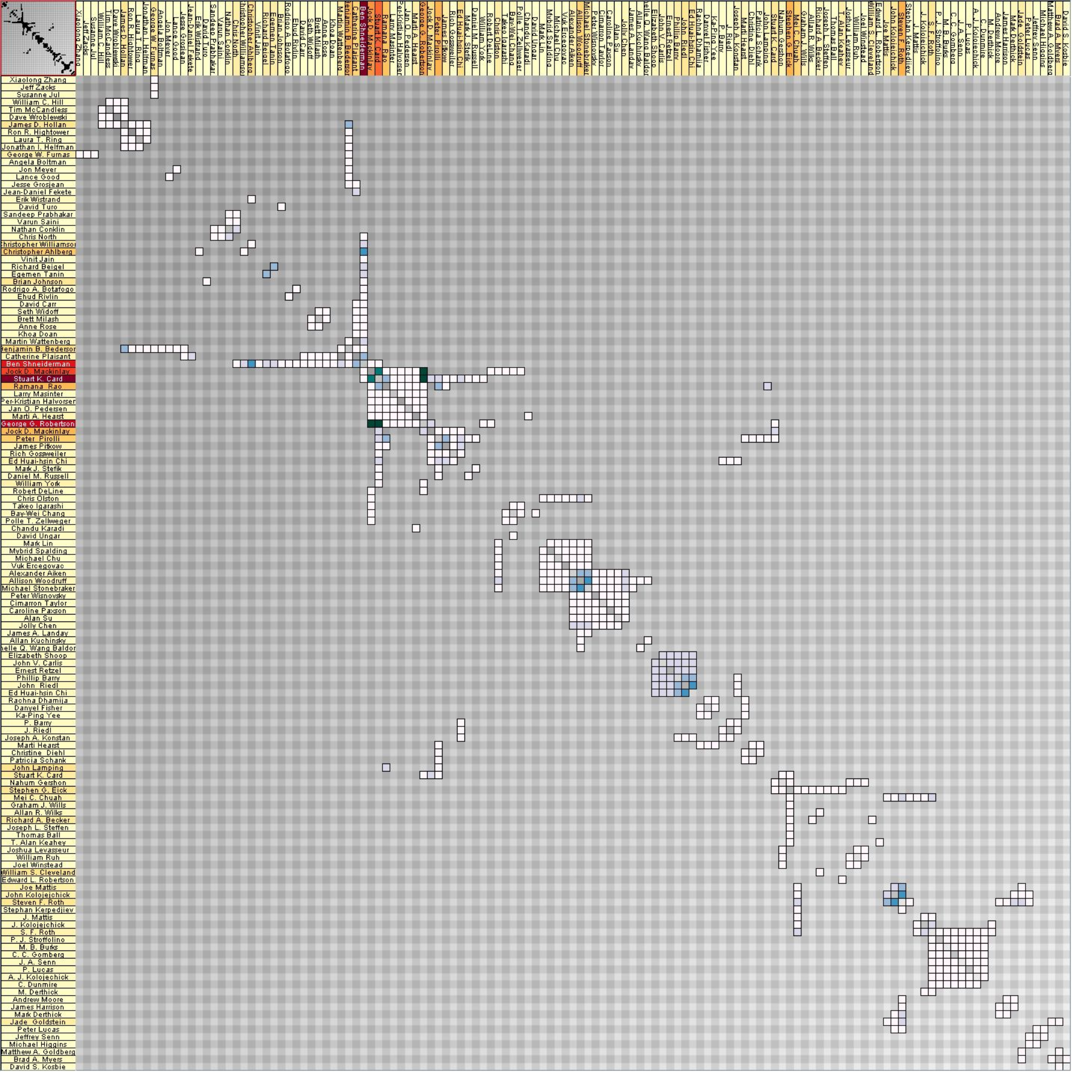
InfoVis Co-authoring (K. Börner et

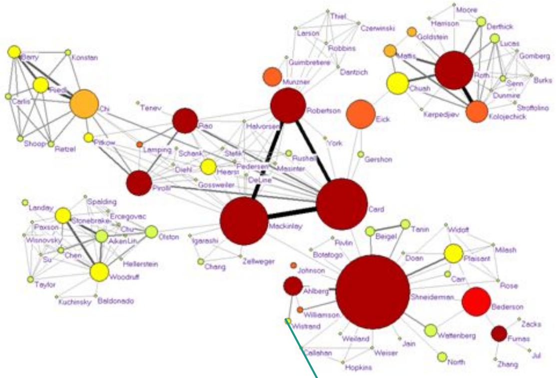


Generally, after loading...



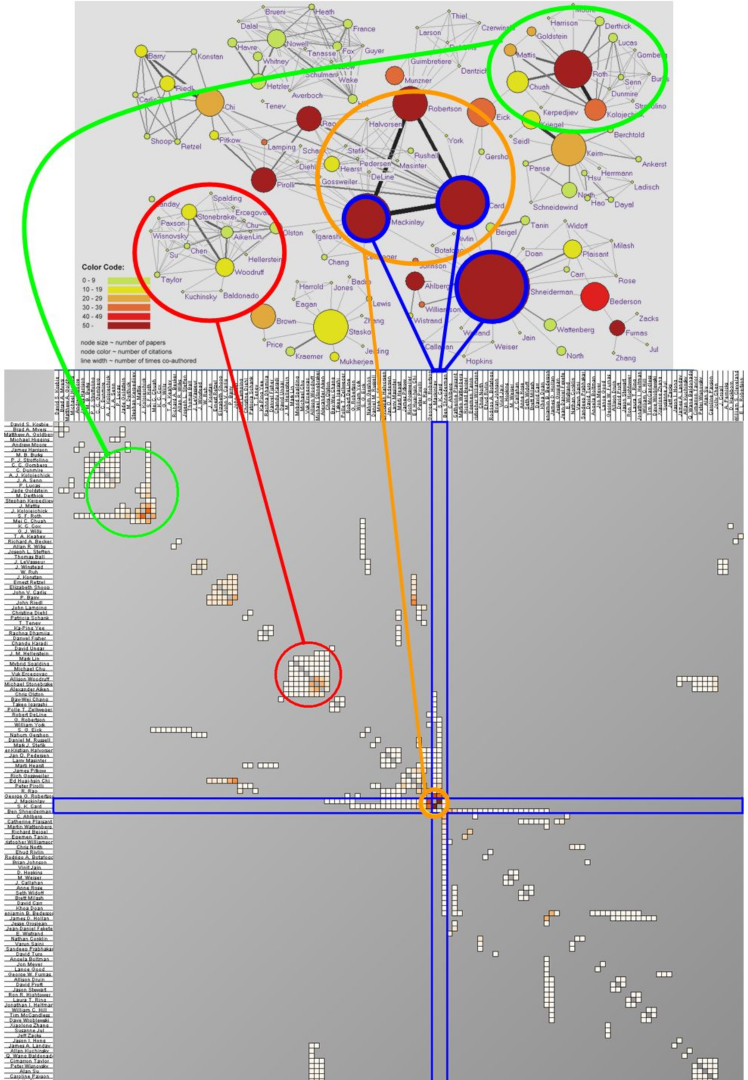




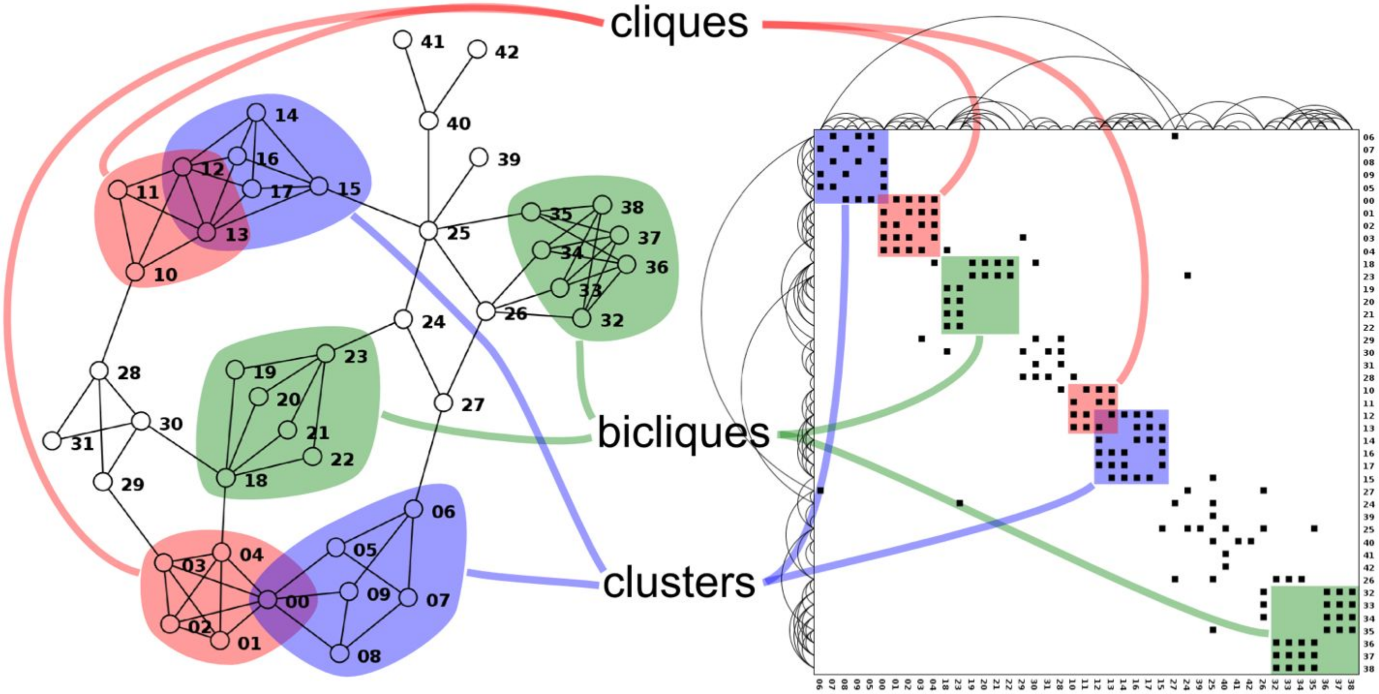


Communicate

Explore

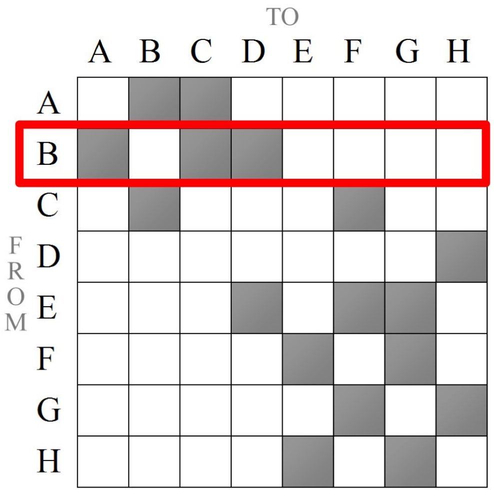

Color Code:
0 - 9
10 - 19
20 - 29
30 - 39
40 - 49
50 -
node size ~ number of papers
node color ~ number of citations
line width ~ number of times co-authored

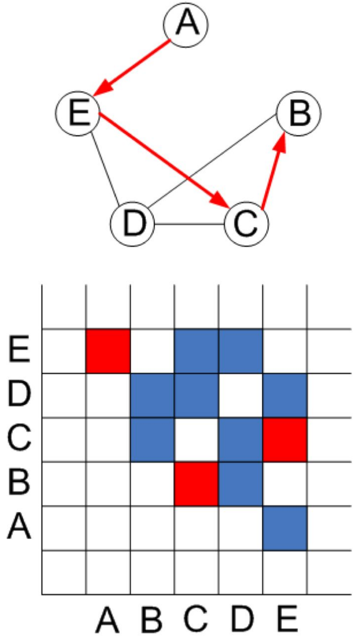
Structures visible in both



http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

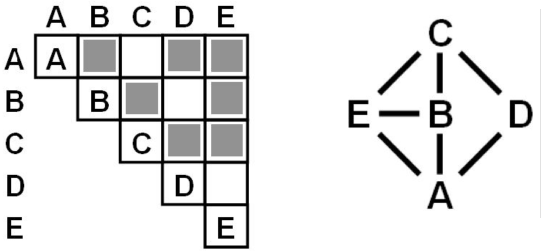
Adjacency matrix



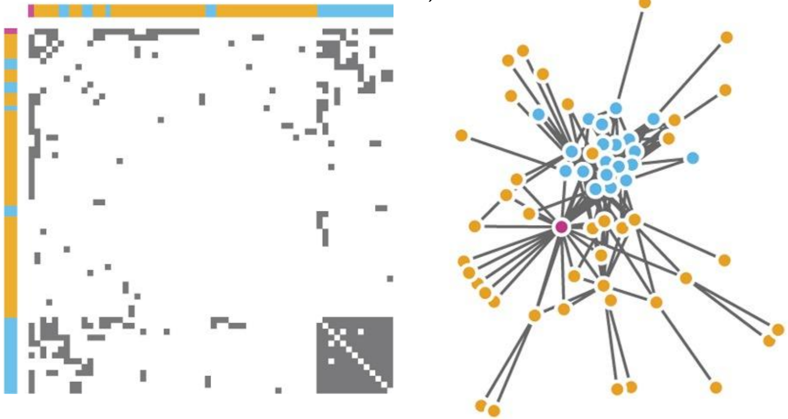


Idiom: **adjacency matrix view**

- data: network
  - transform into same data/encoding as heatmap
- derived data: table from network
  - 1 quant attrib
    - weighted edge between nodes
  - 2 categ attribs: node list x 2
- visual encoding
  - cell shows presence/absence of edge
- scalability
  - 1K nodes, 1M edges



[NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007.]

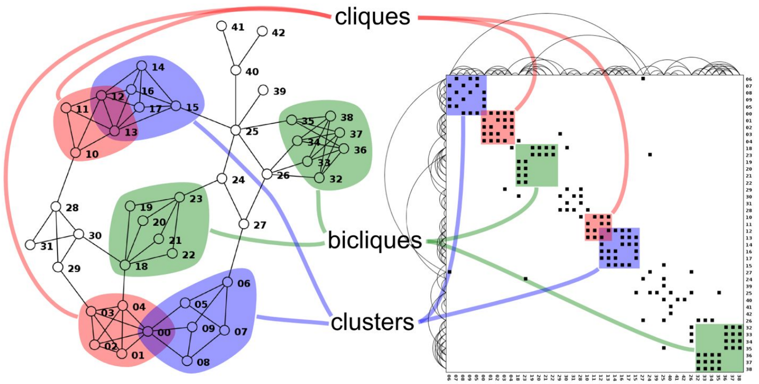


[Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115.]

Node-link vs. matrix comparison

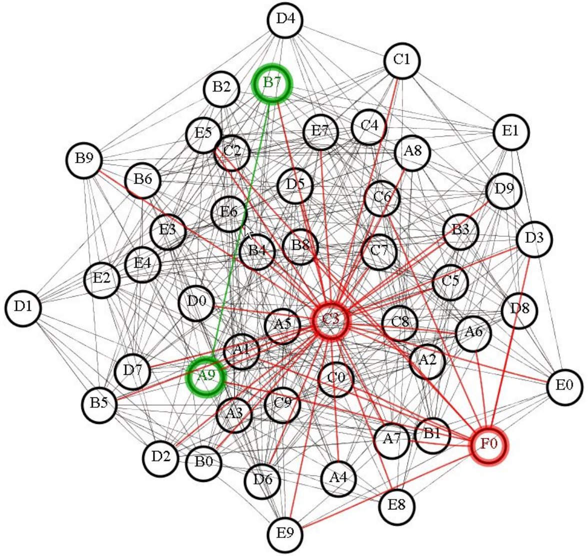
- node-link diagram strengths
  - topology understanding, path tracing
  - intuitive, flexible, no training needed
- adjacency matrix strengths
  - focus on edges rather than nodes
  - layout straightforward (reordering needed)
  - predictability, scalability
  - some topology tasks trainable
- empirical study
  - node-link best for small networks
  - matrix best for large networks
    - if tasks don't involve path tracing!

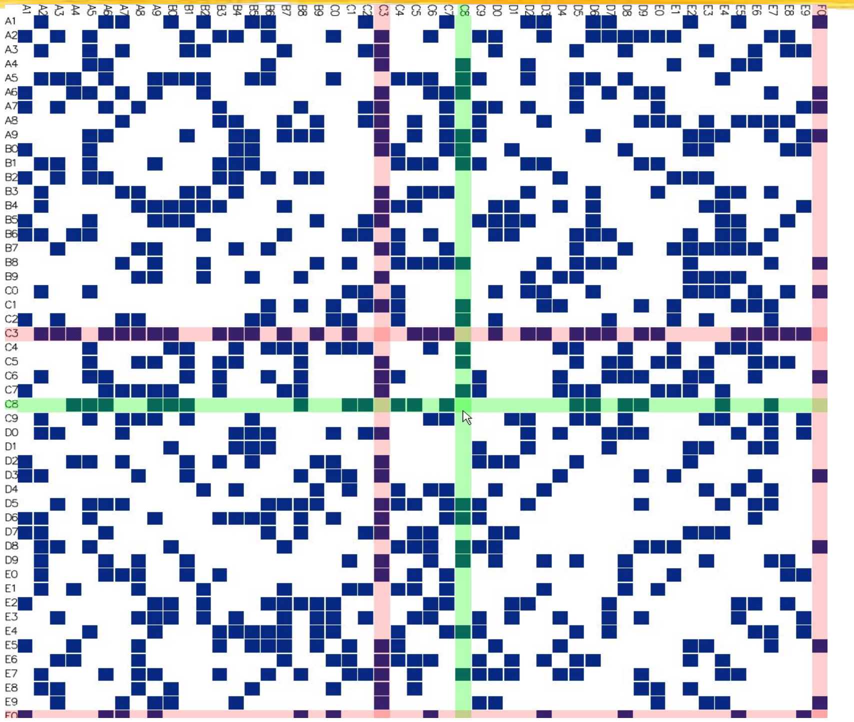
[On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. Ghoniem, Fekete, and Castagliola. Information Visualization 4:2 (2005), 114–135.]



http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

Readability Experiment



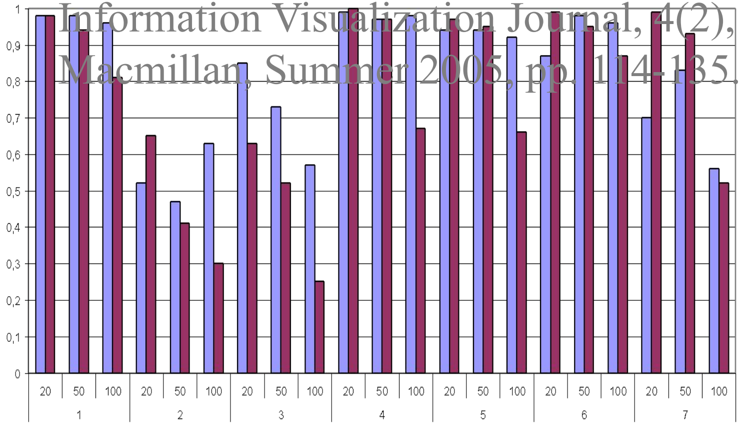


Controlled Experiment: Node Link Diagrams vs. Adjacency Matrices

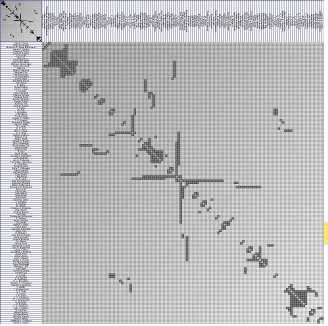
- **The Tasks:**
- Tasks related to the overview
  - Number of vertices
  - Number of arcs
- Tasks related to graph elements
  - Finding an element (a vertex, a link)
  - Finding the most connected vertex (a central actor, a pivot, a hub)
  - Finding a common neighbor
  - Finding a path
- Random graphs (3 sizes et 3 densities)
- 2 representations: Node-Link + Matrix
- **Results:**
- Node-link diagrams are preferable for small sparse graphs (20 vertices)

Matrices are more readable wrt dense graphs and medium/large graphs (> 20 vertices) wrt the

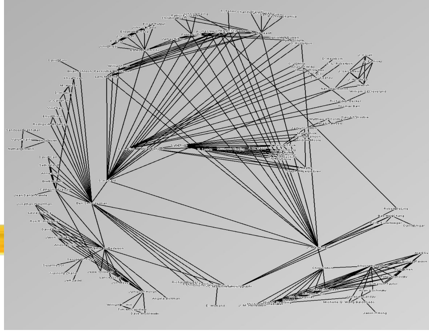
References:
Mohammad Ghoniem, Jean-Daniel Fekete and Philippe Castagliola *Readability of Graphs Using Node-Link and Matrix-Based Representations: Controlled Experiment and Statistical Analysis*, Information Visualization Journal, 4(2), Palgrave Macmillan, Summer 2005, pp. 114-135.



Completion time for the 7 tasks, 3 densities and 2 representations (Node-Link in blue, Matrix in red)



Matrix vs. NodeLink



+

- Usable without reordering
- No node overlapping
  No edge crossing
  → Readable for dense graphs
- Fast navigation
- Fast manipulation
  → Usable interactively

- Familiar
- Compact
- More readable for path following
- More effective for small graphs
- More effective for sparse graphs

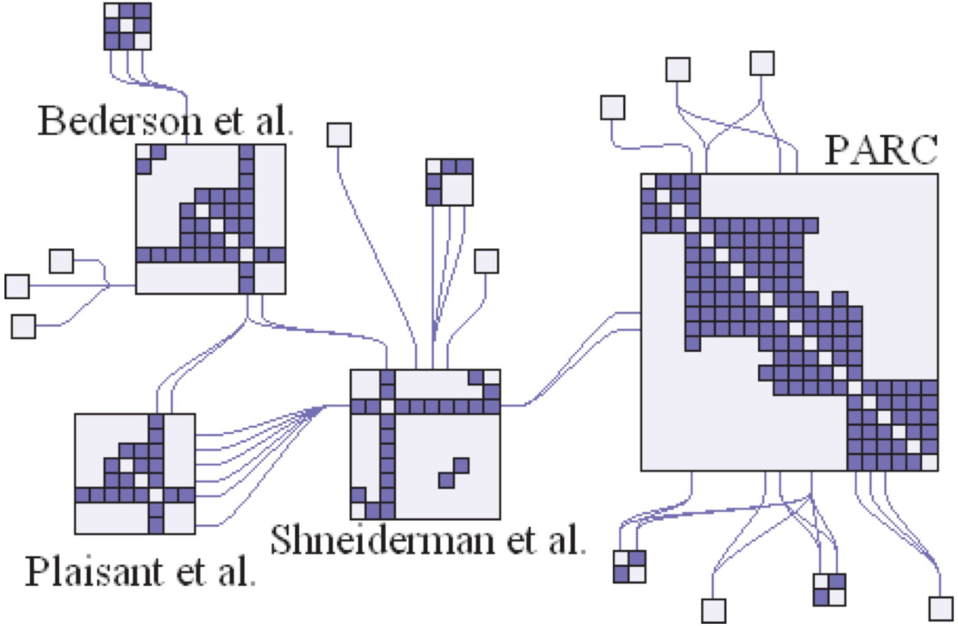
–

- More readable for some tasks
- Less familiar
- Use more space
- Weak for path following tasks

- Useless without layout
- Node overlapping
  Edge crossing
  → Not readable for dense graphs
- Manipulation requires layout

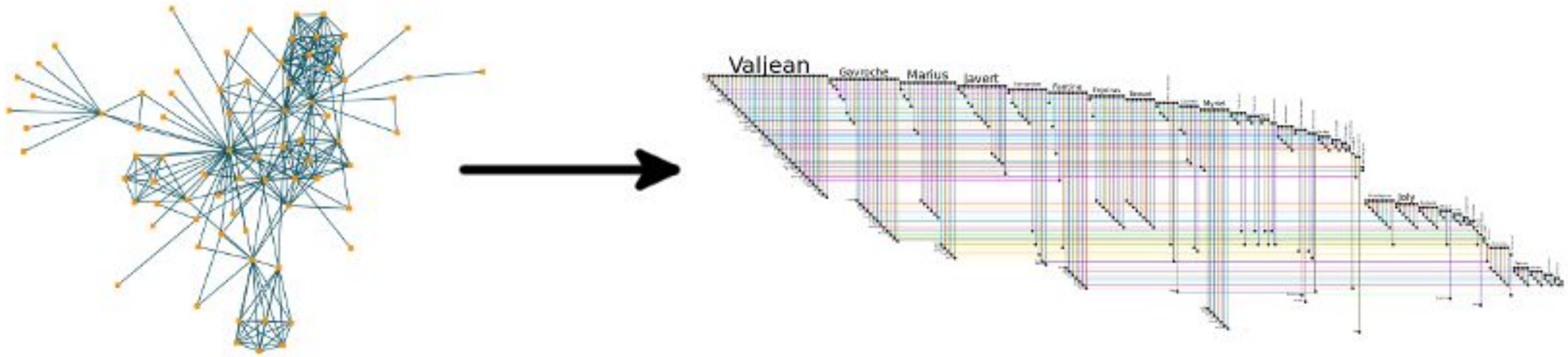
Idiom: **NodeTrix**

- hybrid nodelink/matrix
- capture strengths of both



[NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007.]

More Graph Drawing: Biofabric



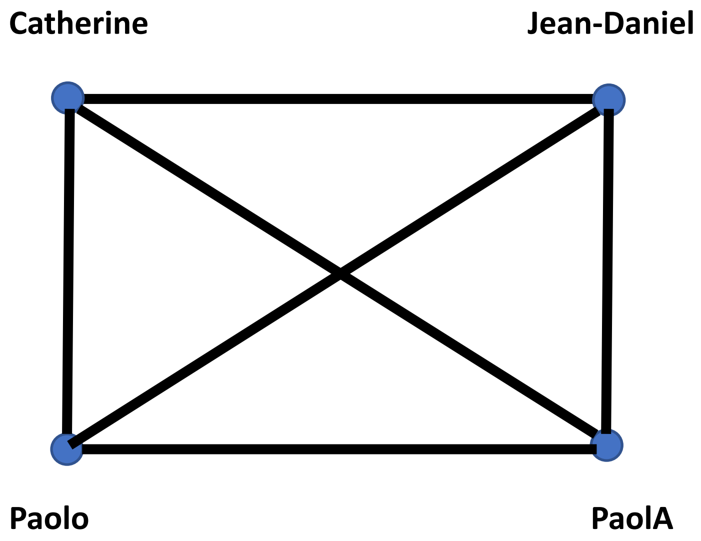
https://biofabric.systemsbiology.net/gallery/pages/SuperQuickBioFabric.html

Hypergraph?

= a generalization of a graph

where EDGES can connect
one, two, or more VERTICES

Graph (e.g. who knows who)



Catherine — Jean-Daniel

Catherine — Paolo

Catherine — PaolA

Paolo — PaolA

Paolo — Jean-Daniel

PaolA — Jean-Daniel

Node link diagram

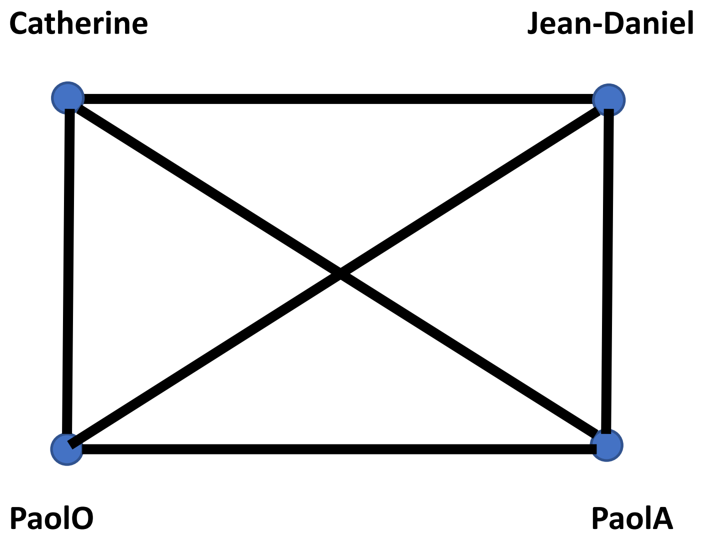
Hypergraph (e.g. teams)

Catherine Jean-Daniel

Catherine Paolo

Paolo PaolA Jean-Daniel

Paolo PaolA Jean-Daniel Catherine



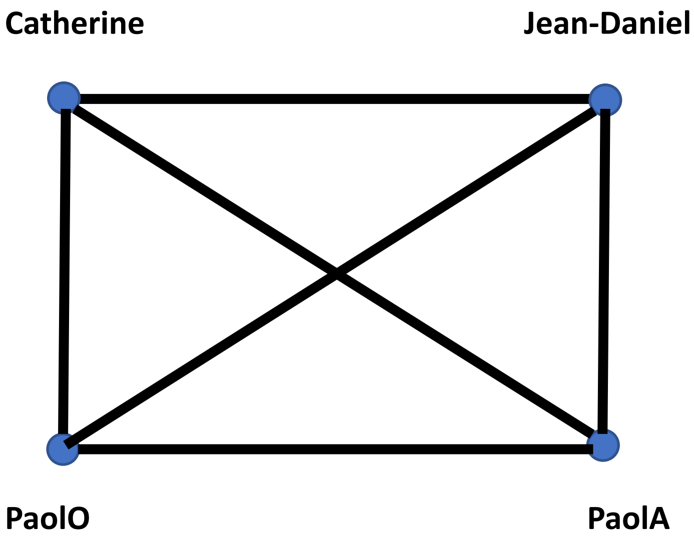
Catherine Jean-Daniel

Catherine Paolo

Paolo PaolA Jean-Daniel

Paolo PaolA Jean-Daniel Catherine

Node link diagram



Catherine Jean-Daniel

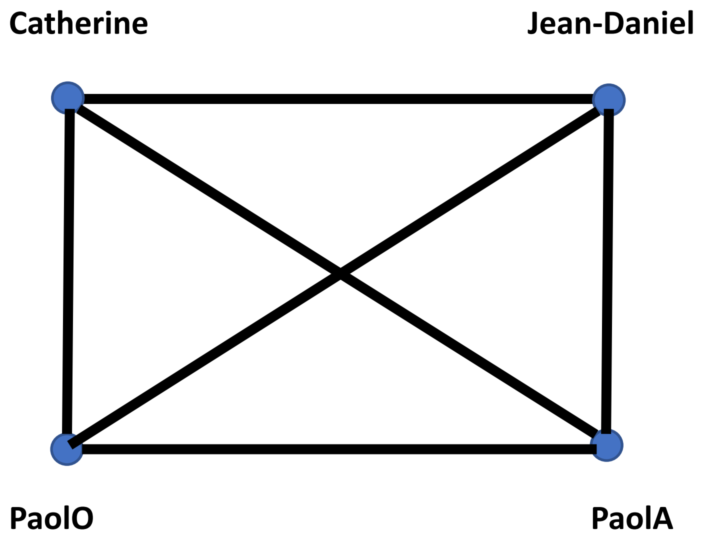
Catherine Paolo

Paolo PaolA Jean-Daniel

Paolo PaolA Jean-Daniel Catherine

4 teams / 4 hyperedges

Looks just the same as…



Catherine — Jean-Daniel

Catherine — Paolo

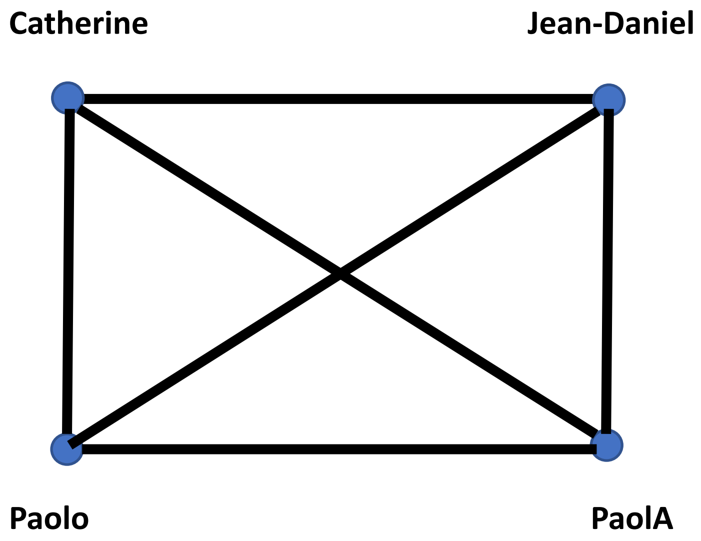
Catherine — PaolA

Paolo — PaolA

Paolo — Jean-Daniel

PaolA — Jean-Daniel

Looks just the same as… **the “who knows who” graph**



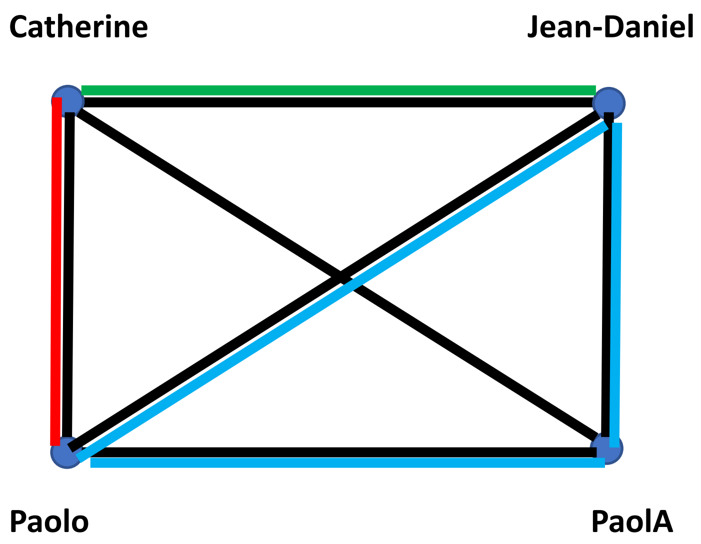
Catherine Jean-Daniel

Catherine Paolo

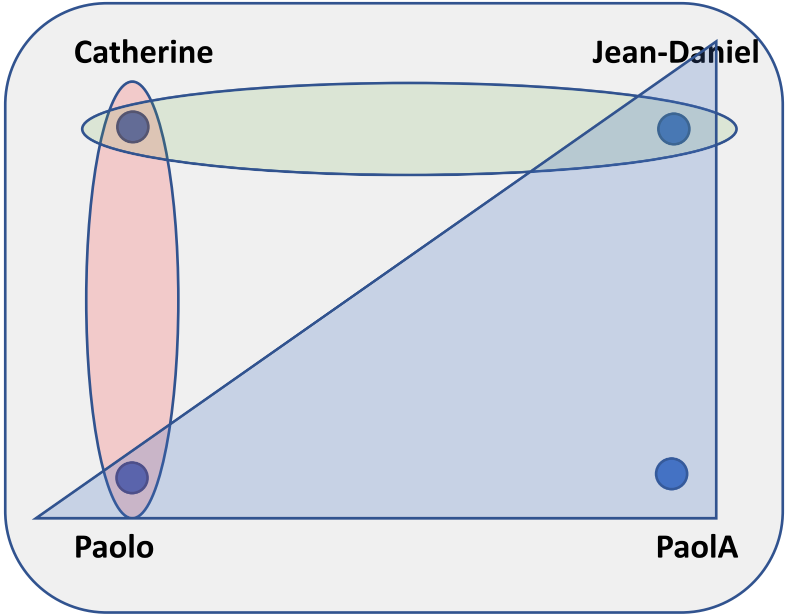
Paolo PaolA Jean-Daniel

Paolo PaolA Jean-Daniel Catherine

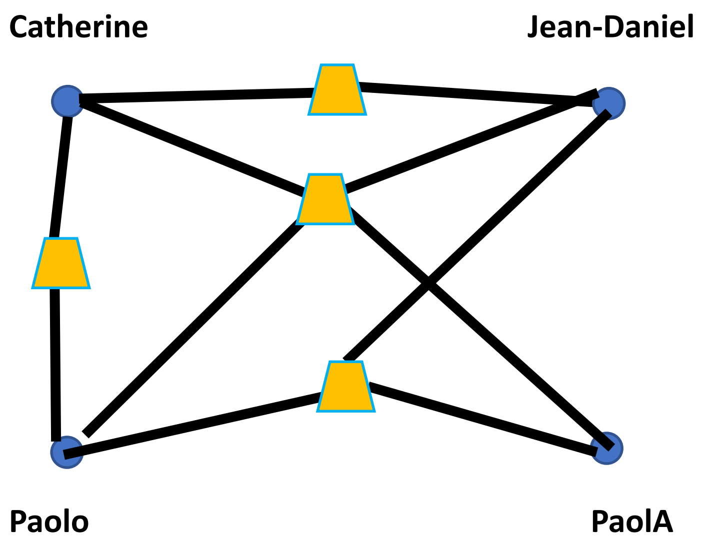
?? How to reveal the hyperedges ??



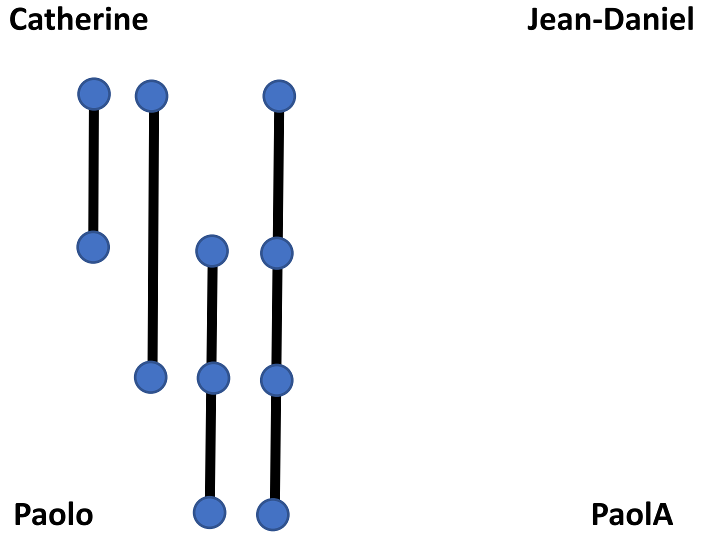
Multiple links – different link attributes



Teams = Sets



Bi-partite network (nodes and teams)



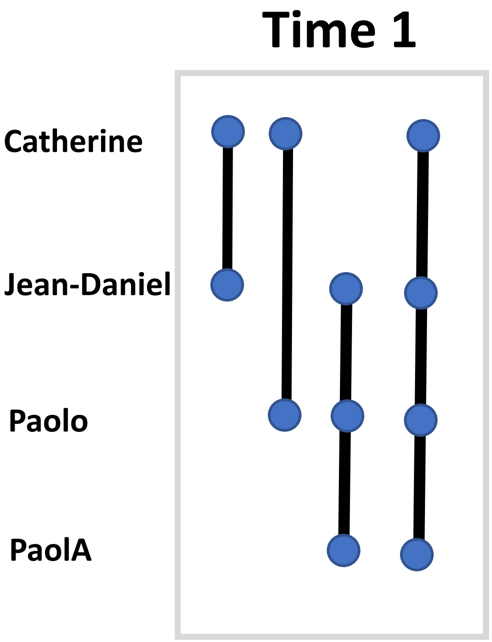
Catherine Jean-Daniel

Catherine Paolo

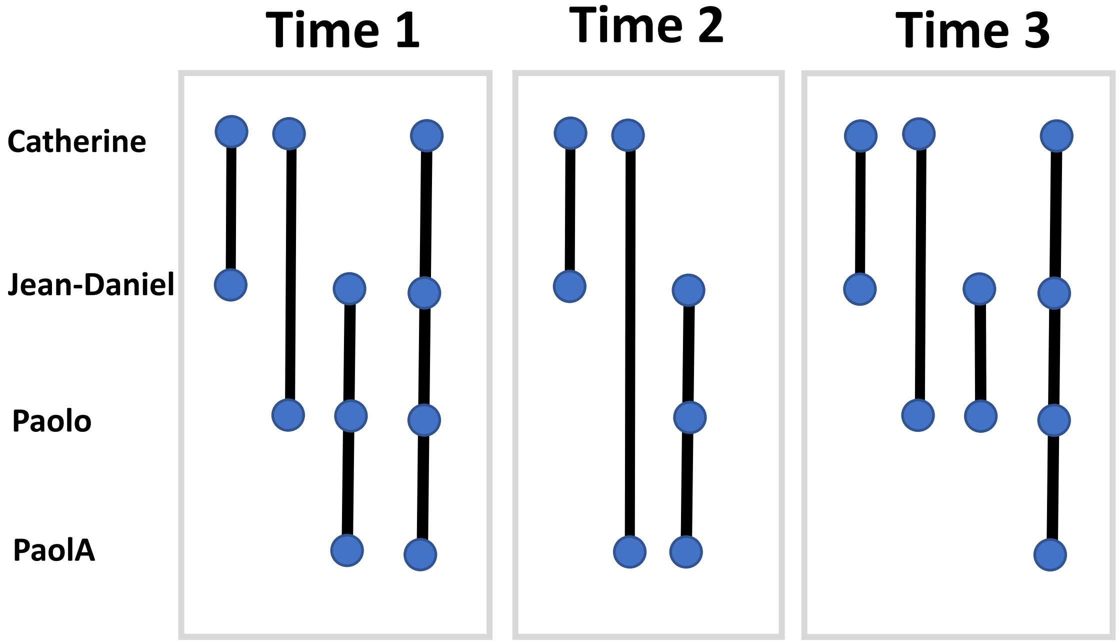
Paolo PaolA Jean-Daniel

Paolo PaolA Jean-Daniel Catherine

Our proposed technique: Paola R Valdivia, Paolo Buono, Catherine Plaisant, Nicole Dufournaud, Jean-Daniel Fekete. Analyzing Dynamic Hypergraphs with Parallel Aggregated Ordered Hypergraph Visualization. IEEE Transactions on Visualization and Computer Graphics, 2021, 27 (1), pp.1-13. ⟨10.1109/TVCG.2019.2933196⟩. ⟨[hal-02264960](hal-02264960)⟩ https://aviz.fr/paohvis

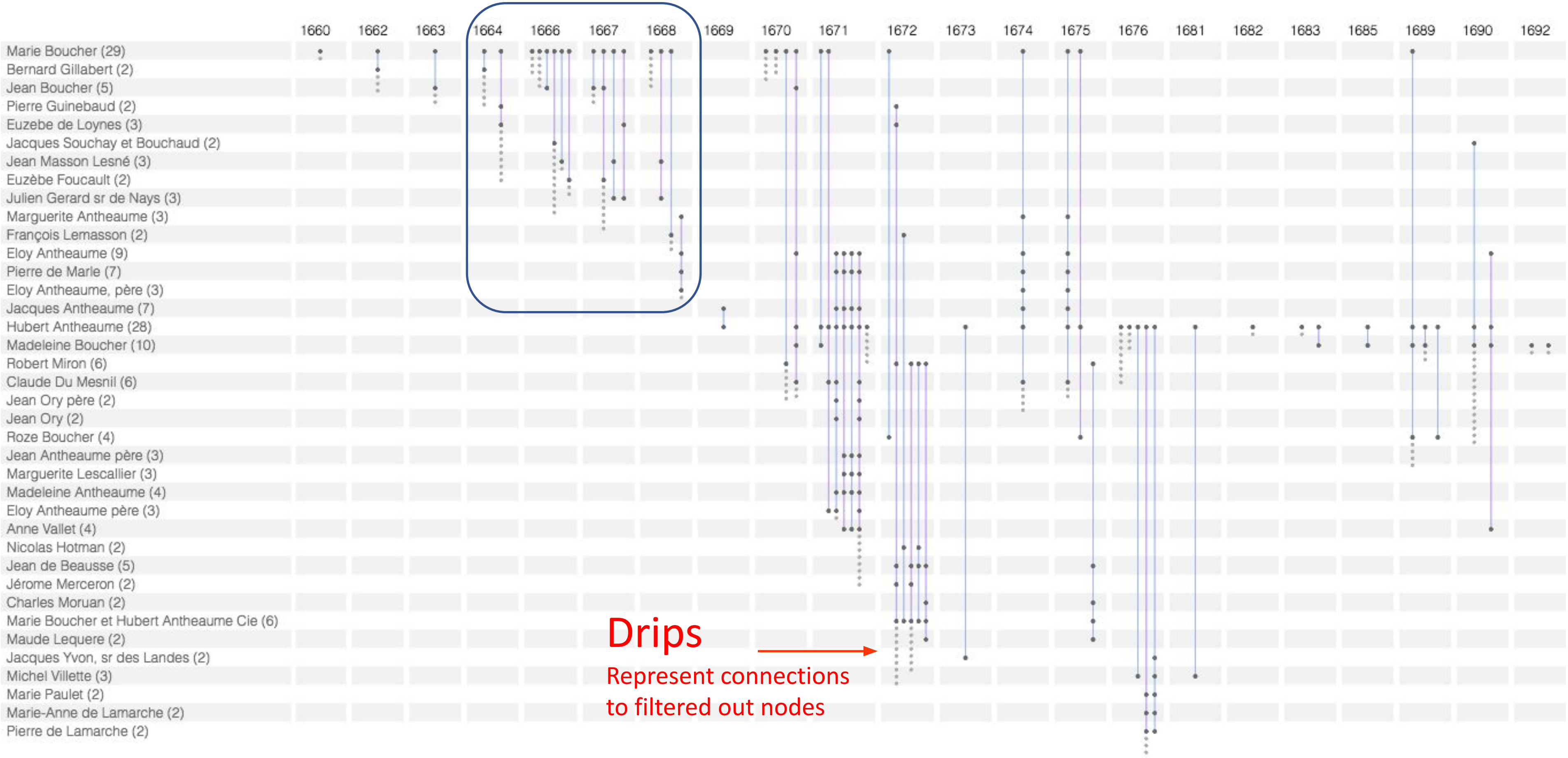


Hypergraph



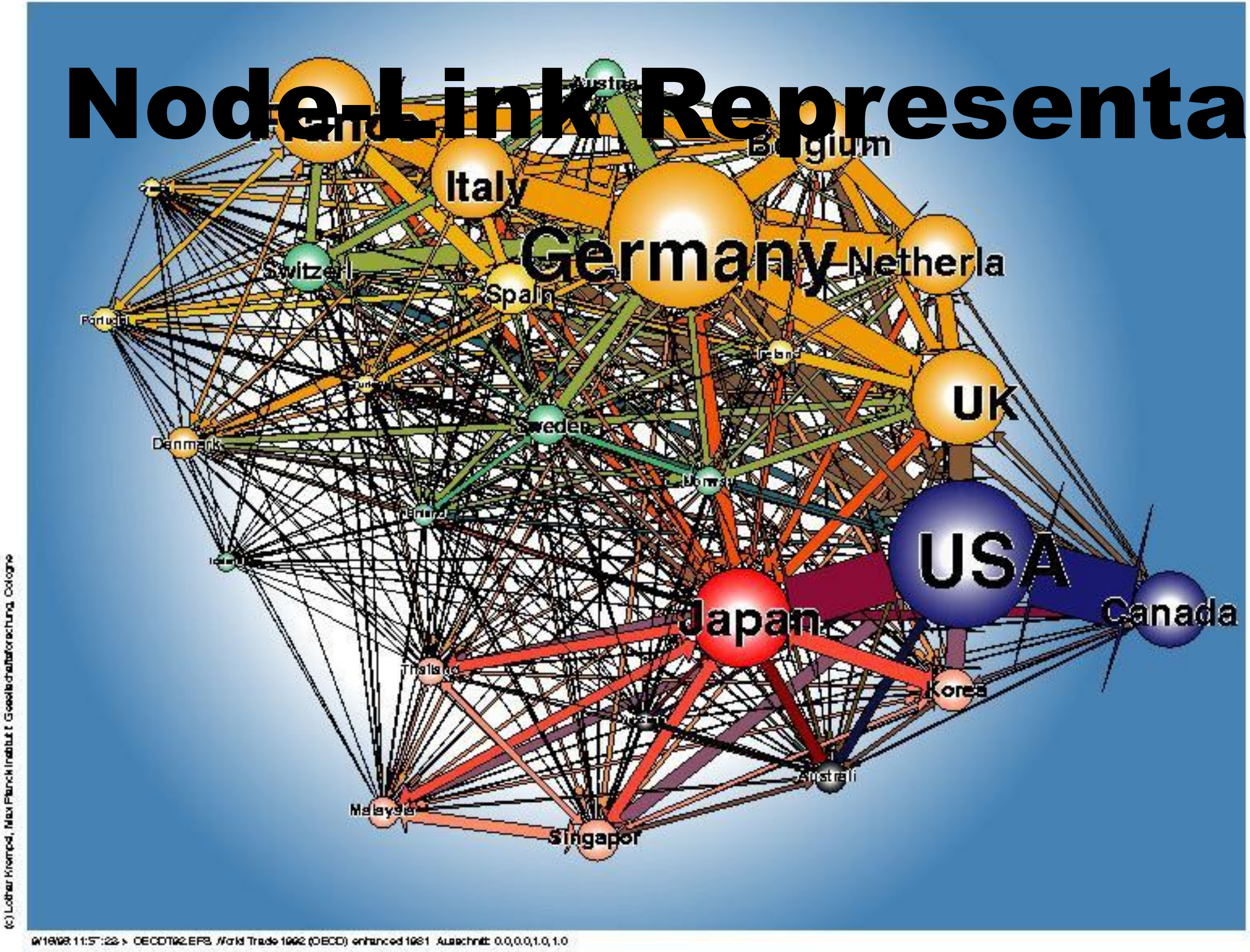
Dynamic Hypergraph (changes overtime)

Parallel Aggregated Ordered Hypergraph
PAOHvis



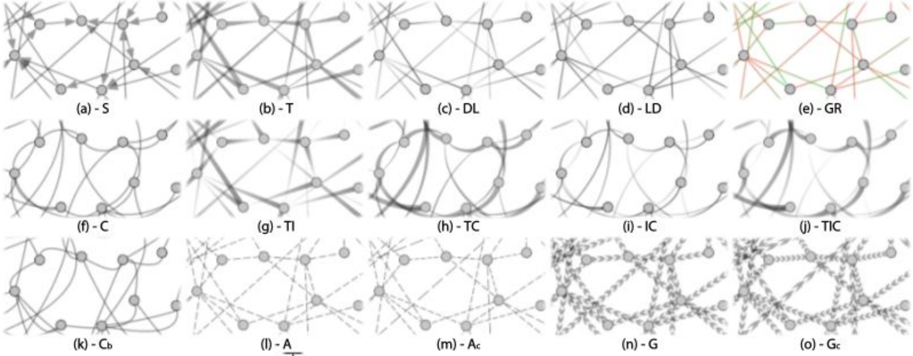
Marie Boucher. 63 documents. 75 people - 48 listed (appear in > 1 document) 28 hinted as "drips"

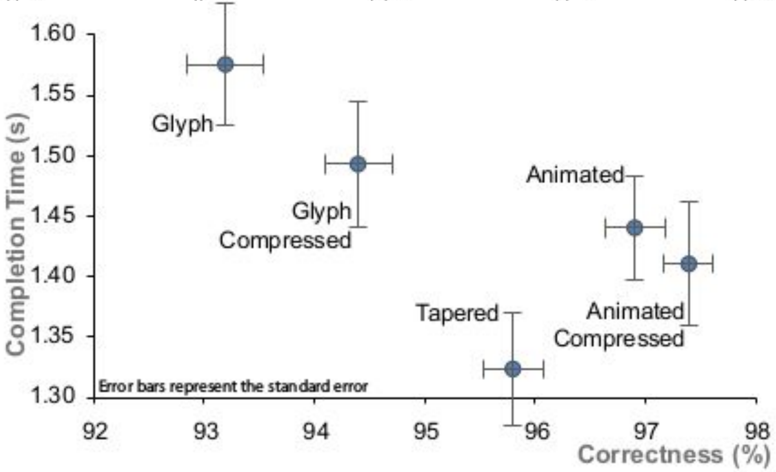
Node-Link Representation



Lisibilité des liens ?

- 3 codages
- 3 densités
- Plusieur graphes aléatoires
- On pose des questions
- On mesure
  - Le temps de réponse
  - La validité des réponses
- On regarde s'il y a une différence statistique
- Oui !

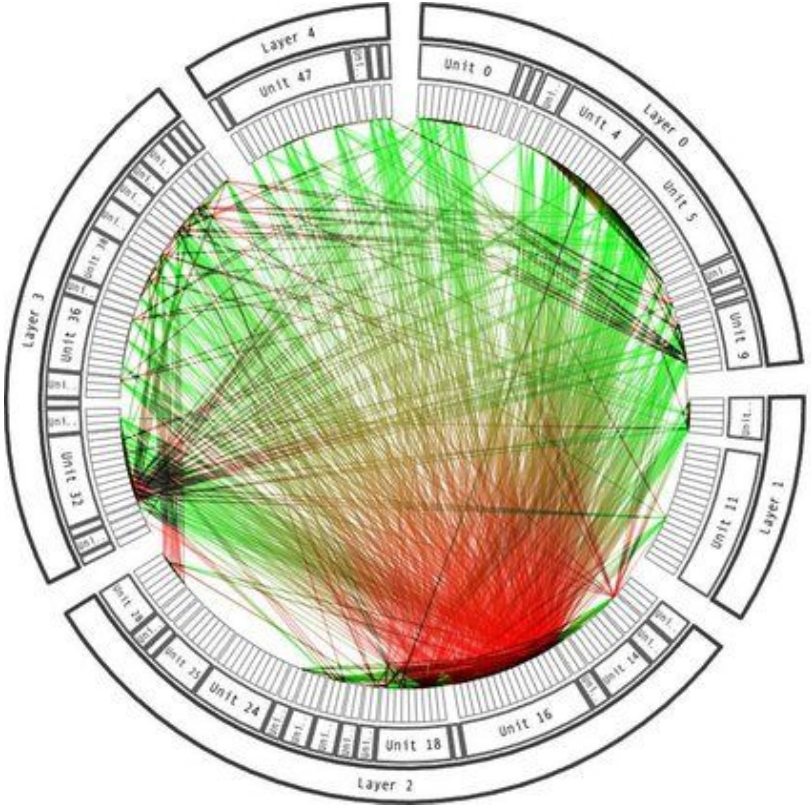


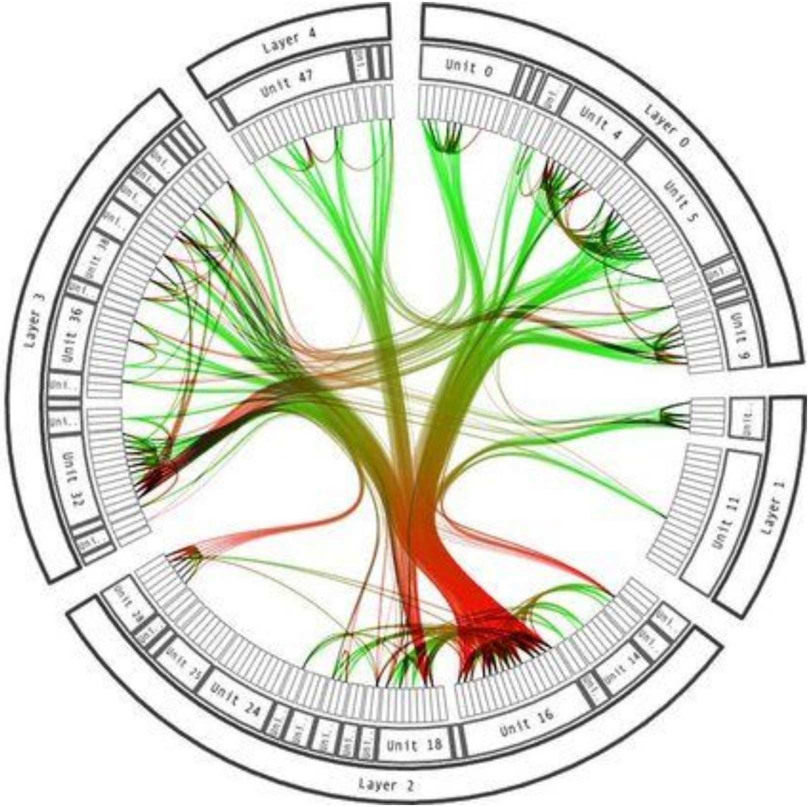


D. Holten and J. J. van Wijk. A User Study on Visualizing Directed Edges in Graphs. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), pages 2299–2308. ACM, 2009.

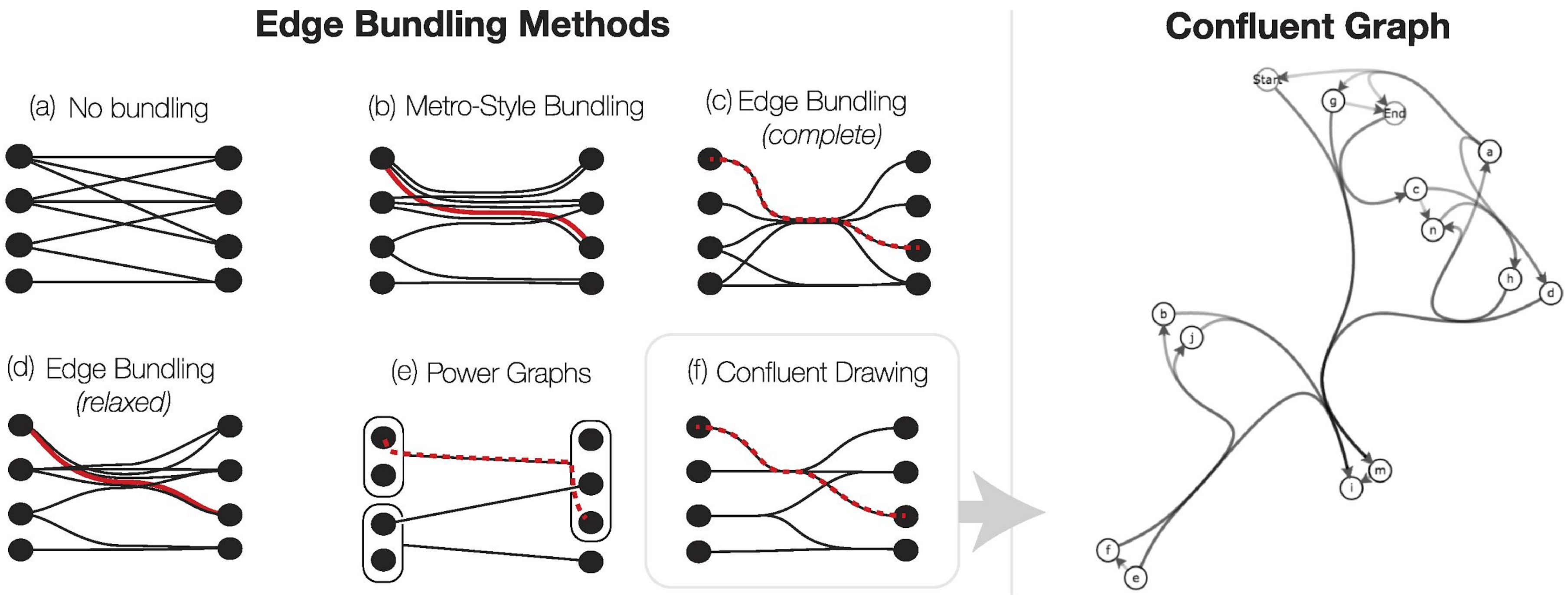
Danny Holten, Petra Isenberg, Jarke Van Wijk, Jean-Daniel Fekete. An Extended Evaluation of the Readability of Tapered, Animated, and Textured Directed-Edge Representations in Node-Link Graphs. IEEE Press. *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, Mar 2011, Hong Kong, China. 8p, 2011.

Edge bundling





Edge Bundling ou pas ?



https://aviz.fr/~bbach/confluentgraphs/

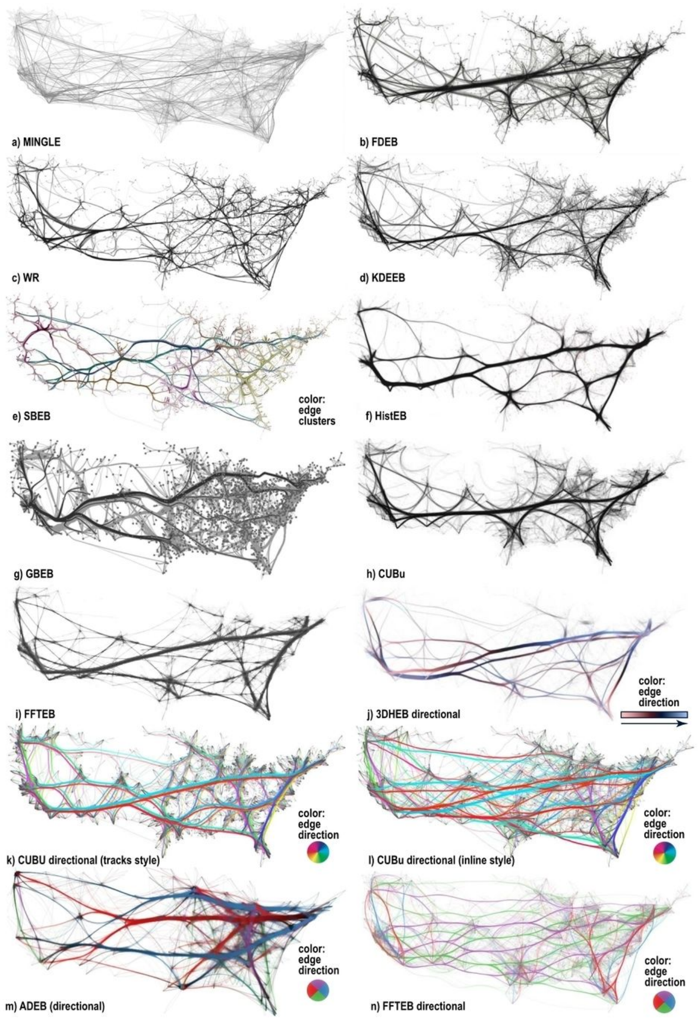
Edge Bundling ou pas ?

Les algorithmes de groupement des liens simplifient le dessin de réseaux

Mais ils produisent des résultats difficiles à contrôler

- Si l'objectif est de rendre le réseau plus beau, OK
- Si les trajectoires doivent avoir un sens, alors il faut les éviter
- Simplifier retire de l'information...

A. Lhuillier, C. Hurter, A. Telea.
State of the Art in Edge and Trail Bundling Techniques, in *Eurovis 2017* June. 2017

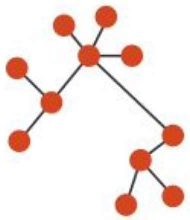


Trees

Arrange Networks and Trees

Node–Link Diagrams
Connection Marks

✔ NETWORKS ✔ TREES



Adjacency Matrix
Derived Table

✔ NETWORKS ✔ TREES

Enclosure
Containment Marks

✘ NETWORKS ✔ TREES

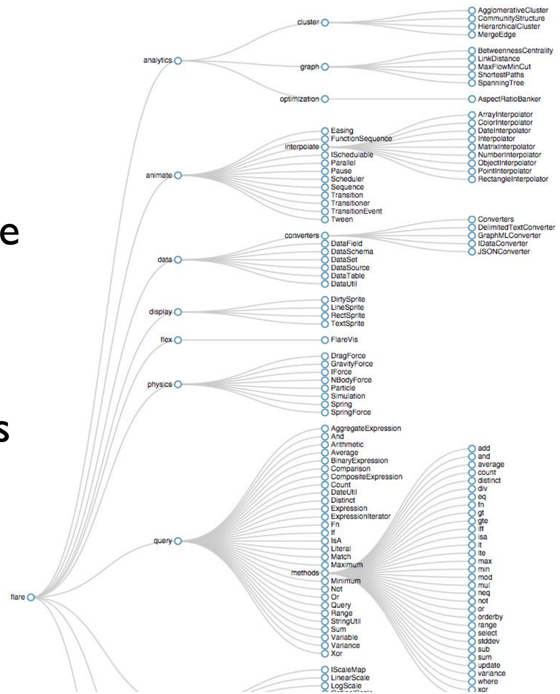
Node-link trees

- Reingold-Tilford
  - tidy drawings of trees
    - exploit parent/child structure
  - allocate space: compact but without overlap
    - rectilinear and radial variants

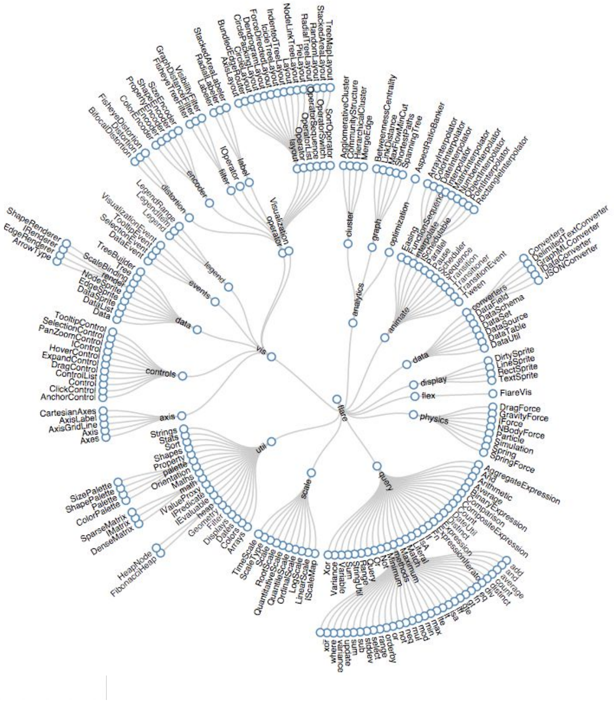
[Tidier drawing of trees. Reingold and Tilford. IEEE Trans. Software Eng., SE-7(2):223–228, 1981.]

  - nice algorithm writeup

http://billmill.org/pymag-trees/



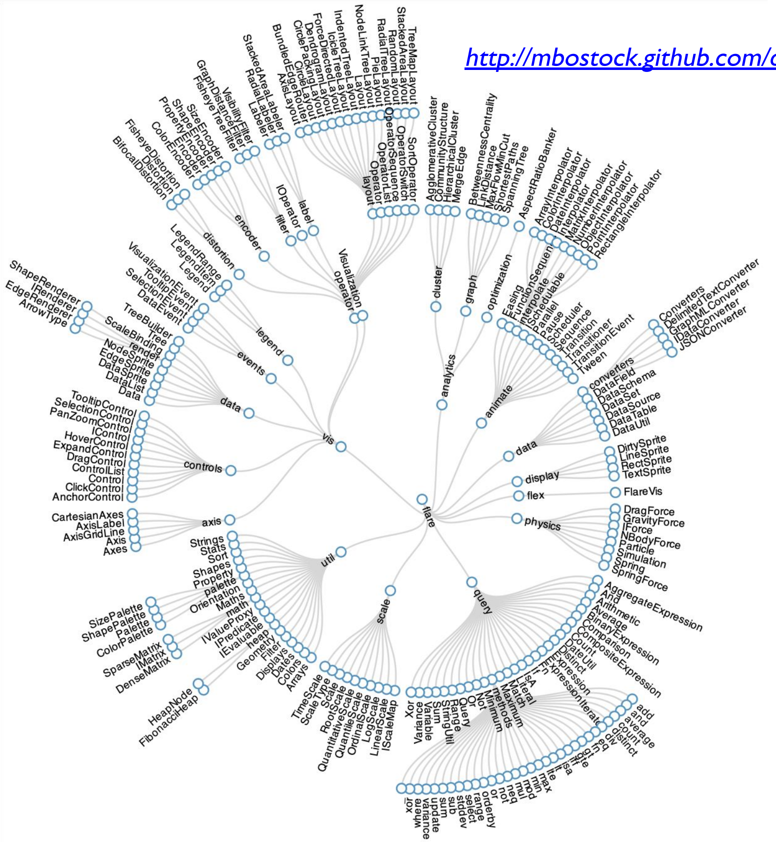
http://bl.ocks.org/mbostock/4339184



http://bl.ocks.org/mbostock/4063550

Idiom: **radial node-link tree**

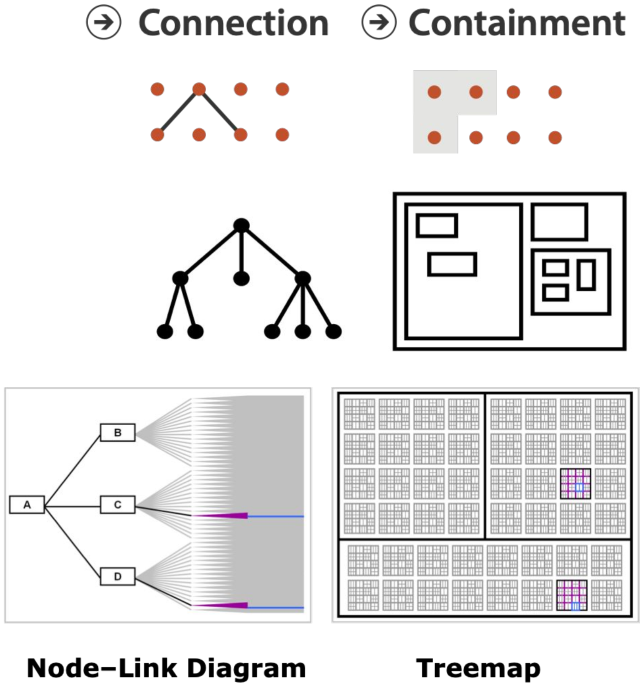
- data
  - tree
- encoding
  - link connection marks
  - point node marks
  - radial axis orientation
    - angular proximity: siblings
    - distance from center: depth in tree
- tasks
  - understanding topology, following paths
- scalability
  - 1K - 10K nodes (with/without labels)



http://mbostock.github.com/d3/ex/tree.h

Link marks: Connection and containment

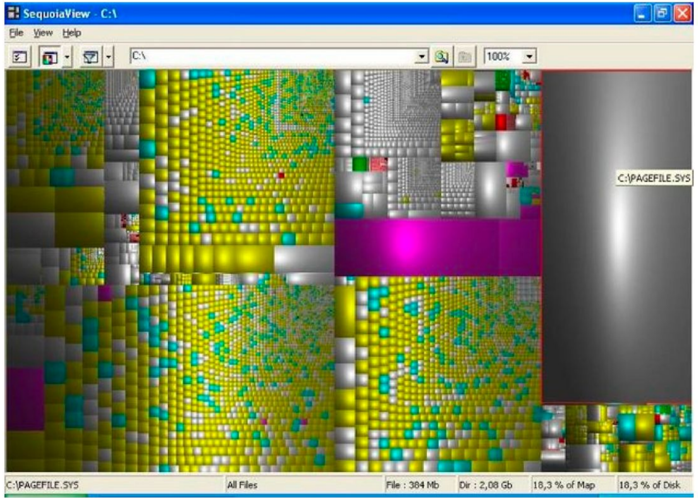
- marks as links (vs. nodes)
  - common case in network drawing
  - 1D case: connection
    - ex: all node-link diagrams
    - emphasizes topology, path tracing
    - networks and trees
  - 2D case: containment
    - ex: all treemap variants
    - emphasizes attribute values at leaves (size coding)
    - only trees



[Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. Dong, McGuffin, and Chignell. Proc. InfoVis 2005, p. 57-64.]

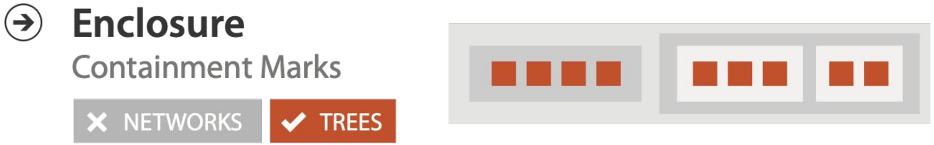
Idiom: **treemap**

- data
  - tree
  - 1 quant attrib at leaf nodes
- encoding
  - area containment marks for hierarchical structure
  - rectilinear orientation
  - size encodes quant attrib
- tasks
  - query attribute at leaf nodes
  - ex: disk space usage within filesystem
- scalability
  - 1M leaf nodes



https://www.win.tue.nl/sequoiaview/

[Cushion Treemaps. van Wijk and van de Wetering. Proc. Symp. InfoVis 1999, 73-78.]



Idiom: implicit tree layouts (sunburst, icicle plot)

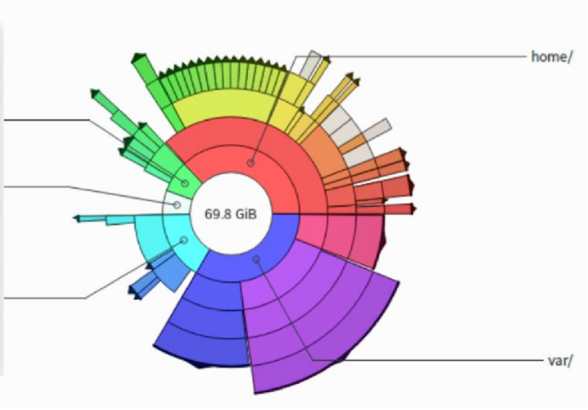
- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

Treemap

containment

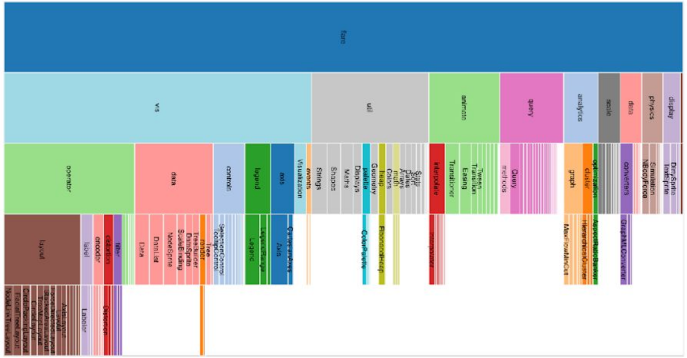
Sunburst

position (radial)



Icicle Plot

position (rectilinear)



Idiom: implicit tree layouts (sunburst, icicle plot)

- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

Treemap

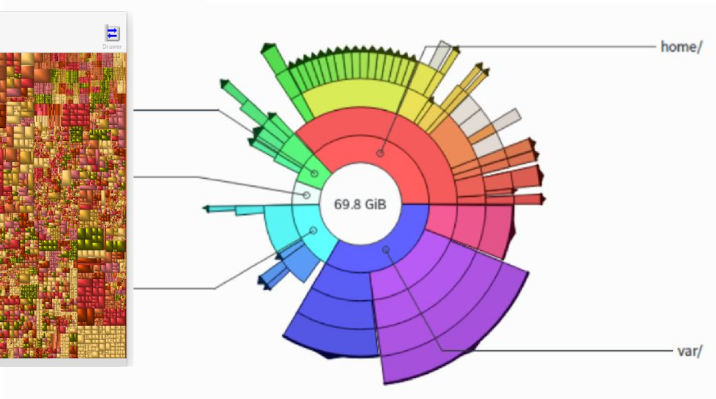
containment

only leaves visible

Sunburst

position (radial)

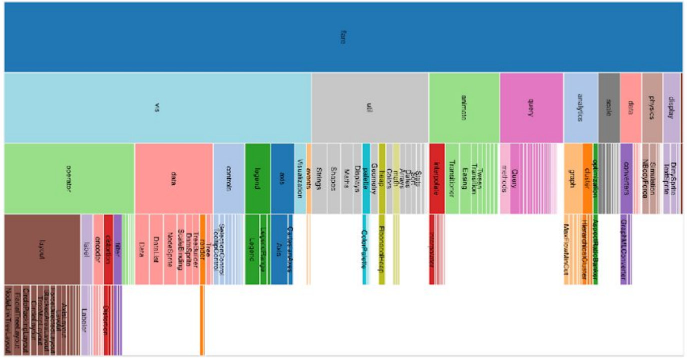
inner nodes & leaves visible



Icicle Plot

position (rectilinear)

inner nodes & leaves visible



Idiom: implicit tree layouts (sunburst, icicle plot)

- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

Treemap

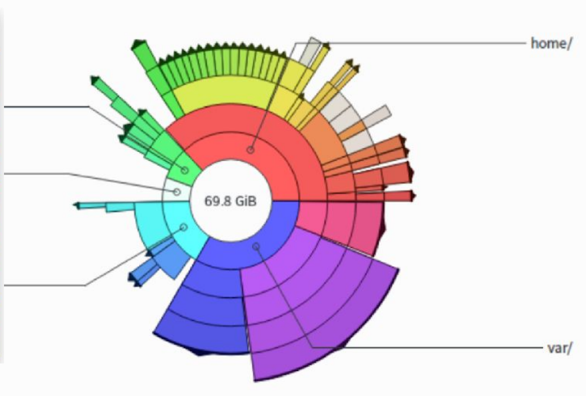
containment

only leaves visible

Sunburst

position (radial)

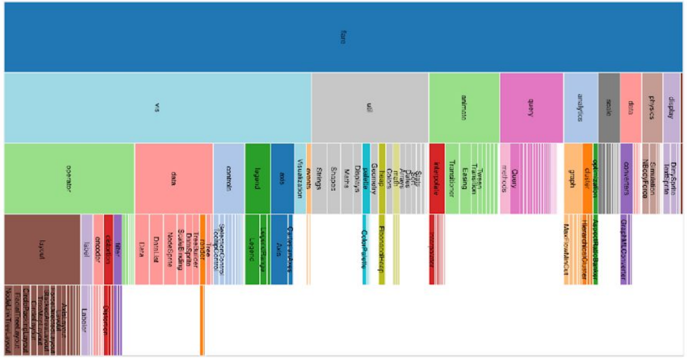
inner nodes & leaves visible



Icicle Plot

position (rectilinear)

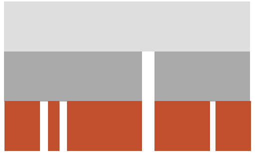
inner nodes & leaves visible



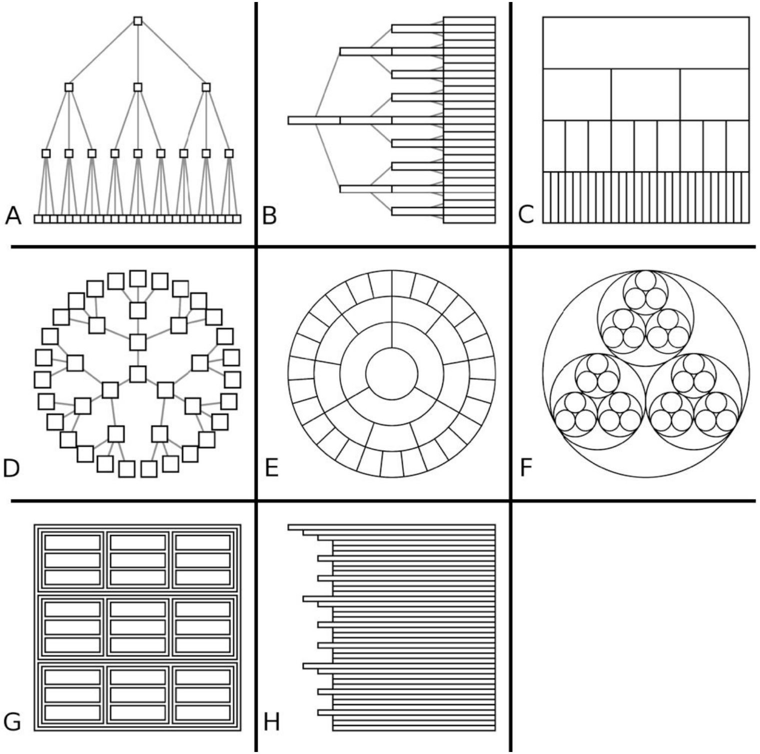
➔ Implicit

Spatial Position

✕ NETWORKS ✔ TREES



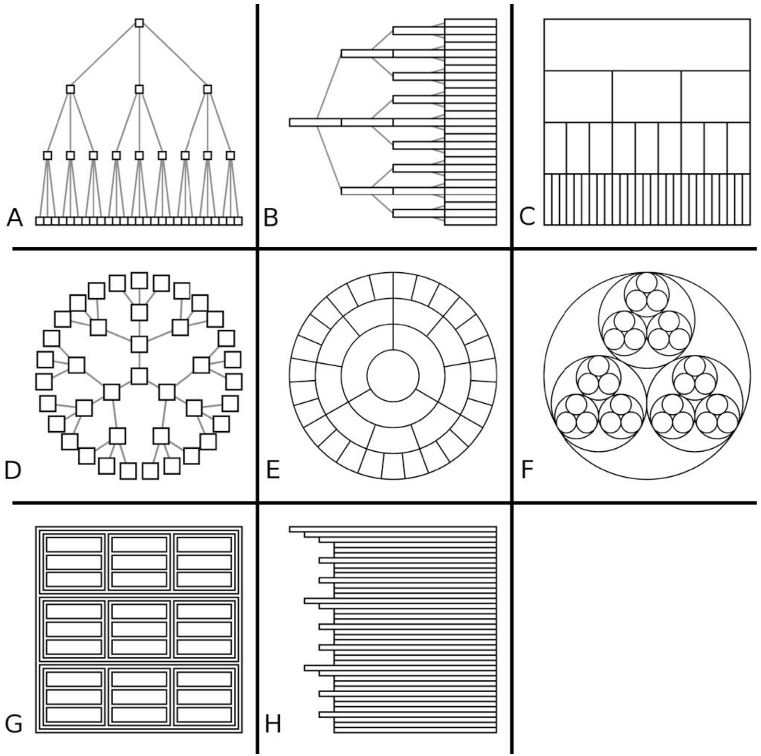
Tree drawing idioms comparison



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

Comparison: tree drawing idioms

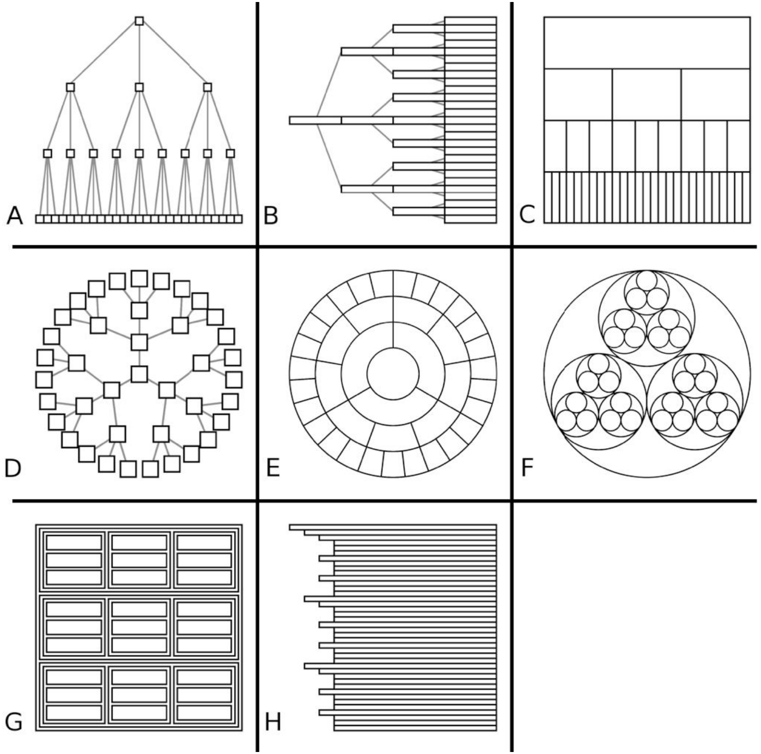
- data shown
  - link relationships
  - tree depth
  - sibling order



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees.
McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

Comparison: tree drawing idioms

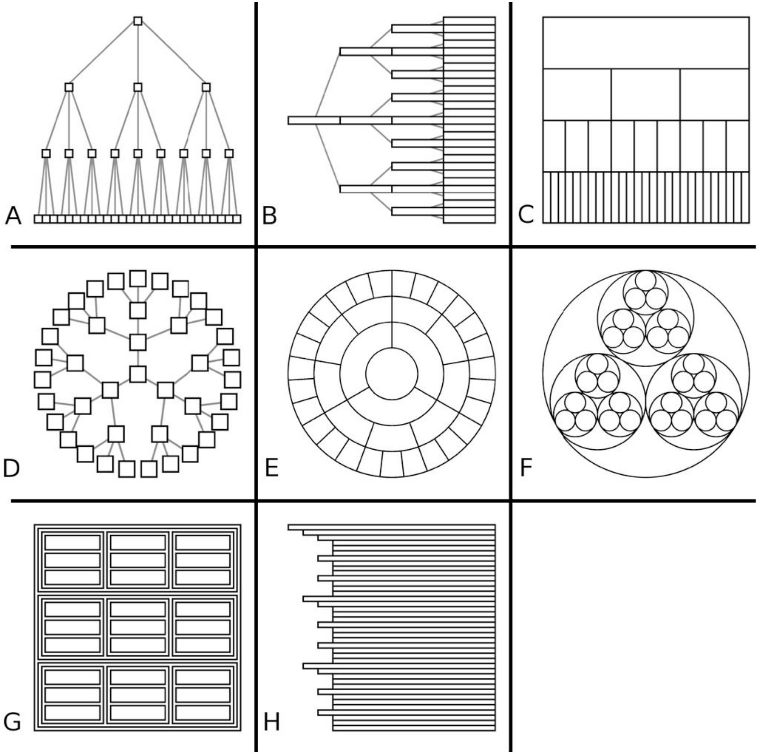
- data shown
  - link relationships
  - tree depth
  - sibling order
- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees.
McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

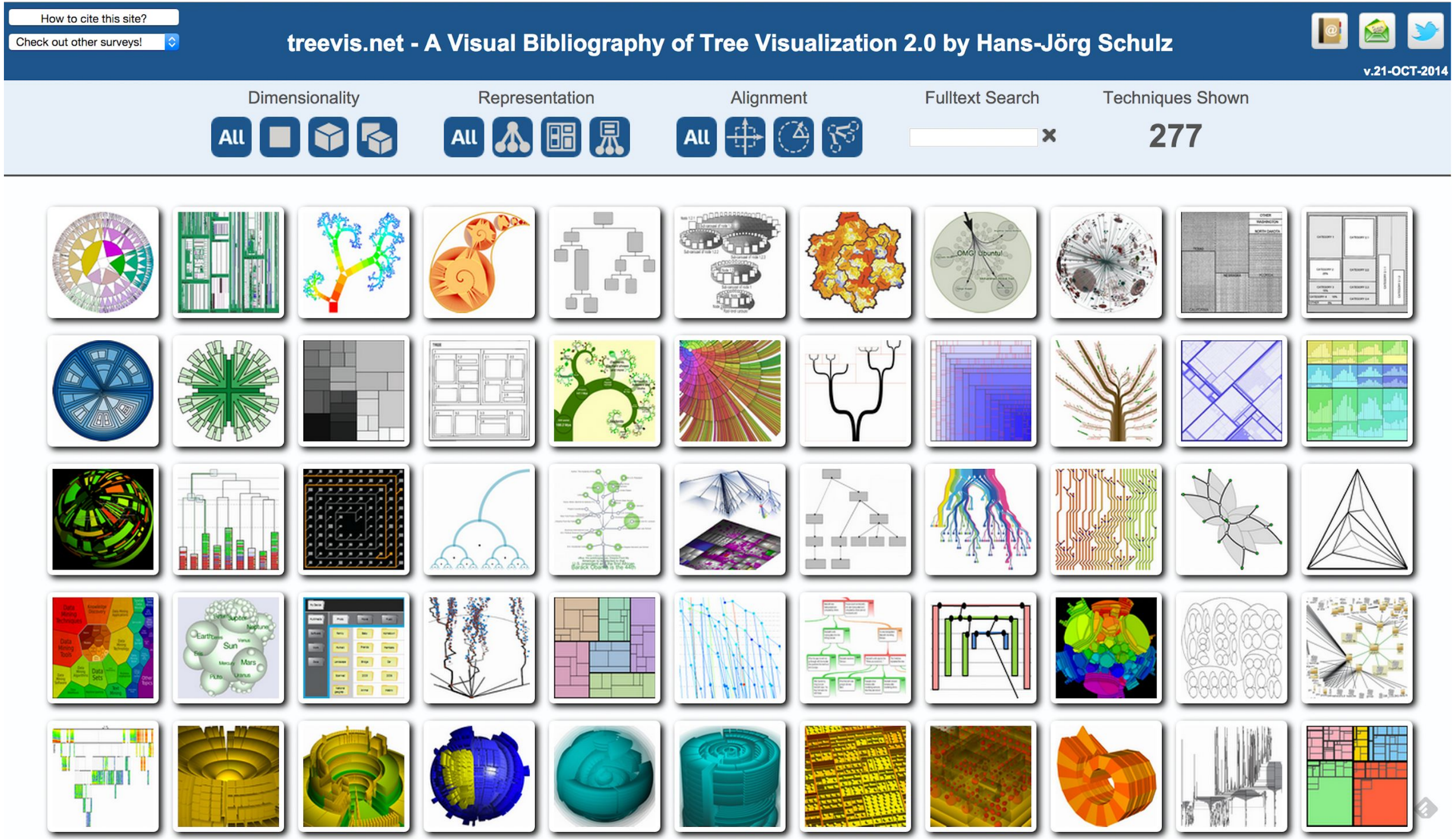
Comparison: tree drawing idioms

- data shown
  - link relationships
  - tree depth
  - sibling order
- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels
- considerations
  - redundant? arbitrary?
  - information density?
    - avoid wasting space
    - consider where to fit labels!

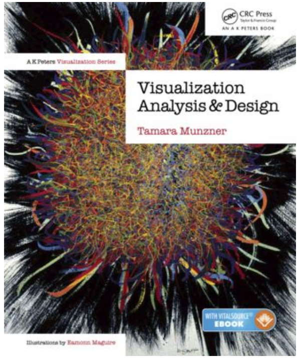


[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

treevis.net: Many, many options!



https://treevis.net/



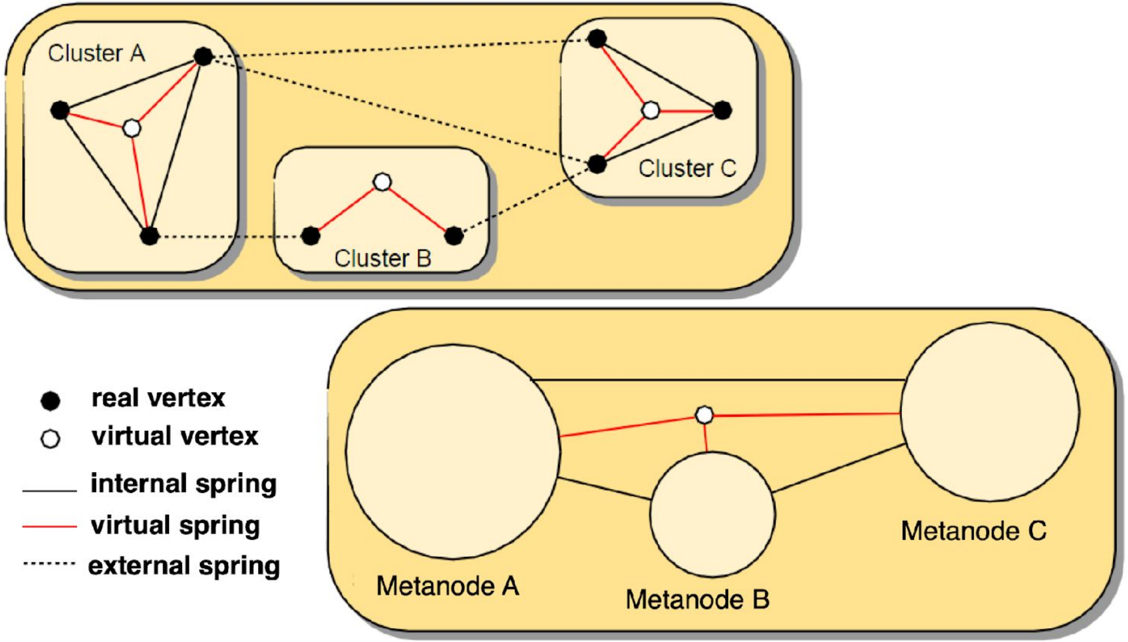
Visualization Analysis & Design

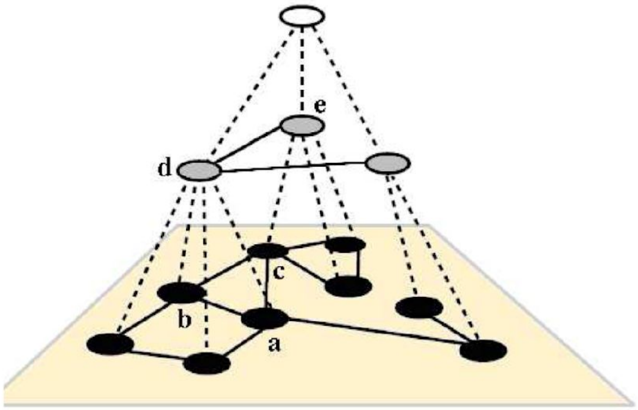
Network Data (Ch 9) II

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

Multilevel networks

- derive cluster hierarchy of metanodes on top of original graph nodes

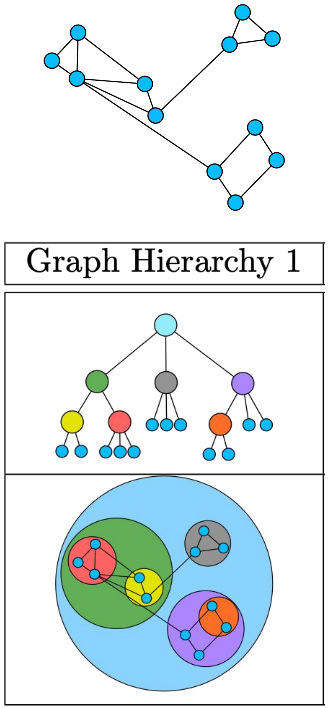




[Schulz 2004]

Idiom: **GrouseFlocks**

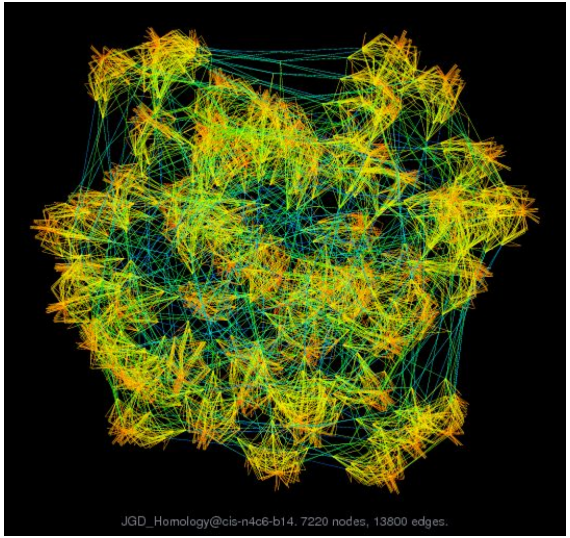
- data: compound network
  - network
  - cluster hierarchy atop it
    - derived or interactively chosen
- visual encoding
  - connection marks for network links
  - containment marks for hierarchy
  - point marks for nodes
- dynamic interaction
  - select individual metanodes in hierarchy to expand/contract



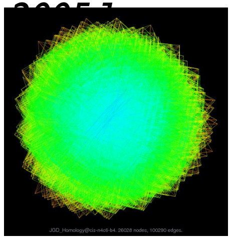
[GrouseFlocks: Steerable Exploration of Graph Hierarchy Space. Archambault, Munzner, and Auber. IEEE TVCG 14(4):900-913, 2008.]

Idiom: **sfdp** (multi-level force-directed placement)

- data: compound graph
  - original: network
  - derived: cluster hierarchy atop it
- considerations
  - better algorithm for same encoding technique
    - same: fundamental use of space
    - hierarchy used for algorithm speed/quality but not shown explicitly
- scalability
  - nodes, edges: 1K-10K
  - hairball problem eventually hits



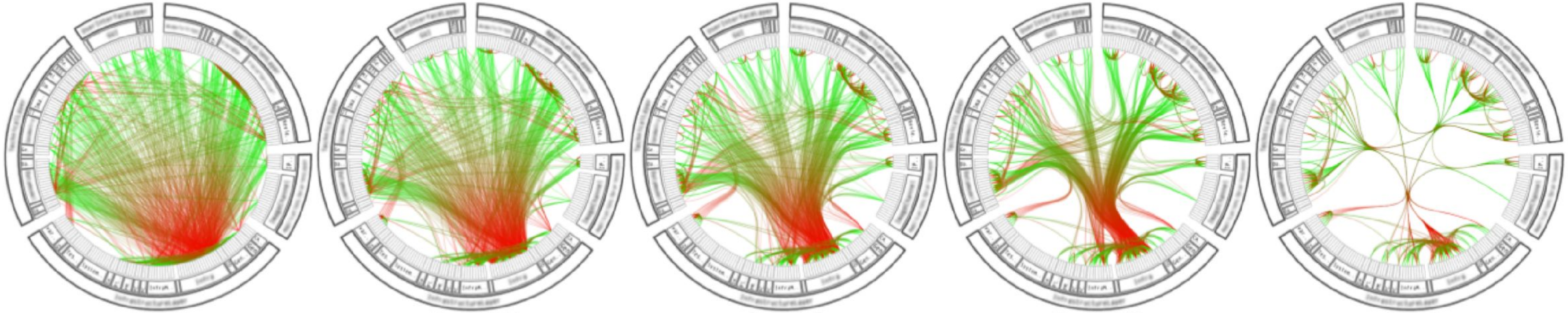
[Efficient and high quality force-directed graph drawing. Hu. The Mathematica Journal 10:37–71, 2005.]



http://www.research.att.com/yifanhu/GALLERY/GRAPHS/index1.html

Idiom: **hierarchical edge bundling**

- data
  - any layout of compound network
    - network: software classes (nodes), import/export between classes (links)
    - cluster hierarchy: class package structure
  - derived: bundles of edges with same source/destination (multi-level)
- idiom: curve edge routes according to bundles
- task: edge clutter reduction

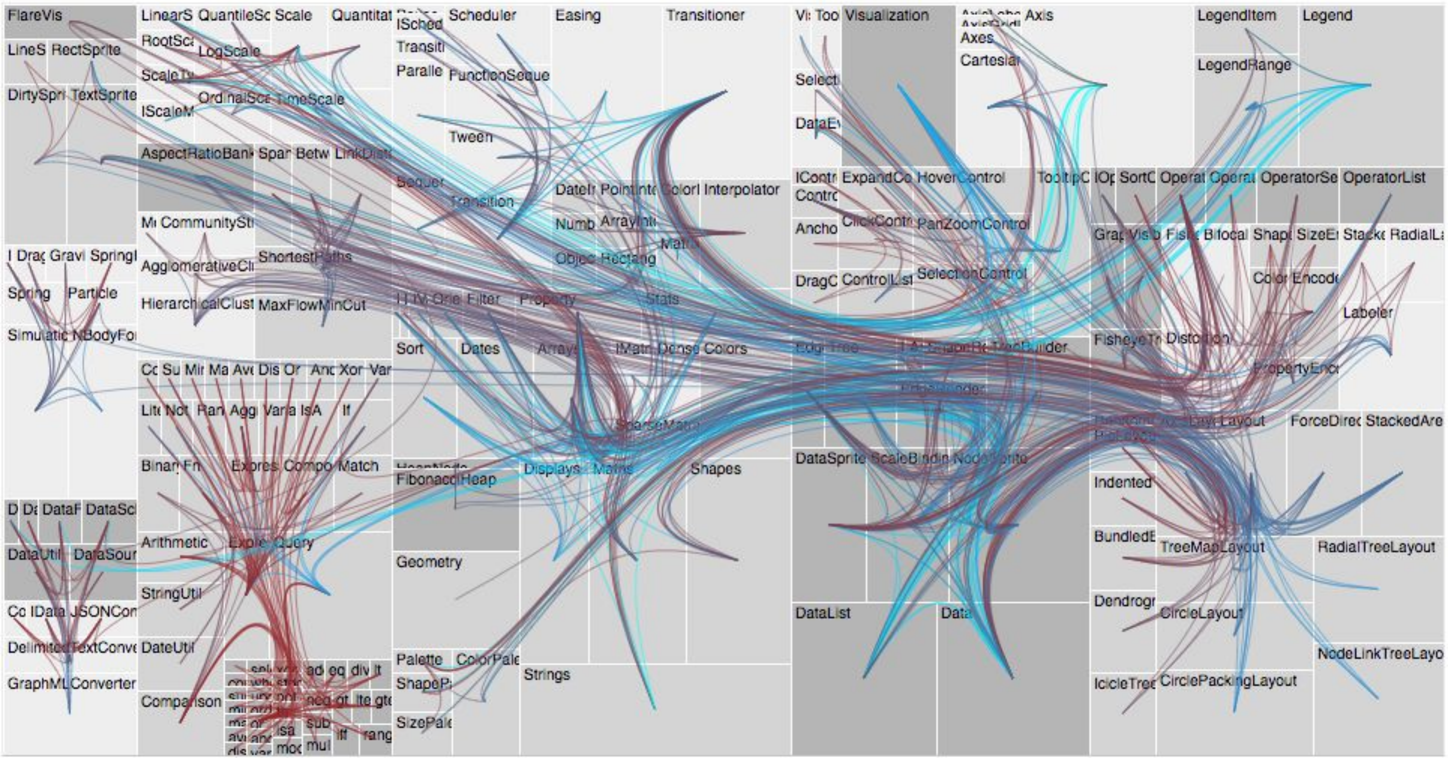


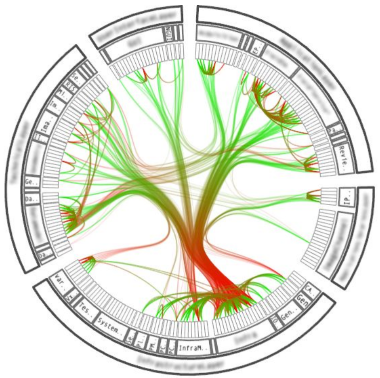
$\beta = 0$ $\beta = 0.25$ $\beta = 0.5$ $\beta = 0.75$ $\beta = 1$

[Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. Danny Holten. TVCG 12(5):741-748 2006]

Hierarchical edge bundling

- works for any layout: treemap vs radial





[Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. Danny Holten. TVCG 12(5):741-748 2006]

Free Software

- Java
  - Cytoscape https://cytoscape.org/, Gephi https://gephi.org/
- C/C++/Python
  - GraphViz https://www.graphviz.org/, igraph https://igraph.org/, NetworkX https://networkx.org/
  - Tulip https://tulip.labri.fr/site/
- Web
  - **D3 https://d3js.org/**
  - **Vega https://vega.github.io/vega/examples/**
  - **The Vistorian https://vistorian.net/**
  - **https://demo.hedgedoc.org/features**
  - **WebCola https://ialab.it.monash.edu/webcola/**