NYU

CS-GY 6313 B: Information Visualization

10/10/2024

Logistics

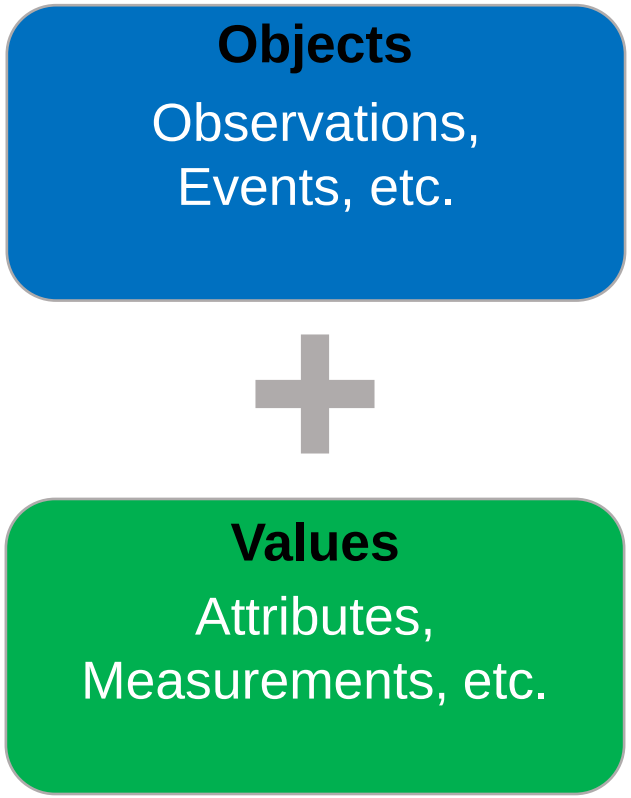
- Assignment 2 due next week

2D Viz: Networks and Trees

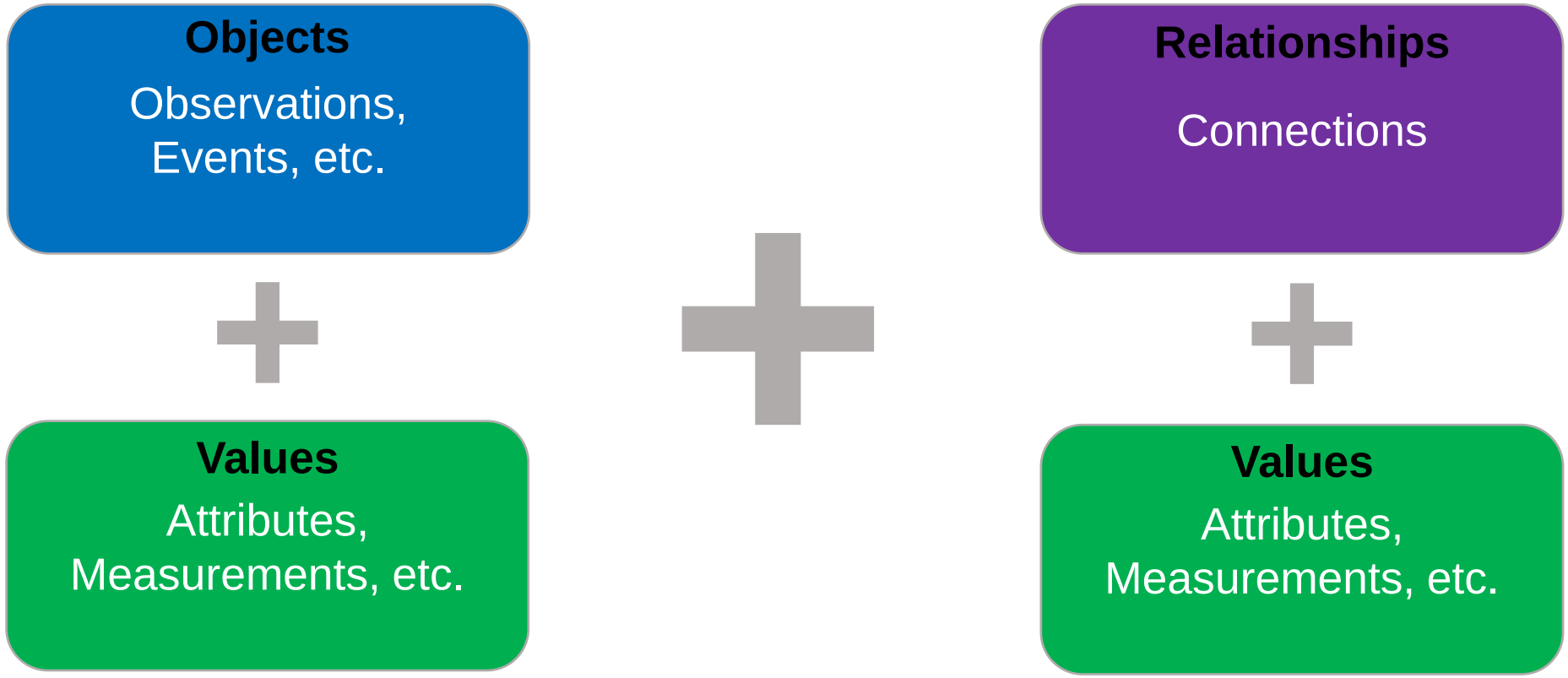
Network data

- A way to model *relationships* between entities
- Also known as:
  - Graph
  - Node-link graph

Tabular data



Network data



Network data

Nodes

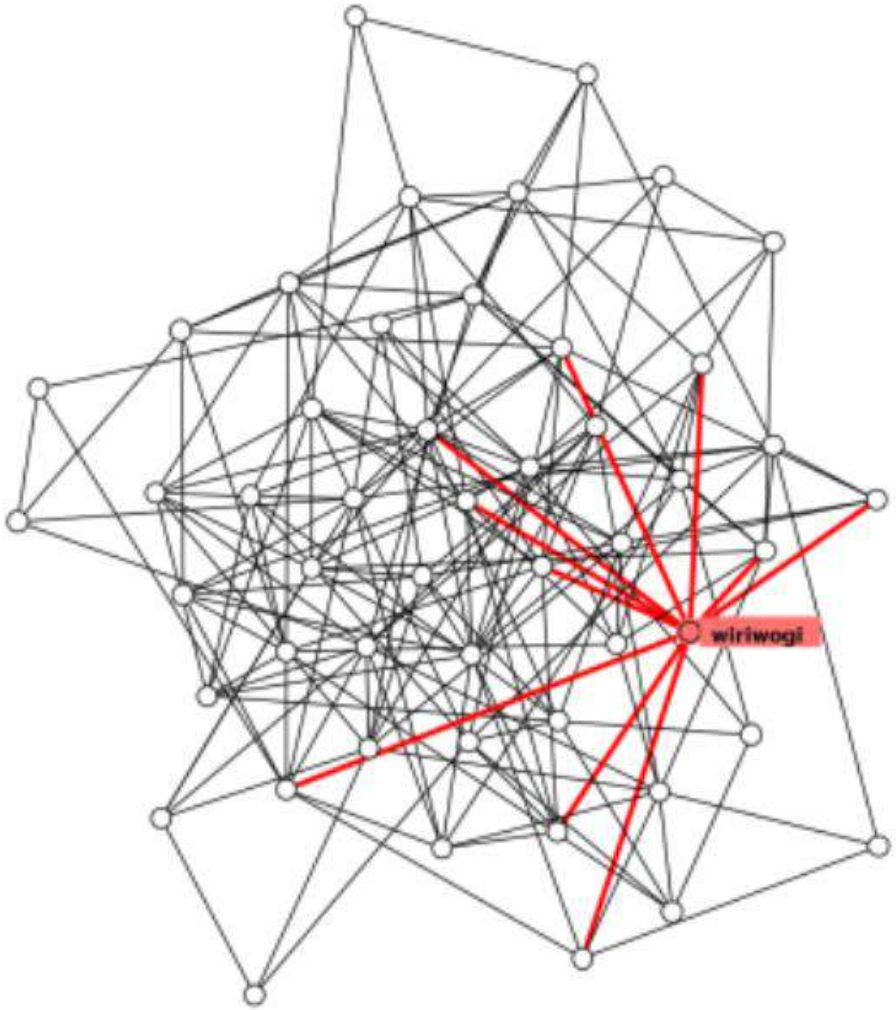
Label	Value
A	5
B	10
C	15

Links

N1	N2	Value
A	B	9
A	C	27
B	C	18

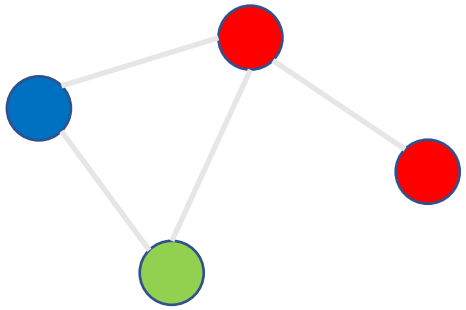
Network tasks: topology-based and attribute-based

- Topology-based tasks:
  - Find paths
  - Find (topological) neighbors
  - Compare centrality
  - Identify clusters/communities
- Attribute-based tasks:
  - Find distributions, extreme values, etc.
- Combination tasks:
  - Use both attribute and connection data!
  - Example?
    - Find all friends-of-friends who like cats
      - Topology: friends of friends (path from you to the other node)
      - Attribute: does the person (node) like cats?

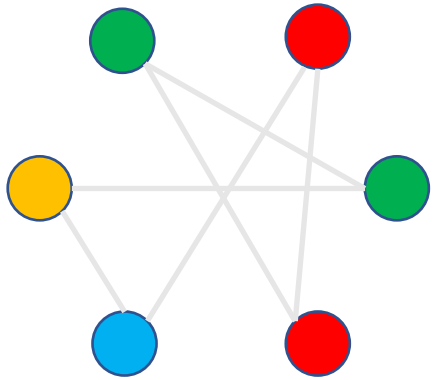


Node-Link Diagram

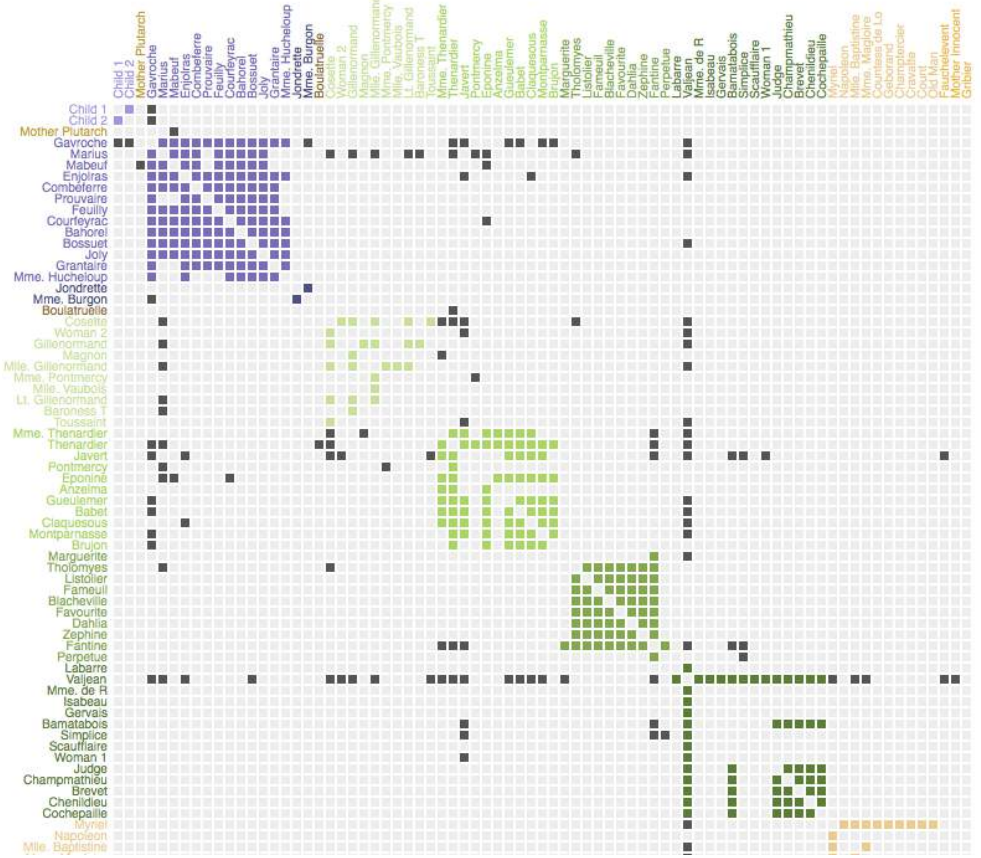
Force-Directed Layouts

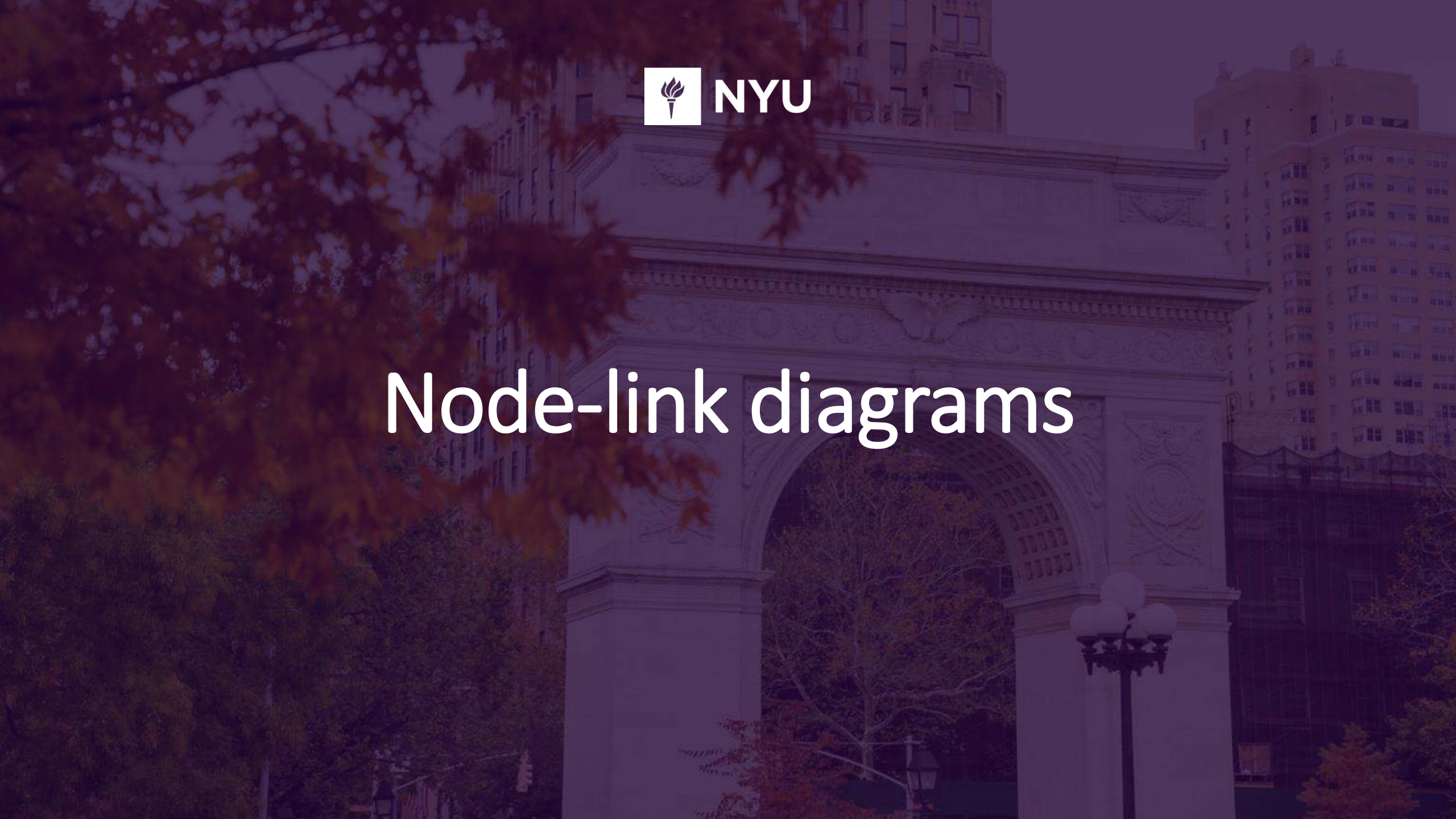


Fixed Layouts



Matrices

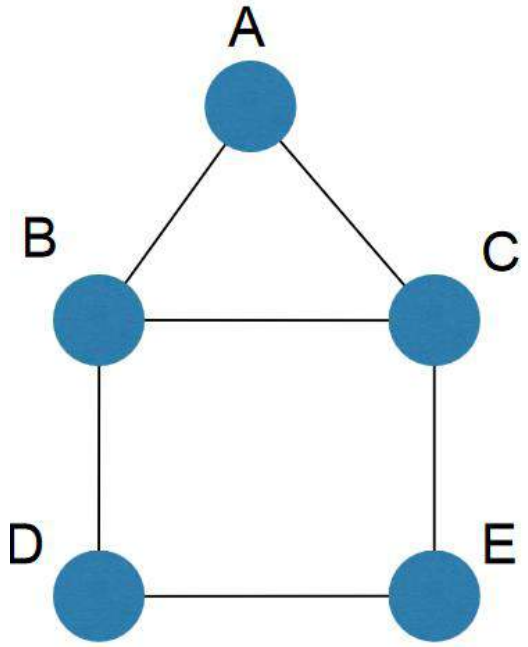


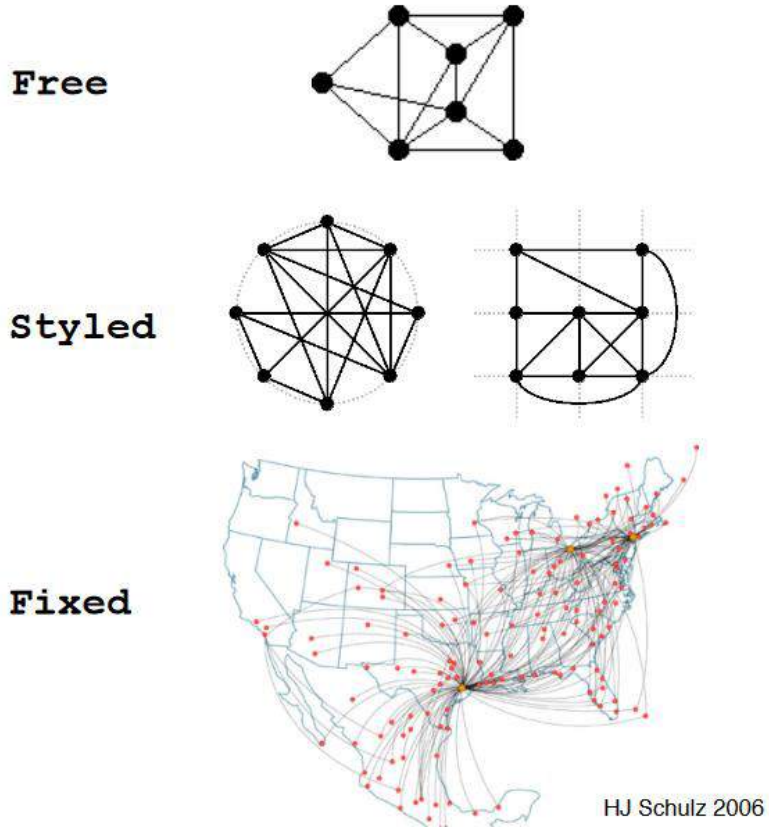


Node-link diagrams

Node-link diagrams

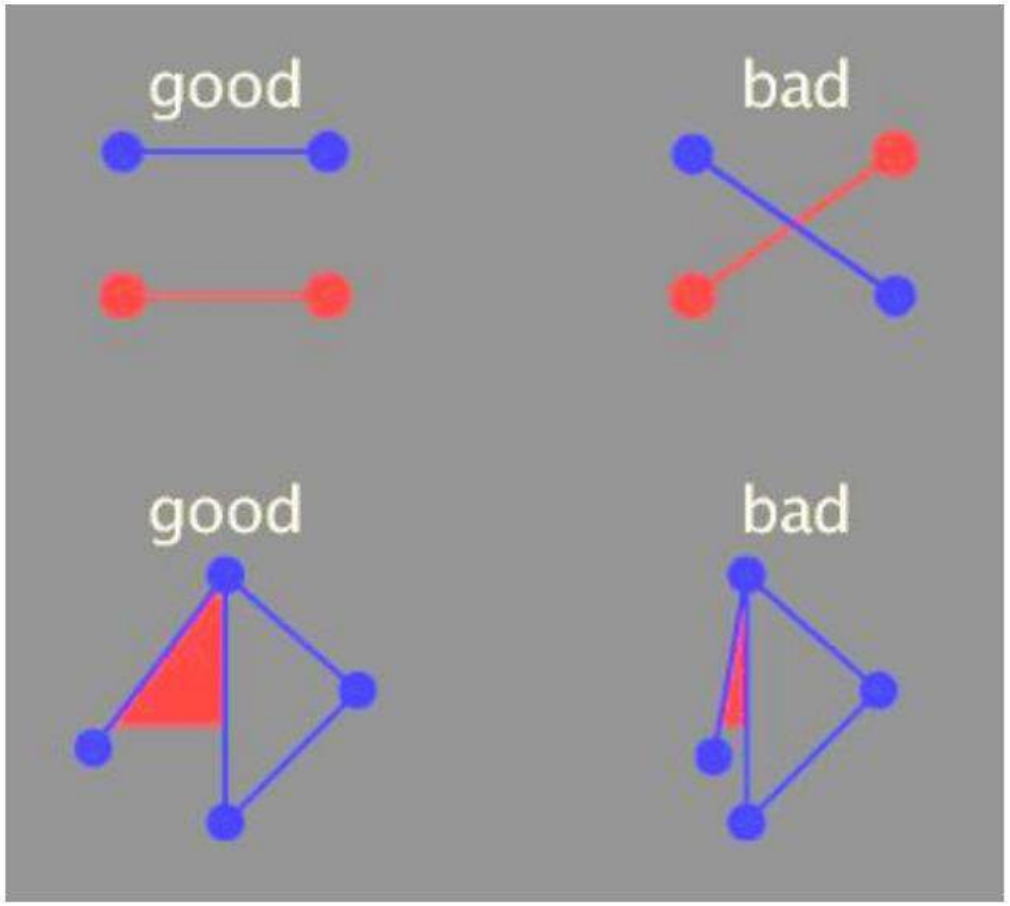
- Nodes: point marks
- Links: line marks
  - Straight lines or arcs
  - Connections between nodes
- Intuitive and familiar
  - Most common
  - Many, many variants





What makes for a good node-link layout?

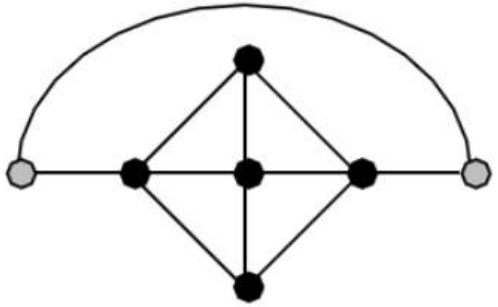
- Minimize:
  - Edge crossings
  - Node overlaps
  - Distances between topological neighbor nodes
  - Total drawing area (white space)
  - Edge bends
- Maximize:
  - Angular distance between edges
  - Aspect ratio disparities
- Emphasize symmetry
  - Similar graph structures should look similar in layout



Criteria conflict

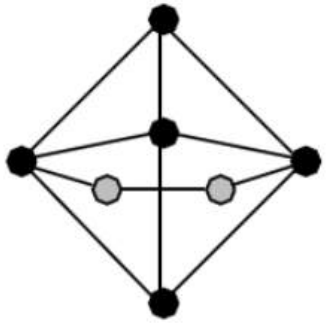
- Meeting all of those criteria is hard
- Most criteria are NP-hard just on their own
- Many criteria directly conflict with each other
- Solution: use heuristics

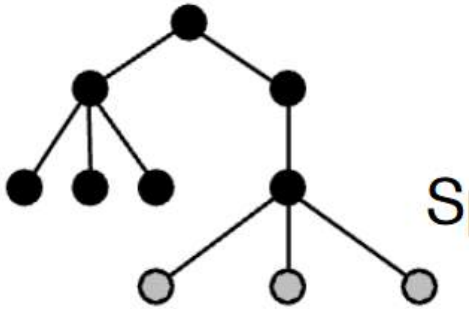
Minimum number of edge crossings



vs.

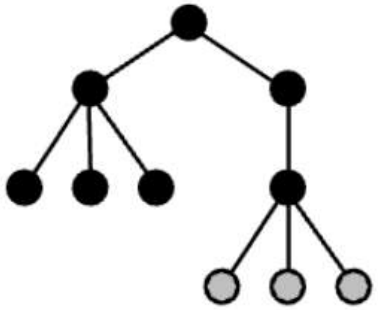
Uniform edge length





Space utilization

vs.



Symmetry

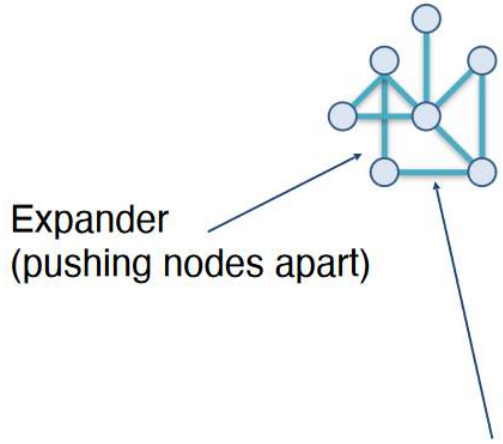
Schulz 2004

Optimization-based layouts

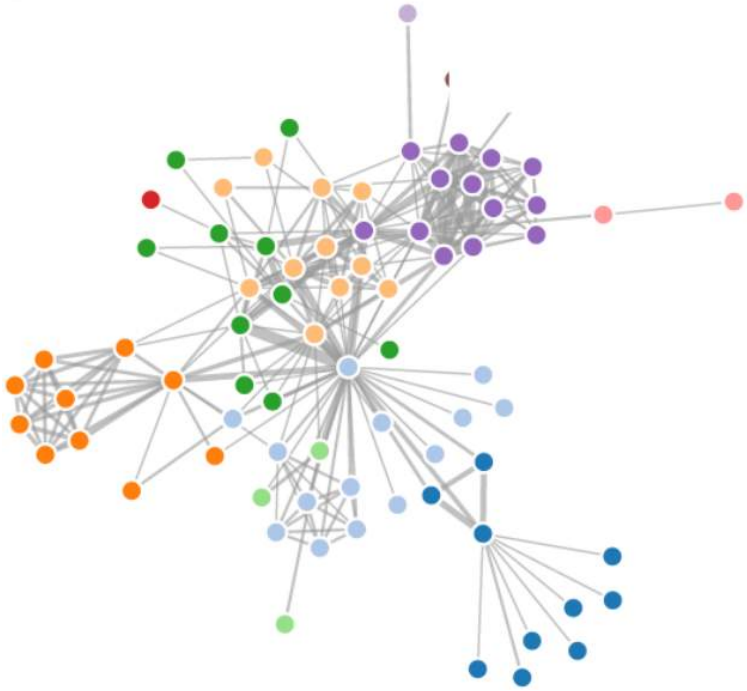
- Formulate layout problem as optimization problem
- Convert criteria into weighted cost function
  - F(layout) = a*[crossing counts] + b*[drawing space used]+…
- Use known optimization techniques to find layout at minimal cost
  - Energy-based physics models
  - Force-directed placement
  - Spring embedders

Force-directed placement

- Physics model
  - Links = springs pull together
  - Nodes = magnets repulse away



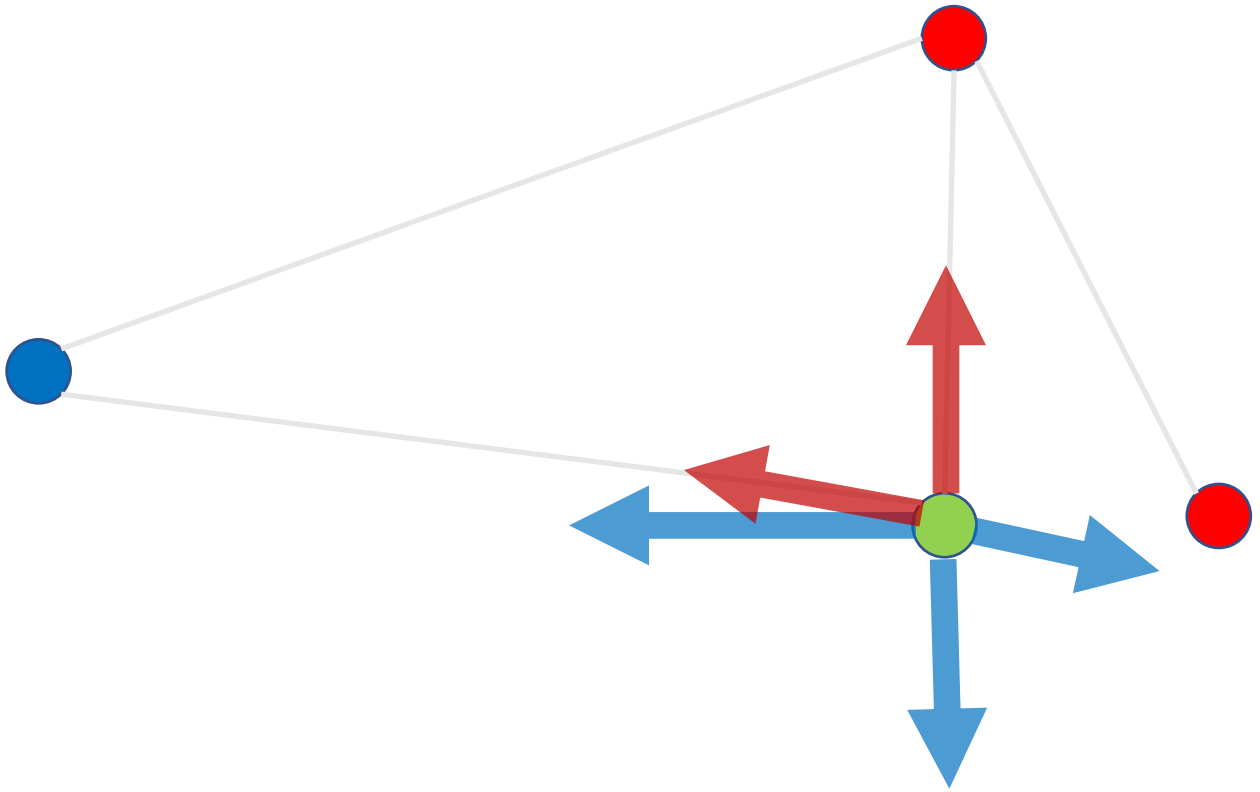
- Algorithm
  - Place vertices in random locations
  - While not in equilibrium:
    - For each vertex:
      - Calculate force on vertex
        - Sum of:
          - Pairwise repulsion of all nodes
          - Attraction between connected nodes
      - Move vertex by c * vertex_force



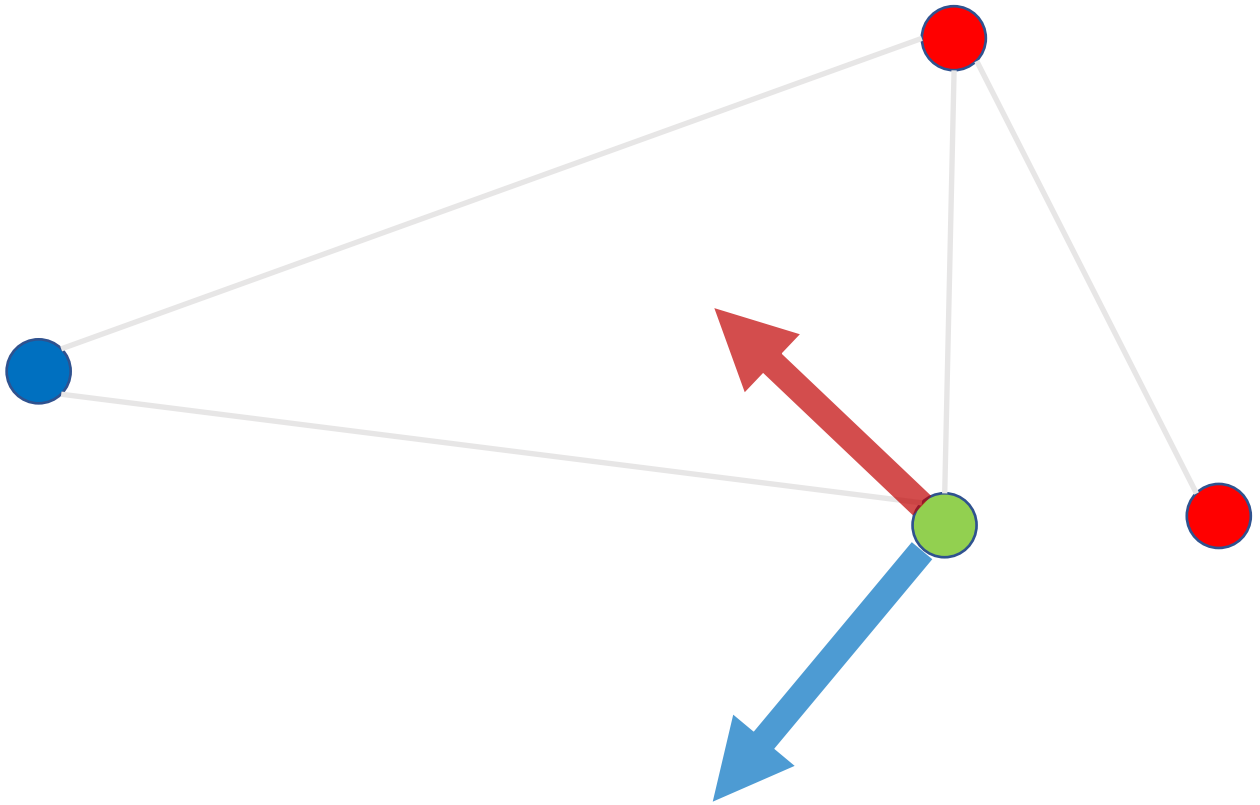
https://observablehq.com/@d3/force-directed-graph-component

Initialize with random layout

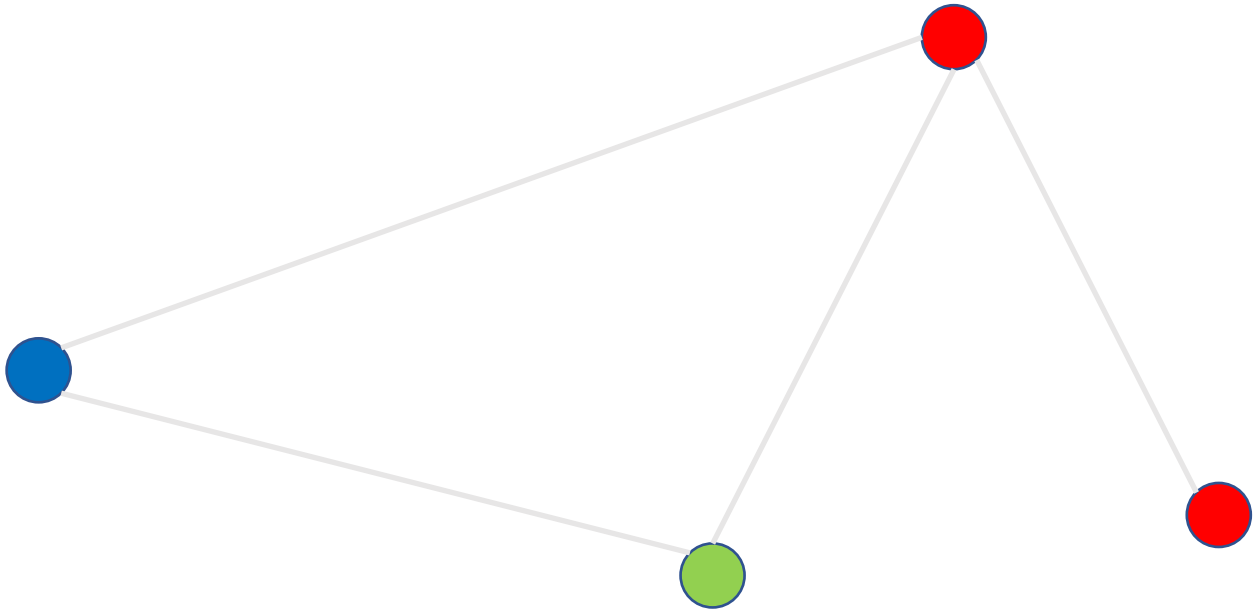
Compute attraction and repulson forces per node



Compute the net force per node



Move the node according to the net force – Hooke's Law



And repeat…

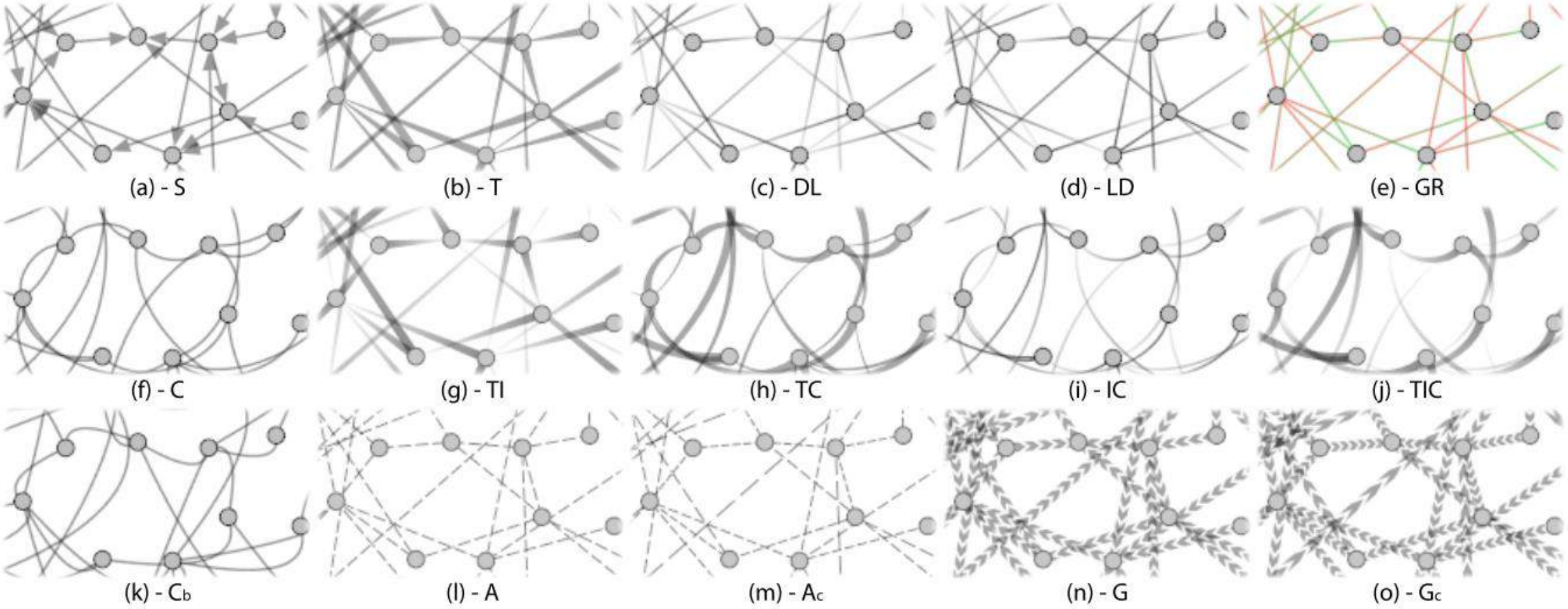
Force-directed placement properties

- Strengths:
  - Reasonable layout for small, sparse graphs
  - Clusters typically visible
  - Edge length uniformity
- Weaknesses:
  - Nondeterministic
  - Computationally expensive: O(n^3) for n nodes
    - Each step is n^2, takes ~n cycles to reach equilibrium
  - Naïve FD doesn’t scale well beyond ~1000 nodes
  - Iterative progress of viz: cool but distracting
  - Visual complexity: E < 4N
    - Max # of edges in a graph is N^N, so 4N is quite sparse!

Force-directed placement

- Considerations:
  - Spatial position: no meaning directly encoded!
    - Left free to minimize crossings
  - Semantics of proximity?
    - Sometimes meaningful
    - Sometimes arbitrary (artifact of layout algorithm)

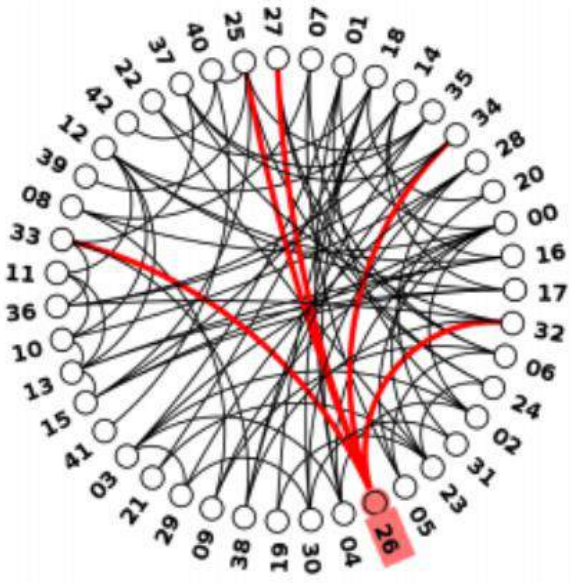
Directed graphs

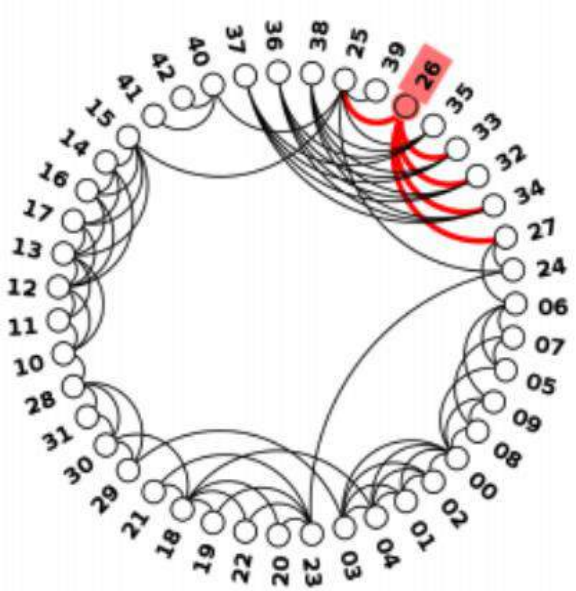


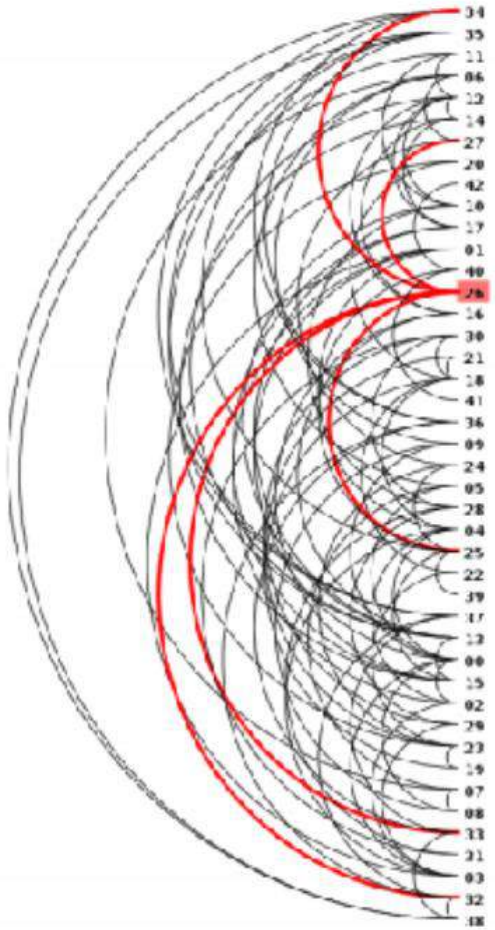
Holten, Danny, et al. "An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs." *IEEE Pacific Visualization Symposium (PacificVis), 2011*.

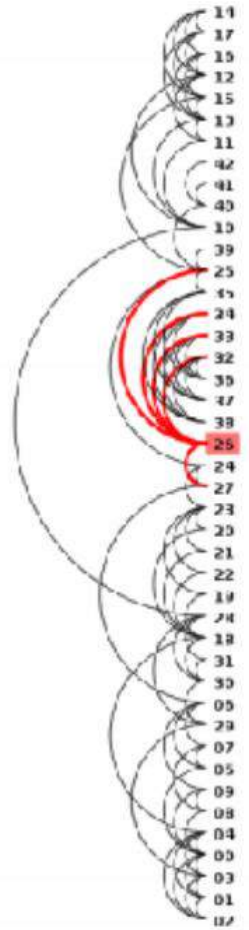
Fixed layout: circular layout/arc diagram (node-link)

- Alternative node-link layouts: lay out nodes around circle or along line
- Data:
  - Original: network
  - Derived: node ordering attribute (global computation)
- Considerations: node ordering is crucial to avoid excessive clutter from edge crossing
  - Example: Barycentric ordering





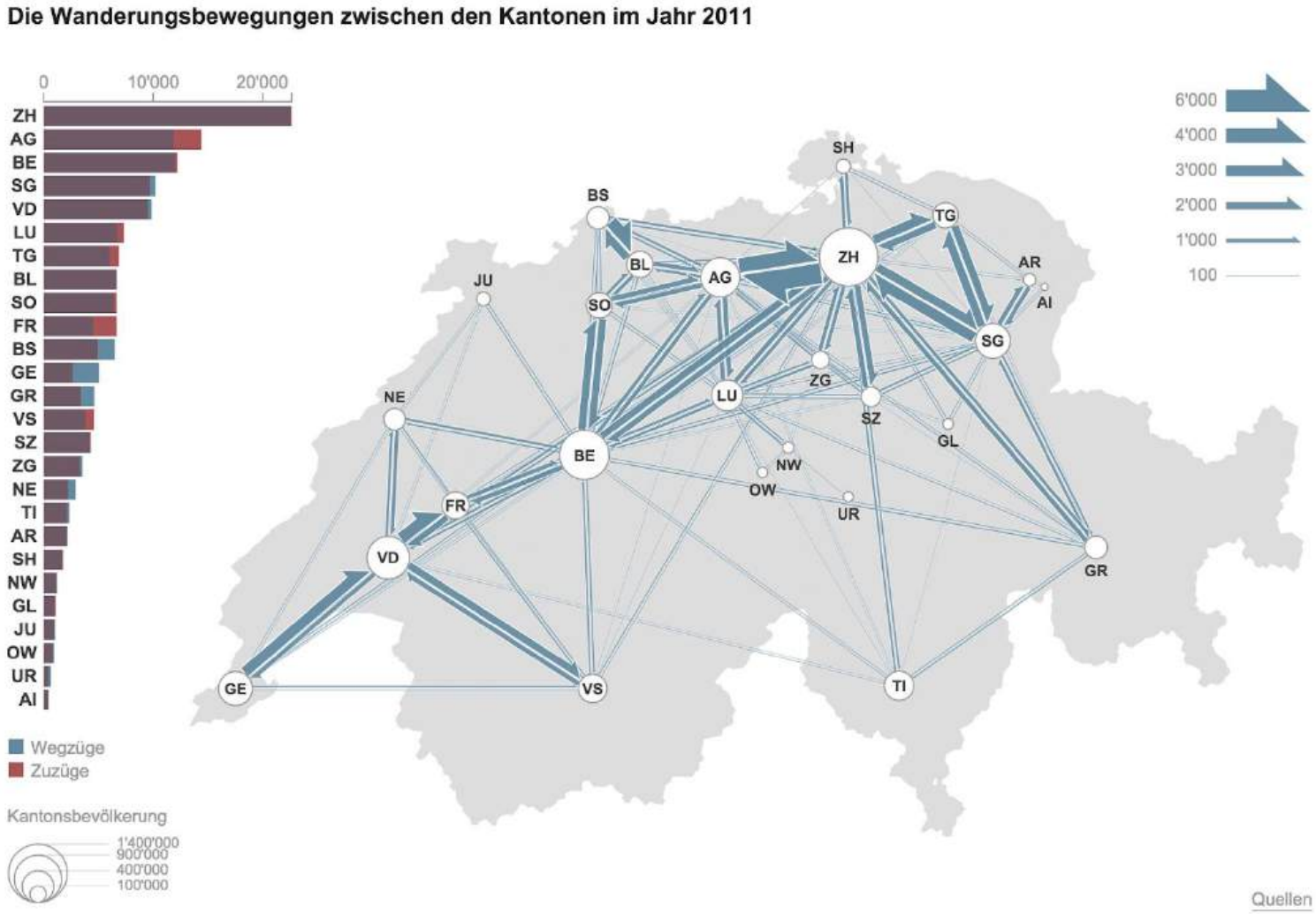




http://profs.etsmtl.ca/mmcguffin/research/2012-mcguffin-simpleNetVis/mcguffin-2012-simpleNetVis.pdf

Fixed layout: spatial

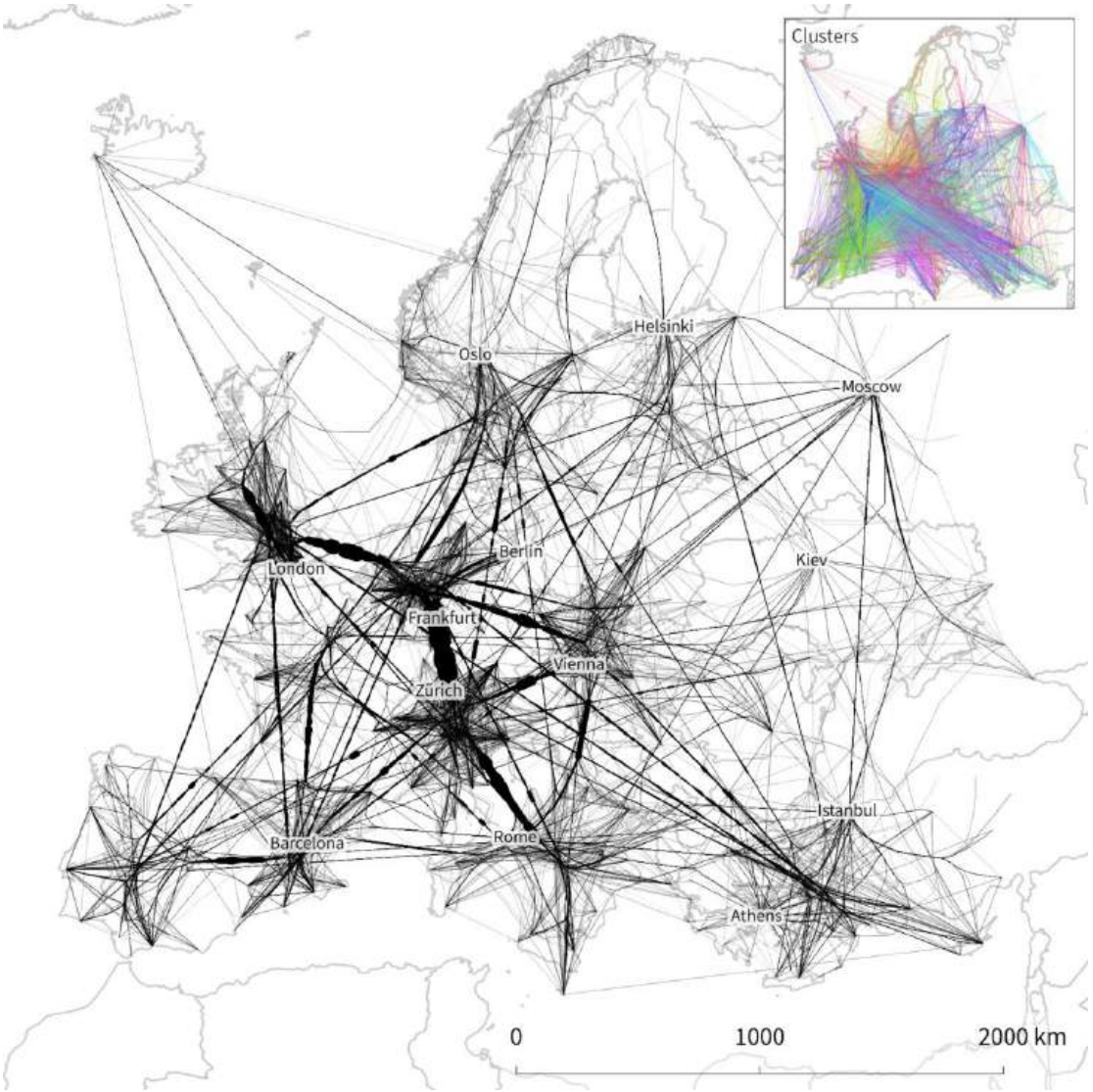
- Nodes are placed based on geographical coordinates
- Gains all the benefits of spatial visualizations:
  - Familiarity
  - Semantic meaning of position



https://ilya.boyandin.me/works/2013/08/30/summerseries

Fixed layout: spatial

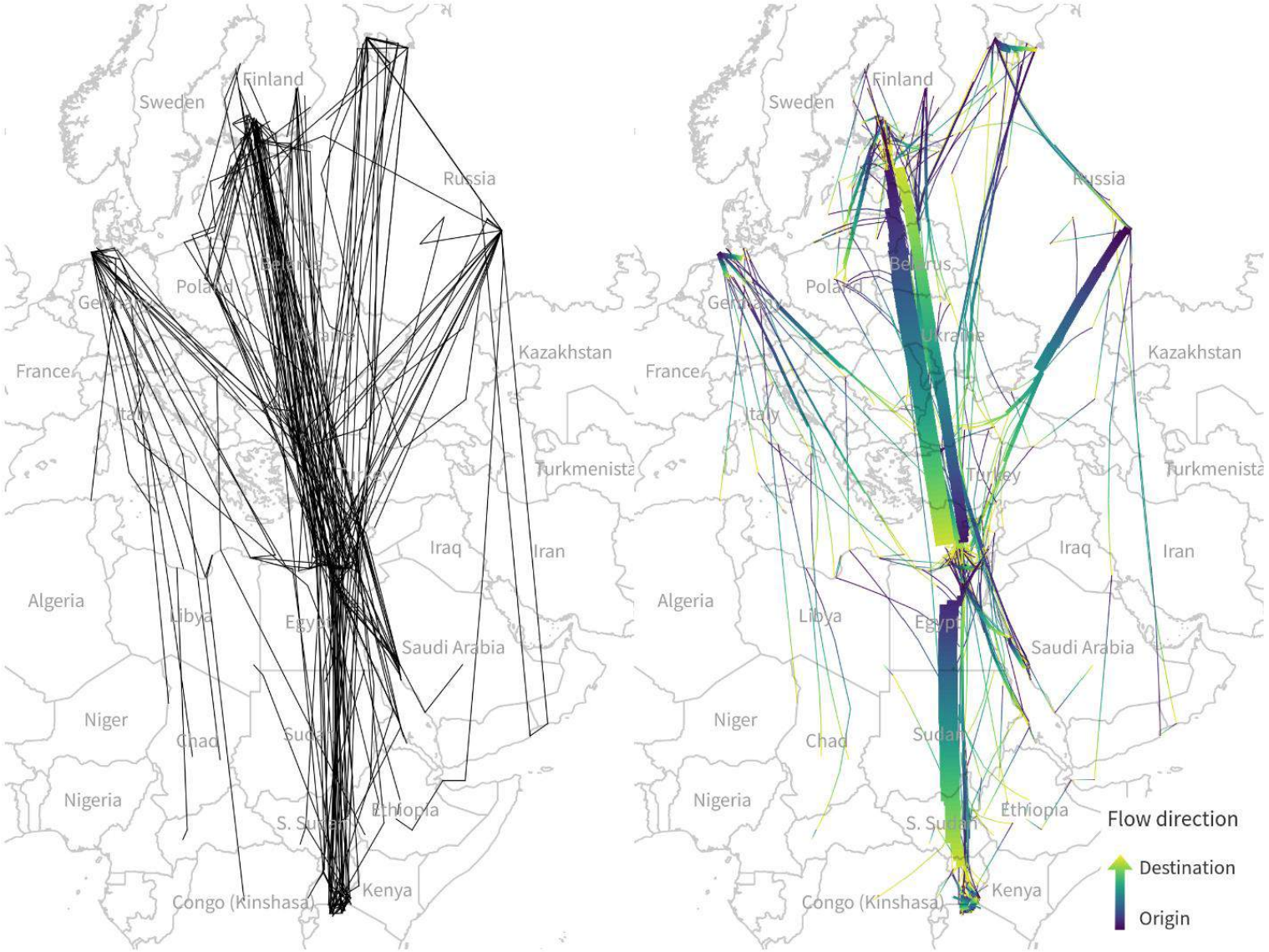
- Watch out for clutter!



https://github.com/dts-ait/qgis-edge-bundling

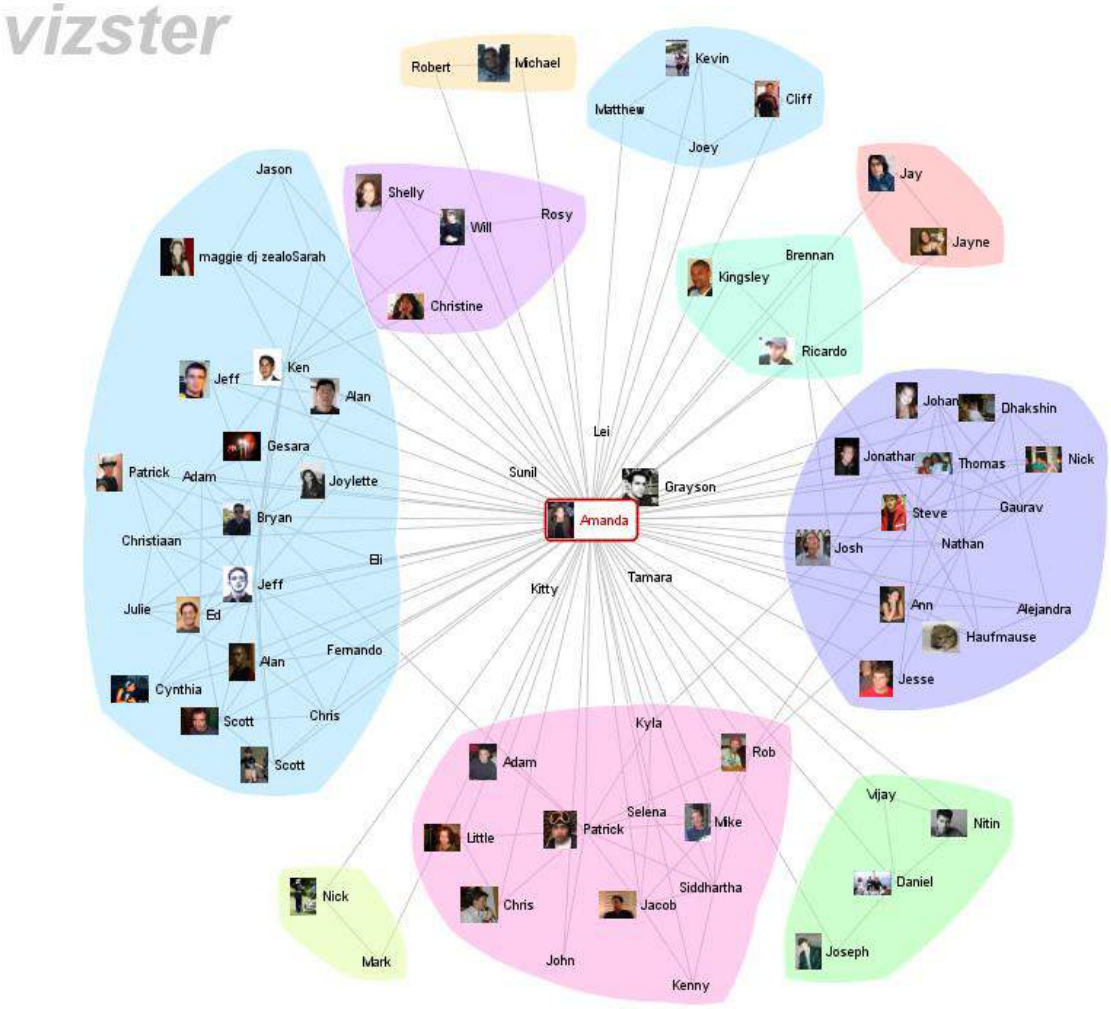
Reducing clutter

- Edge bundling



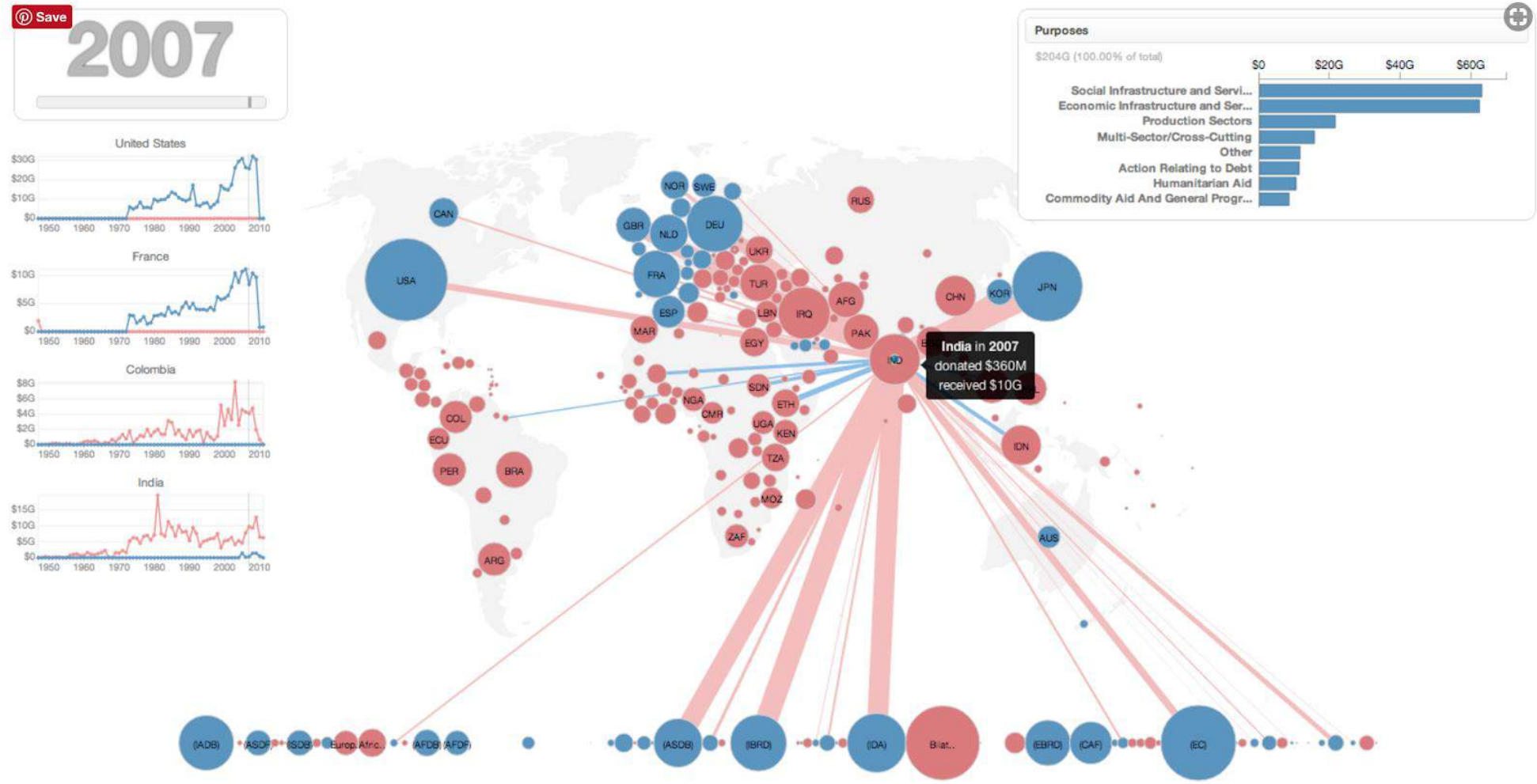
Reducing clutter

- Edge bundling
- Clustering



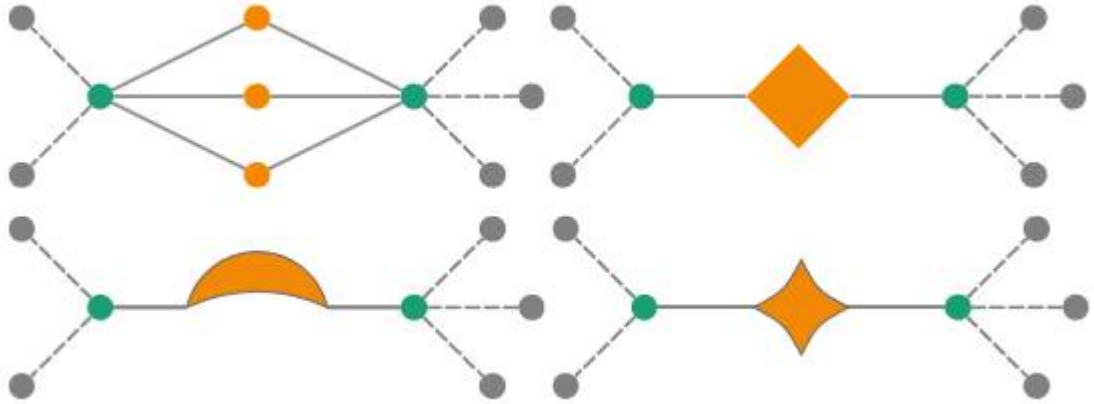
Reducing clutter

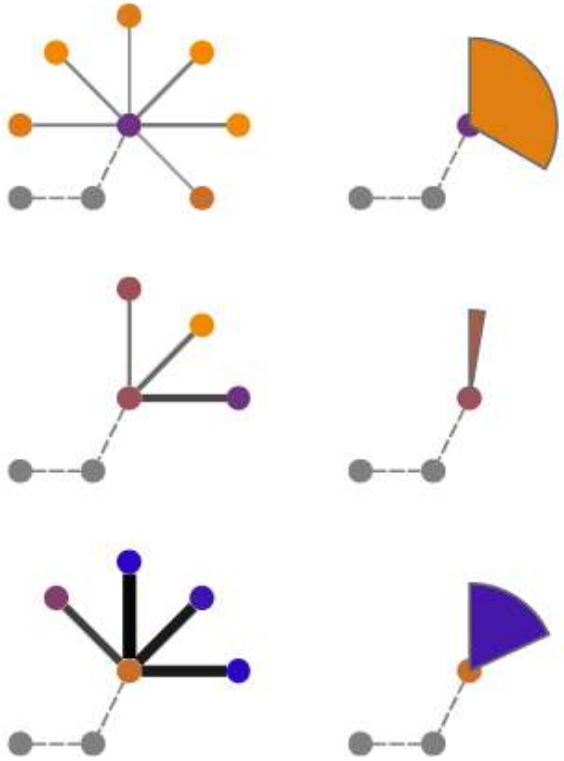
- Edge bundling
- Clustering
- Edges on demand

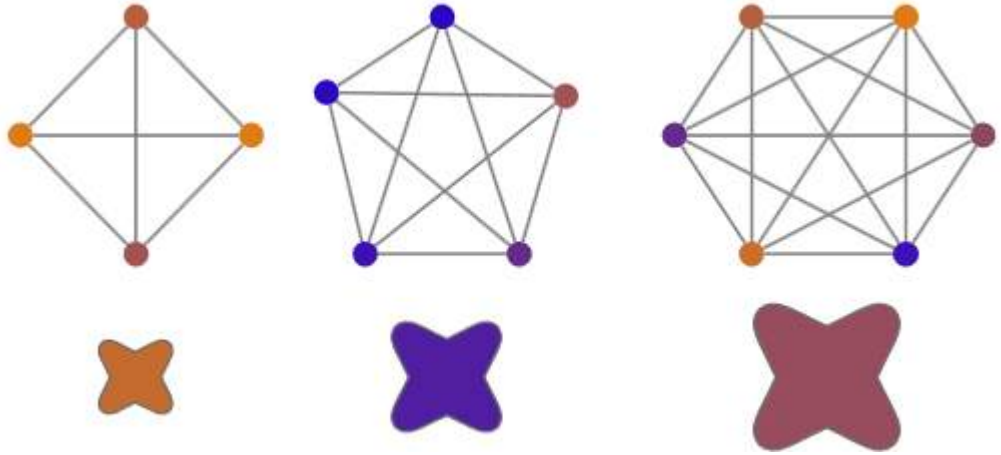


Reducing clutter

- Edge bundling
- Clustering
- Removing Edges
- Aggregation/Simplification



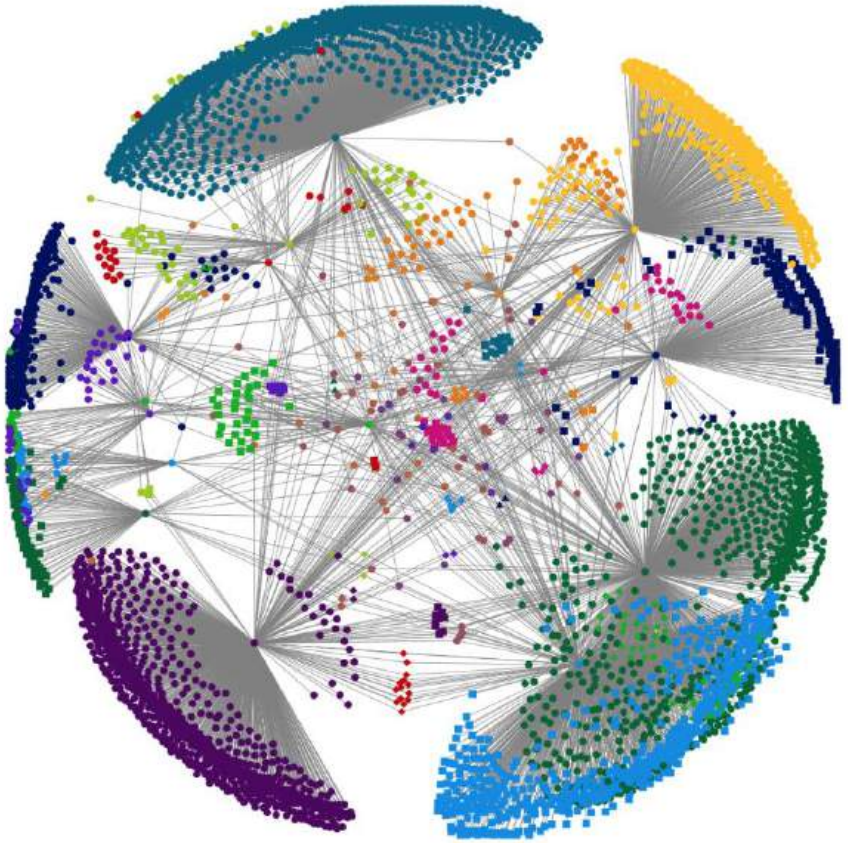




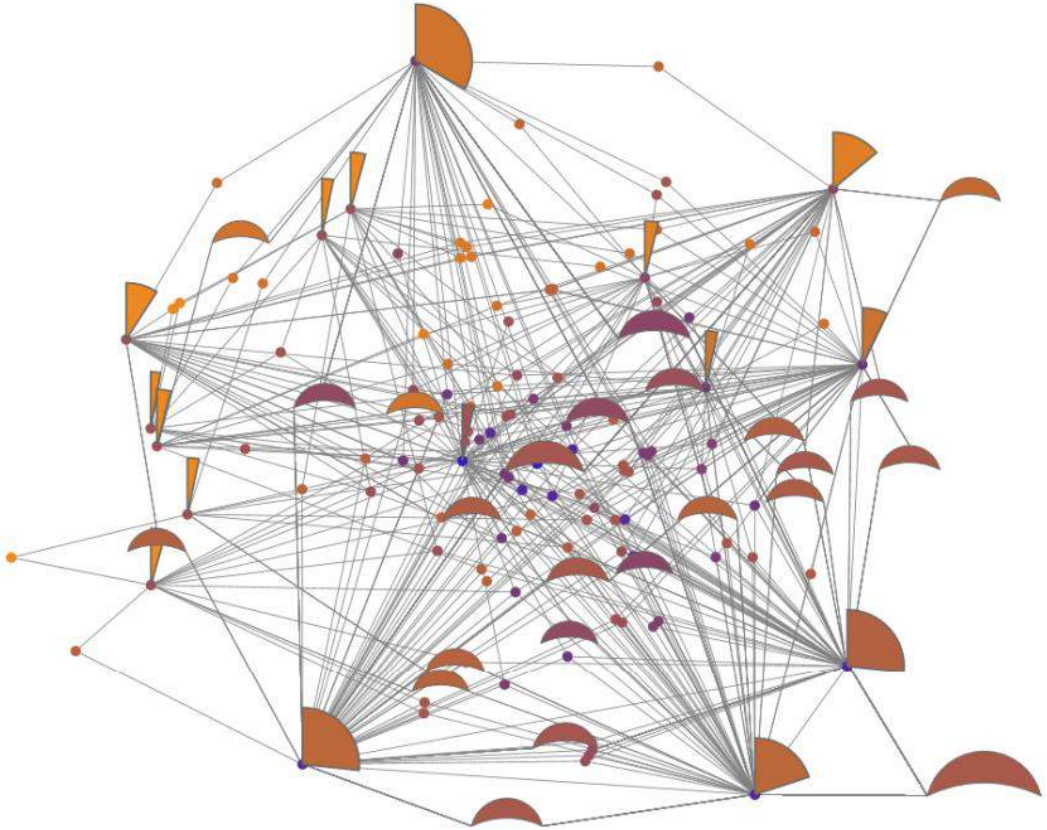
Dunne, Cody, and Ben Shneiderman. "Motif simplification: improving network visualization readability with fan, connector, and clique glyphs." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.

Reducing clutter

- Edge bundling
- Clustering
- Removing Edges
- Aggregation/Simplification

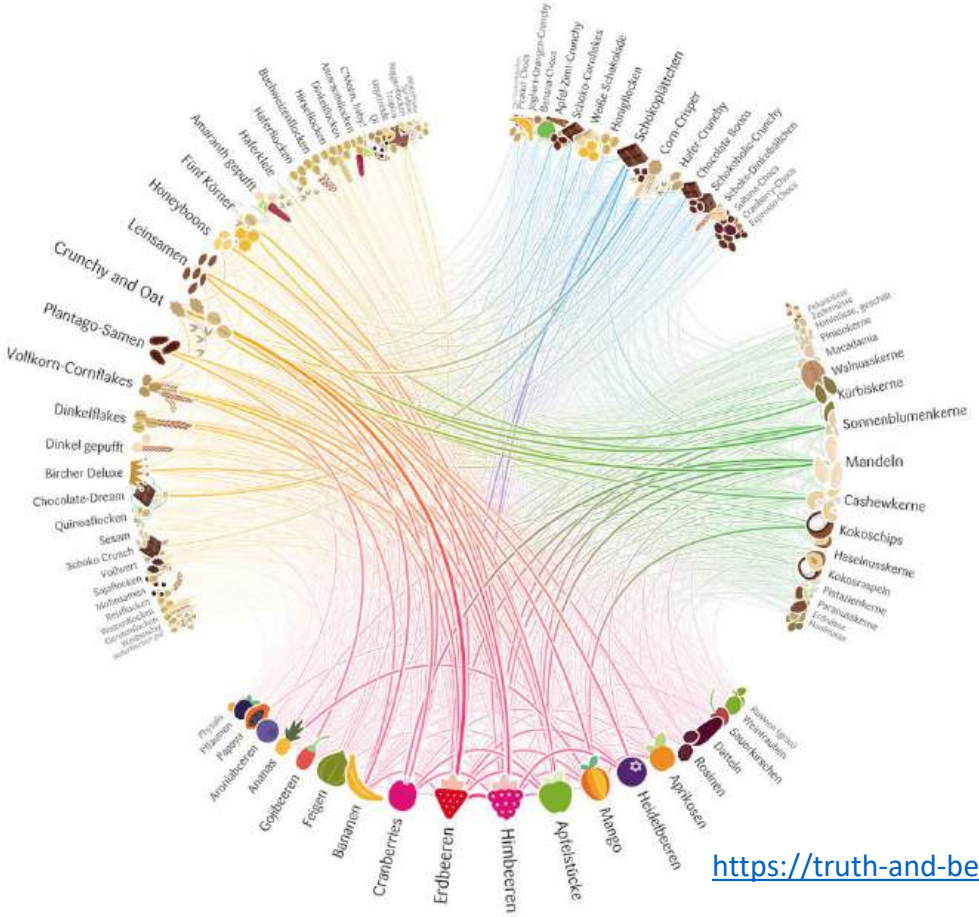


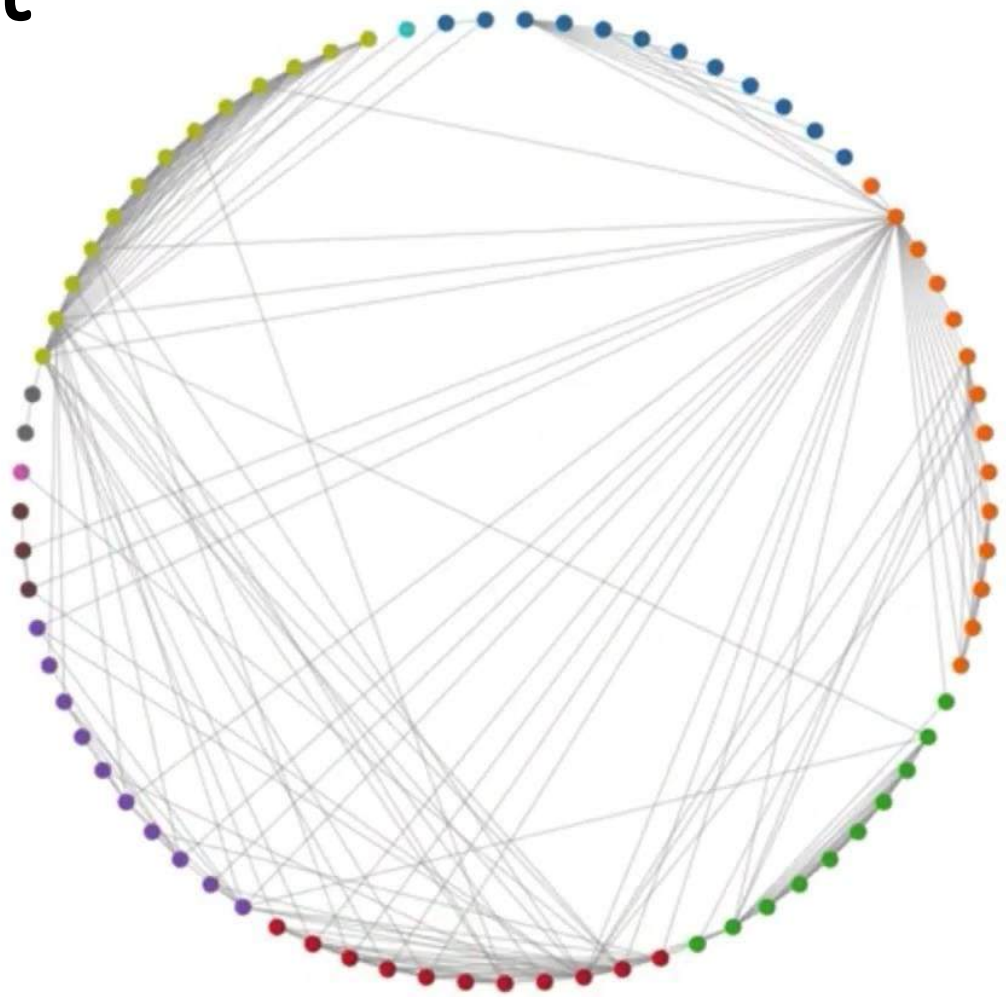
Dunne, Cody, and Ben Shneiderman. "Motif simplification: improving network visualization readability with fan, connector, and clique glyphs." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.



Force-directed vs fixed layout

- Force-directed:
  - Reveals the **structure** of the network
- Fixed layout:
  - Improves visibility of nodes and edges



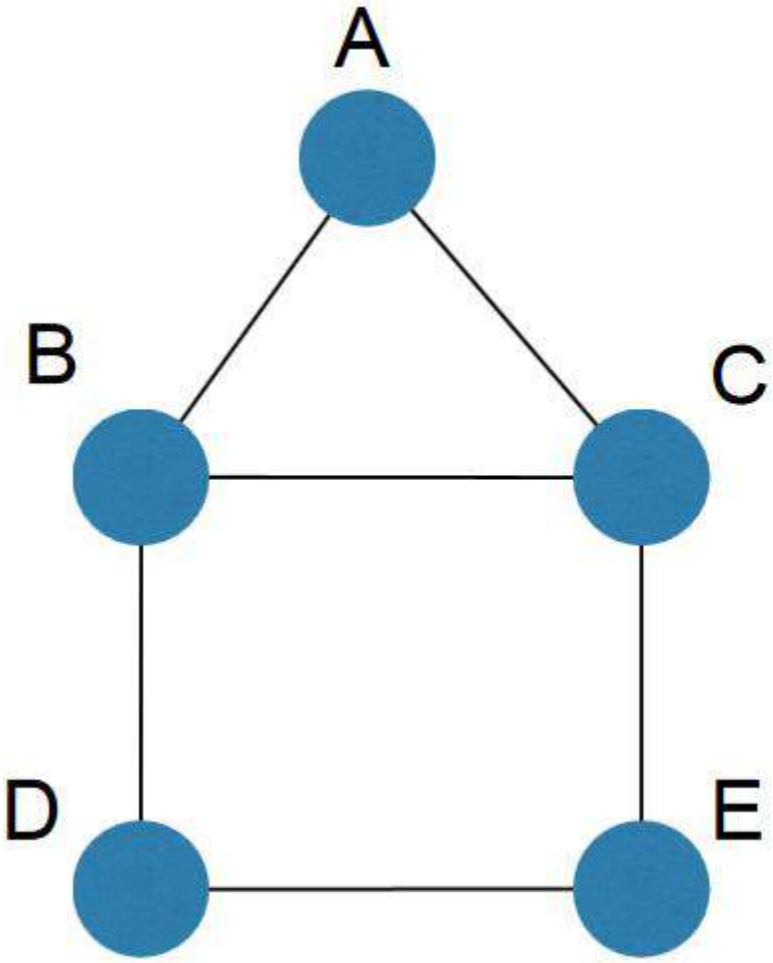


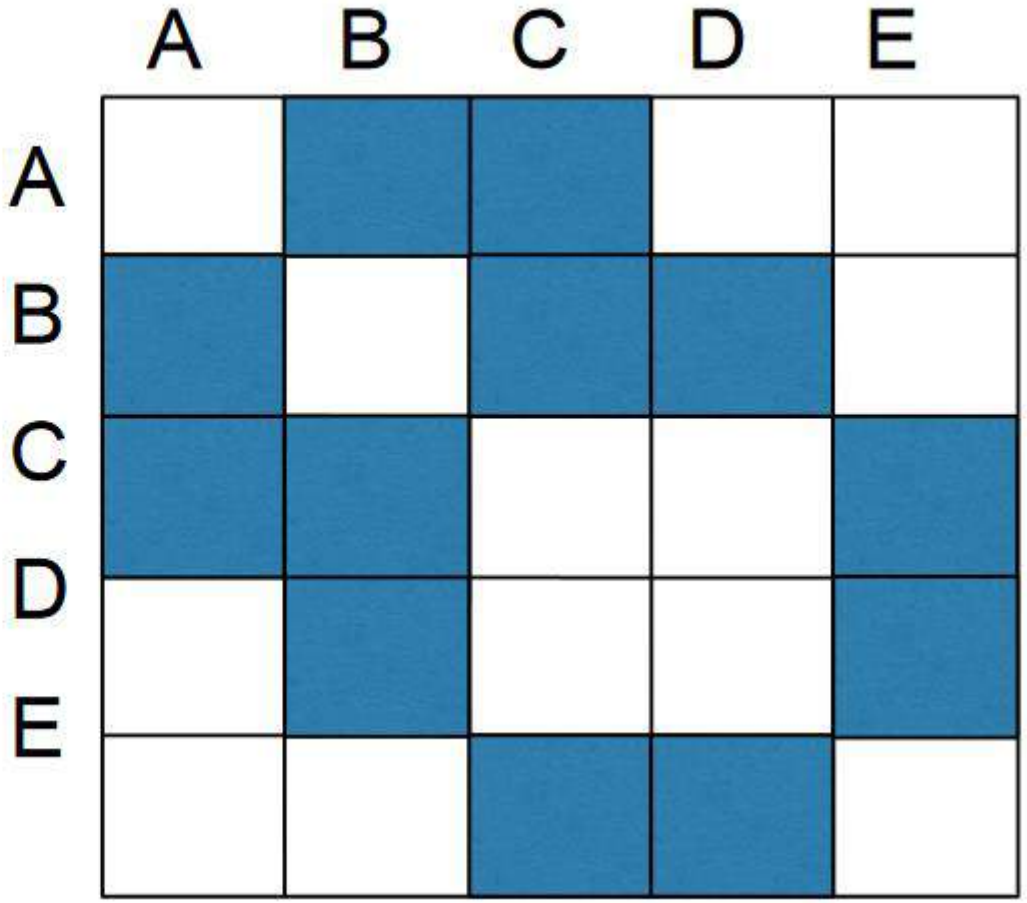
https://truth-and-beauty.net/projects/muesli-ingredient-network

Matrices

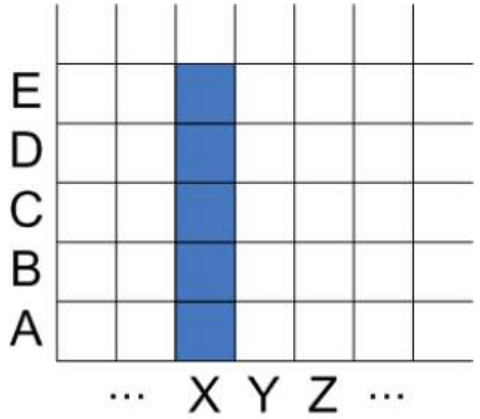
Adjacency matrix representations

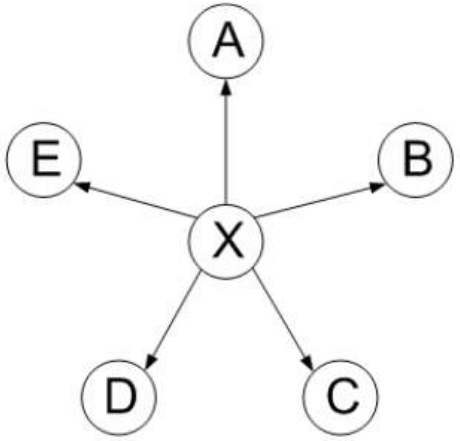
- Derive adjacency matrix from network connections
  - Basically turning it into a table

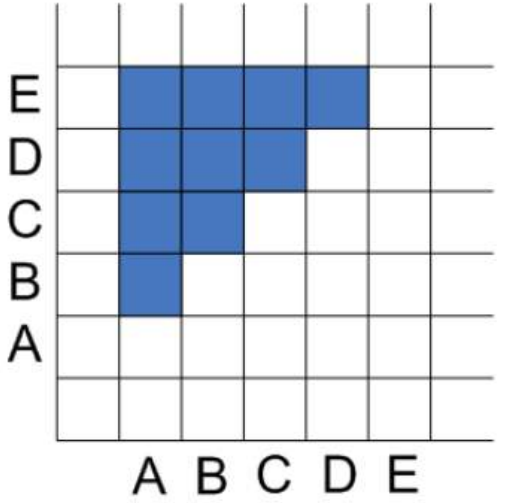


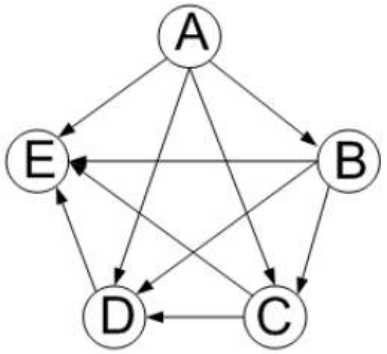


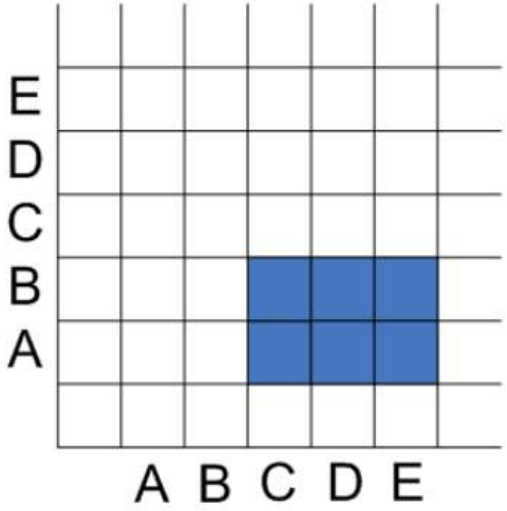
Adjacency matrix examples

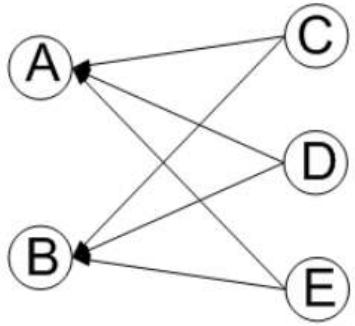


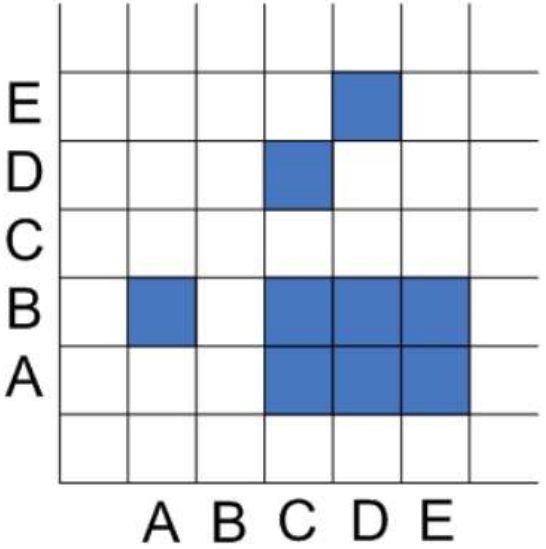


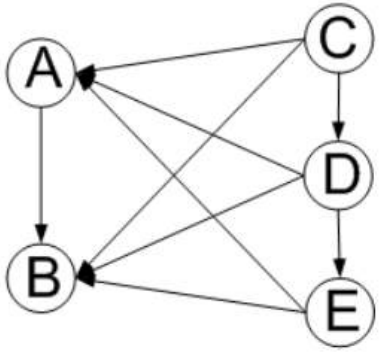






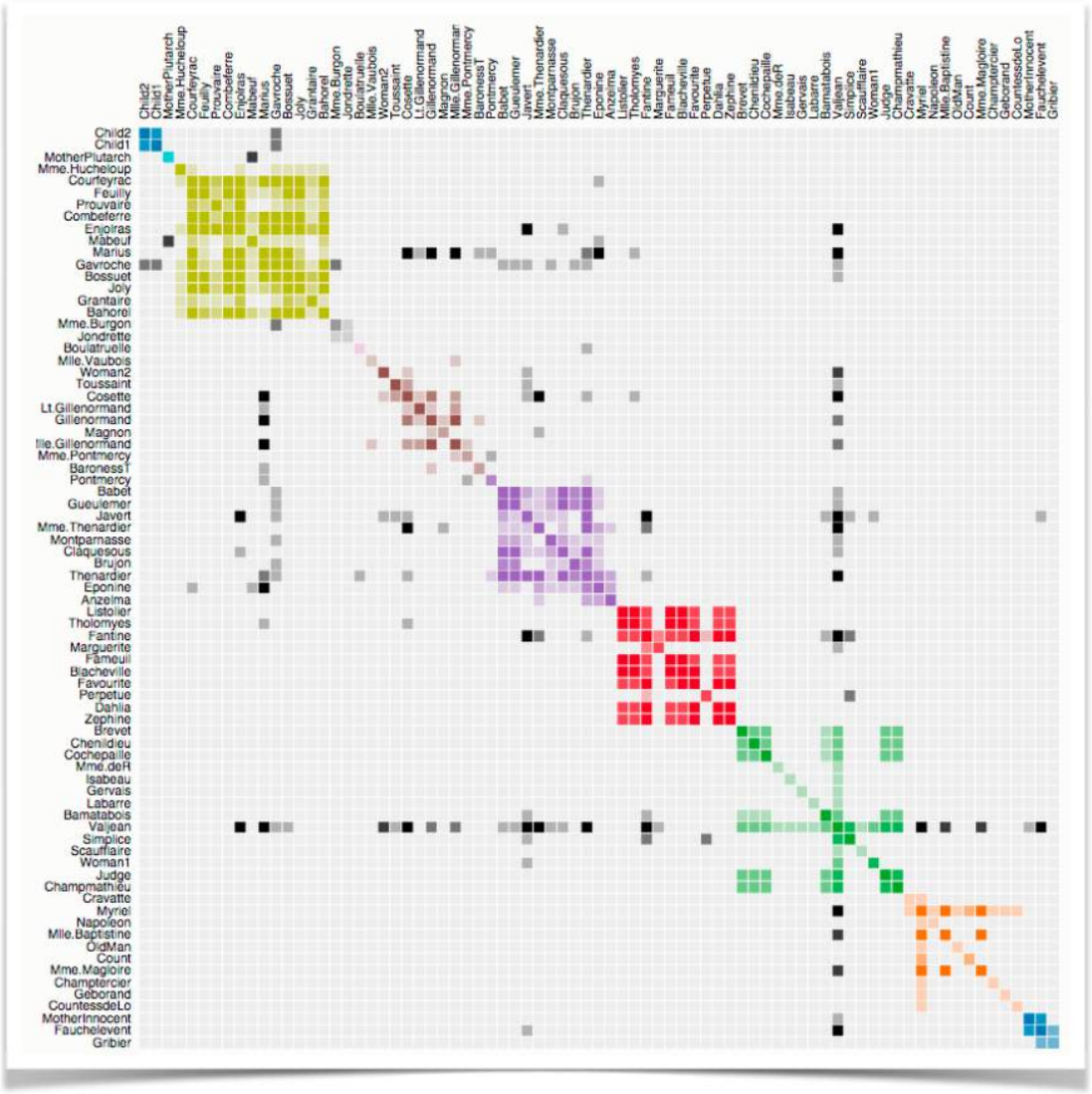






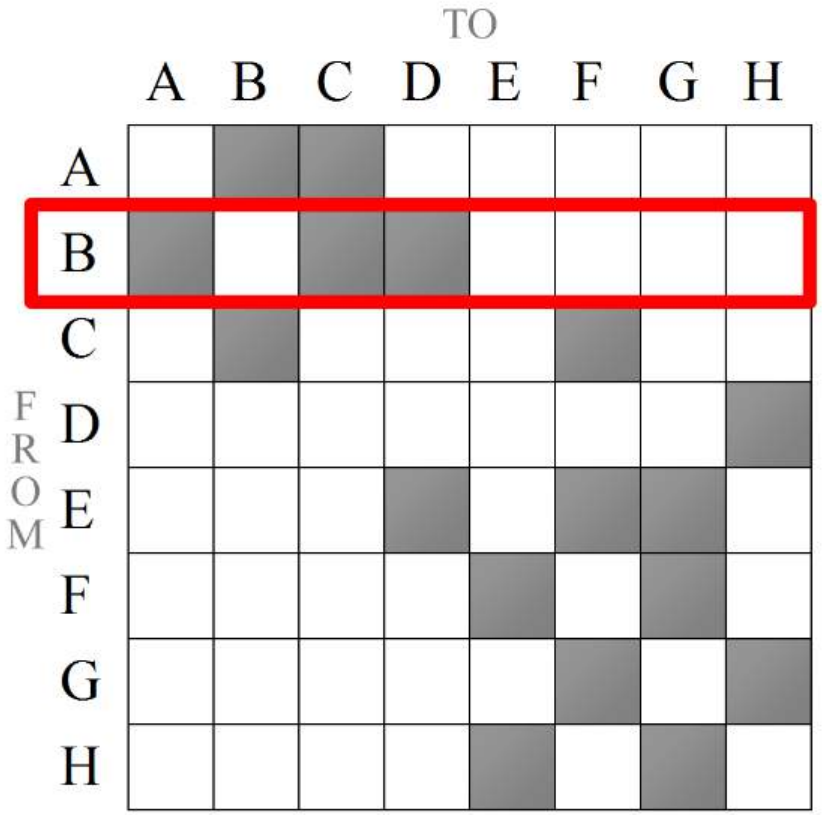
Node order is crucial: Reordering

- Just like circular/arc node-link diagrams

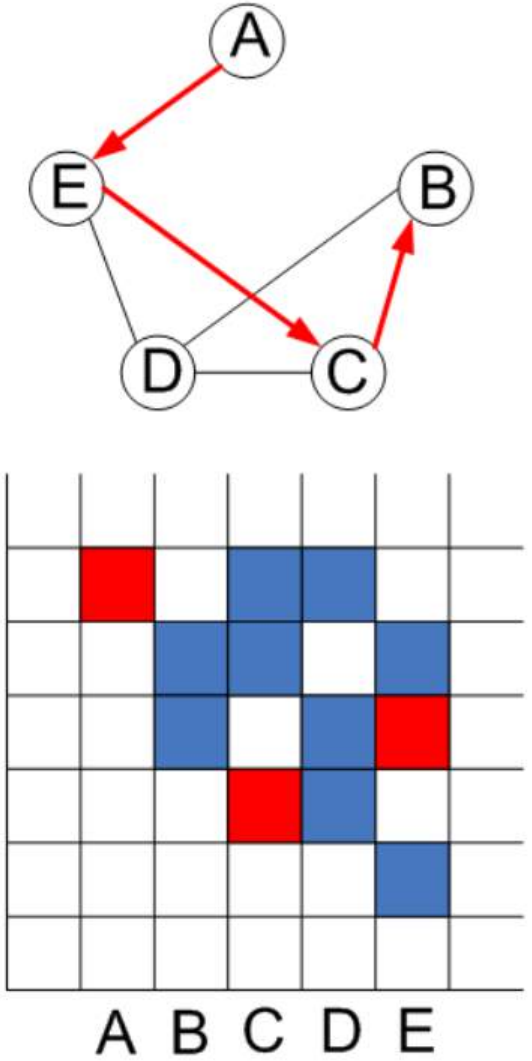


https://bost.ocks.org/mike/miserables/

Adjacency matrix pros and cons



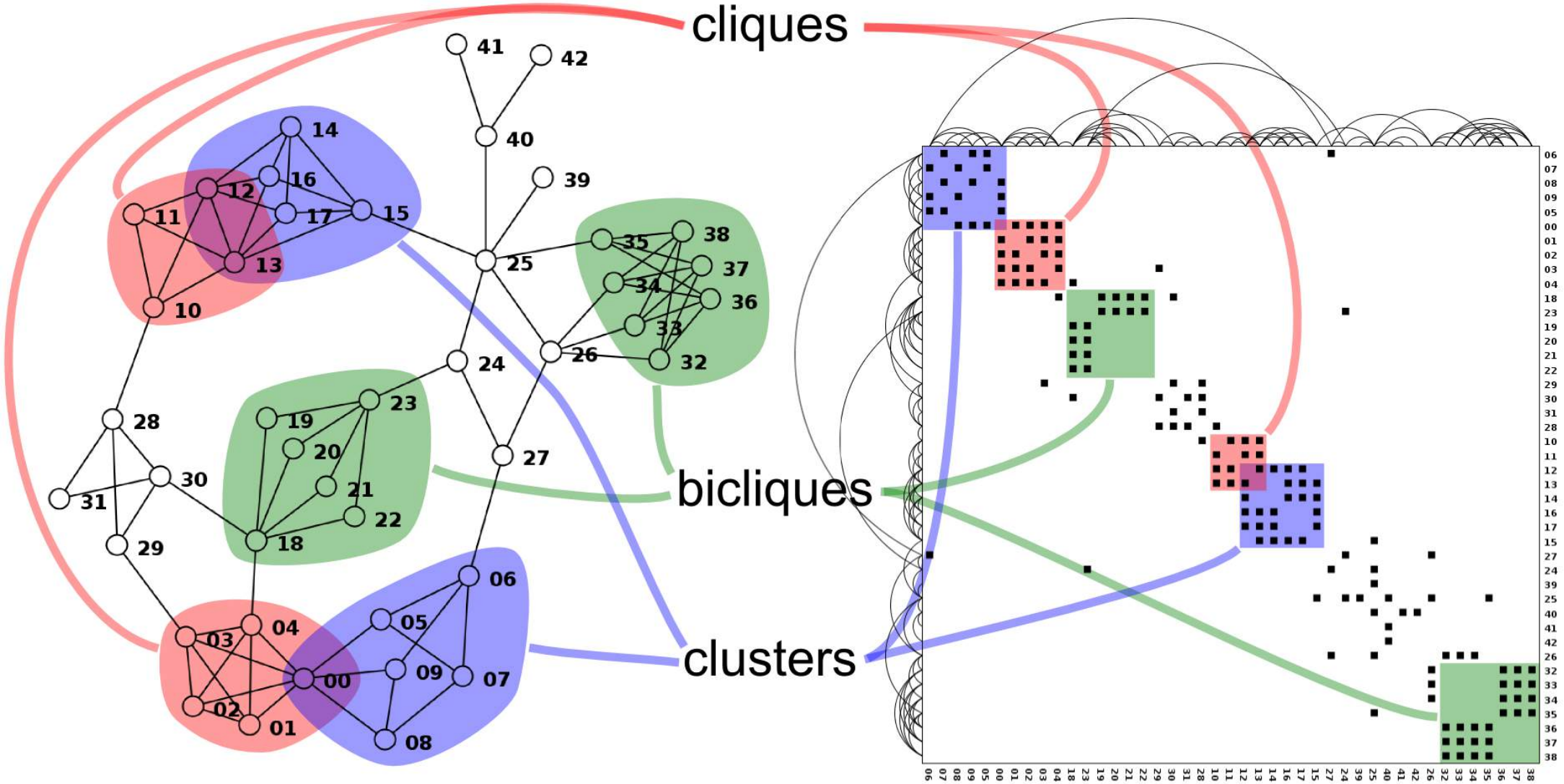
Good for neighborhood tasks
(node 1-hop neighbors)



Bad for tasks related to paths

Structure in both

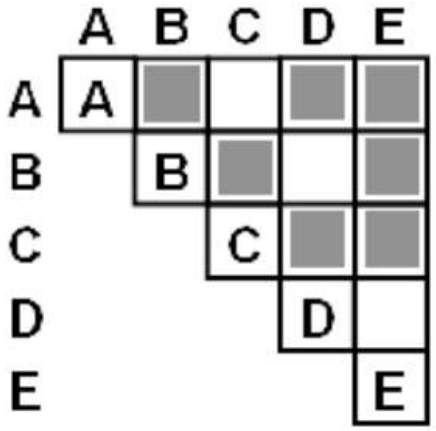
- Typically need to be taught to recognize

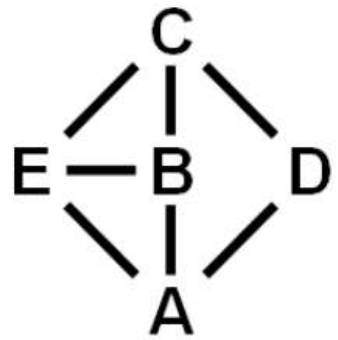


https://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

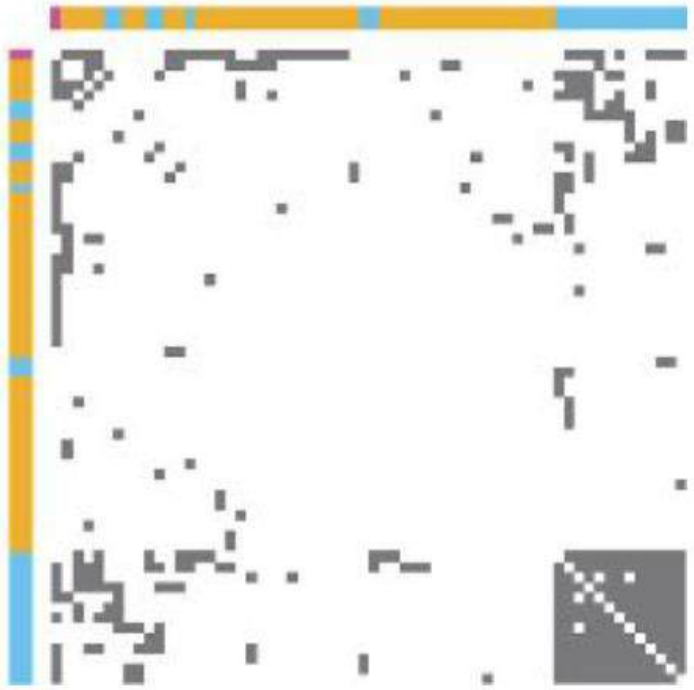
Adjacency matrix

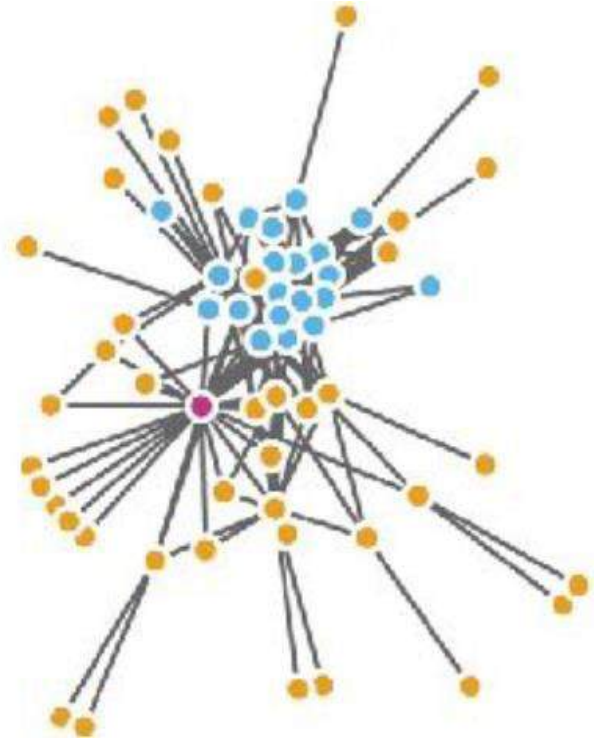
- Data: network
- Derived data: table from network
  - 1 quant. attribute
    - Weighted edge between nodes
  - 2 categorical attributes: node list 2x
- Visual encoding:
  - Cell shows presence/absence of edge
- Scalability
  - 1000 nodes, 1M edges (no clutter!)





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

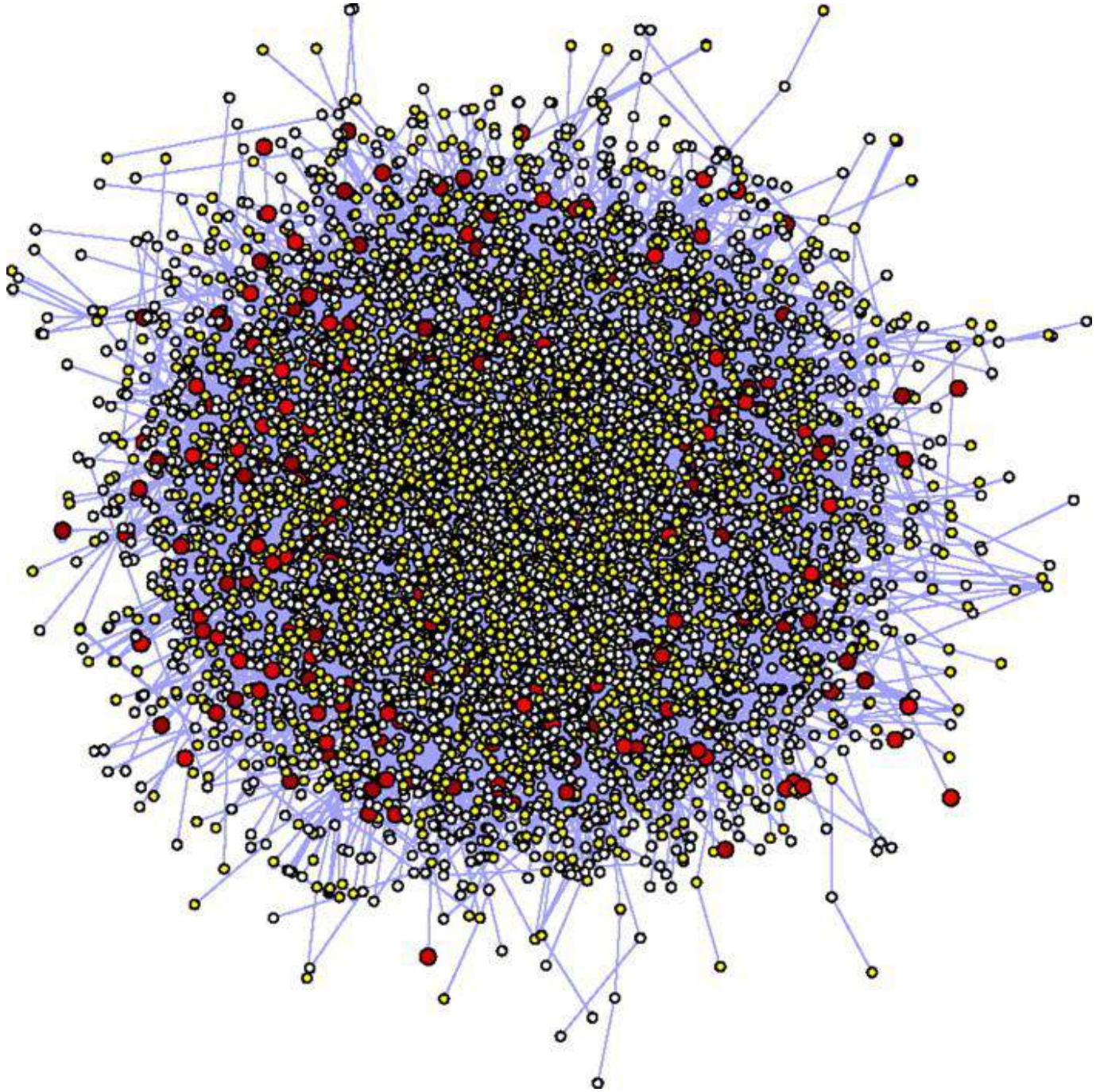




Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115

Adjacency matrix

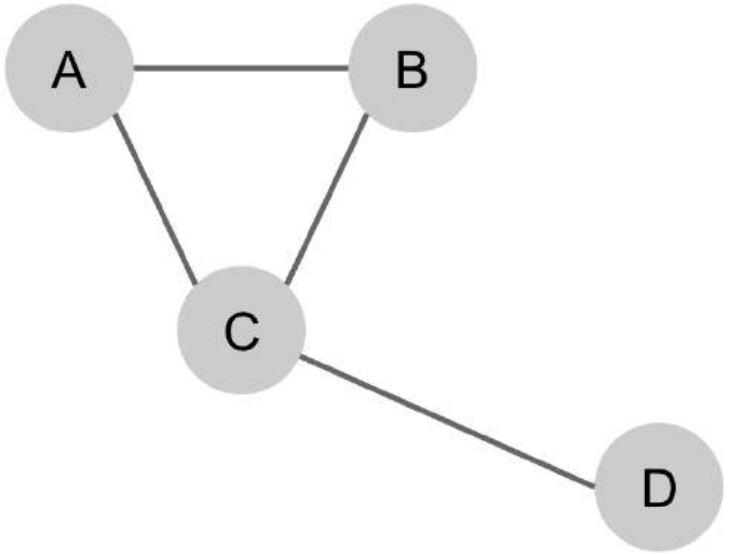
- Data: network
- Derived data: table from network
  - 1 quant. attribute
    - Weighted edge between nodes
  - 2 categorical attributes: node list 2x
- Visual encoding:
  - Cell shows presence/absence of edge
- Scalability
  - 1000 nodes, 1M edges (no clutter!)
- Avoid hairball effect!



Adjacency matrix

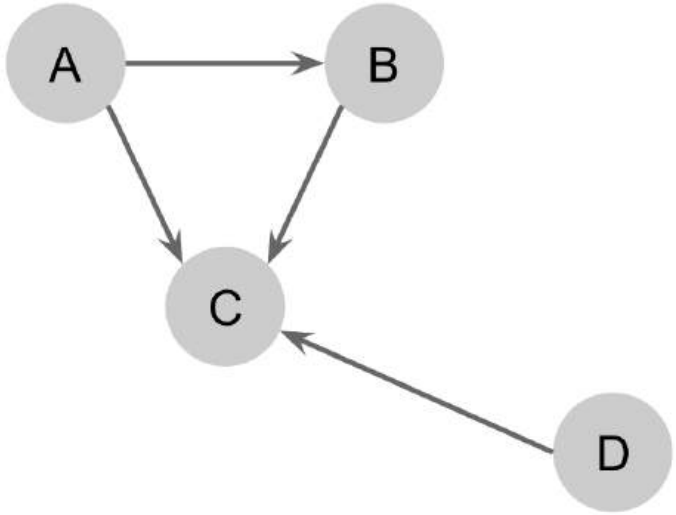
- Can also show directed graphs

Undirected Network



	A	B	C	D
A		■	■	
B	■		■	
C	■	■		■
D			■	

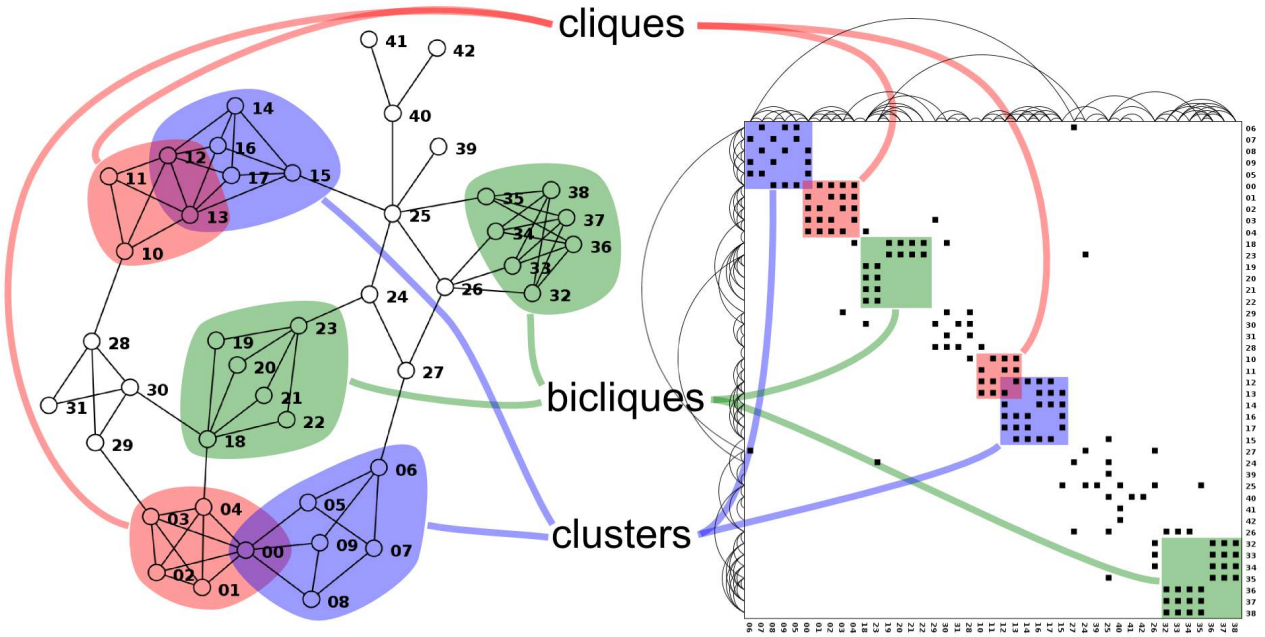
Directed Network



	A	B	C	D
A		■	■	
B			■	
C				
D			■	

Node-link vs matrix comparison

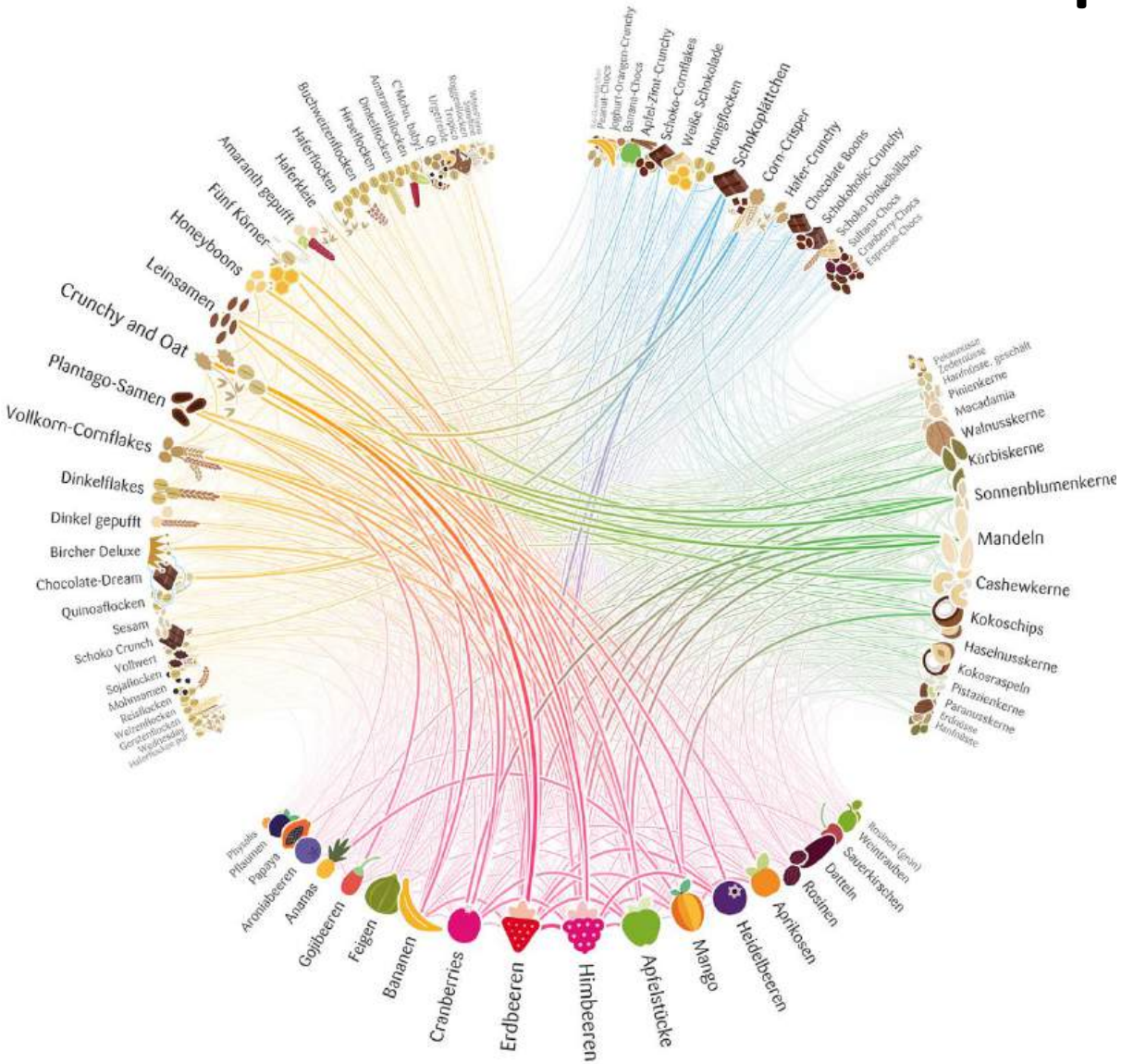
- Node-link diagram strengths
  - Topology understanding, path tracing
  - Intuitive, flexible, no training needed
- Adjacency matrix strengths
  - Focus on edges rather than nodes
  - Layout is straightforward (reordering needed)
  - Predictable space requirements, scalability
  - Some topology tasks are easy with training
- Empirical study:
  - Node-link best for small networks
  - Matrix best for large networks
    - As long as no path tracing

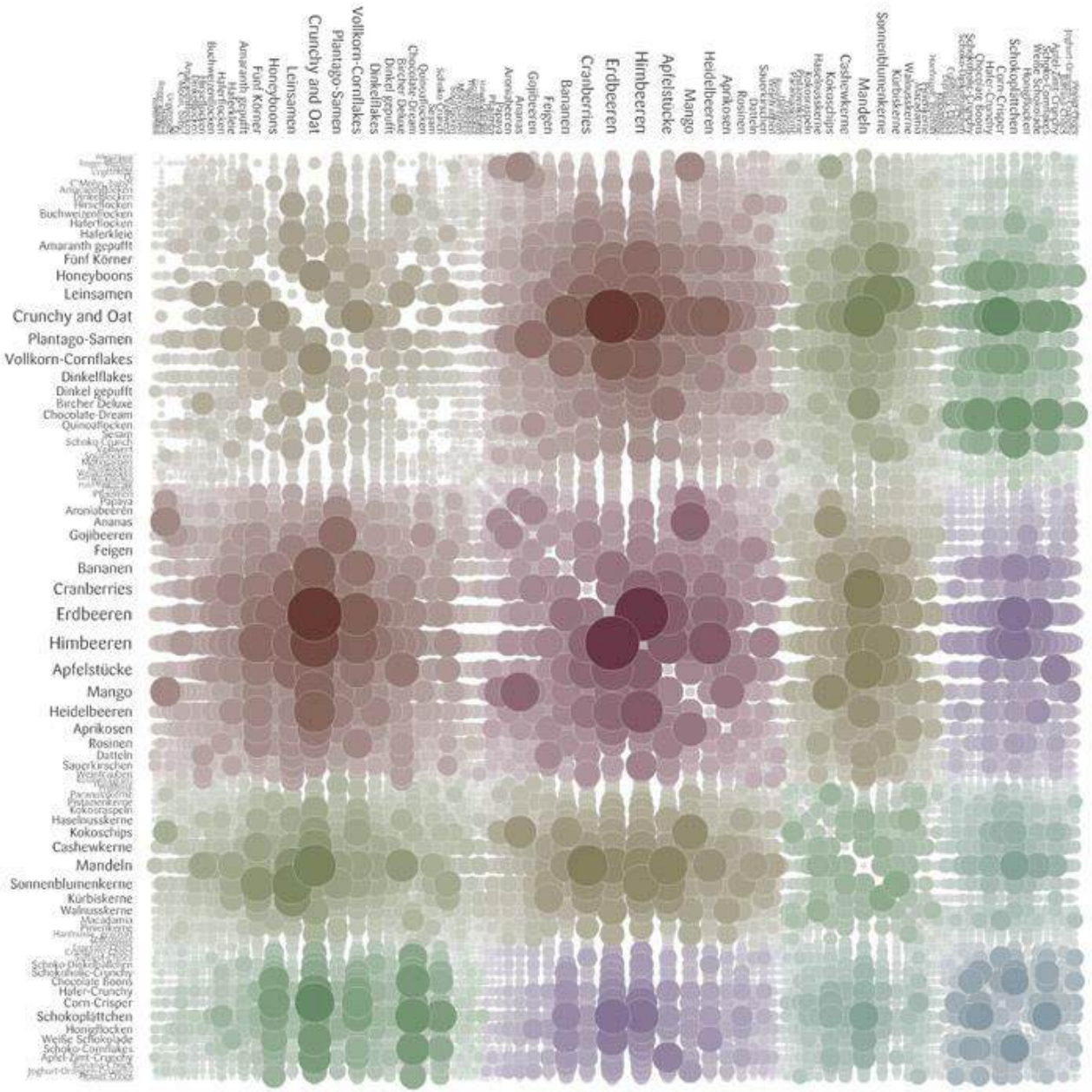


https://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

- 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

Node-link vs matrix comparison

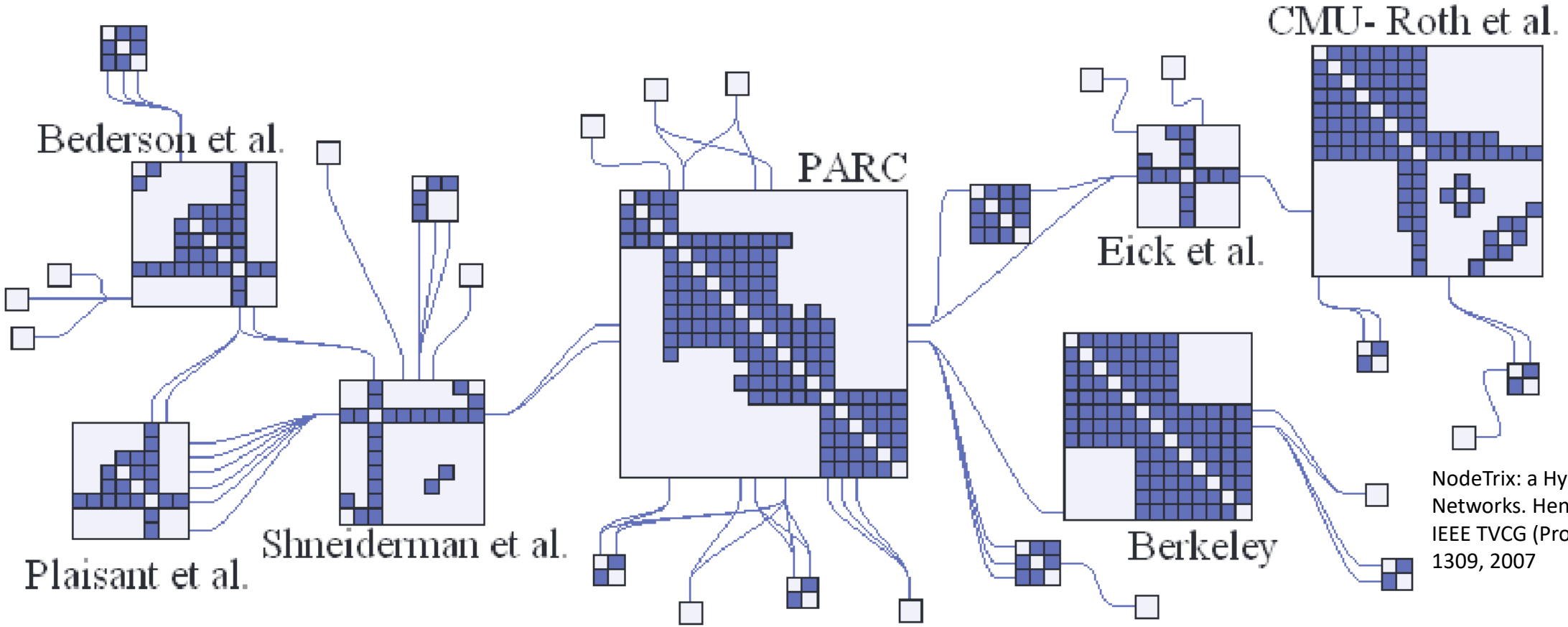




https://truth-and-beauty.net/projects/muesli-ingredient-network

NodeTrix

- What if I need path tracing for a large network?
- NodeTrix: hybrid node-link/matrix
- Captures strengths of both
- https://www.youtube.com/watch?v=7G3MxyOcHKQ

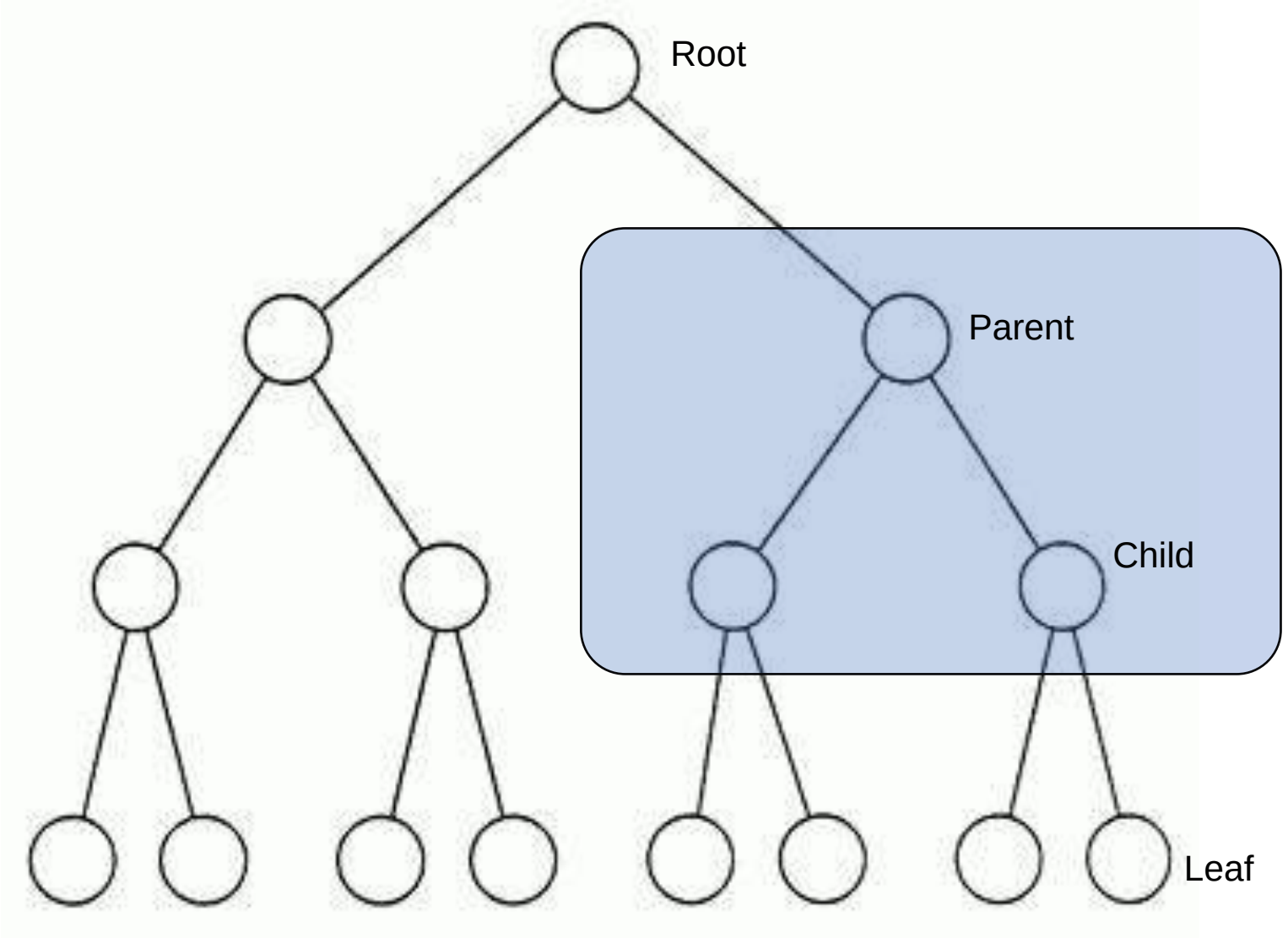


NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007

Trees

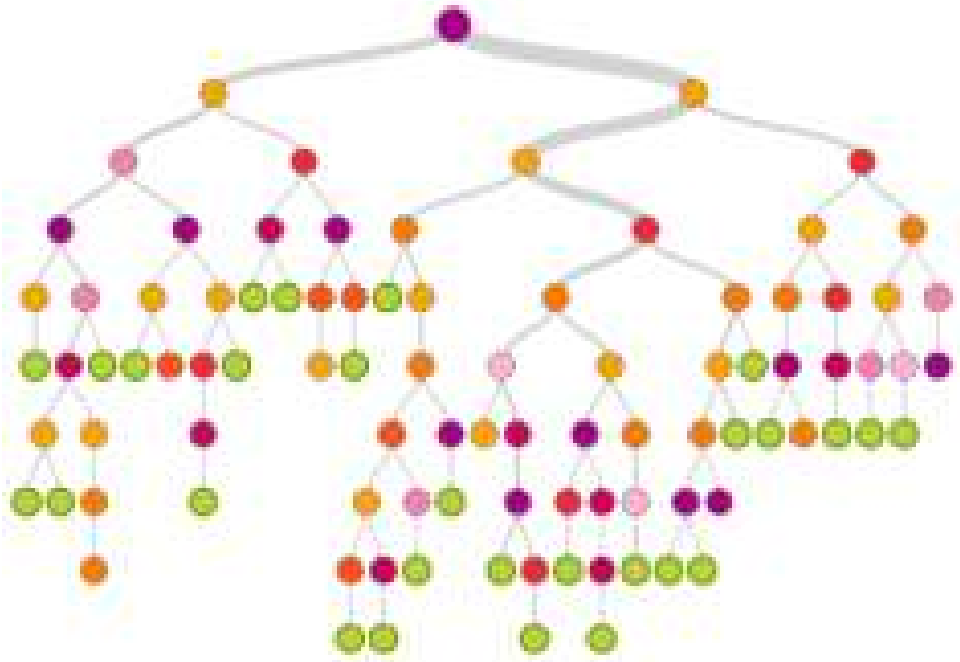
Node-link trees

- Applications of trees:
  - File structure
  - Evolutionary tree
  - Language structure
  - More?



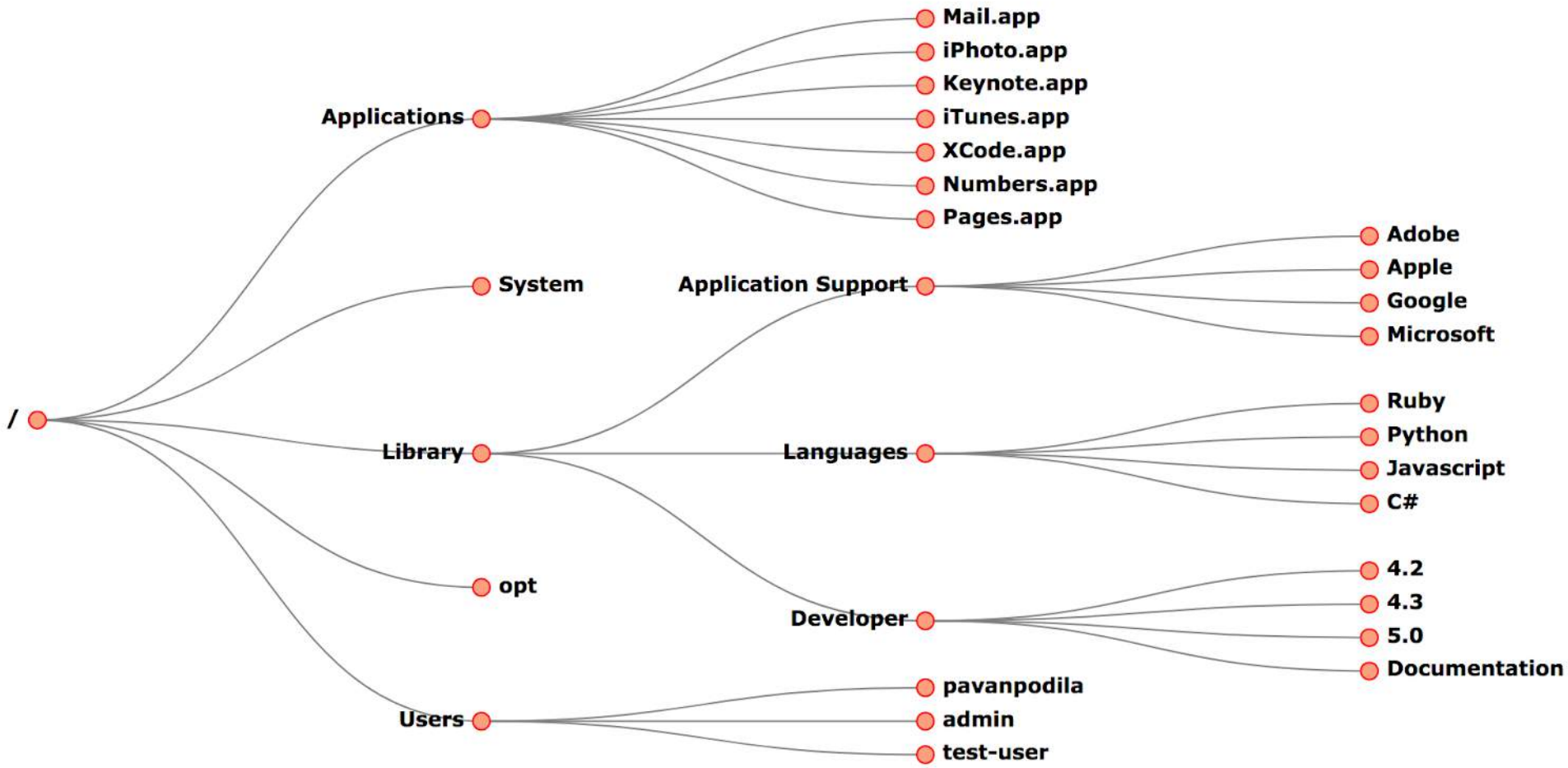
Visualizing trees

Node-link



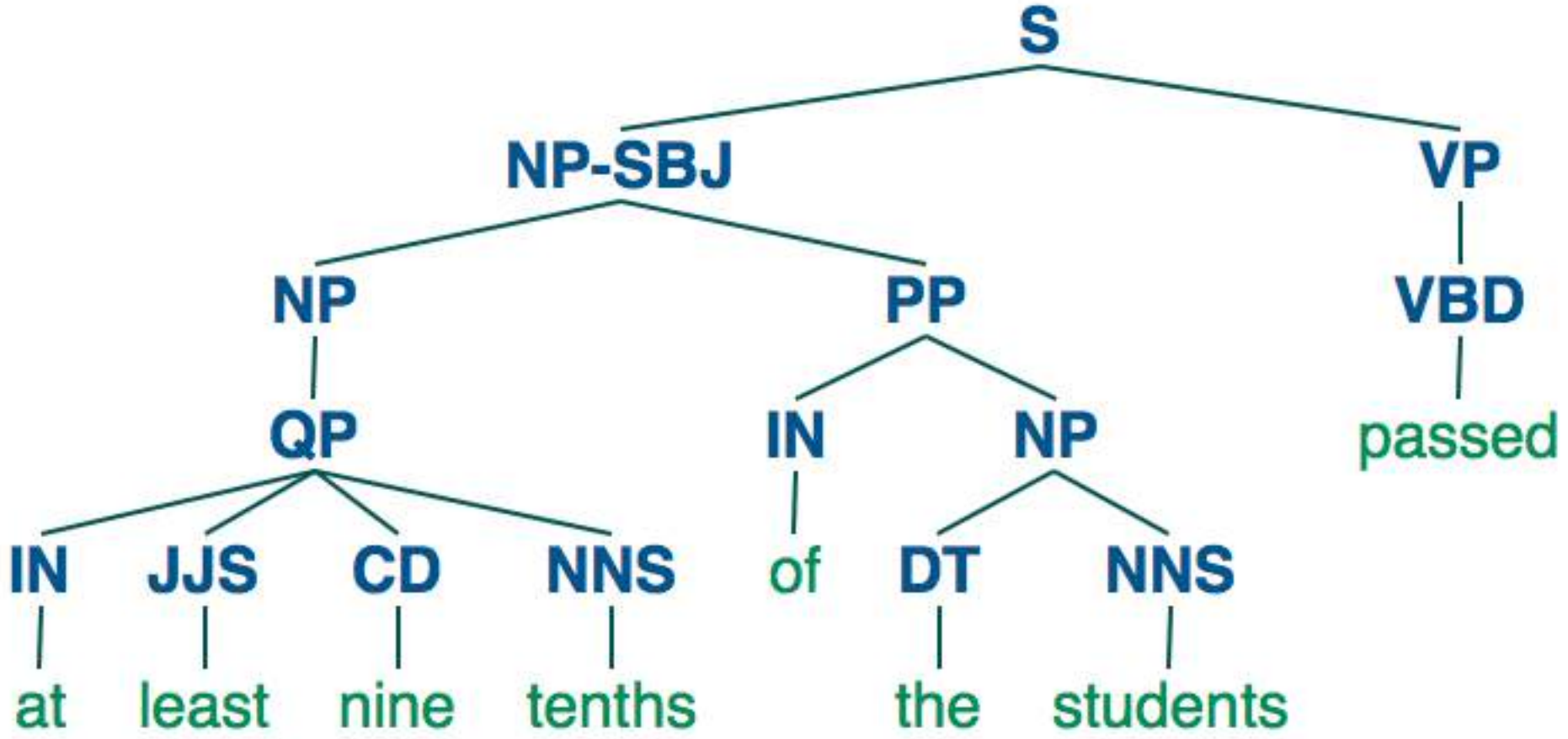
Containment

Node-link trees: file system



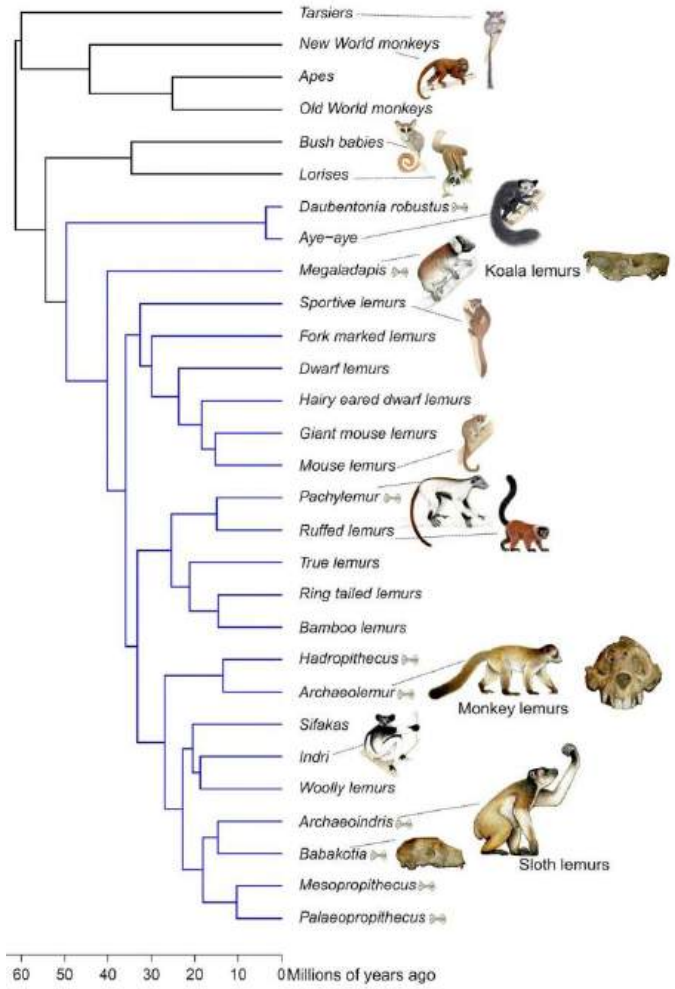
Source: https://blog.pixelingene.com/demos/d3_tree/

Node-link trees: language parser

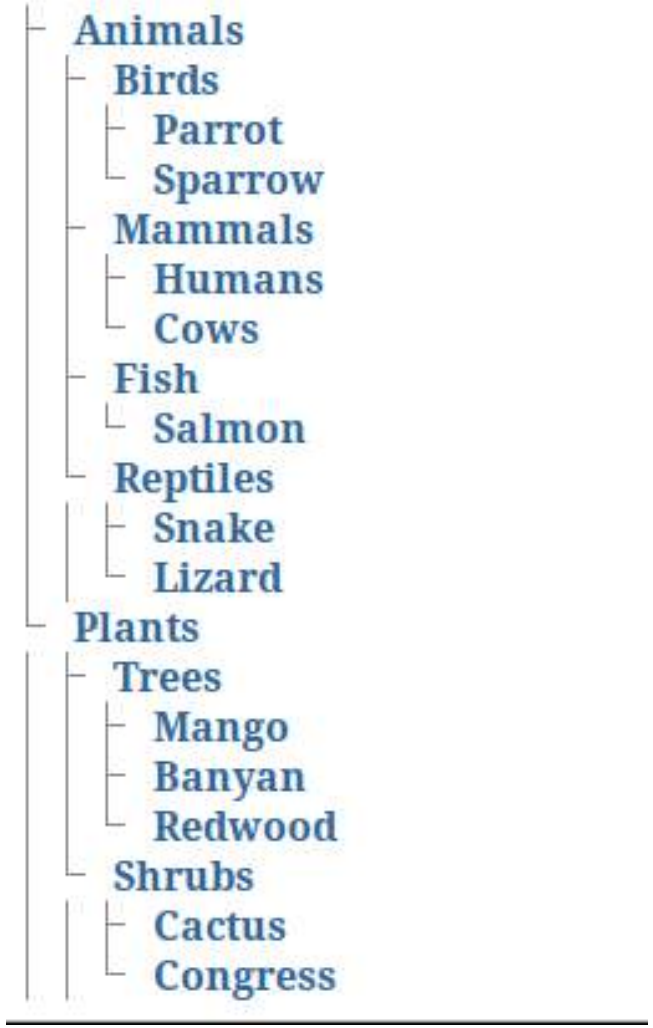


Source: https://stackoverflow.com/questions/23429117/saving-nltk-drawn-parse-tree-to-image-file

Node-link trees: species ancestry



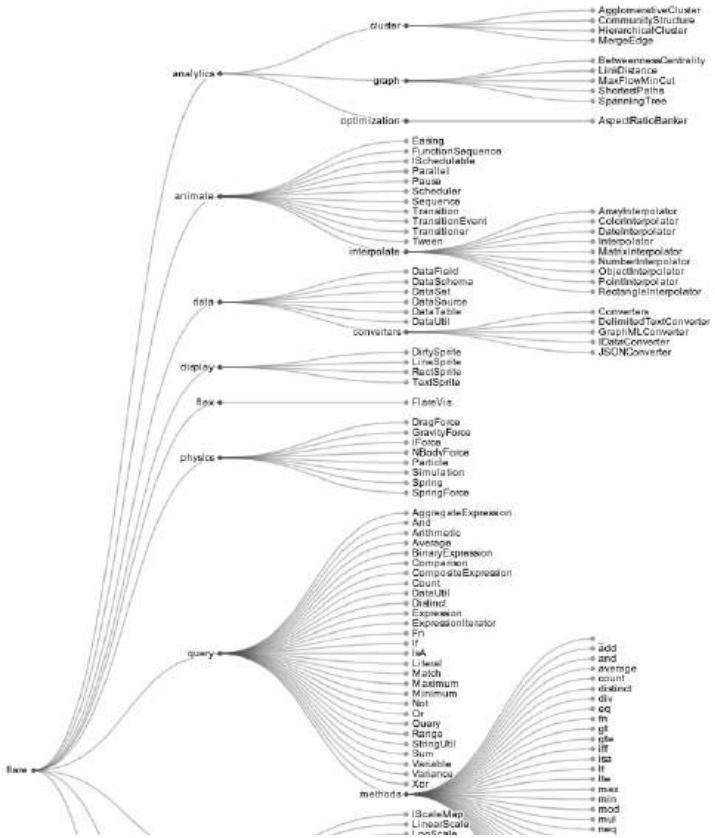
Source: http://novataxa.blogspot.com/2016/04/



Source: https://stackoverflow.com/questions/14922247/how-to-get-a-tree-in-html-using-pure-css

Node-link trees

- Special type of network
- Reingold-Tilford algorithm
  - Tidy drawings of trees
    - Exploit parent-child structure
  - Allocate space: compact but without overlap
    - Rectilinear and radial variants
- Algorithm writeup: https://llimllib.github.io/pymag-trees/



https://observablehq.com/@d3/tree/2

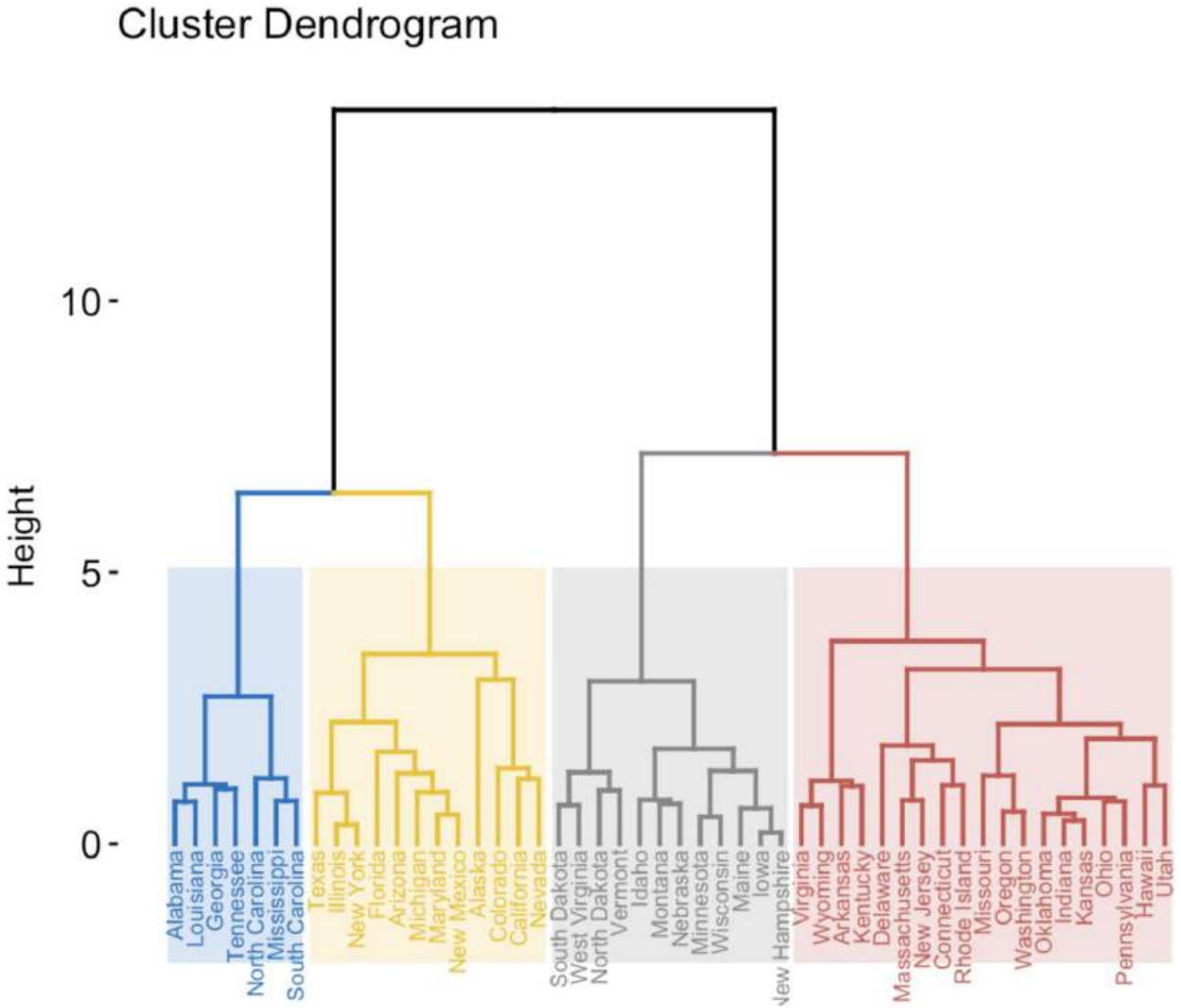
https://observablehq.com/@d3/radial-tree/2

Radial node-link trees

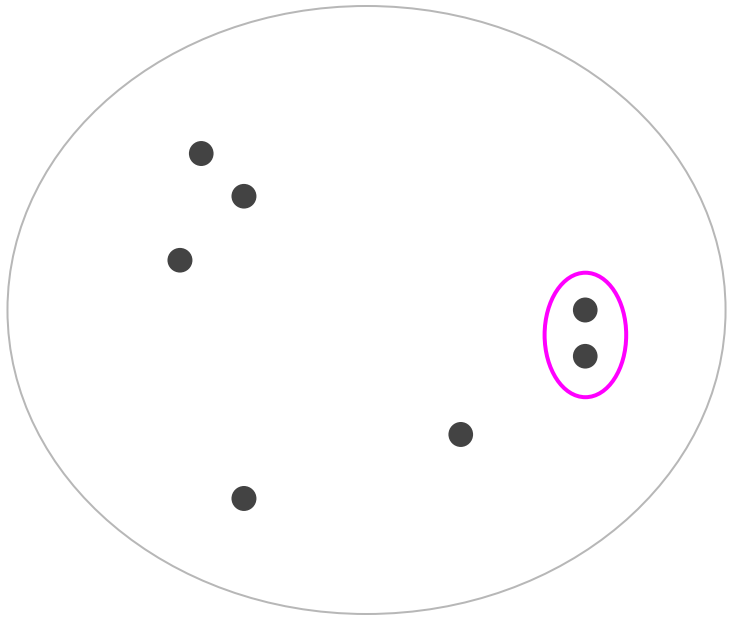
- Data:
  - Tree
- Encoding:
  - Point node marks
  - Link connection marks
  - Radial axis orientation
    - Angular proximity → siblings
    - Distance from center → depth in tree
- Tasks:
  - Understand topology, follow paths
- Scalability:
  - 1k – 10k nodes (with/without labels)

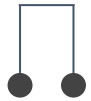
Dendrogram

- Represents **hierarchical clusters**
  - Data processing algorithm to organize objects into a hierarchy according to a similarity metric defined between the objects.
- Binary tree!

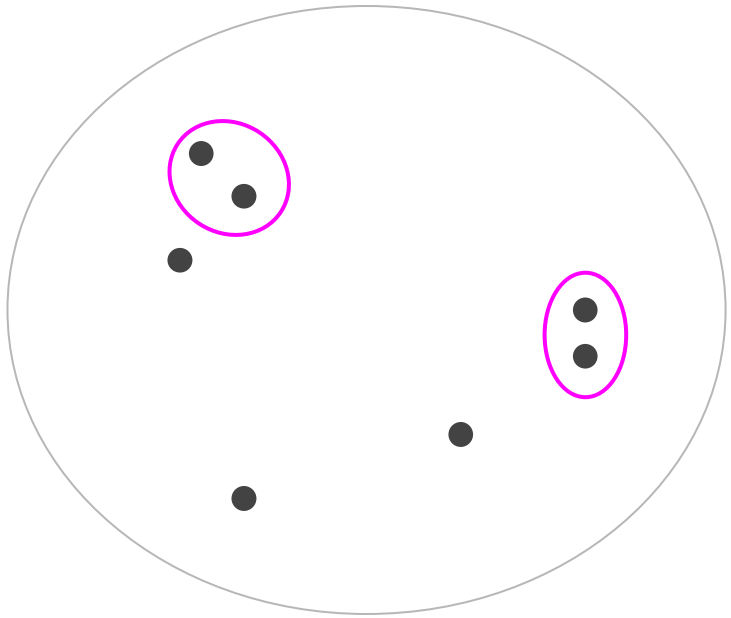


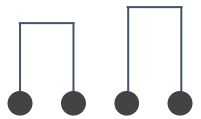
Hierarchical Clustering



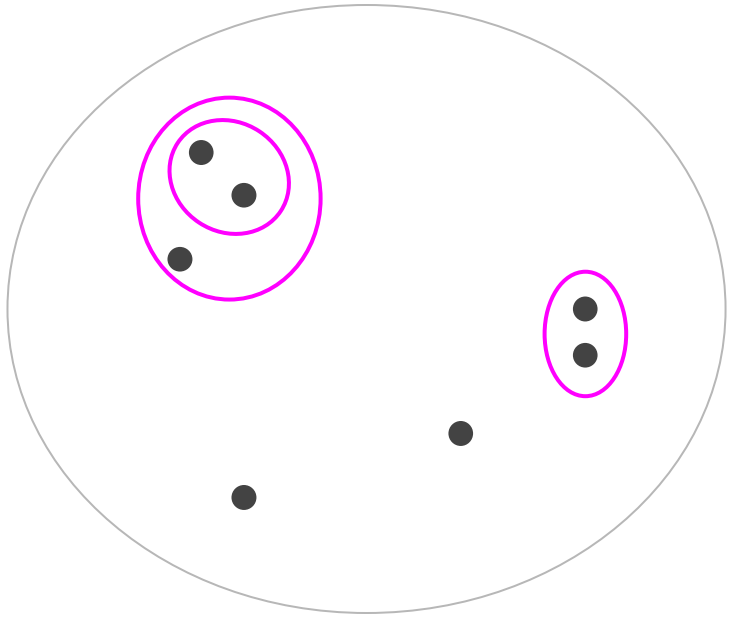


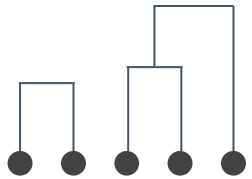
Hierarchical Clustering



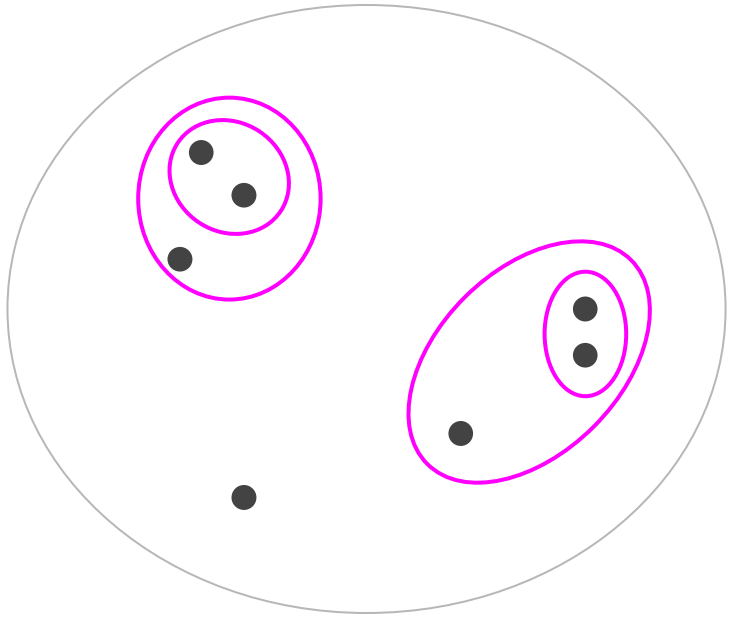


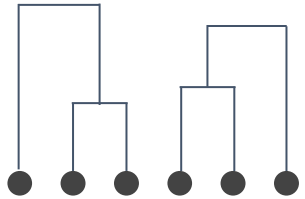
Hierarchical Clustering



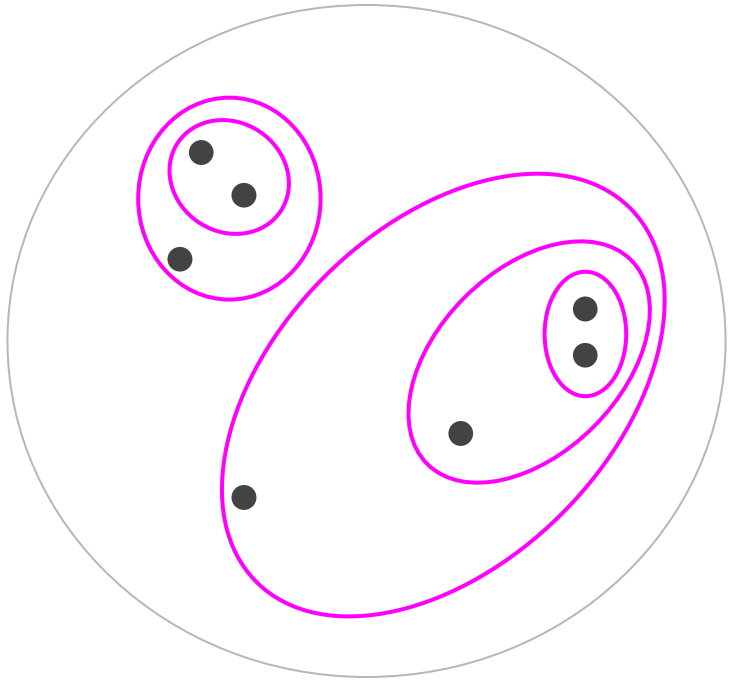


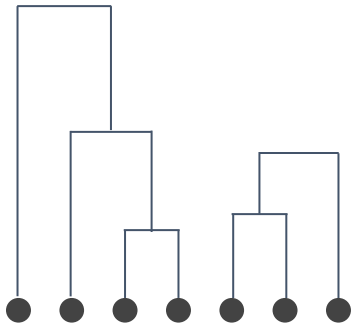
Hierarchical Clustering



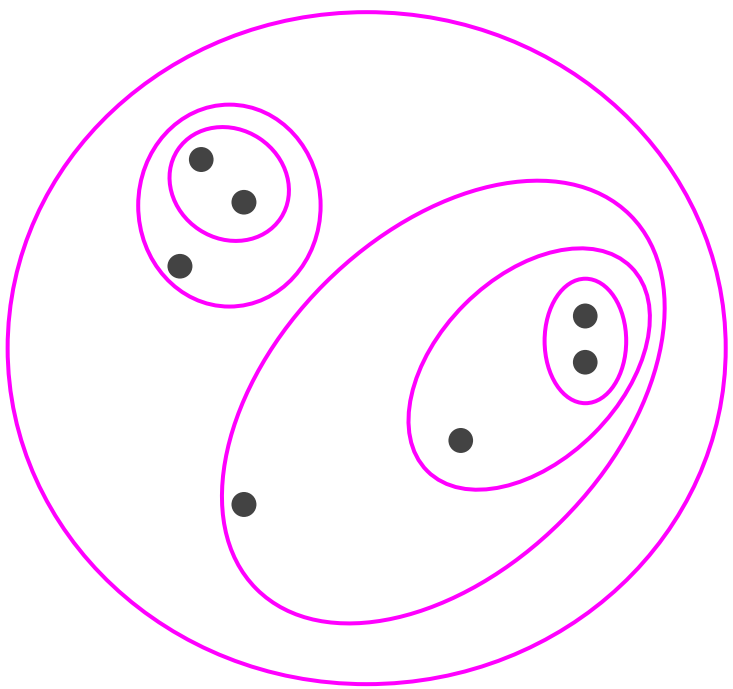


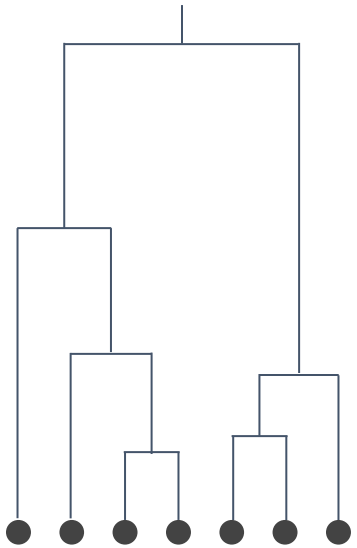
Hierarchical Clustering





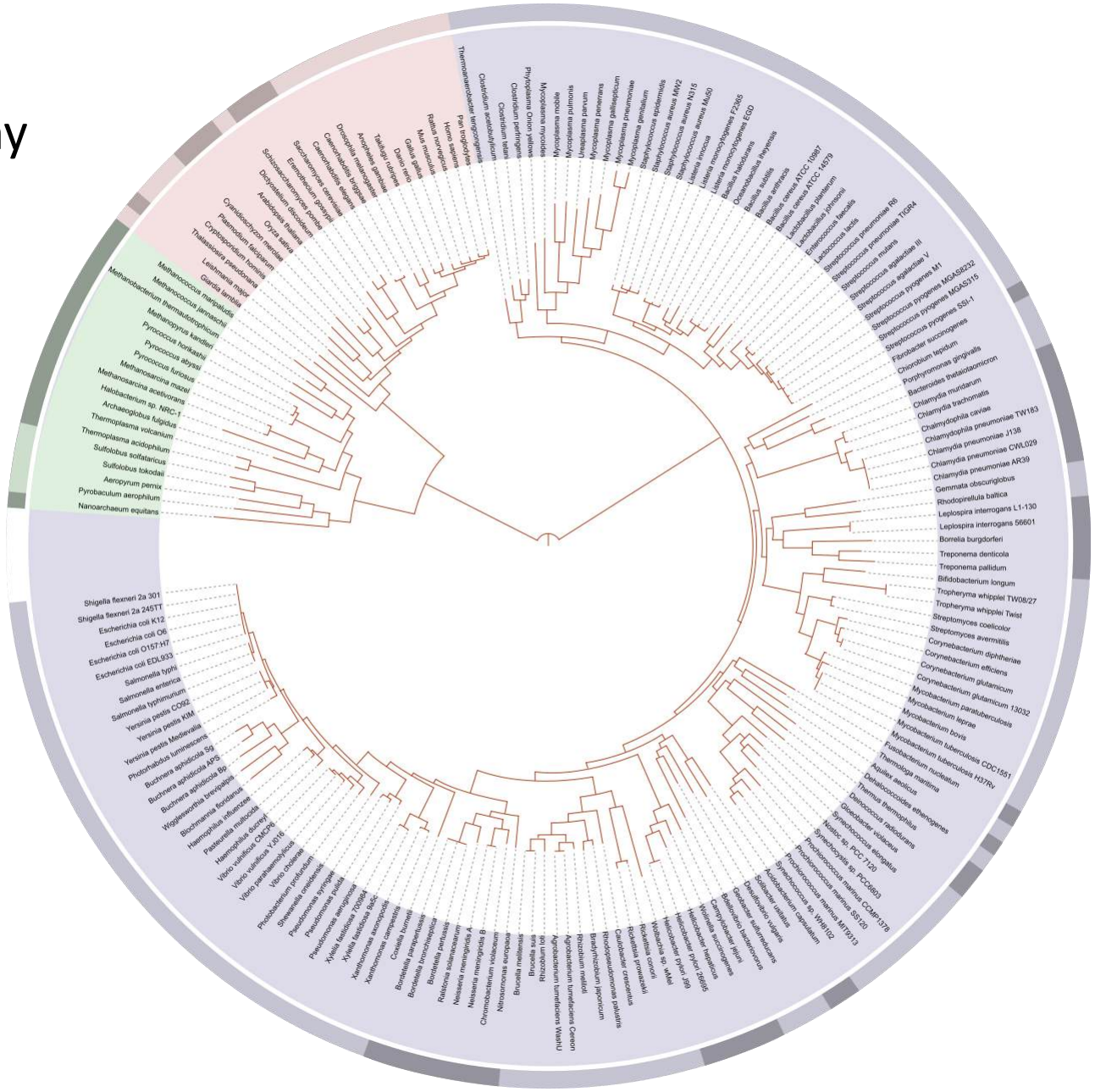
Hierarchical Clustering





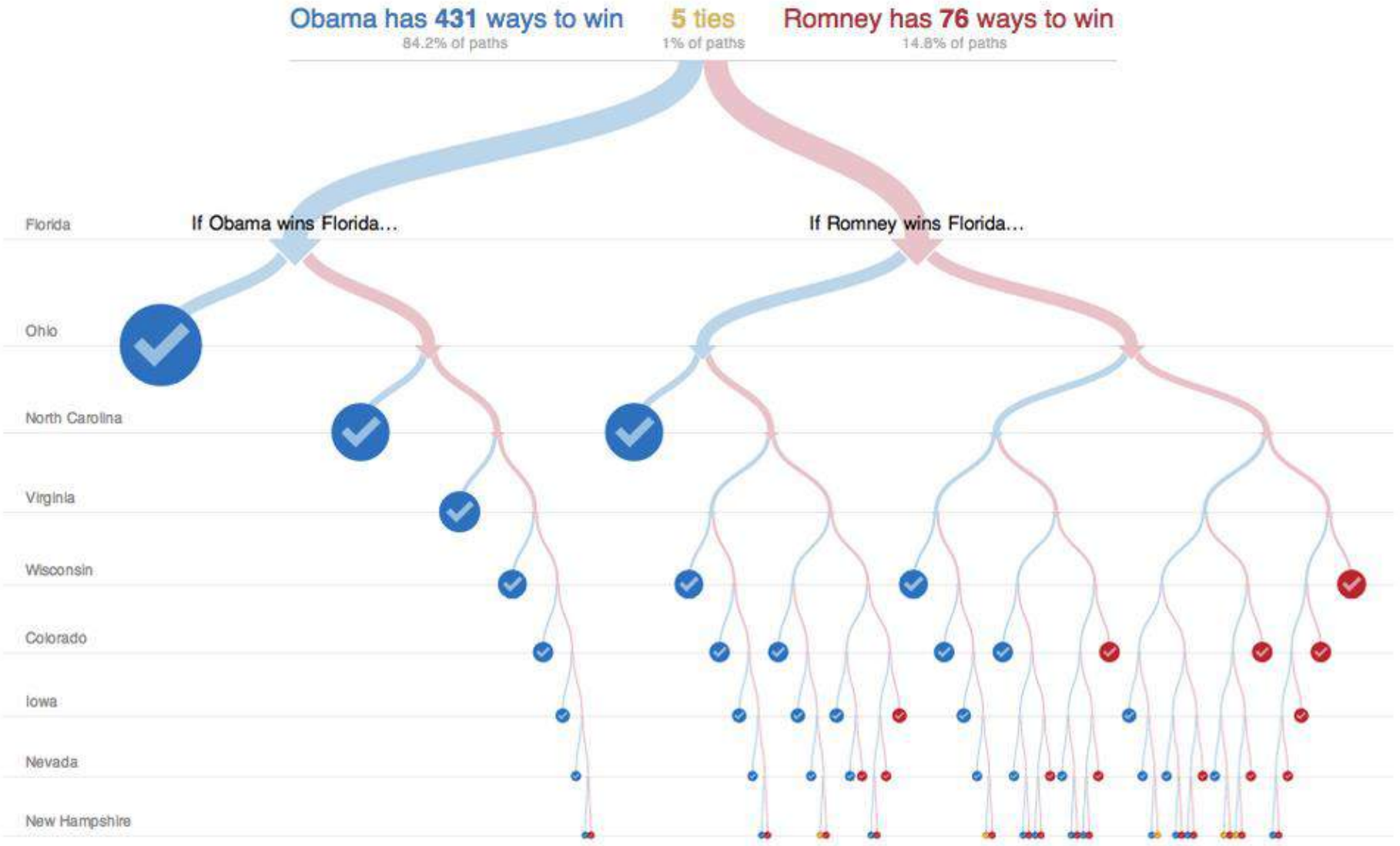
Hierarchical Clustering

- Useful to group together and visualize any set of complex objects on top of which a distance function can be defined
  - Eg: Phylogenetic Trees Computed from “Genetic Distance” from multiple sequence alignments.



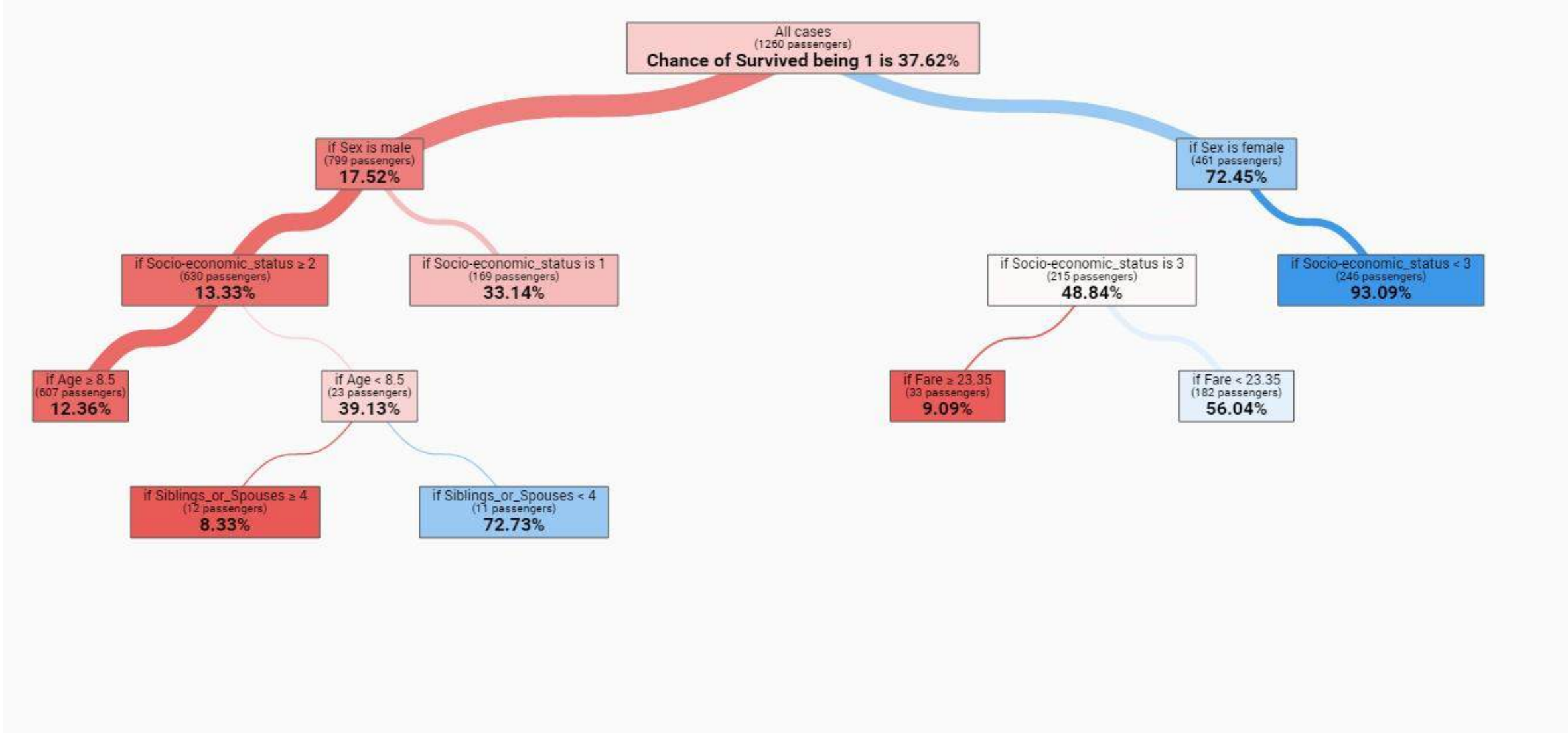
Decision trees

- Visualize different combinations of decisions/events/outcomes



Decision trees

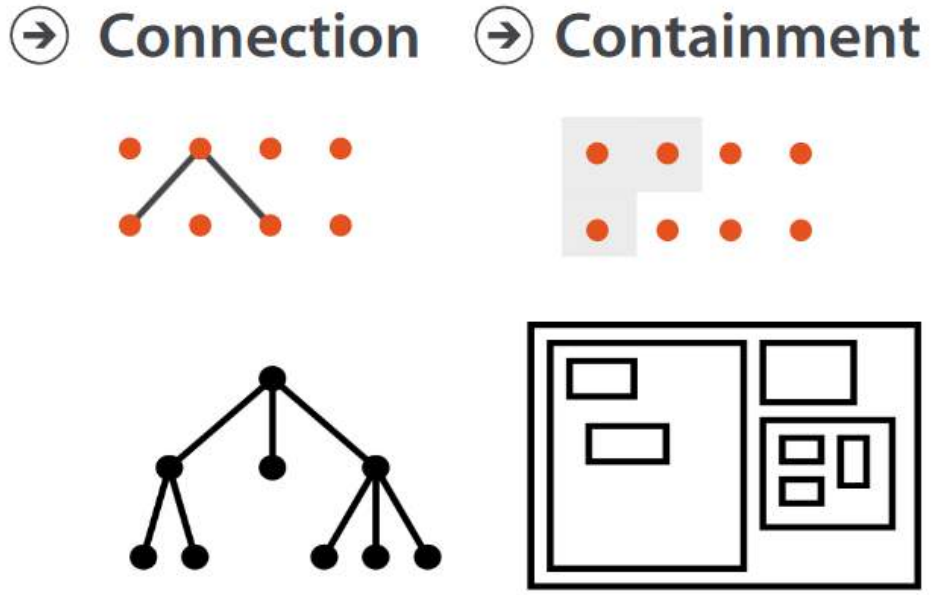
- Visualize different combinations of decisions/events/outcomes
- Can be built automatically from data

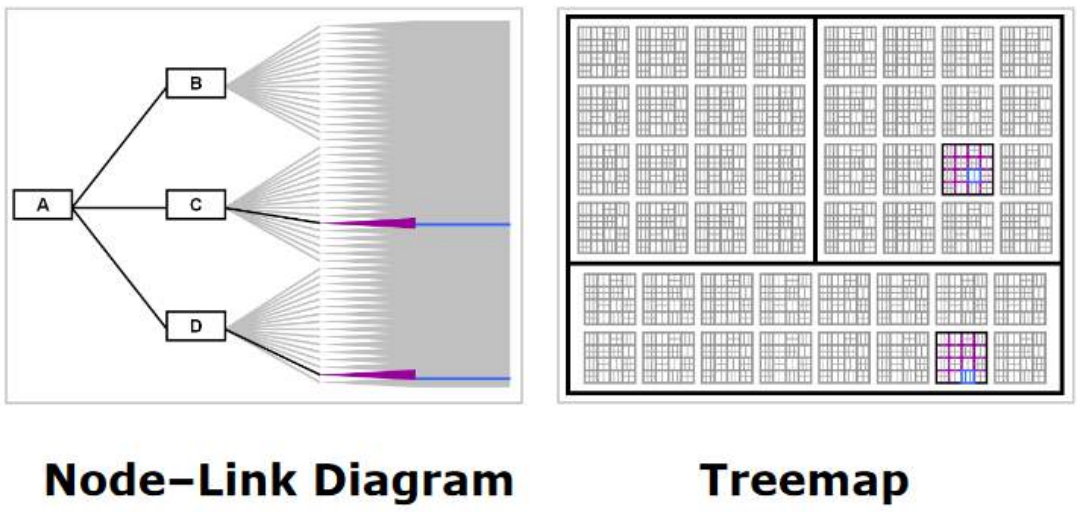


Alternative: link marks w. connection & containment

- What we've seen so far:
  - 1D case: connection (lines)
    - Eg: all node-link diagrams
    - Emphasizes topology, path tracing
    - Networks and trees

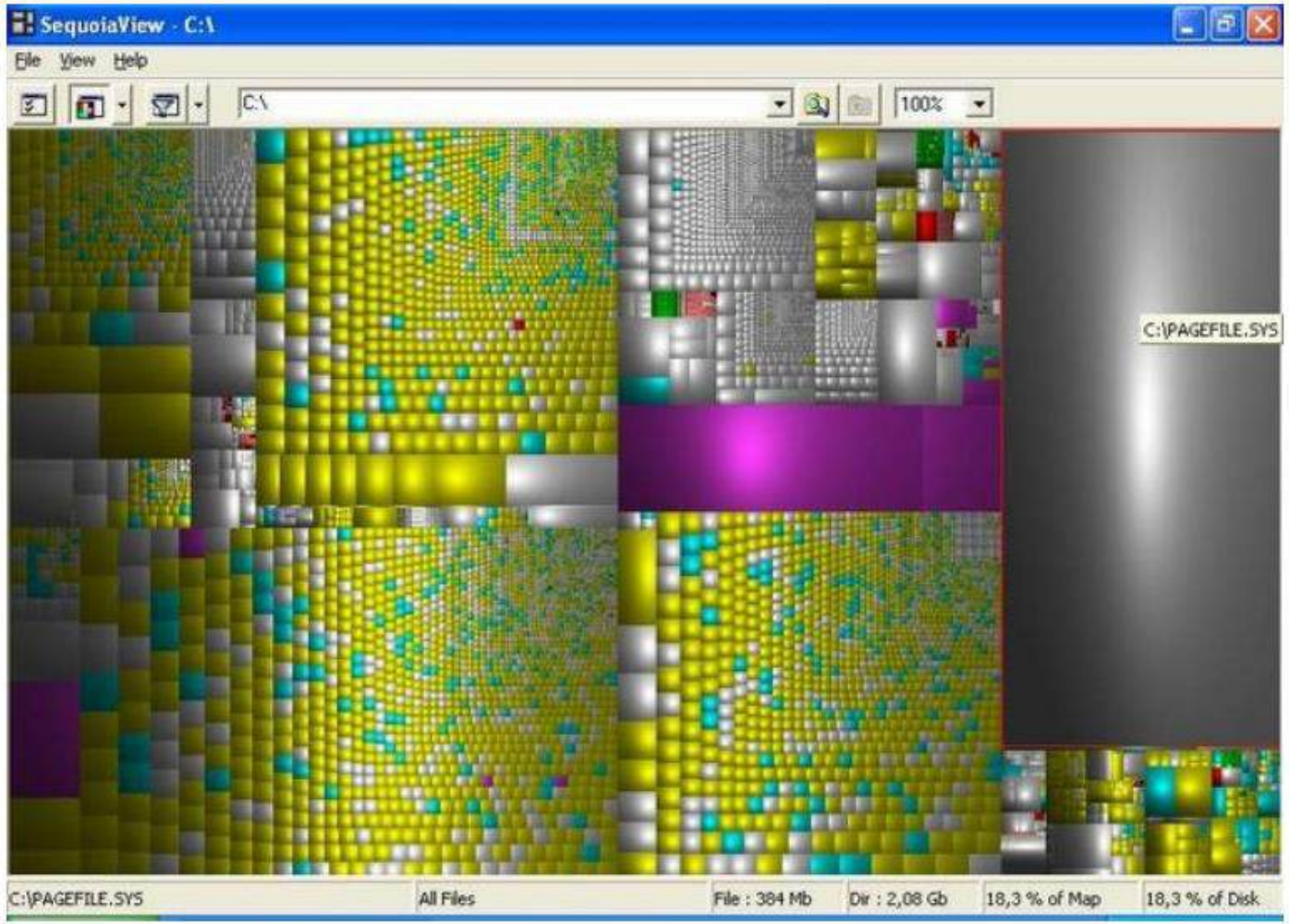
- Alternative: marks as links (rather than nodes)
  - Common case in network drawings
  - 2D case: containment (areas)
    - Eg: treemaps
    - Emphasizes attribute values at leaves (size encoding)
    - Only works for trees





Treemap

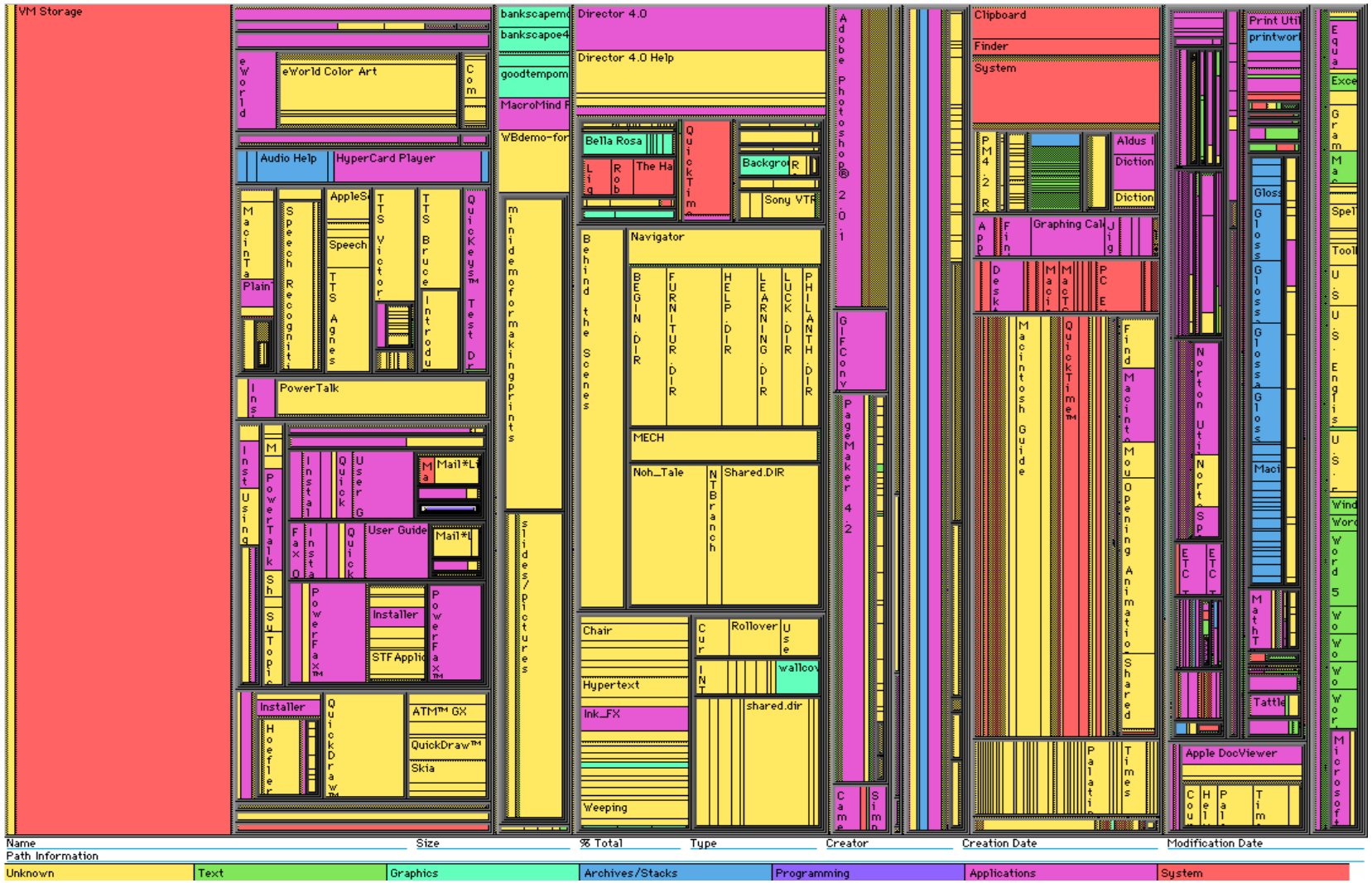
- Data:
  - Tree
  - 1 quant. attribute at leaf nodes
- Encoding
  - Area containment marks for hierarchical structure
  - Rectilinear orientation
  - Size encodes quant. Attributes
- Tasks:
  - Query attribute at leaf nodes
  - Eg: disk space usage
- Scalability
  - 1M leaf nodes



Cushion Treemaps. van Wijk and van de Wetering. Proc. Symp. InfoVis 1999, 73-78.
https://sequoiaview.win.tue.nl/

Treemap

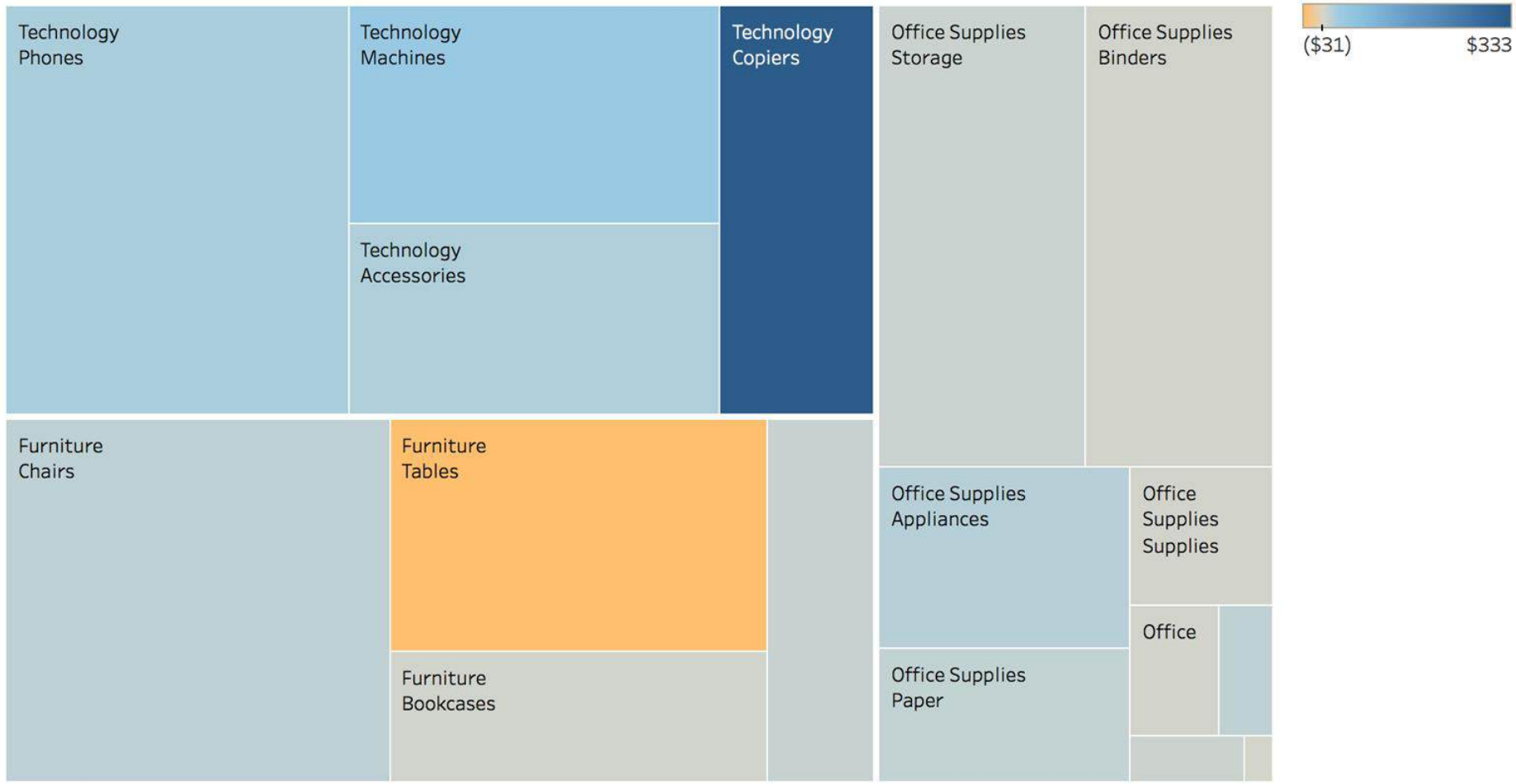
- What information can be visualized with a tree map?
  - Area: Quantity
  - Color: Quantity/Category
  - Hierarchy: Nesting



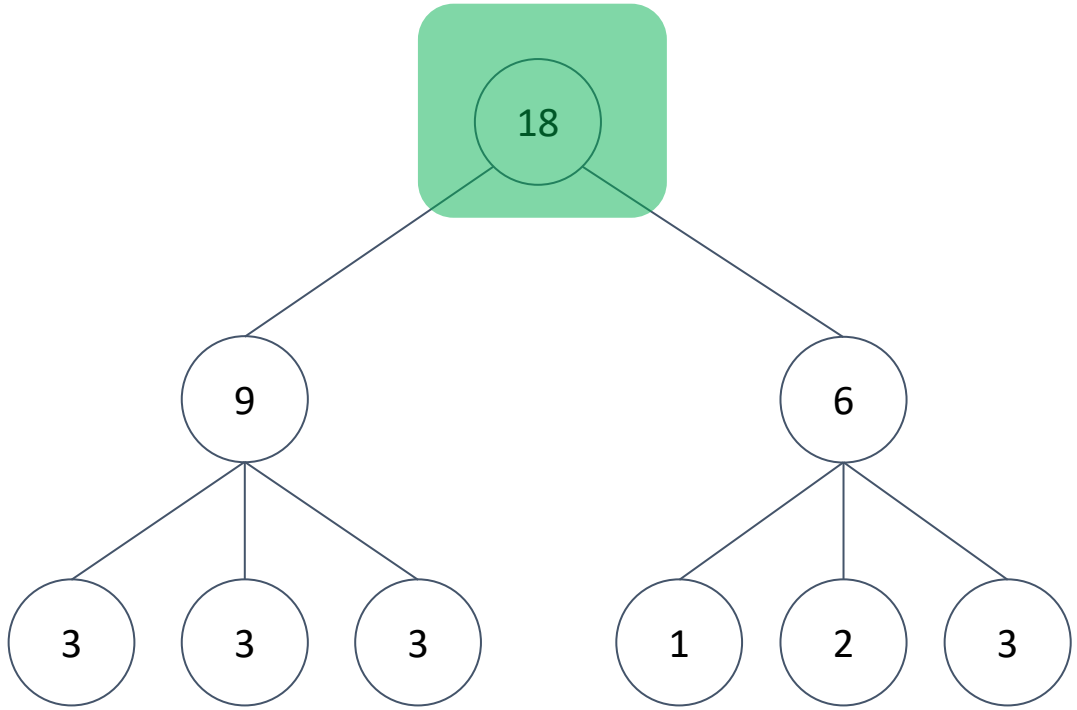
Source: http://www.cs.umd.edu/hcil/treemap-history/

Treemap

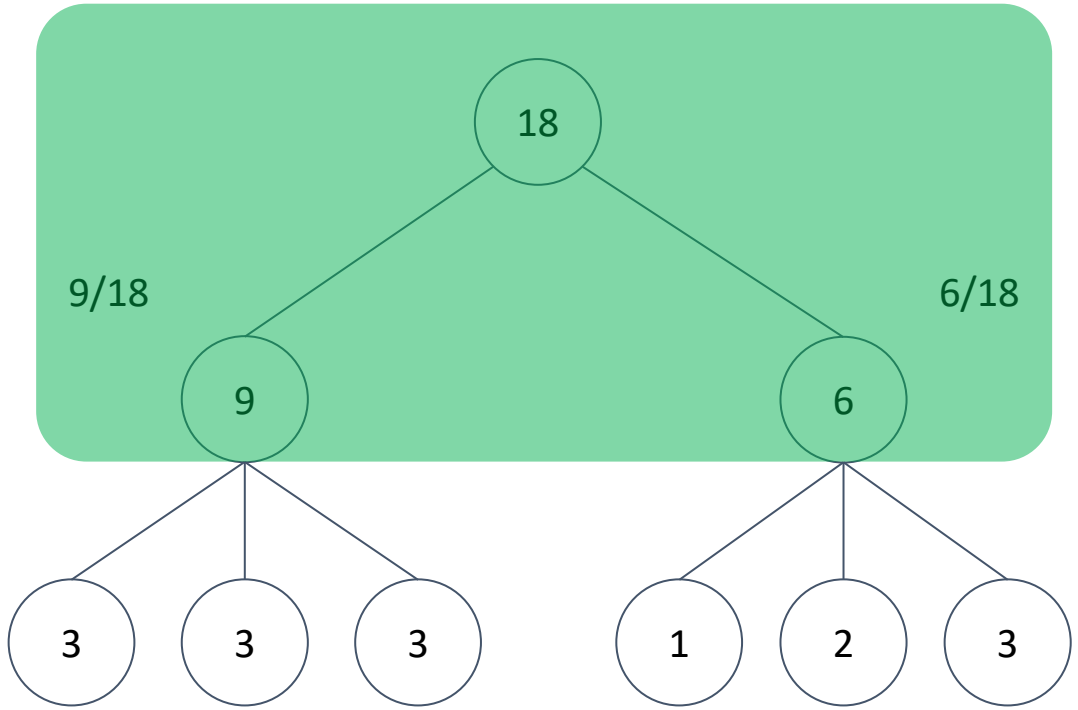
- What information can be visualized with a tree map?
  - Area: Quantity
  - Color: Quantity/Category
  - Hierarchy: Nesting

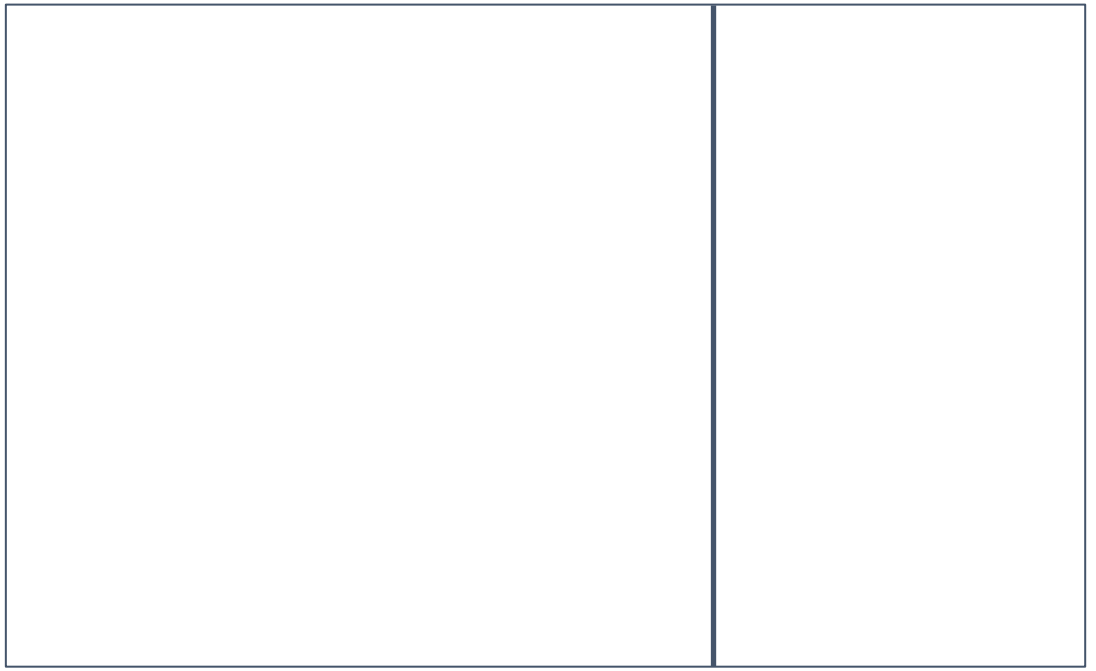


Treemap construction

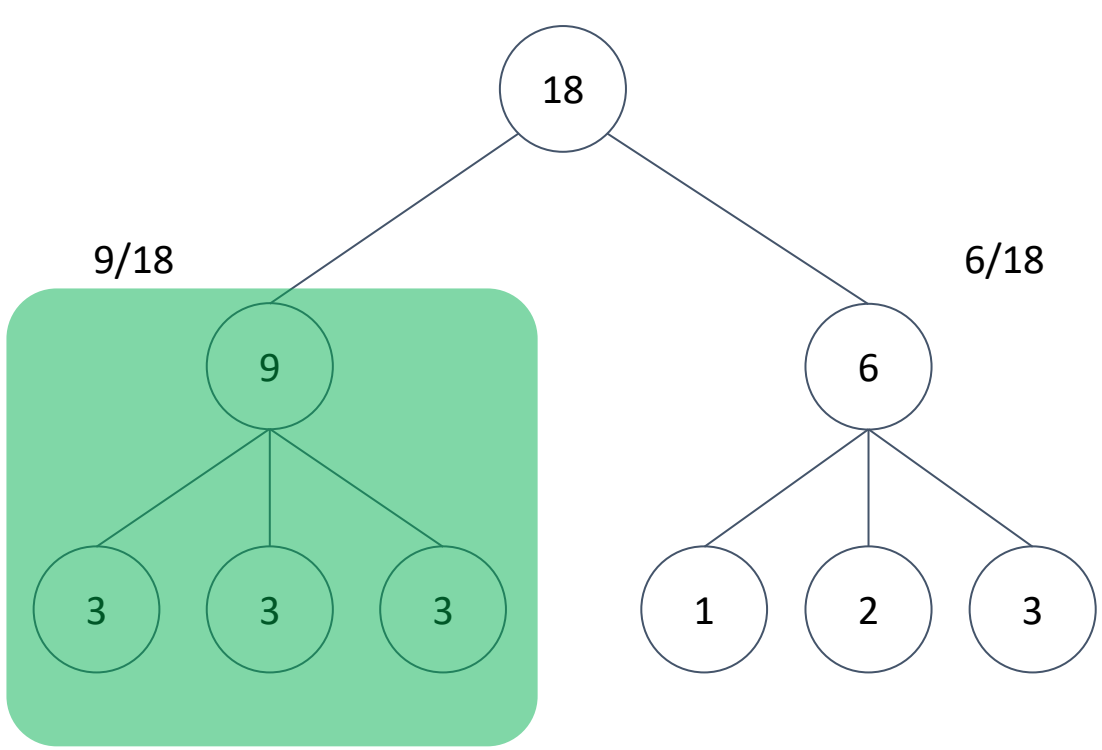


Treemap construction

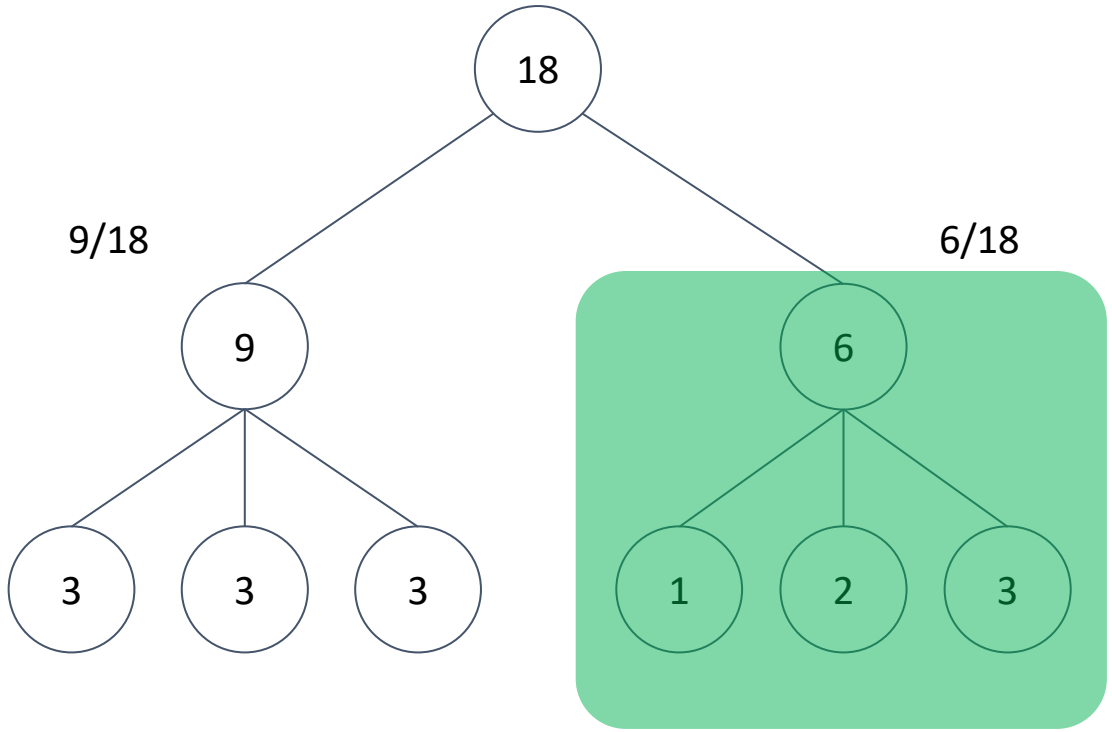


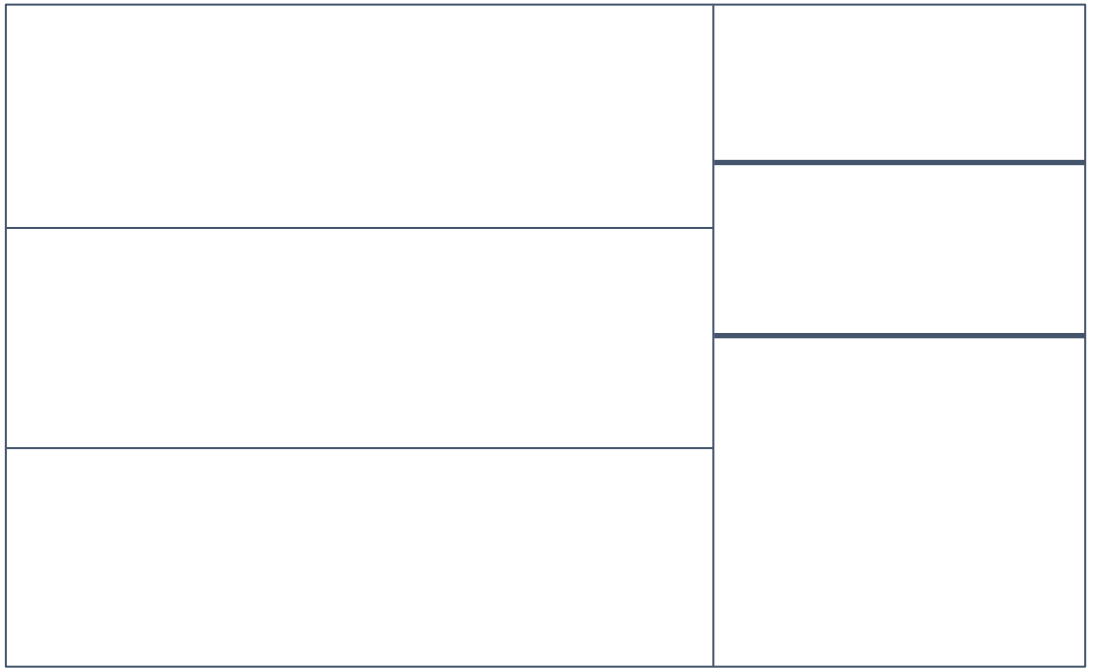


Treemap construction



Treemap construction

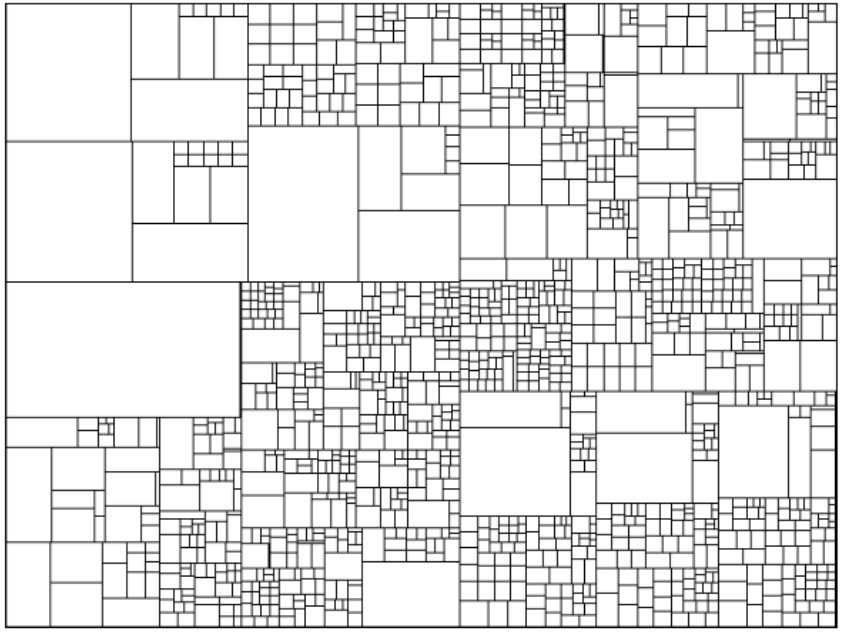


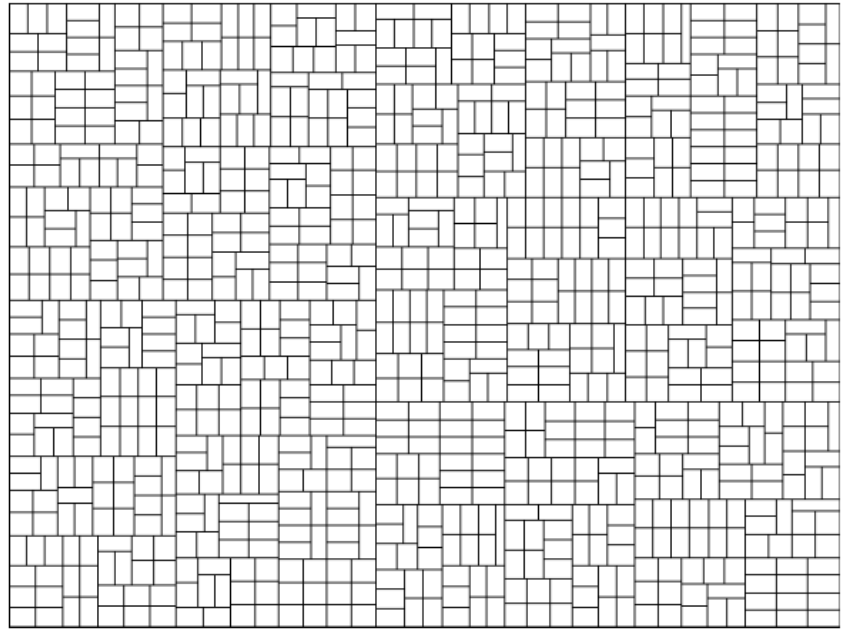


Treemap drawback

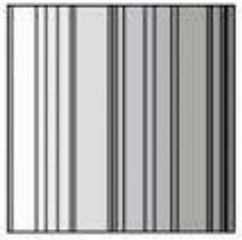
- When the rectangles have very different aspect ratios (proportion of height vs. width), comparing areas gets harder (especially with thin elongated rectangles).

- Solution: Squarified Tree Maps.

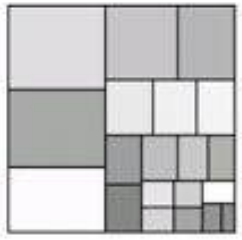




SliceAndDice



Squarified



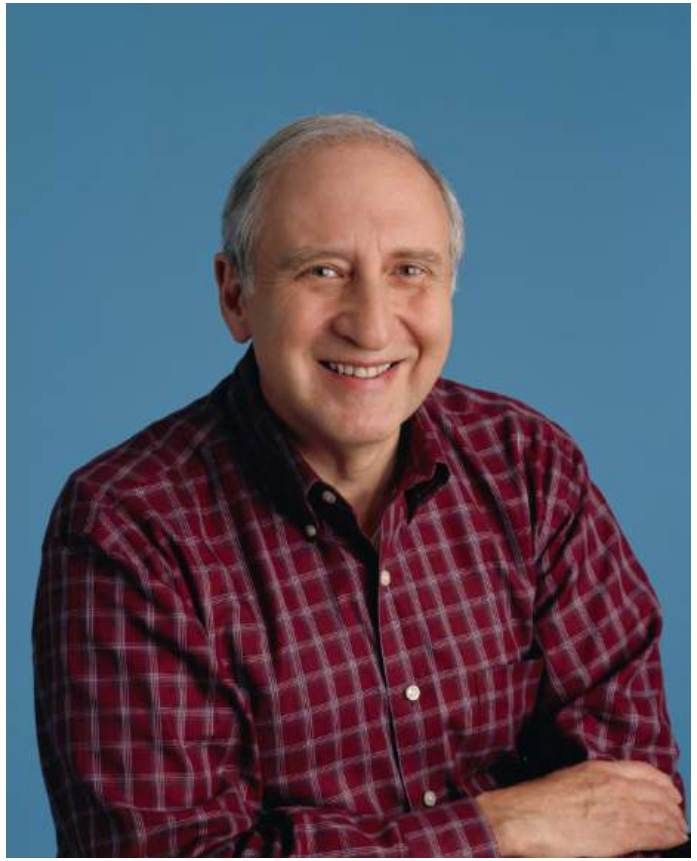
Strip

Treemap summary

- Advantages:
  - Node visibility
  - No overlapping marks
  - Supports size and color channels

- Disadvantages:
  - Size is not the most accurate channel
  - Structures can be hard to discern

Treemap lore



"During 1990, in response to the common problem of a filled hard disk, I became obsessed with the idea of producing a compact visualization of directory tree structures." - Ben Shneiderman

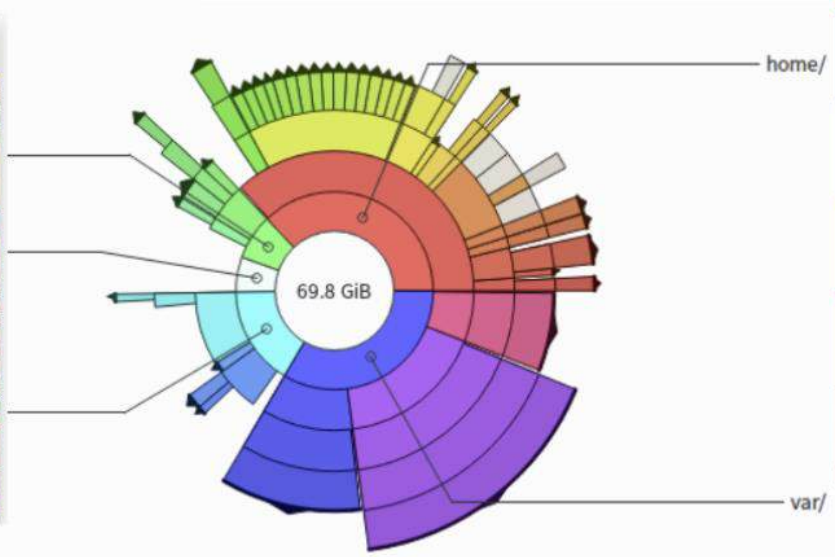
http://www.cs.umd.edu/hcil/treemap-history/

Alternative: implicit tree layout

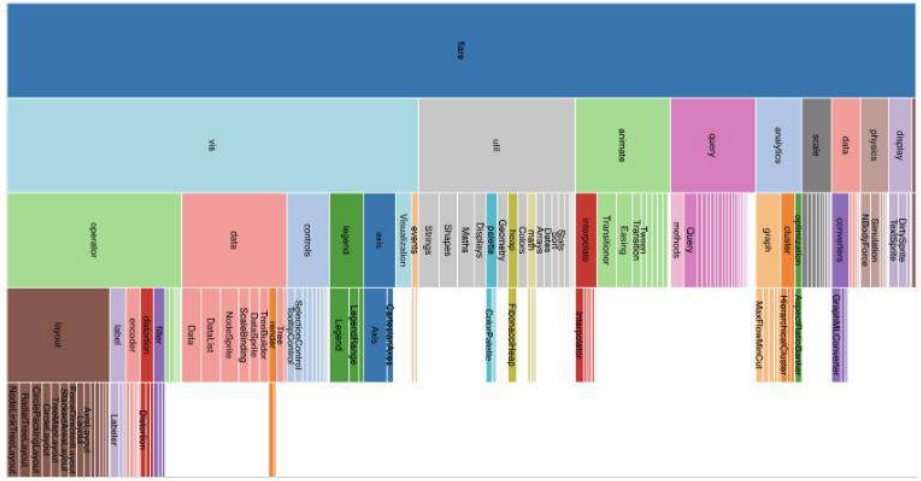
- Alternative to node-link trees (connection) and treemaps (containment)
  - Show parent-child relationship only through relative positions

Treemap
- Containment
- Only leaves visible

Sunburst
- Position (radial)
- Inner nodes & leaves visible

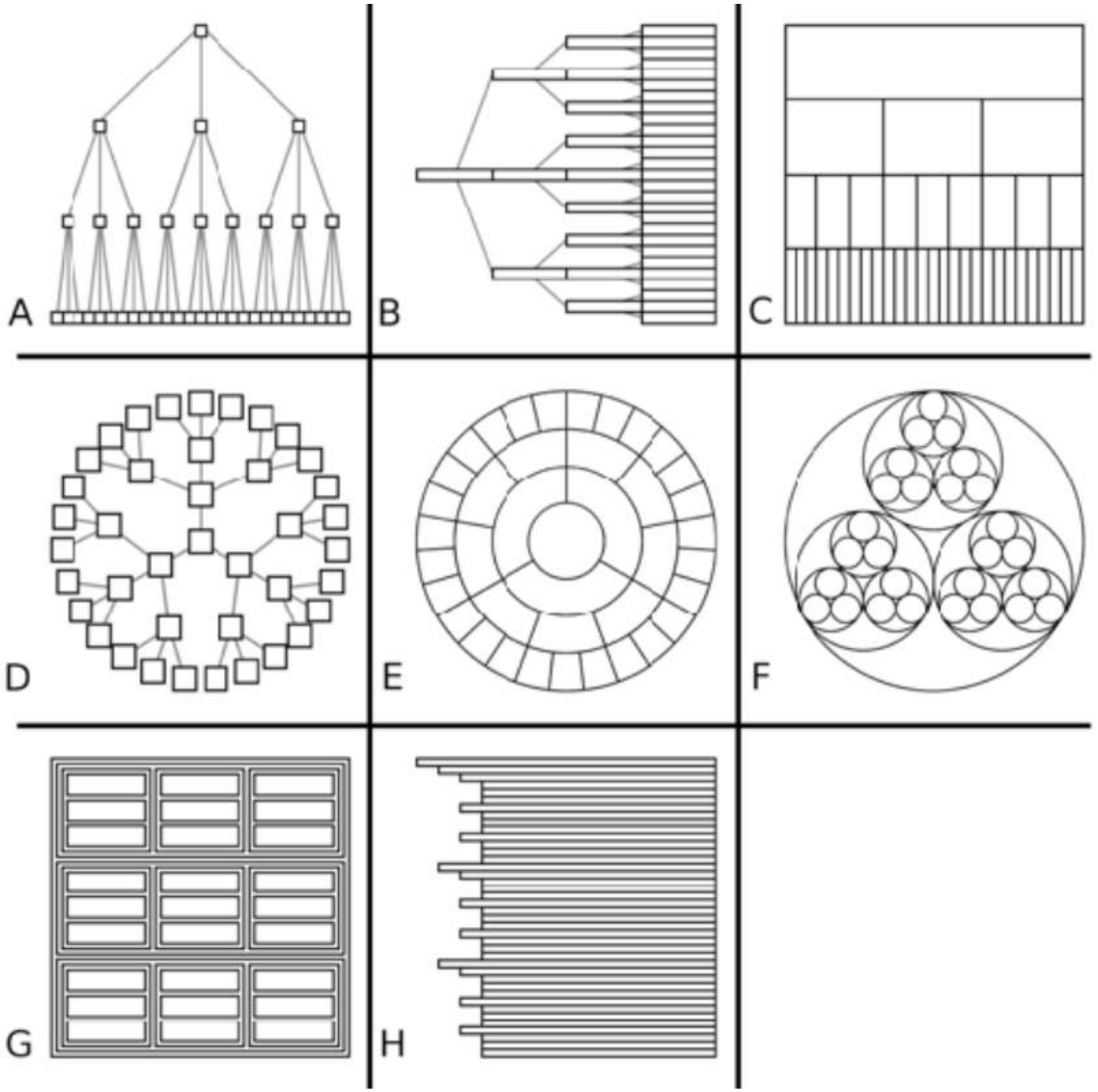


Icicle plot
- Position (rectilinear)
- Inner nodes & leaves visible



Tree viz considerations & comparisons

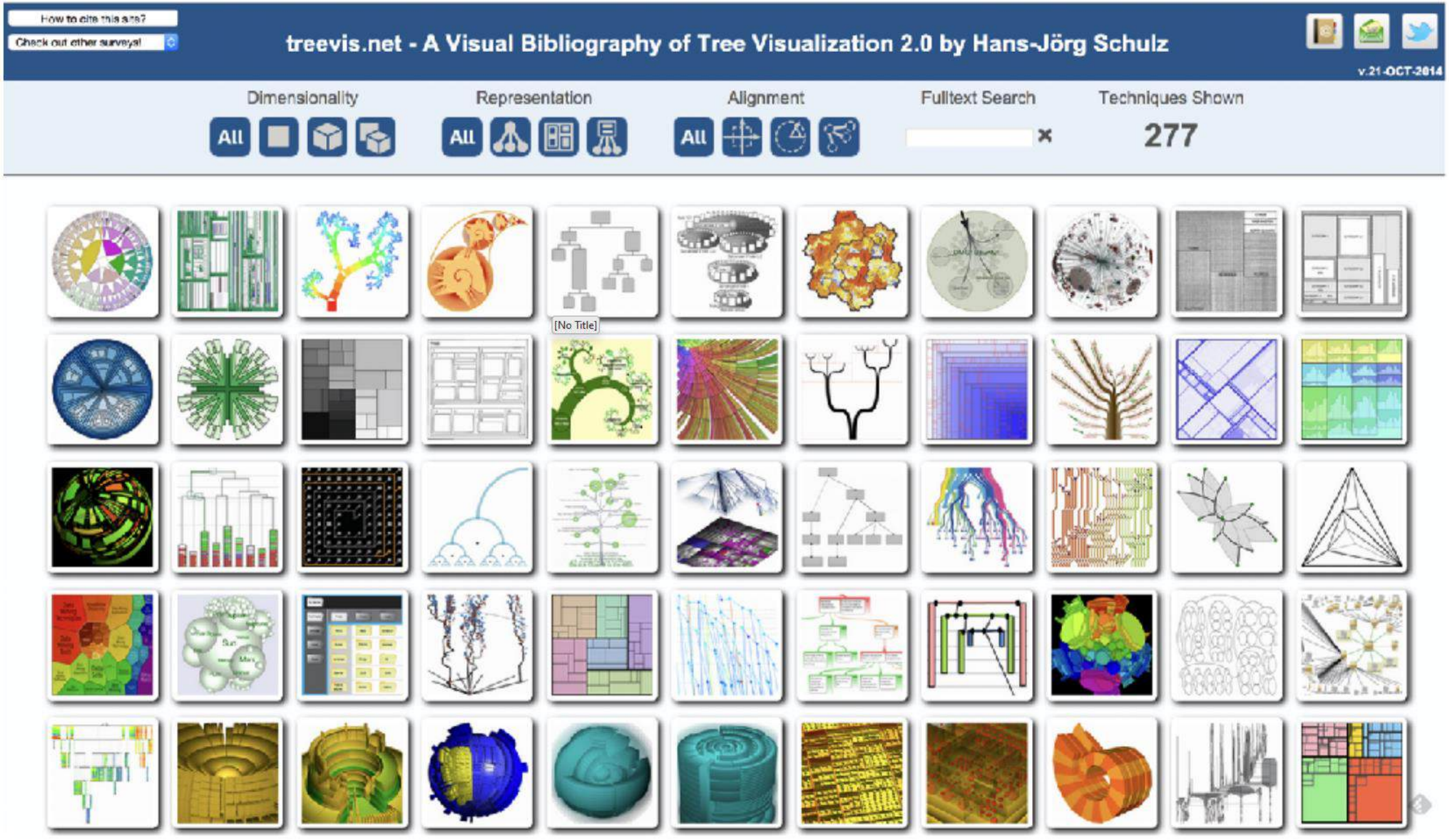
- Data shown
  - Link relationships
  - Tree depth
  - Sibling order
- Design choices
  - Connection vs containment link marks
  - Rectilinear vs radial layout
  - Spatial position channels
- Considerations
  - Redundant encoding? Arbitrary encoding?
  - Information density?
    - Avoid wasting space, but don’t overlap
    - 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/

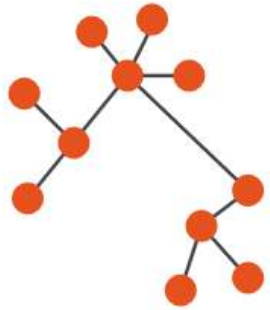


Summary

Node–Link Diagrams

Connection Marks

✔ NETWORKS ✔ TREES



Adjacency Matrix

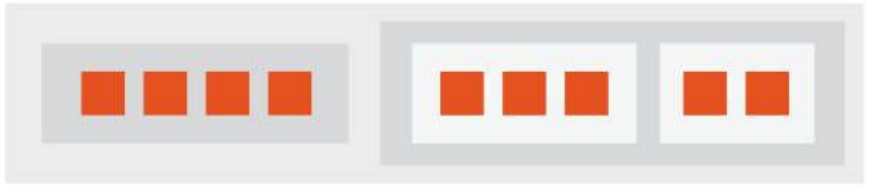
Derived Table

✔ NETWORKS ✔ TREES

Enclosure

Containment Marks

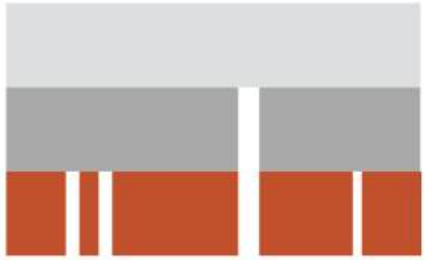
✕ NETWORKS ✔ TREES

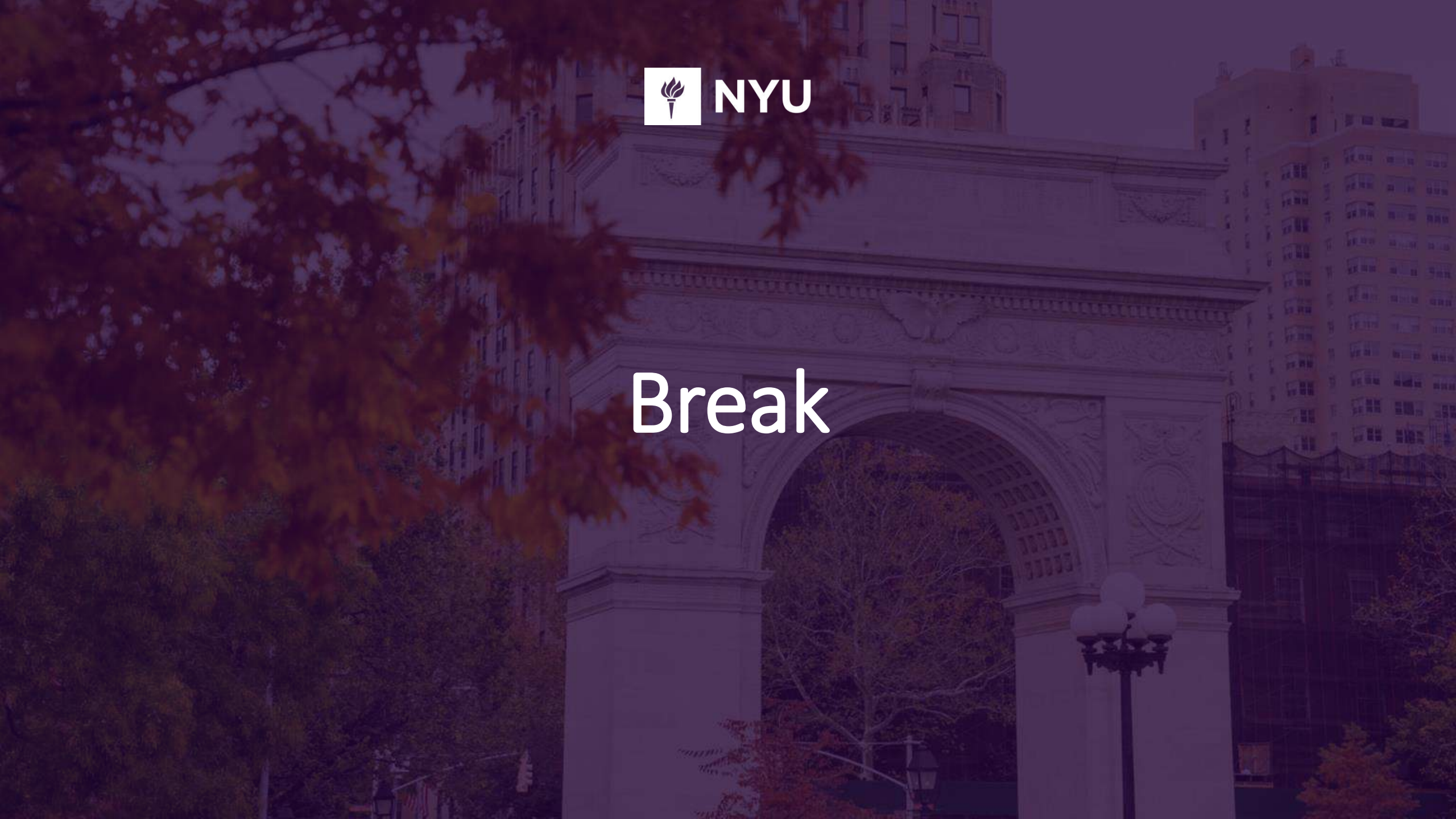


Implicit

Spatial Position

✕ NETWORKS ✔ TREES





Break