Information Visualization
Networks & Trees 1/2

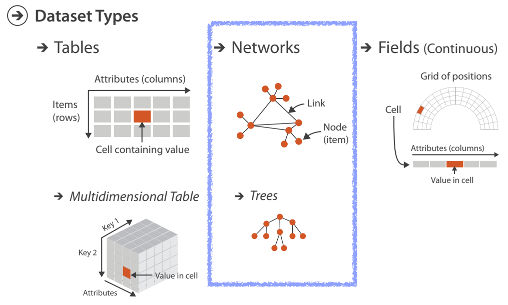
Tamara Munzner
Department of Computer Science
University of British Columbia

Lect 14/15, Feb 27 & Mar 3 2020

https://www.cs.ubc.ca/~tmm/courses/436V-20

Network data

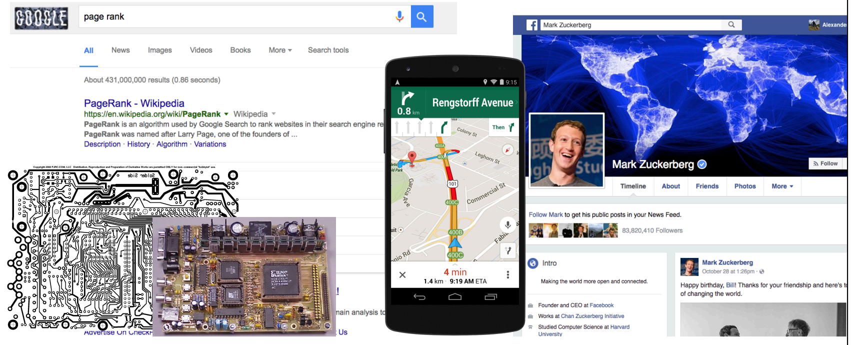
- networks
  - model relationships between things
    - aka graphs
  - two kinds of items, both can have attributes
    - nodes
    - links
- tree
  - special case
  - no cycles
    - one parent per node



Networks

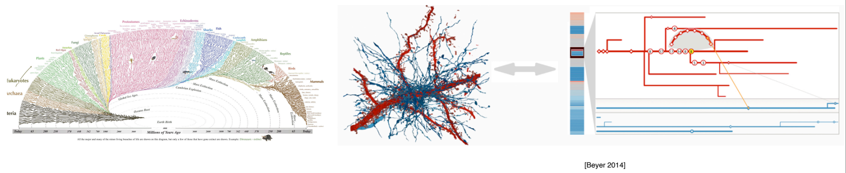
Applications of networks

- without networks, couldn't have any of these:

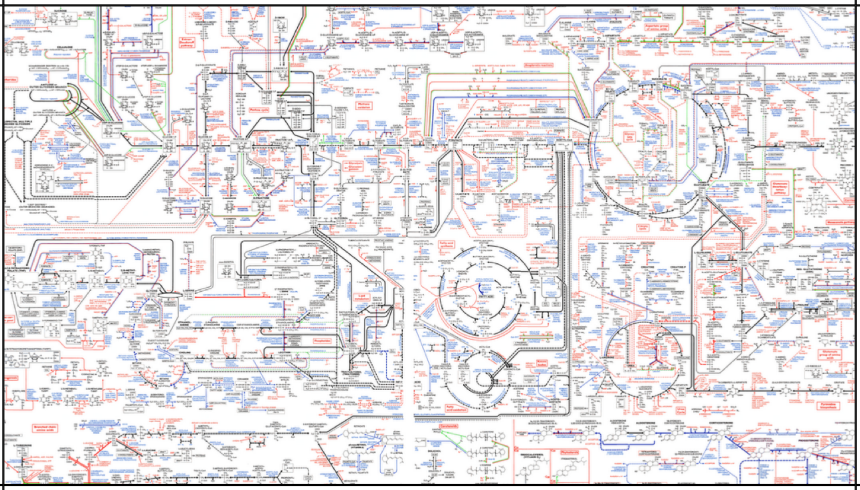


Applications of networks: biological networks

- interactions between genes, proteins, and chemical products
- the brain: connections between neurons
- your ancestry: the relations between you and your family
- phylogeny: the evolutionary relationships of life



[Beyer 2014]

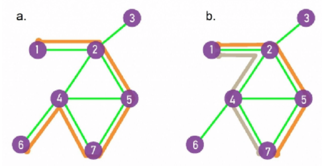


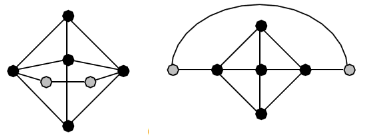
Exercise: Friend networks, two ways

- Imagine a network of friends
  - you want to visualize if and how often the friends have met each other
  - for each friend you have age, gender, and origin info
- Create two different sketches that visualize this friendship relation
- In Socrative quiz, click on true when done [~5 min]

Network tasks: topology-based

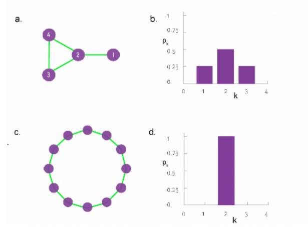
- topological structure of network
  - path following
    - path is route along links
      - *hops* from one node to another
    - path length is number of links along route
      - shortest path connects nodes i & j with smallest # of hops
- topology vs geometry
  - topological hops different from geometric distance given specific layout
    - topology does not depend on layout
    - geometry does

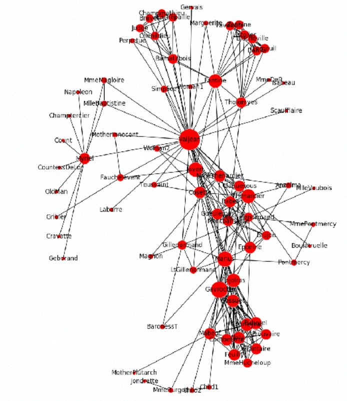




Network tasks: topology-based

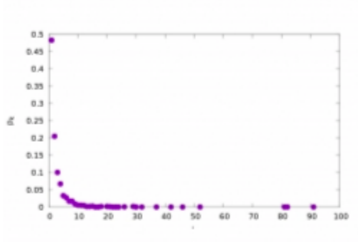
- topological structure of network
  - node importance metrics
    - node degree: attribute on nodes
      - number of links connected to a node
      - local measure of importance
      - average degree, degree distribution



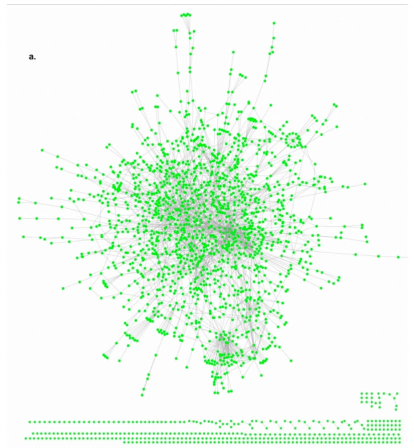


Degree distribution

- real network
  - power law distributions are common

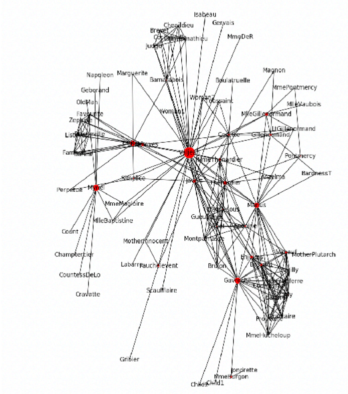


Protein interaction network, Barabasi

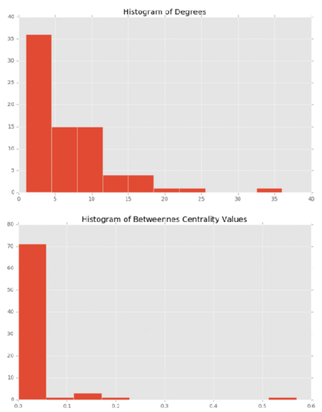


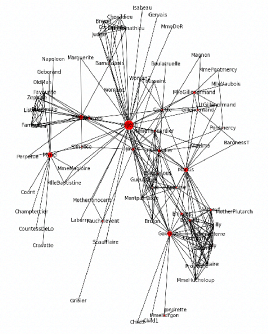
Network tasks: topology-based

- topological structure of network
  - node importance metrics
    - betweenness centrality: attribute on nodes
      - how many shortest paths pass through a node
      - global measure of importance
      - good measure for overall relevance of node in network



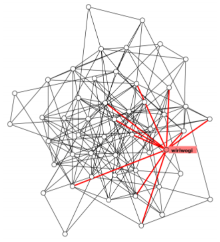
Centrality measures: Degree vs betweenness centrality



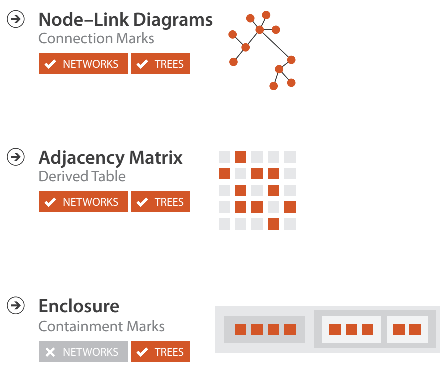


Network tasks: attribute-based vs topology-based

- topology based tasks
  - find paths
  - find (topological) neighbors
  - compare centrality/importance measures
  - identify clusters / communities
- attribute based tasks (similar to table data)
  - find extreme values, ...
- combination tasks - incorporating both
- example: locate - find single or multiple nodes/links with a given property
  - topology: find all adjacent nodes of given node
  - attributes: find edges with maximum edge weight

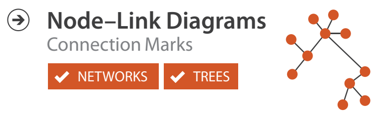


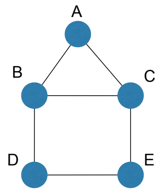
Three kinds of network visual encodings

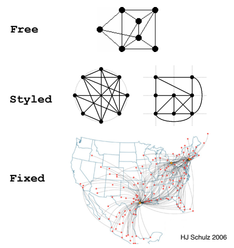


Node-link diagrams

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







Exercise

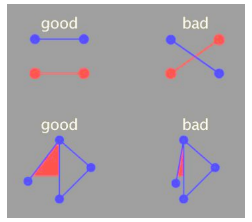
- sketch an aesthetically pleasing node-link diagram of this network
  - there are five nodes: A,B,C,D,E
  - each row in the table describes an edge

A	B
C	D
C	B
A	D
A	C
B	D
D	E
A	E

- Socrative quiz: pick true when done [~5 min]

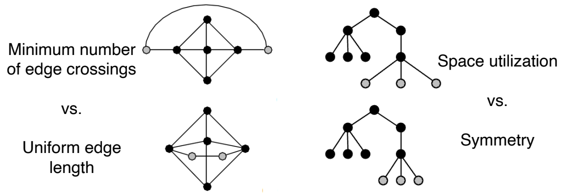
Criteria for good node-link layouts

- minimize
  - edge crossings
  - distances between topological neighbor nodes
  - total drawing area
  - edge bends
  - edge length disparities (sometimes)
- 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



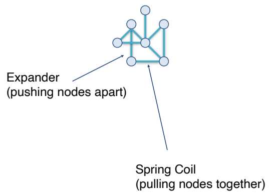
Schulz 2004

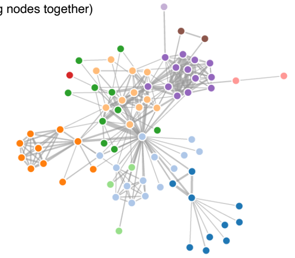
Optimization-based layouts

- 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

Force-directed placement

- physics model
  - links = springs pull together
  - nodes = magnets repulse apart
- algorithm
  - place vertices in random locations
  - while not equilibrium
    - calculate force on vertex
      - sum of
        - pairwise repulsion of all nodes
        - attraction between connected nodes
    - move vertex by c * vertex_force

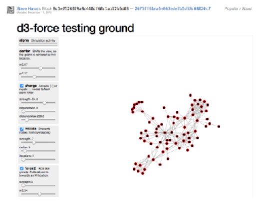




http://mbostock.github.com/d3/ex/force.html

Force-directed placement properties

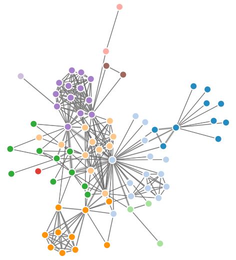
- strengths
  - reasonable layout for small, sparse graphs
  - clusters typically visible
  - edge length uniformity
- weaknesses
  - nondeterministic
  - computationally expensive: O(n^3) for n nodes
    - each step is n^2, takes ~n cycles to reach equilibrium
  - naive FD doesn't scale well beyond 1K nodes
  - iterative progress: engaging but distracting



https://bl.ocks.org/steveharoz/8c3e2524079a8c440df60c1ab72b5d03

Idiom: **force-directed placement**

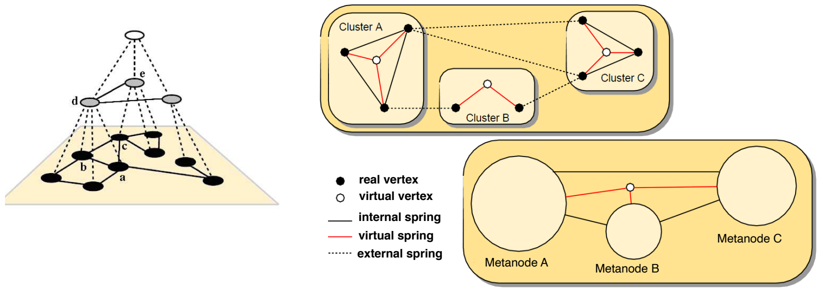
- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length
      - 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

Multilevel approaches

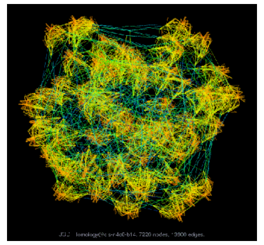
- derive cluster hierarchy of metanodes on top of original graph nodes



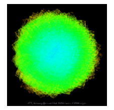
[Schulz 2004]

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

- data
  - 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



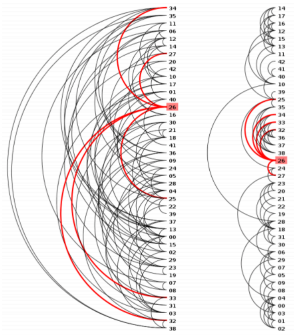
[Efficient and high quality force-directed graph drawing. Hu. The Mathematica Journal 10:37–71, 2005.]

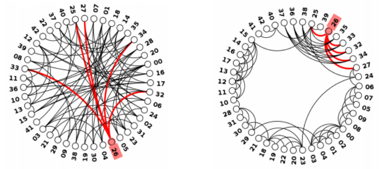


http://www.research.att.com/yifanhu/GALLERY/GRAPHS/index1.html

Restricted layouts: Circular, arc

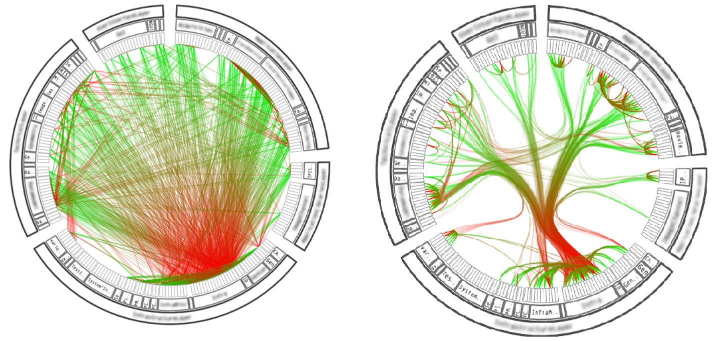
- lay out nodes around circle or along line
  - circular layouts
  - arc diagrams
- node ordering crucial to avoid excessive clutter from edge crossings
  - barycentric ordering before & after
  - derived attribute: global computation





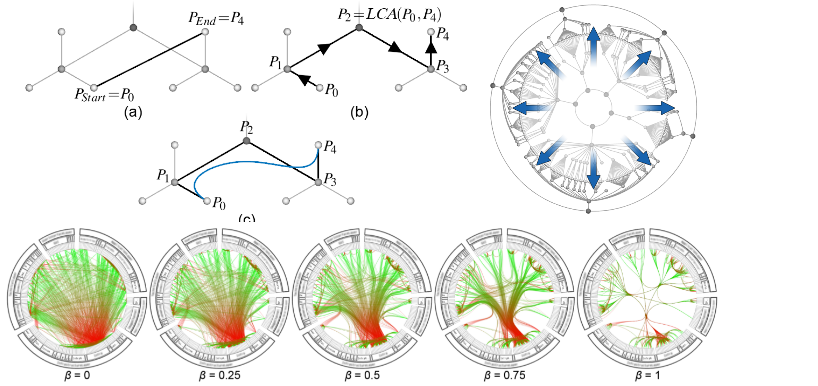
http://profs.etsmtl.ca/mmcguffin/research/2012-mcguffin-simpleNetVis/mcguffin-2012-simpleNetVis.pdf

Edge clutter reduction: hierarchical edge bundling



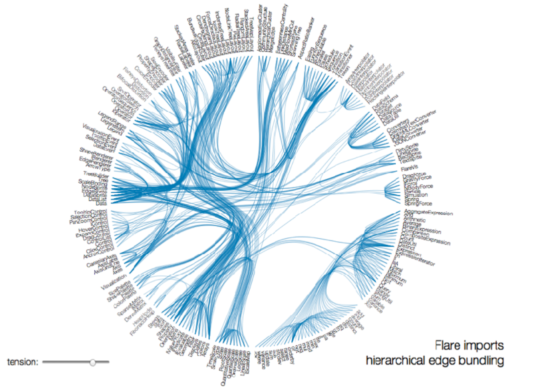
[Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. Danny Holten. TVCG 12(5):741-748 2006]

Hierarchical edge bundling



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

Bundle strength

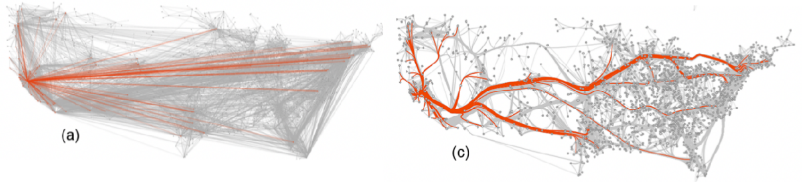


http://mbostock.github.io/d3/talk/20111116/bundle.html

Fixed layouts: Geographic

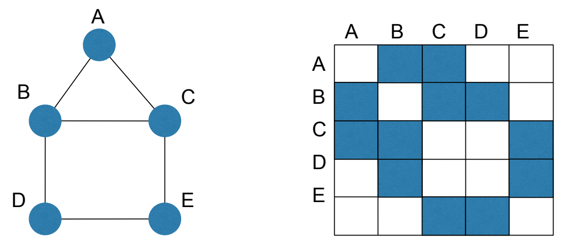
- lay out network nodes using given/fixed spatial data
  - route edges accordingly
  - edge bundling also applicable

https://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919

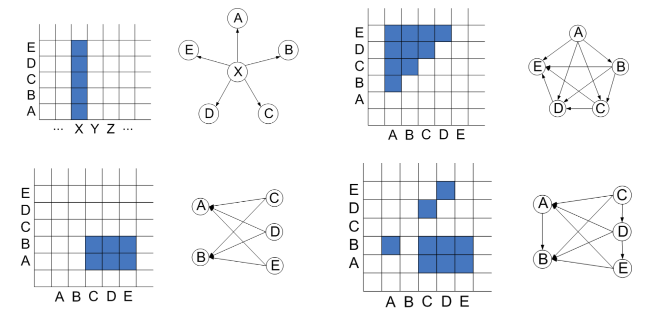


Adjacency matrix representations

- derive adjacency matrix from network

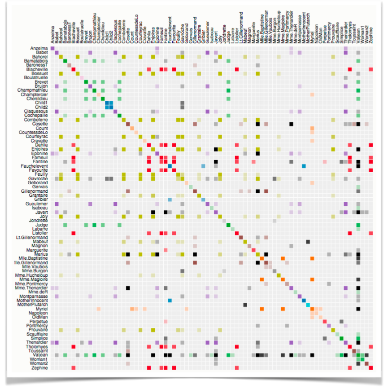


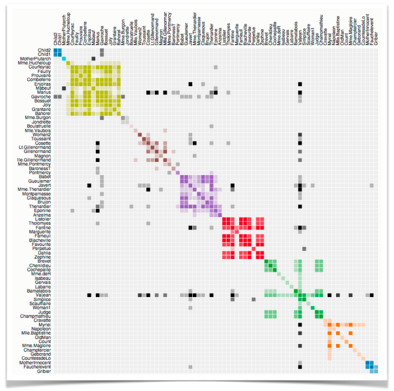
Adjacency matrix examples



HJ Schulz 2007

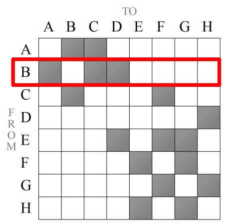
Node order is crucial: Reordering

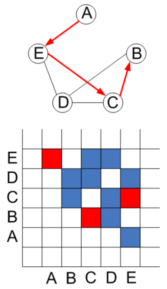




https://bost.ocks.org/mike/miserables/

Adjacency matrix

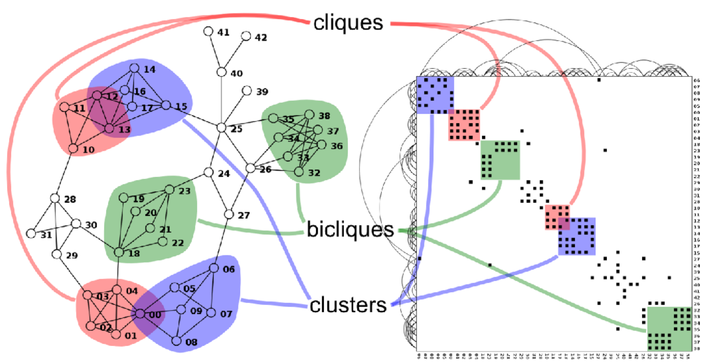




good for topology tasks related to neighborhoods (node 1-hop neighbors)

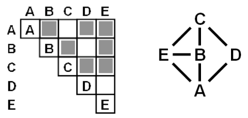
bad for topology tasks related to paths

Structures visible in both



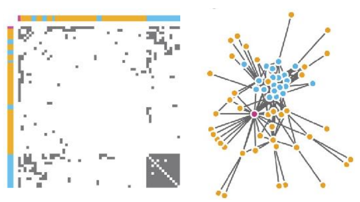
http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

Idiom: **adjacency matrix view**



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

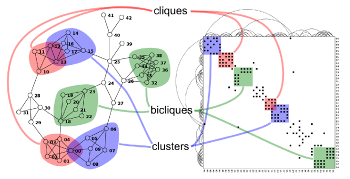
- data: network
  - transform into same data/encoding as heatmap
- derived data: table from network
  - 1 quant attrib
    - weighted edge between nodes
  - 2 categ attribs: node list x 2
- visual encoding
  - cell shows presence/absence of edge
- scalability
  - 1K nodes, 1M edges



[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!

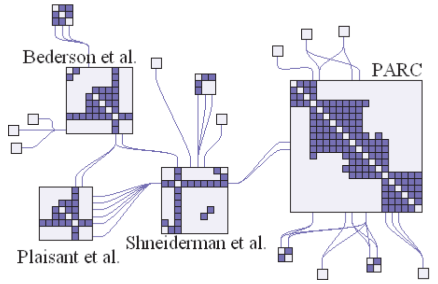


http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

[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.]

Idiom: **NodeTrix**

- hybrid nodelink/matrix
- capture strengths of both



NodeTrix
[Henry et al. 2007]

Trees

Tree exercise

Here is part of a directory structure used for the material for this class and the relative file size.

datavis-17/
 lectures/
  Intro.key (110 MB)
  perception/
   Perception.key (113 MB)
   Blindness.mov (15MB)
  Data.key (12 MB)
  Graphs.key (180 MB)
 exams/
  Exam1-solution.doc (5MB)
  Exam1.doc (1MB)
 exercise/
  Graph.doc (3MB)
  Graph-video.doc (210MB)

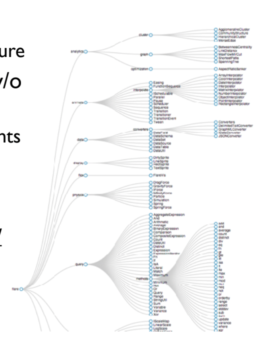
Sketch two different visualizations that show both, the directory structure and the size of the directories and the contained files.

- socrative: true when done with both

Node-link trees

- Reingold-Tilford
  - tidy drawings of trees
    - exploit parent/child structure
  - allocate space: compact w/o overlap
    - rectilinear and radial variants
  - nice algorithm writeup

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



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

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

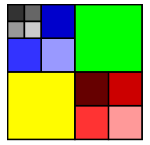
Idiom: **radial node-link tree**

- 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

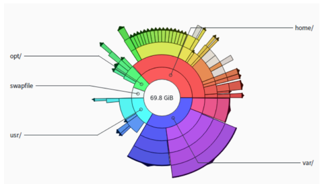
http://mbostock.github.com/d3/ex/tree.html

Containment (implicit) layouts

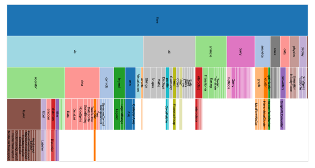
Treemap



Sunburst

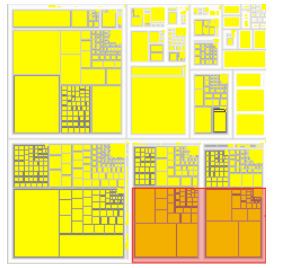


Icicle Plot



Idiom: **treemap**

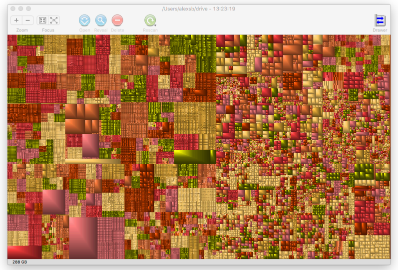
- data
  - tree
  - 1 quant attrib at leaf nodes
- encoding
  - area containment marks for hierarchical structure
  - rectilinear orientation
  - size encodes quant attrib
- tasks
  - query attribute at leaf nodes
- scalability
  - 1M leaf nodes



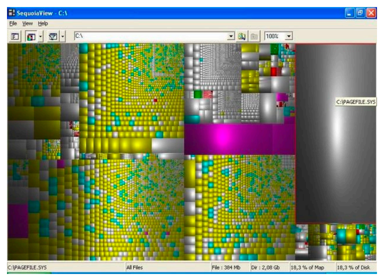
http://tulip.labri.fr/Documentation/3_7/userHandbook/html/ch06.html

Treemap software: disk space

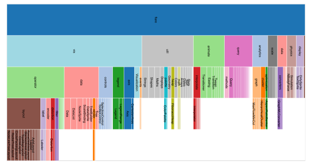
Mac: GrandPerspective

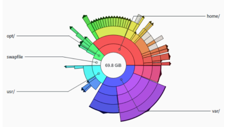


Windows: Sequoia View

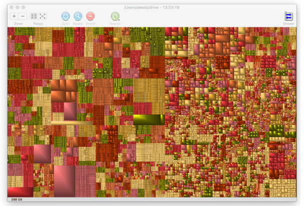


Containment: Approaches compared





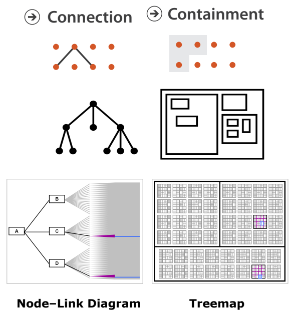
Inner Nodes and Leaves Visible



Only Leaves Visible

Link marks: Connection and containment

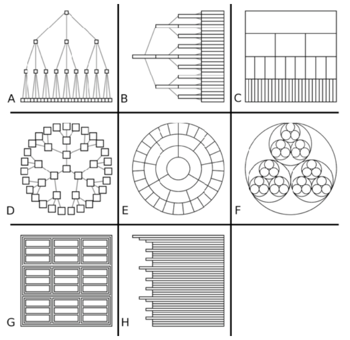
- marks as links (vs. nodes)
  - common case in network drawing
  - 1D case: connection
    - ex: all node-link diagrams
    - emphasizes topology, path tracing
    - networks and trees
  - 2D case: containment
    - ex: all treemap variants
    - emphasizes attribute values at leaves (size coding)
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



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

Tree drawing idioms comparison

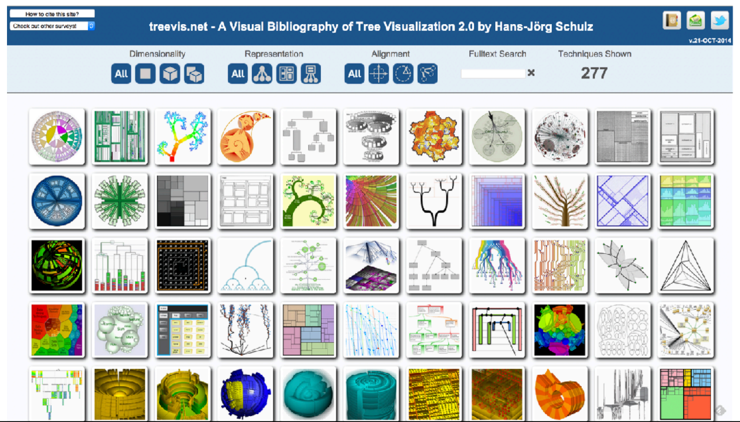
- data shown
  - link relationships
  - tree depth
  - sibling order
- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels
- considerations
  - redundant? arbitrary?
  - information density?
    - avoid wasting space
    - consider where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



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

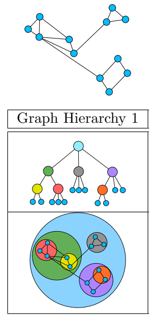
treevis.net: many, many options

https://treevis.net/



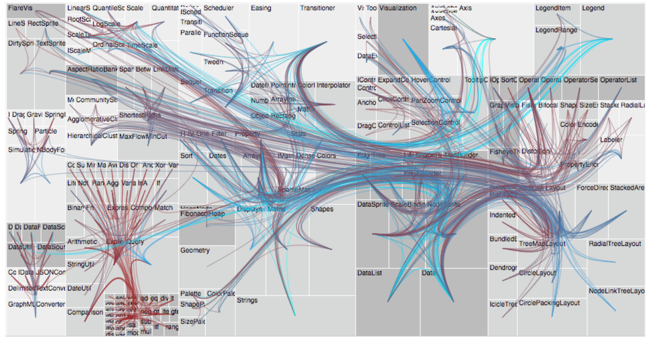
Idiom: **GrouseFlocks**

- data: compound graphs
  - 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.]

Hierarchical edge bundling: treemap vs radial



Credits

- Visualization Analysis and Design (Ch 9)
- Alex Lex & Miriah Meyer, http://dataviscourse.net/