Ch 9: Networks
Paper: Genealogical Graphs
Paper: ABySS-Explorer

Tamara Munzner
Department of Computer Science
University of British Columbia

CPSC 547, Information Visualization
Week 9: 30 October 2025

www.cs.ubc.ca/~tmm/courses/547-25

Plan for today

- mini-lecture / Q&A
  - networks
  - paper: Abyss-Explorer [type: **design study**]
  - paper: Geneaology [type: **technique**]
  - Q&A roundup from previous weeks
- break
- meet with project teams individually
  - discuss my written feedback

Upcoming

- week 10 reading
  - paper: Polaris/Tableau [type: **system**]
  - paper: D3 [type: **system**]
  - paper: Vega-Lite [type: **system**]
  - reminder: NO CLASS Nov 6
- week 11
  - no more reading!
  - project updates due Wed Nov 12 noon
  - project peer reviews in class (Thu Nov 13)
    - Gale-Ricky :: Alice-Minju
    - Haeji-Yuri :: Kevin-Raymond
    - Clarkson :: Tamara

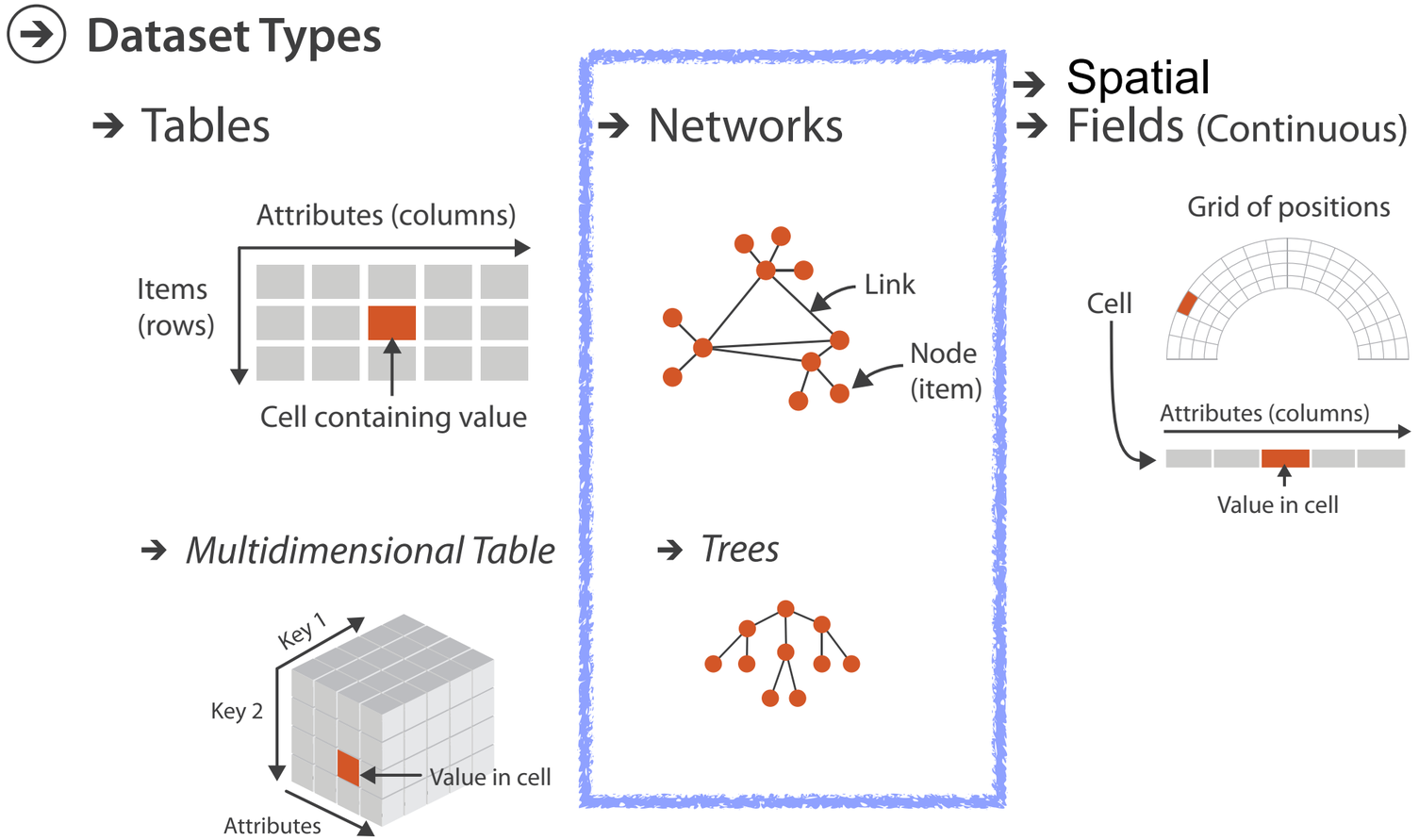
Upcoming

- week 12
  - no readings, work on projects
  - in class: post-update meetings with me (1on1 with each team)
- week 13
  - no readings, work on projects
  - in class: lecture, paper writing and research process; evals
- week 14
  - no readings, work on projects
  - in class: lecture, advanced topics
- week 15
  - final presentations, exact timing TBD

Networks

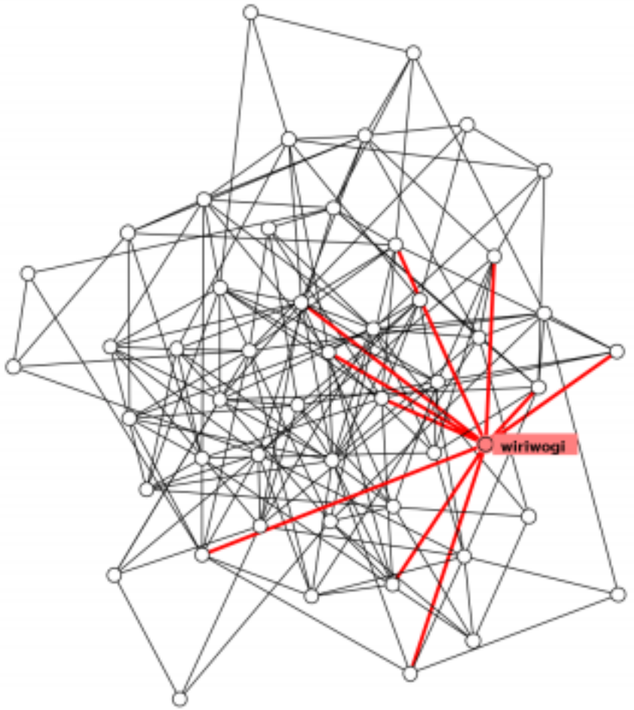
Network data

- 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



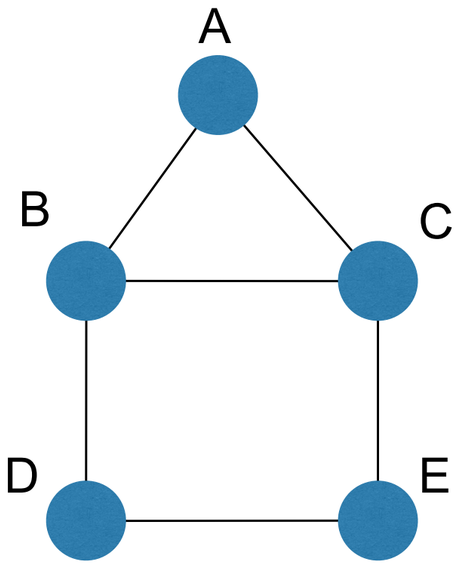
Network tasks: topology-based and attribute-based

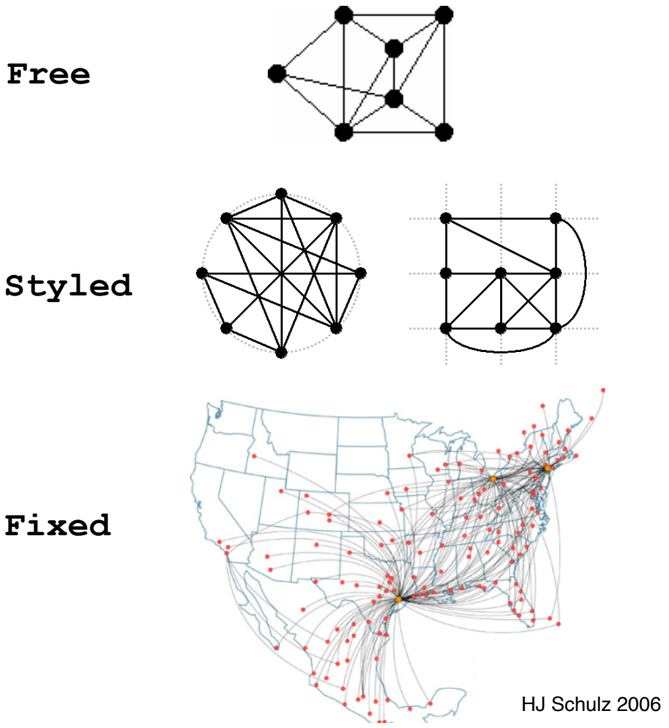
- topology based tasks
  - find paths
  - find (topological) neighbors
  - compare centrality/importance measures
  - identify clusters / communities
- attribute based tasks (similar to table data)
  - find distributions, ...
- combination tasks, incorporating both
  - example: find friends-of-friends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



Node-link diagrams

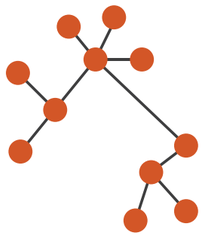
- nodes: point marks
- links: line marks
  - straight lines or arcs
  - connections between nodes
- intuitive & familiar
  - most common
  - many, many variants





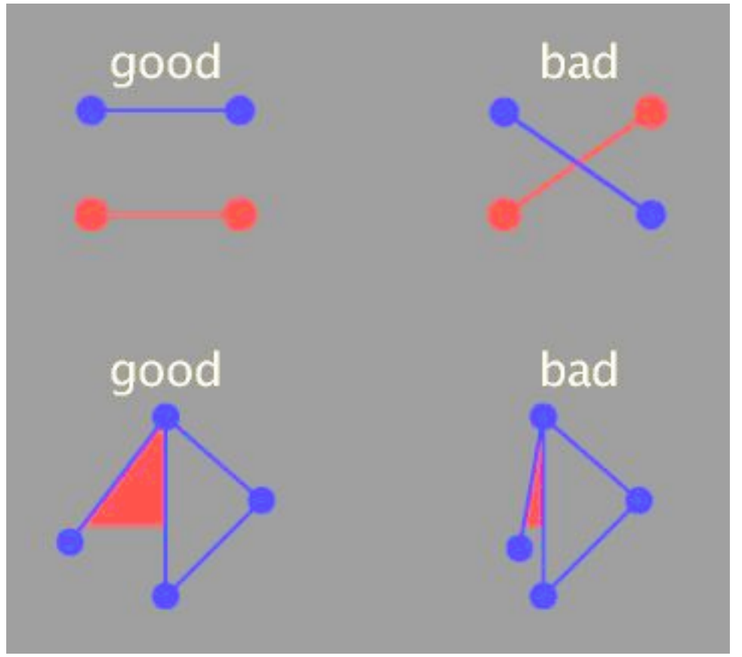
Node–Link Diagrams
Connection Marks

✔ NETWORKS ✔ TREES



Criteria for good node-link layouts

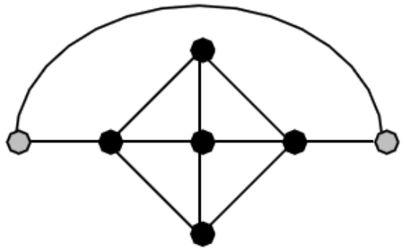
- minimize
  - edge crossings, node overlaps
  - distances between topological neighbor nodes
  - total drawing area
  - edge bends
- maximize
  - angular distance between different edges
  - aspect ratio disparities
- emphasize symmetry
  - similar graph structures should look similar in layout



Criteria conflict

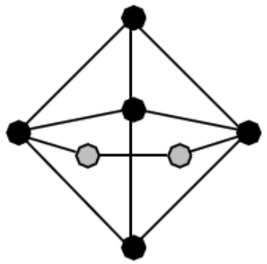
- most criteria NP-hard individually
- many criteria directly conflict with each other

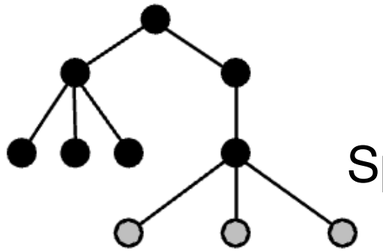
Minimum number of edge crossings



vs.

Uniform edge length

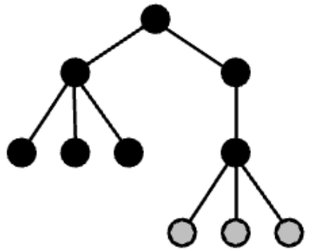




Space utilization

vs.

Symmetry



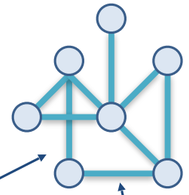
Schulz 2004

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

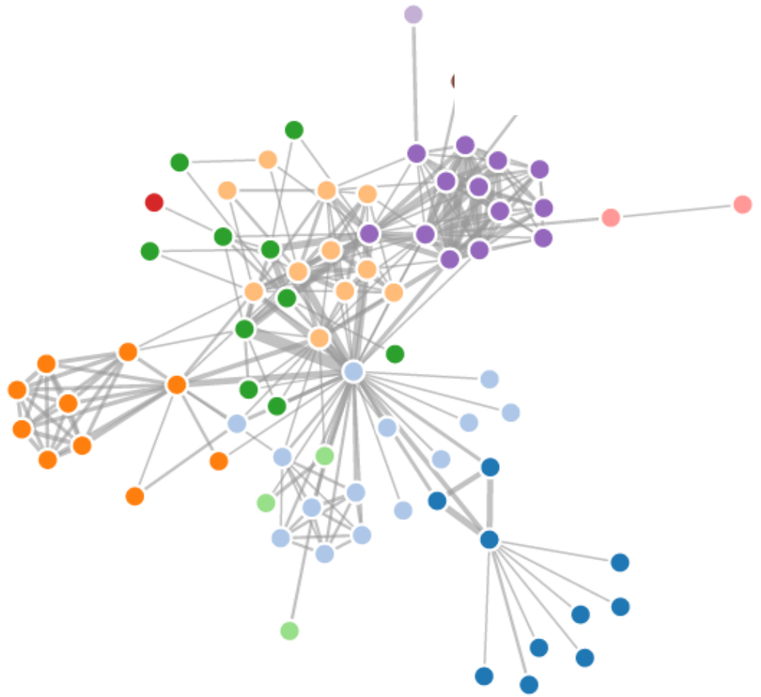
- physics model
  - links = springs pull together
  - nodes = magnets repulse apart



Expander
(pushing nodes apart)

Spring Coil
(pulling nodes together)

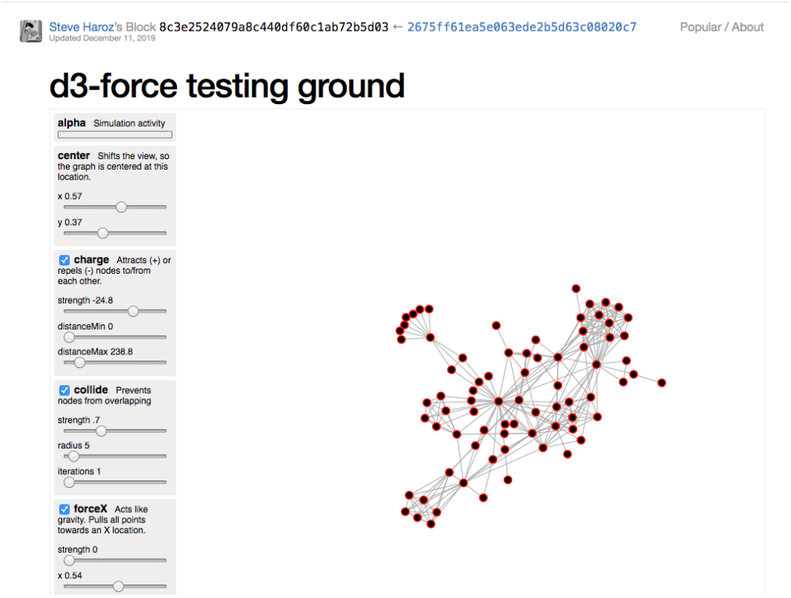
- algorithm
  - place vertices in random locations
  - while not equilibrium
    - calculate force on vertex
      - sum of
        - » pairwise repulsion of all nodes
        - » attraction between connected nodes
    - move vertex by c * vertex_force



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

Force-directed placement properties

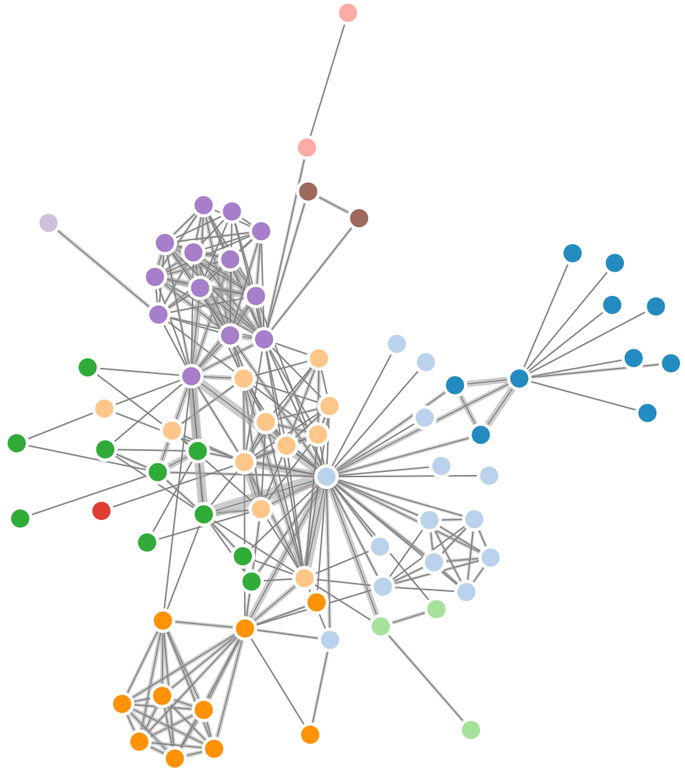
- strengths
  - reasonable layout for small, sparse graphs
  - clusters typically visible
  - edge length uniformity
- weaknesses
  - nondeterministic
  - computationally expensive: O(n^3) for n nodes
    - each step is n^2, takes ~n cycles to reach equilibrium
  - naive FD doesn't scale well beyond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



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

Idiom: **force-directed placement**

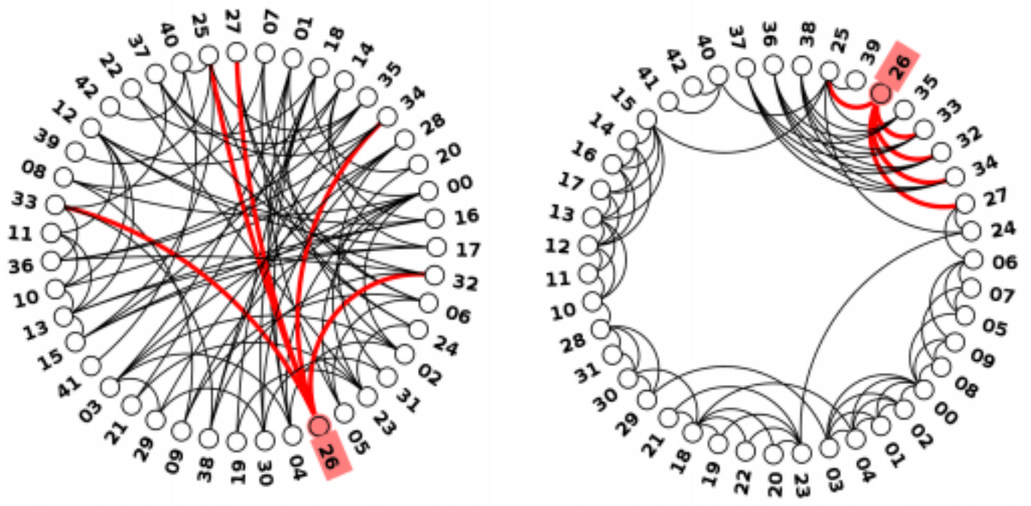
- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length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

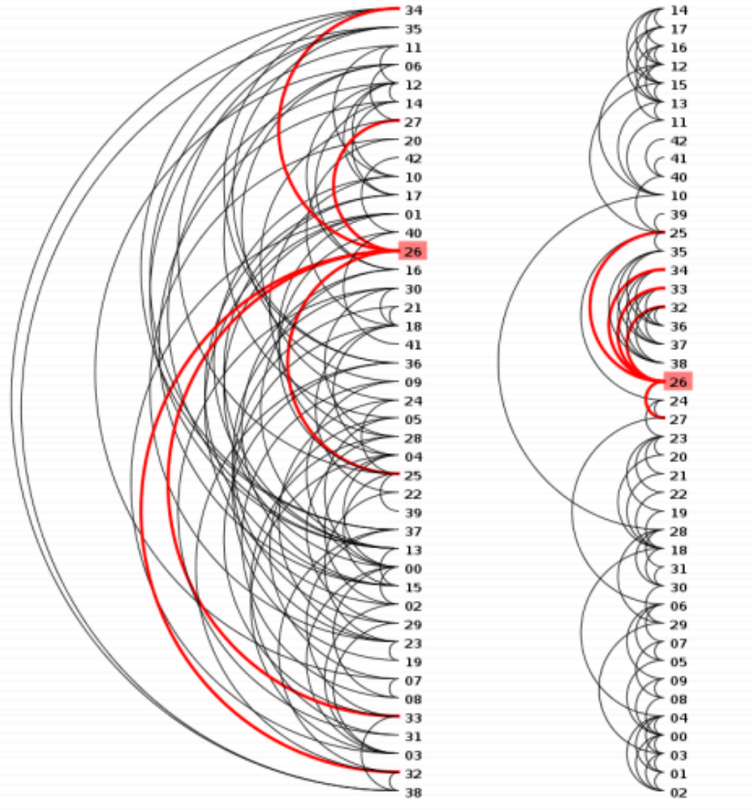


http://mbostock.github.com/d3/ex/force.html

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

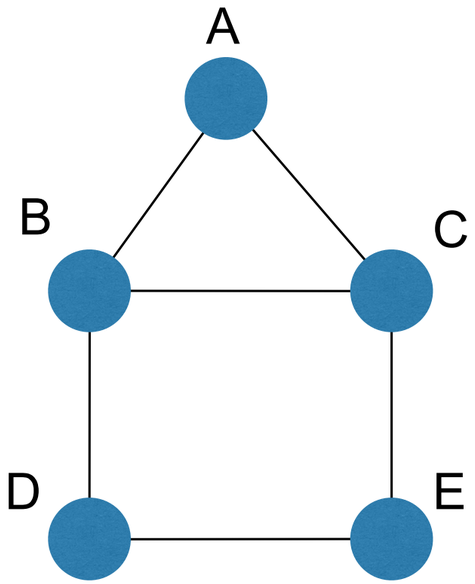


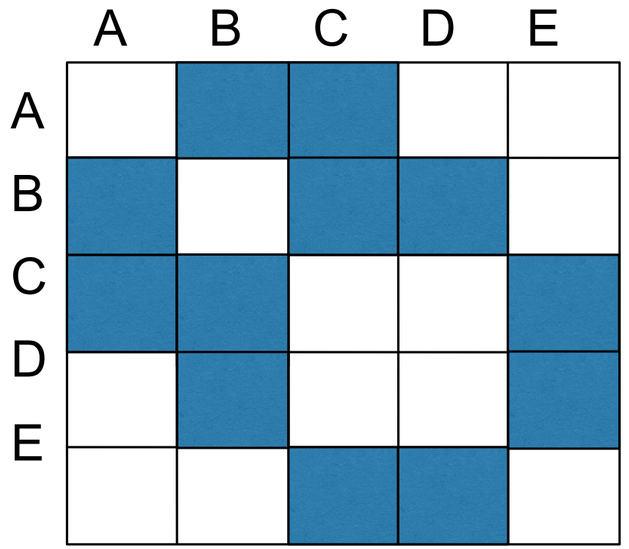


http://profs.etsmtl.ca/mmcguffin/research/2012-mcguffin-simpleNetVis/mcguffin-2012-simpleNetVis.pdf

Adjacency matrix representations

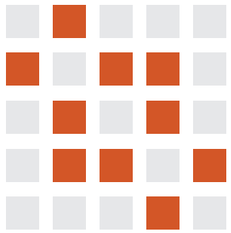
- derive adjacency matrix from network



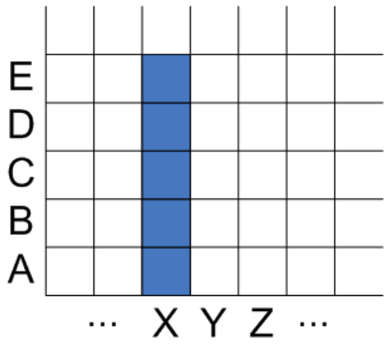


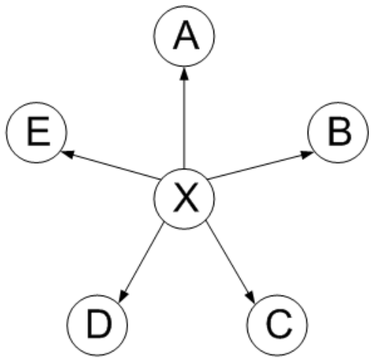
Adjacency Matrix
Derived Table

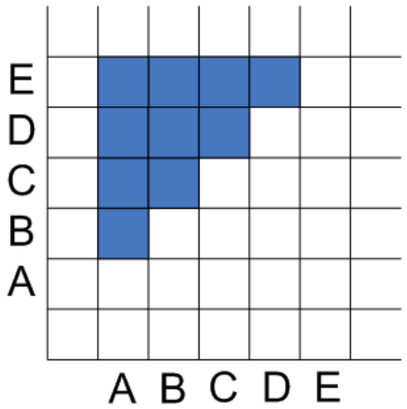
✔ NETWORKS ✔ TREES

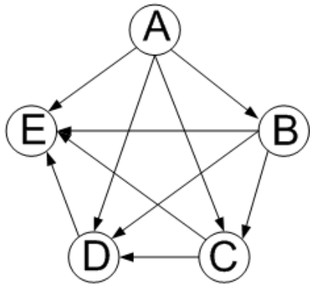


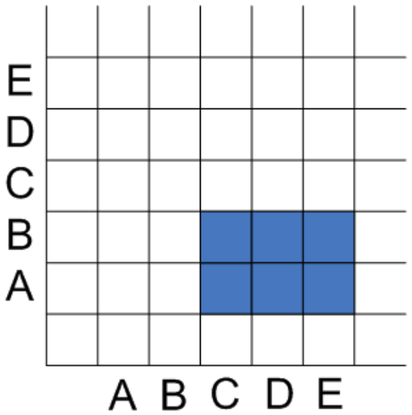
Adjacency matrix examples

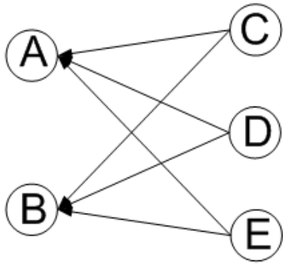


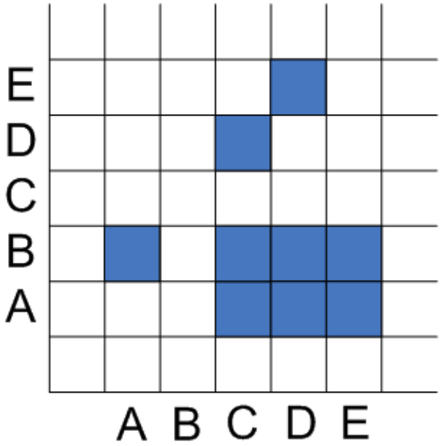


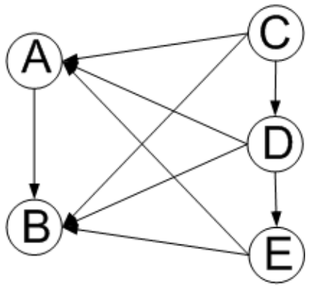






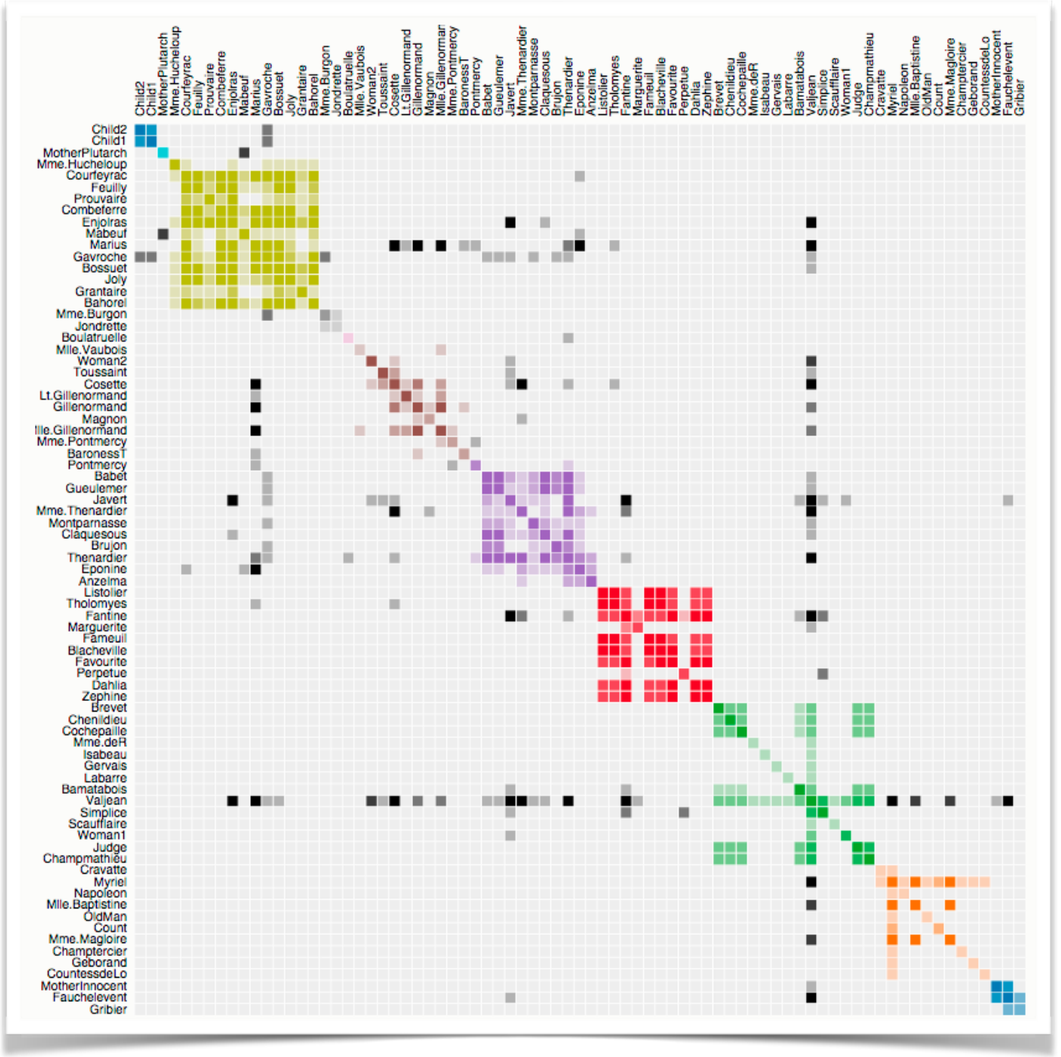






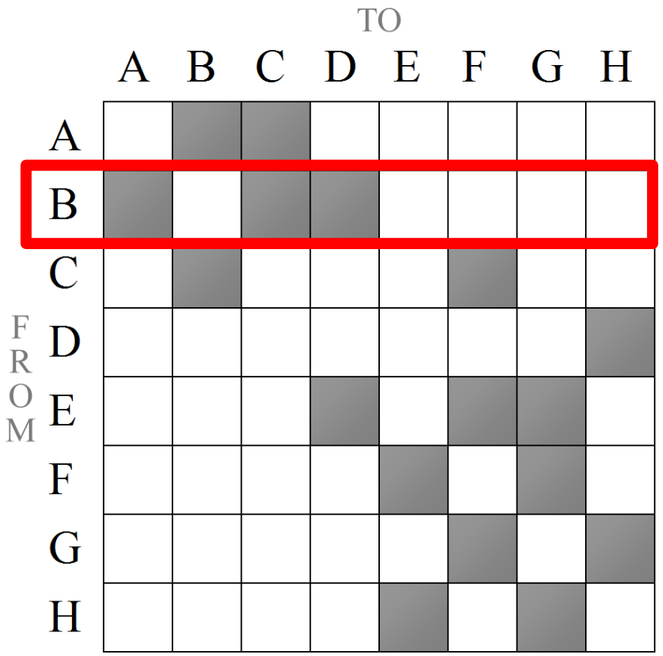
HJ Schulz 2007

Node order is crucial: Reordering

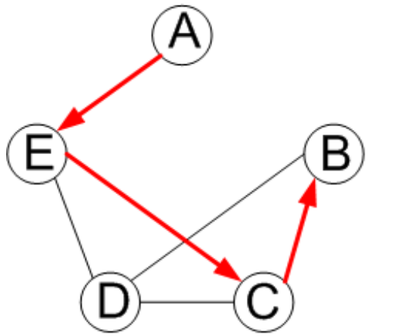


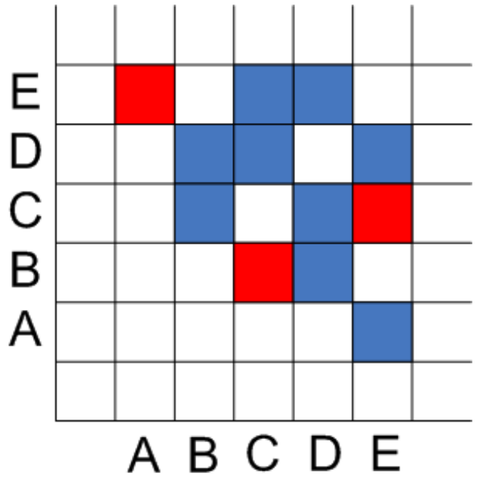
https://bost.ocks.org/mike/miserables/

Adjacency matrix



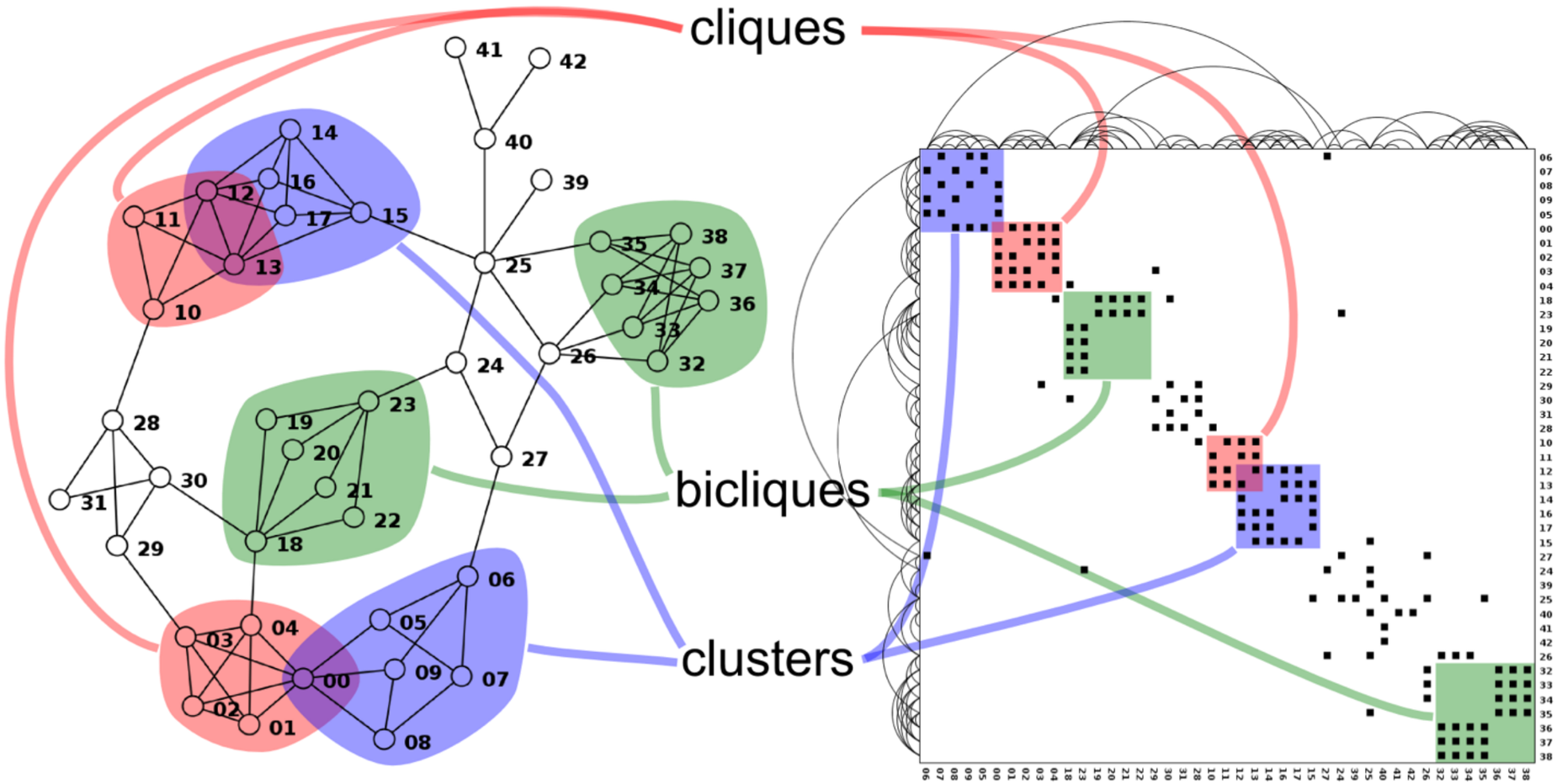
good for topology tasks
related to neighborhoods
(node 1-hop neighbors)





bad for topology tasks
related to paths

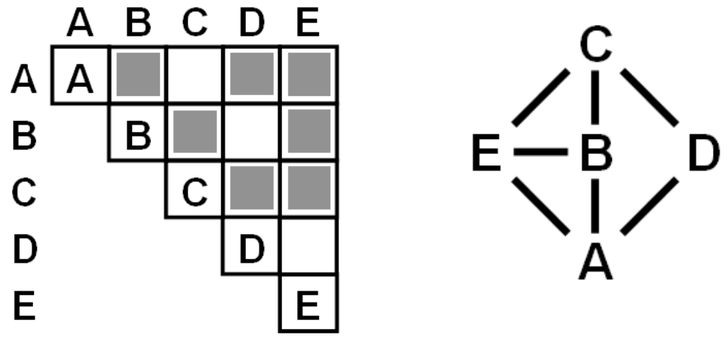
Structures visible in both



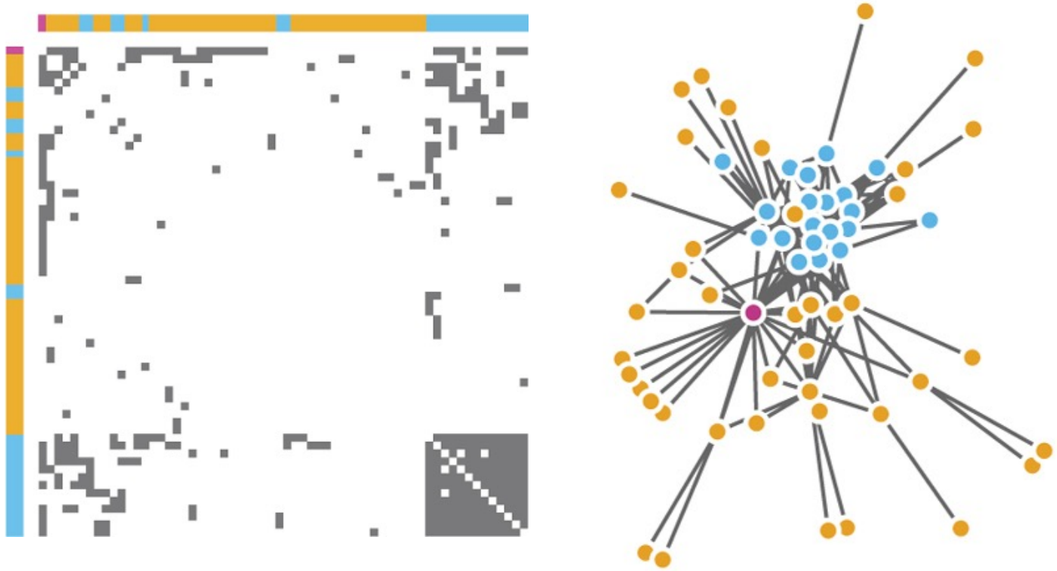
http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

Idiom: **adjacency matrix view**

- 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.]

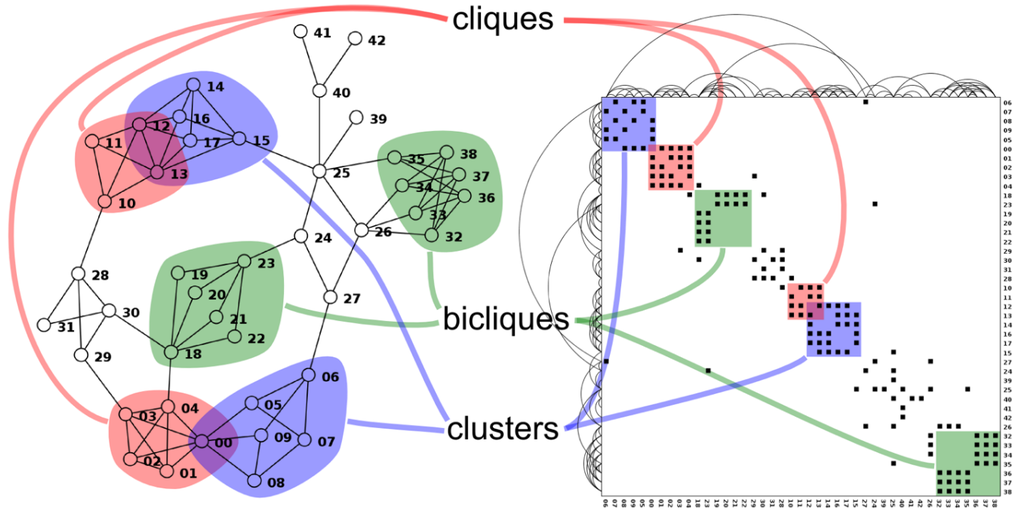


[Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115.]

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!

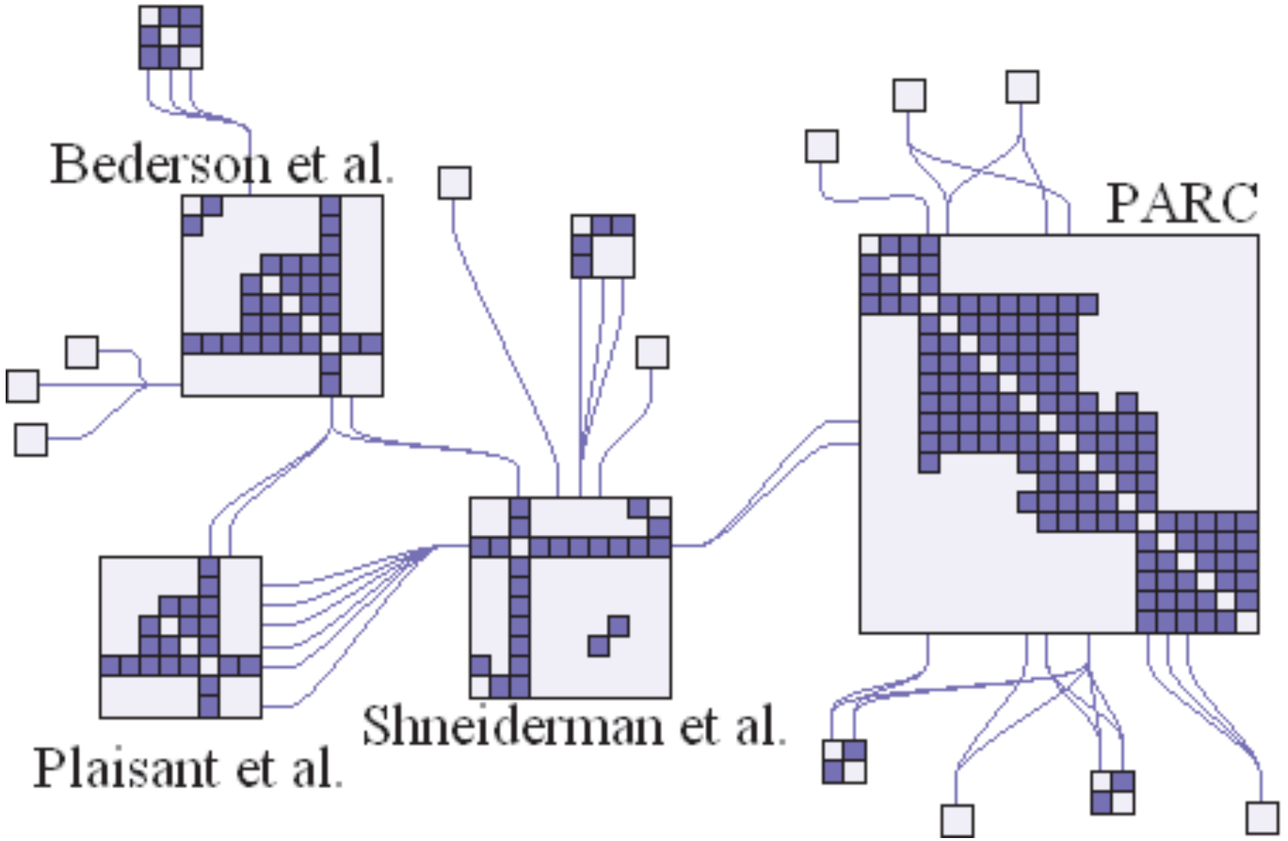
[On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. Ghoniem, Fekete, and Castagliola. Information Visualization 4:2 (2005), 114–135.]



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

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.]

Trees

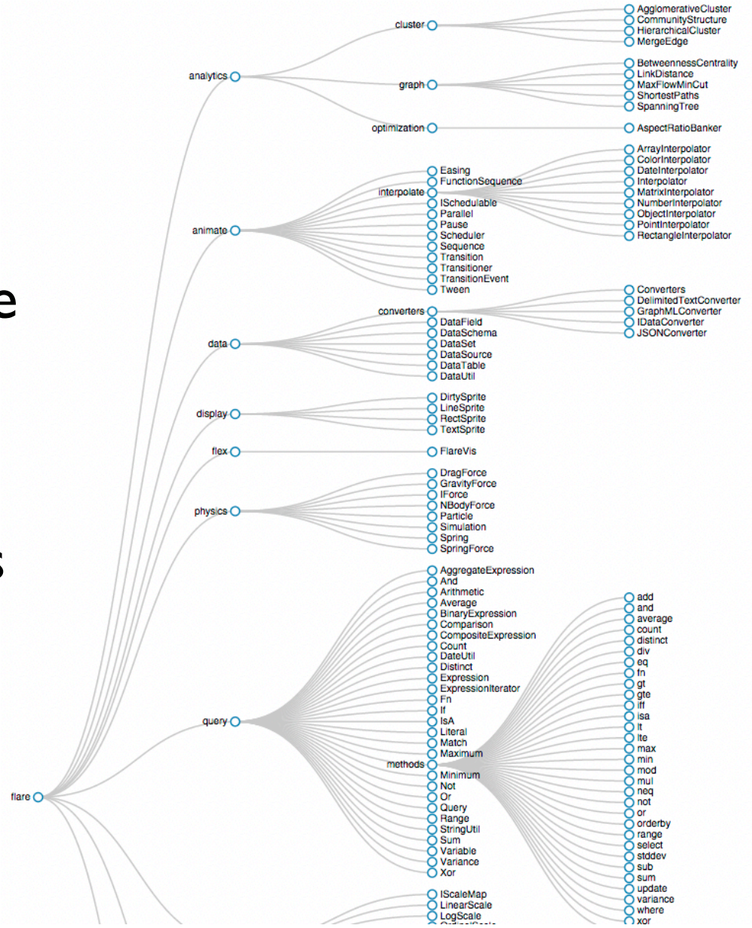
Node-link trees

- 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/

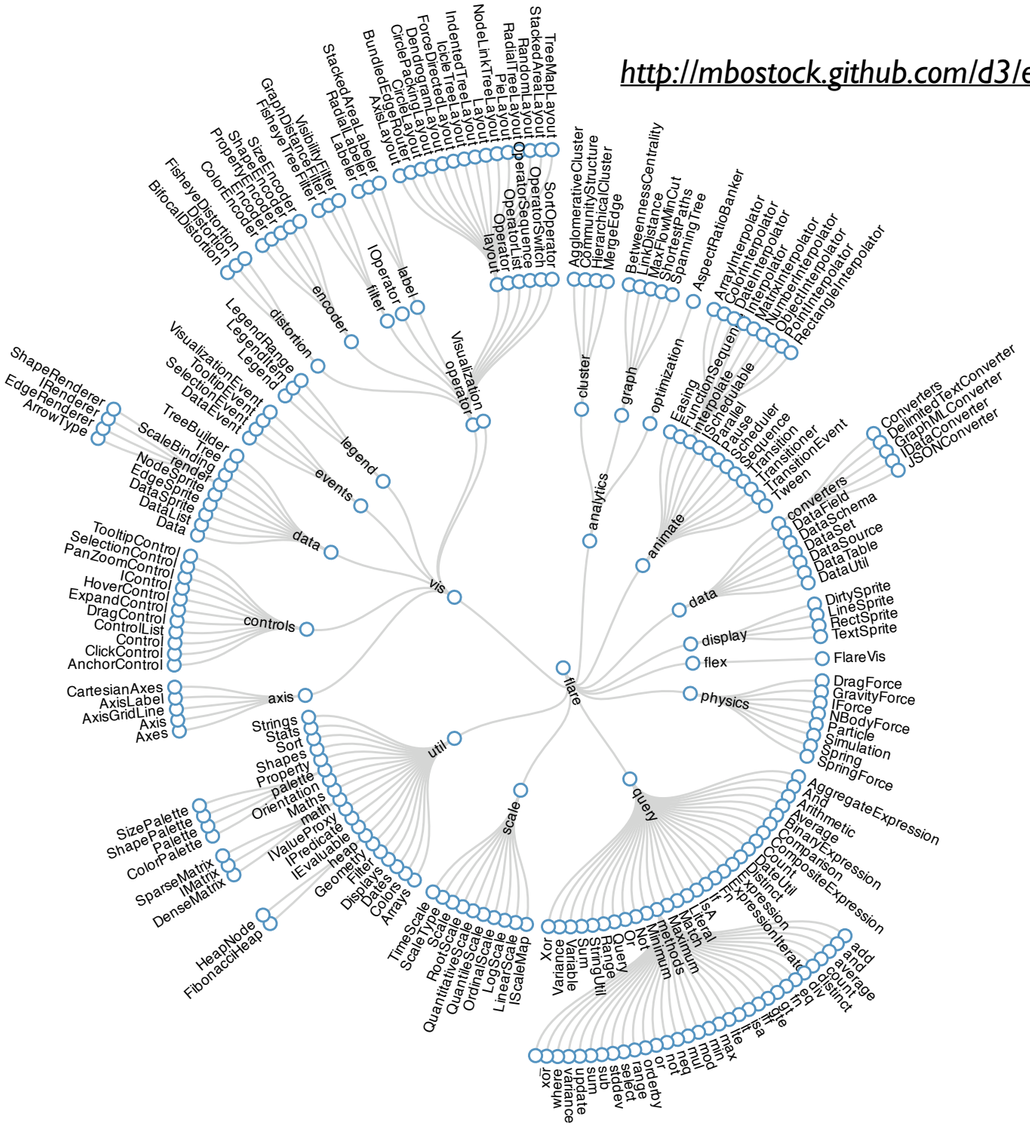


http://bl.ocks.org/mbostock/4339184

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

Idiom: **radial node-link tree**

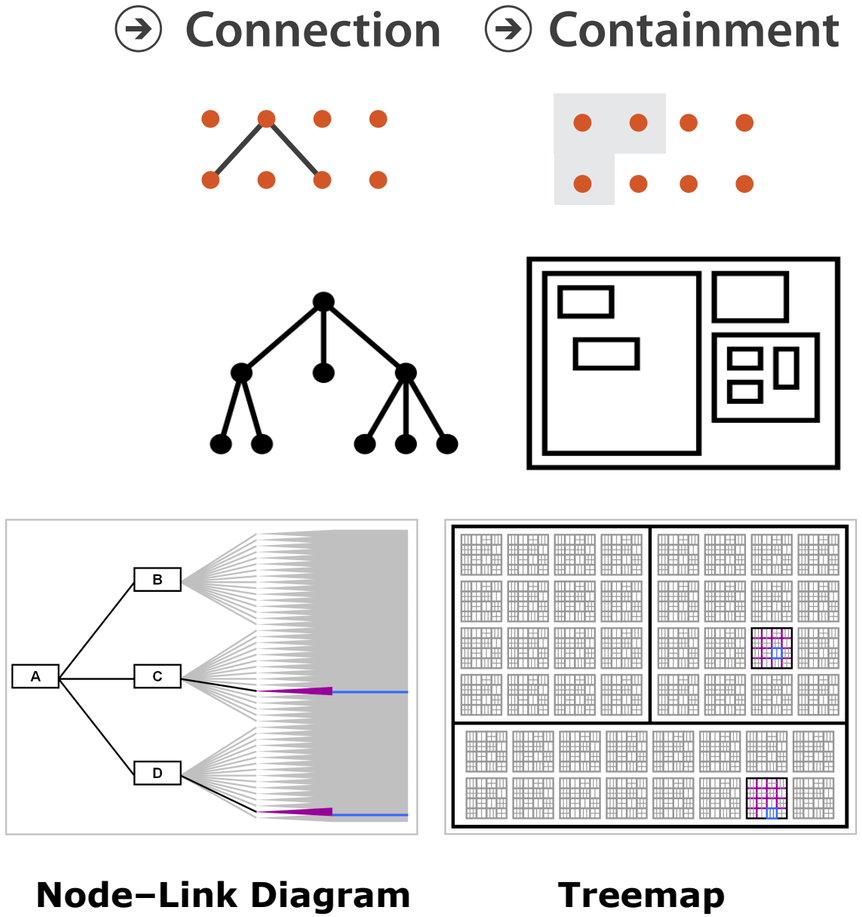
- data
  - tree
- encoding
  - link connection marks
  - point node marks
  - radial axis orientation
    - angular proximity: siblings
    - distance from center: depth in tree
- tasks
  - understanding topology, following paths
- scalability
  - 1K - 10K nodes (with/without labels)



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

Link marks: Connection and containment

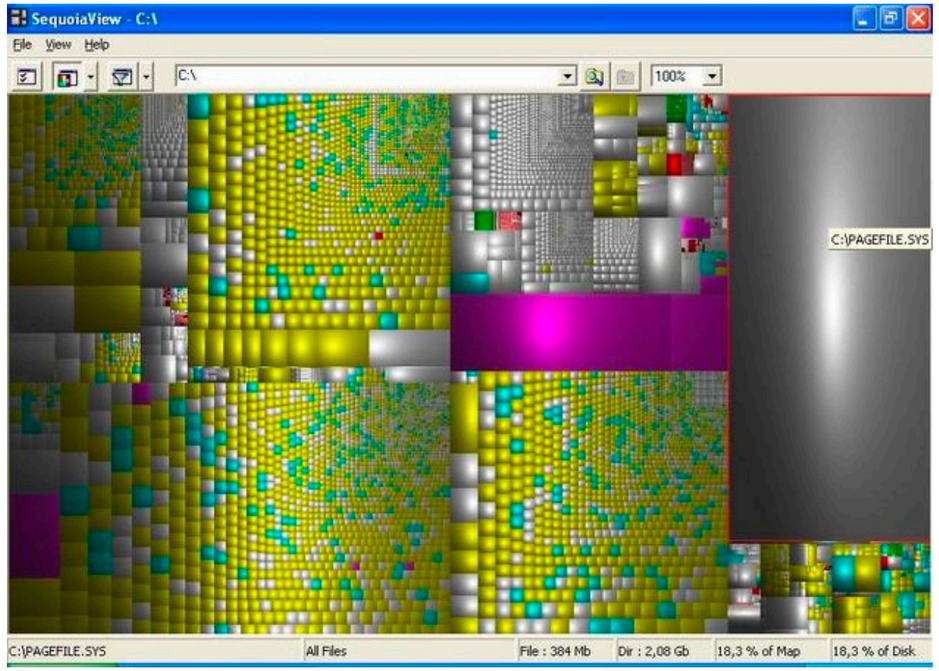
- marks as links (vs. nodes)
  - common case in network drawing
  - 1D case: connection
    - ex: all node-link diagrams
    - emphasizes topology, path tracing
    - networks and trees
  - 2D case: containment
    - ex: all treemap variants
    - emphasizes attribute values at leaves (size coding)
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



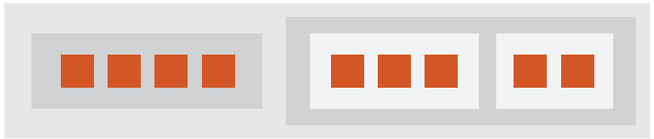
https://www.win.tue.nl/sequoiaview/

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

Enclosure

Containment Marks

✕ NETWORKS ✓ TREES



Idiom: implicit tree layouts (sunburst, icicle plot)

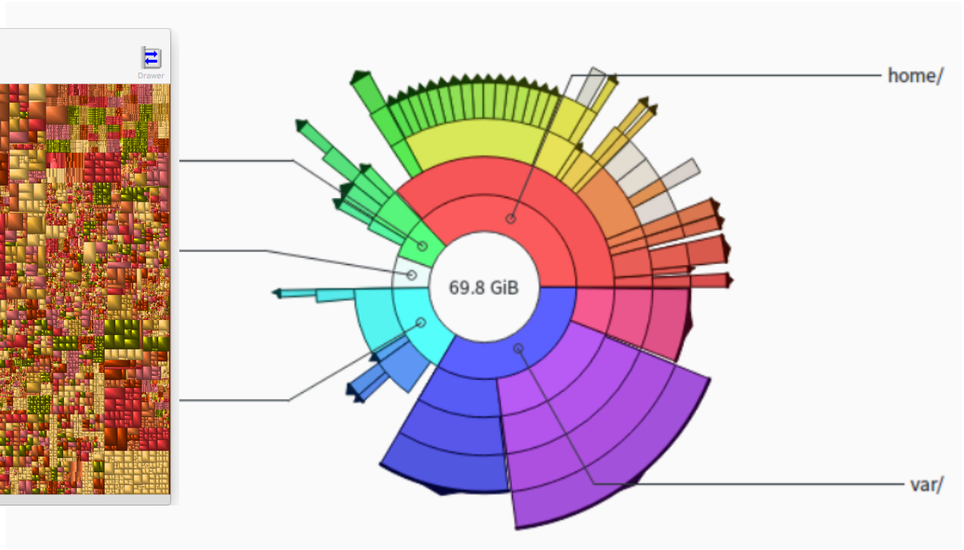
- alternative to connection and containment: position
  - show parent-child relationships only through relative positions

Treemap

containment

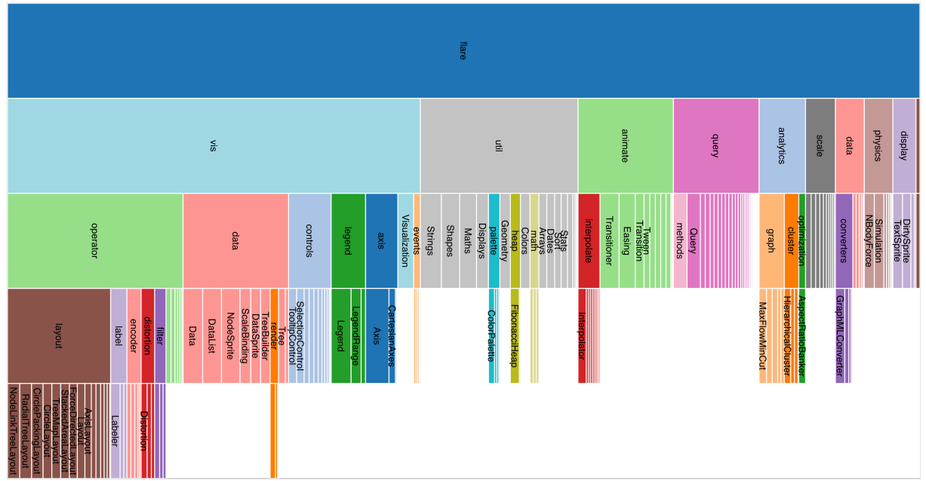
Sunburst

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

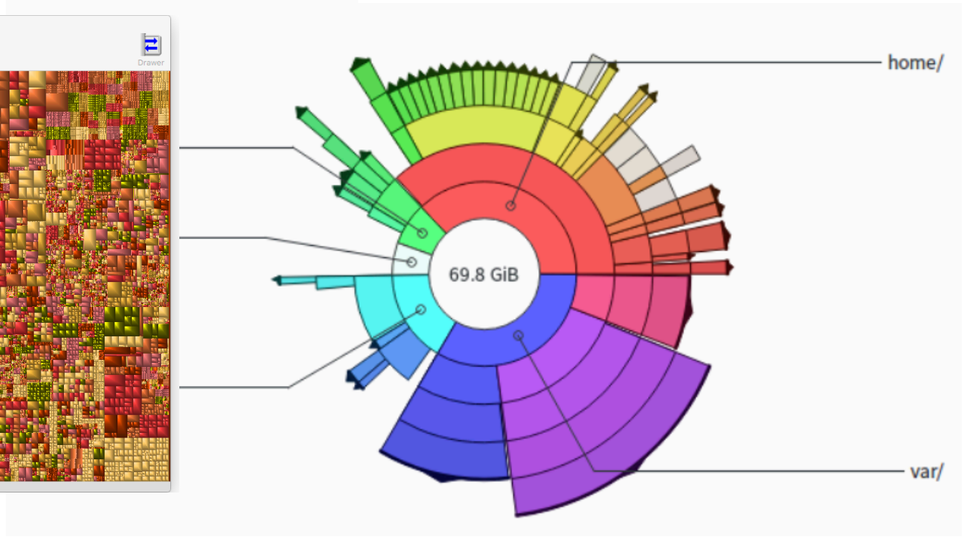
containment

only leaves visible

Sunburst

position (radial)

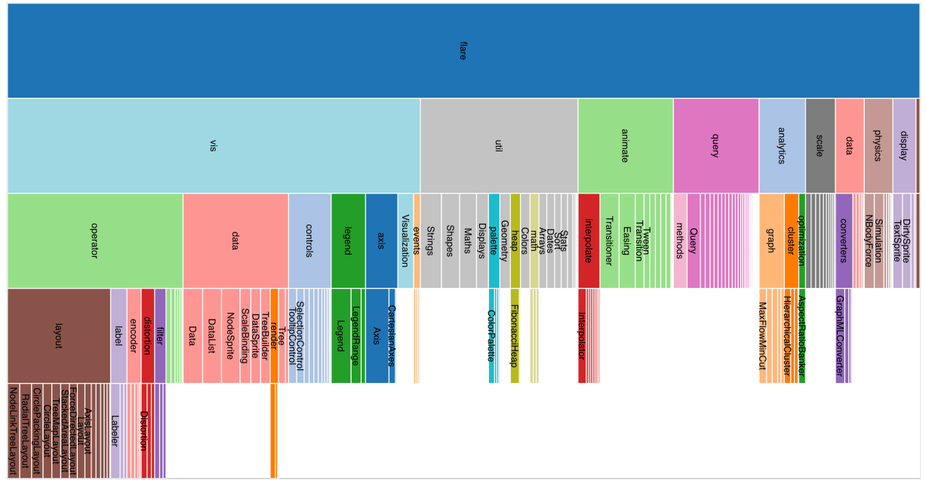
inner nodes & leaves visible



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

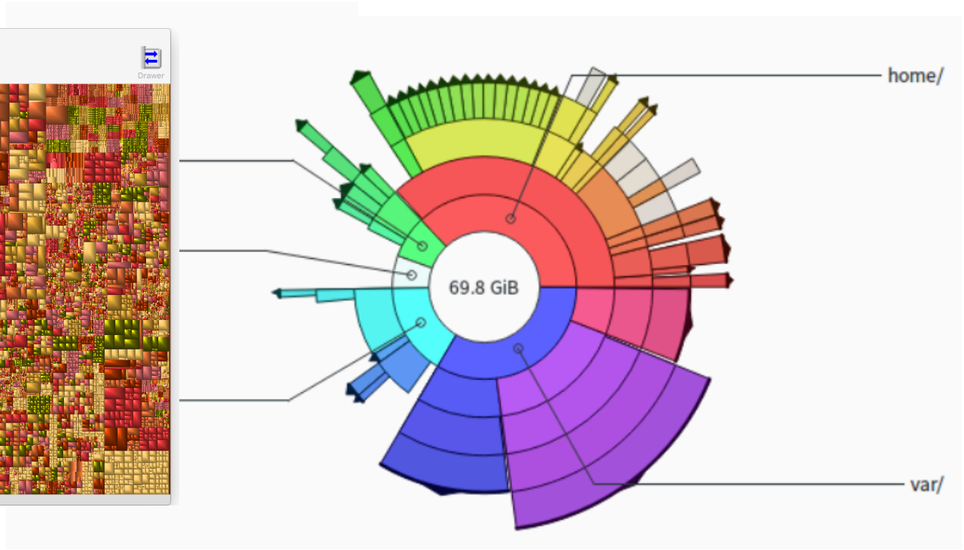
containment

only leaves visible

Sunburst

position (radial)

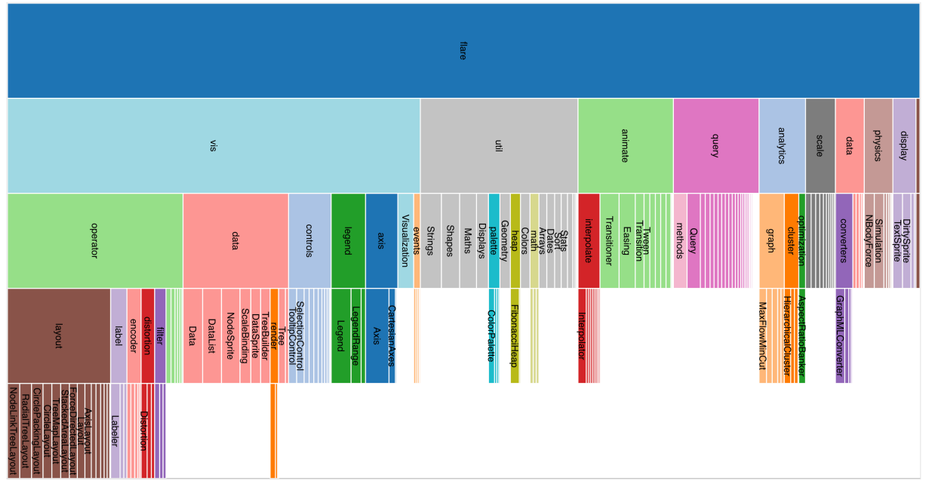
inner nodes & leaves visible



Icicle Plot

position (rectilinear)

inner nodes & leaves visible

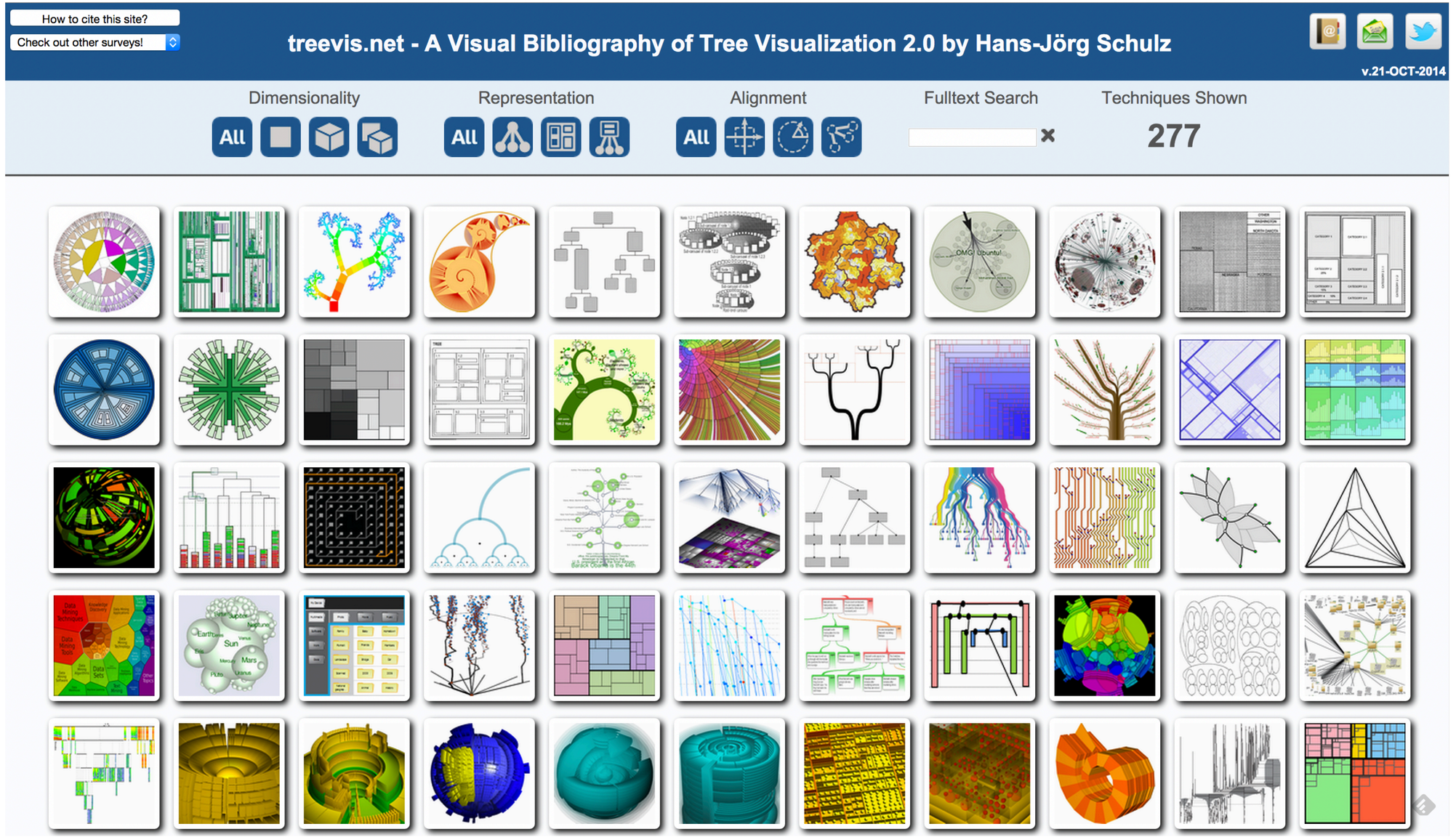


Implicit

Spatial Position

✕ NETWORKS ✓ TREES

treevis.net: Many, many options!



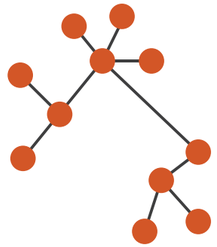
https://treevis.net/

Arrange networks and trees

Node–Link Diagrams

Connection Marks

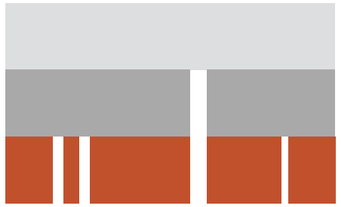
✔ NETWORKS ✔ TREES



Implicit

Spatial Position

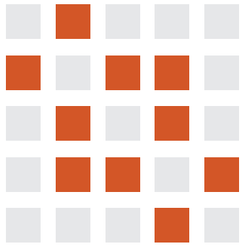
✘ NETWORKS ✔ TREES



Adjacency Matrix

Derived Table

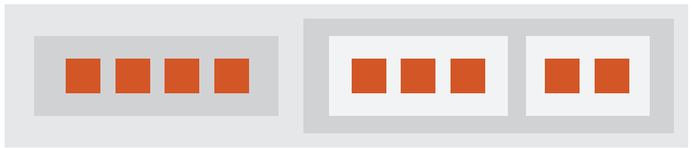
✔ NETWORKS ✔ TREES



Enclosure

Containment Marks

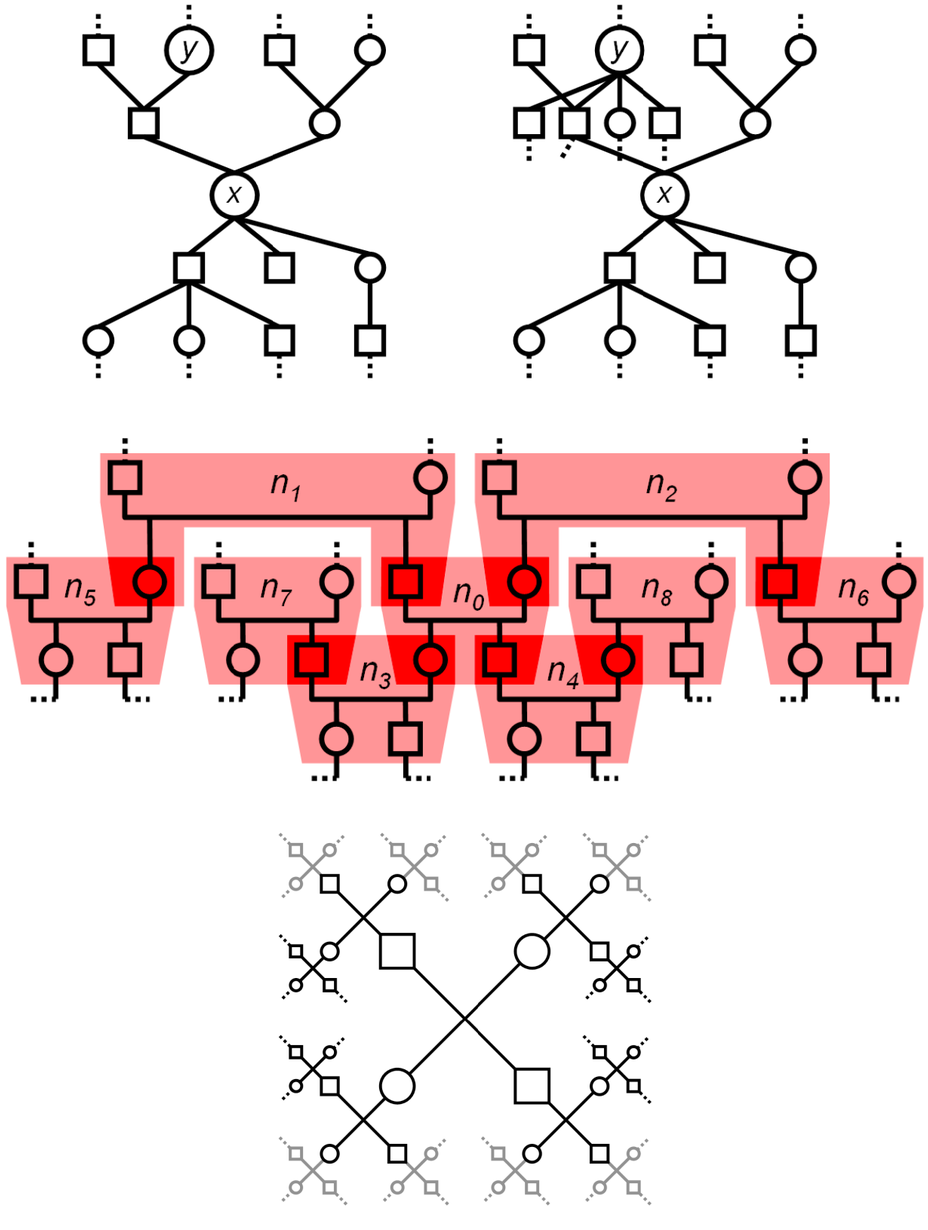
✘ NETWORKS ✔ TREES



Paper: Genealogical Graphs

Genealogical graphs: Technique paper

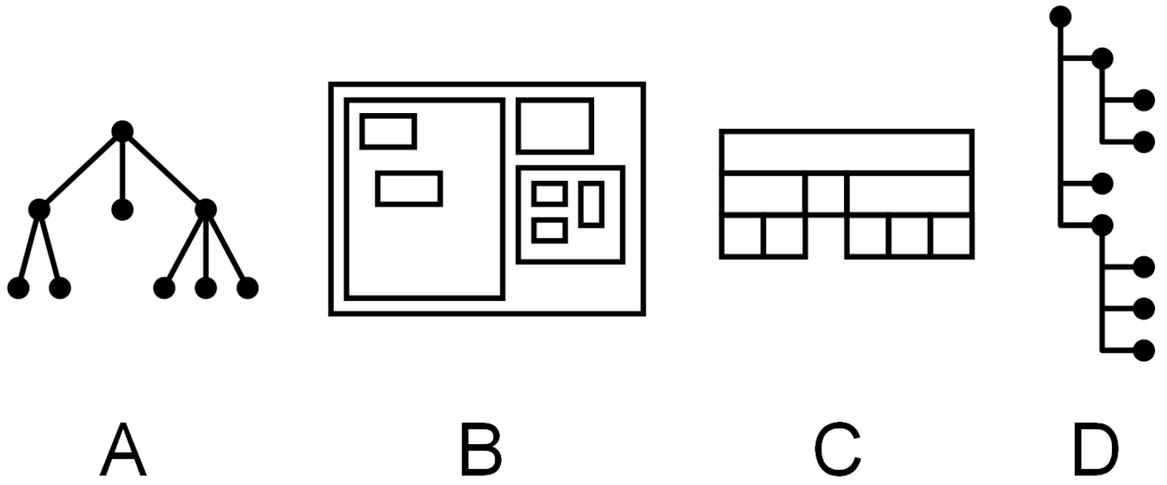
- motivation:
  extensive problem analysis!
- family tree is a misnomer
  - single person has tree of ancestors, tree of descendants
  - pedigree collapse inevitable
    - diamond in ancestor graph
- crowding problem
  - exponential
- fractal layout
  - poor info density
  - no spatial ordering for generations



[Fig 2, 6, 7. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Layouts

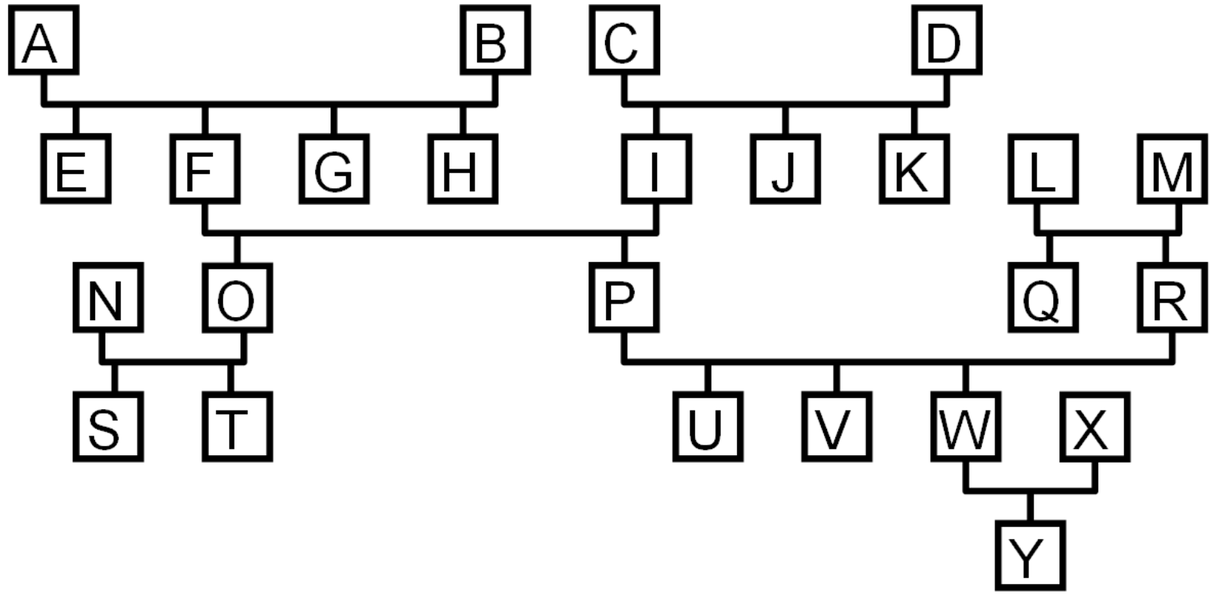
- rooted trees: standard layouts
  - connection
  - containment
  - adjacent aligned position
  - indented position



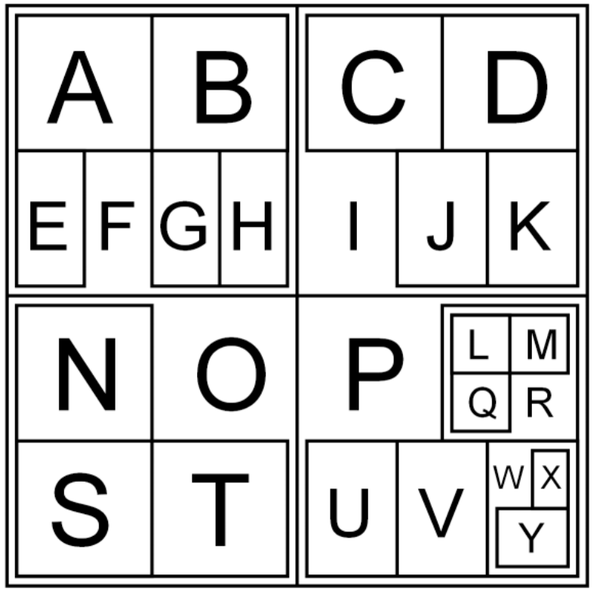
[Fig 8. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

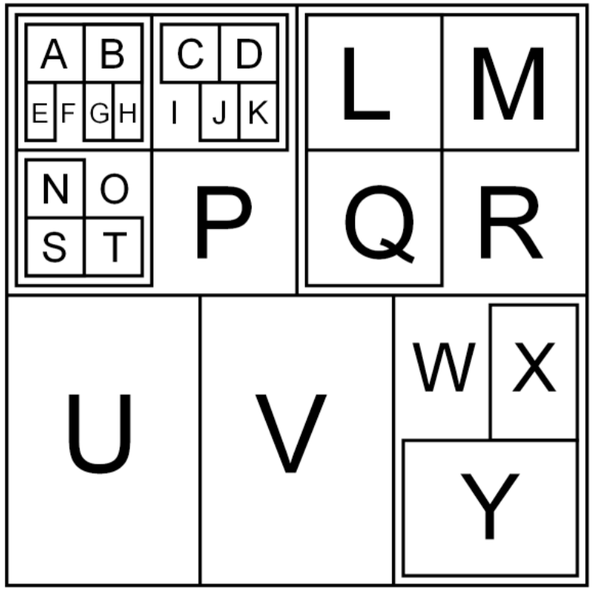
Layouts

- free trees
  - no root



- adapting rooted methods
  - temporary root for given focus
  - containment (nested)

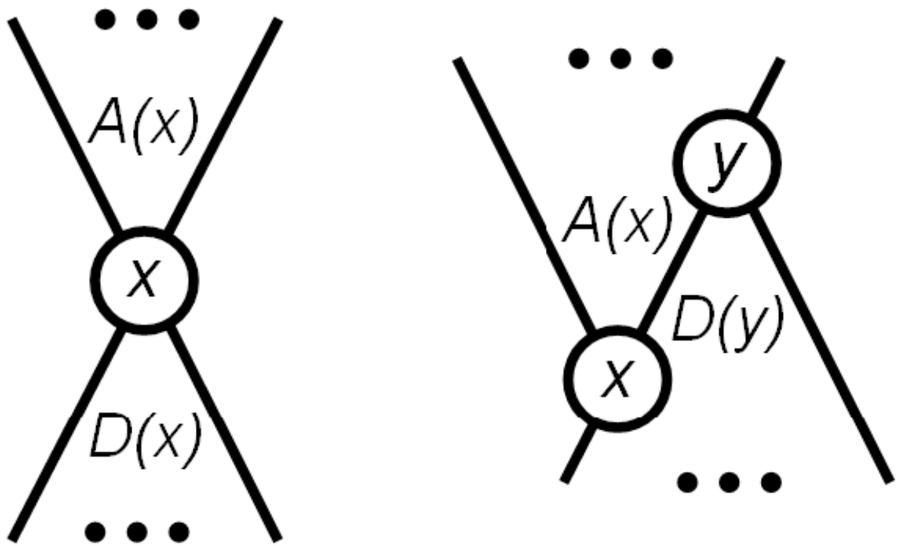


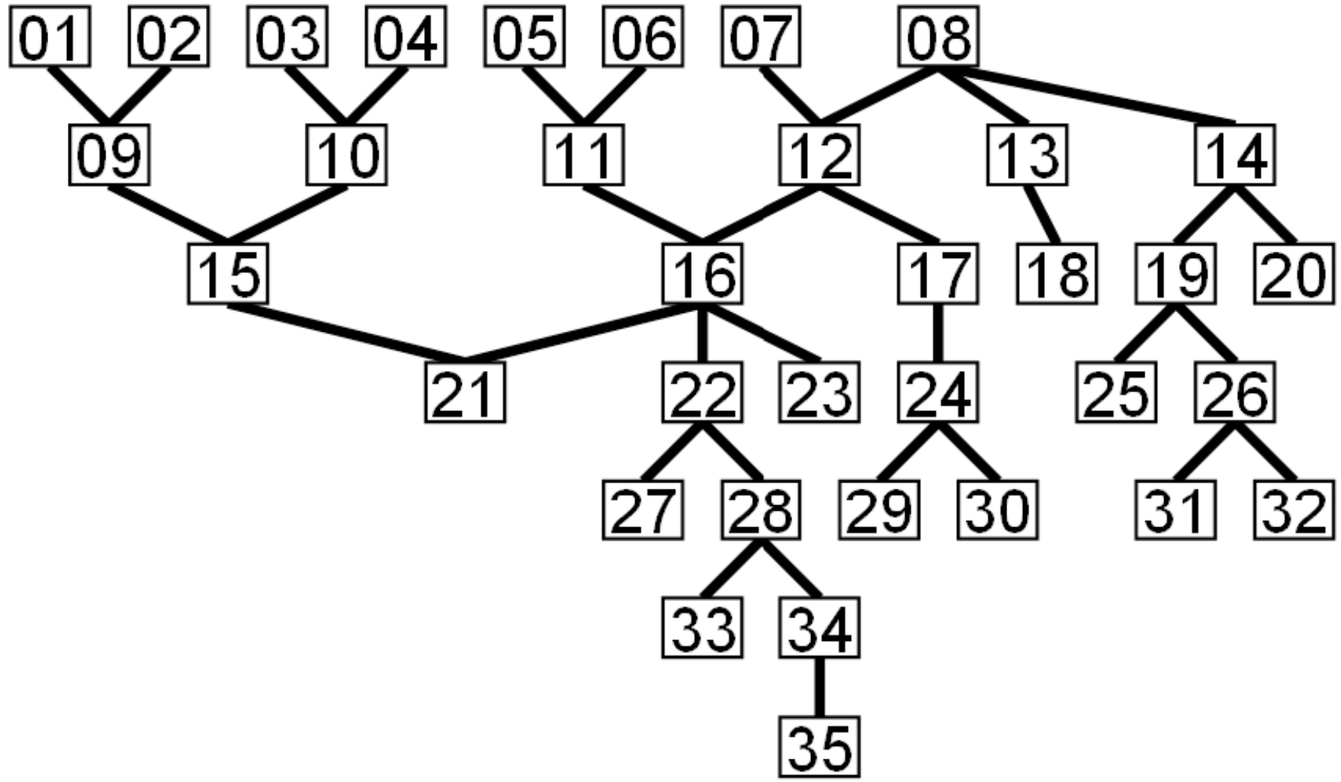


[Fig 9. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Dual trees: data abstraction **and** visual encoding approach

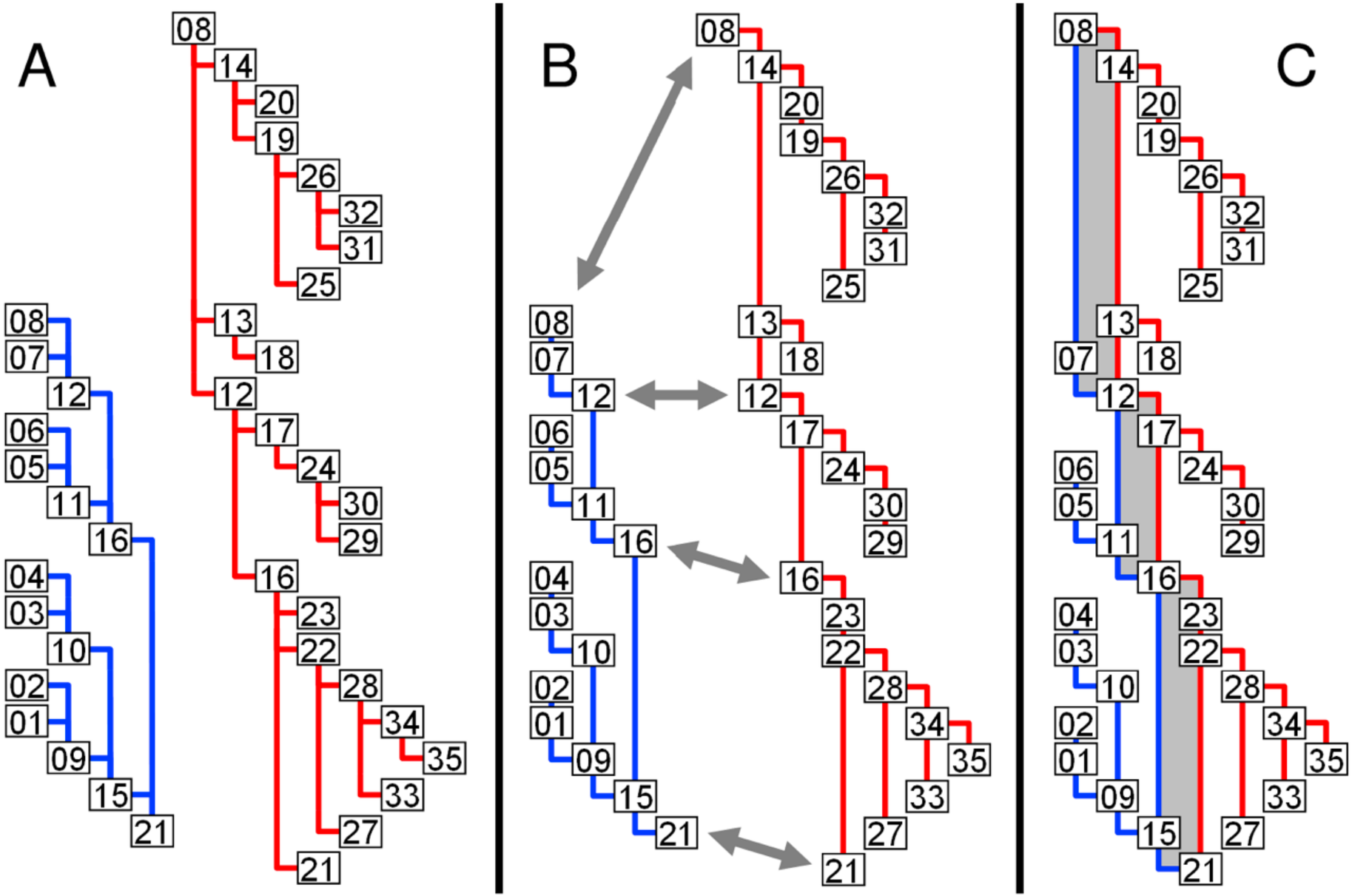
- explore canonical subsets and combinations, easy to interpret, scales well
- no crossings, nodes ordered by generation
- doubly rooted: x leftmost descend, y rightmost ancestor
  - offset roots from hourglass diagram





[Fig 10. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Indented, flipped, combined

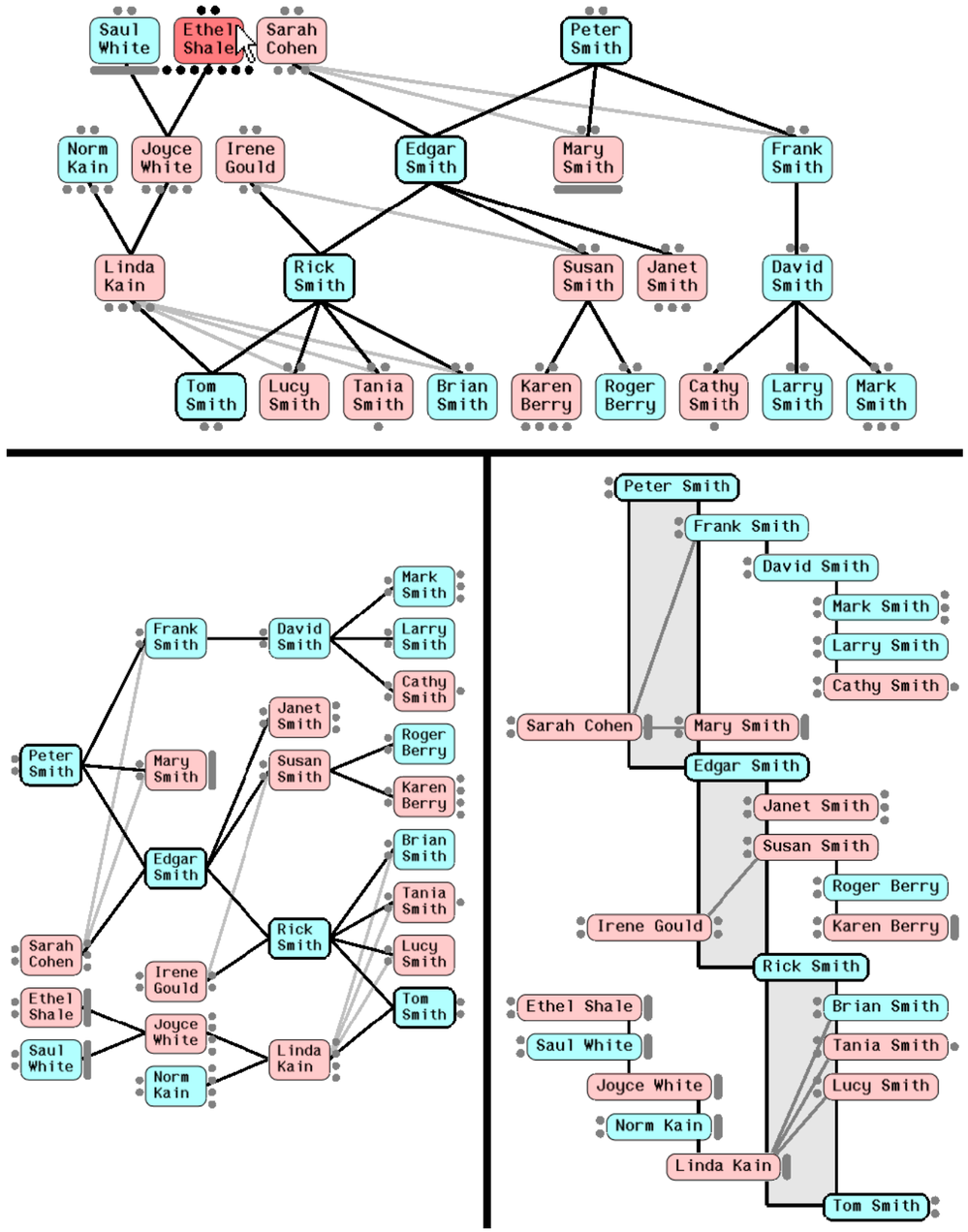


[Fig 11. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Another example

- vertical connection
- horizontal connection
- indented

- previously covered ideas
  - layering
  - aggregation



[Fig 13. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

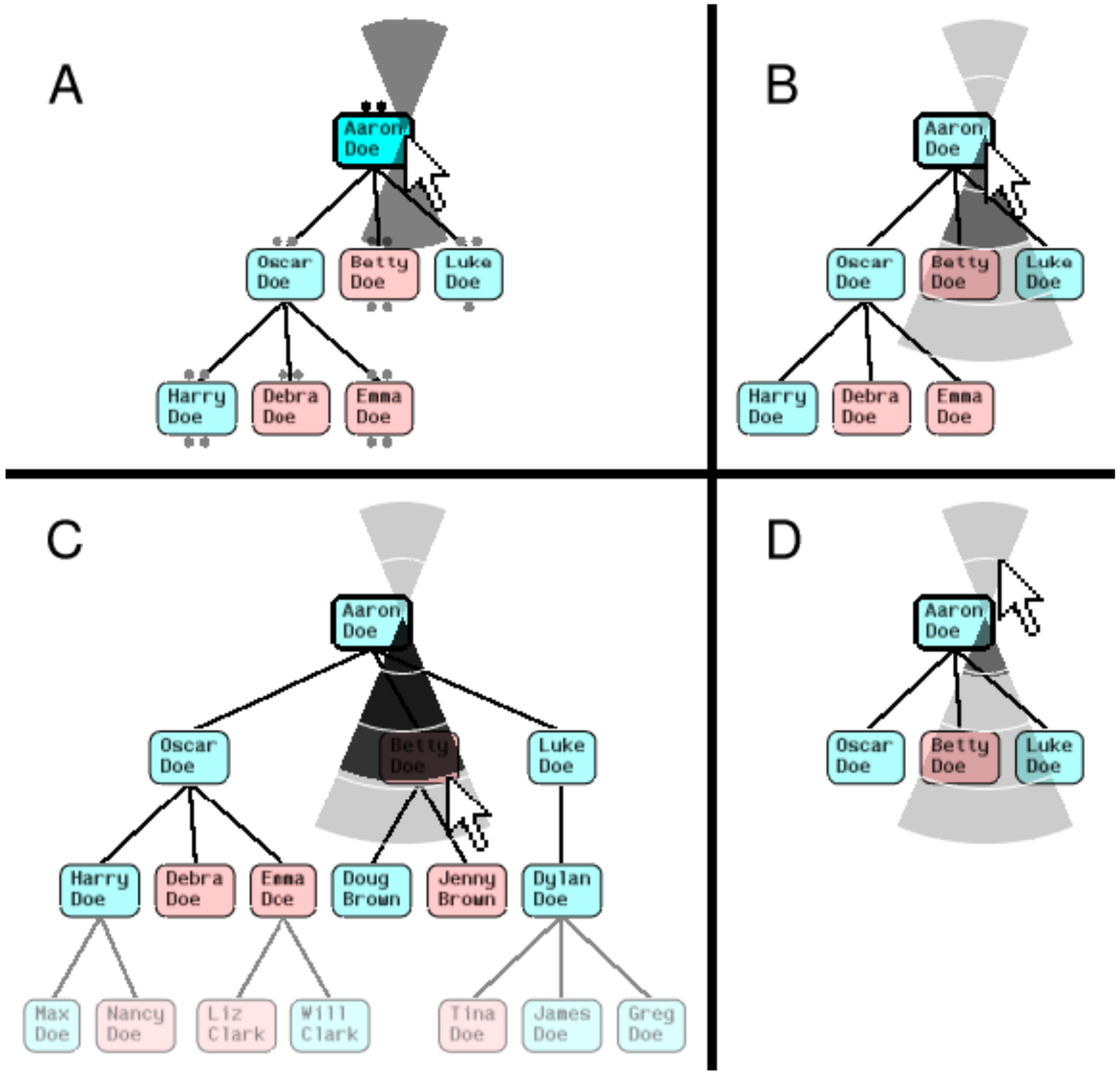
Interaction as fundamental to design

- navigation
  - topological navigation via collapse/expand on selection
    - parents, children
    - expand can trigger rotation
      - collapsing others
      - layout driven by navigation
  - geometric zoom/pan
  - constrained navigation: automatic camera framing
- animated transitions
  - 3 phases: fade out, move, fade in
- mouseover hover
  - preview dots: expand if collapsed

[Fig 14. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Custom widget

- popup marking menu
  - flick up or down, ballistic
  - subtree drag-out widget



[Fig 14. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

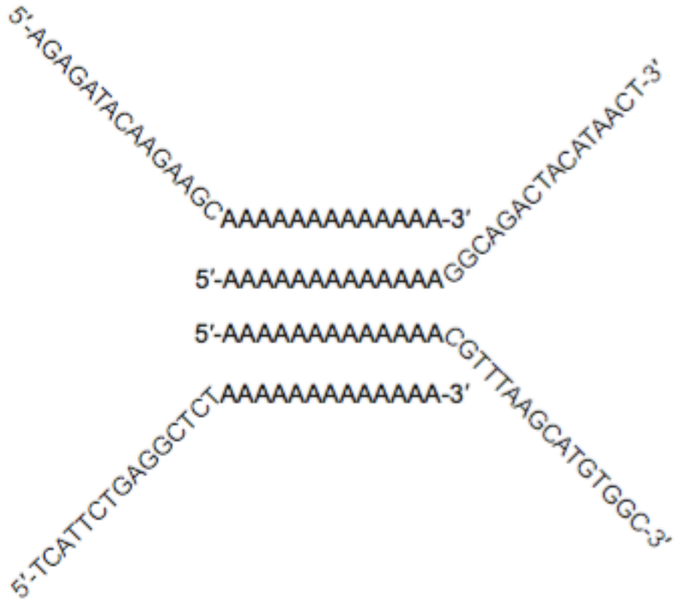
Evaluation approach

- framing: "initial user feedback"
  - domain expert usage & comments
  - less rigorous, eg not a semi-structure interview with script
- my take: cleared bar because of heft of other contributions
  - extensive initial analysis up front, vs extensive user testing at end

Paper: ABySS-Explorer

ABySS-Explorer: Design study

- reconstructing genome with ABySS algorithm (Assembly By Short Sequences)
- domain task
  - go from short subsequences to ***contigs***, long contiguous sequences
  - extensive automatic support, but still human in the loop for visual inspection and manual editing
  - ambiguities, like repetitions longer than read length
- data, domain:abstract
  - millions of reads of 25-100 nucleotides (nt): strings
  - read coverage, proxy for quality: quant attrib
  - read pairing distances, proxy for size distribution: quant



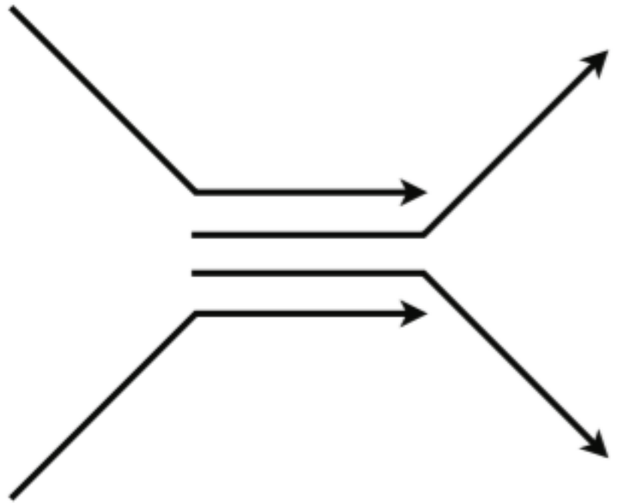
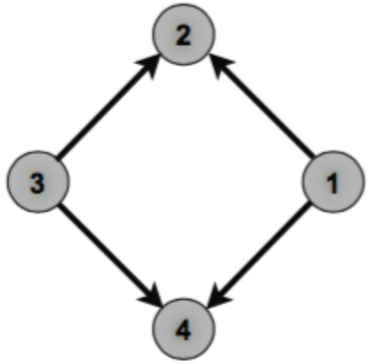
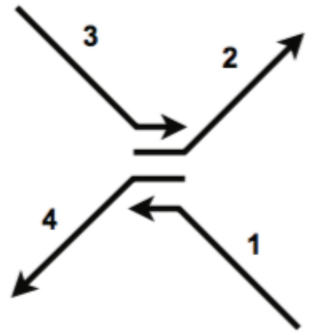


Fig 2. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

Contigs: abstraction as derived network data

- derived data: de Bruijn graph/network
  - directed network, compact representation of sequence overlaps
  - node: contig
  - edge: overlap of k − 1 nt between two contigs
  - good for computing, bad for reasoning about sequence space
- derived data: dual de Bruijn graph
  - node: points of contig overlap
  - edge: contig
  - better match for arrow diagrams used in hand drawn sketches
- base layout: force-directed





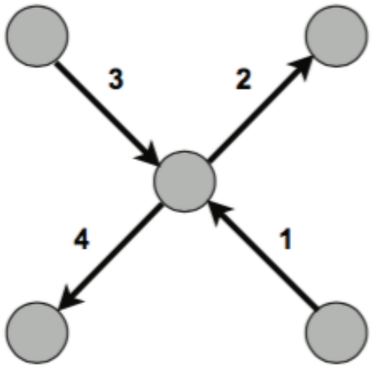


Fig 3. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

DNA as double stranded: idiom for encoding & interaction

- rejected option: 2 nodes per contig
  - excess clutter if one for each direction
  - choice at data abstraction level
- encoding & interaction idiom: *polar* node
  - encoding: upper vs lower attachment point
    - redundant with arc direction
      - large-scale visibility, without need to zoom
    - arbitrary but consistent
  - interaction: click to reverse direction
    - switches polarity of vertex connections
    - changes sign of label

Fig 4. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

Contig length: encoding

- rejected option: scale edge lengths by sequence lengths
  - short contigs are important sources of ambiguity, would be hard to distinguish
  - task guidance: only low-res judgements needed, relatively long or short
- encoding idiom: wave pattern
  - oscillation shows fixed number, shapes distinguishable
  - min amplitude at connections so edges visible
  - orientation with max amplitude asymmetric wrt start
    - rejected initial option: max in middle
    - rejected options:
      - color (keep for other attribute)
      - half-lines
      - curvature (used for polar nodes)
    - aligned with empirical guidance for tapered edges

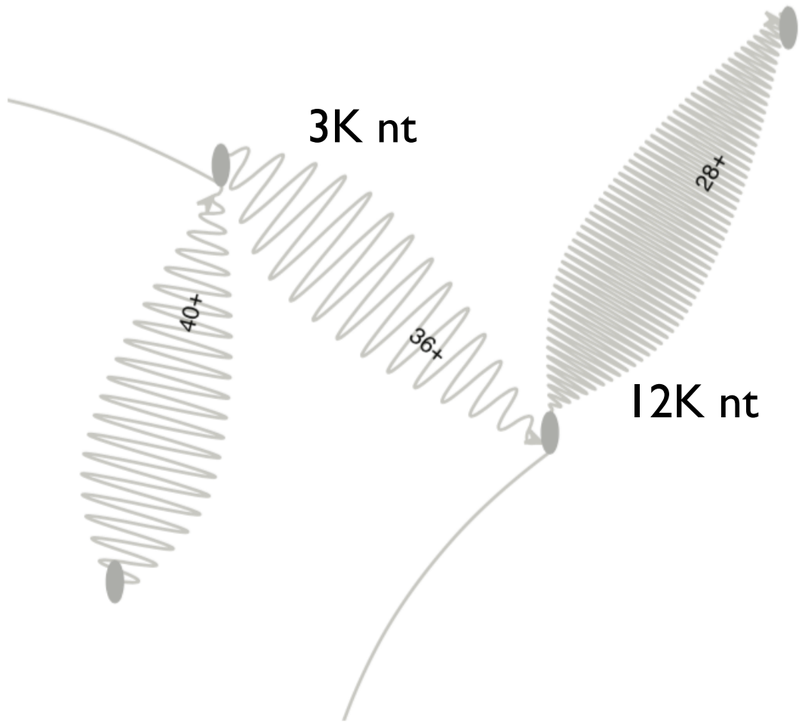


Fig 5. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

Contig coverage: encoding

- rejected options: luminance/lightness
  - not distinguishable given denseness variation from wave shapes
  - also problematic with desire for separable color/hue encoding
- chosen: line thickness
  - not distinguishable for extremely long contigs
  - can address by adjusting oscillation frequency to suitable size

Read pairs: encoding

- data:
  - distance estimate
  - orientation
- encoding:
  - dashed line (shape channel for line mark)
    - implying inferred vs observed sequences
  - color for both dashed line and contig leaf
  - [same length as for contigs]
  - rejected initial option: line color alone
    - too ambiguous
  - interaction to fully resolve remaining ambiguity
  - or color by unambiguous paths in grey

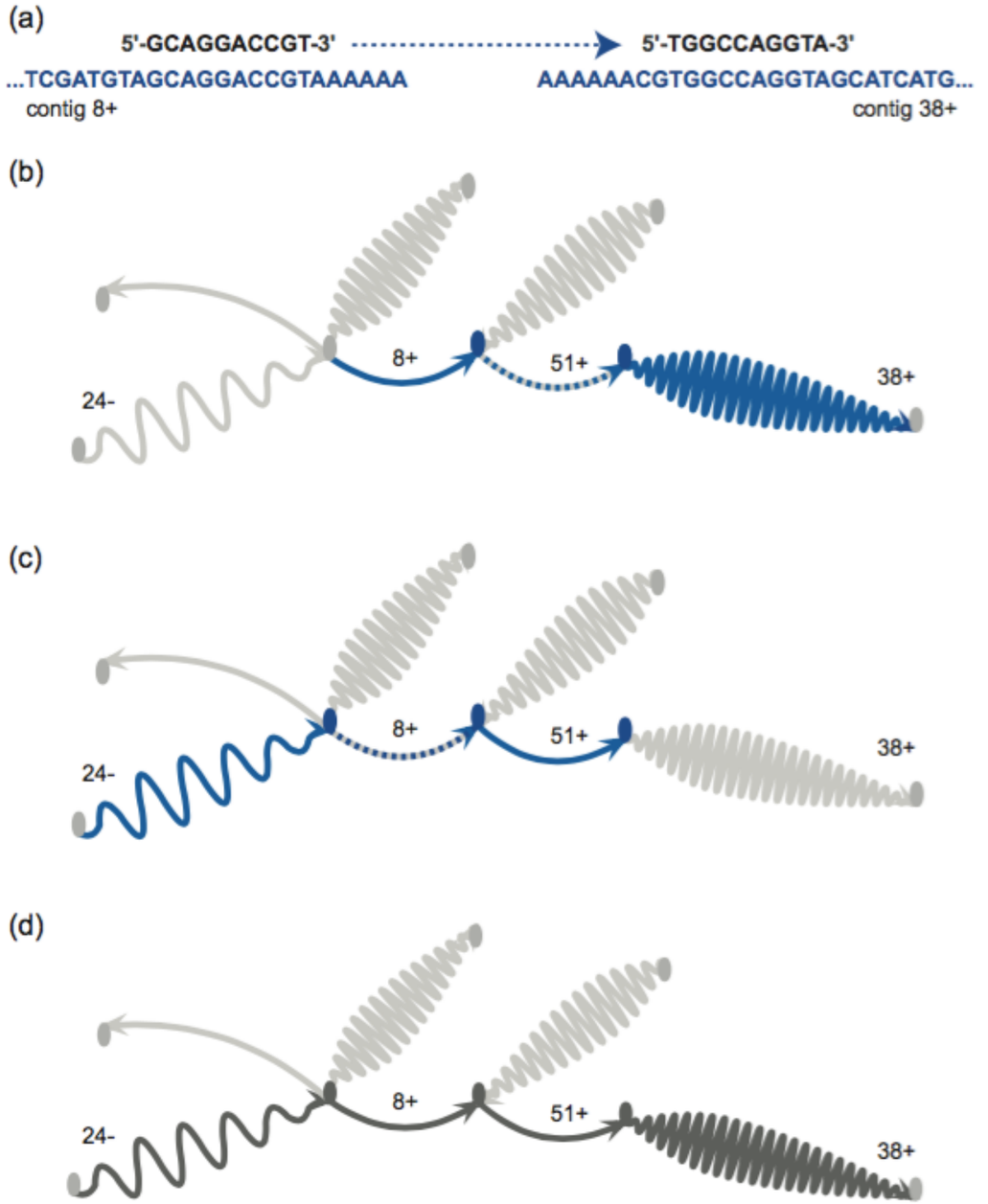
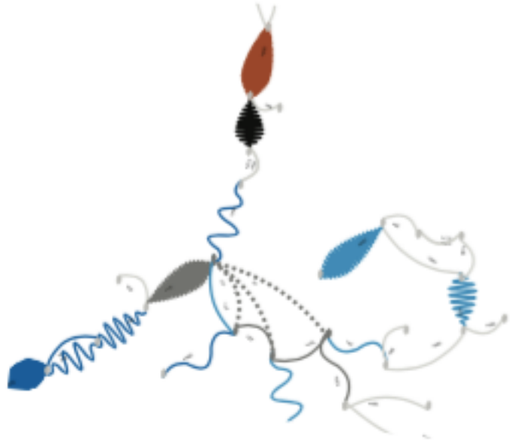


Fig 6. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

Displaying meta-data

- reserve color for additional attributes
- ex: color to compare reference human to lymphoma genome
  - inconsistencies visible as interconnections between different colors
  - inversion breakpoint visible
  - interaction to check if error in metadata from experiments vs assembly
    - read pair info supports metadata
      - speedup claim vs prev work



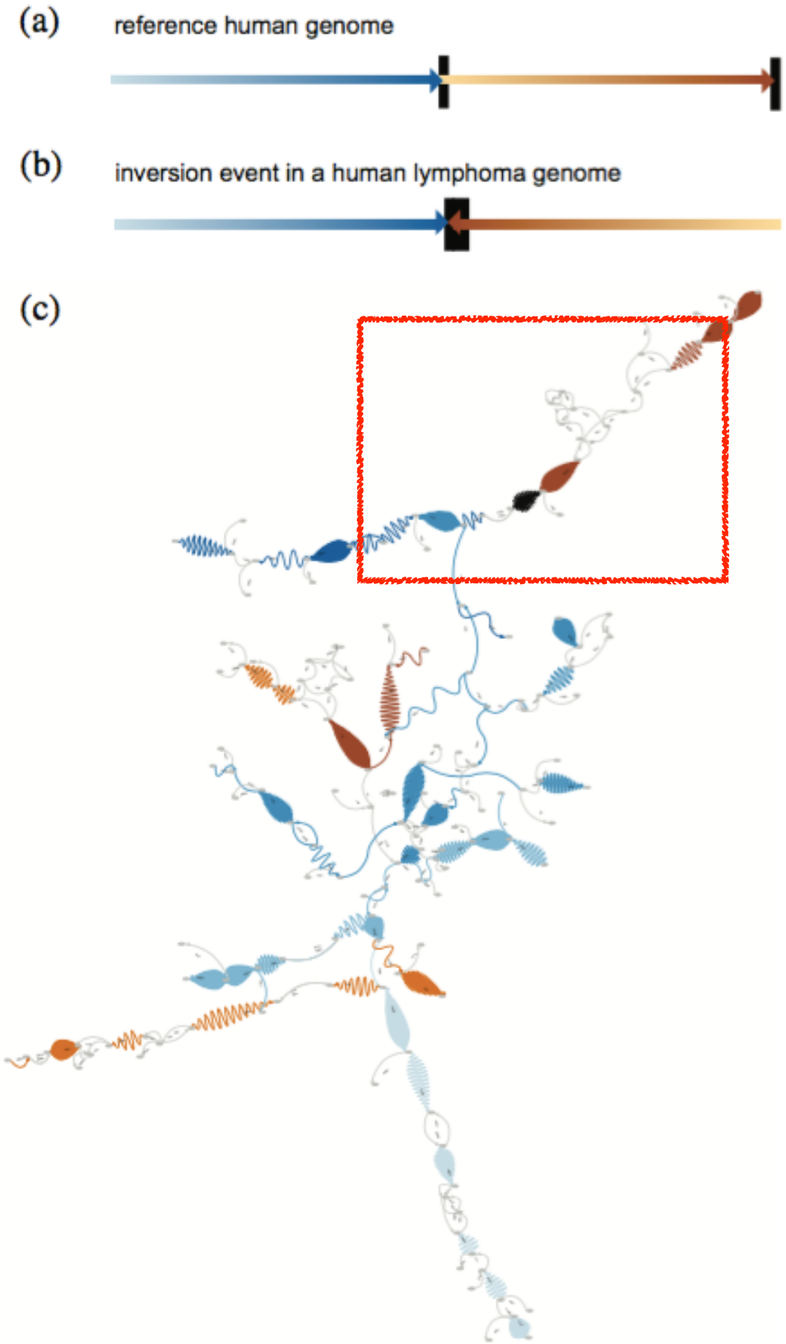


Fig 10. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

Assembly examples

- ideal: single large contig
  - overview/gist: many small contigs remain
- interaction to resolve
  - integrate paired read highlighting on top of contig paths structure

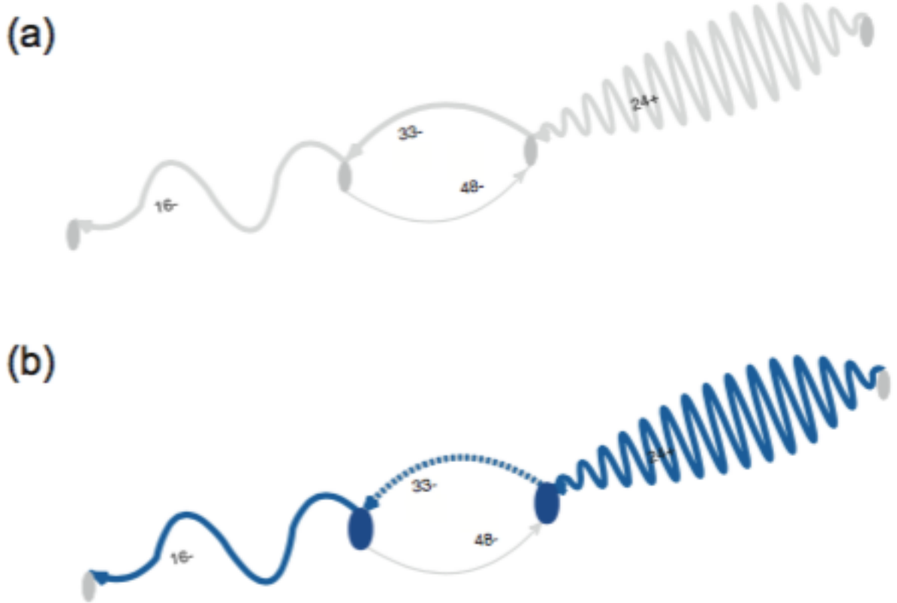


Fig 7/9. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

Evaluation

- more implicit than other design studies
  - researcher embedded with domain scientists, vs reaching out across the wall
  - reporting on applications, rather than in-depth case studies of deployed systems
  -

Things to note: both papers

- data abstraction :: visual encoding relationship
  - novelty for abstraction underlies novelty of visual encoding
- extensive discussion of alternatives as framing/justification for design

Q&A Roundup

Break

Project Team lonl Followups