Arrange networks and trees

- **Node-link Diagrams**
  - Connections and Marks
  - NETWORKS, TREES

- **Adjacency Matrix**
  - Derived Table
  - NETWORKS, TREES

- **Enclosure**
  - Containment Marks
  - NETWORKS, TREES
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 $E < 4N$

[Link to diagram and example](http://mbostock.github.com/d3/ex/force.html)
Idiom: \textbf{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
    • (more on algorithm vs encoding in afternoon)

• scalability
  – nodes, edges: 1K-10K
  – hairball problem eventually hits


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

![Matrix view of a five-node network](image)

**Figure 7.5:** Comparing matrix and node-link views of a five-node network.

a) Matrix view. b) Node-link view. From [Henry et al. 07], Figure 3b and 3a.

(Permission needed.)

Network matrix views can achieve very high information density, up to a limit of one thousand nodes and one million edges, just like cluster heatmaps and all other matrix views that uses small area marks.

**Network matrix view**

**Data Types**
- **network**

**Derived Data**
- table: network nodes as keys, link status between two nodes as values

**View Comp.**
- space: area marks in 2D matrix alignment

**Scalability**
- nodes: 1K
- edges: 1M

---


Connection vs. adjacency comparison

- adjacency matrix strengths
  - predictability, scalability, supports reordering
  - some topology tasks trainable

- node-link diagram strengths
  - topology understanding, path tracing
  - intuitive, no training needed

- empirical study
  - node-link best for small networks
  - matrix best for large networks
    - if tasks don’t involve topological structure!

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

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

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

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

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

---

Further reading

  – Chap 9: Arrange Networks and Trees


Topological Fisheye Views

• derived data
  – input: laid-out network (spatial positions for nodes)
  – output: multilevel hierarchy from graph coarsening

• interaction
  – user changed selected focus point

• visual encoding
  – hybrid view made from cut through several hierarchy levels

Topological Fisheye Views

• derived data
  – input: laid-out network (spatial positions for nodes)
  – output: multilevel hierarchy from graph coarsening

• interaction
  – user changed selected focus point

• visual encoding
  – hybrid view made from cut through several hierarchy levels

[Fig 4,8. Topological Fisheye Views for Visualizing Large Graphs. Gansner, Koren and North, IEEE TVCG 11(4), p 457-468, 2005]
Coarsening requirements

• uniform cluster/metanode size
• match coarse and fine layout geometries
• scalable

Coarsening strategy

• must preserve graph-theoretic properties
• use both topology and geometry
  – topological distance (hops away)
  – geometric distance - but not just proximity alone!
    • just contracting nodes/edges could create new cycles
• derived data: proximity graph

what not to do!

2-D point set  Delaunay triangulation  relative neighborhood graph

[Fig 10, 12. Topological Fisheye Views for Visualizing Large Graphs. Gansner, Koren and North, IEEE TVCG 11(4), p 457-468, 2005]
Candidate pairs: neighbors in original and proximity graph

• proximity graph: compromise between larger DT and smaller RNG
  – better than original graph neighbors alone
    • slow for cases like star graph

• maximize weighted sum of
  – geometric proximity
    • goal: preserve geometry
  – cluster size
    • goal: keep uniform cluster size
  – normalized connection strength
    • goal: preserve topology
  – neighborhood similarity
    • goal: preserve topology
  – degree
    • goal: penalize high-degree nodes to avoid salient artifacts and computational problems
Hybrid graph creation

- cut through coarsening hierarchy to get active nodes
  - animated transitions between states

[Fig 10, 12. Topological Fisheye Views for Visualizing Large Graphs. Gansner, Koren and North, IEEE TVCG 11(4), p 457-468, 2005]
Final distortion

• geometric distortion for uniform density
• (colorcoded by hierarchy depth just to illustrate algorithm)
  – compare to original
  – compare to simple topologically unaware fisheye distortion

[Fig 2,15. Topological Fisheye Views for Visualizing Large Graphs. Gansner, Koren and North, IEEE TVCG 11(4), p 457-468, 2005]