Bioinformatics II
Winter Term 2016/17

universitätbonn
Rheinische
Friedrich-Wilhelms-
Universität Bonn

Chapter 5:
Graphs and Networks

Jun.-Prof. Dr.-Ing. Thomas Schultz

URL: http://cg.cs.uni-bonn.de
E-Mail: schultz@cs.uni-bonn.de
Office: Friedrich-Ebert-Allee 144, 53113 Bonn

November 29, 2016

Protein-Protein Interactions in Yeast

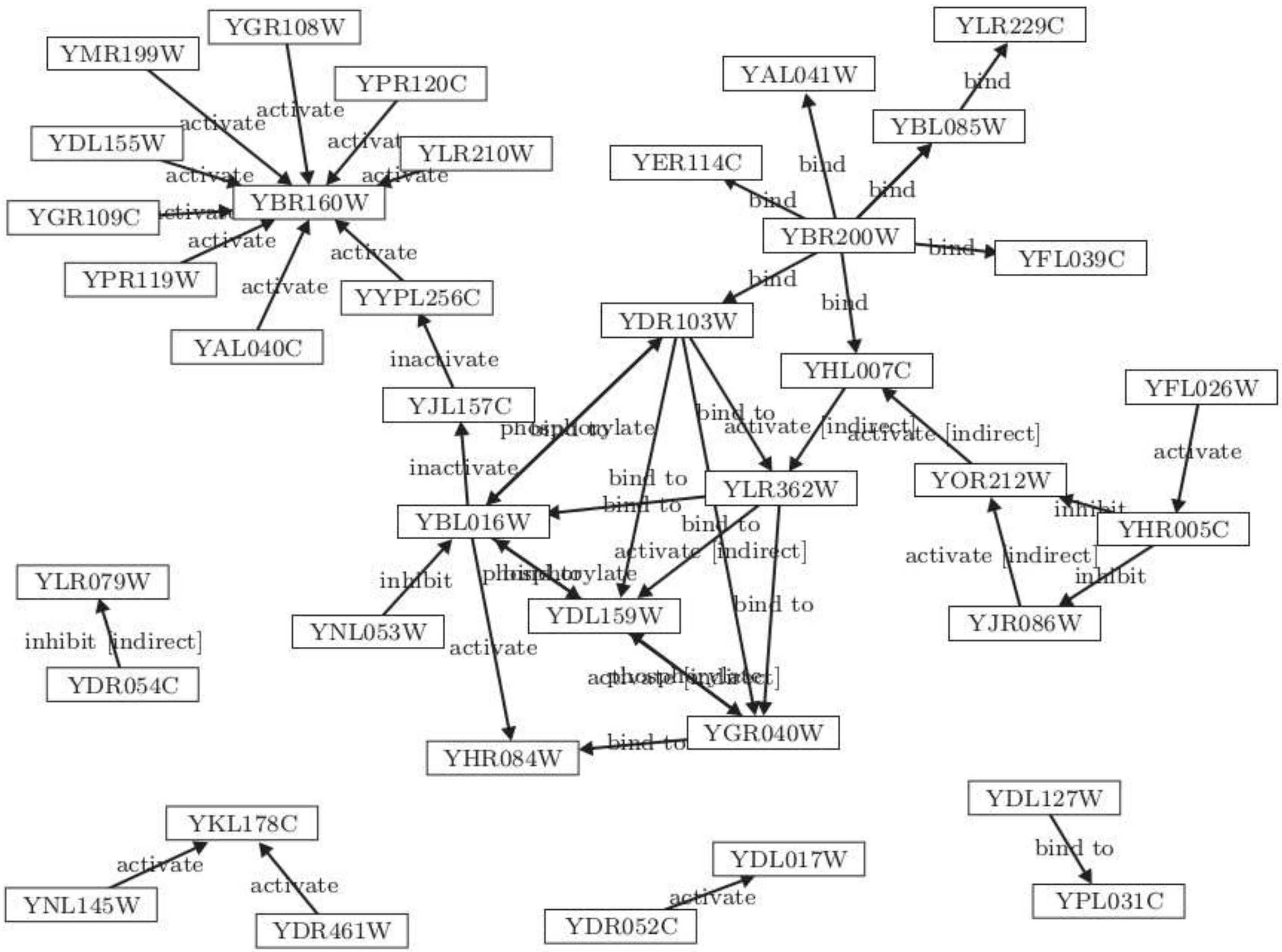
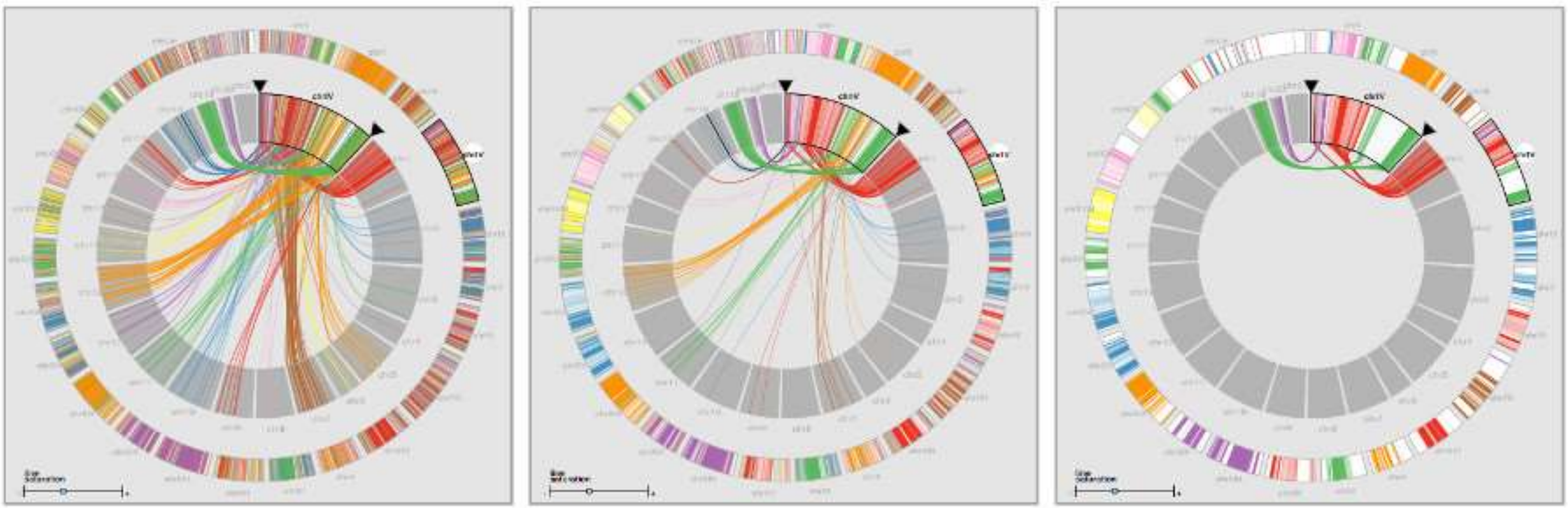


Image from [Bachmaier et al. 2013]

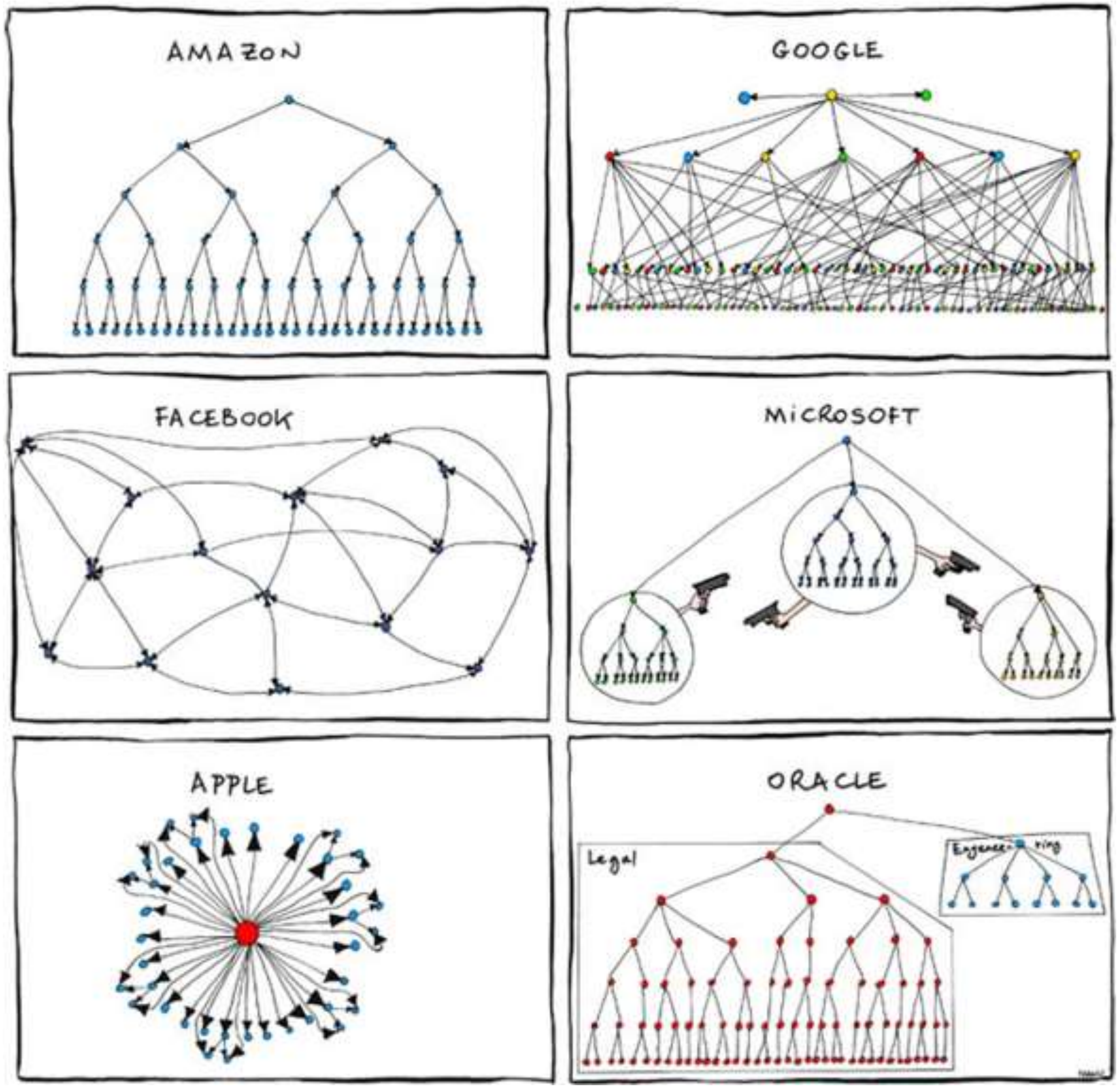
Comparative Genomics

- [Meyer et al. 2009]: Synteny (conservation of genomic features) between genomes



Organization Charts

- By Manu Cornet



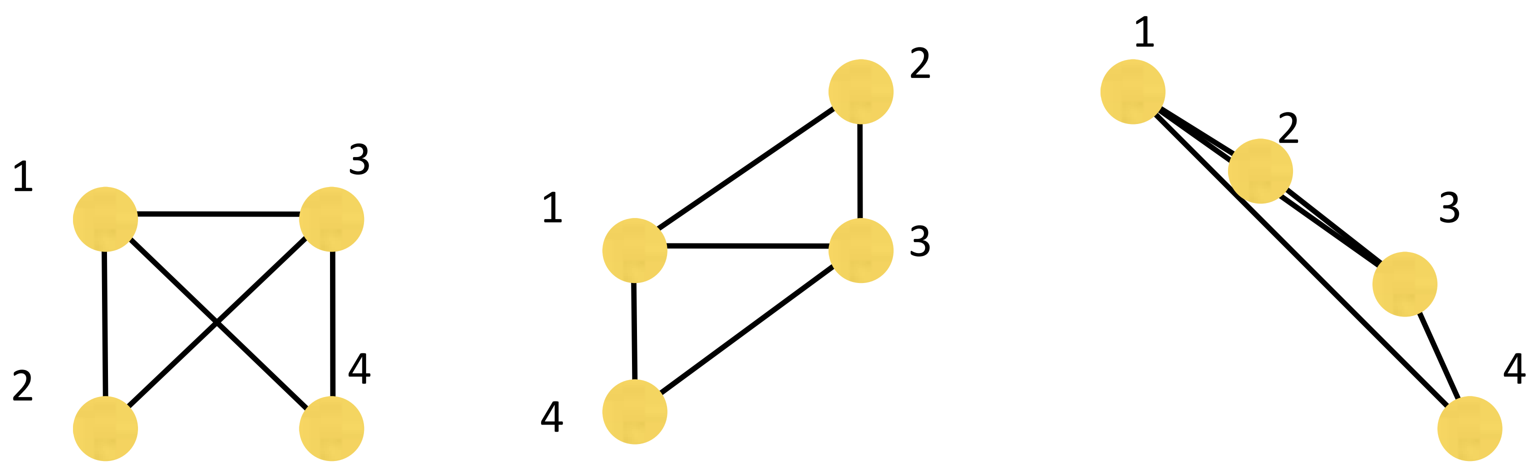

facebook

Section 5.1:
Basic Definitions

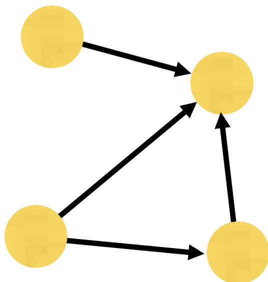
Graphs & Trees: Definitions

- A graph G consists of
  - a collection of nodes (or vertices) V
  - a set of edges E, consisting of vertex pairs.
- An edge $e_{xy} = (x,y)$ connects two nodes x and y.

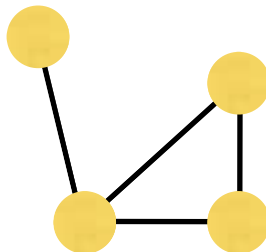
– Example: V={1,2,3,4}, E={(1,2),(1,3),(2,3),(3,4),(4,1)}



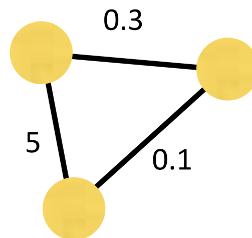
Graphs & Trees: Definitions



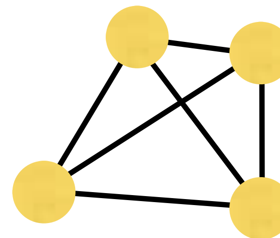
A directed graph



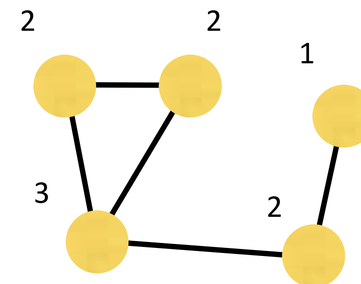
An undirected graph



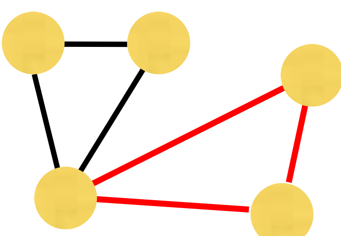
Weighted



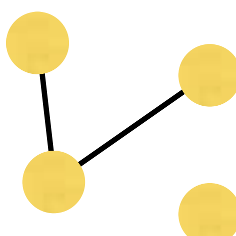
Complete



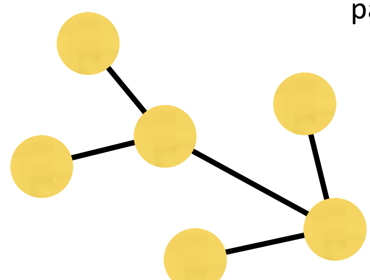
Node degrees



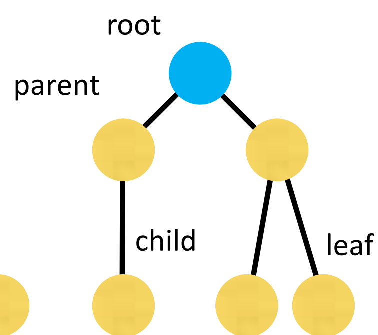
A cycle



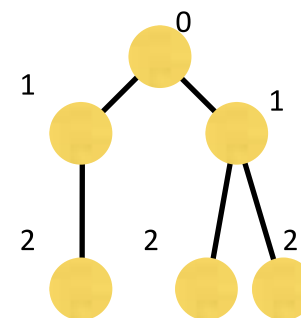
An acyclic graph



A connected acyclic graph, a.k.a. a tree



A rooted tree or hierarchy

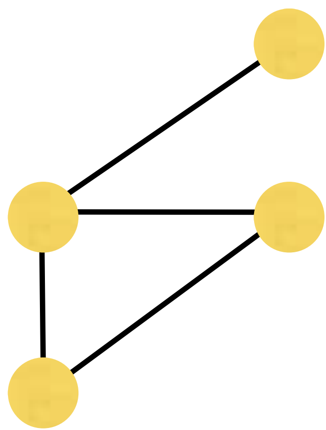


Node depths

Graphs & Trees: Definitions

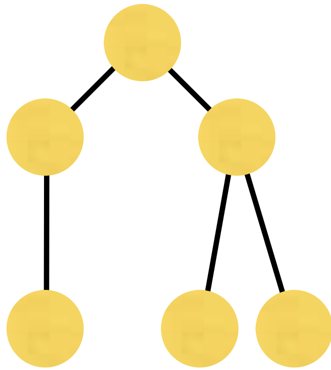
Graphs

- models relations among data
- *nodes* and *edges*



Trees

- graphs with a hierarchical structure
  - connected graph with n-1 edges
- nodes as *parents* and *children*



Graphs & Trees: Definitions

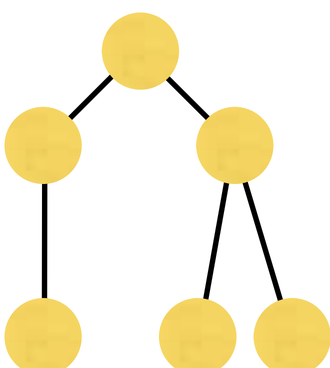
- Primary concern of graph drawing is the spatial layout of nodes and edges
- Often, the goal is to effectively depict the graph structure
  - connectivity, path-following
  - network distance
  - clustering
  - ordering (e.g., hierarchy level)

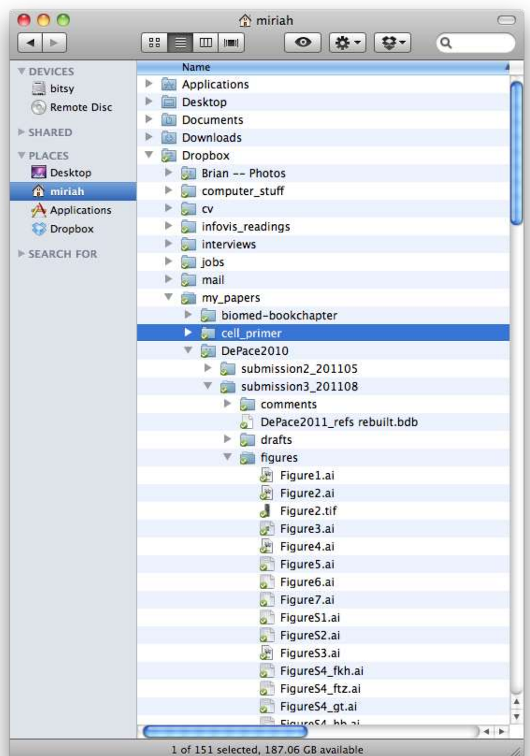
Section 5.2:
Visualizing Trees / Hierarchies

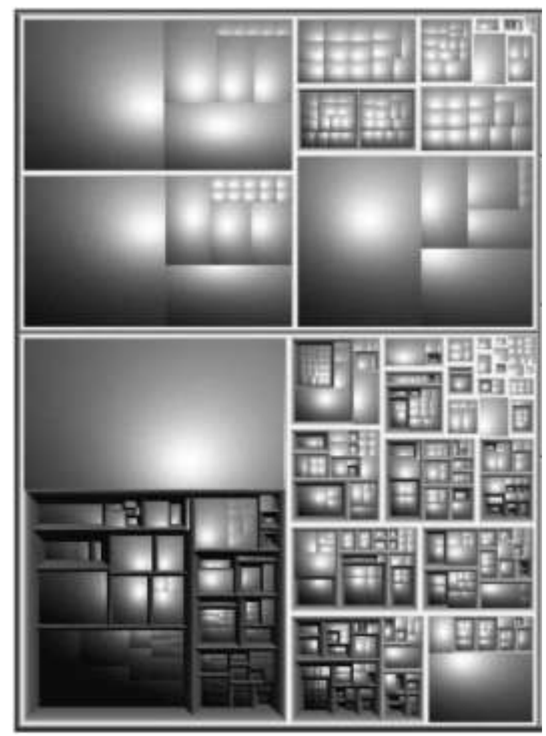
Visualizing Trees

Rooted trees

- Recursion makes it elegant and fast to draw trees
- Approaches:
  - node link
  - indentation
  - enclosure (treemaps)

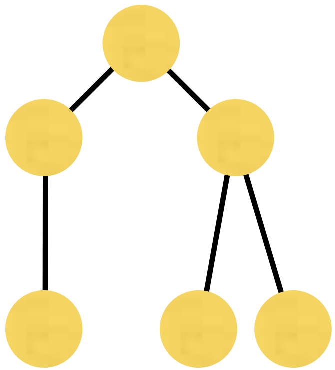






Visualizing Trees: Node-Link Diagrams

- Nodes are distributed in space, connected by straight or curved lines
- Typical approach is to use 2D space to break apart breadth and depth
- Frequent **design goals:**
  - Nodes at same depth share the same vertical position
  - Horizontal whitespace communicates hierarchy
  - Minimize required area
  - Minimize total length of edges
  - Achieve good aspect ratio



Aesthetics of Reingold-Tilford

- *Tidier Drawings of Trees* [Reingold/Tilford 1981]
  – Formulation for binary trees, can be generalized
- **Aesthetic Goals:**
  – Nodes at the same level should be aligned
  – Maintain the relative ordering of left and right subtrees
  – Parent should be centered over the children
  – A tree and its mirror image should be drawn as reflections of each other
  – A subtree should be drawn the same way regardless of where it occurs in the tree

Reingold-Tilford: Recursive Construction

- Assume left and right subtrees have already been drawn
- Shift them to a fixed horizontal distance
- Center parent between them

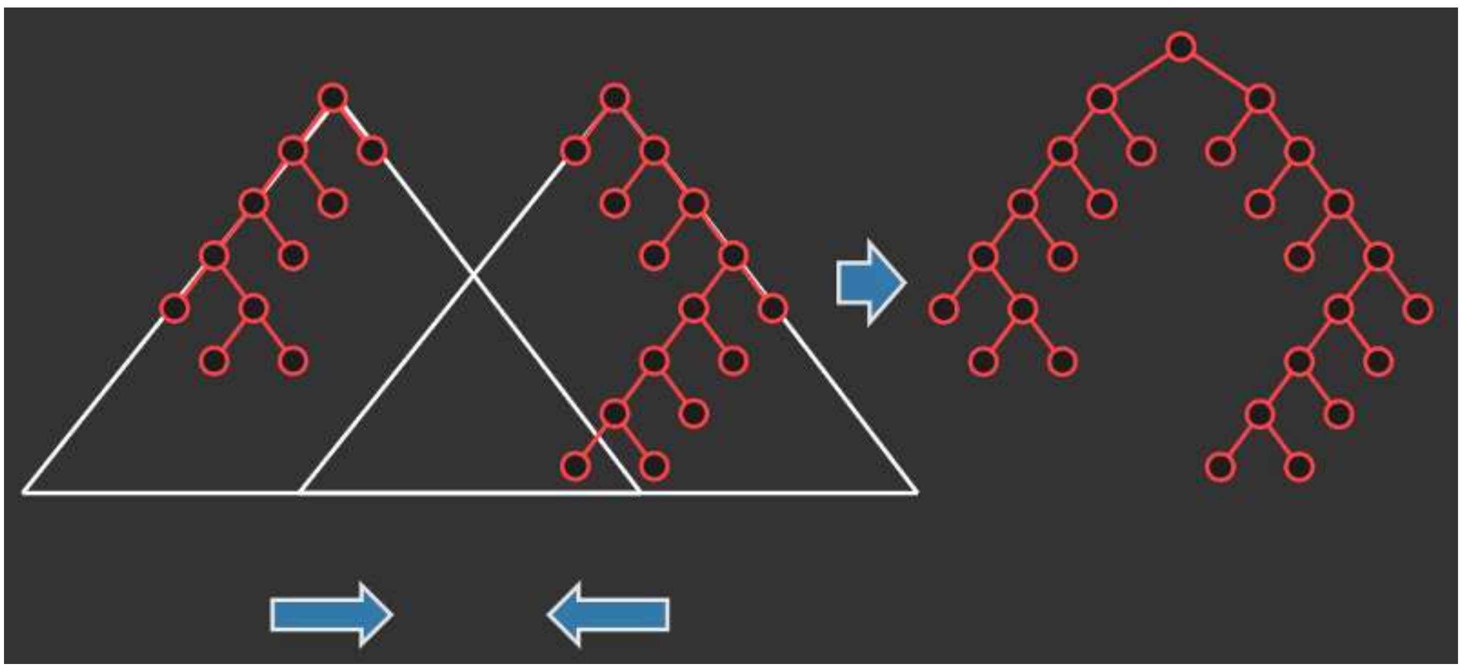


Image from Pat Hanrahan

Reingold-Tilford: Threading

- Finding the correct distance requires traversal of the contours of each subtree
- If a contour node at depth k is a leaf, store *thread* to contour node at depth $k+1$

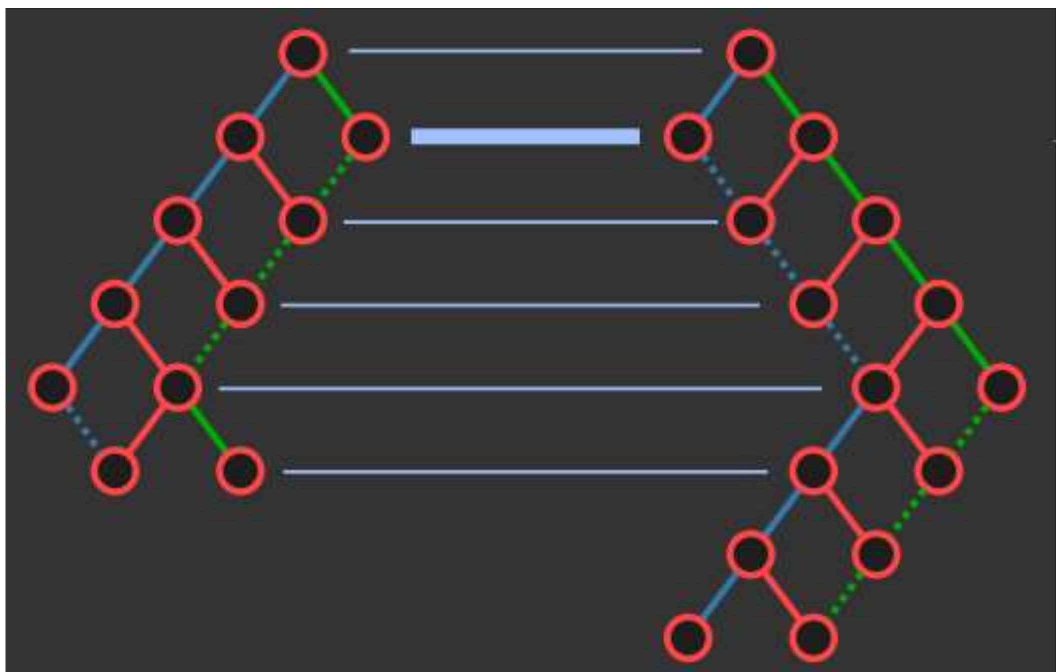
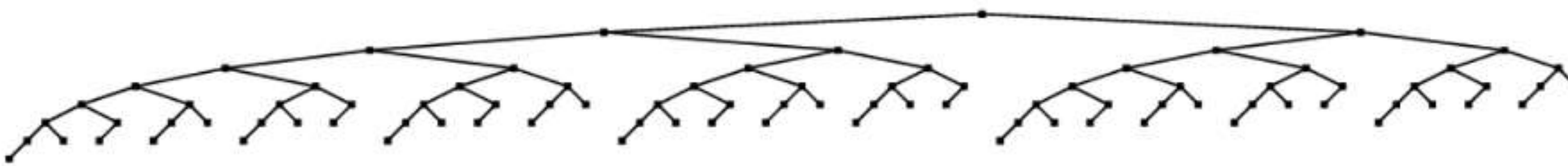


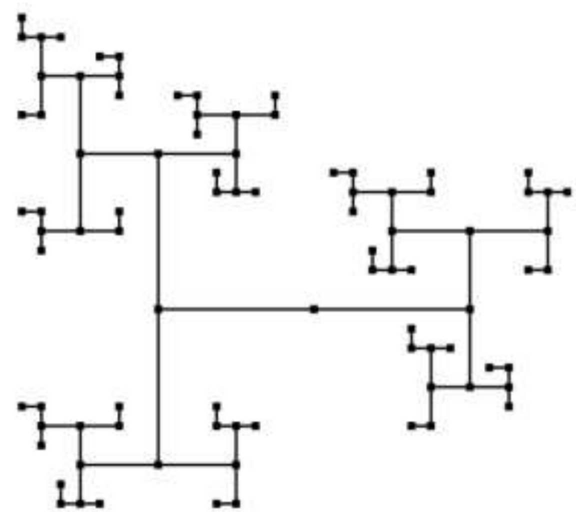
Image from Pat Hanrahan

Reingold-Tilford: Pros and Cons

- The Reingold-Tilford algorithm is
  - easy to understand and implement, but
  - can lead to poor aspect ratios:



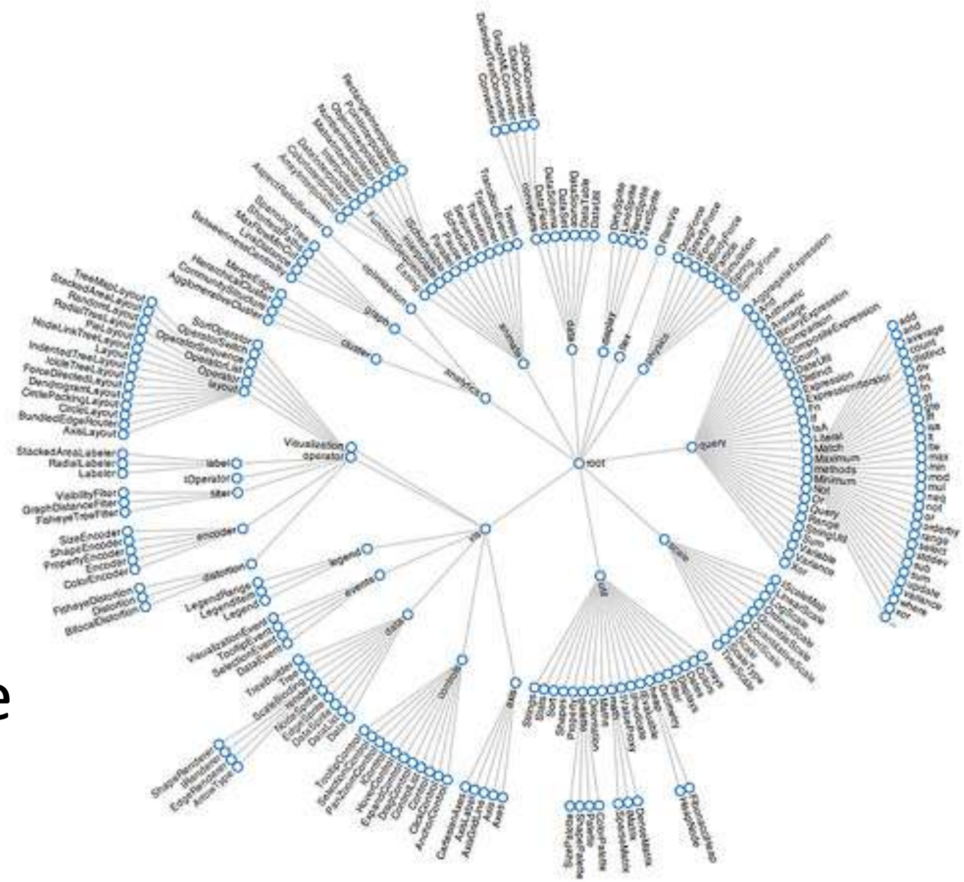
  - alternative, non-level based layout of the same tree:



Images from Adrian Rusu

Node-Link Diagrams: Radial Layout

- node-link diagram in polar coordinates
- radius encodes depth with root in center
- angular sectors assigned to subtrees
- Reingold-Tilford can be applied



http://seeingcomplexity.wordpress.com/2011/02/05/hierarchical-edge-bundles/

Node-Link Diagrams: Bubble Tree Layout

- Variant: Each inner node becomes the center of all its children, makes it easy to distinguish subtrees

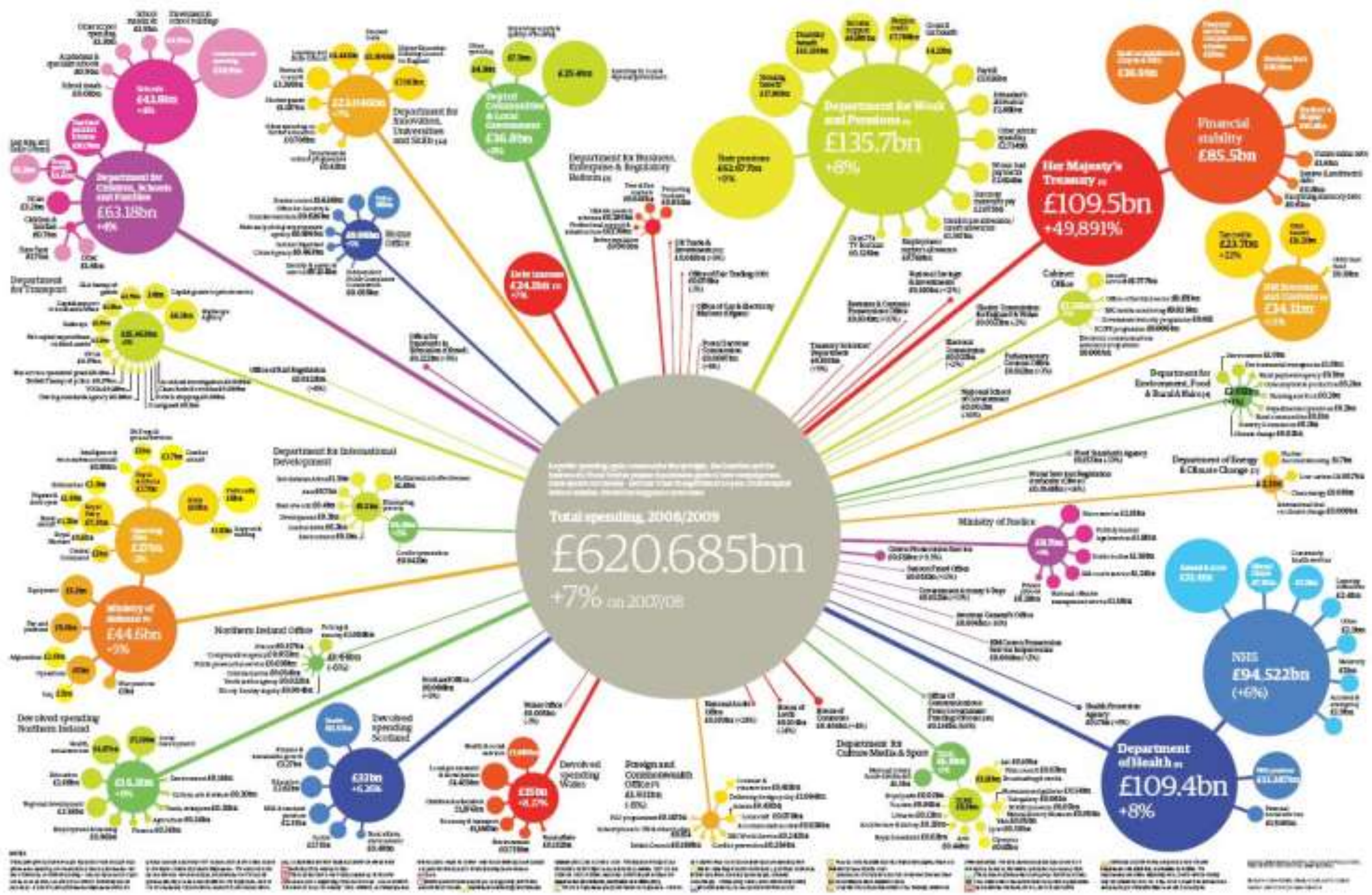
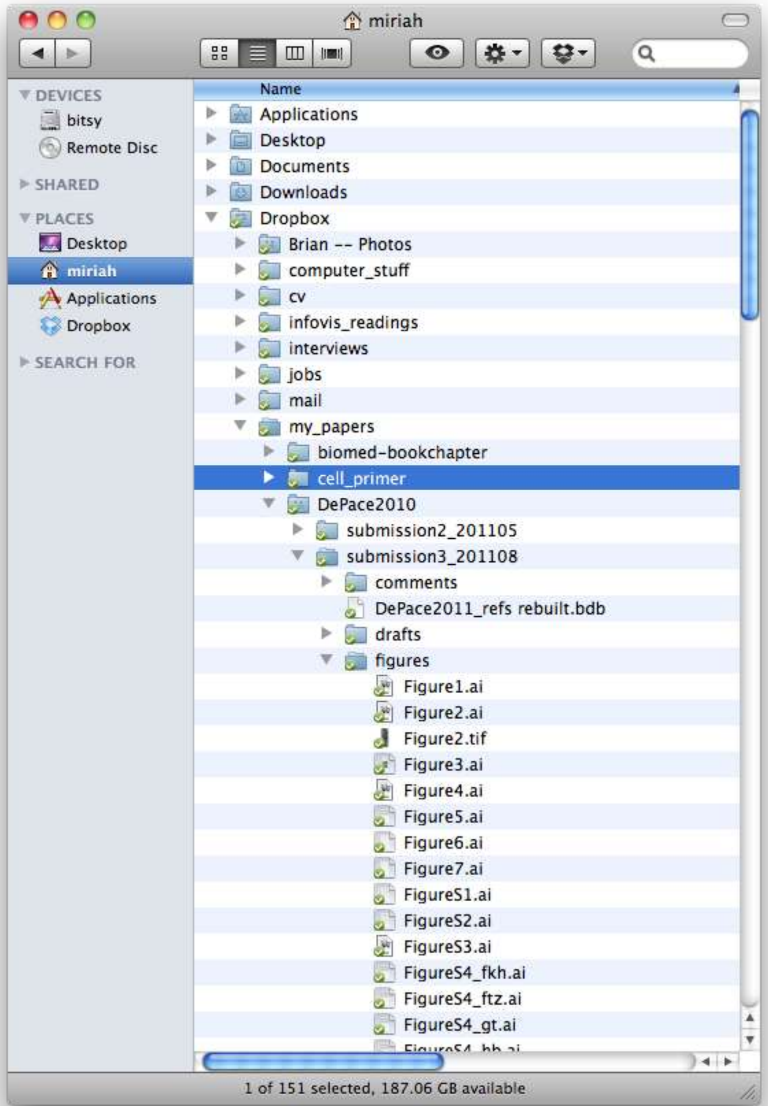


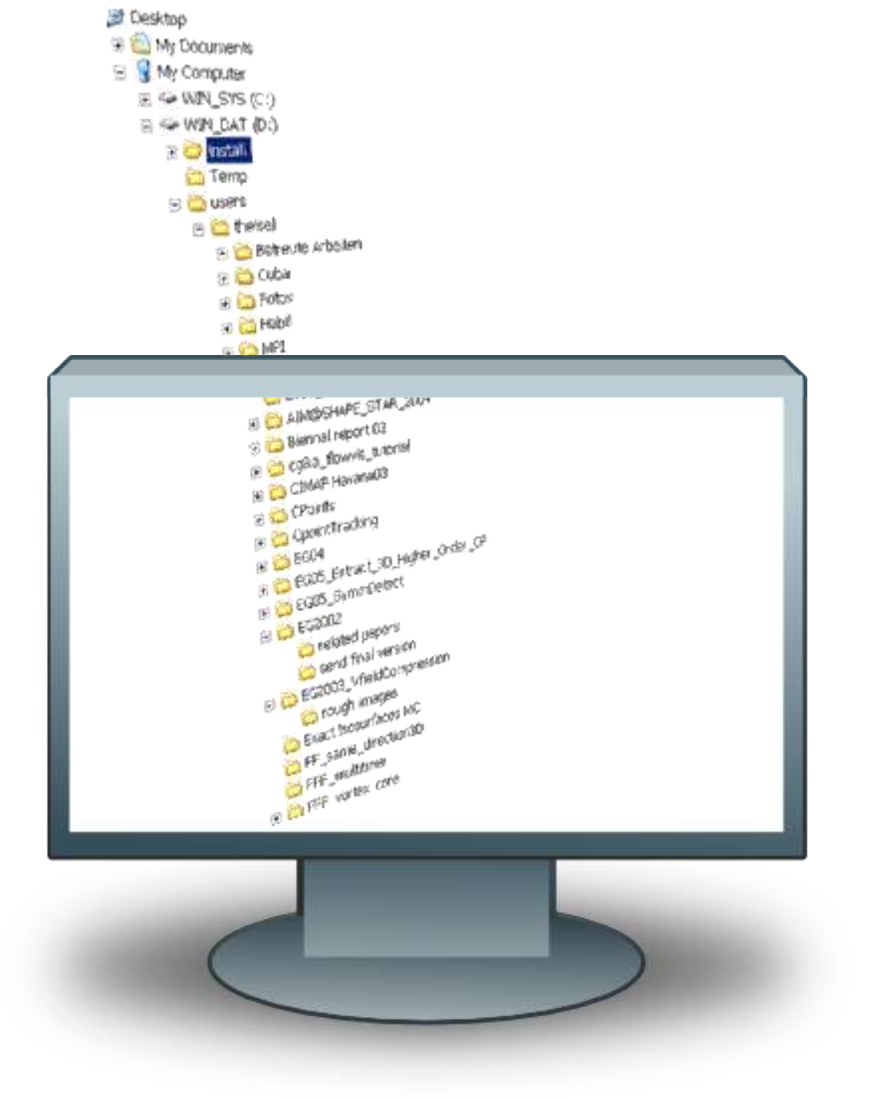
Image: improving-visualization.org

Visualizing Trees: Indentation

- place all items along vertically spaced rows
- indentation used to show parent/child relationships
- commonly used as a component in user interfaces
- breadth and depth contend for space
- often requires a great deal of scrolling



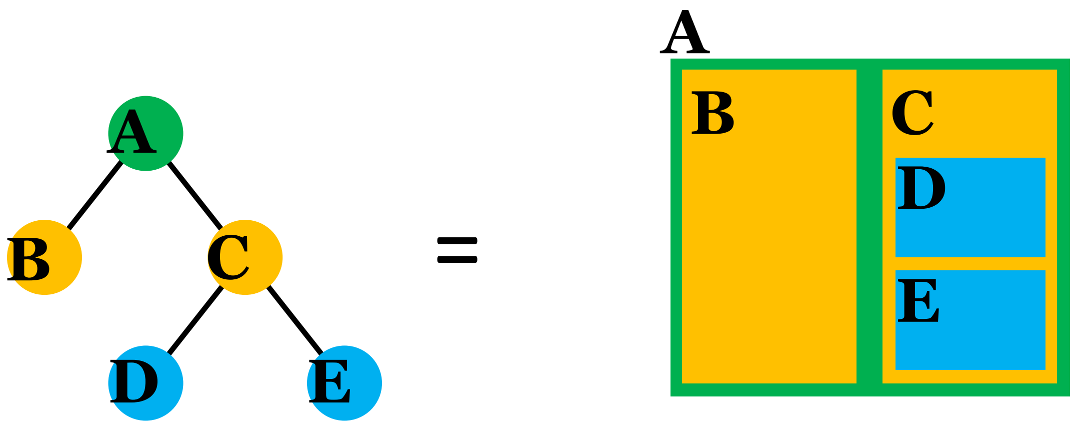
Visualizing Trees: Indentation



- Trees are usually large and unbalanced
- How to use the rectangular screen space optimally?

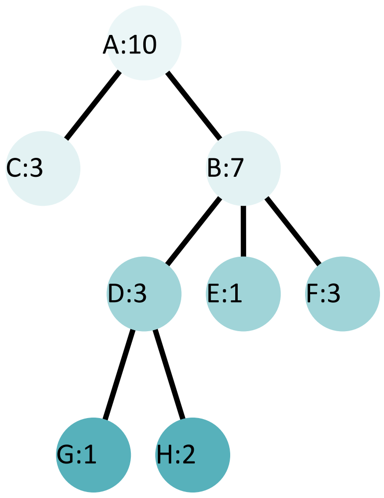
Visualizing Trees: Treemaps

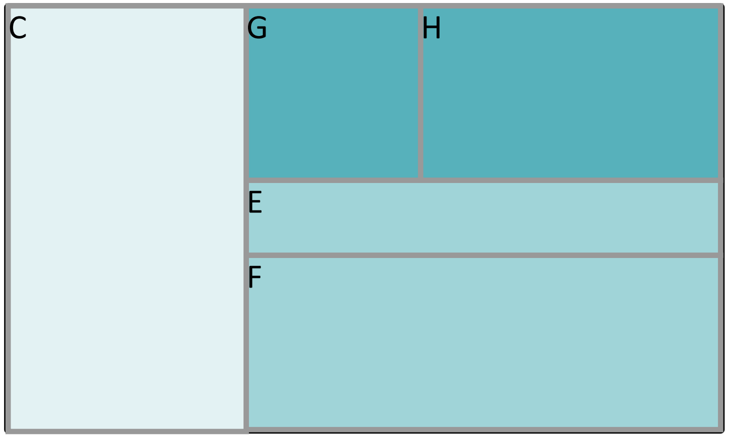
- encode structure using spatial enclosure
  - often referred to as **treemaps**
- benefits
  - provides single view of entire tree
  - easy to tell “size” of node
  - easy to encode additional attributes (color)
- problems
  - difficult to accurately read depth



Visualizing Trees: Treemaps

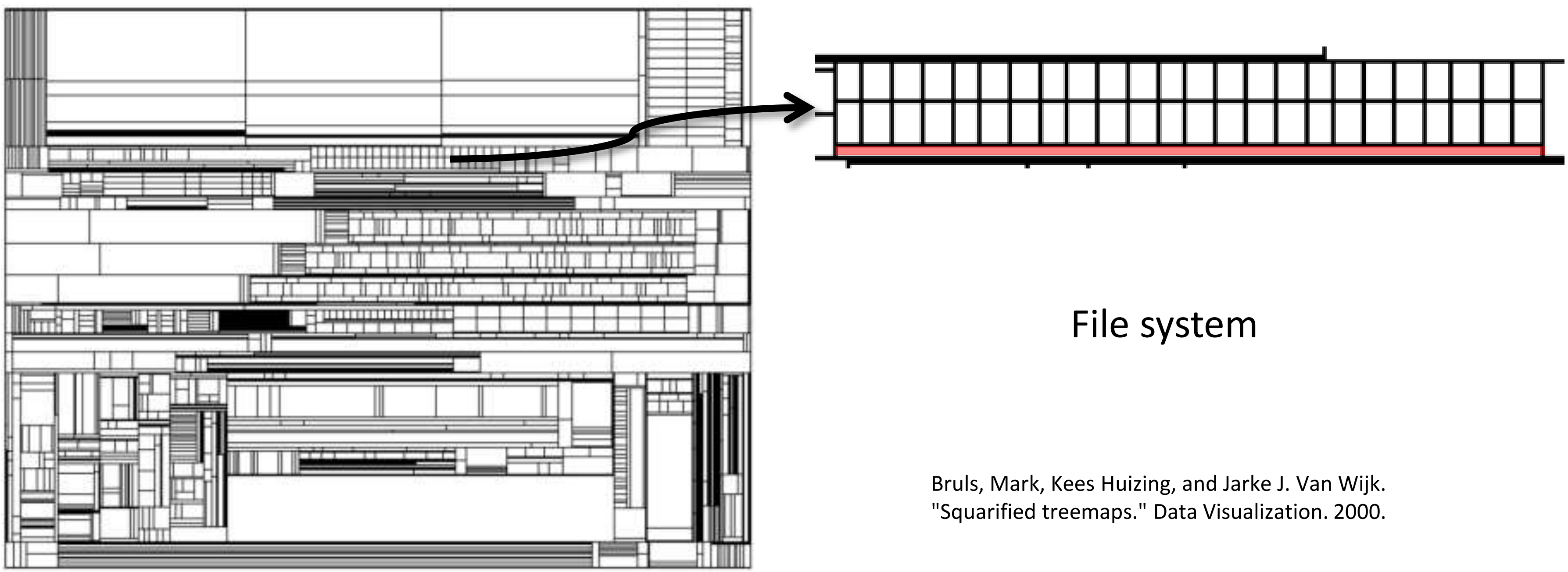
- recursively fill space based on a size metric for nodes
- enclosure indicates hierarchy
- additional measures can control aspect ratio of cells





Visualizing Trees: Treemaps

- **Problem:** Naïve splitting can lead to rectangles with poor aspect ratios

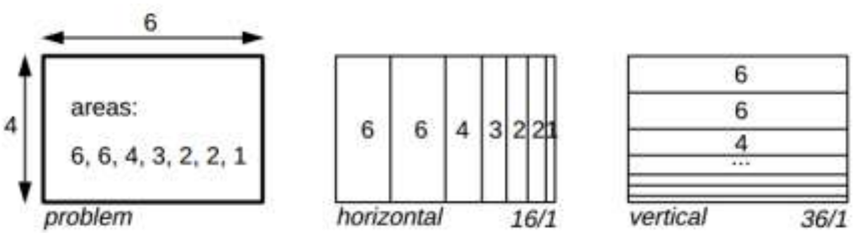


File system

Bruls, Mark, Kees Huizing, and Jarke J. Van Wijk. "Squarified treemaps." Data Visualization. 2000.

Visualizing Trees: Squarified Treemaps

- **Squarified treemaps:** greedy heuristic to favor squares over elongated rectangles
  - finding an optimal solution is NP-hard



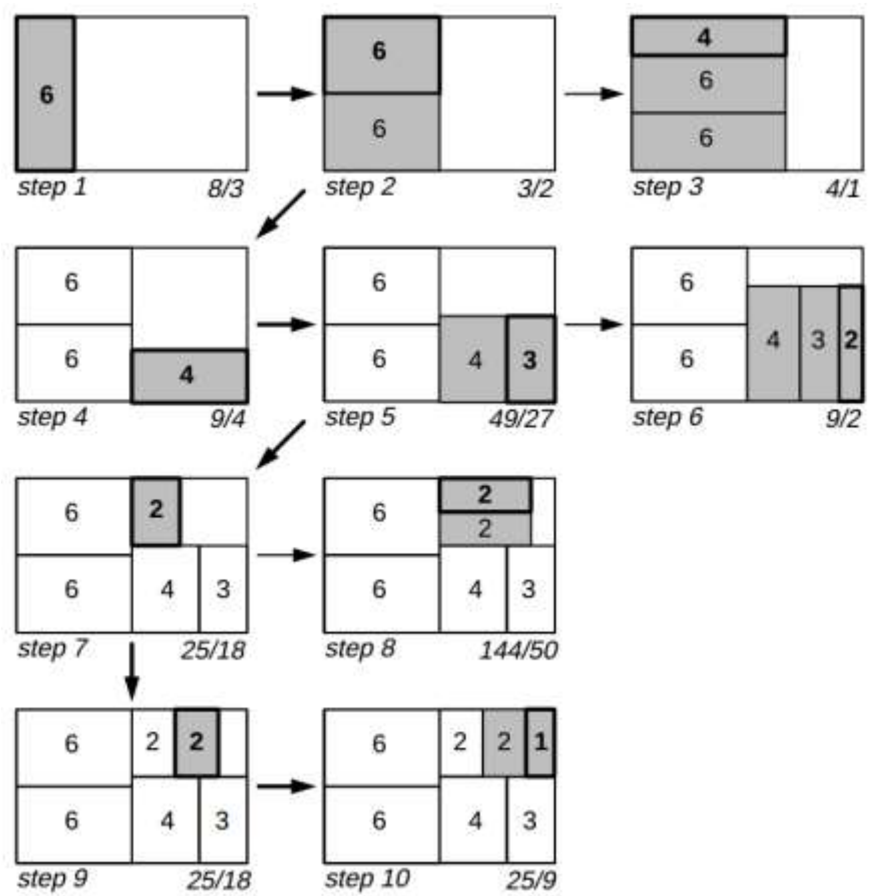
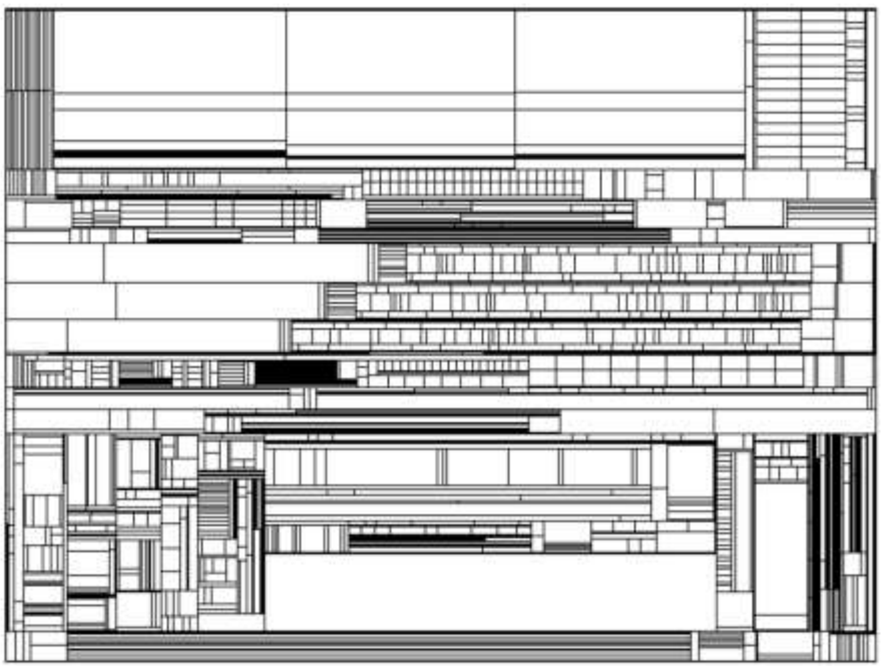


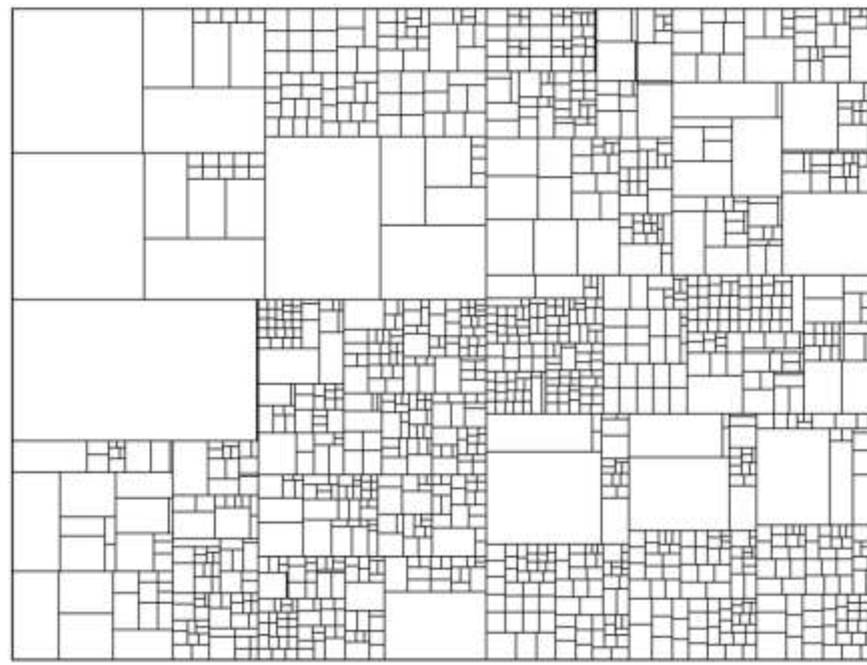
Image from [Bruls et al. 2000]

Example Result: Squarified Treemap

- File system example from [Bruls et al. 2000]
  - **Drawback:** Hierarchical structure more difficult to perceive

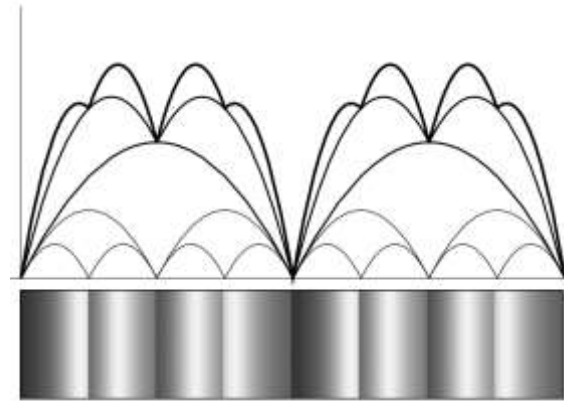


Traditional Treemap

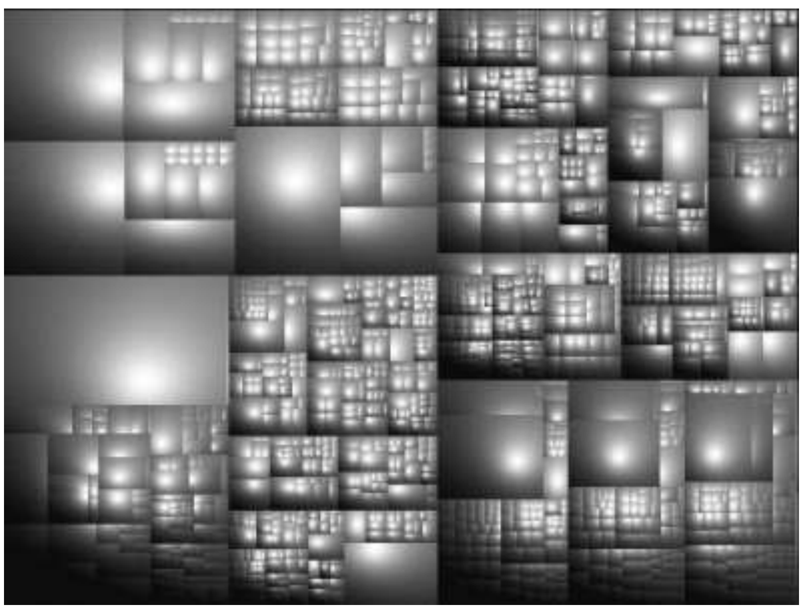


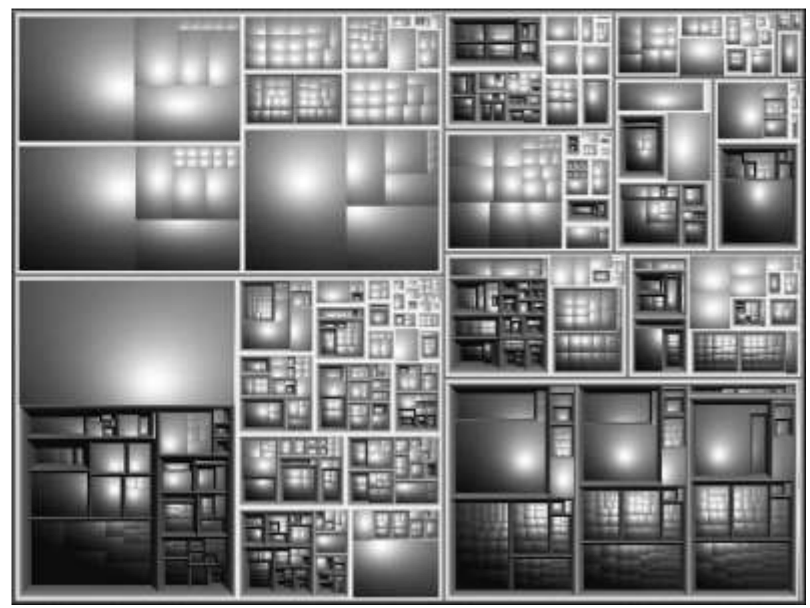
Squarified Treemap

Visualizing Trees: Cushion Shading



- **Cushion shading** supports perception of hierarchies
- $I = I_a + I_d \max(0, \mathbf{n} \cdot \mathbf{l})$
- Can add **frames and interaction**





Images from [van Wijk et al. 1999], [Bruls et al. 2000]

Sunburst Displays

- **Sunburst displays**
  - Root at center, nested rings around it
  - Space-filling like treemaps
  - Layout similar to radial node-link diagrams

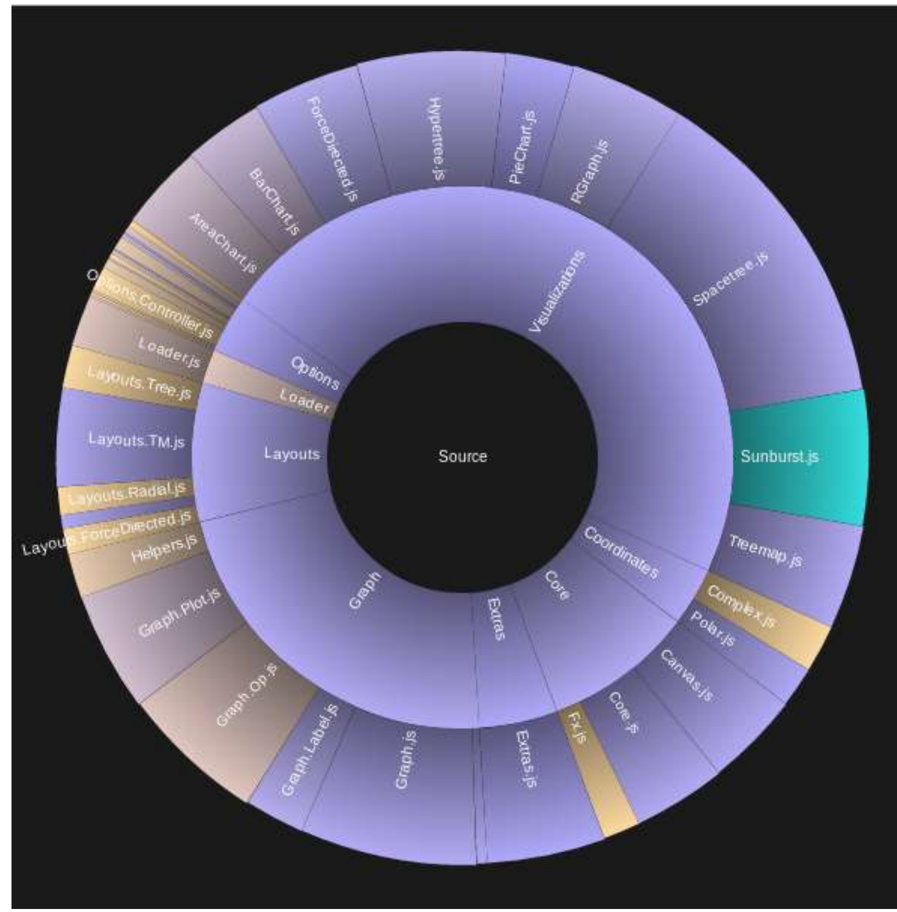


Image: http://philogb.github.io/jit/static/v20/Jit/Examples/Sunburst/example2.html

Summary: Visualizing Trees / Hierarchies

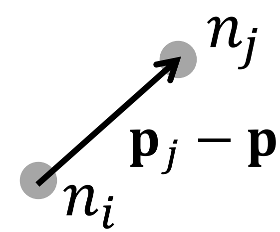
- Main strategies for visualizing trees:
  - Node-link diagrams
    - Reingold-Tilford
    - Radial layouts
  - Indentation
  - Containment (treemaps)
    - Squarification
    - Cushions and frames

Section 5.3:
General Graphs

Force-Directed Layouts

- **Goal:** Place groups of strongly connected nodes close to each other, preserve minimum distance between nodes
  - *Spring force* that node n_j exerts on n_i (natural spring length s_{ij}, tension k_{ij}) if an edge connects them

$$\mathbf{f}_{ij}(x) = k_{ij}\left(\|\mathbf{p}_j - \mathbf{p}_i\| - s_{ij}\right)\frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_j - \mathbf{p}_i\|}$$



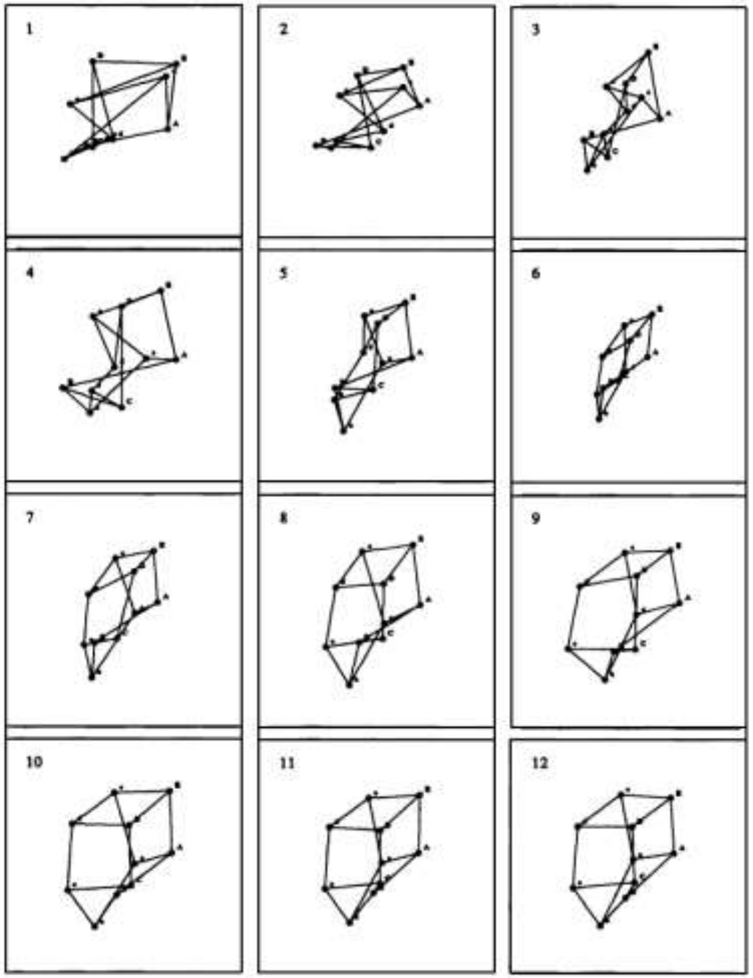
  - *Electrical repulsion* that node n_j exerts on n_i (repulsion strength r_{ij})

$$\mathbf{g}_{ij}(x) = -\frac{r_{ij}}{\|\mathbf{p}_j - \mathbf{p}_i\|^2}\frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_j - \mathbf{p}_i\|}$$

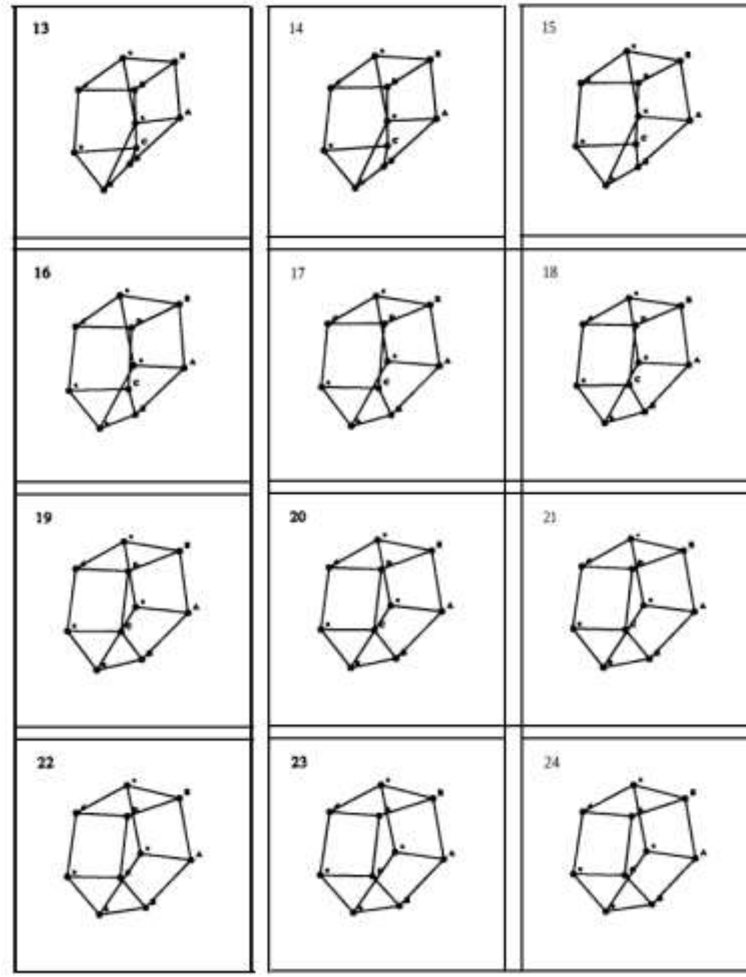
Force-Directed Layouts: General Algorithm

- Initialize (randomly or through a heuristic)
- Iterate:
  - For each node:
    - Sum all attractive and repulsive forces
    - Multiply overall force by stepsize (“temperature”)
    - Impose a maximum displacement (e.g., keep within image)
    - Move node
  - Adjust temperature

Quenching and Simmering



Quenching: Rapid cooling

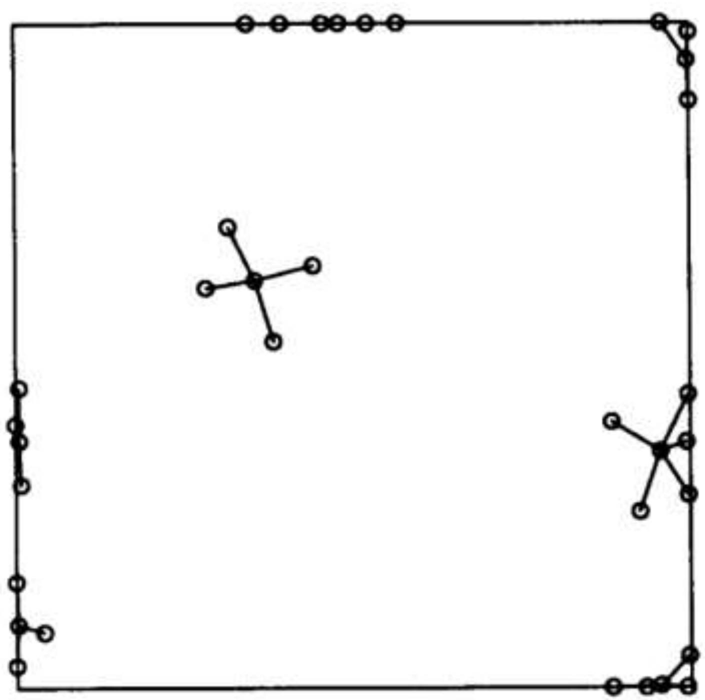


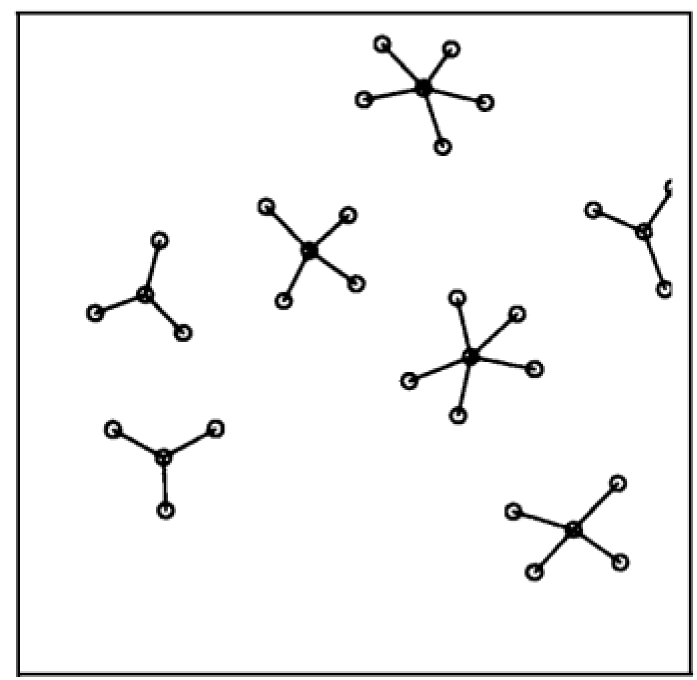
Simmering: Constant low temperature

Images from [Fruchtermann/Reingold 1991]

Grid Variant

- **Speedup:** Divide plane into grid cells and only compute repulsive forces between nodes in the same cell
  – Produces better results for disconnected graphs





Images from [Fruchtermann/Reingold 1991]

Layered Layouts

- Nodes of **Directed Acyclic Graphs** can be organized in layers
  - Assign vertical position to each layer
  - Suggests a hierarchy, similar to tree visualization

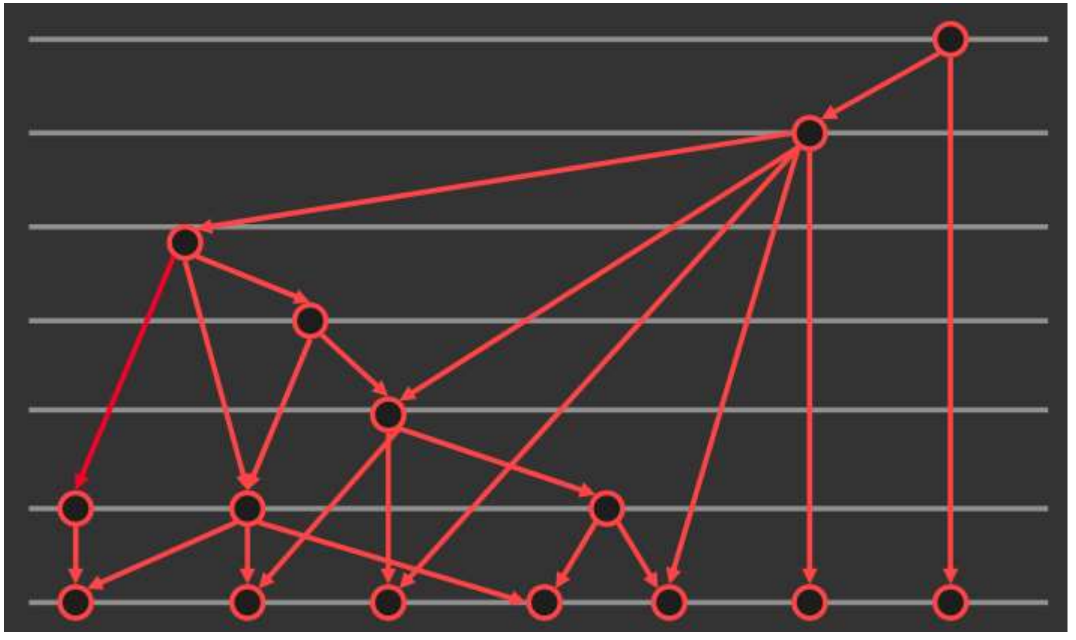
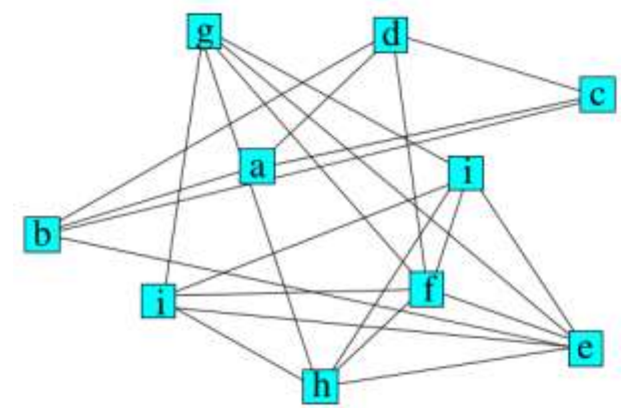
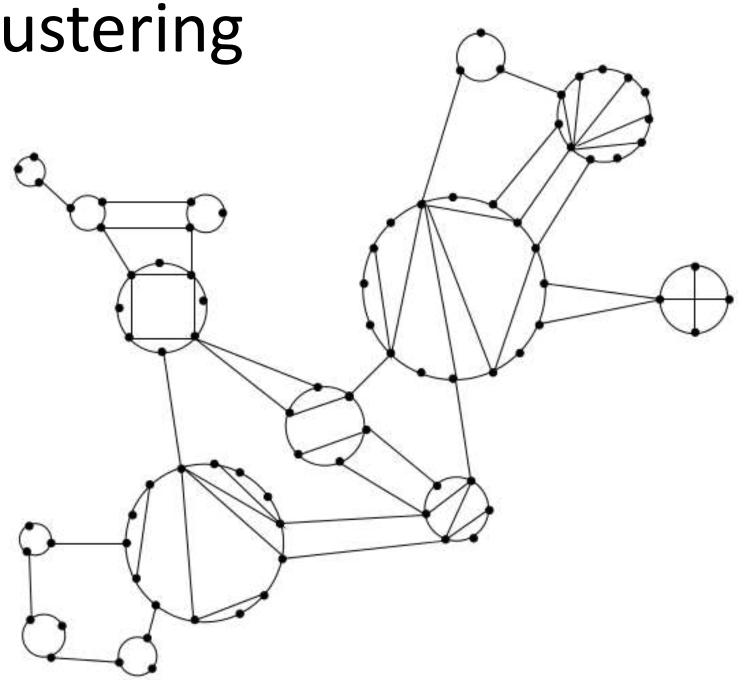


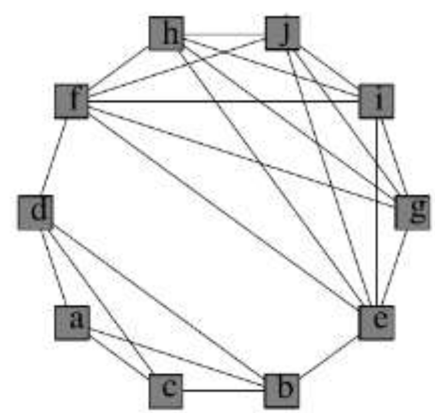
Image from Pat Hanrahan

Circular Layouts

- **Circular layouts** distribute nodes along the circumference of a circle
  - Try to minimize edge crossings
  - Can be combined with clustering



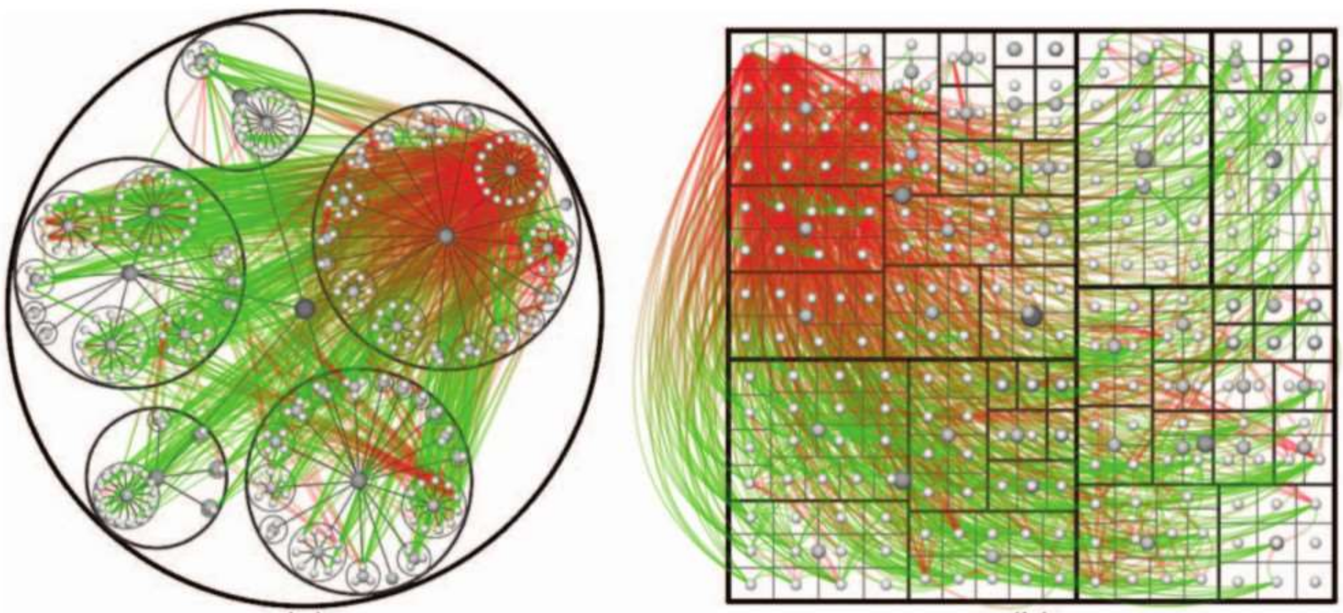




Images from [Six/Tollis 2013]

Hierarchical Edge Bundling

- What if you have hierarchical *(inclusion)* and non-hierarchical *(adjacency)* relationships at the same time?
  - *Example:* software system; caller green, callee red

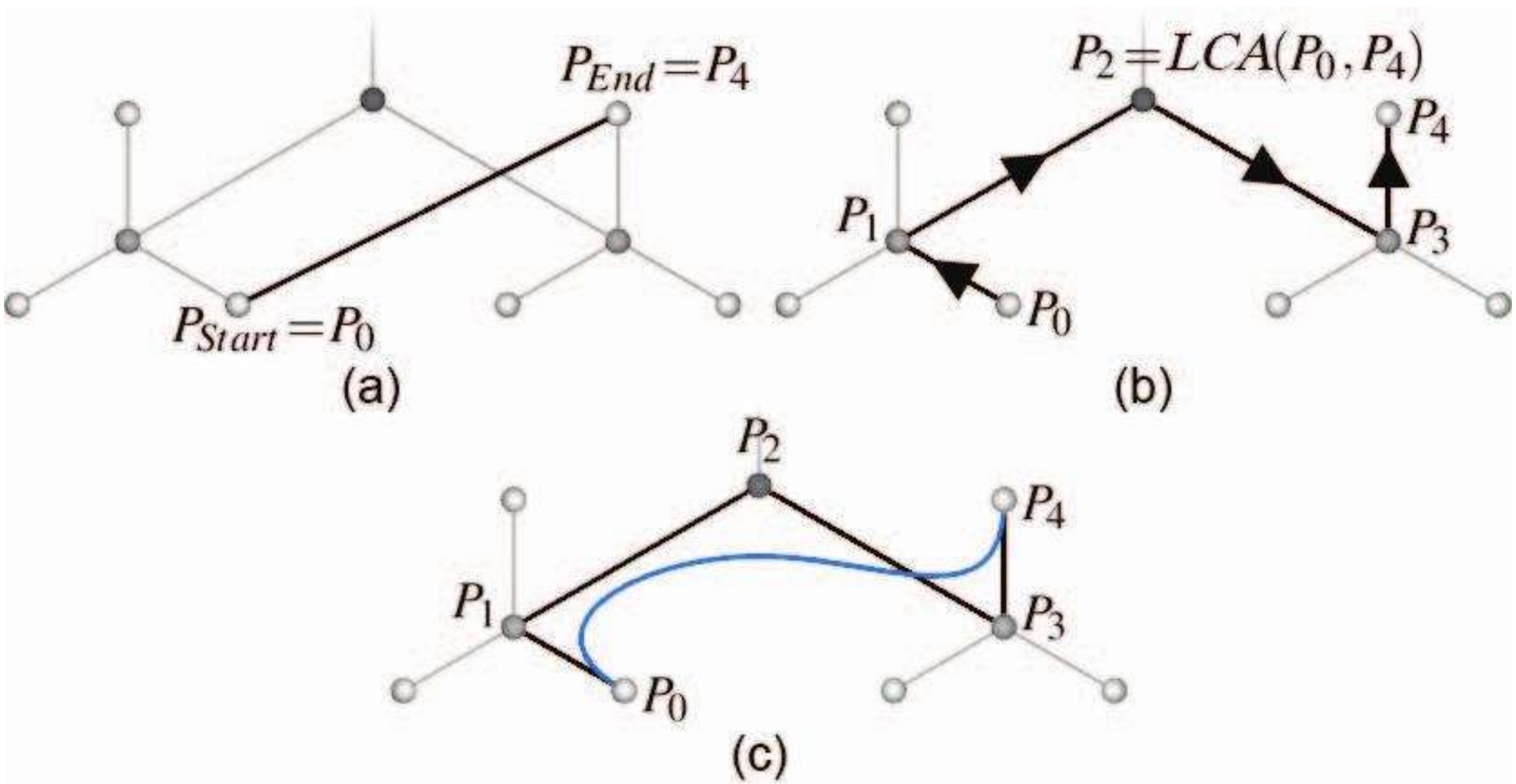


(a) (b)

Images by [Holten 2006]

Hierarchical Edge Bundling: Concept

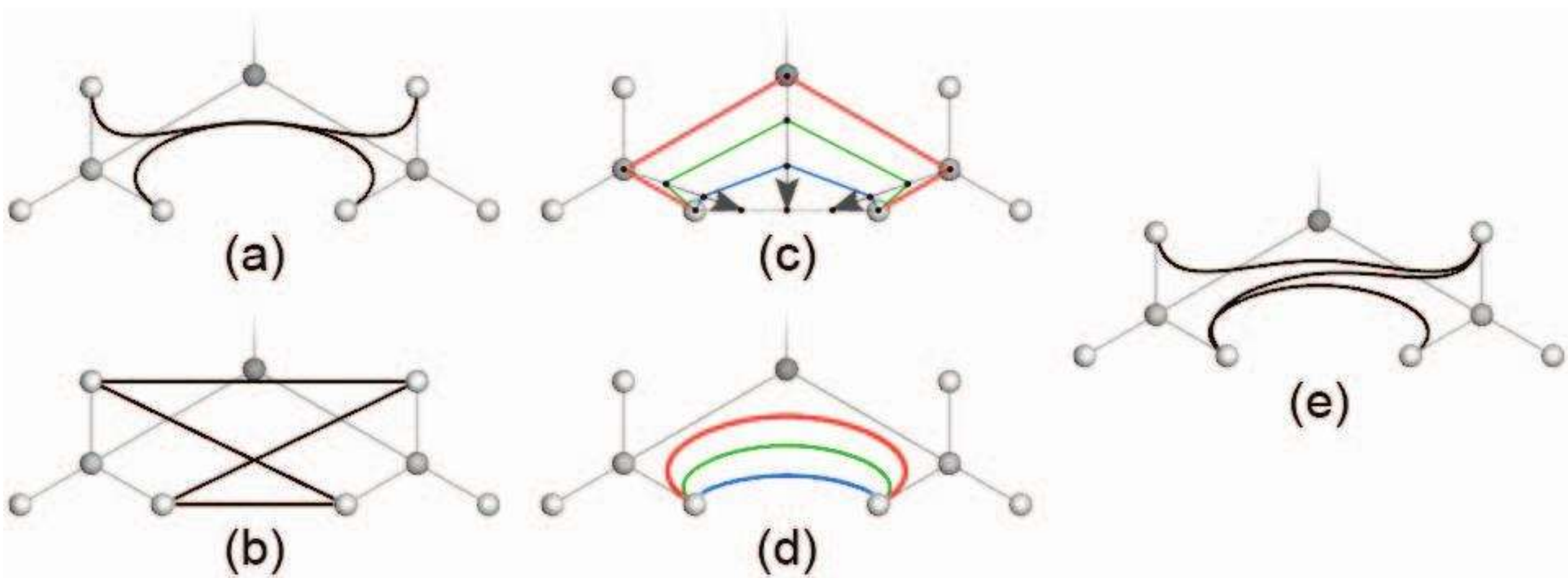
- **Idea:**
  - Use standard tree layout for hierarchy
  - Path along tree as control polygon for graph edge



Images by [Holten 2006]

Edge Bundling Strength

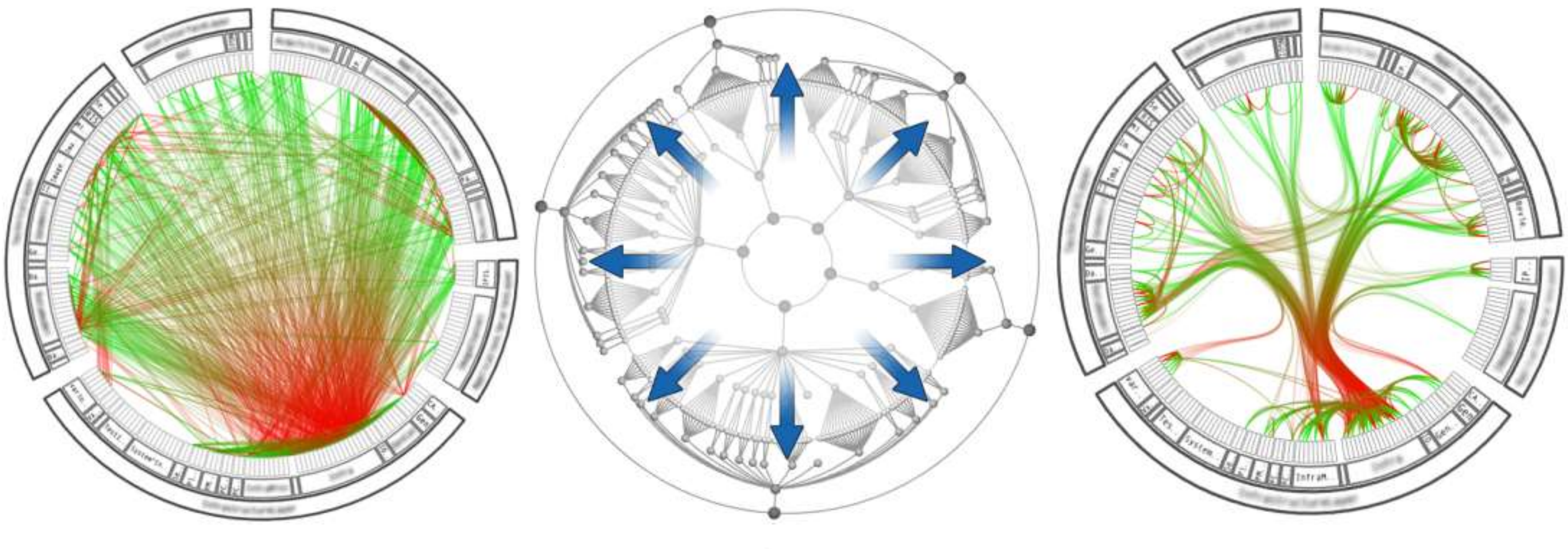
- Bundling can lead to ambiguities
- Continuously straightening edges using a **bundling strength** $\beta \in [0,1]$ resolves this



Images by [Holten 2006]

Edge Bundling: Radial Layout

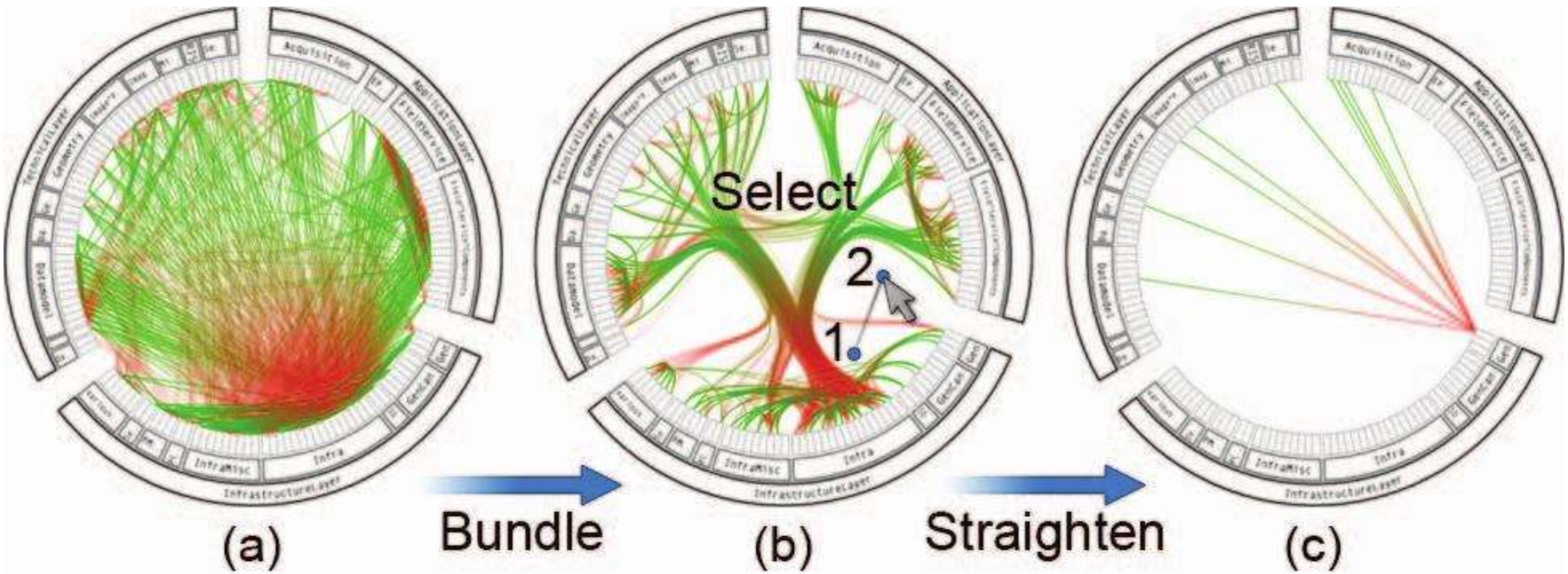
- **Radial Graph Layout** can be used for control polygons; nodes mirrored to the outside for labeling



Images by [Holten 2006]

Edge Bundling: Interaction

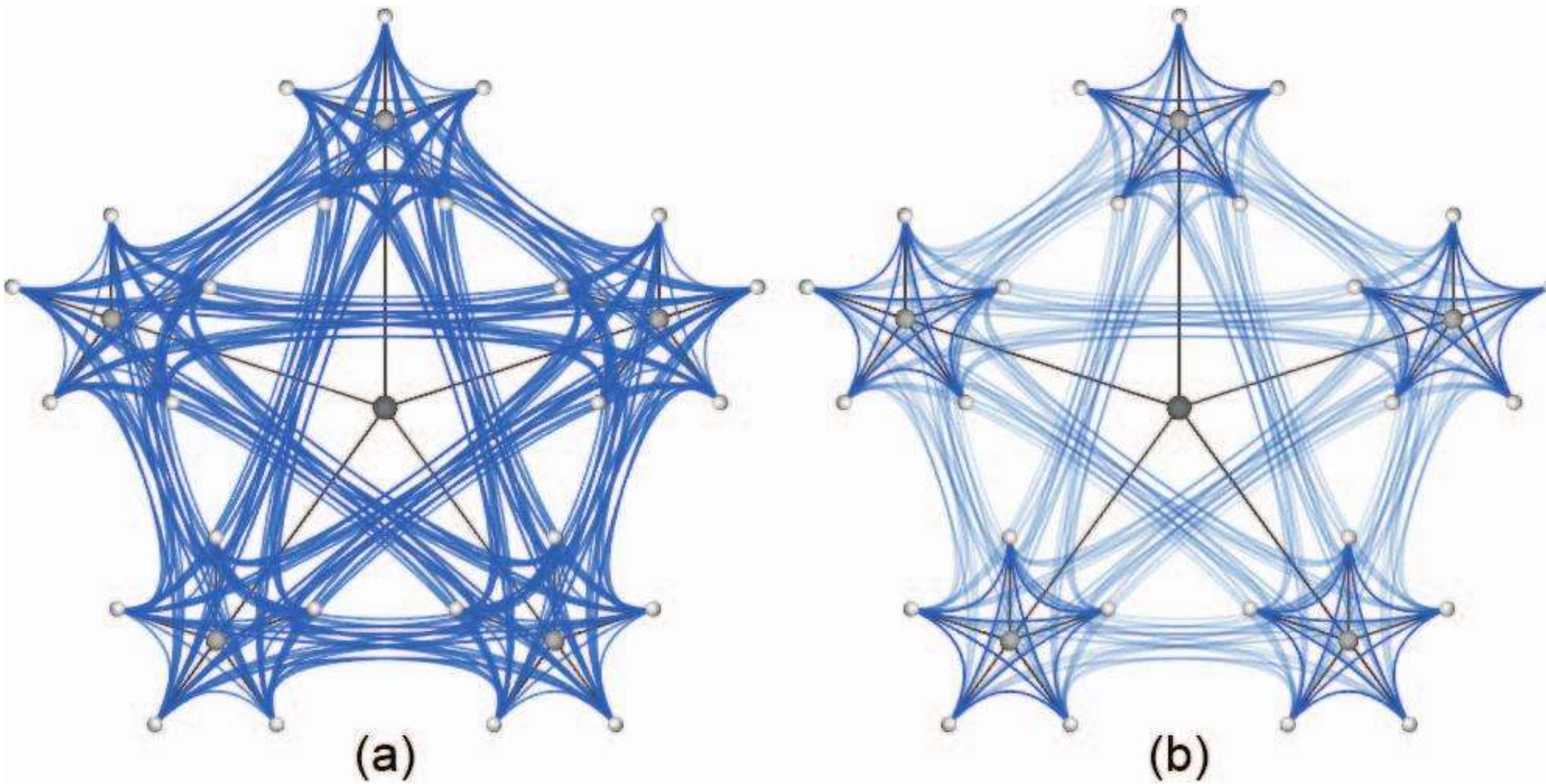
- **Interactive filtering** and straightening helps with disambiguation:



Images by [Holten 2006]

Edge Bundling: Transparency

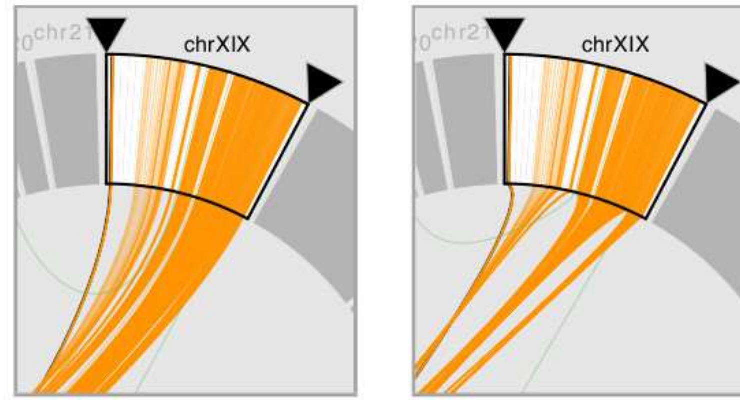
- Making longer bundles more **transparent** allows us to more easily see shorter edges



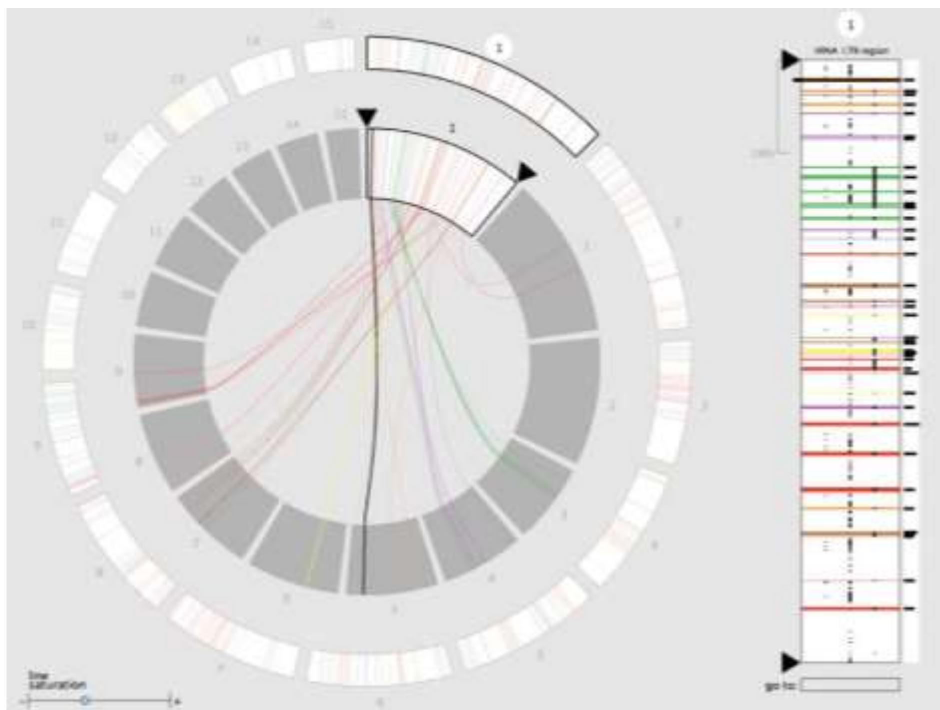
Images by [Holten 2006]

Edge Bundling in Comparative Genomics

- MizBee [Meyer et al. 2009] bundles edges from the same source block that are preserved in the same target chromosome

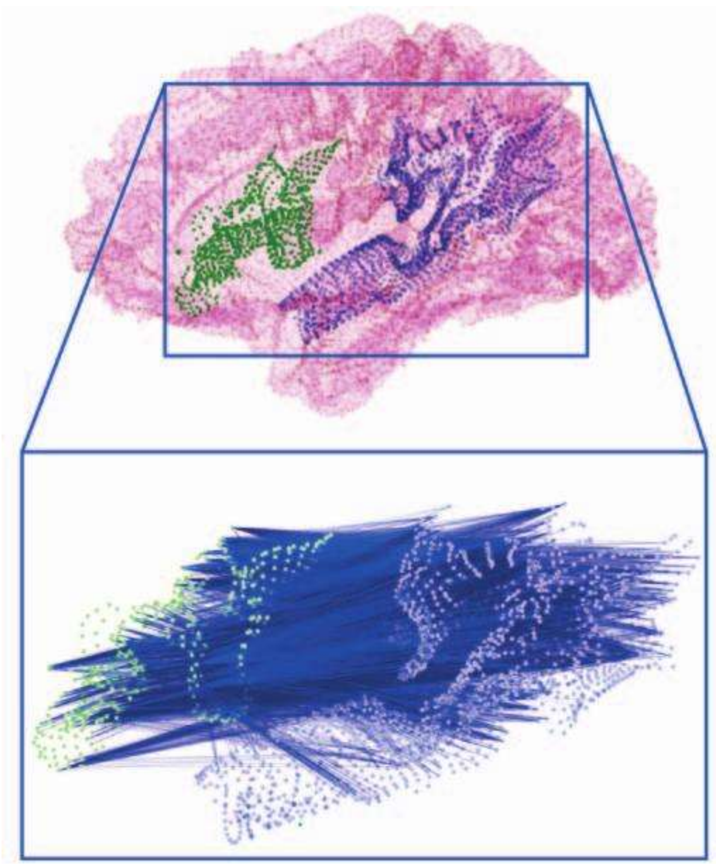


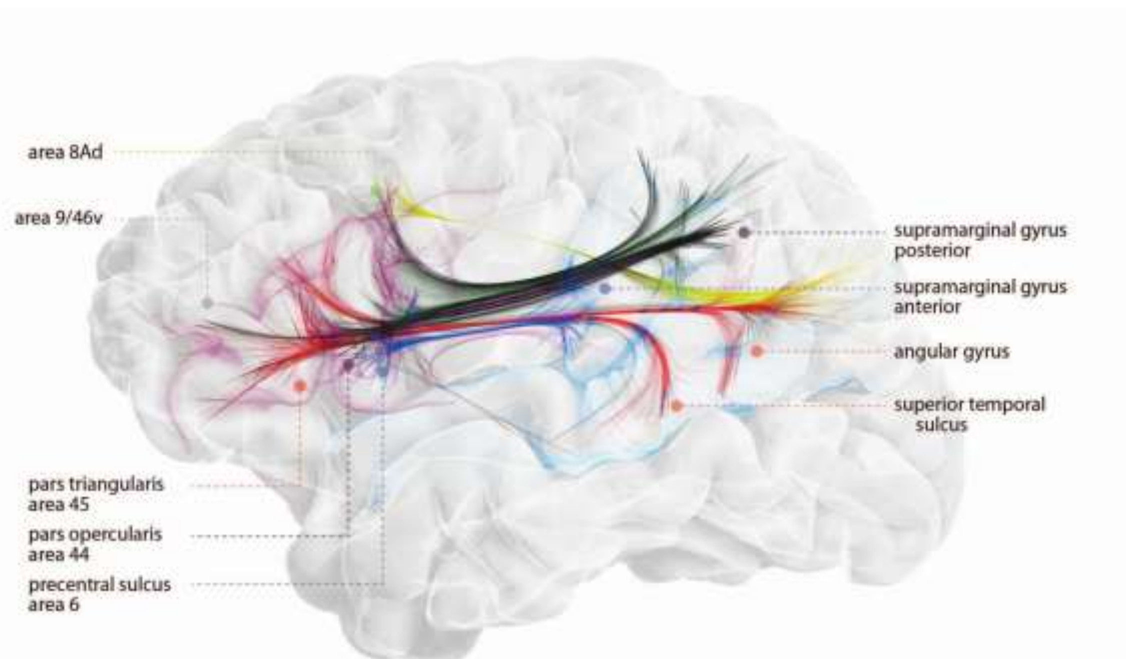
(a) (b)



Edge Bundling in Neuroimaging

- [Böttger et al. 2014] iteratively move similar edges in 3D fMRI connectivity graphs towards each other





Summary: Visualizing Graphs

- **Node-and-link** visualizations of general graphs
  – Force-directed layouts
- Techniques for **specific classes of graphs**
  – Layered layouts for directed acyclic graphs
  – Circular layouts
- **Edge bundling** can clean up visualizations of graphs with many edges
  – First introduced for “inclusion+adjacency” graphs
  – Idea has been applied widely

References

- Matthew O. Ward, Georges Grinstein, Daniel Keim. **Interactive Data Visualization: Foundations, Techniques, and Applications.** A. K. Peters/CRC Press, 2010
- Roberto Tamassia (Ed). **Handbook of Graph Drawing and Visualization.** CRC Press, 2013