NETWORK VISUALIZATION

CS 448B | Fall 2024

MANEESH AGRAWALA

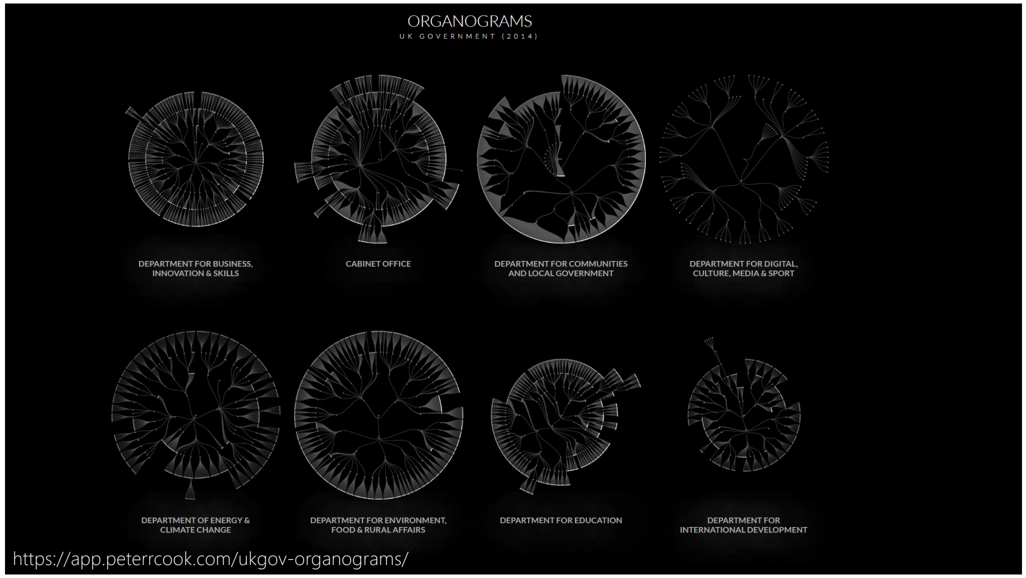
1

READING RESPONSE: QUESTIONS/THOUGHTS

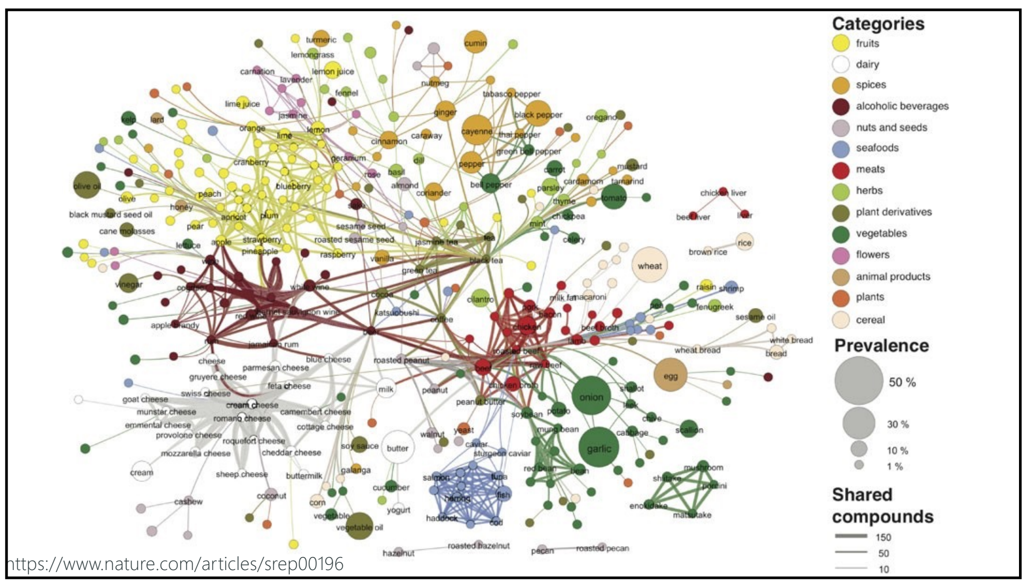
While Segel and Heer's martini glass structure actively encourages users to construct personal narratives from the data, Tufte's focus remains on the clear communication of data, with personal interpretation emerging incidentally. This contrast between the two sets of authors raises an interesting question: ***Could actively encouraging personal narratives in data visualization introduce unintended biases, especially when exploring complex or sensitive topics?***

I'm curious how we should think picking color for communicating across cultures. The article by Heer and Stone mentions that categorical color perception is affected by language. Additionally, we saw in lecture results from the world color survey that show south pacific respondents distinguished blue and green far more than respondents from Mexico. Clearly, to what degree we view colors as distinct is subjective and contingent upon our culture and our language. I wonder, how much might these linguistic and cultural differences affect how viewers perceive our visualizations? ***How should we choose colors for our visualizations if we're trying to reach a multicultural audience?***

2


ORGANOGRAMS
UK GOVERNMENT (2014)
DEPARTMENT FOR BUSINESS, INNOVATION & SKILLS
CABINET OFFICE
DEPARTMENT FOR COMMUNITIES AND LOCAL GOVERNMENT
DEPARTMENT FOR DIGITAL, CULTURE, MEDIA & SPORT
DEPARTMENT OF ENERGY & CLIMATE CHANGE
DEPARTMENT FOR ENVIRONMENT, FOOD & RURAL AFFAIRS
DEPARTMENT FOR EDUCATION
DEPARTMENT FOR INTERNATIONAL DEVELOPMENT
https://app.peterrcook.com/ukgov-organograms/

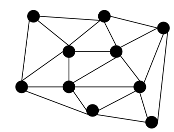
3


Categories
fruits
dairy
spices
alcoholic beverages
nuts and seeds
seafoods
meats
herbs
plant derivatives
vegetables
flowers
animal products
plants
cereal
Prevalence
50 %
30 %
10 %
1 %
Shared compounds
150
50
10
https://www.nature.com/articles/srep00196

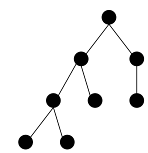
4

GRAPHS AND TREES

Graphs
Model relations among data
Nodes and *edges*



Trees
Graphs with hierarchical structure
Connected graph with N-1 edges
Nodes as *parents* and *children*



5

SPATIAL LAYOUT

Primary concern – positioning of nodes and edges

Often (but not always) goal is to depict structure

- Connectivity, path-following
- Topological distance
- Clustering/grouping
- Ordering (e.g., hierarchy level)

6

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity

Attribute-based: properties associated with node or link

Browsing: follow paths in the data

Estimation: summarization and temporal changes

8

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity

- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link

Browsing: follow paths in the data

Estimation: summarization and temporal changes

9

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity
- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link
- Find all friends of Fozzie that are students at Stanford (node property)
- Find all friends of Fozzie that are their family (link property)

Browsing: follow paths in the data

Estimation: summarization and temporal changes

10

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity
- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link
- Find all friends of Fozzie that are students at Stanford (node property)
- Find all friends of Fozzie that are their family (link property)

Browsing: follow paths in the data
- Find Kermit’s friend with first name Beaker and then find Beaker’s mentor Bunsen

Estimation: summarization and temporal changes

11

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity
- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link
- Find all friends of Fozzie that are students at Stanford (node property)
- Find all friends of Fozzie that are their family (link property)

Browsing: follow paths in the data
- Find Kermit's friend with first name Beaker and then find Beaker's mentor Bunsen

Estimation: summarization and temporal changes
- How does Miss Piggy's friend group change over the course of a year

12

TODAY

Learning Objectives

1. Techniques for visualizing trees
2. Techniques of laying out graphs
3. Alternative techniques for visualizing node-link data

13

TREE VISUALIZATION

14

COMMON TYPES OF TREE VISUALIZATION

Indentation

Linear list, indentation encodes depth

Node-Link diagrams

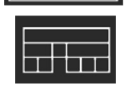
Nodes connected by lines/curves

Enclosure diagrams

Represent hierarchy by enclosure

Layering

Layering and alignment



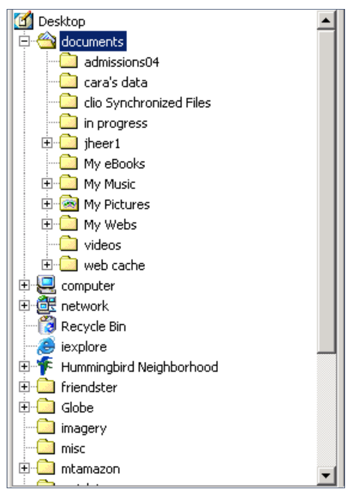
Tree layout is fast: O(n) or O(n log n), enabling real-time layout for interaction

15

INDENTATION

16

INDENTATION



Items along vertically spaced rows

Indentation shows parent/child relationships

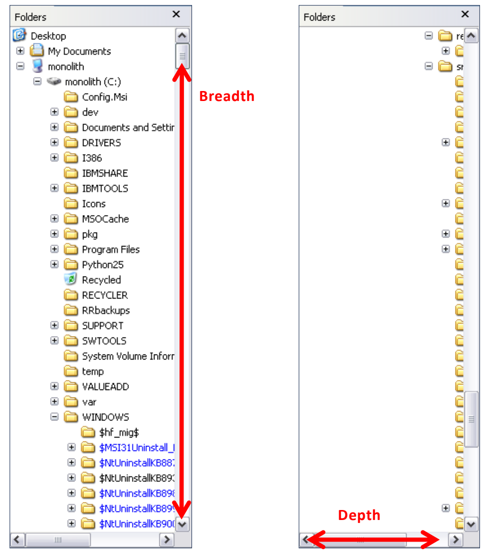
Often used in interfaces

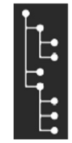
Breadth/depth contend for space

Often requires scrolling

17

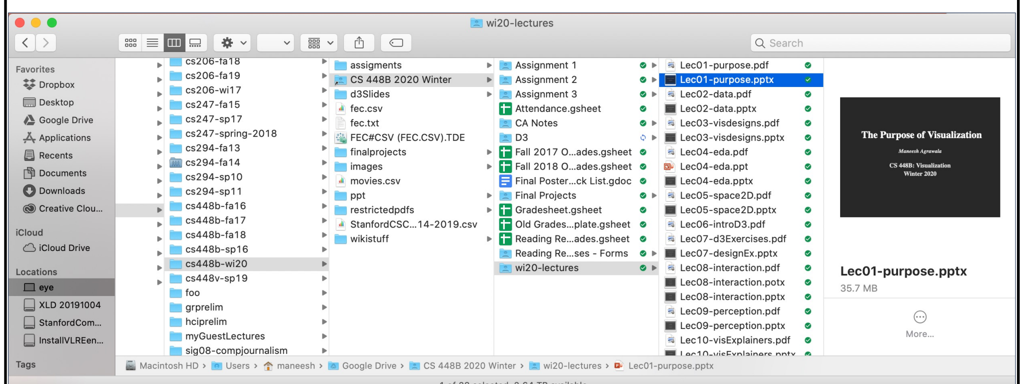
INDENTATION





18

SINGLE-FOCUS ACCORDION LIST

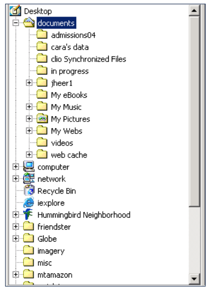


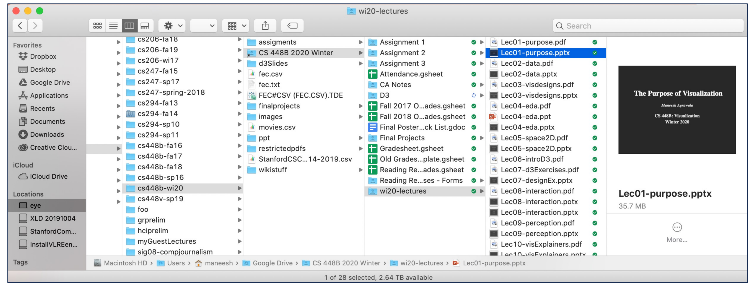
Separate breadth & depth in 2D

Focus on single path at a time

19

WHAT TASKS ARE THESE GOOD FOR?





Benefits

Navigation + Browsing, Parent-child relationships

Disadvantages

Network overview, Estimation, Comparison

20

NODE-LINK DIAGRAMS

21

NODE-LINK DIAGRAMS

Nodes distributed in space, connected by straight or curved lines

Use 2D space to break apart breadth and depth

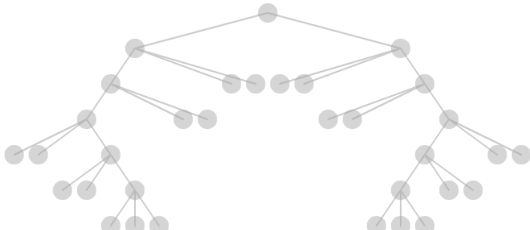
Space used to communicate hierarchical orientation

(e.g., towards *authority* or *generality*)

22

BASIC RECURSIVE LAYOUT

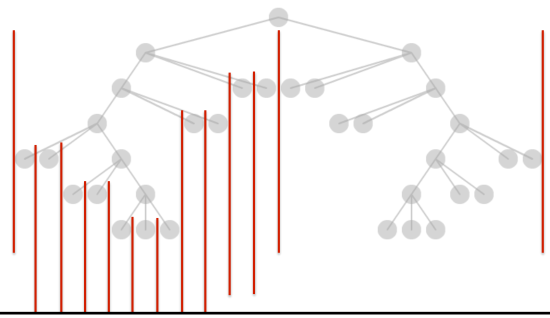
Repeatedly divide space for subtrees by leaf count



23

BASIC RECURSIVE LAYOUT

Repeatedly divide space for subtrees by leaf count



24

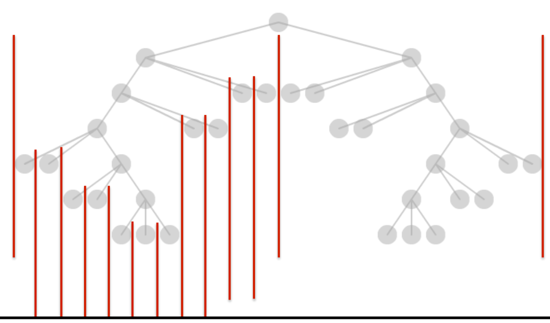
BASIC RECURSIVE LAYOUT

Repeatedly divide space for subtrees by leaf count

Breadth of tree along one dimension

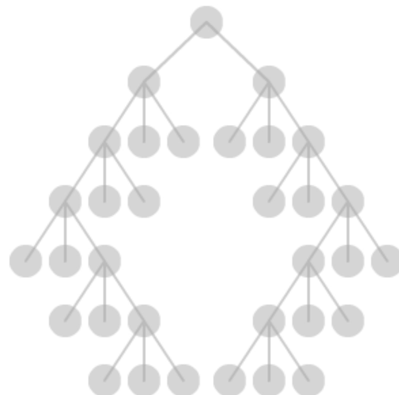
Depth along the other dimension

Problem: Exponential growth of breadth



25

REINGOLD & TILFORD'S "TIDY" LAYOUT



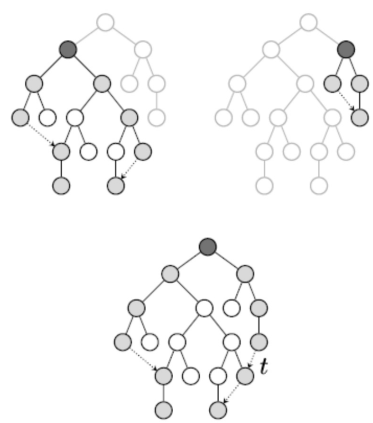
Goal: maximize density and symmetry

Originally for binary trees, extended by Walker to cover general case

Corrected by Buchheim et al. to achieve a linear time algorithm

26

REINGOLD & TILFORD LAYOUT



Design Considerations

Clearly encode depth

No edge crossings

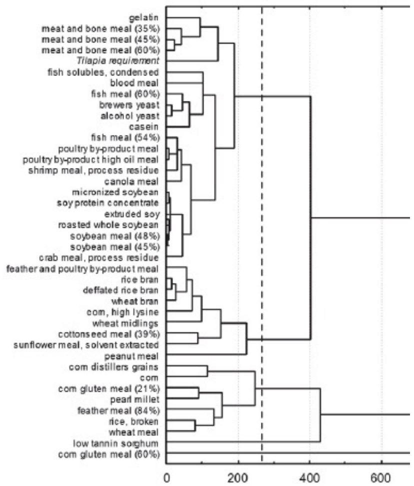
Draw isomorphic subtrees identically (same shape)

Preserve layout ordering and symmetry

Compact space saving layout (don't waste space)

27

CLUSTER DENDROGRAMS



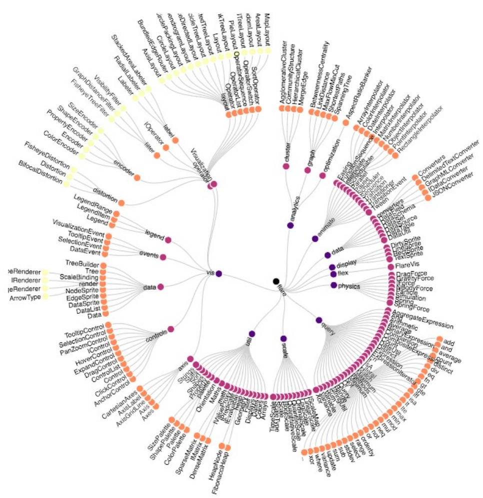
Depicts cluster trees produced by hierarchical clustering algorithms

Leaf nodes arranged in line, internal node depth indicates order/value at which clusters merge

Apply basic recursive layout with orthogonal two-segment edges

59

RADIAL LAYOUT



Node-link diagram in polar coordinates

Radius encodes depth, root at center

Angular sectors assigned to subtrees (basic recursive approach)

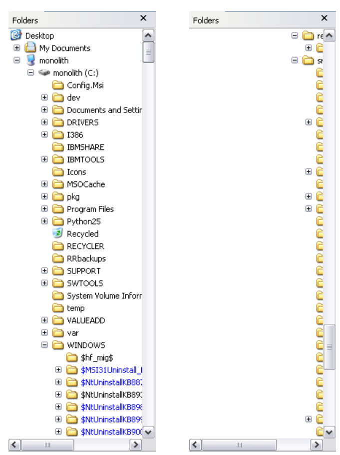
Reingold-Tilford approach can also be applied here

60

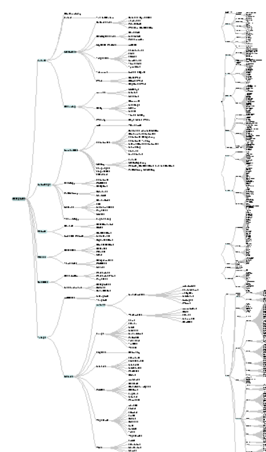
ANALYSIS TASKS: FOCUS + CONTEXT

61

VISUALIZING LARGE HIERARCHIES



Indented Layout



Reingold-Tilford Layout

62

MORE NODES, MORE PROBLEMS…

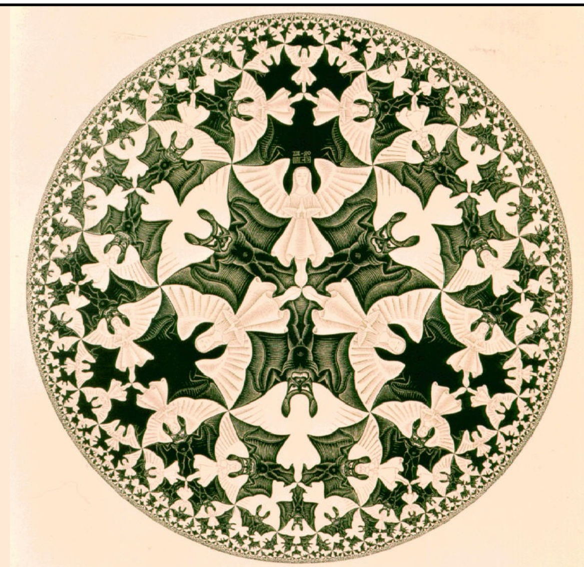
Scale

- Tree breadth often grows exponentially
- Even with tidier layout, quickly run out of space

Possible solutions

- Filtering
- Scrolling or Panning
- Zooming
- Aggregation
- Focus+Context

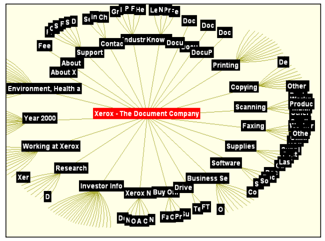
63

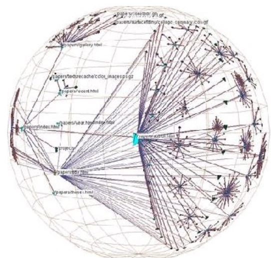


MC Escher, *Circle Limit IV*

64

HYPERBOLIC LAYOUT





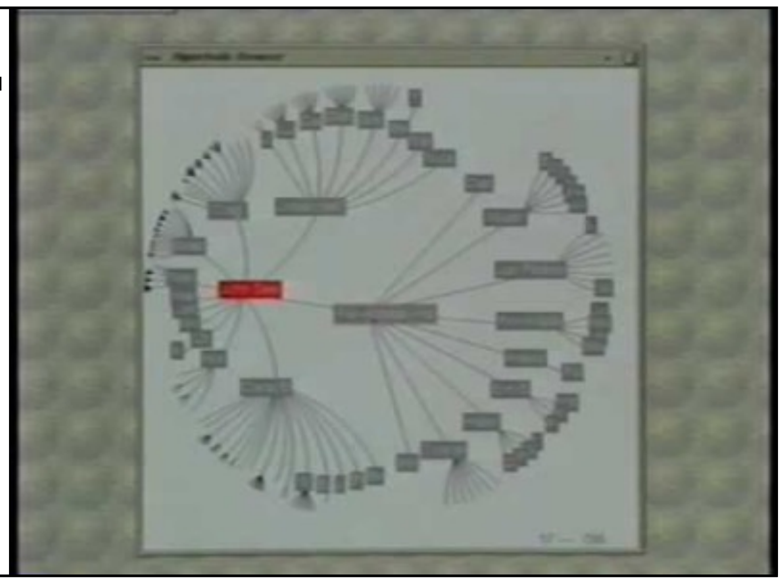
Perform tree layout in hyperbolic space, then project the result on to the Euclidean plane

Why? Like tree breadth, the hyperbolic plane expands exponentially!

Also computable in 3D, projected into a sphere

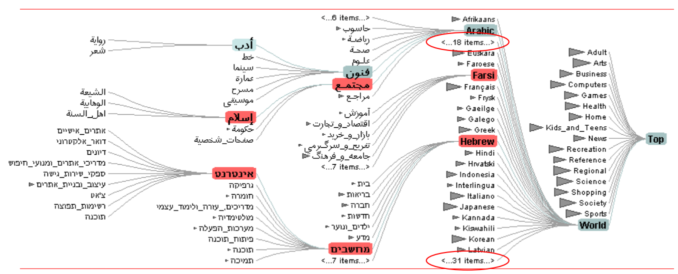
65

HYPERBOLIC LAYOUT [Rao 1995]



66

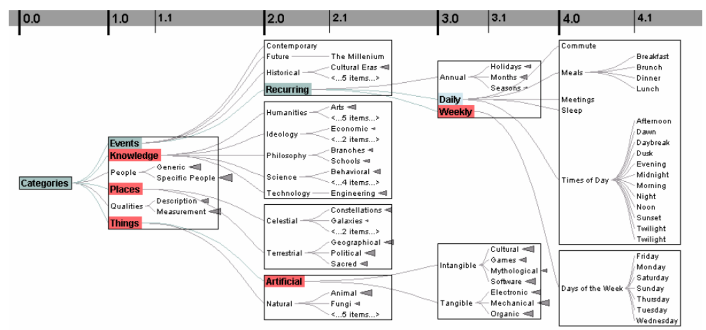
DEGREE OF INTEREST TREES



Space-constrained, multi-focal tree layout

https://www.youtube.com/watch?v=RTQ0N4QY0vc

DEGREE OF INTEREST TREES



Cull “low interest” nodes on a given depth level until all blocks in level fit within bounds
Attempt to center child blocks under parents

https://www.youtube.com/watch?v=RTQ0N4QY0vc

INDENTATION & NODE-LINK DIAGRAMS

Encode structure in **2D space** (breadth/depth)

Benefits

Clearly depicts node relationships / structure

Structure-based or browsing tasks

Problems

Even with tidy layout, quickly run out of space

Missing

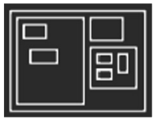
Attribute-based encodings

69

ENCLOSURE

70

ENCLOSURE DIAGRAMS



Encode structure using **spatial enclosure**
Popularly known as **treemaps**

Benefits
Provides a single view of an entire tree
Easier to spot large/small nodes

Problems
Difficult to accurately read structure/depth

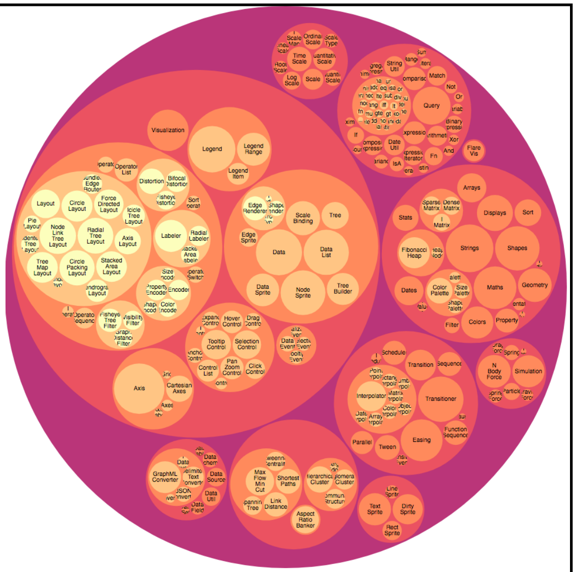
71

CIRCLE PACKING LAYOUT

Nodes represented as sized circles

Nesting to show parent-child relationships

Problems?
- Inefficient use of space
- Parent size misleading?



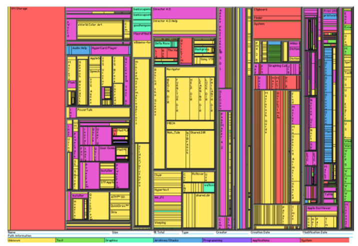
72

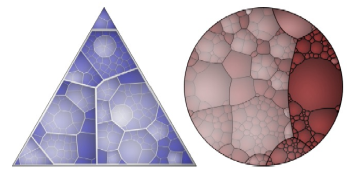
TREEMAPS

Hierarchy visualization that emphasizes values of nodes via area encoding

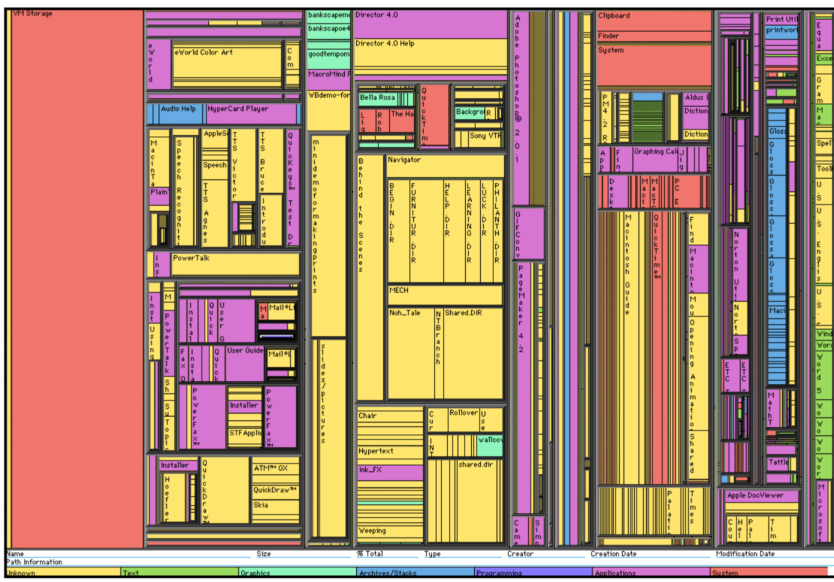
Partition 2D space such that leaf nodes have sizes proportional to data values

First algorithms designed to show file sizes on a hard drive [Shneiderman 1990]





74



***Slide & Dice* layout:** Alternate horizontal / vertical partitions

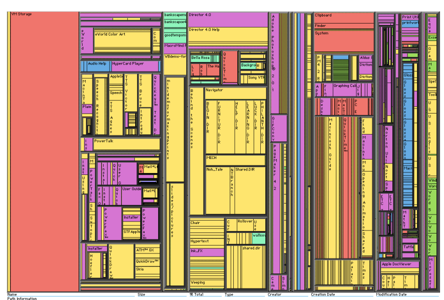
75

***Squarified* layout:** Try to produce square (1:1) aspect ratios

SQUARIFIED TREEMAPS [Bruls 2000]

Slice & Dice layout suffers from extreme aspect ratios. How might we do better?

Squarified layout: greedy optimization with objective of square rectangles. Slice/dice within siblings; alternate whenever ratio worsens



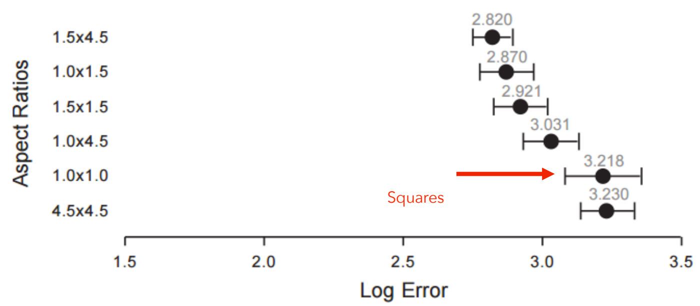
WHY SQUARES?

Posited Benefits of 1:1 Aspect Ratios

1. Minimize perimeter, reducing border ink
   Mathematically true!
2. Easier to select with a mouse cursor.
   Validated by empirical research & Fitt's Law!
3. Similar aspect ratios are easier to compare.
   Seems intuitive, but is this true?

78

COMPARISON: ERROR VS. ASPECT RATIO



Study by Kong et al. 2010

Comparison of squares has higher error! Comparison of extreme ratios even worse

Perhaps squarify works well because it fails to meet its objective?

79

WHY SQUARES?

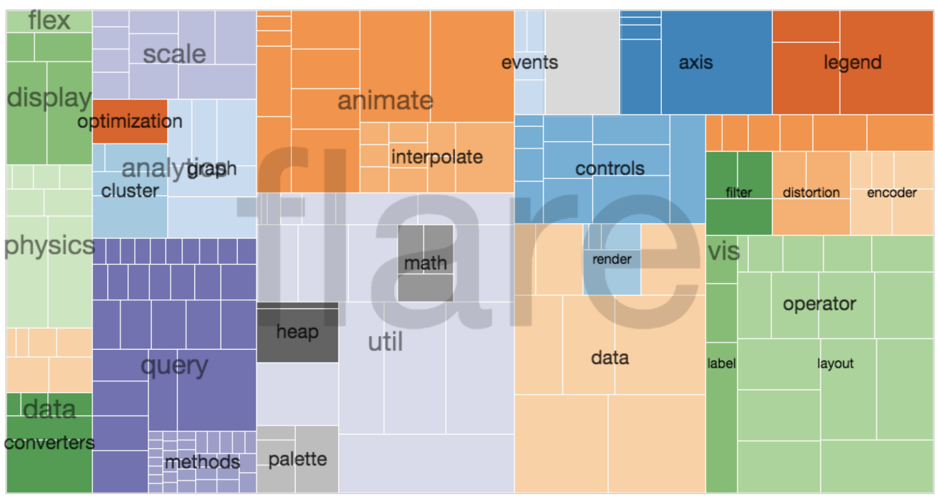
Posited Benefits of 1:1 Aspect Ratios

1. Minimize perimeter, reducing border ink
   Mathematically true!
2. Easier to select with a mouse cursor.
   Validated by empirical research & Fitt's Law!
3. Similar aspect ratios are easier to compare.
   ~~Seems intuitive, but is this true?~~
   Extreme ratios & squares-only, are more inaccurate.
   Balanced ratios better? Target golden ratio?

80

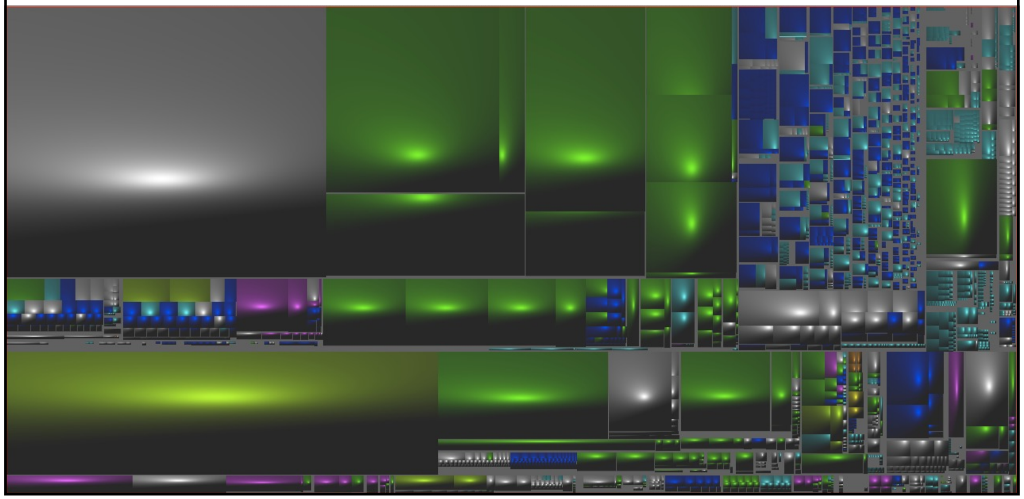
INTERACTIVE EXAMPLE

https://vega.github.io/vega/examples/treemap/



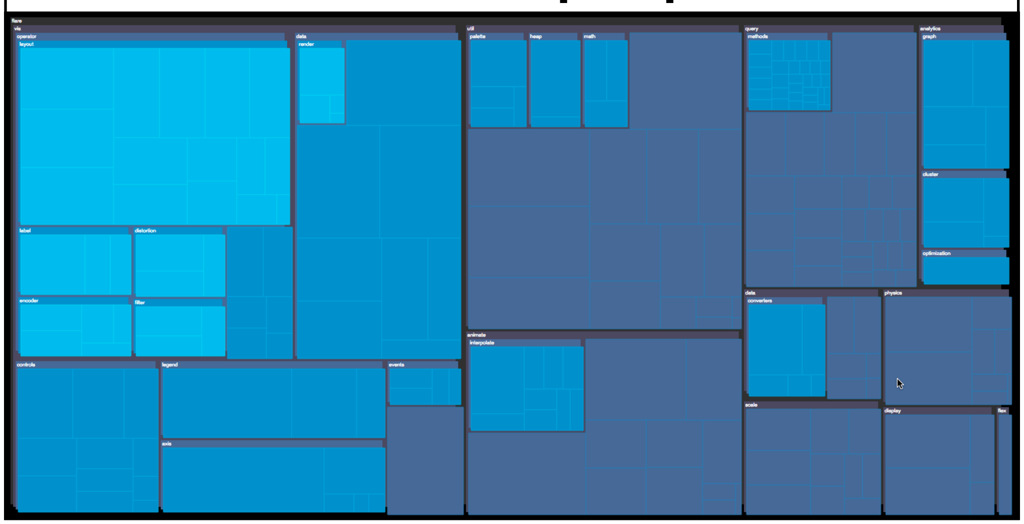
81

CUSHION TREEMAPS [van Wijk 1999]



84

CASCADED TREEMAPS [Lü 2008]

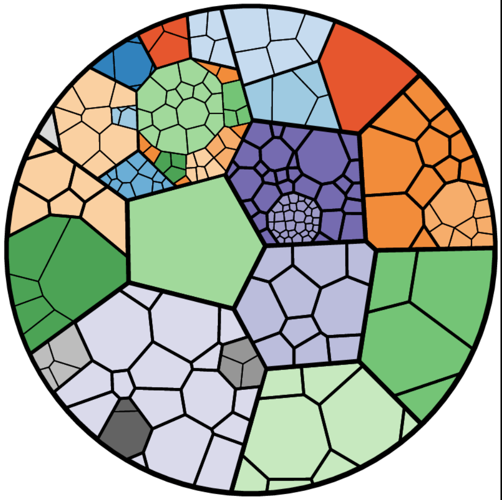


85

VORONOI TREEMAPS [Balzer 2005]

Treemaps with arbitrary polygonal shapes and boundary

Uses iterative, weighted Voronoi tessellations to generate cells with value-proportional areas

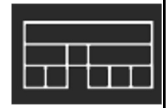


86

LAYERING

88

LAYERED DIAGRAMS



Signify tree structure using
- Layering
- Adjacency
- Alignment

Involves recursive sub-division of space

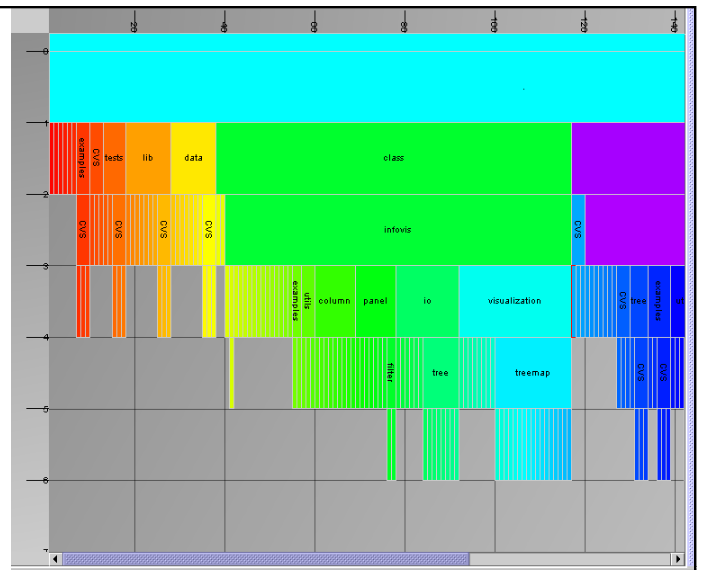
Leaf nodes may be sized by value, parent size visualizes sum of descendant leaf values

89

ICICLE TREES

Parent nodes get a larger layer area, (e.g. horizontal extent)

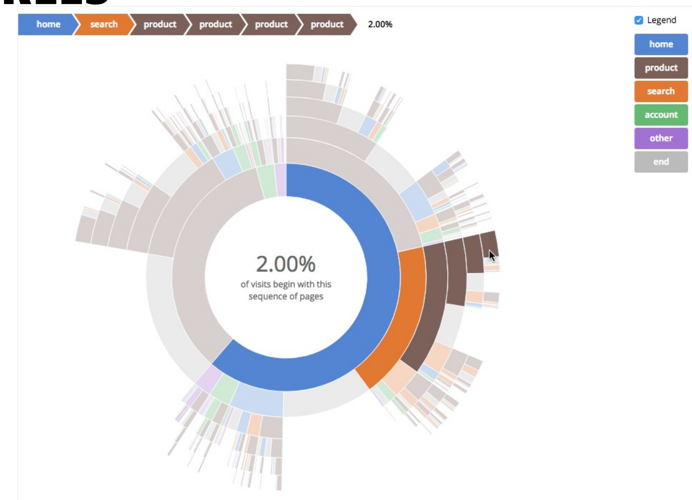
Child levels are layered, constrained to parent's extent



90

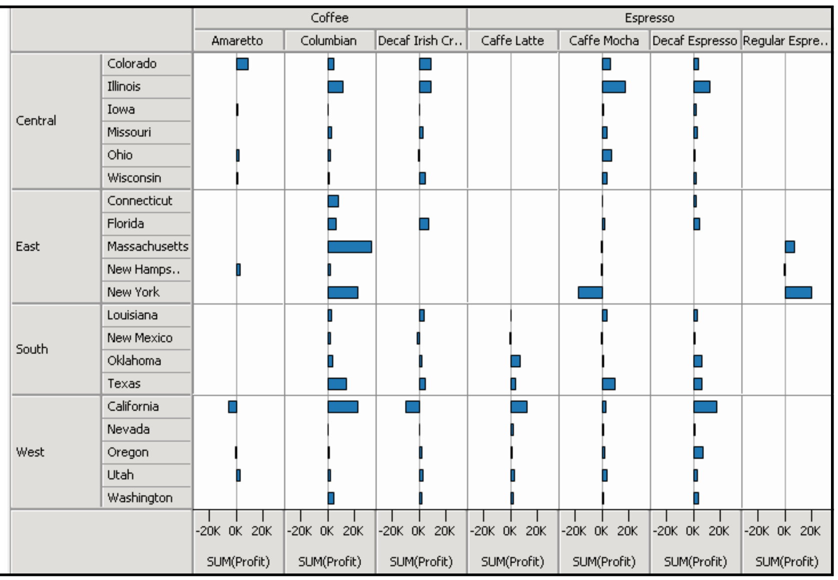
SUNBURST TREES

- Parent nodes get a larger layer area, (e.g. angular extent)
- Child levels are layered, constrained to parent's extent



91

ANOTHER USE OF LAYERED TREES



92

ANNOUNCEMENTS

94

FINAL PROJECT

Design Reviews Dec 2 and Dec 4

Data analysis/explainer

Analyze dataset in depth & make a visual explainer

Deliverables

An article with multiple different interactive visualizations

Short video (2 min) demoing and explaining the project

Schedule

Project proposal: Today!

Design Review and Feedback: 10th week of quarter, 12/2 and 12/4

Final code and video: Sun 12/8 8pm

Grading

Groups of up to 3 people, graded individually

Clearly report responsibilities of each member

95

FINAL PROJECT GUIDELINES

Consider the audience
Your visual explainer should be of interest to a group of people beyond your immediate circle (an explainer about your own Spotify data unlikely be of interest to others you don't know)

Pick relatively less explored topics/datasets
Do some research on what has already been done for the topic/dataset(s)

Certain data like songs (e.g. Spotify) or movies (e.g. IMDB) are already well analyzed and should be avoided, unless you want to try to take a very different angle or use innovative analysis methods

Develop a narrative
In the early stages of the analysis process, try to uncover patterns to help you form and shape a narrative through-line for the explainer

96

FINAL PROJECT GUIDELINES

Design visualization interactions
Choose base visualizations that can support a high level of interactivity
- Bubble charts, tree maps, and word clouds typically aren't the most effective choices

Design interactive features that would enable viewers to interact with the data in a way that strengthens your narrative
- Tooltip is typically not enough interaction

Draw inspiration from sites like the New York Times and the Pudding

97

NODE-LINK GRAPH LAYOUT

98

NODE-LINK GRAPH VISUALIZATION

Nodes connected by lines/curves

Sugiyama-Style Layout - arranged by depth

Force-Directed Layout - physical simulation

Attribute-Driven Layout - arranged by value

Constraint-Based Layout – optimization

Arc Diagrams - aligned layout

99

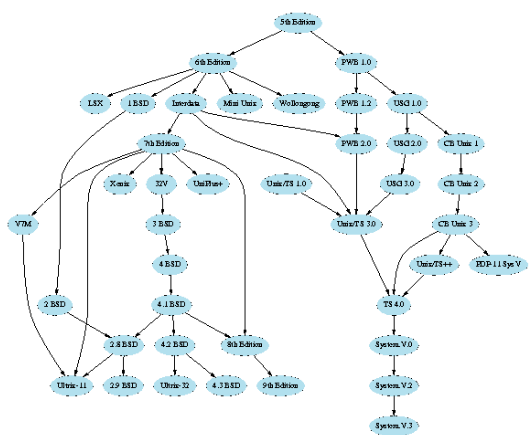
SUGIYAMA-STYLE LAYOUT

100

SUGIYAMA-STYLE GRAPH LAYOUT

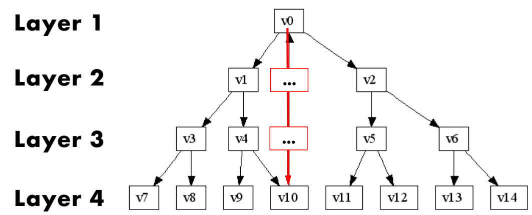
Evolution of the UNIX operating system

Hierarchical layering based on descent



101

SUGIYAMA-STYLE GRAPH LAYOUT

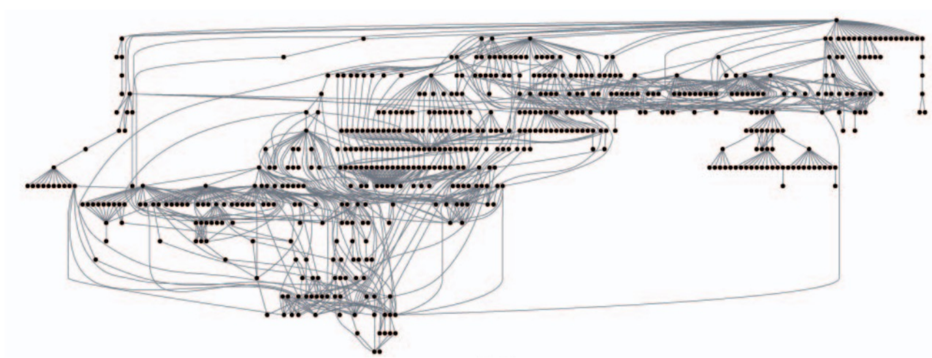


Reverse some edges to remove cycles (if not already a DAG)
Assign nodes to hierarchy layers → Longest path layering
- Create dummy nodes to "fill in" missing layers

Arrange nodes within layer, minimize edge crossings
- Route edges – layout splines if needed

102

PRODUCES HIERARCHICAL LAYOUT

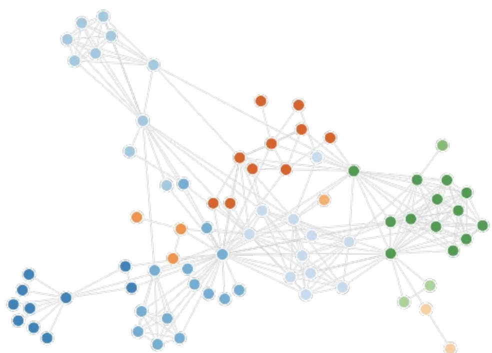


Sugiyama-style layout emphasizes hierarchy
- However, cycles in the graph may not be as apparent, and hierarchy may mislead
- Long edges can impede perception of proximity

103

FORCE-DIRECTED LAYOUT

104



Interactive Example: Configurable Force Layout

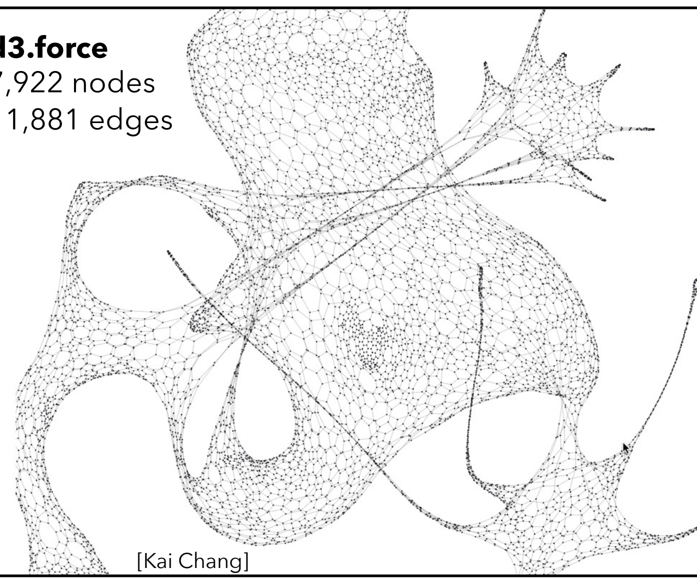
105

Use the Force!

http://mbostock.github.io/d3/talk/20110921/

107

d3.force
7,922 nodes
11,881 edges



[Kai Chang]

108

LAYOUT BY PHYSICS SIMULATION

Nodes = charged particles with air resistance — $F = q_i * q_j / d_{ij}^2$, $F = -b * v_i$

Edges = springs — $F = k * (L - d_{ij})$

At each timestep, calculate forces acting on nodes.

Integrate for updated velocities and positions.

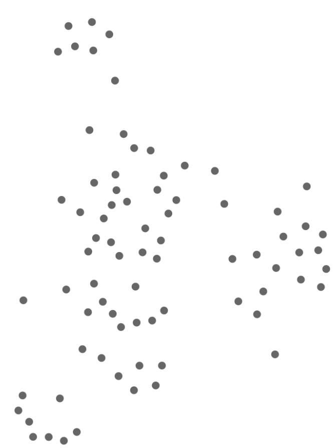
D3's force layout uses **velocity Verlet** integration

Assume uniform mass *m* and timestep *Δt*:

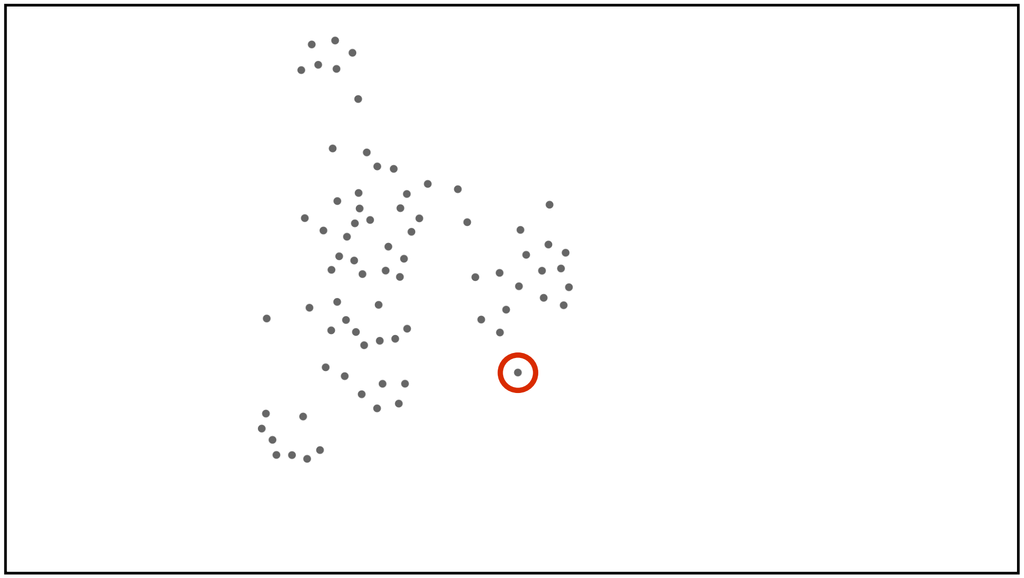
$F = ma \rightarrow F = a \rightarrow F = \Delta v / \Delta t \rightarrow F = \Delta v$

Forces simplify to velocity offsets!

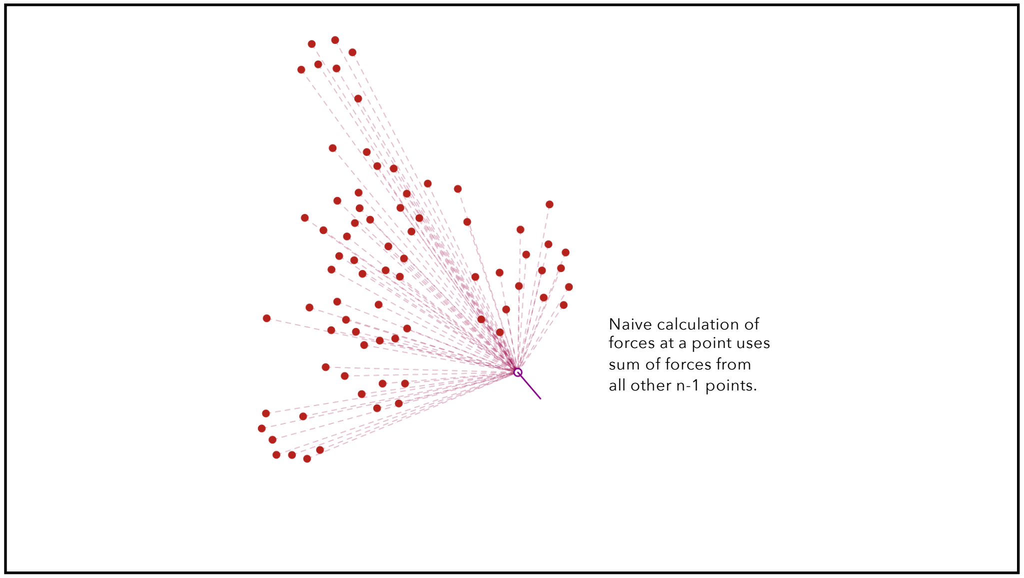
109



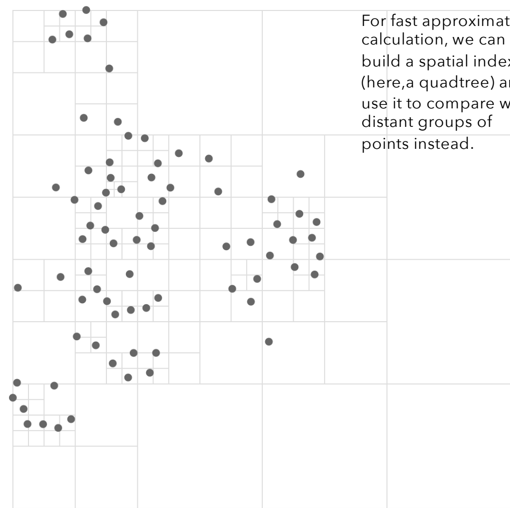
110



111

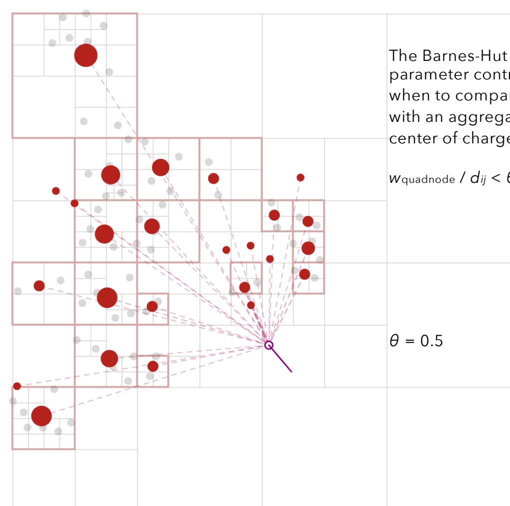

Naive calculation of
forces at a point uses
sum of forces from
all other n-1 points.

112



For fast approximate calculation, we can build a spatial index (here,a quadtree) and use it to compare with distant groups of points instead.

113

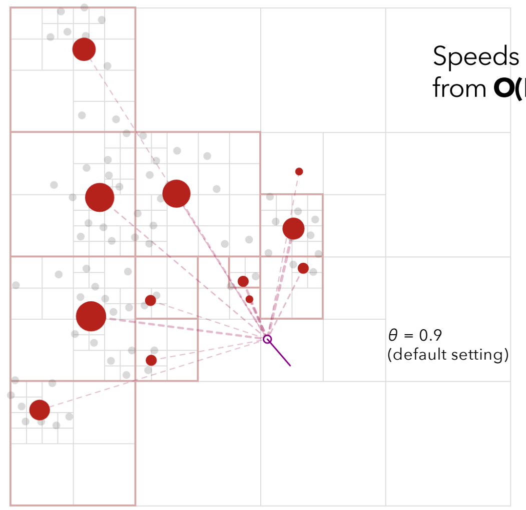


The Barnes-Hut θ parameter controls when to compare with an aggregate center of charge.

$w_{\text{quadnode}} / d_{ij} < \theta$?

$\theta = 0.5$

114



Speeds up algorithm from **O(N²)** to **O(NlogN)**

115

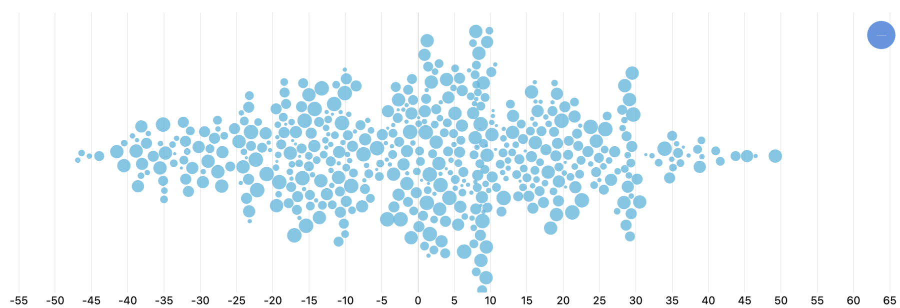
CUSTOMIZED FORCE LAYOUTS

https://www.amcharts.com/demos/beeswarm/

Different forces can be composed to create variety of custom layouts

A **beeswarm plot** can be made by combining:

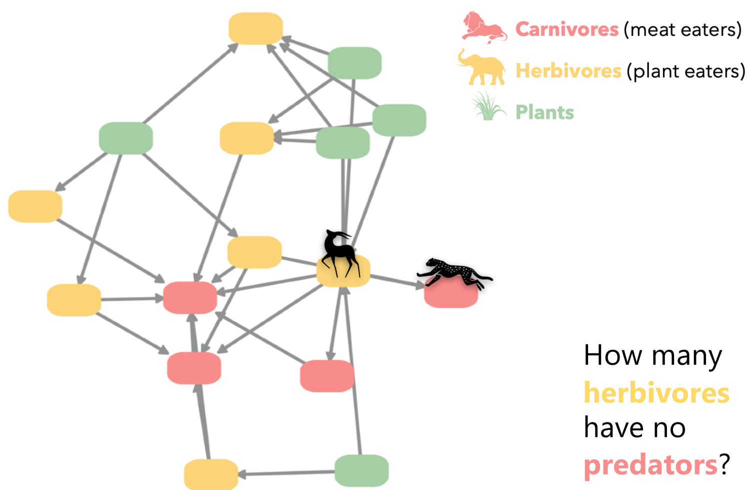
- Attractive **X** and **Y** forces to draw nodes of a certain category to a desired point
- **Collide** force to detect collision & remove overlap



119

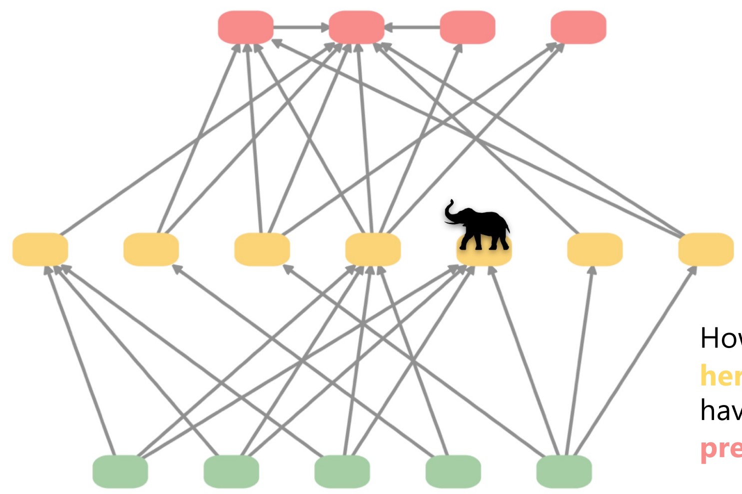
ATTRIBUTE-DRIVEN LAYOUT

120



How many **herbivores** have no **predators**?

121



122

ATTRIBUTE-DRIVEN LAYOUT

Large node-link diagrams **get messy**!

Can we exploit additional structure?

Idea: Use **data fields/attributes** associated with nodes or edges to perform layout (e.g., scatter plot based on node values)

Attributes may also be statistical properties of the graph

Can apply dynamic queries & brushing on attributes/fields to explore…

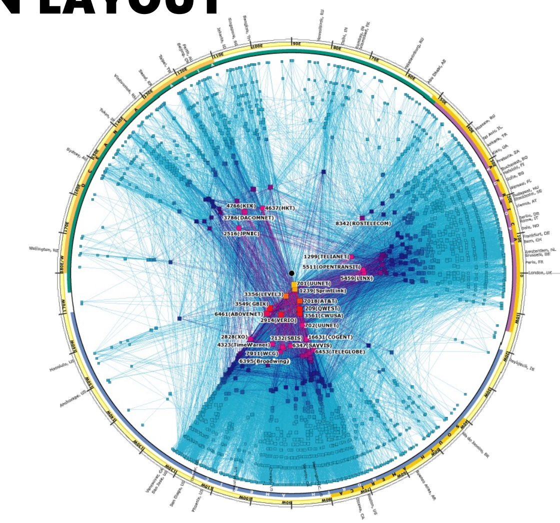
123

ATTRIBUTE-DRIVEN LAYOUT

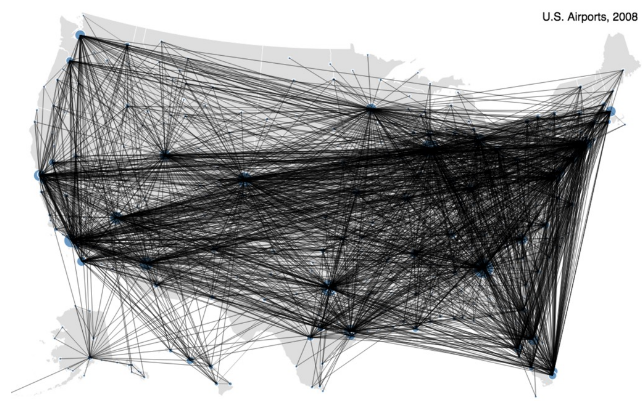
The "Skitter" Layout
Internet Connectivity
Radial Scatterplot

Angle = Longitude
Geography

Radius = Degree
# of connections
(a statistic of the nodes)

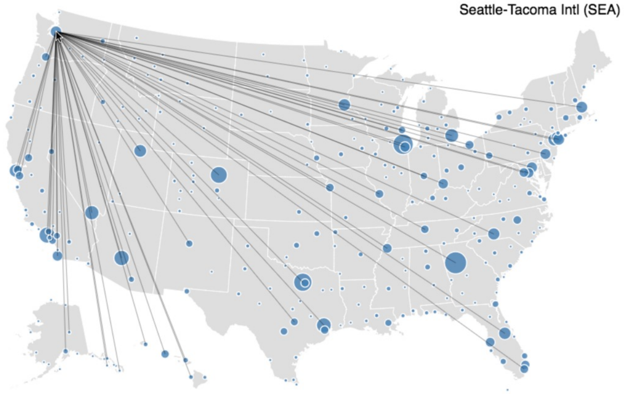


124



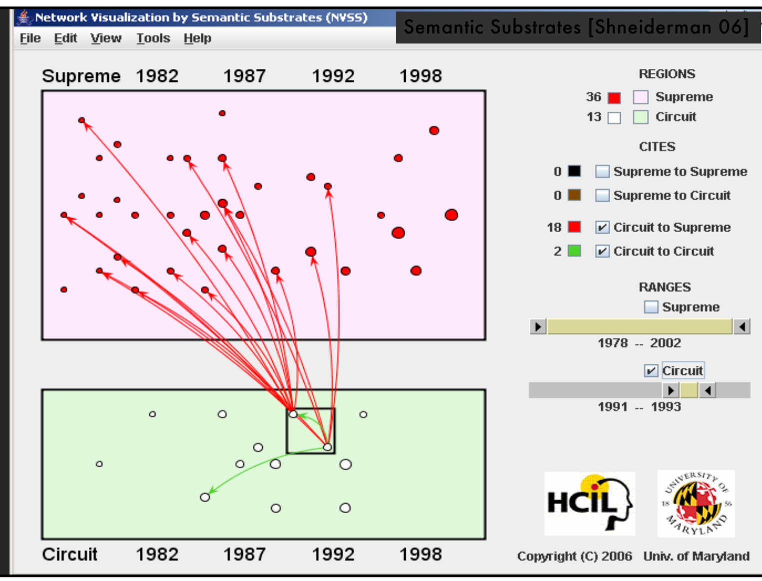
Drawing all edges is not particularly useful here…

125



Node layout determined by geographic location.
Adjacent edges shown on node selection.

126



127

CONSTRAINT-BASED LAYOUT

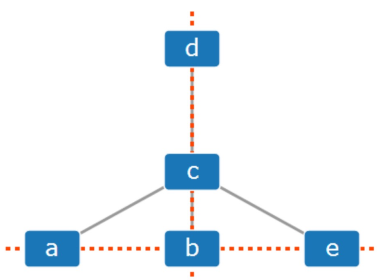
134

CONSTRAINT-BASED LAYOUT

Treat layout as an optimization problem

Define layout using an *energy model* along with *constraint equations* the layout should obey

Use optimization algorithms to solve:



Position Constraints

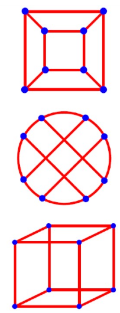
a must be to the ***left*** of b

d, c, and b must have the same ***x position***

a, b, and e must have the same ***y position***

135

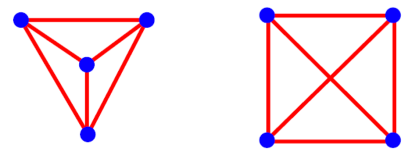
OPTIMIZING AESTHETICS



- Minimize edge crossings
- Minimize area
- Minimize line bends
- Minimize line slopes
- Maximize smallest angle between edges
- Maximize symmetry

but, can't do it all

Optimizing these criteria is often NP-Hard, and requires approximations

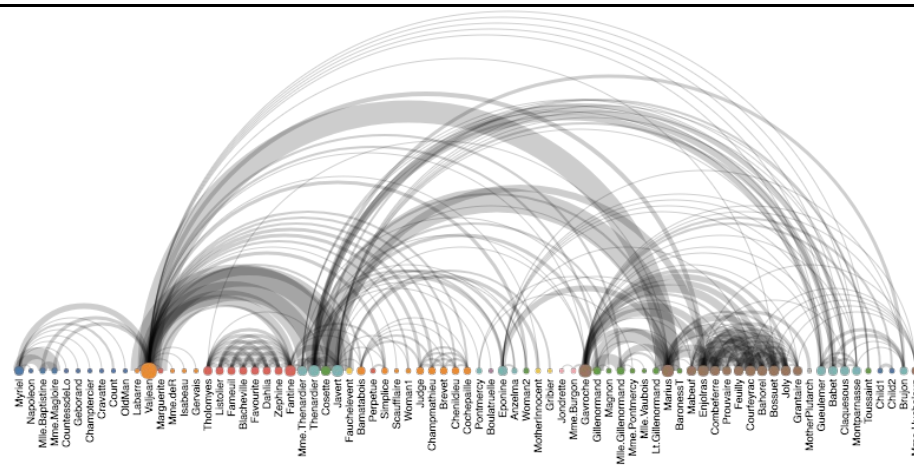


min # crossings   max symmetries

136

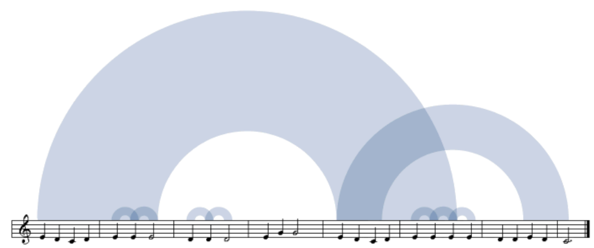
ARC DIAGRAMS

138



Linear node layout, circular arcs show connections.
Layout quality sensitive to node ordering!

139



The Shape of Song
[Wattenberg 2001]

140

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

Force-Directed Layout - physical simulation

Attribute-Driven Layout - arranged by value

Constraint-Based Layout – optimization

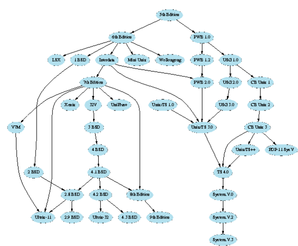
Arc Diagrams - aligned layout

141

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

- **Good:** Structure-based analysis of hierarchical relationships
- **Bad:** Browsing and path following due to long edges



Force-Directed Layout - physical simulation

Attribute-Driven Layout - arranged by value

Constraint-Based Layout – optimization

Arc Diagrams - aligned layout

142

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

Good: Structure-based analysis of hierarchical relationships

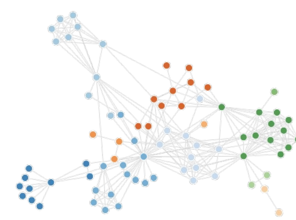
Bad: Browsing and path following due to long edges

Force-Directed Layout - physical simulation

Good: Structure-based analysis of closely related elements

Bad: Browsing and summarization of dense networks

Attribute-Driven Layout - arranged by value



Constraint-Based Layout – optimization

Arc Diagrams - aligned layout

143

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

Good: Structure-based analysis of hierarchical relationships

Bad: Browsing and path following due to long edges

Force-Directed Layout - physical simulation

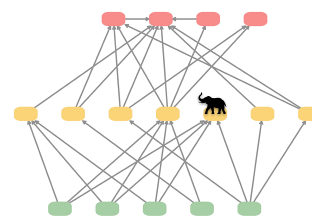
Good: Structure-based analysis of closely related elements

Bad: Browsing and summarization of dense networks

Attribute-Driven Layout - arranged by value

Good: Enables attribute-based analysis tasks

Bad: Difficult to design layouts appropriate to revealing attributes and network structure



Constraint-Based Layout – optimization

Arc Diagrams - aligned layout

144

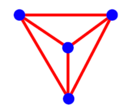
NODE-LINK GRAPH VISUALIZATION

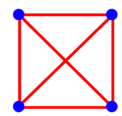
Sugiyama-Style Layout - arranged by depth
- **Good:** Structure-based analysis of hierarchical relationships
- **Bad:** Browsing and path following due to long edges

Force-Directed Layout - physical simulation
- **Good:** Structure-based analysis of closely related elements
- **Bad:** Browsing and summarization of dense networks

Attribute-Driven Layout - arranged by value
- **Good:** Enables attribute-based analysis tasks
- **Bad:** Difficult to design layouts appropriate to revealing attributes and network structure

Constraint-Based Layout – optimization
- **Good:** Graph layout based on structural/aesthetic properties
- **Bad:** Difficult to select appropriate constraints


min # crossings


max symmetries

Arc Diagrams - aligned layout

145

NODE-LINK GRAPH VISUALIZATION

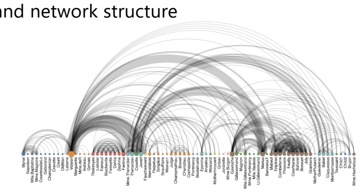
Sugiyama-Style Layout - arranged by depth
- **Good:** Structure-based analysis of hierarchical relationships
- **Bad:** Browsing and path following due to long edges

Force-Directed Layout - physical simulation
- **Good:** Structure-based analysis of closely related elements
- **Bad:** Browsing and summarization of dense networks

Attribute-Driven Layout - arranged by value
- **Good:** Enables attribute-based analysis tasks
- **Bad:** Difficult to design layouts appropriate to revealing attributes and network structure

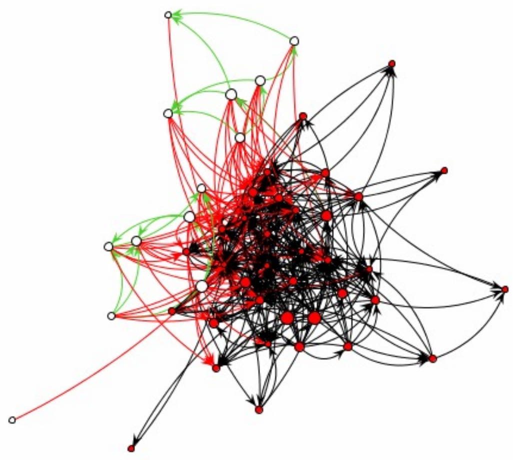
Constraint-Based Layout – optimization
- **Good:** Graph layout based on structural/aesthetic properties
- **Bad:** Difficult to select appropriate constraints

Arc Diagrams - aligned layout
- **Good:** Summarization and comparison of overall structure
- **Bad:** Order matters for node layout; Structure-based and path following



146

LIMITATIONS OF NODE-LINK LAYOUTS



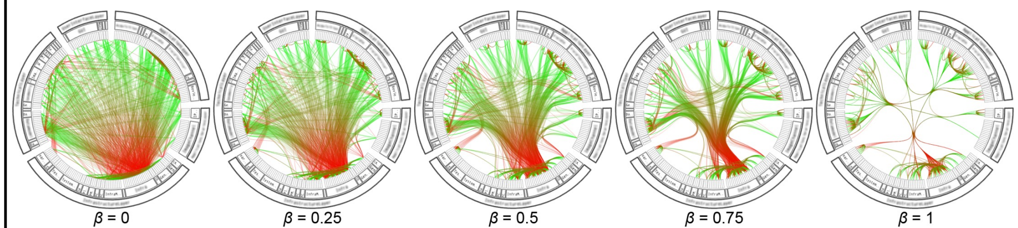
Edge crossings and occlusions!
Poor scalability…

147

HIERARCHICAL EDGE BUNDLING

148

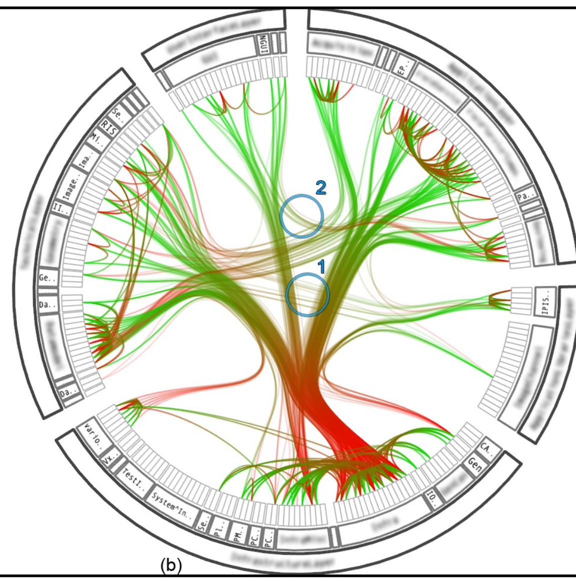
HIERARCHICAL EDGE BUNDLING



Given a tree with additional *adjacency* edges (usually between leaves)

Bundle edges with varying amounts of tension – helping to reveal common connections between subtrees

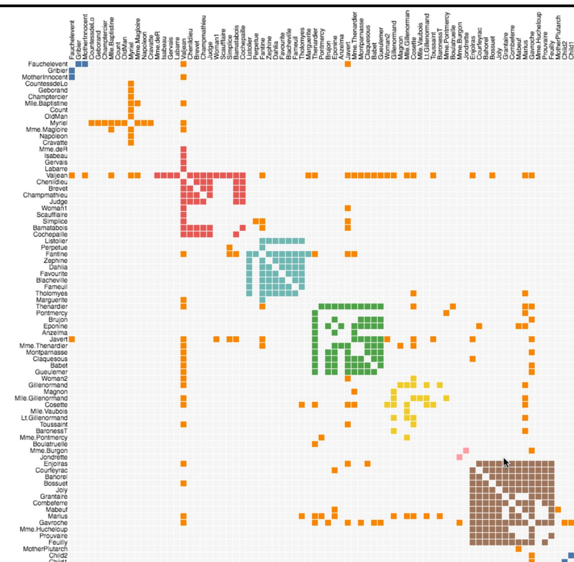
150



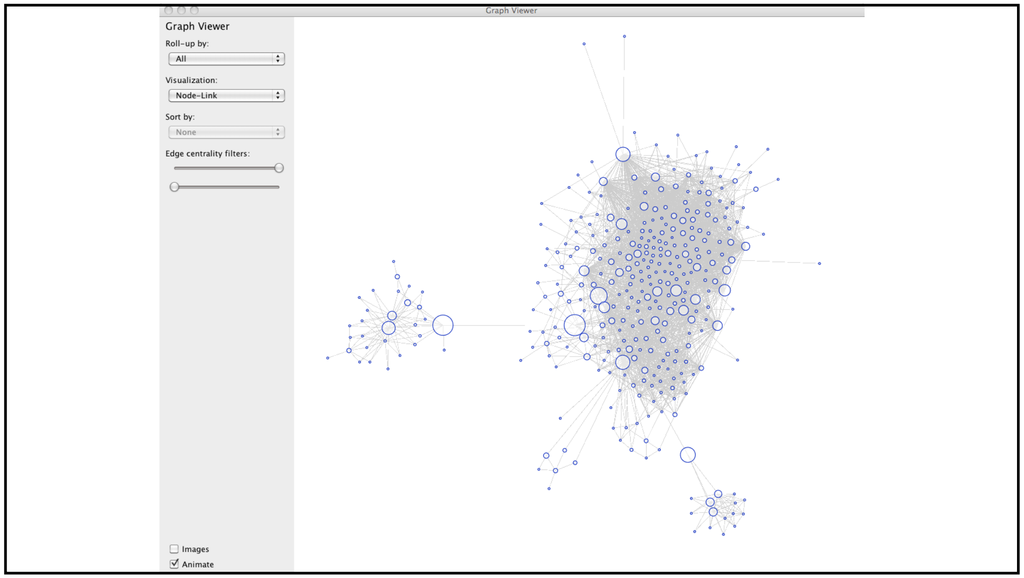
151

MATRIX DIAGRAMS

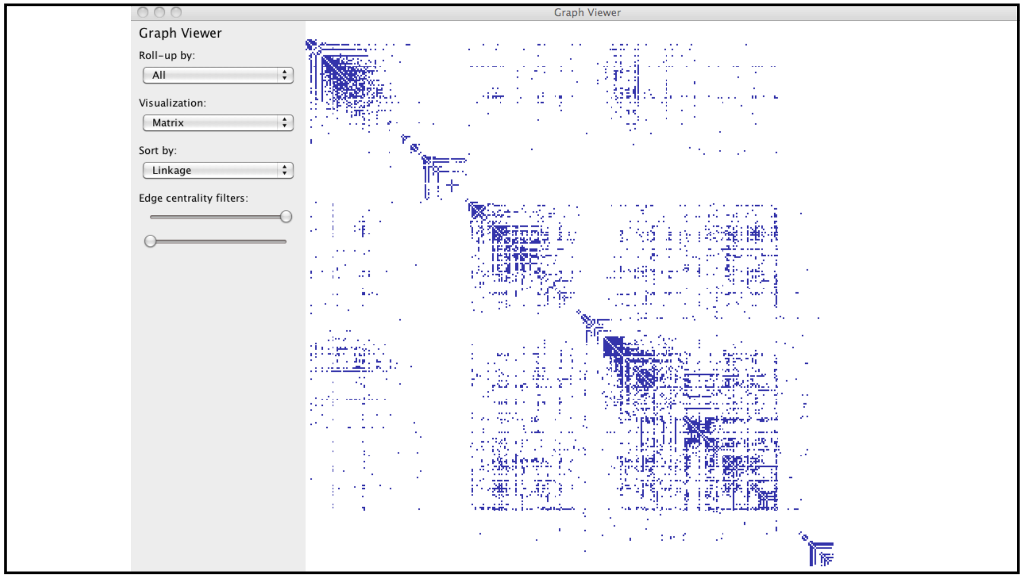
156



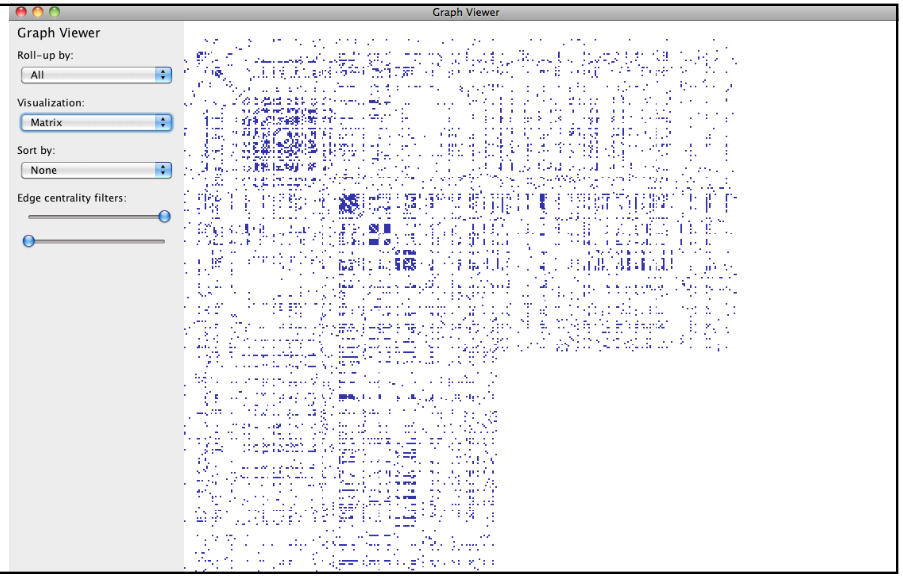
157


Graph Viewer
Graph Viewer
Roll-up by:
All
Visualization:
Node-Link
Sort by:
None
Edge centrality filters:
Images
Animate

158


Graph Viewer
Graph Viewer
Roll-up by:
All
Visualization:
Matrix
Sort by:
Linkage
Edge centrality filters:

159



160

SUMMARY: TREES AND NETWORKS

Tree Layout

Indented / Node-Link / Enclosure / Layers

Focus+Context techniques for scale

Graph Layout

Sugiyama Layout

Force-Directed Layout

Attribute-Driven Layout

Constraint Layout

Arc Diagrams

Matrix Diagrams

161