# VISUALIZING TREES AND GRAPHS

Petra Isenberg

informatics mathematics
Inria

# RECAP

you have learned about

- simple plots
- multi-attribute data visualization

# DATA AND ITS STRUCTURE

## STRUCTURED DATA

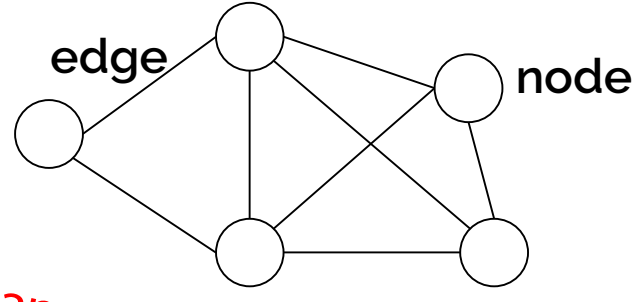
| | | | | |
|---|---|---|---|---|
| 0.103 | 0.176 | 0.387 | 0.300 | 0.379 |
| 0.333 | 0.384 | 0.564 | 0.587 | 0.857 |
| 0.421 | 0.309 | 0.654 | 0.729 | 0.228 |
| 0.266 | 0.750 | 1.056 | 0.936 | 0.911 |
| 0.225 | 0.326 | 0.643 | 0.337 | 0.721 |
| 0.187 | 0.586 | 0.529 | 0.340 | 0.829 |
| 0.153 | 0.485 | 0.560 | 0.428 | 0.628 |

## UNSTRUCTURED DATA

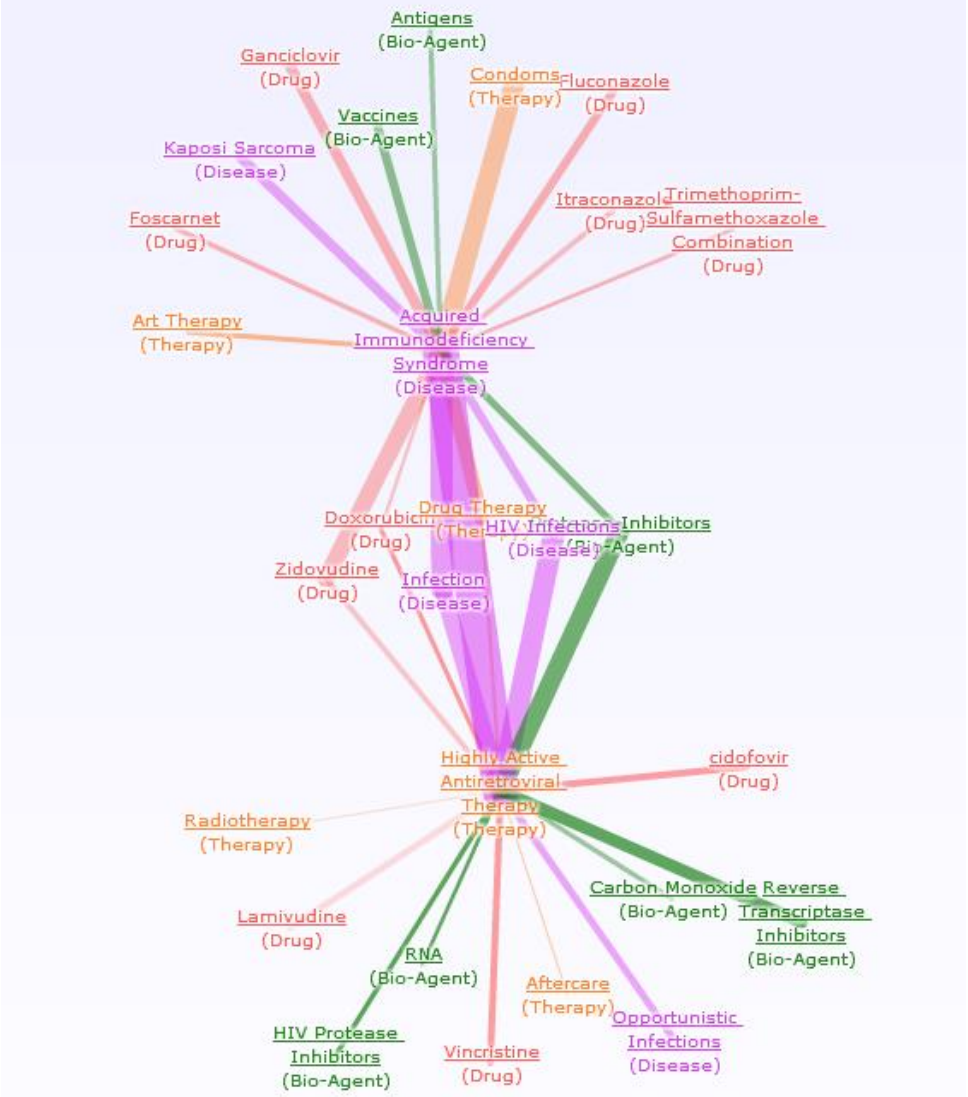

# STRUCTURED DATA

- there are relationships between the data items
- you can use a graph representation

| | | | | |
|---|---|---|---|---|
| 0.103 | 0.176 | 0.387 | 0.300 | 0.379 |
| 0.333 | 0.384 | 0.564 | 0.587 | 0.857 |
| 0.421 | 0.309 | 0.654 | 0.729 | 0.228 |
| 0.266 | 0.750 | 1.056 | 0.936 | 0.911 |
| 0.225 | 0.326 | 0.643 | 0.337 | 0.721 |
| 0.187 | 0.586 | 0.529 | 0.340 | 0.829 |
| 0.153 | 0.485 | 0.560 | 0.428 | 0.628 |




Almost anything can be a graph



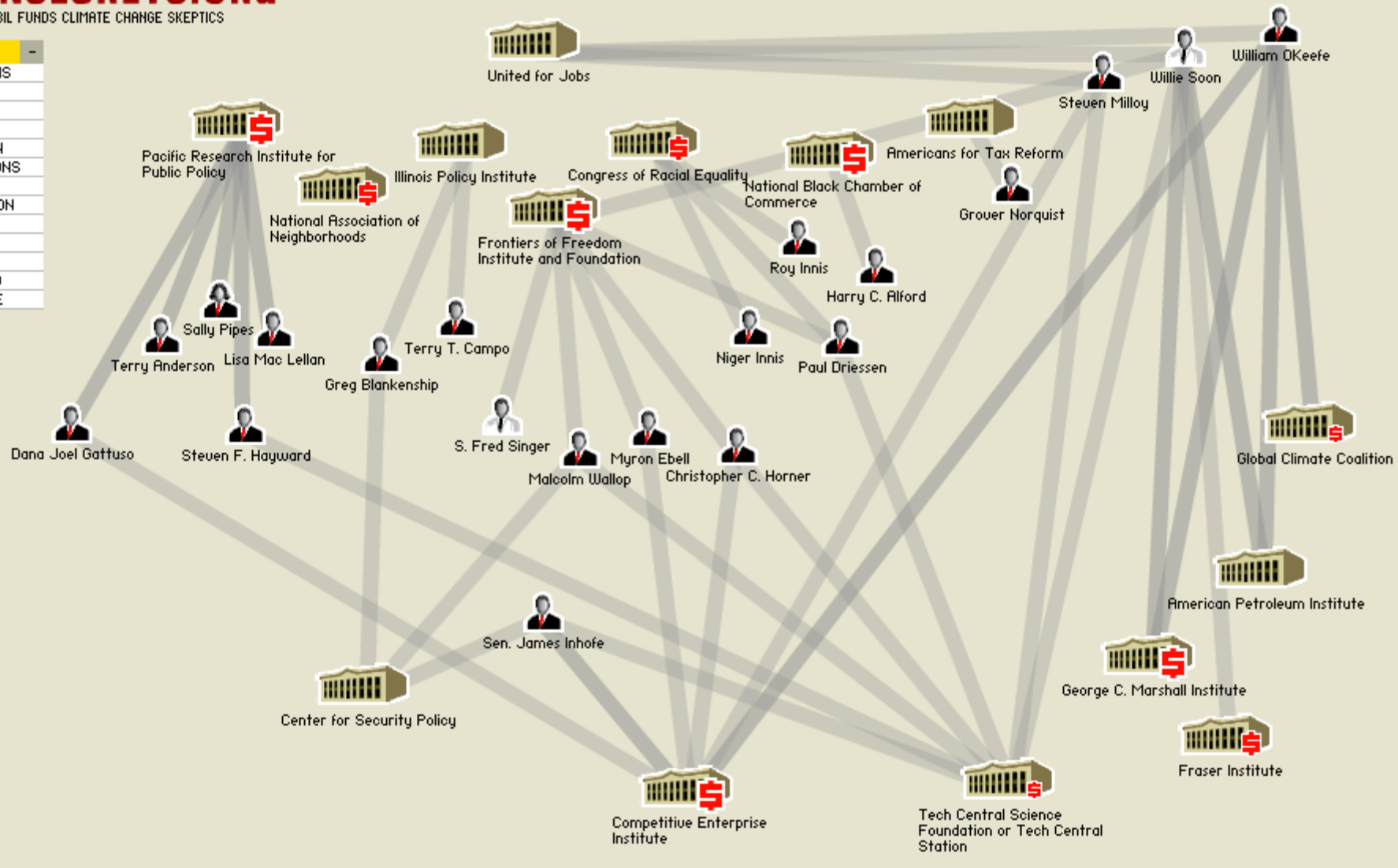

http://www.curehunter.com

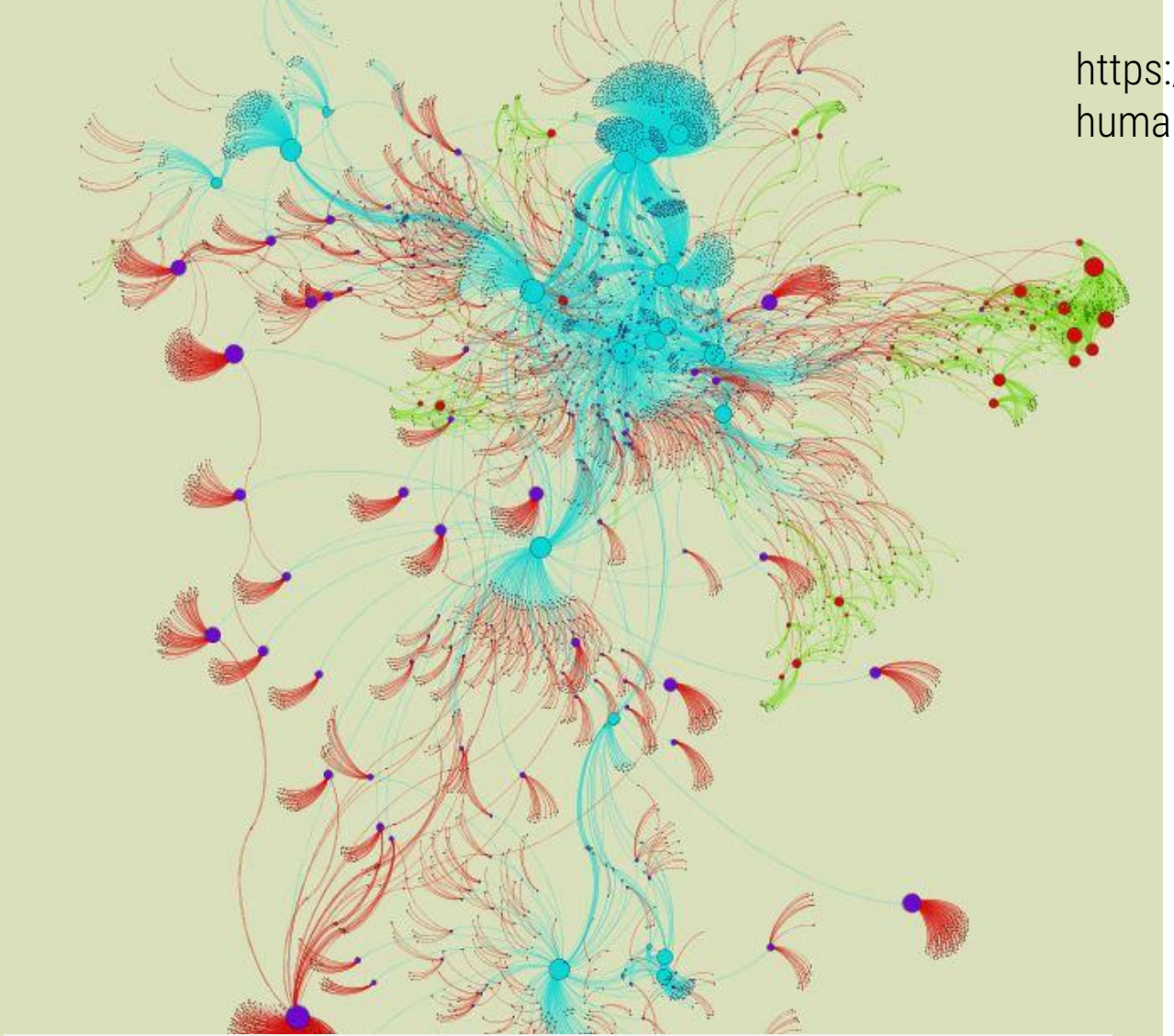
visual dictionary of drugs, diseases and therapies

# EXXONSECRETS.ORG

HOW EXXONMOBIL FUNDS CLIMATE CHANGE SKEPTICS

: : TOOLS -
INSTRUCTIONS
LOAD MAP
SAVE MAP
PRINT MAP
FIND PERSON
ORGANIZATIONS
CLEAR MAP
HTML VERSION
ABOUT
LINKS
SEARCH
KEY/LEGEND
+ SUBSCRIBE

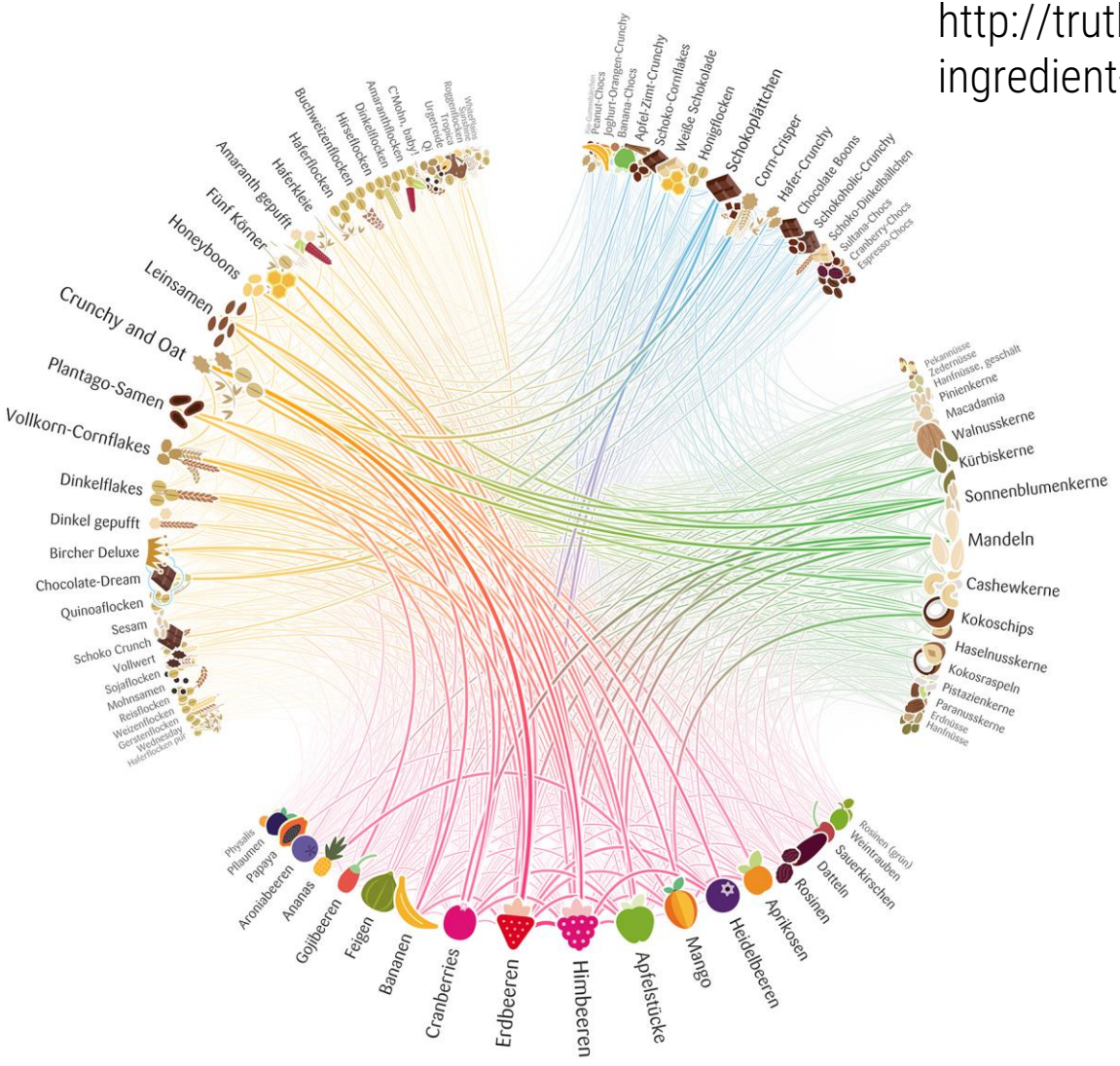




Top Contributors to the Catalogue of Life and their associated species, references and databases

https://dhs.stanford.edu/spatial-humanities/visualizing-databases/

http://truth-and-beauty.net/projects/muesli-ingredient-network

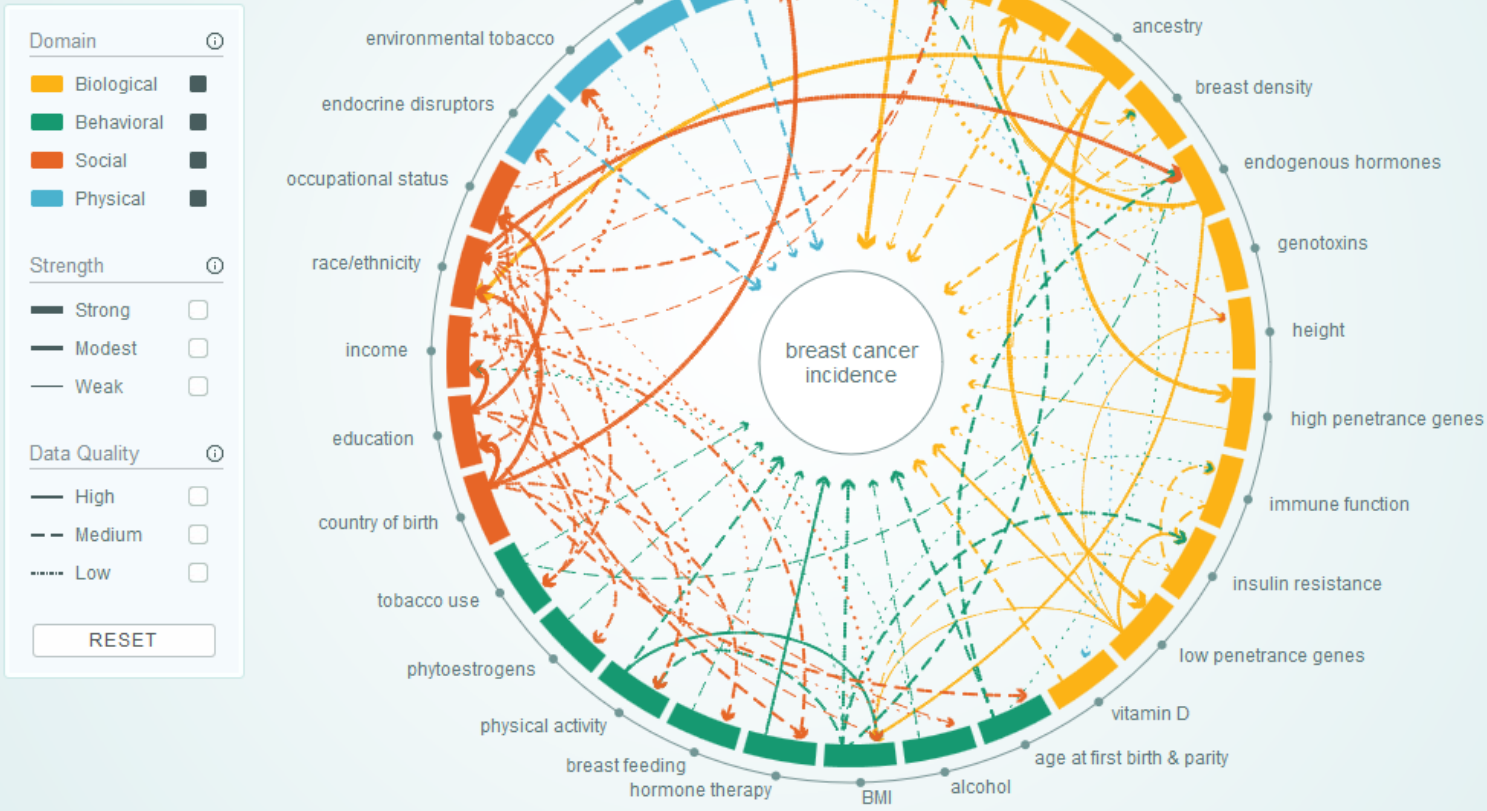


http://www.cabreastcancer.org/causes/#

# A Model of Breast Cancer Causation

*Visualizing the many factors and relationships influencing breast cancer incidence in postmenopausal women*

Definitions | References

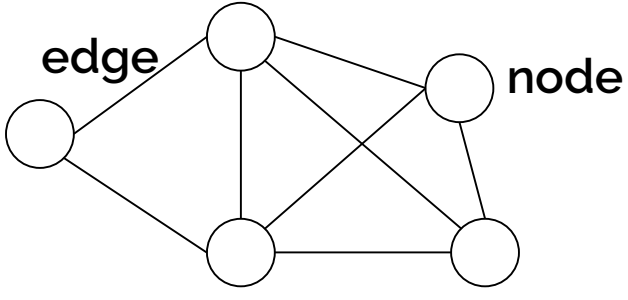


naturevideo

# GRAPHS

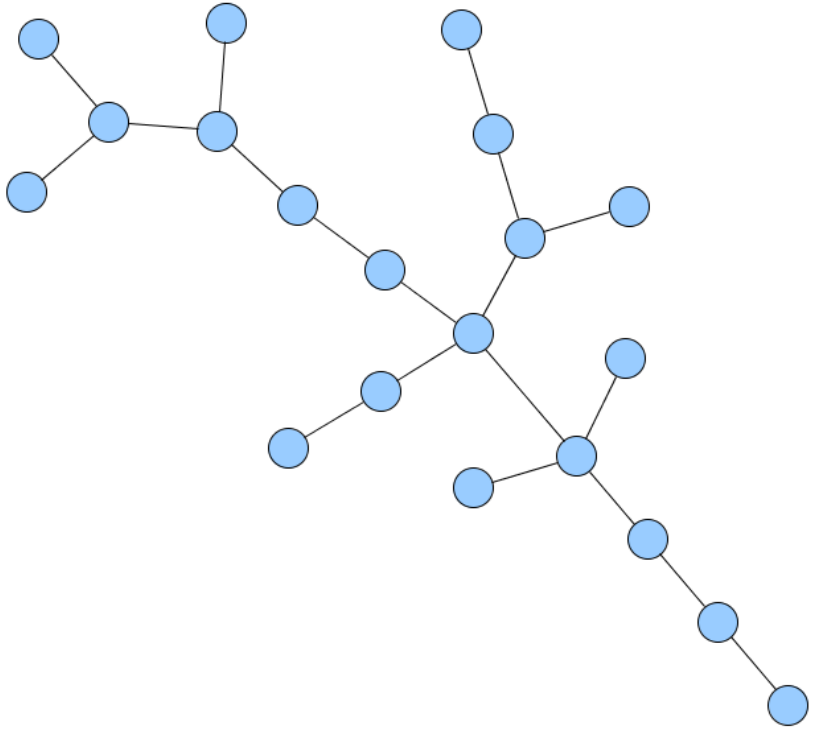
Graphs

- Describe relations among data items
- Using **nodes** and **edges**



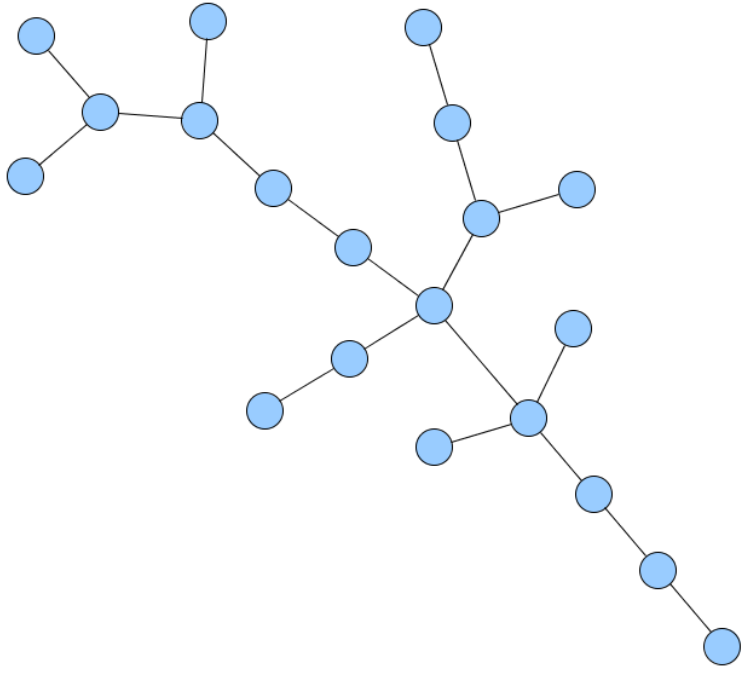
# DEFINITIONS

a tree is a **connected graph with no cycles**



# DEFINITIONS

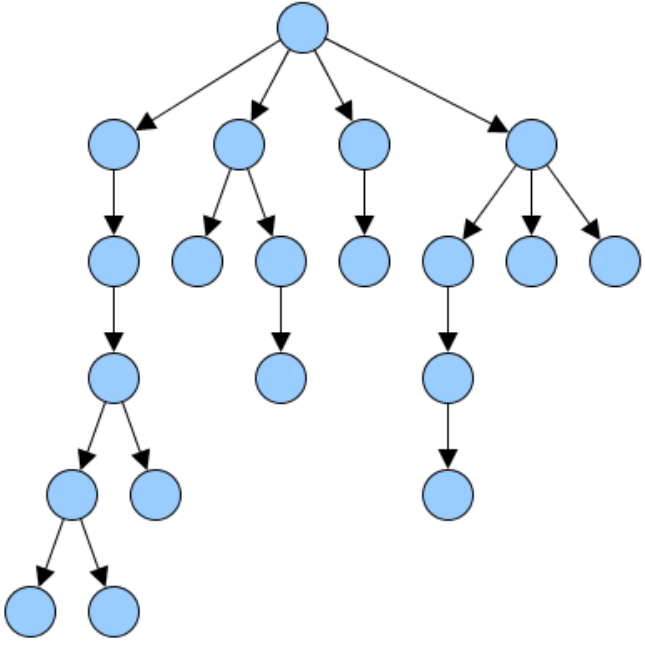
an **undirected tree** is an undirected graph in which any two vertices are connected by exactly one simple path (= a path with no repeated vertices)



# DEFINITIONS

a directed tree is a digraph (directed graph) whose underlying graph is a tree

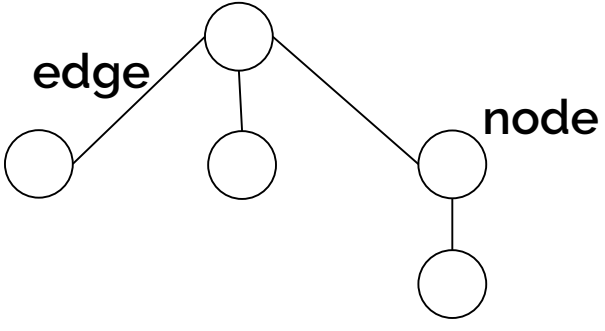
- a directed tree consists of a number of **nodes** and **parent-child relationships**
- every node has just one **parent** and any number of **children**
- *directed trees are the most common form in computer science*



# TREES

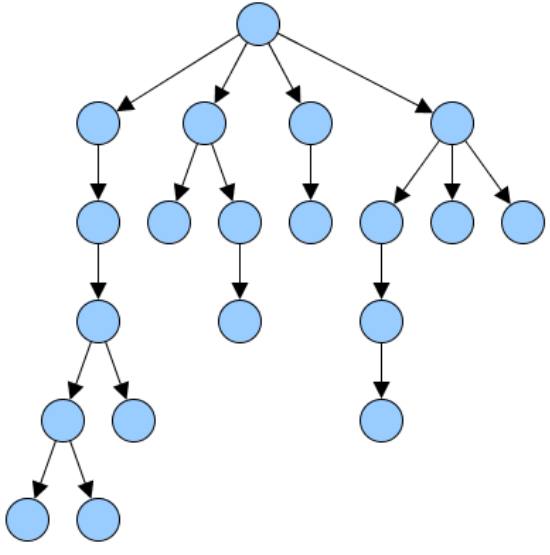
*The most important nonlinear data structure in computer science (Donald Knuth, 1997)*

- Directed acyclic graph / connected graph with n-1 edges
- Nodes have one parent & 0-N children

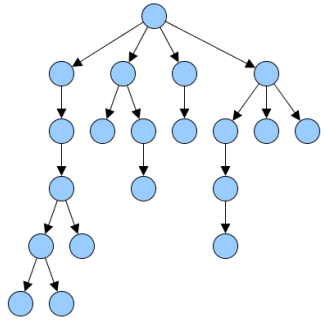


# DEFINITIONS

- the number of children of a node is called its **degree**
- **leaf** nodes are nodes without children



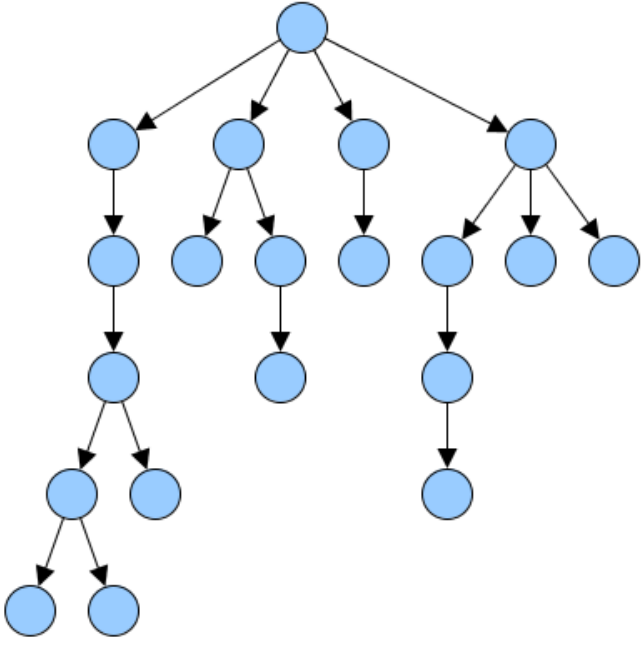
# DEFINITIONS



- a **rooted tree** is a directed tree with a distinguished vertex r, called the **root**, such that for every other vertex v there is directed path from r to v. The root node is the only node with no parent
- any node may act as a root in undirected trees

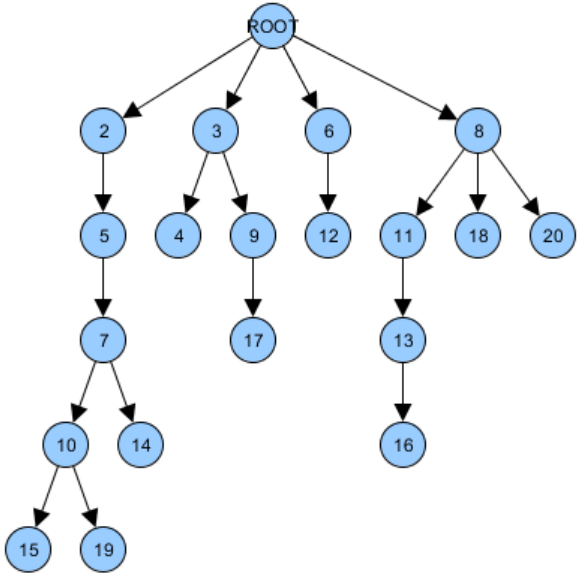
# DEFINITIONS

the connection between parent and child nodes is called an **edge**



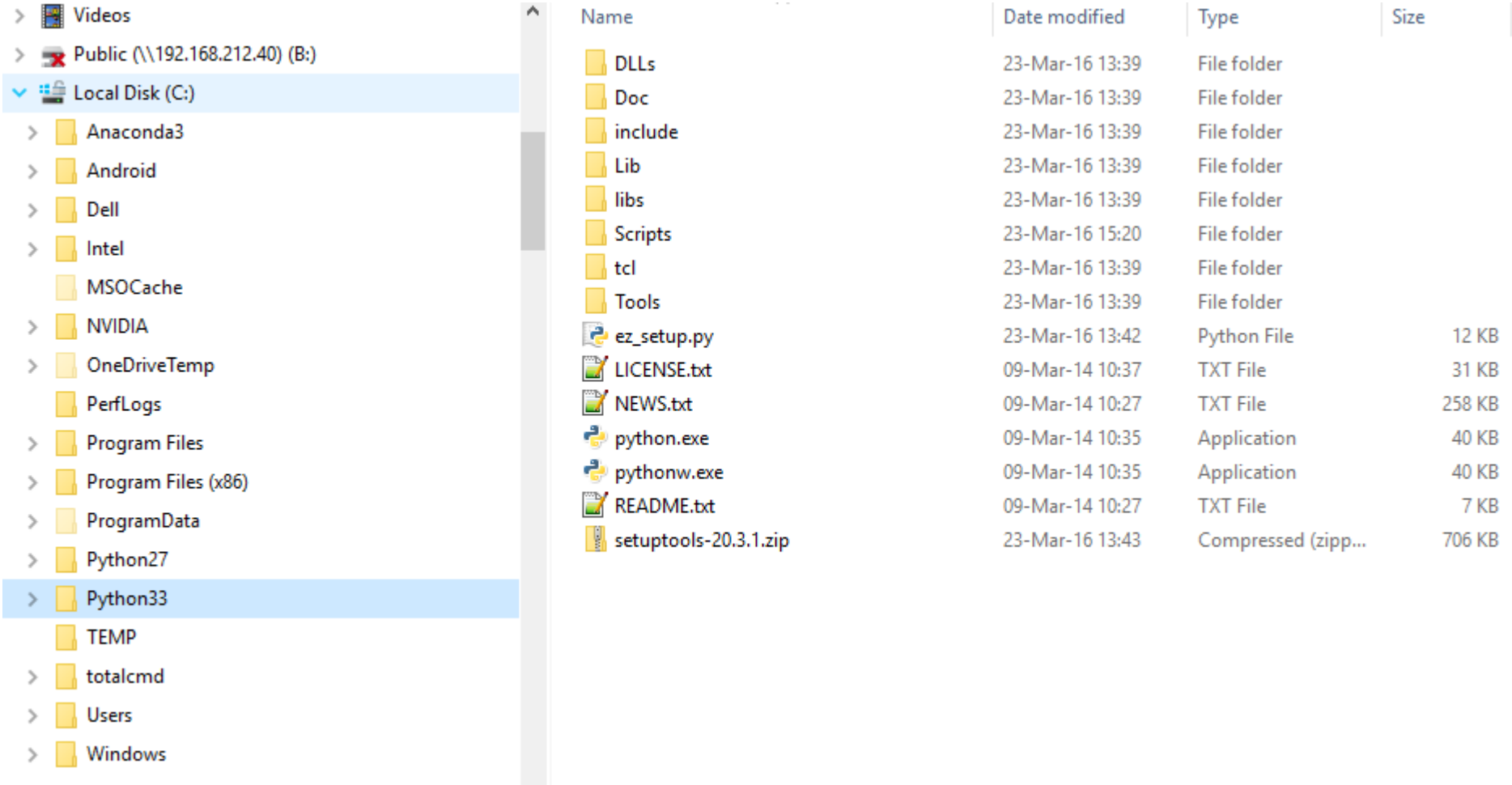
# DEFINITIONS

an **ordered tree** is a rooted tree in which the children of each vertex are assigned a fixed ordering.

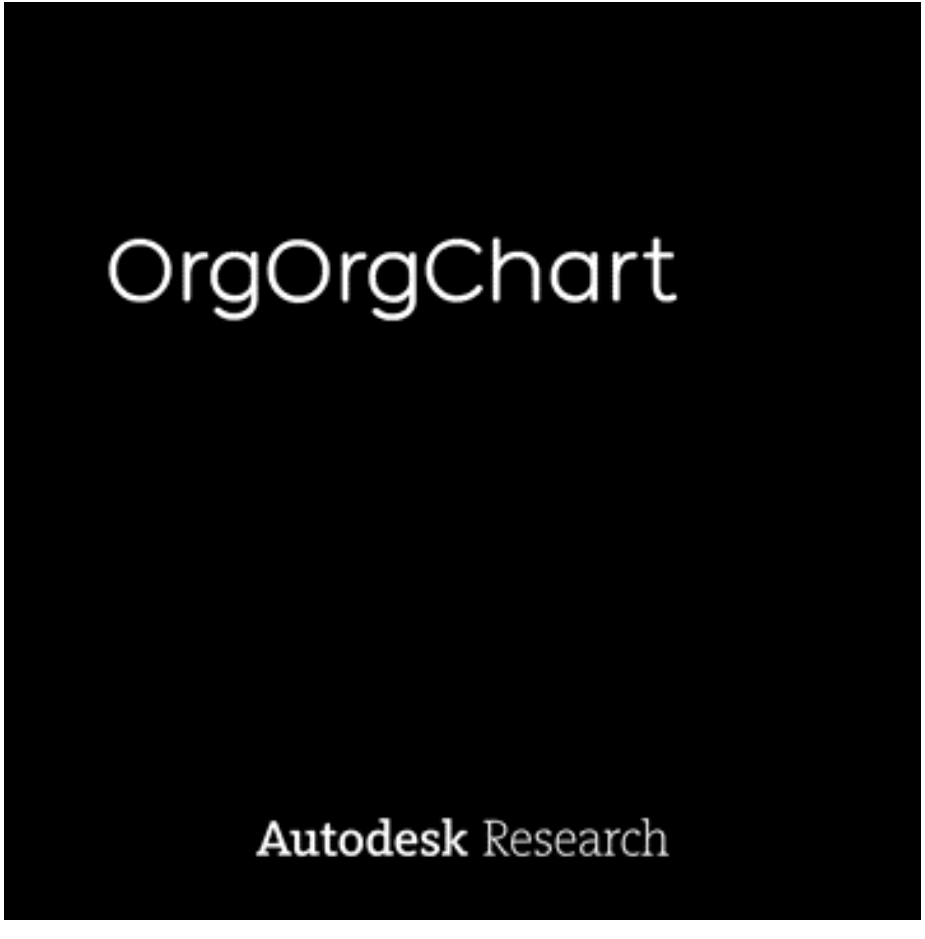


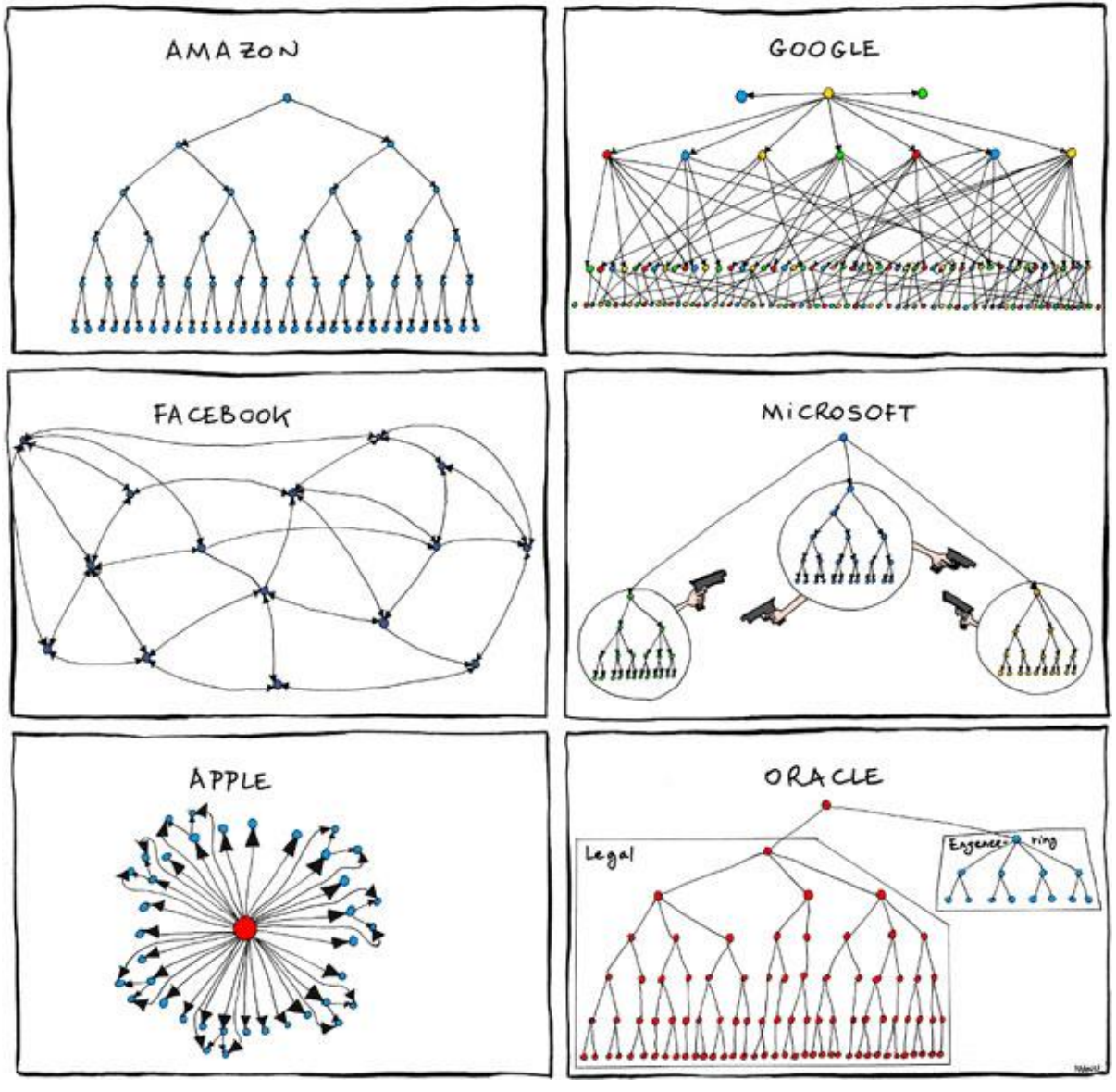
# EXAMPLES OF TREES

# HIERARCHIES



# HIERARCHIES





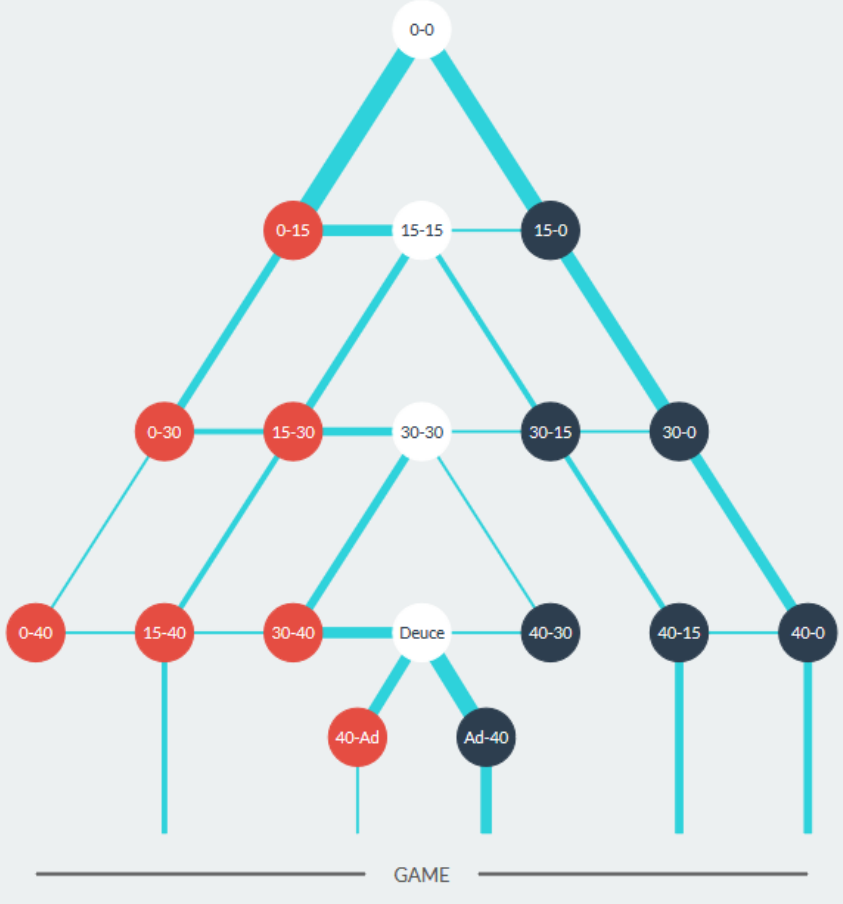

org charts aren't always trees, though

http://www.bonkersworld.net/organizational-charts/

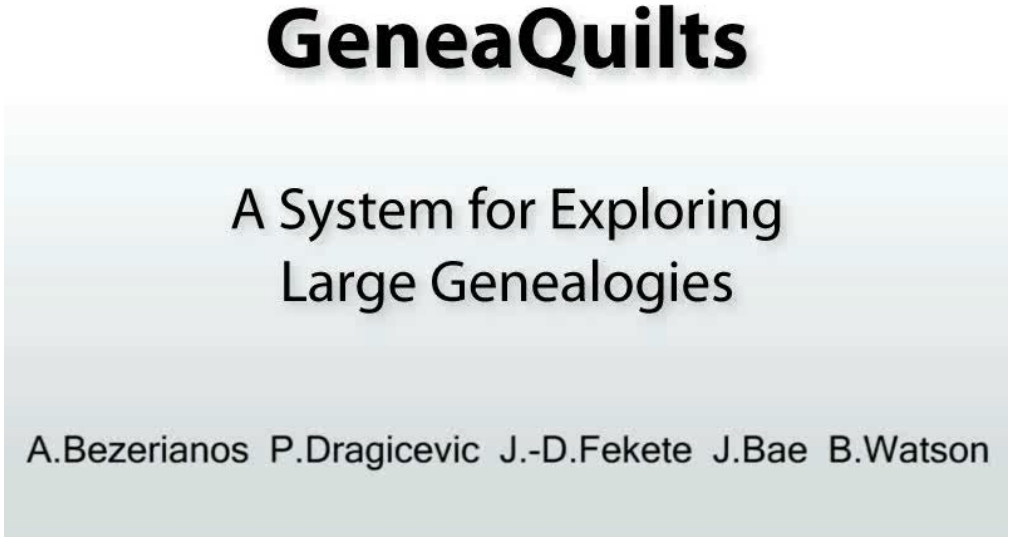
# DECISION PROCESS

## NADAL

- Indian Wells ›
- Monte-Carlo ›
- Madrid ›
- Rome ›
- Roland Garros ›
  - Brands 4-6, 7-6(4), 6-4, 6-3
  - Klizan 4-6, 6-3, 6-3, 6-3
  - Fognini 7-6(5), 6-4, 6-4
  - Nishikori 6-4, 6-1, 6-3
  - Wawrinka 6-2, 6-3, 6-1
  - Djokovic 6-4, 6-3, 6-1, 6-7(3), 9-7
  - Ferrer 6-3, 6-2, 6-3
- Wimbledon ›
- Rogers Cup ›



# BRANCHING PROCESSES




Think about it: is a family tree really a tree?

# TREE REPRESENTATION

TECHNIQUES

How to cite this site?

Check out other surveys!

# treevis.net - A Visual Bibliography of Tree Visualization 2.0 by Hans-Jörg Schulz

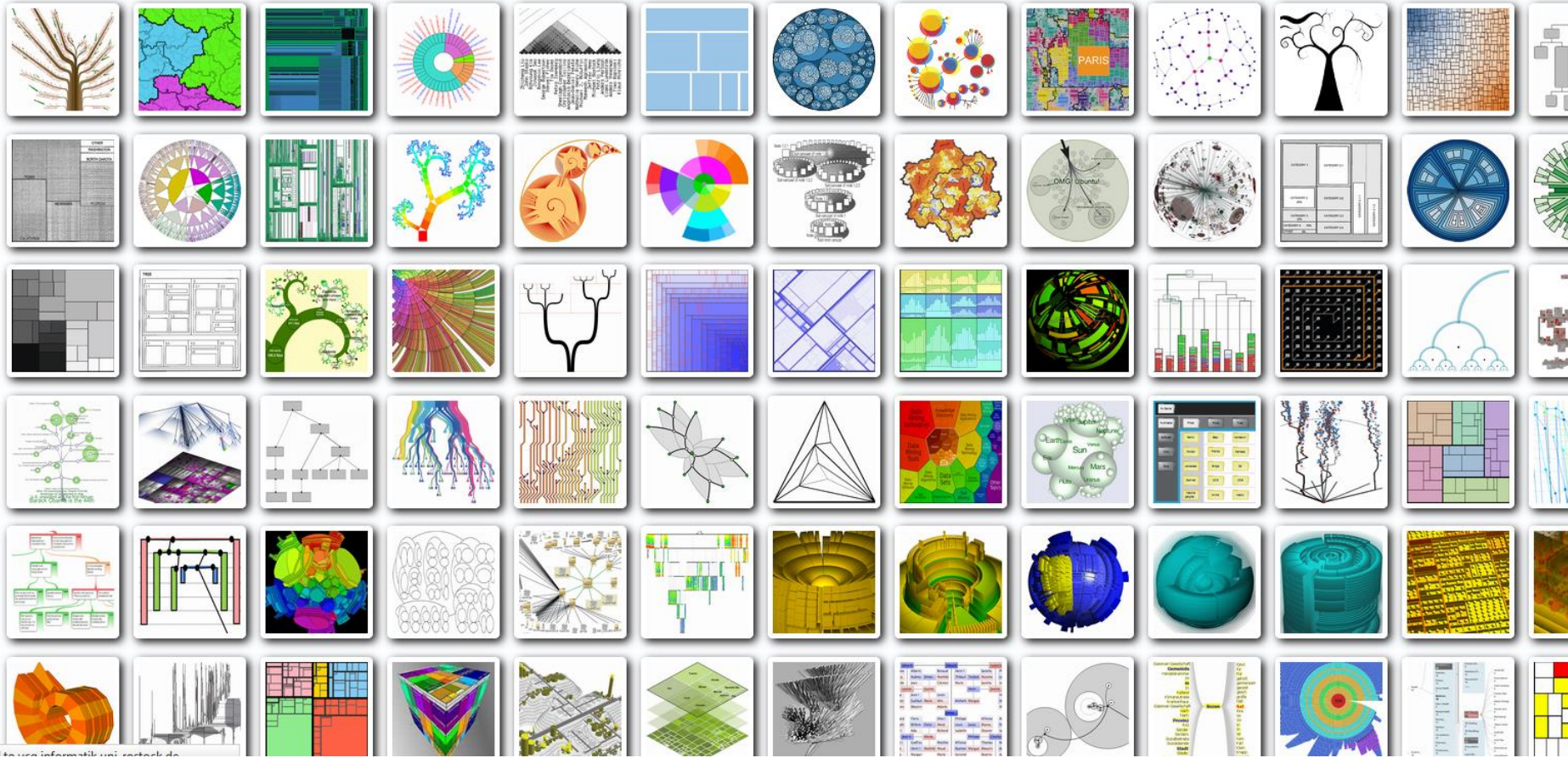
Dimensionality: All

Representation: All

Alignment: All

Fulltext Search

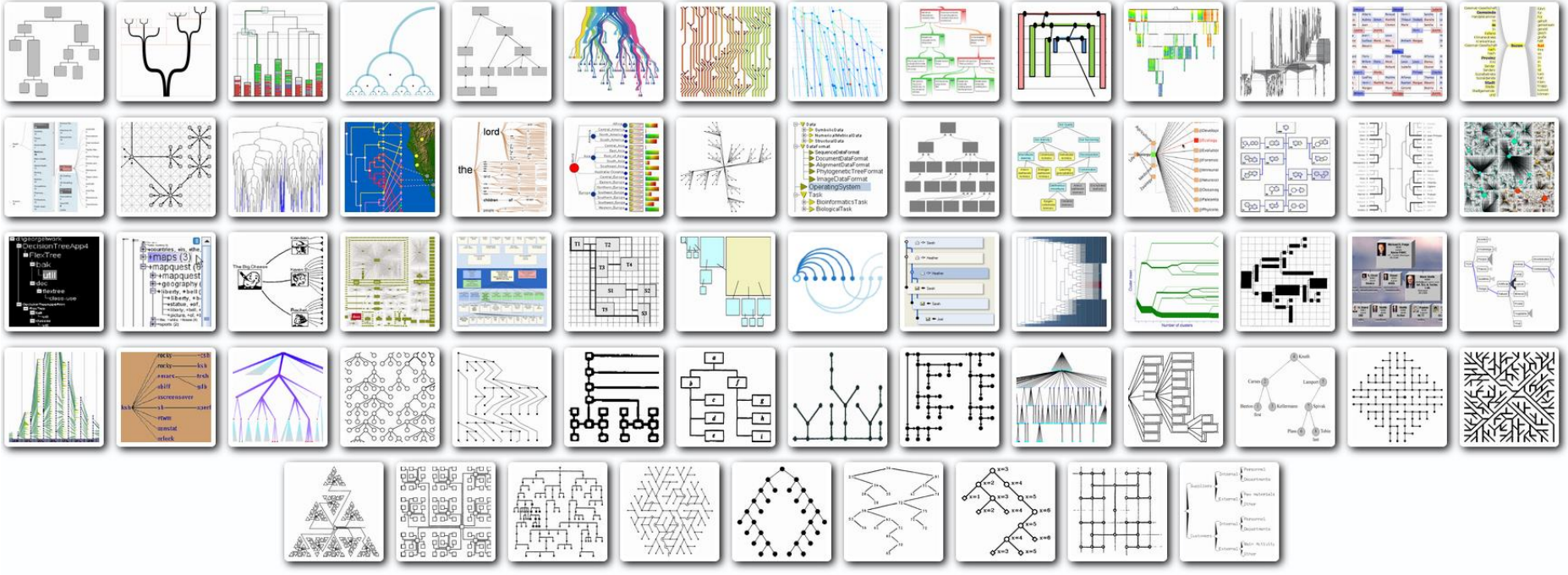
Techniques Shown: 292



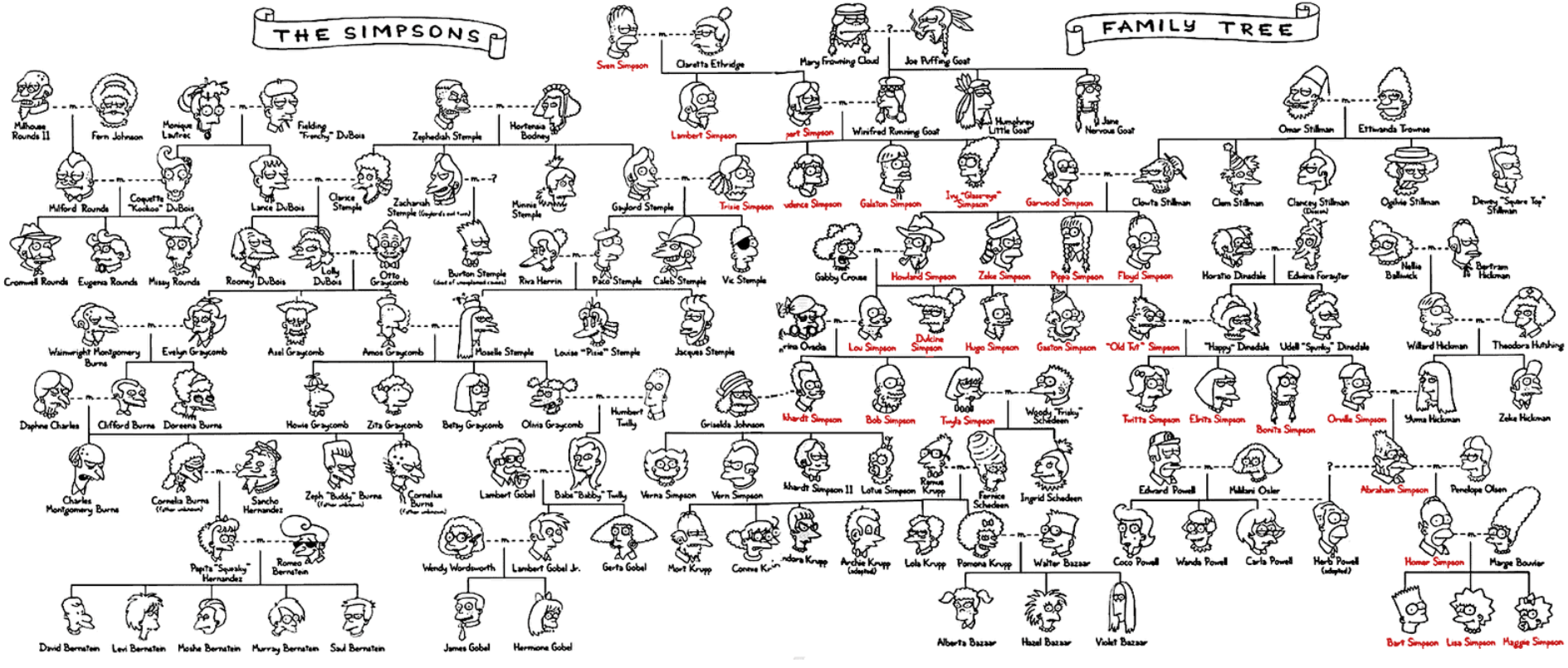
# CATEGORIZATIONS OF LAYOUTS

- many possible
- here we follow the categorization on treevis.net:
  - Dimensionality of the layout
  - Representation type
  - Alignment of nodes in space

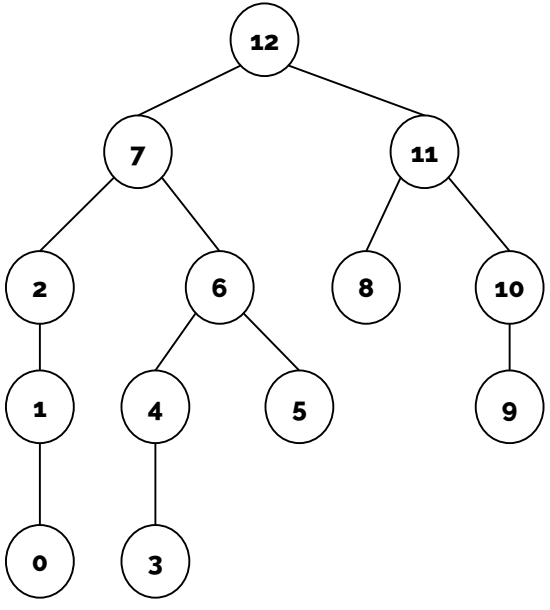
# 2D, AXIS-PARALLEL, EXPLICIT EDGES



# NODE-LINK



# NODE-LINK ALGORITHM

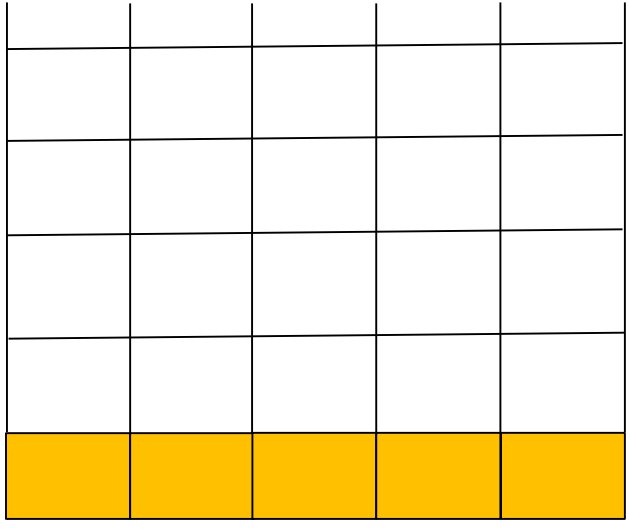


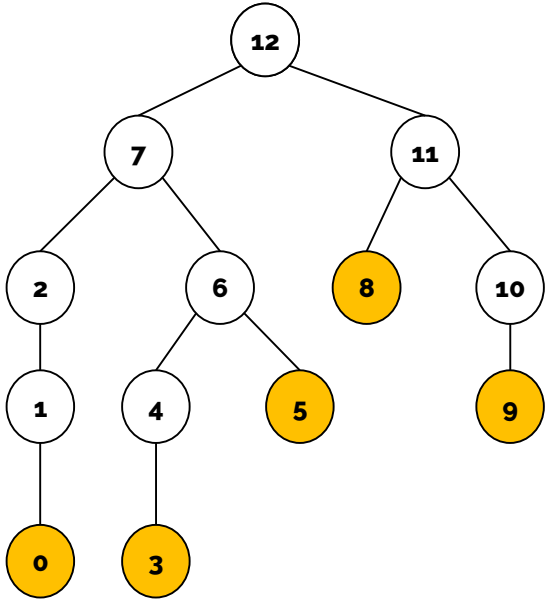

THE LAYOUT WE WANT – HOW DO WE GET THERE?

# NODE-LINK ALGORITHM

SIMPLE APPROACH

1) COUNT THE LEAVES
2) PLACE THE ROOT

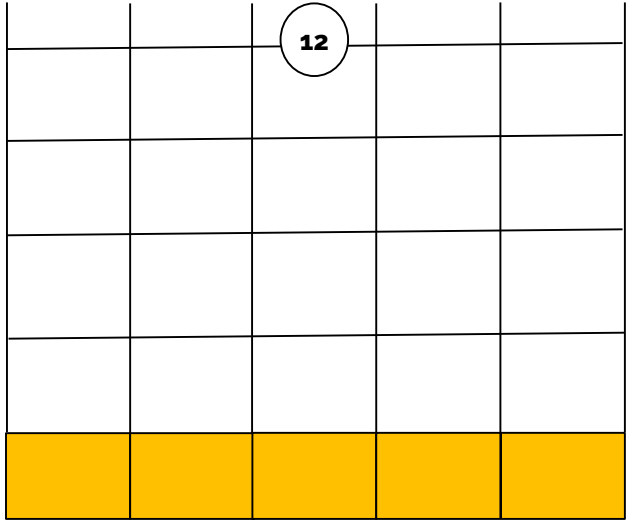


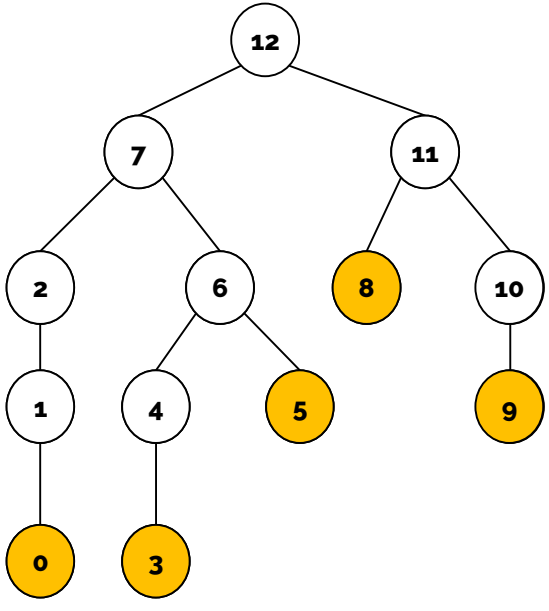


# NODE-LINK ALGORITHM

SIMPLE APPROACH

1) COUNT THE LEAVES
2) PLACE THE ROOT
3) RECURSIVELY DIVIDE

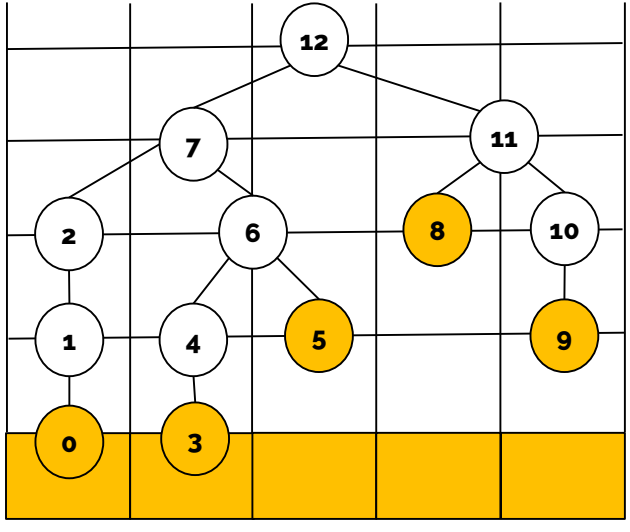


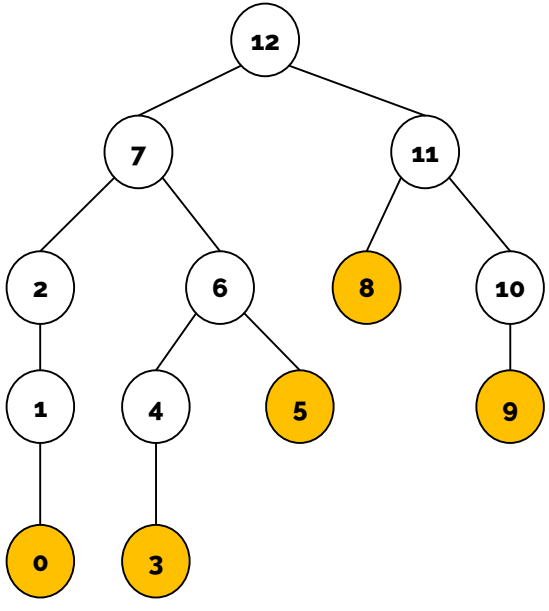



# NODE-LINK ALGORITHM

SIMPLE APPROACH

1) COUNT THE LEAVES
2) PLACE THE ROOT
3) RECURSIVELY DIVIDE

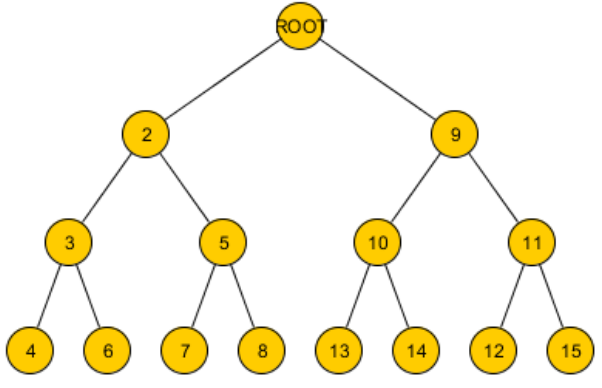


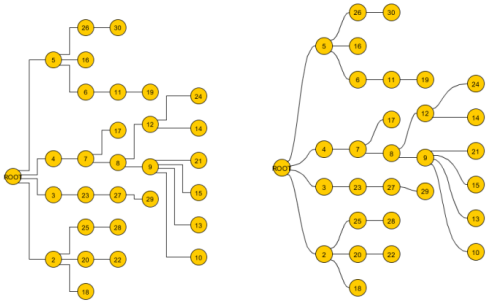
# PROS/CONS

- nodes at the same distance from the root are horizontally aligned
- positive: simple to understand, clear symmetries
- negative: needs large area, often bad aspect ratio (much wider than tall)

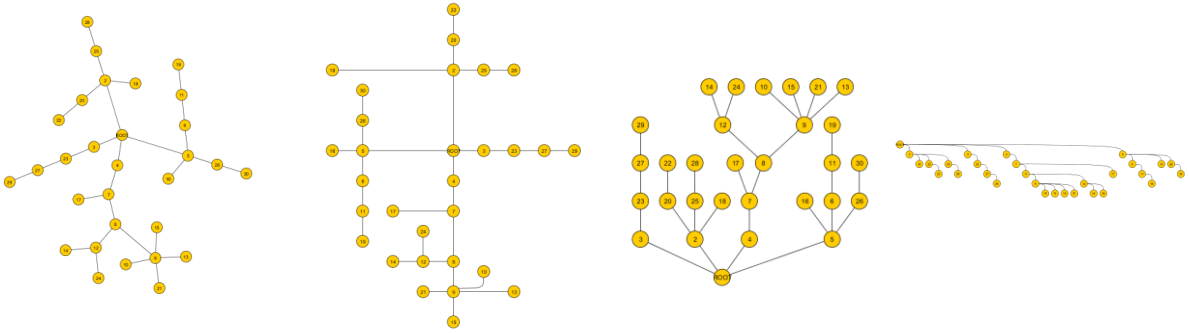
- what can we vary in this representation?
  - marks that depict nodes
  - visual variables used on marks to depict metadata
  - type of links
  - visual variables used on marks that depict the links
  - placement of nodes



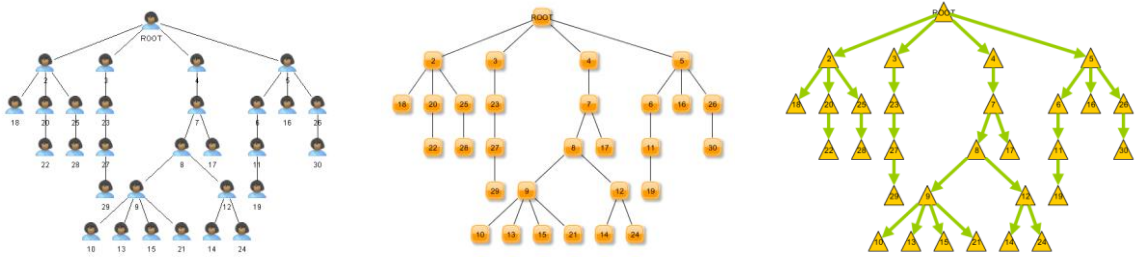
vary edges



node placement



marks



Images created with yEd: http://www.yworks.com

# SPECIFIC ALGORITHMS

- usually described recursively
- well known: Reingold-Tilford algorithm
- lots of research in this direction:
  - Wetherell and Shannon 1978, Tidy Drawings of Trees
  - **Reingold and Tilford 1981**, Tidier Drawing of Trees
  - Walker 1990, A Node-positioning Algorithm for General Trees
  - Buchheim et al 2002, Improving Walker's Algorithm to Run in Linear Time

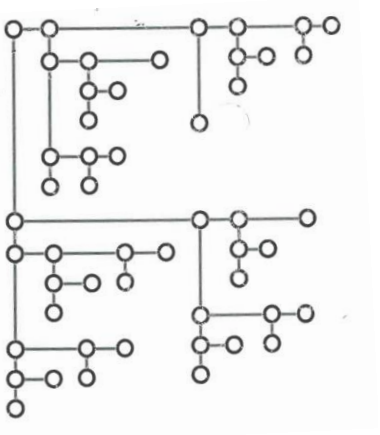
# AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being layed out

- **area**: match area of your layout to the size of the display and data
- **aspect ratio**: usually optimal if close to 1
- **subtree separation**: try not to overlap subtrees
- **root-leaf distance**: minimize distance from root to leaves
- **edge lengths**: minimize total, average, maximum, edge lengths & try to make edge lengths uniform
- **angular resolution**: increase angles formed by edges
- **symmetry**: symmetric layouts usually considered pleasing

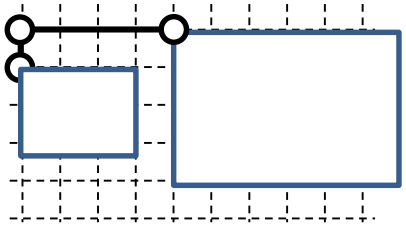
# HV DRAWING

- for binary trees
- straight line grid drawing
  - every child of a vertex u, is either horizontally aligned with and to the right of u, or vertically aligned with and below u
  - the bounding rectangles (smallest rectangles with horizontal and vertical sides covering the drawings) of the subtrees of u do not intersect

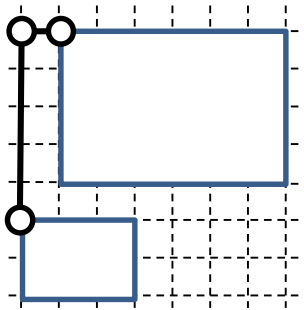


# HV DRAWING STRATEGY

- divide: recursively construct hv-drawings for left and right subtrees
- conquer: perform either a horizontal combination or a vertical combination

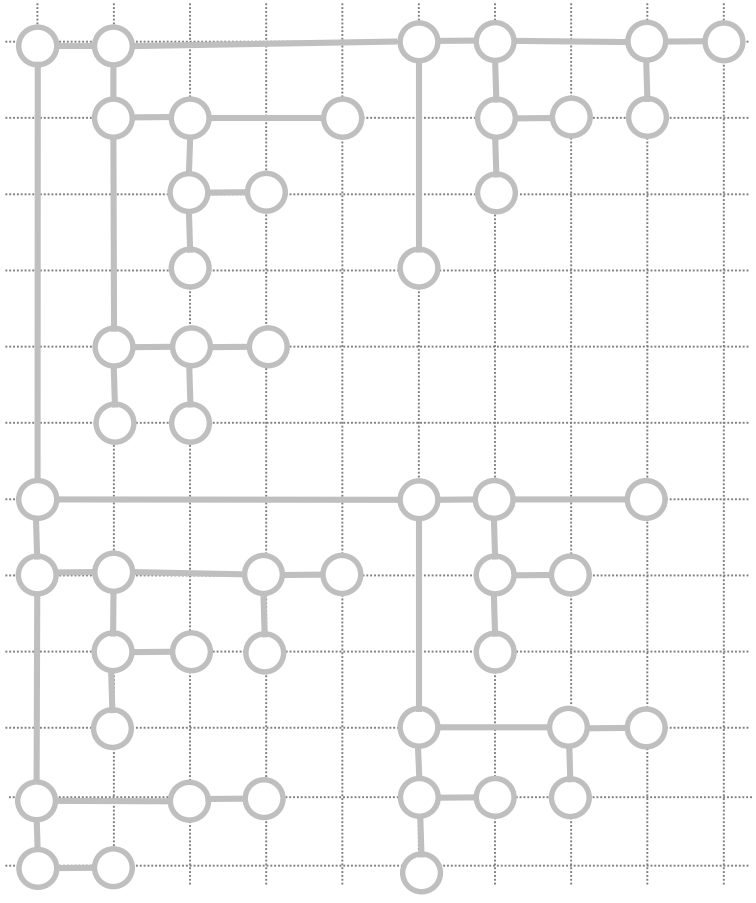


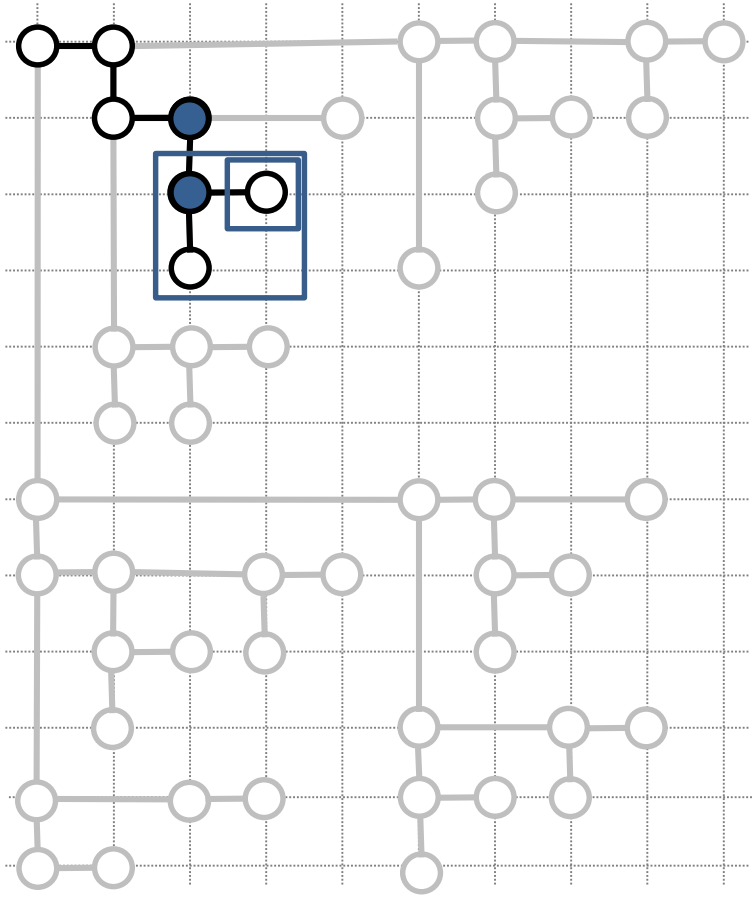
horizontal combination

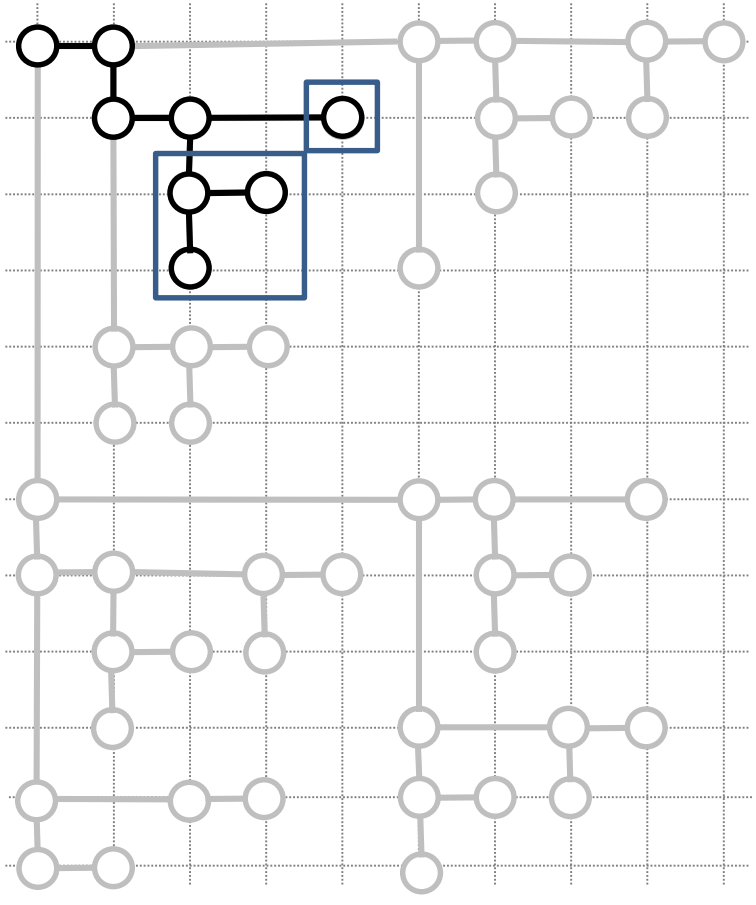


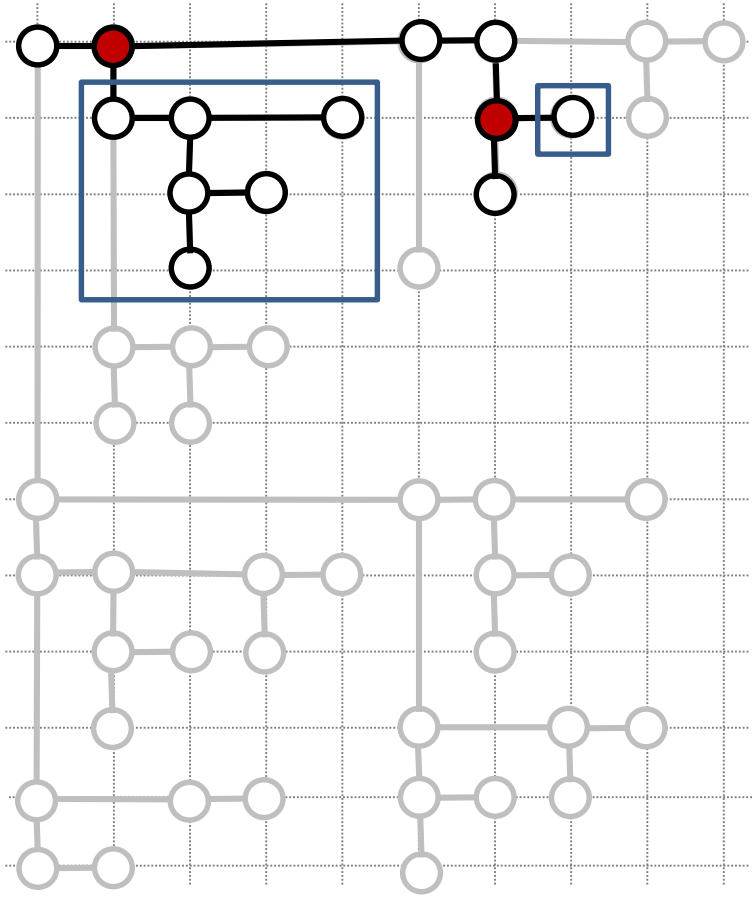
vertical combination

Resource: Battista et al. Graph Drawing





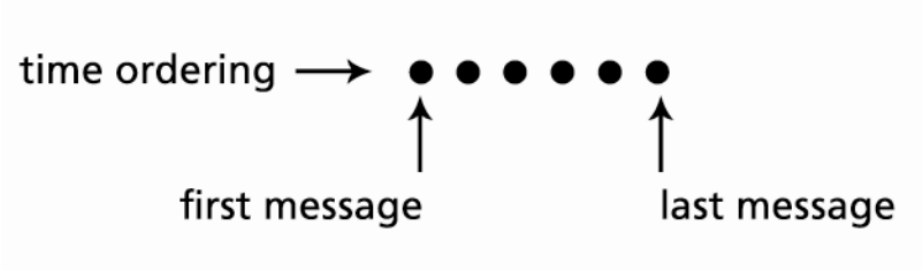


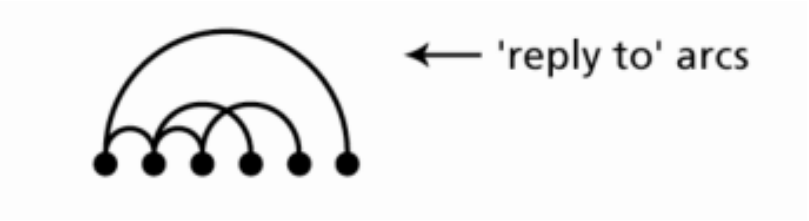


As an exercise,
fill in the rest by yourself

# LAYOUT DIMENSIONALITY: 2D – THREAD ARCS

- email visualization



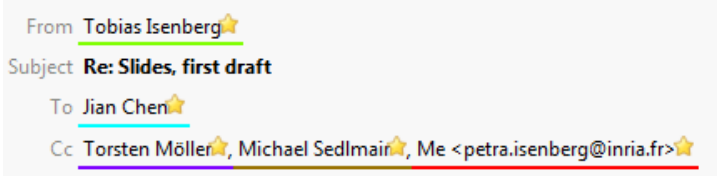



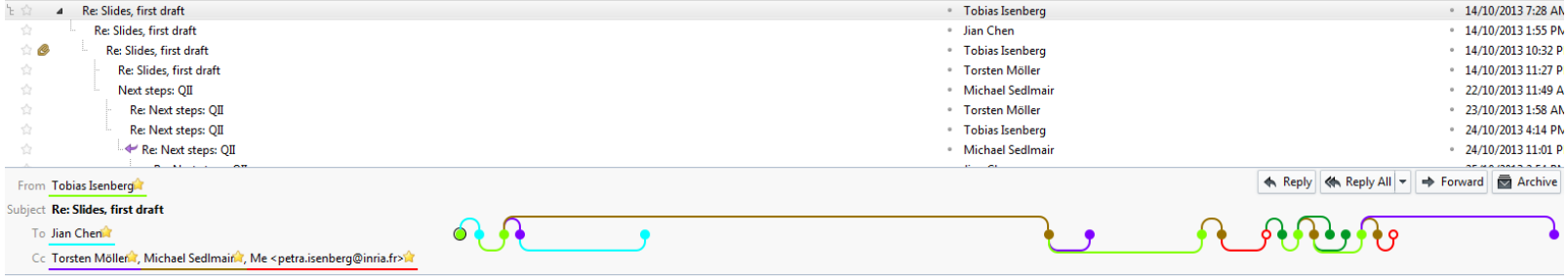

Thread Arcs: An email thread visualization
InfoVis'03: Proceedings of the IEEE Symposium on Information Visualization, pages 211-218.

# THREADVIS

http://threadvis.github.io/

- time-scaling
- coloring people



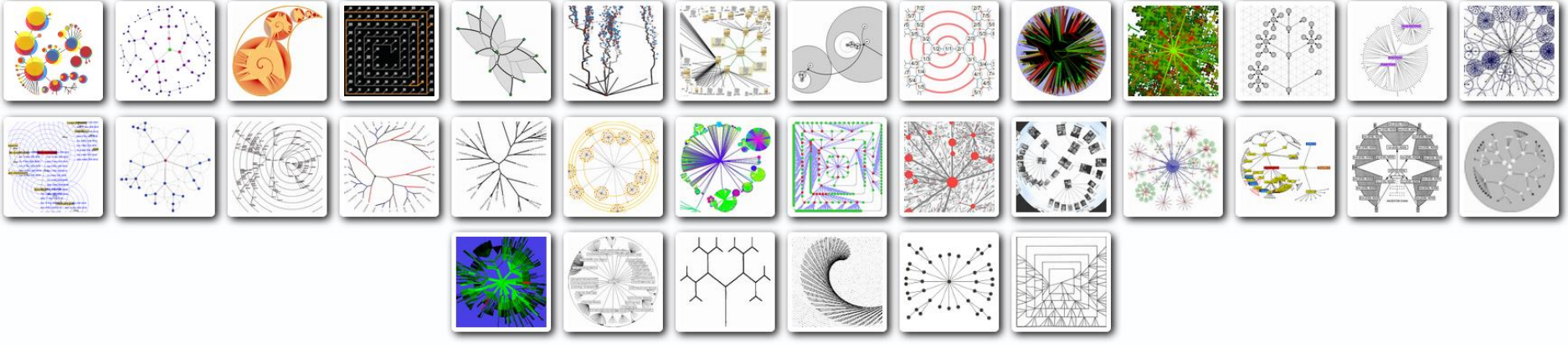



# TREEJUXTAPOSER

Rectilinear layout and interaction for comparison of very large trees

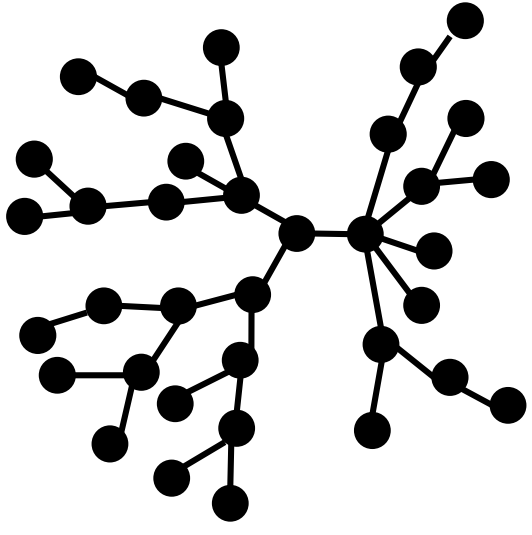

http://www.cs.ubc.ca/~tmm/papers/tj/#video

# 2D, RADIAL, EXPLICIT EDGES



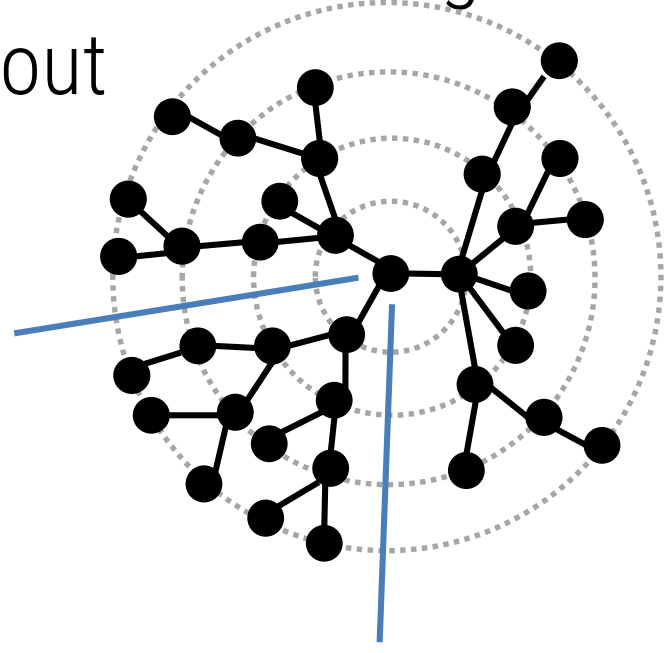
# RADIAL NODE-LINK DRAWING

variation of layered drawing from beginning of lecture



# RADIAL NODE-LINK DRAWING

- nodes drawn on concentric circles
- nodes drawn within wedges of the circular layout

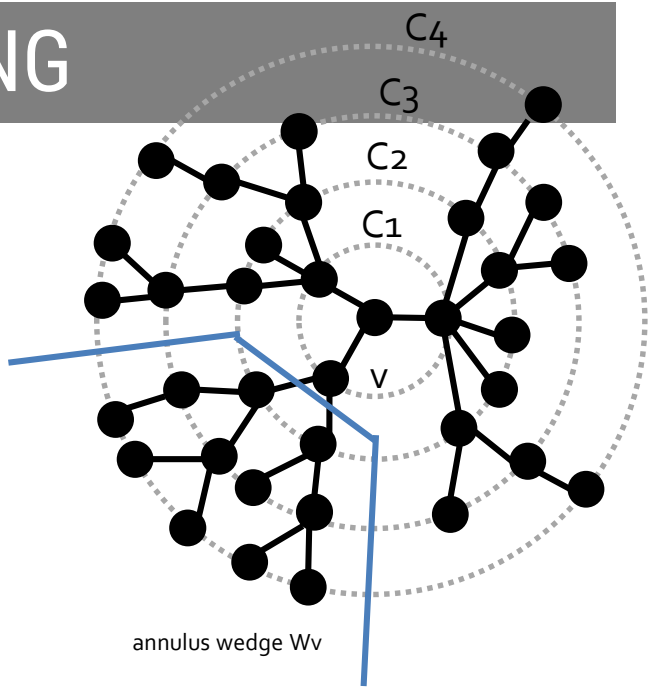


# RADIAL NODE-LINK DRAWING

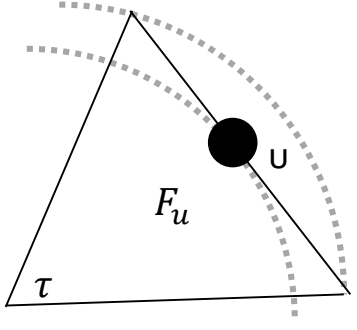
- radius of $C_i$ given by function p(i)
- subtree of v drawn within $W_v$
- to guarantee planarity (no edge crossings), wedge has to be convex
- several algorithms exist for figuring out the correct angles, e.g.

$$\beta_u = \left(\frac{l(u)\beta_v}{l(v)}, \tau\right)$$

For each child u of v:
- $\beta_u$ is the angle of $W_u$
- $\tau$ is the angle formed by region $F_u$
- l(v): number of leaves in subtree rooted at v
- place u at center of $W_u$



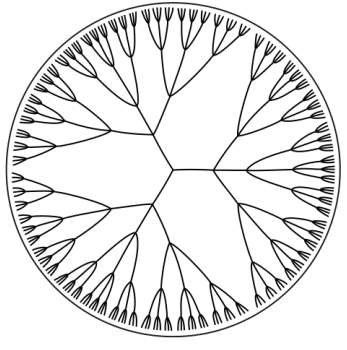
annulus wedge Wv



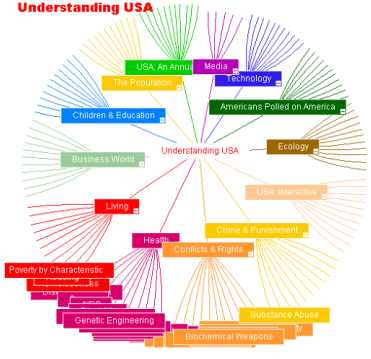

Resource: Battista et al. Graph Drawing

# HYPERBOLIC BROWSER

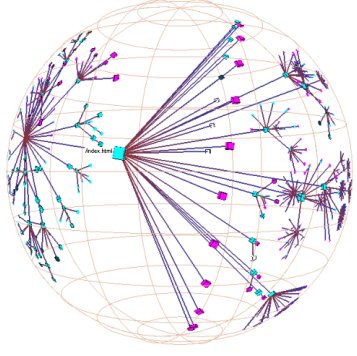
- uses hyperbolic geometry (not euclidean geometry)
- a hyperbolic plane can be displayed using the Poincaré disk model
  - a tree structure of any size fits within a finite area (circle)
  - node is displayed in center
  - all oder nodes move away from center and become exponentially



(a) Uniform hyperbolic tree.

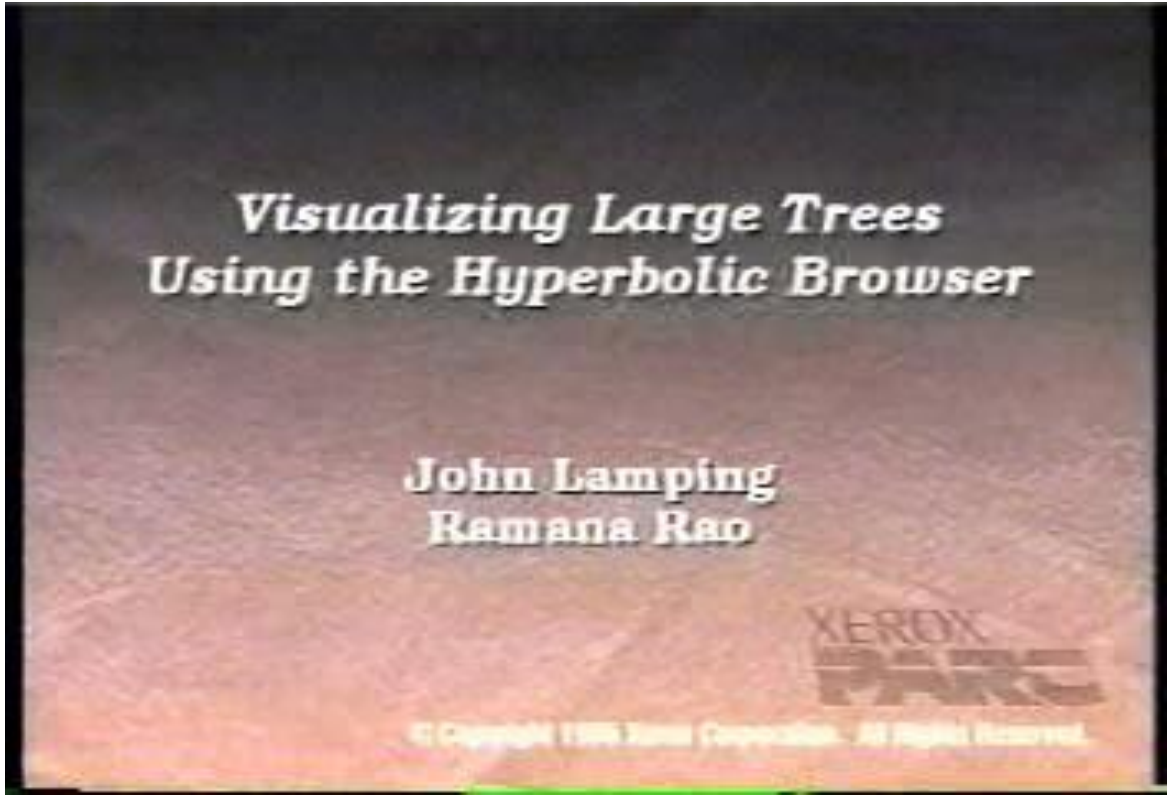



(b) StarTree by Inxight Software.

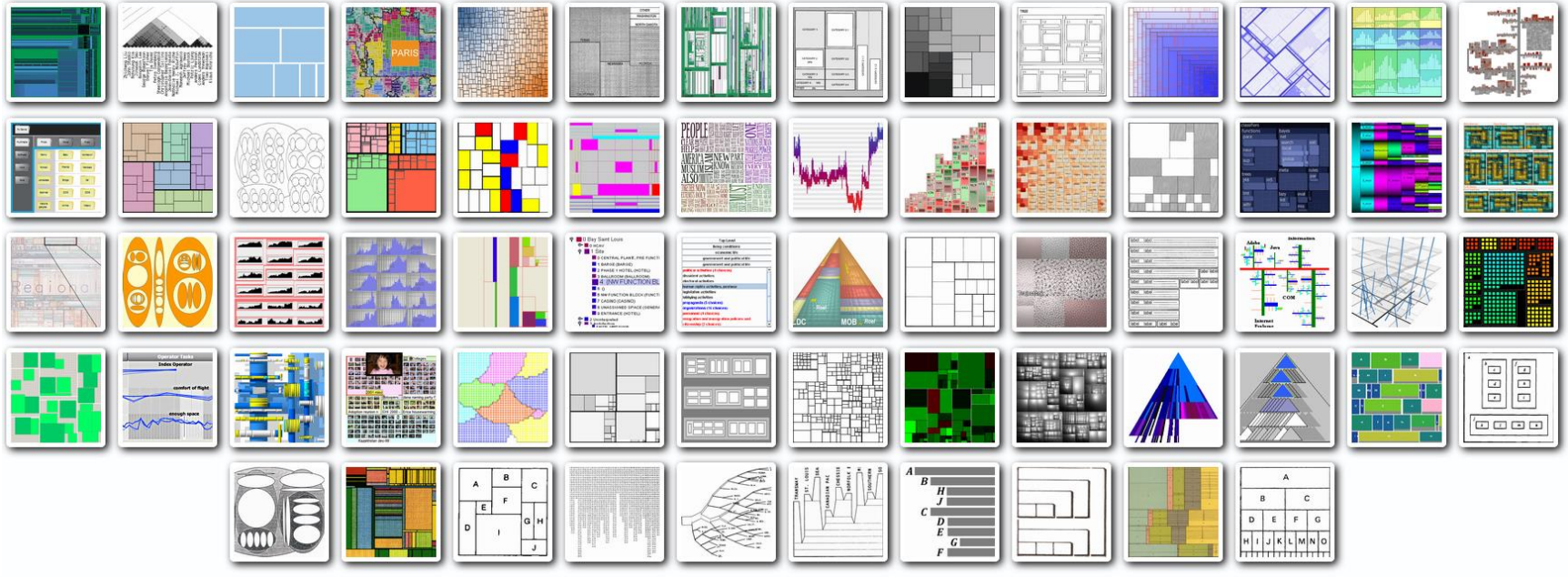


(c) H3 Browser.

# CHI 1996 VIDEO OF HYPERBOLIC BROWSER

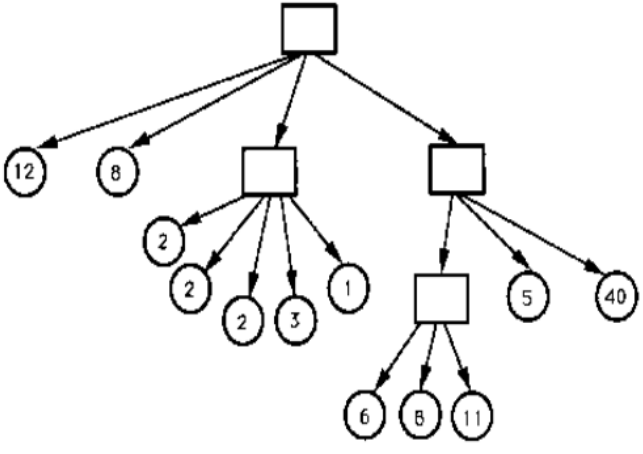


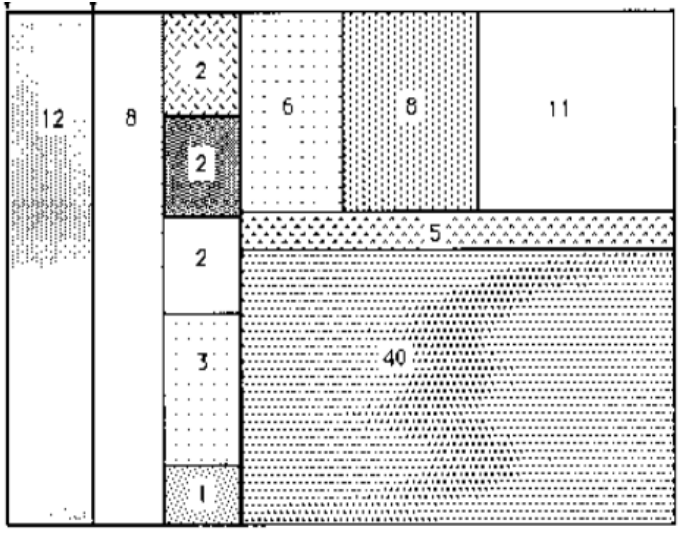
# 2D, AXIS-PARALLEL, IMPLICIT EDGES



# A CLASSIC CONTAINMENT LAYOUT

- example tree to rebuild with treemap algorithm
- size of each node as numbers in leaves



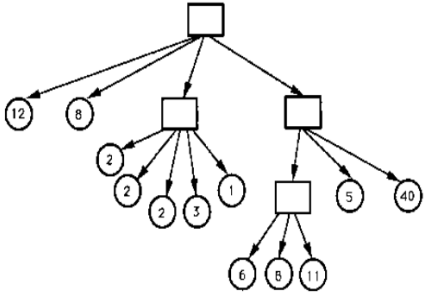
# TREEMAP ALGORITHM

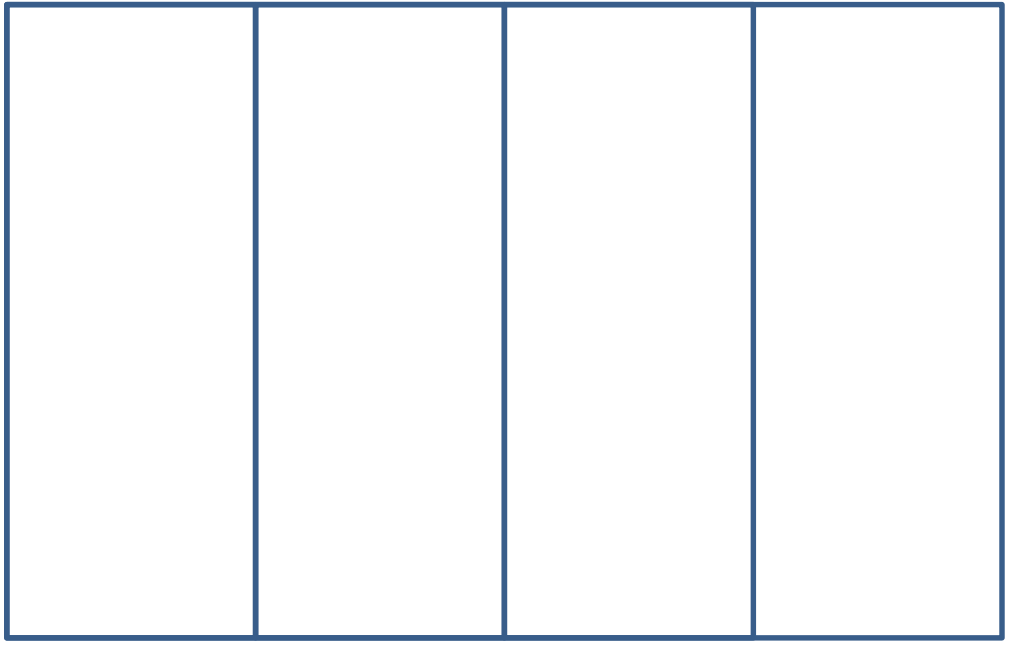
- Take a rectangular display area P1(x1,y1), Q1(x2,y1)
- This area represents the root of the tree

# TREEMAP ALGORITHM

recursive algorithm

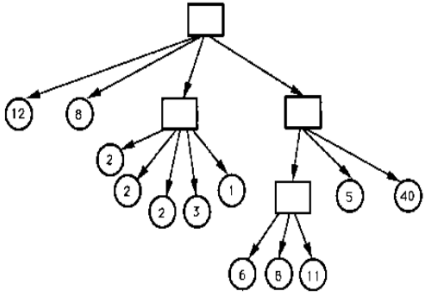
- the number of children of the current node define the number of partitions of the current node

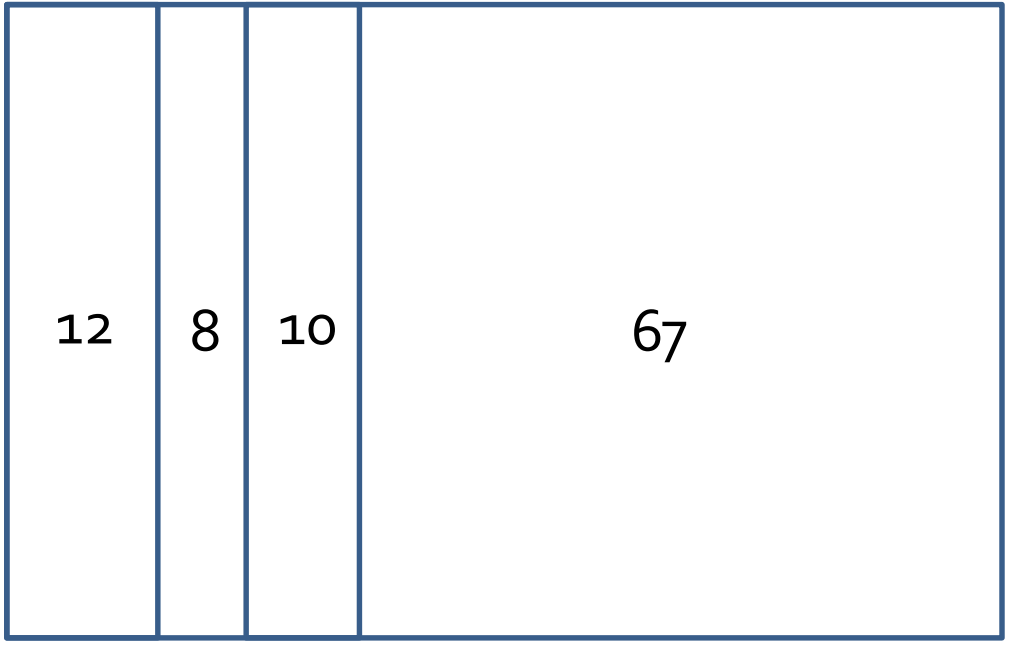


# TREEMAP ALGORITHM

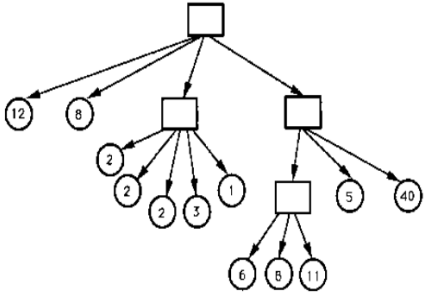
the weight of each node determines the size of each partition

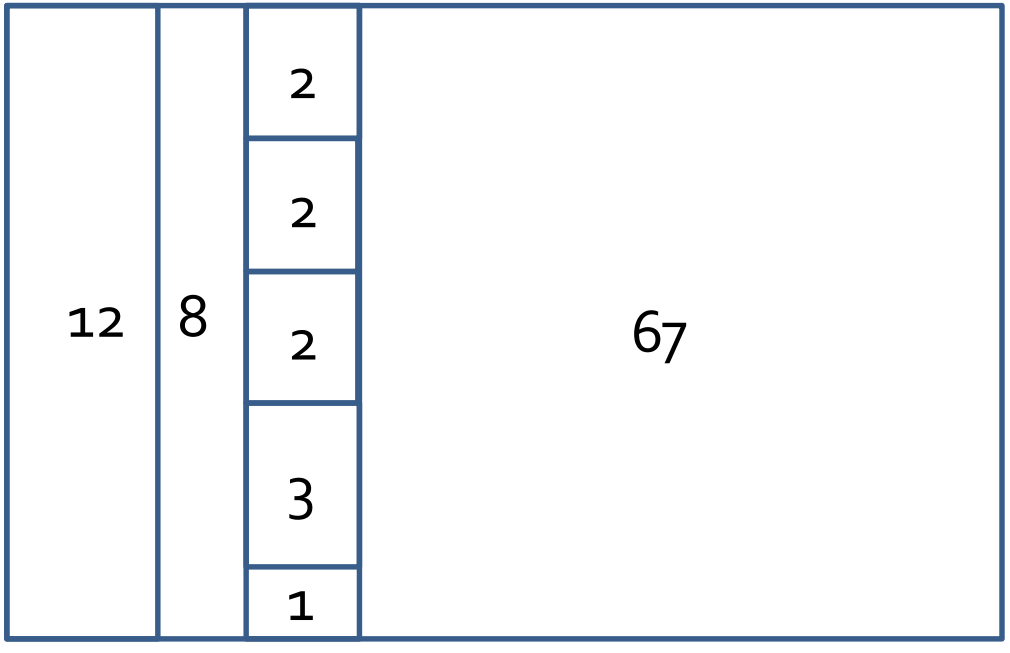


# TREEMAP ALGORITHM

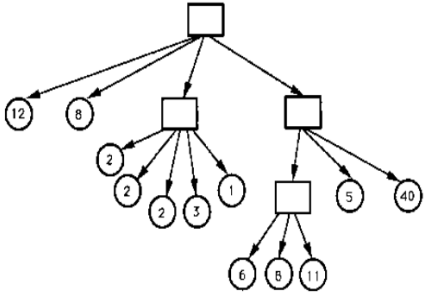
at each change of level, rotate orientation of split by 90 degrees

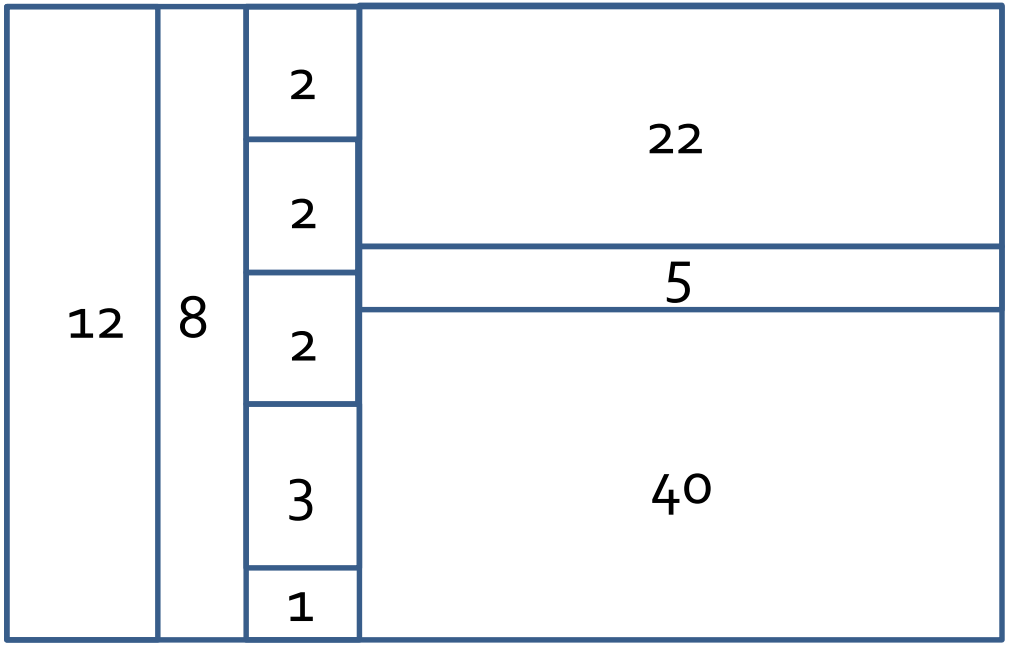


# TREEMAP ALGORITHM

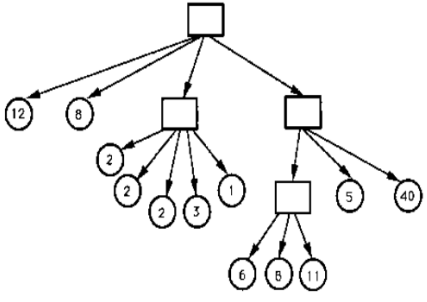
at each change of level, rotate orientation of split by 90 degrees

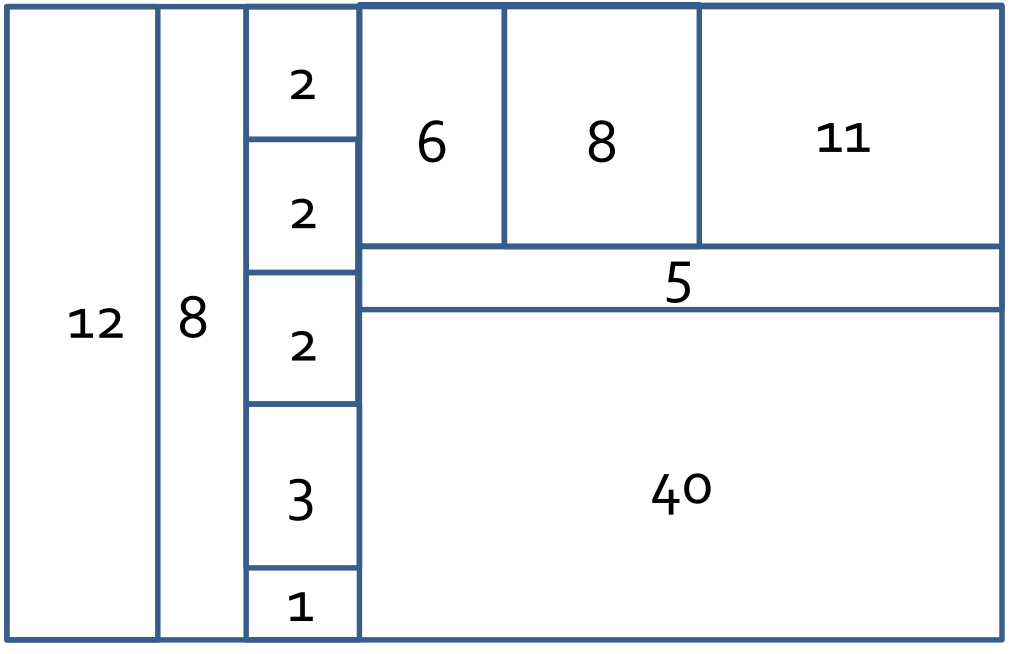


# TREEMAP ALGORITHM

at each change of level, rotate orientation of split by 90 degrees



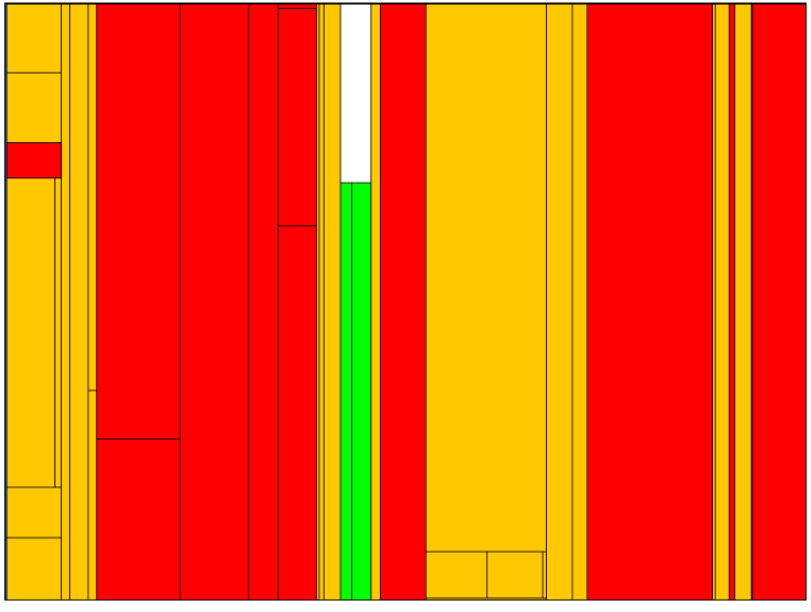



# TREEMAP

- a 2-D space-filling layout
- for further references and to try out a treemap in various applications: http://www.cs.umd.edu/hcil/treemap-history/

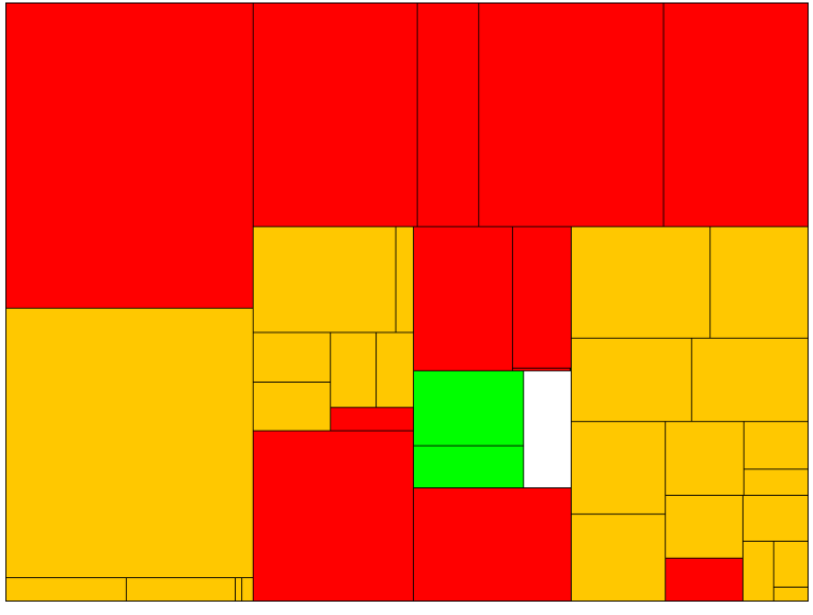
# TREEMAP VARIATIONS

- problem with original treemap: lots of long stripes
- for long stripes the areas are difficult to compare



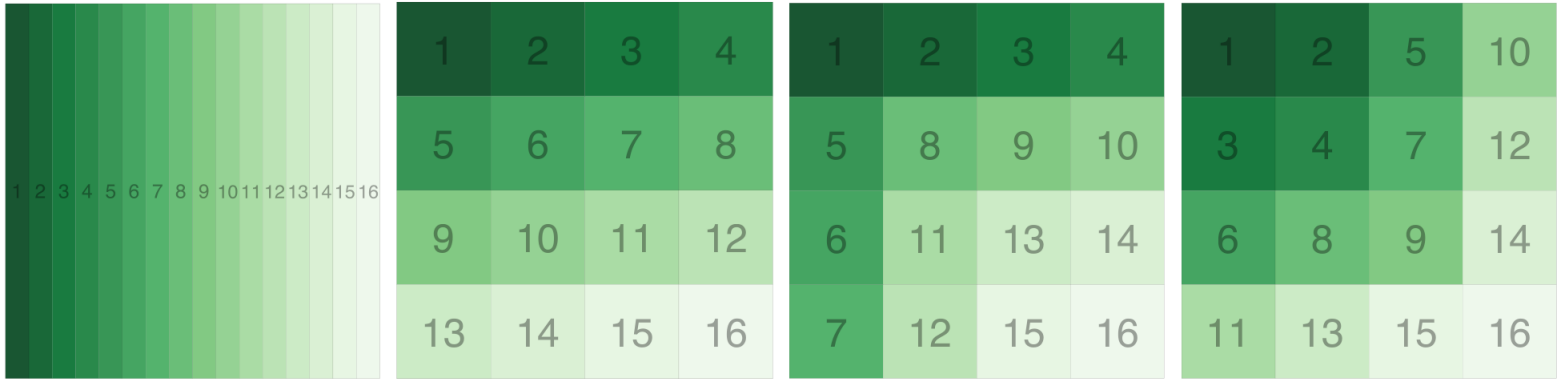
# SQUARIFIED TREEMAP

- calculates more squared regions
- problem: order not as easily read, not very stable with dynamically changing data



# ORDERED TREEMAP

## several algorithms in comparison




slice and dice

B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. ACM Transactions on Graphics, 11:92–99, 1992.

strip

B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. ACM Transactions on Graphics, 21:833–854, 2002.

squarified

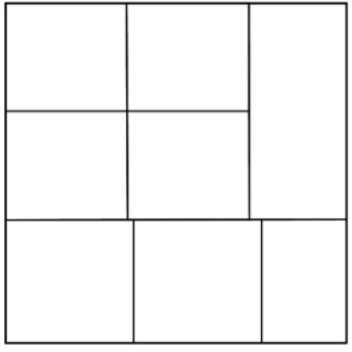
M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. EuroVis, pages 33–42, 2000.

ordered squarified

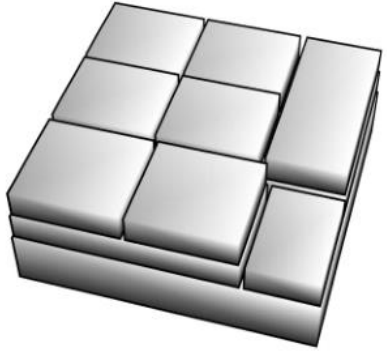
B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In Infovis01, pages 73–78, 2001.

Readability scores: 1.0, 0.625, 0.375, 0.125 (1 = no angular change, 0=every jump between sequential nodes requires an abrupt angular change

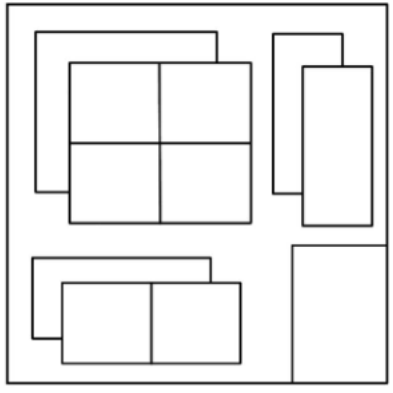
# OTHER



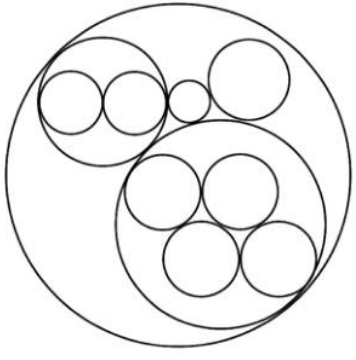
original squarified:
emphasizes leafs and their attributes



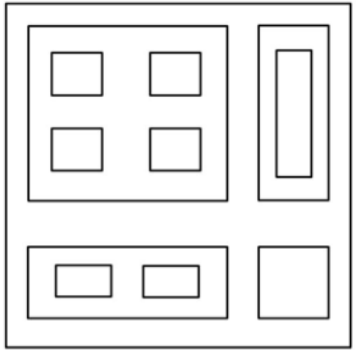
steptree:
emphasizes structure with extrusion



cascaded layout:
emphasizes structure with overlap

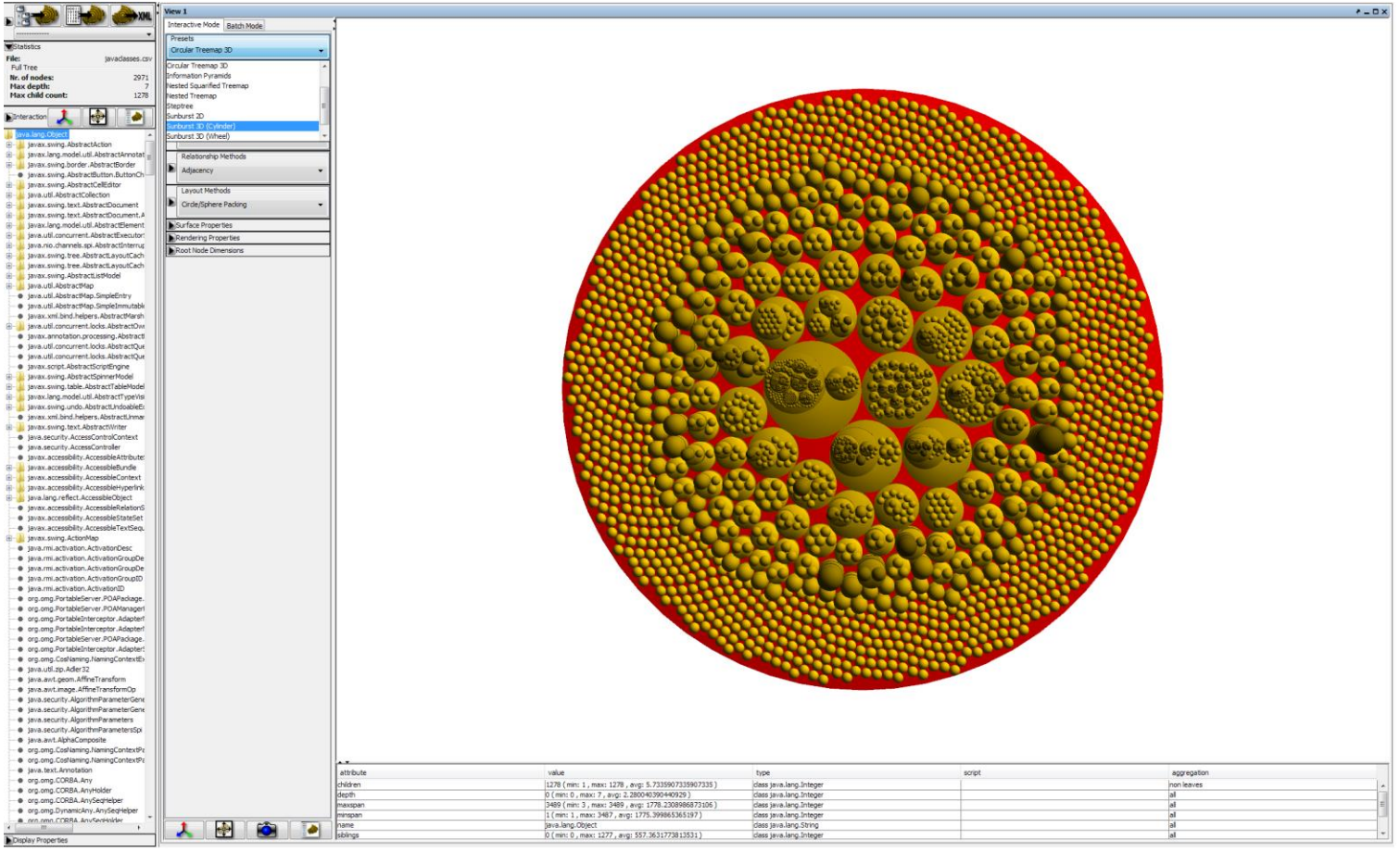


circular treemap:
emphasizes structure with
non-space-filling primitive

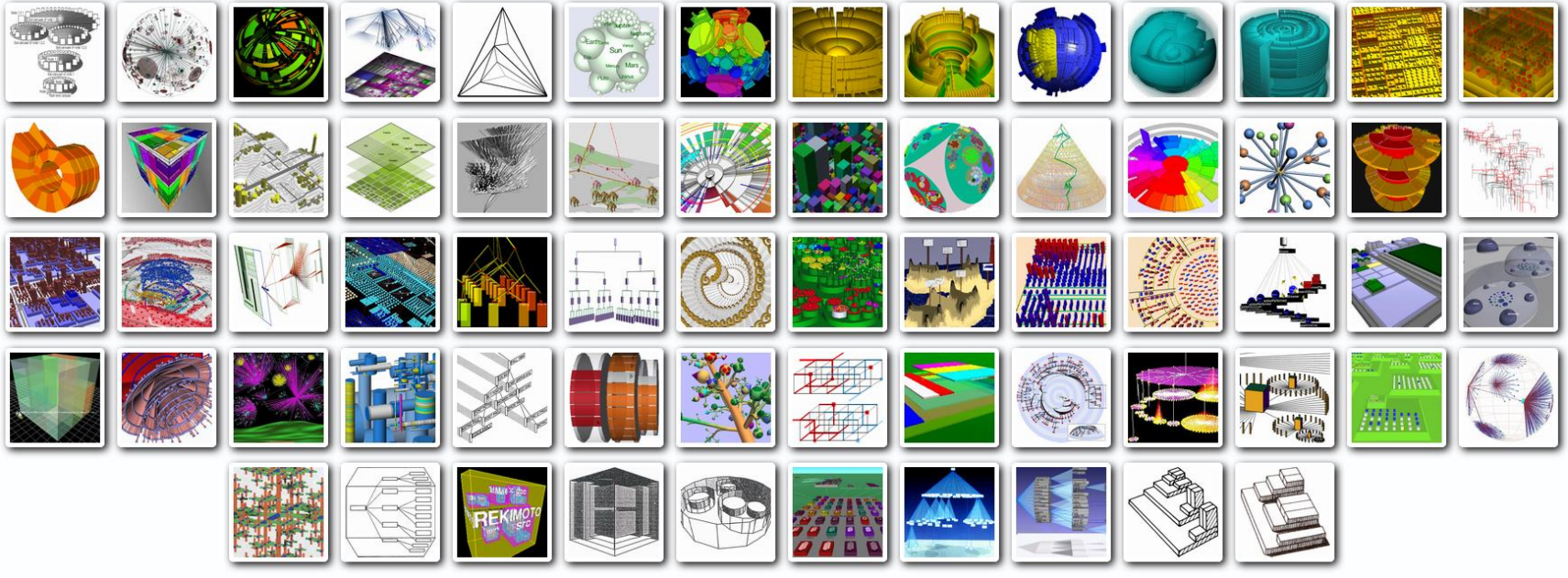


nested layout:
emphasizes structure with whitespace

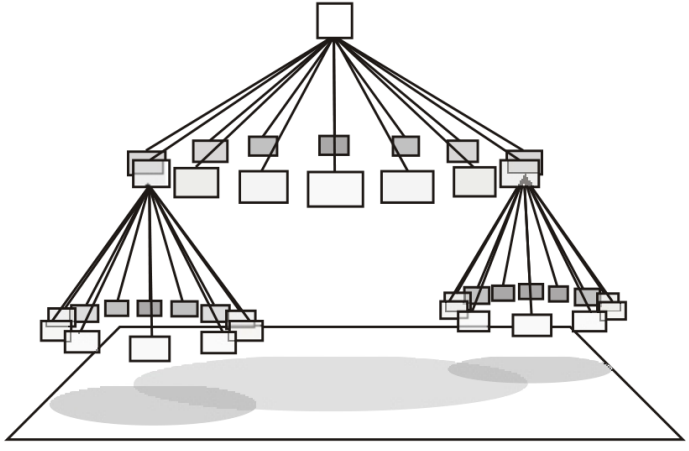
# TRY THE IMPLICIT TREE VISUALIZATION TOOLKIT: HTTP://VCG.INFORMATIK.UNI-ROSTOCK.DE/~HS162/ITVTK/START.HTML



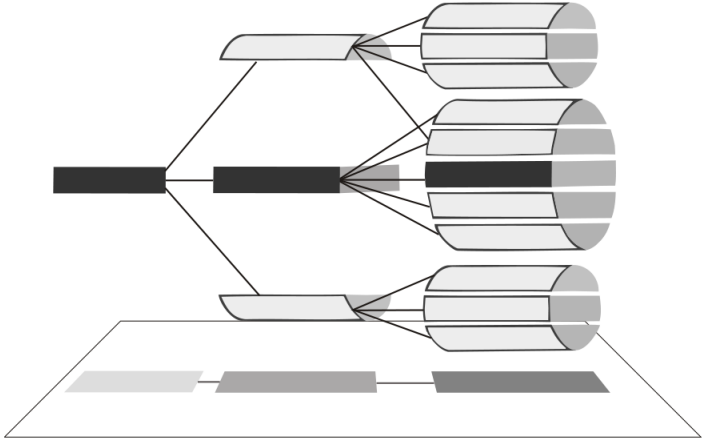
# 3D LAYOUTS



# HISTORIC EXAMPLE: CONETREE / CAMTREE



ConeTree

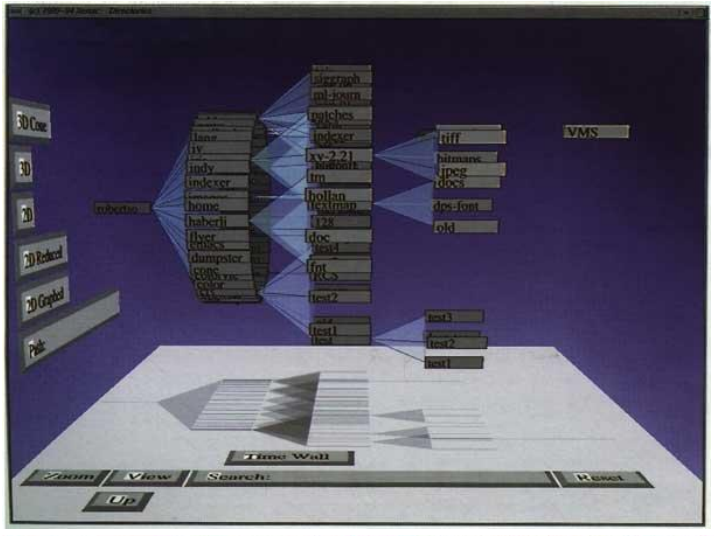


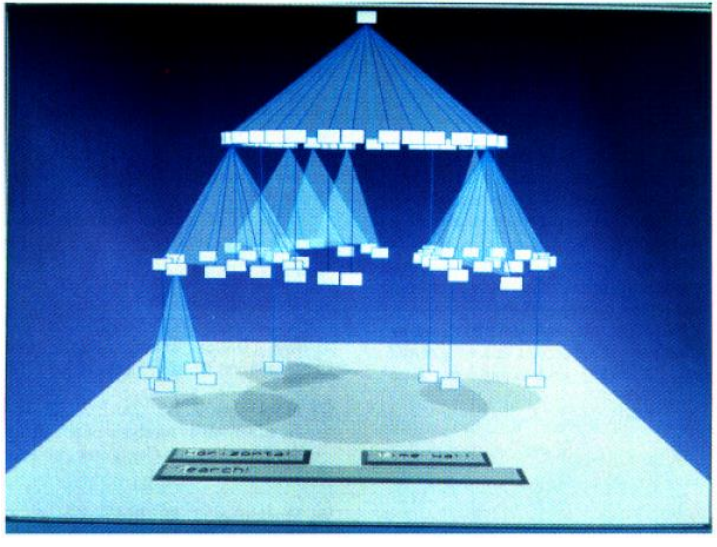
CamTree

[Robertson et al., 1991] George Robertson, Jock D. Mackinlay, Stuart Card. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of the ACM CHI 91*

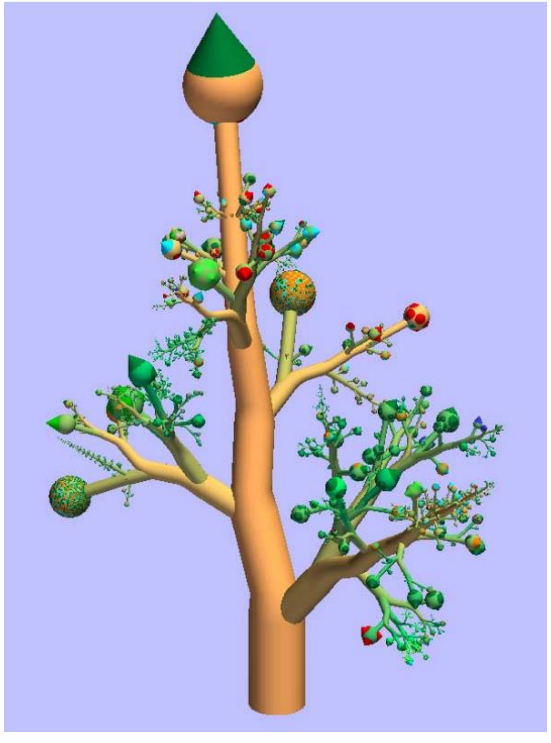
# CONE/CAMTREE

- children of a node are laid out in a cylinder “below” the parent
- siblings located on the same 2D circle
- use of animation
- shadows to enhance structure



# BOTANICAL VISUALIZATION OF HUGE HIERARCHIES



InfoVis 2001

# 3D LAYOUTS

- advantages
  - fit more data into same aspect ratio
  - aesthetically pleasing (aka can look very sexy)
- disadvantages
  - occlusion
  - requires interaction or animation
  - no overviews

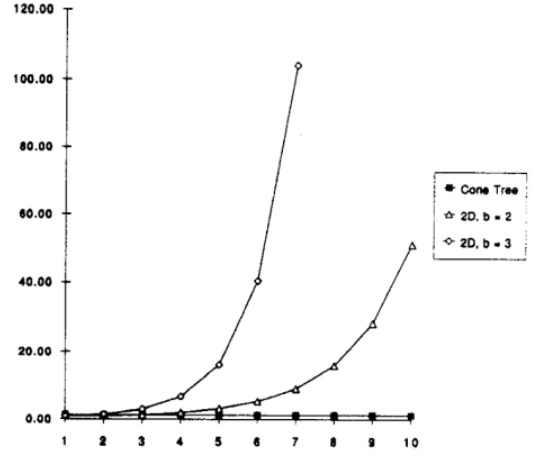



Figure 1: Aspect Ratio of 2D and 3D Trees.

# TREE VISUALIZATION SUMMARY

- there are lots of tree visualizations
  - there is also lots of free software, try it out, (see links earlier in the lecture plus e.g.):
    - http://www.informatik.uni-rostock.de/~hs162/optreedemo/TestBed.htm#
    - http://w3.win.tue.nl/nl/onderzoek/onderzoek_informatica/visualization/sequoiaview/
  - there are a few overview articles, e.g.:
    - **A Generative Layout Approach for Rooted Tree Drawings** by Hans-Jörg Schulz, Zabed Akbar, and Frank Maurer - at IEEE PacificVis 2013
    - **The Design Space of Implicit Hierarchy Visualization: A Survey** by Hans-Jörg Schulz, Steffen Hadlak, and Heidrun Schumann - in IEEE TVCG 17(4)

# TREE VISUALIZATIONS

- can be categorized by
  - edge representations (implicit, explicit)
  - dimensionality of layout
  - radial vs. axis-parallel

- can be modified by
  - layout parameters
  - which marks are used
  - visual variables on marks (which meta-data is represented?)

# GRAPHS / NETWORKS

# DEFINITION GRAPH

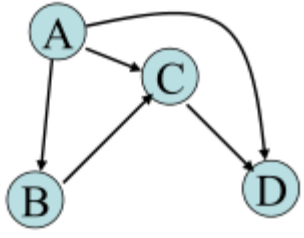
- A set of **vertices** V = $\{v_i\}$
- A set of **edges** E = $\{e_{ij}\}$ with $e_{ij} = \{v_i, v_j\}$
- When the order of vertices of an edge is meaningful, the graph is **directed**

# GRAPH MEASURES

- SIZE = #nodes
- DENSITY = edges/vertices (roughly)
- PATH = sequence of edges connecting (different) vertices
- VERTEX DEGREE = #edge connections
- DISTANCE = #hops between vertices

# TWO CLASSICAL VISUAL REPRESENTATIONS

Node-Link Diagram

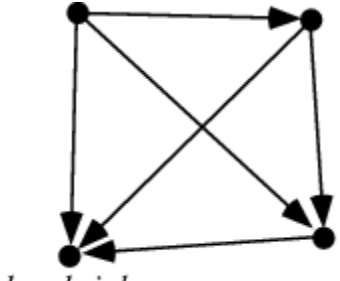



Adjacency Matrix

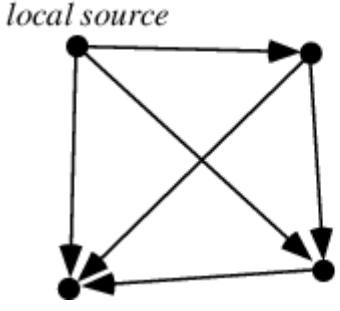
| ↗ | A | B | C | D |
|---|---|---|---|---|
| A | | X | X | X |
| B | | | X | |
| C | | | | X |
| D | | | | |

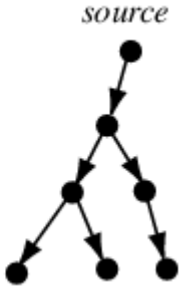
# TASKS

Find # of neighbors of a vertex (e.g. source vs. sink)







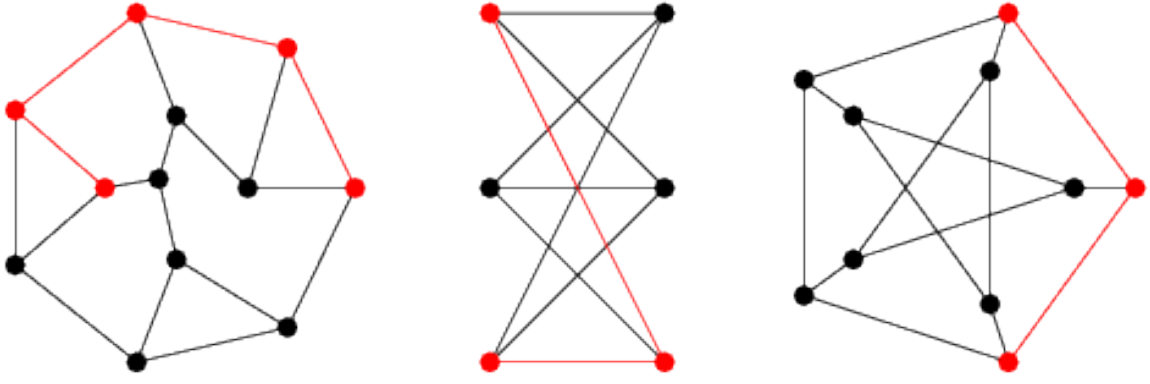




http://mathworld.wolfram.com/DigraphSink.html

http://mathworld.wolfram.com/Source.html

# TASKS

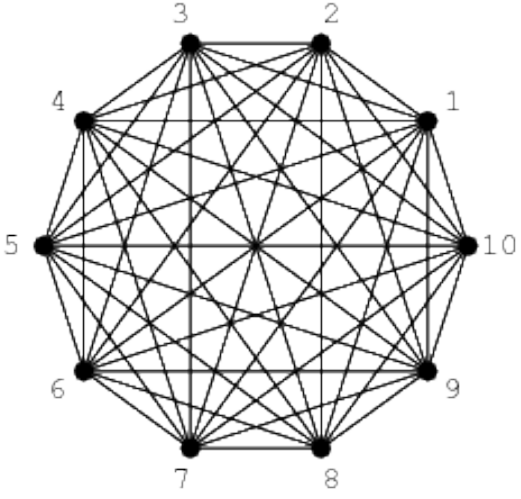
See paths (overviews, shortest,  cycles)

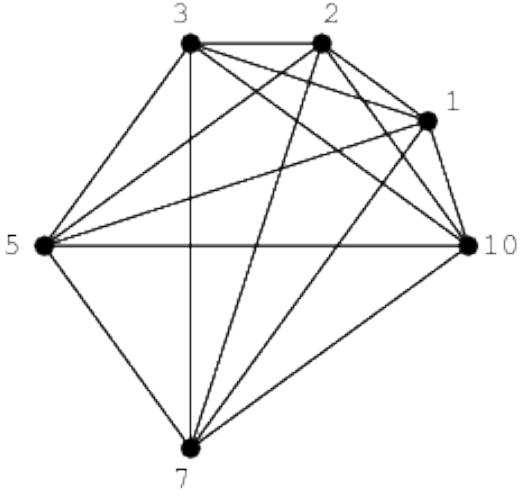


http://mathworld.wolfram.com/GraphGeodesic.html

# TASKS

## Identify Sub-graphs




http://mathworld.wolfram.com/Vertex-InducedSubgraph.html

# TASKS

## HIGHER-LEVEL

involves many elements
involves more human judgment

- which nodes are important?
- where are clusters?
- what are attribute and connection correlations?
- how does the network change over time?

# TASKS

- Many many more specific tasks
- Each application domain will add more

## Task Taxonomy for Graph Visualization

Bongshin Lee, Catherine Plaisant,
Cynthia Sims Parr
Human-Computer Interaction Lab
University of Maryland,
College Park, MD 20742, USA
+1-301-405-7445
{bongshin, plaisant, csparr}@cs.umd.edu

Jean-Daniel Fekete,
Nathalie Henry
INRIA Futurs/LRI Bat. 490
Université Paris-Sud,
91405 ORSAY, France
+33-1-69153460
Jean-Daniel.Fekete@inria.fr, nhenry@lri.fr

**ABSTRACT**

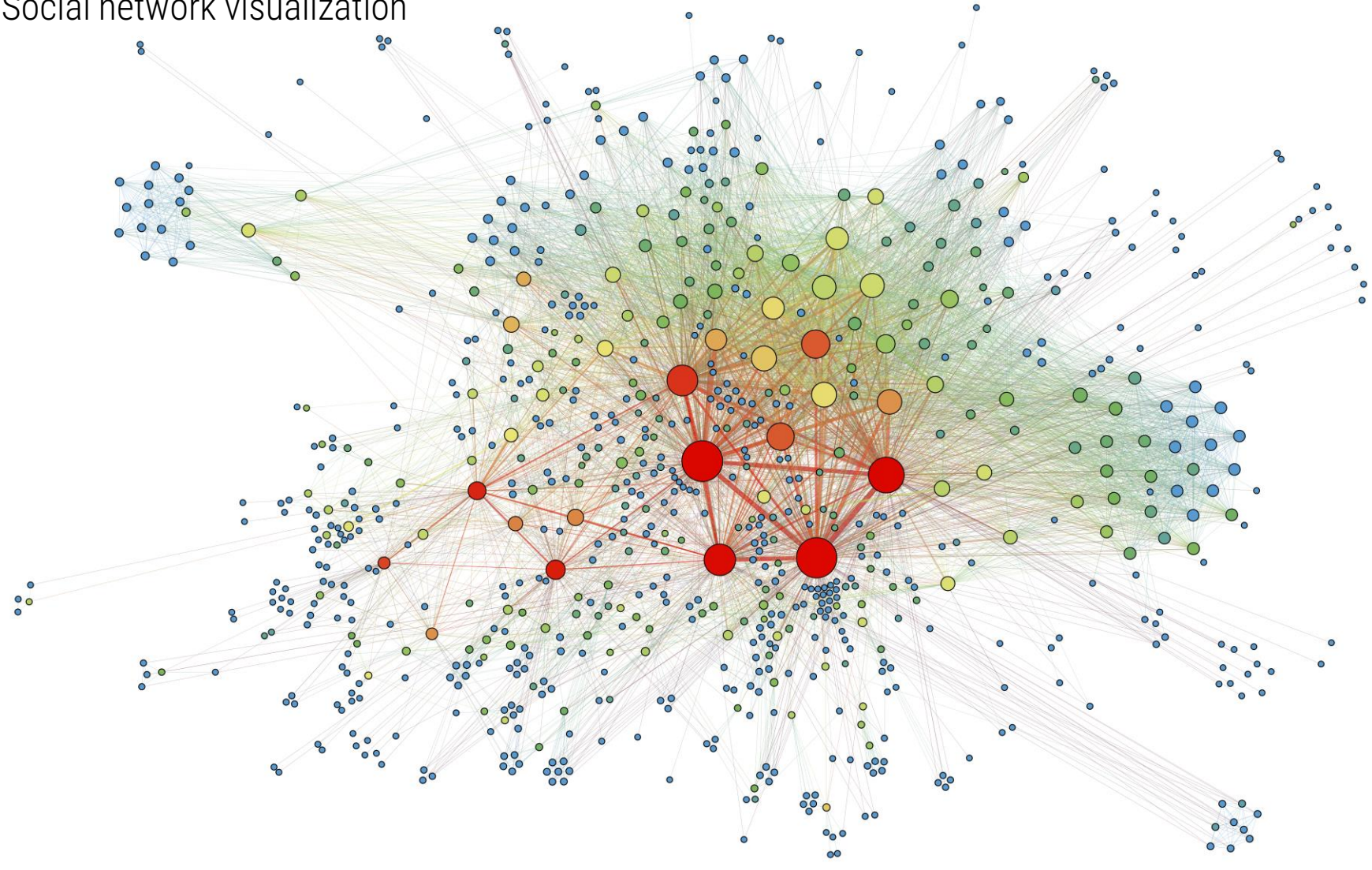
Our goal is to define a list of tasks for graph visualization that has enough detail and specificity to be useful to: 1) designers who want to improve their system and 2) to evaluators who want to compare graph visualization systems. In this paper, we suggest a list of tasks we believe are commonly encountered while analyzing graph data. We define graph specific objects and demonstrate how all complex tasks could be seen as a series of low-level tasks performed on those objects. We believe that our user studies of graph visualization techniques and extracted the tasks used in those studies.

After making those two lists, we considered the set of low-level Visual Analytics tasks proposed by Amar *et al.* [2]. These tasks were extracted from a corpus of questions about tabular data. We realized that our tasks all seem to be compound tasks made up of Amar *et al*'s primitive tasks applied to the graph objects. When some tasks could not be represented with those tasks and objects, we added either an object or a low-level task. In this paper, we

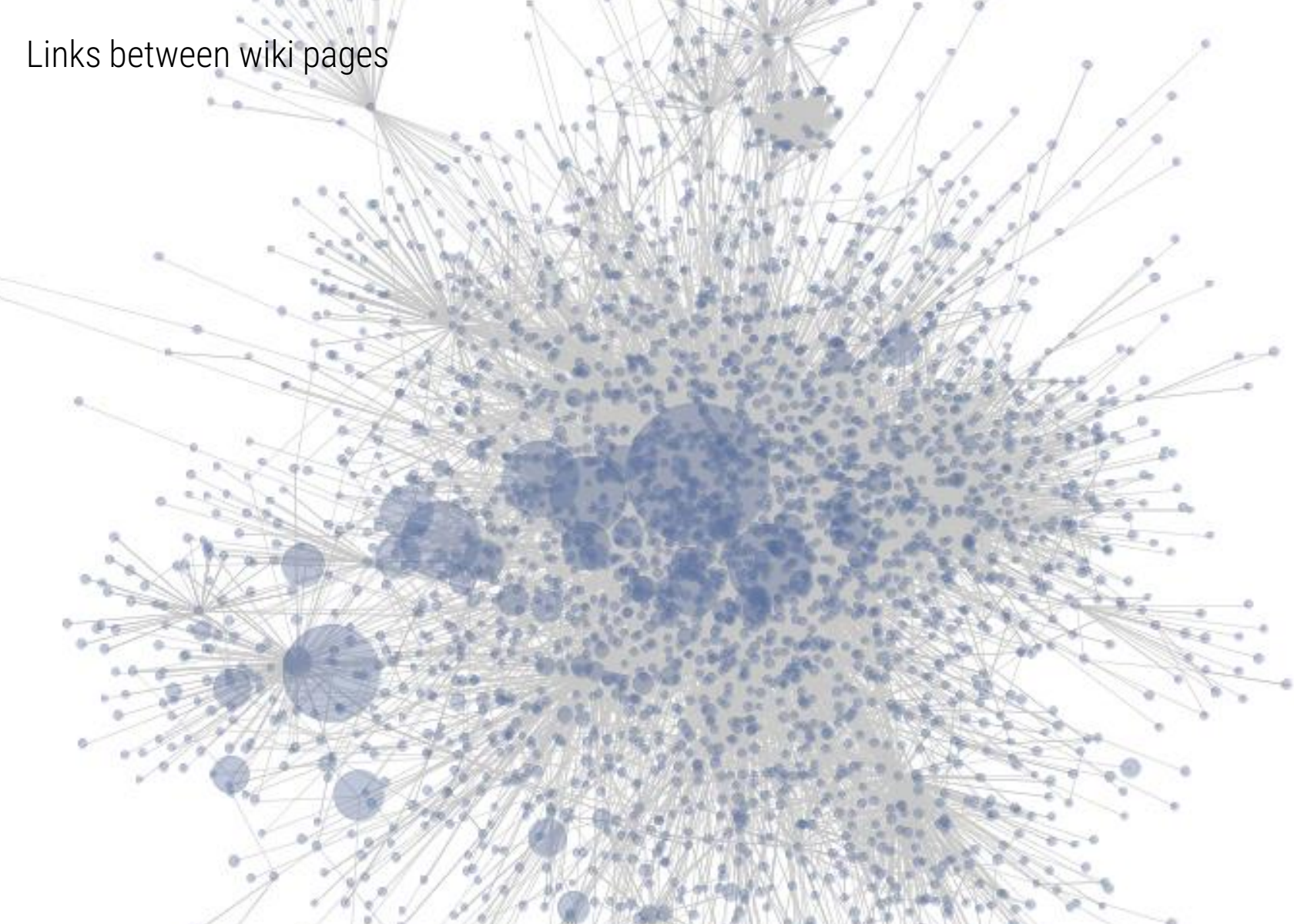
# ATTRIBUTES

- Graph structure describes TOPOLOGY
- Vertices and edges can have attributes
  - labels, weights, lengths, counts…
- Attributes can be used to adjust layouts
- Attributes add new tasks
  - *shortest cycle through cities with largest population*

# Social network visualization

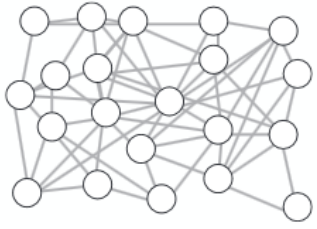


Links between wiki pages



# GRAPH CHALLENGES

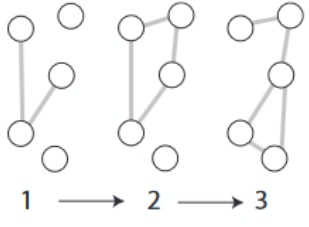
Size



Density

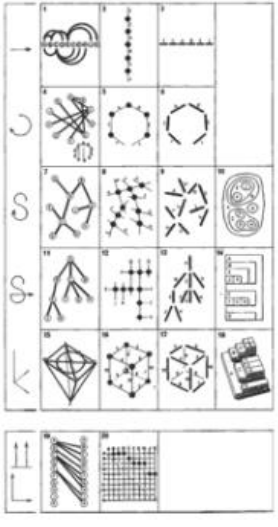
Types +
Attributes

Geography

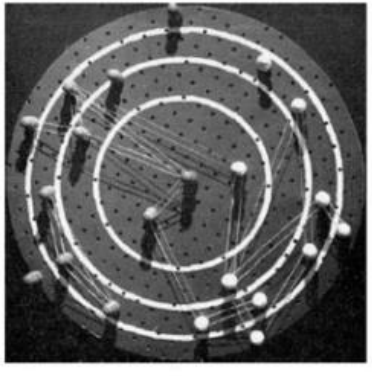



Time

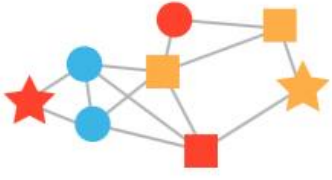
# GRAPH VISUALIZATION CHALLENGES



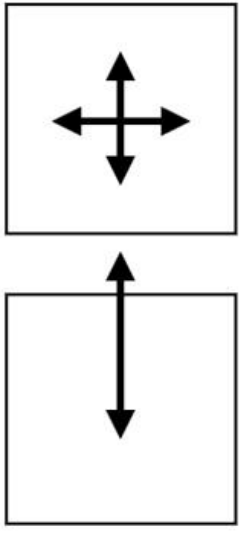
Representation



Layout



Types + Attributes



Navigation

# SOCIAL NETWORK ANALYSIS

https://datascientistinsights.com/2014/02/18/art-of-resistance-the-social-network-anatomy-of-a-kinetic-activist-group/

- determine if Greenpeace was or could become a significant disruptive geopolitical force
- first: identify who/what to concentrate resources on

# SOCIAL NETWORK ANALYSIS

https://datascientistinsights.com/2014/02/18/art-of-resistance-the-social-network-anatomy-of-a-kinetic-activist-group/



## Art of Resistance – The Social Network Anatomy of a Kinetic Activist Group

BY DR. J *on* FEBRUARY 18, 2014 · ( 0 )

As a data scientist that works in the intelligence community, we are often asked to help identify where intelligence gathering and analysis resources should be allocated. Governmental and non-governmental

# SOCIAL NETWORK ANALYSIS

## 1) get Facebook data using Netvizz

### Studying Facebook via Data Extraction: The Netvizz Application

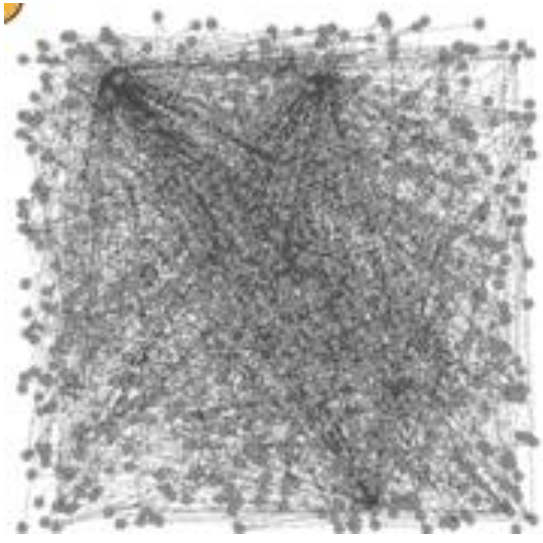
**Bernhard Rieder**
University of Amsterdam
Turfdraagsterpad 9
1012TX Amsterdam
rieder@uva.nl

**ABSTRACT**
This paper describes Netvizz, a data collection and extraction application that allows researchers to export data in standard file formats from different sections of the Facebook social networking service. Friendship networks, groups, and pages can thus be analyzed quantitatively and qualitatively with regards to demographical, post-

numerous publications employing conceptual and/or critical approaches. While traditional empirical methods such as interviews, experiments, and observations are widely used, a growing number of studies rely on what the authors call "data crawling", i.e. "gleaning information about users from their profiles without their active participation" [19]. This paper presents a software tool, Netvizz, designed to

# SOCIAL NETWORK ANALYSIS

## 2) load the data into Gephi https://gephi.org/



585 nodes, interconnected by 1788 edges.
“Somewhere in that spaghetti is a potential bad guy, but where?”

# SOCIAL NETWORK ANALYSIS

## 3) choose a layout algorithm that makes sense for social networks

Force Atlas 2

provides some transparency into the network but still lacks any real clarity around behavioral importance

# SOCIAL NETWORK ANALYSIS

## 4) map an attribute to size of the nodes

betweenness centrality (number of shortest paths from all vertices to all others that pass through that node)

Bigger nodes are more central to behavioral dynamics.

Several nodes become central figures in the overall network.

# SOCIAL NETWORK ANALYSIS

## 5) highlight communities

color nodes by modularity / clusters

We now begin to see a clearer picture of who is doing what with whom.

What becomes really interesting at this stage is understanding some of the more nuanced relationships.

# SOCIAL NETWORK ANALYSIS

## 6) filter, explore, label

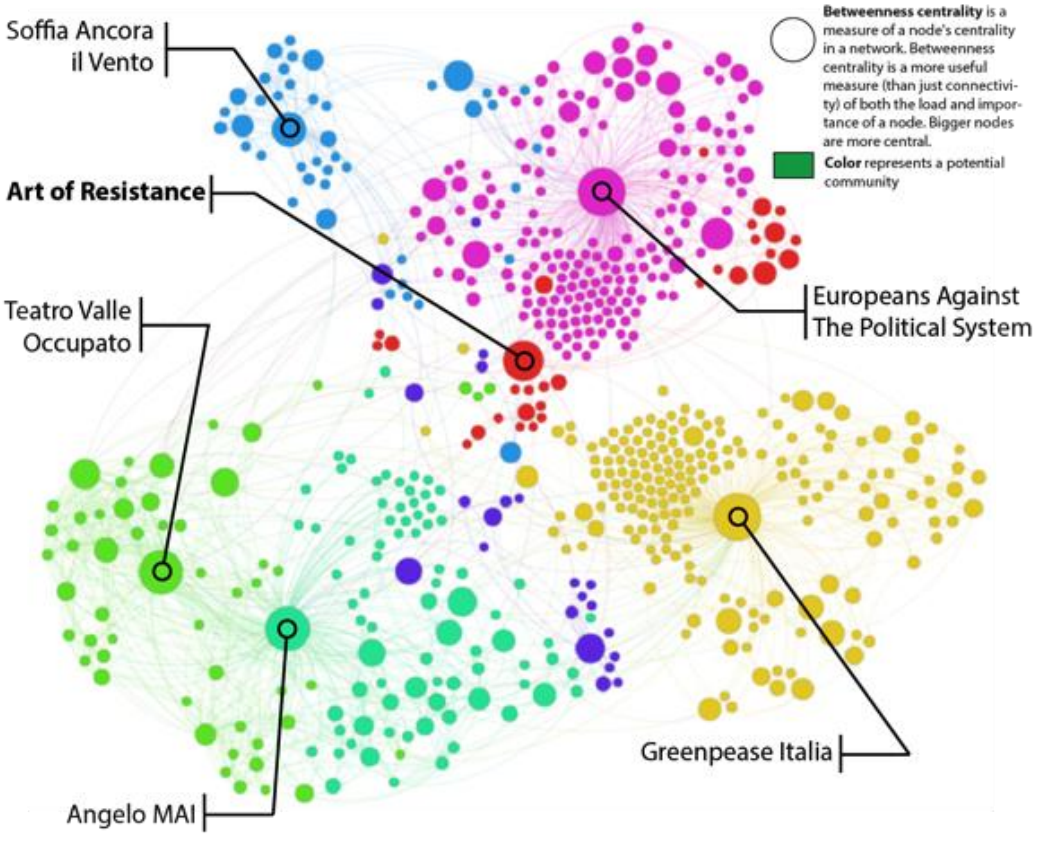

Five outlying nodes in the network (blue, maroon, yellow, dark green, and light green).

Center: an equally important red node

Emergence of a previously un-recognized activism player: [Art of Resistance](#).

# SOCIAL NETWORK ANALYSIS

7) communicate & explain



# LAYOUTS

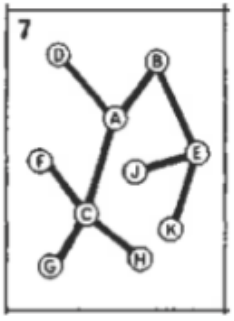
Important to the success of your analysis

# LAYOUT FREE

- Physical forces
- Proximity based

- Spring Model
- Kamanda&Kawai
- Frucherman&Reingold
- Davidson&Harel
- LinLog

**Force Directed**




# LAYOUT FREE

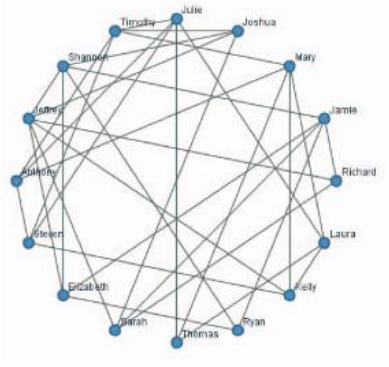
Aesthetic Criteria

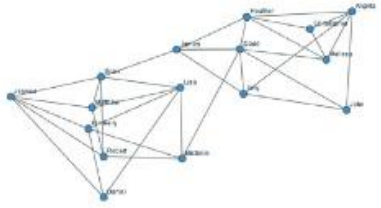
- Reduce number of edge crossing
- Foster Symmetry
- Uniform edge length
- Aspect Ratio
- Equal Angles
- ...

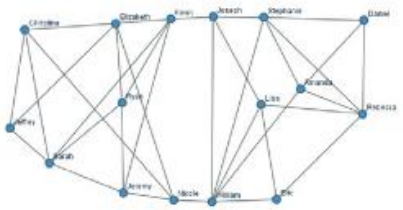
GRAPH DRAWING

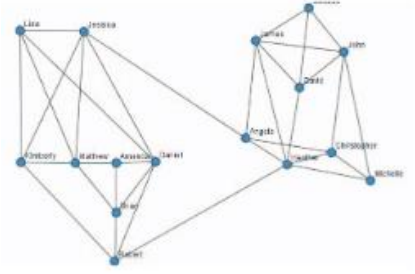
# LAYOUT FREE

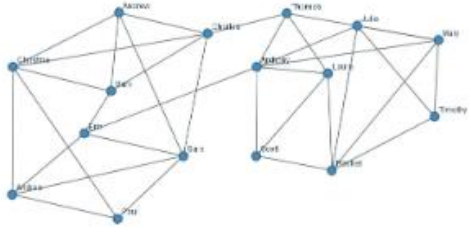
## HAND MADE

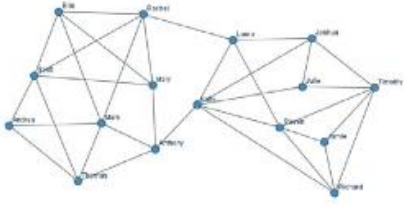


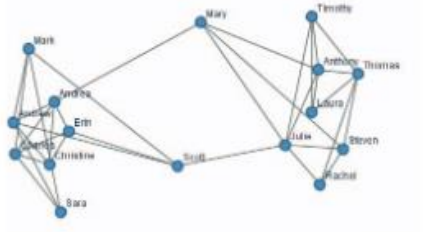



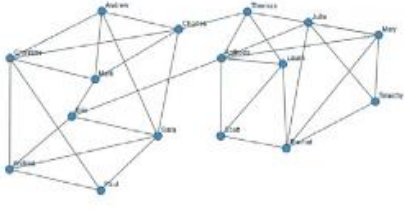


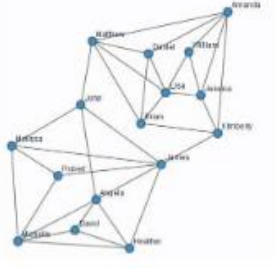






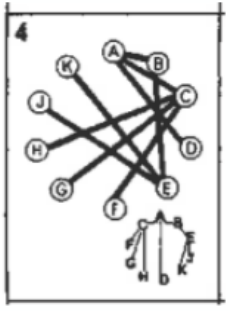


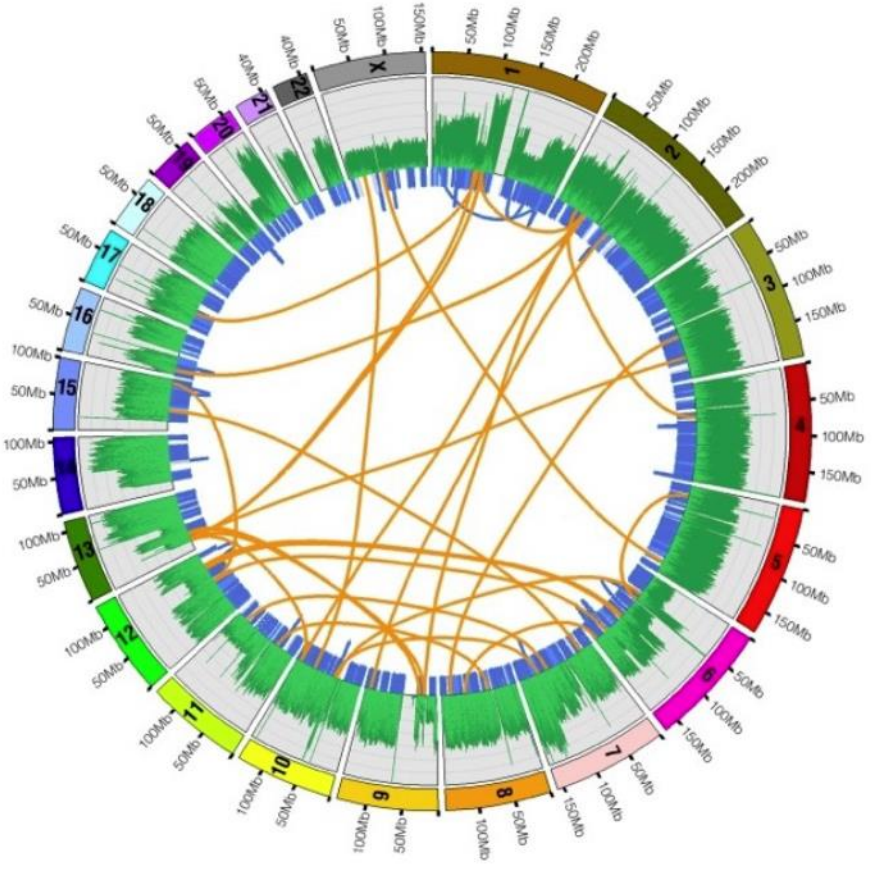




eptual organization in user-generated graph layouts
Ham, F.J.J.; Rogowitz, B.

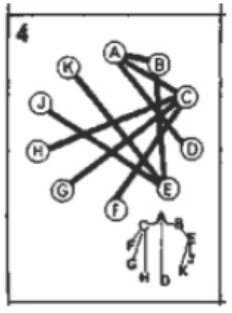
# LAYOUT CIRCULAR

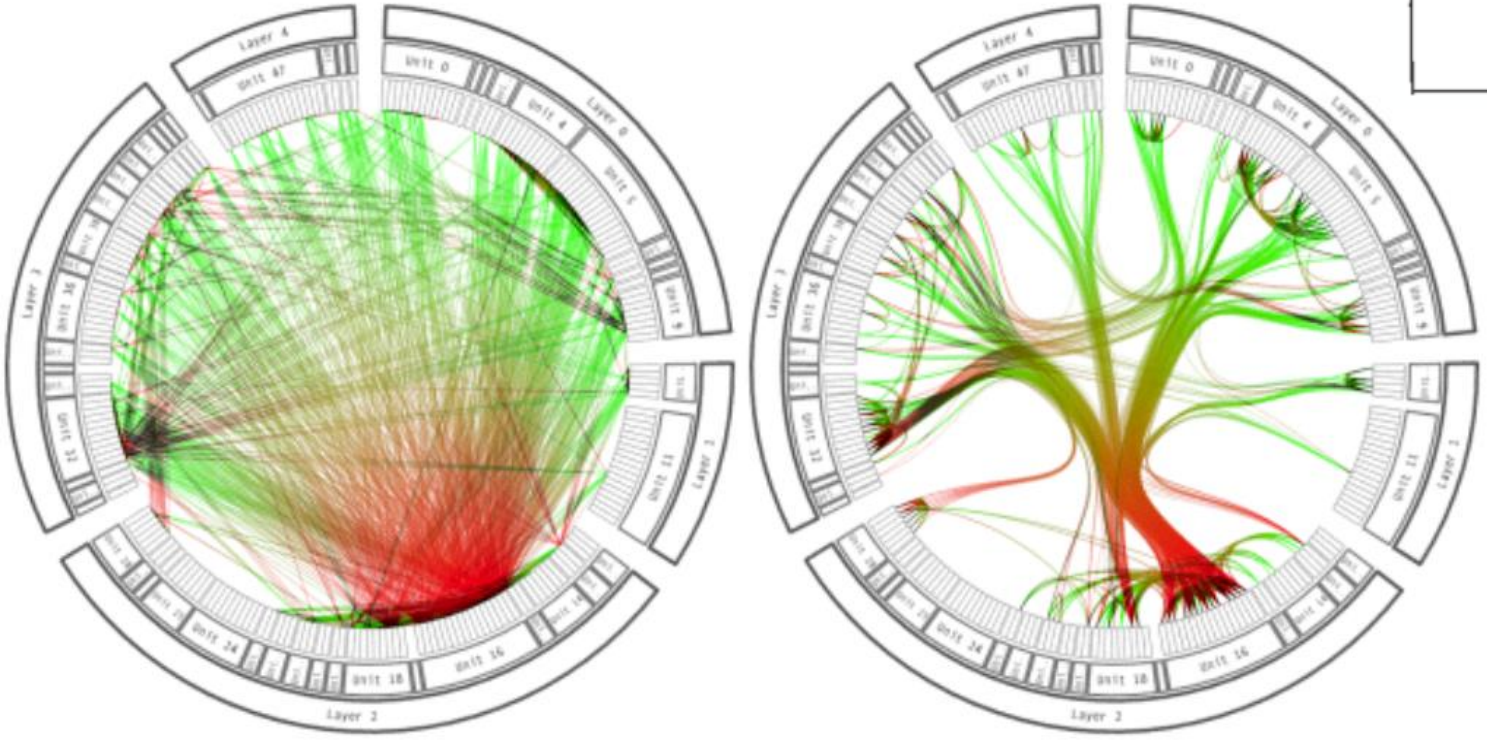



- Edges on the inside
- Vertices & attributes on the outside
- Ordering possible

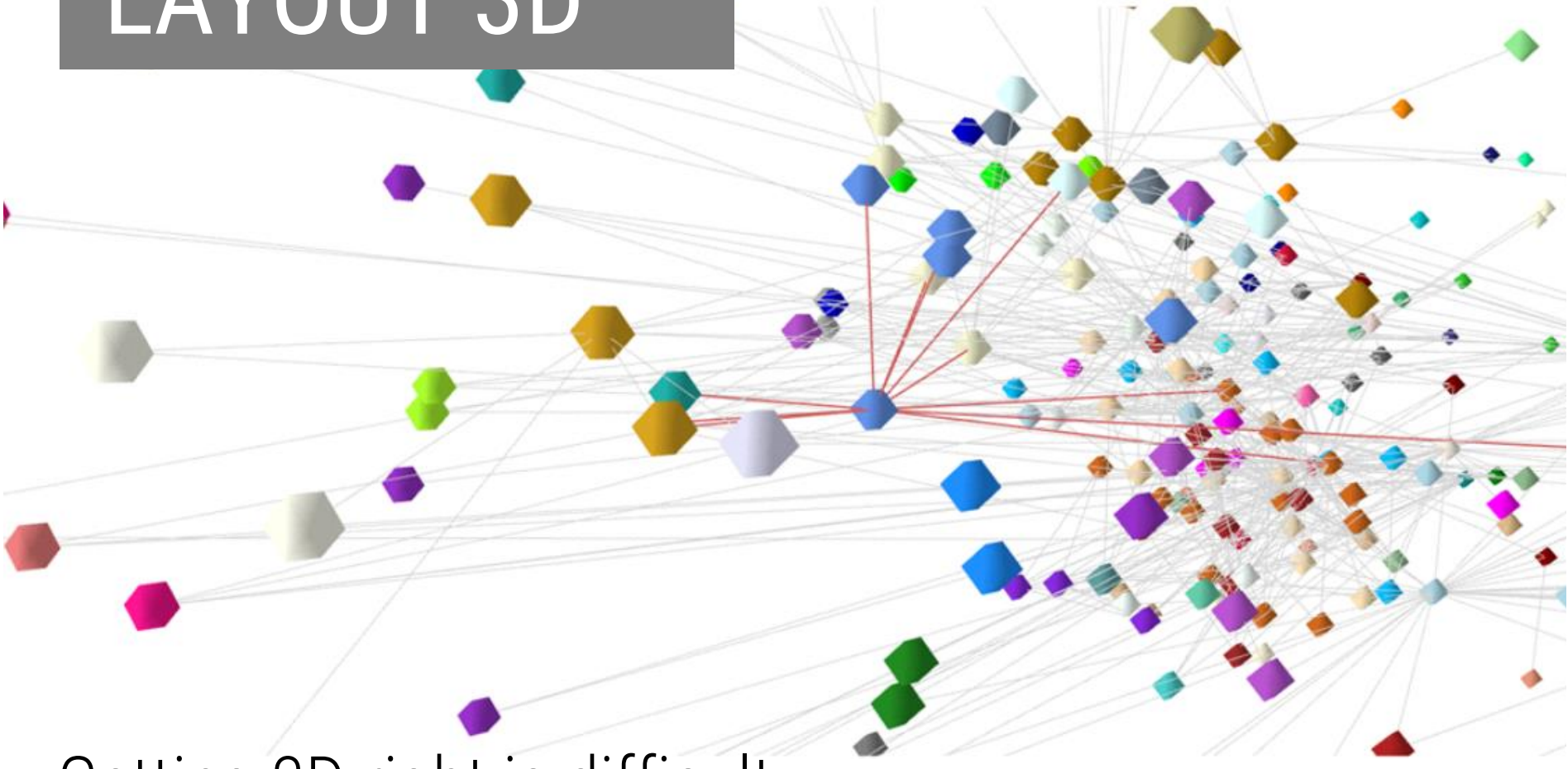
# LAYOUT CIRCULAR




**Edge Bundling**
Holten 2006

# LAYOUT 3D

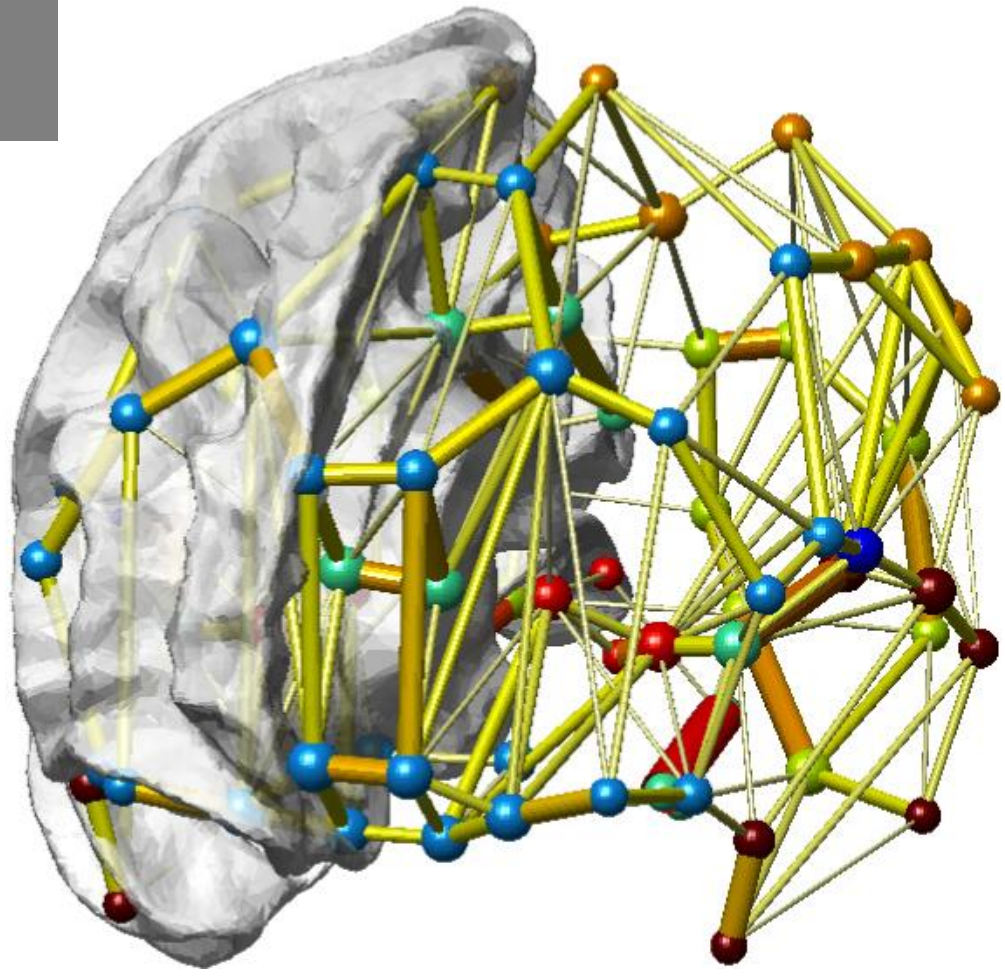


Getting 3D right is difficult

https://fwaris.wordpress.com/2012/07/08/a-simple-technique-for-creating-3d-graphs-from-2d-ones/

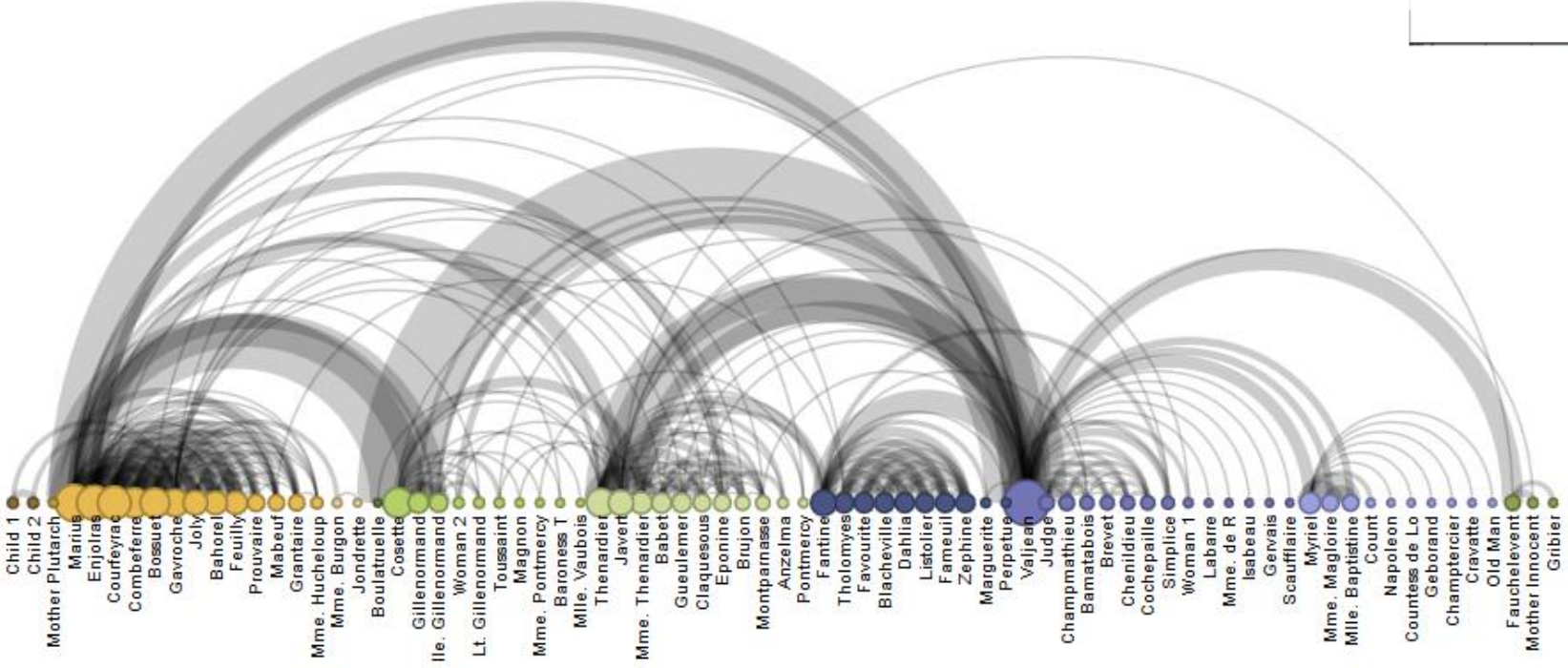
# LAYOUT 3D

Sometimes necessary (!?)



http://gephi.forumatic.com/viewtopic.php?f=26&t=2616

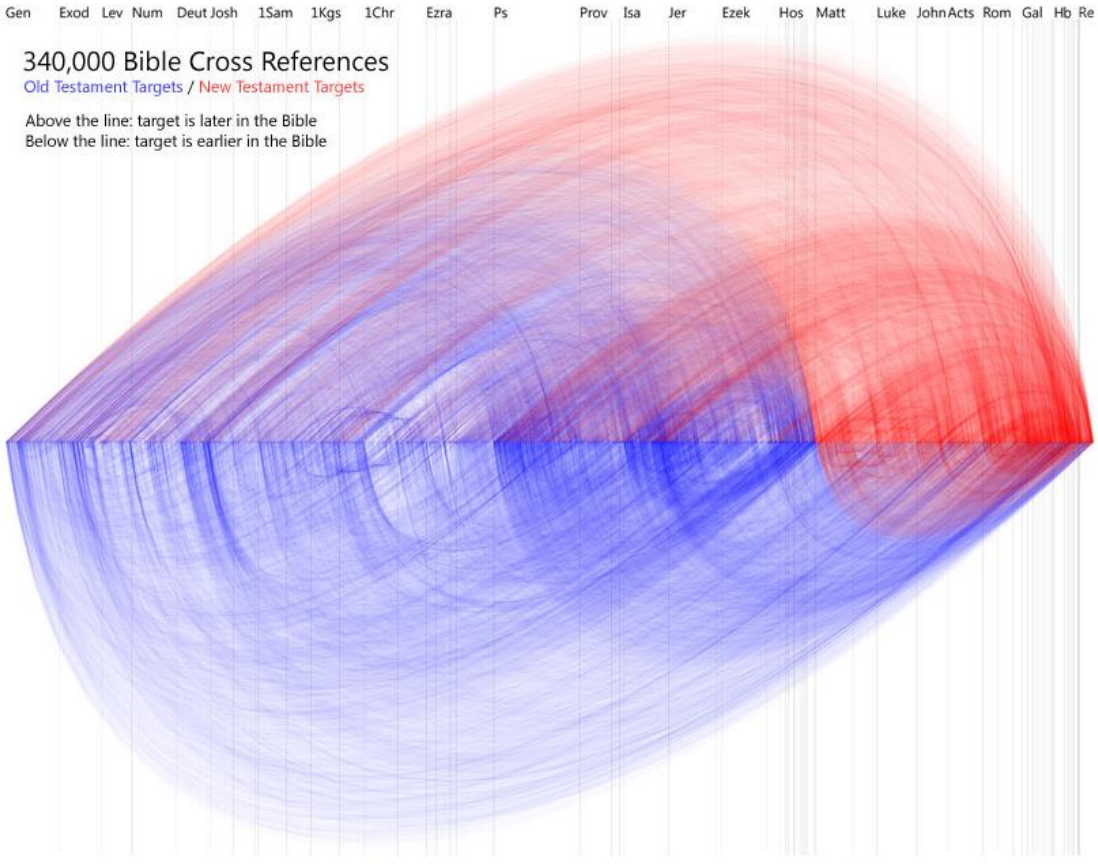
# LAYOUT LINEAR


http://mbostock.github.io/protovis/ex/arc-full.html

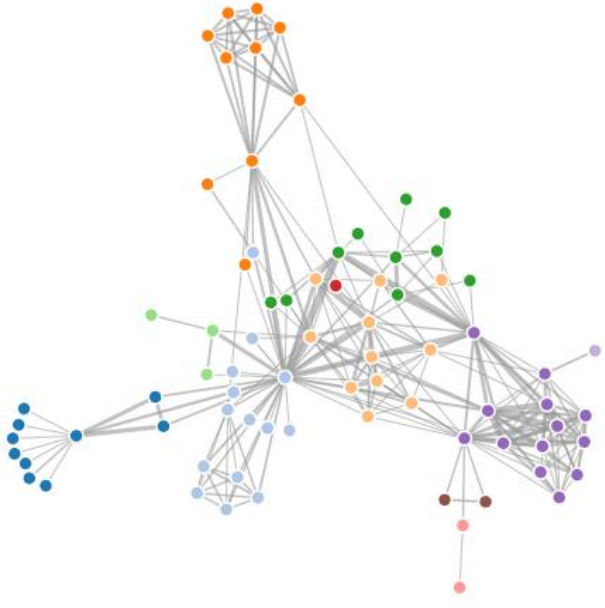
http://www.bewitched.com/song.html

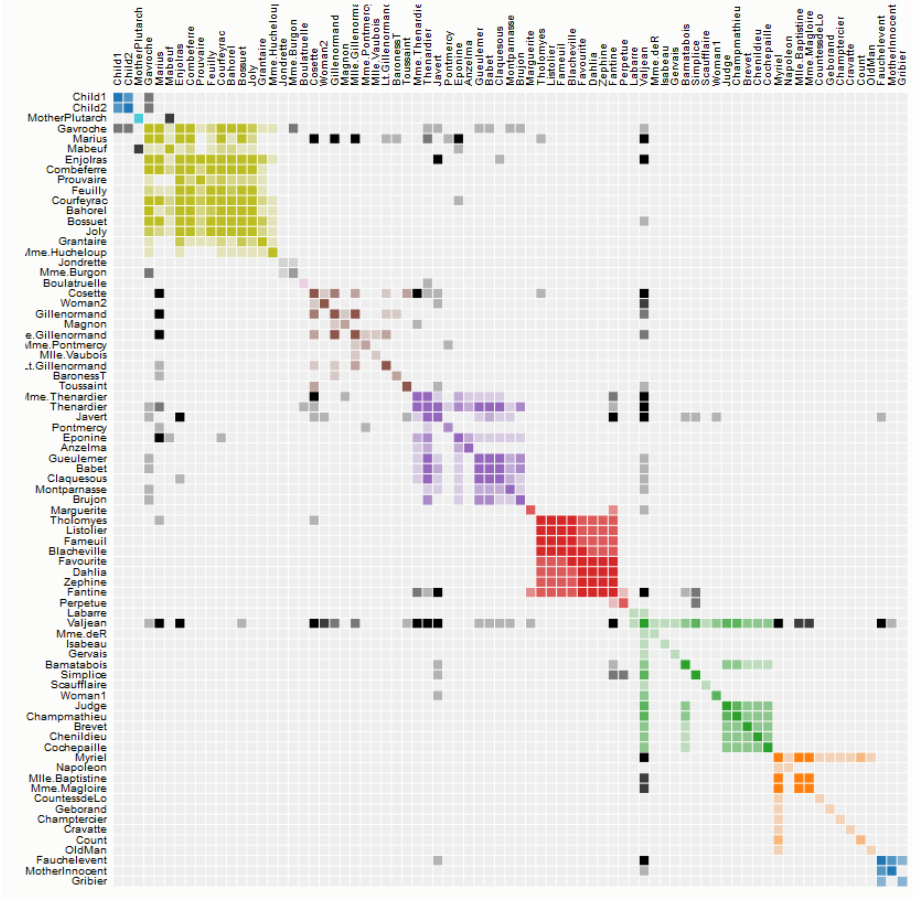
# LAYOUT LINEAR



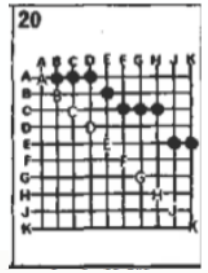

https://www.openbible.info/labs/cross-references/

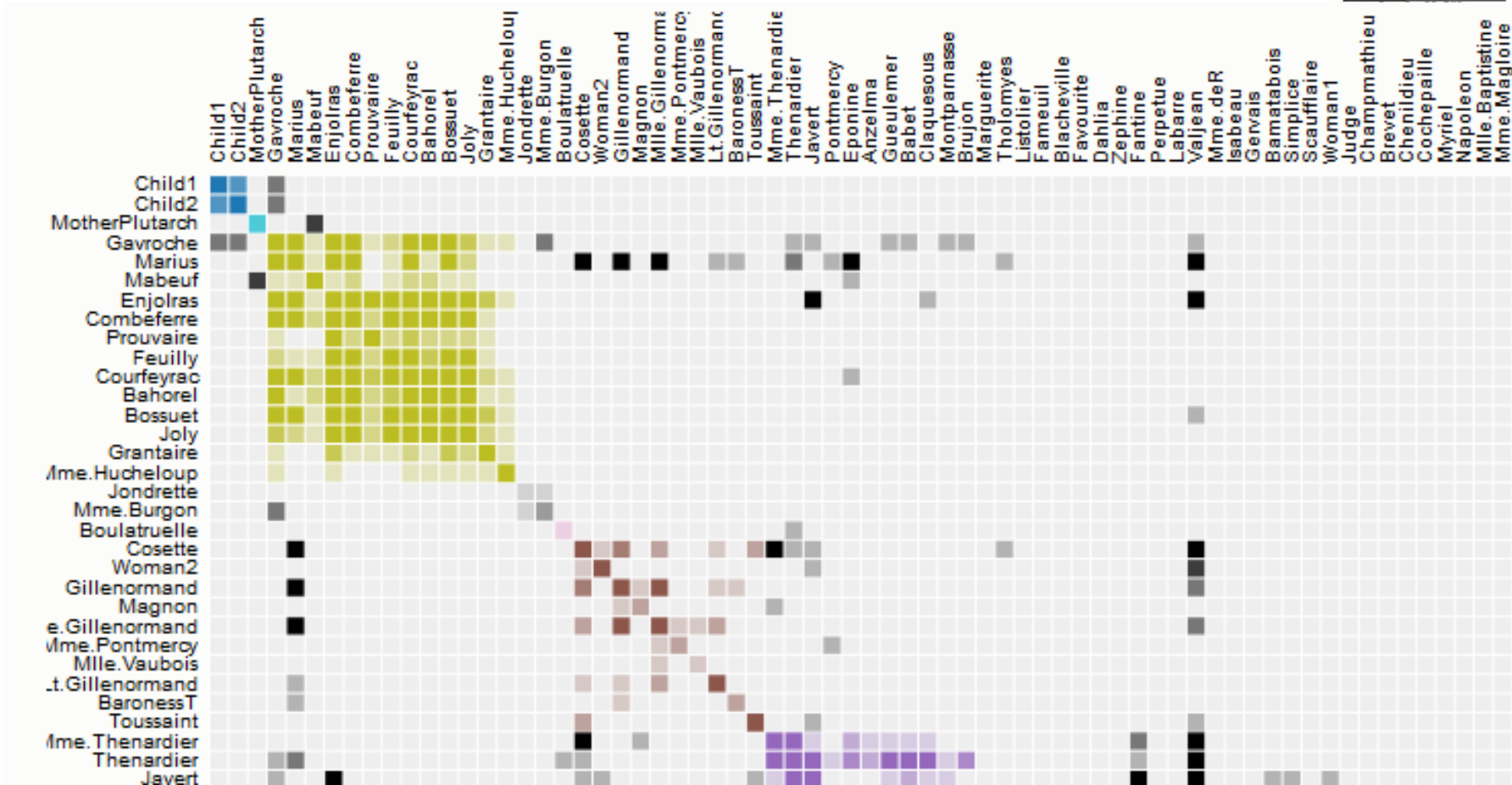
# LAYOUT ADJACENCY MATRIX



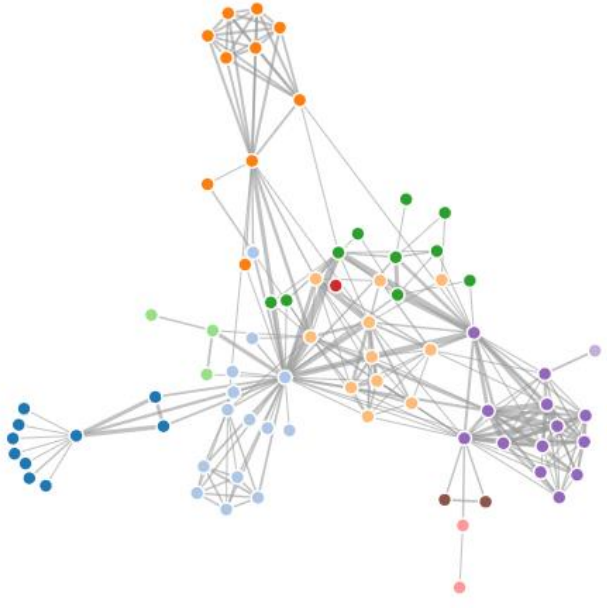


# ADJACENCY MATRIX

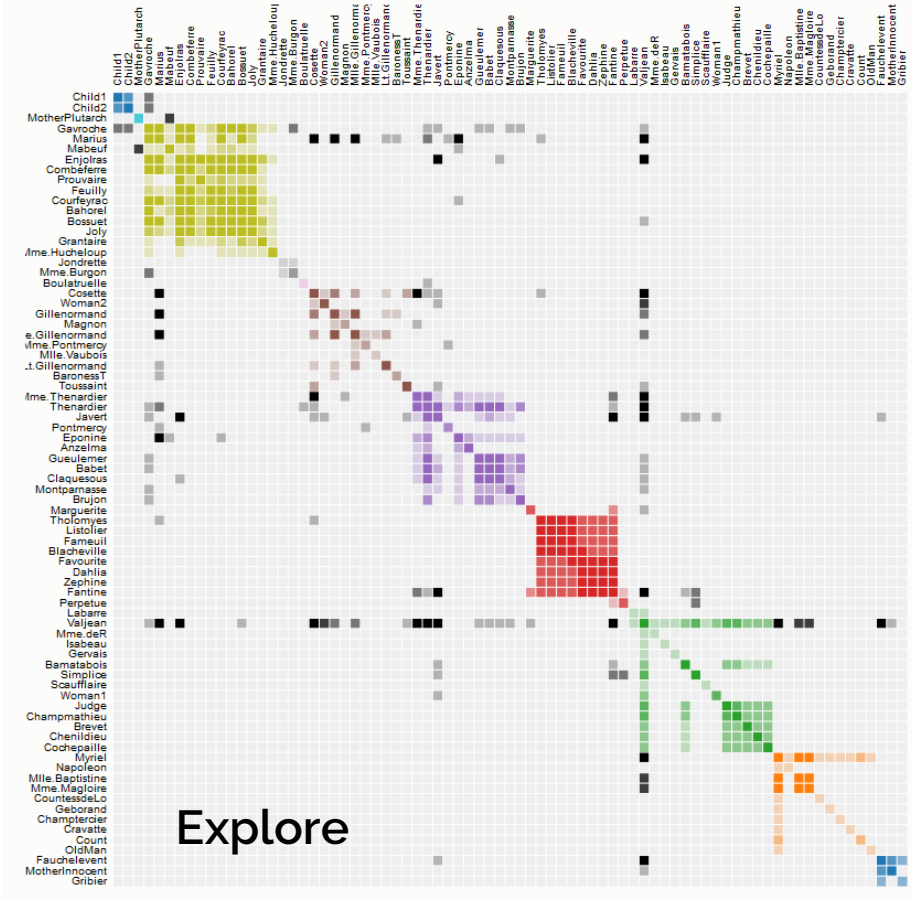


# LAYOUT ADJACENCY MATRIX



Communicate




Explore

# PROS/CONS

## Matrix

- No vertex/edge overlap or crossings
- Readable for dense graph
- Fast navigation

## Node-Link

- Familiar
- Compact
- Path following easier
- Effective for small and sparse graphs

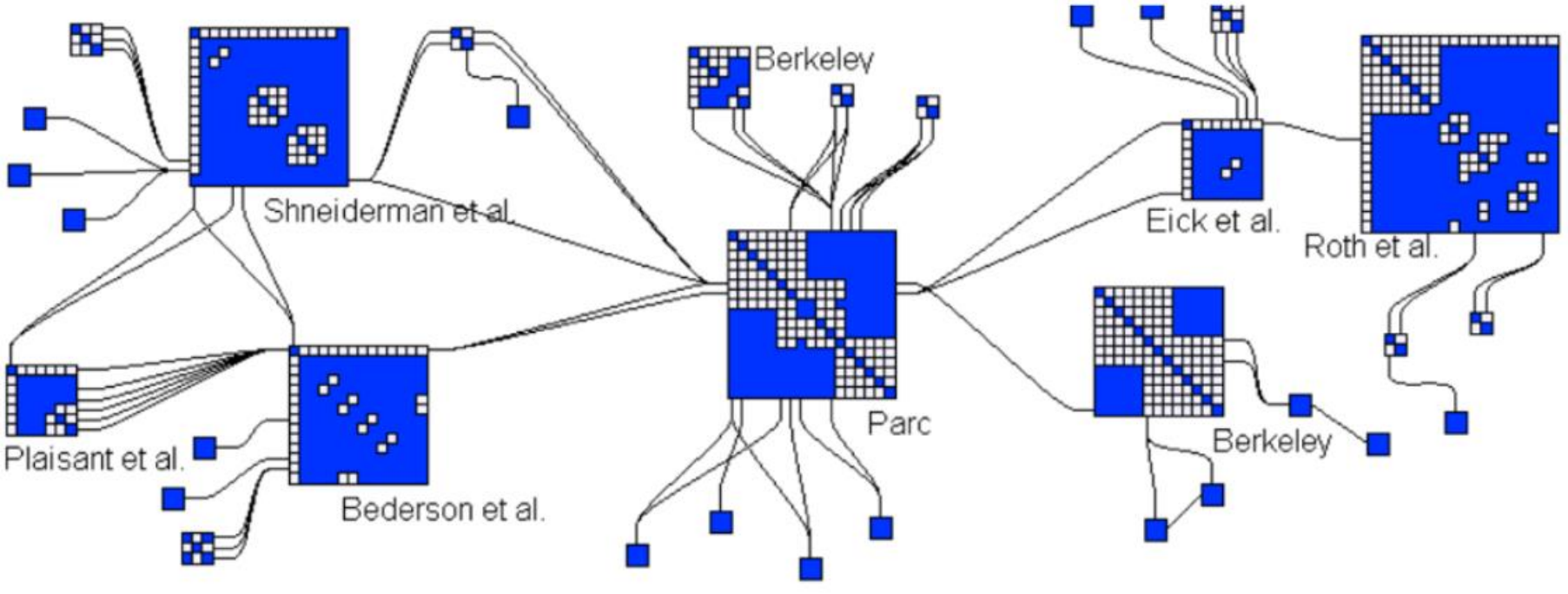
---

- Less familiar
- Space intensive
- Weak for path following tasks

- Useless without layout
- Not readable for dense graphs
- Manipulation requires layout computation

# HYBRID

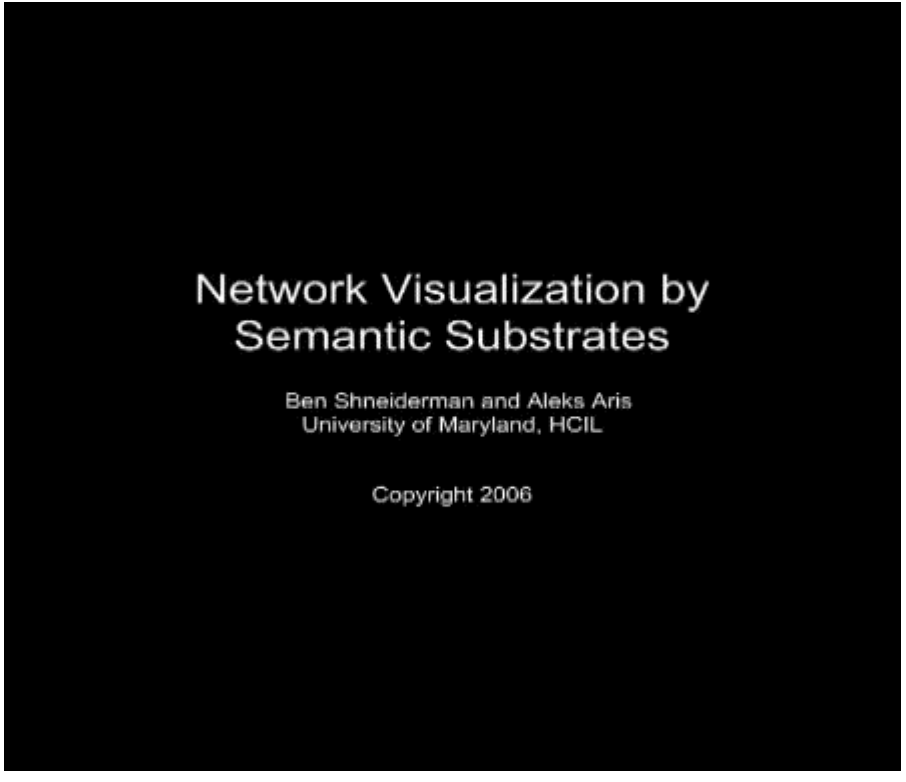
Henry et al., NodeTrix



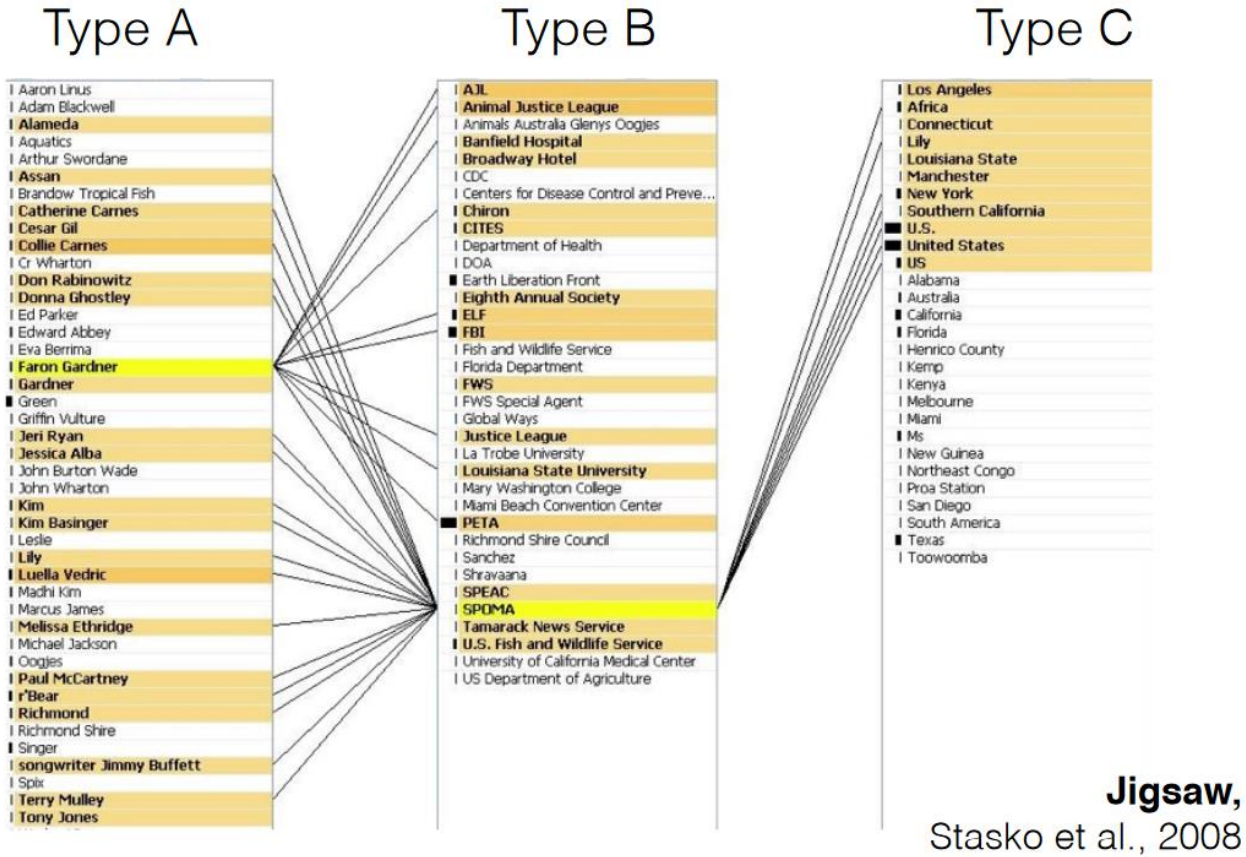

Infovis Coauthorship (133 actors)

Dense = matrices, sparse = node-link

# MULTIVARIATE NETWORKS

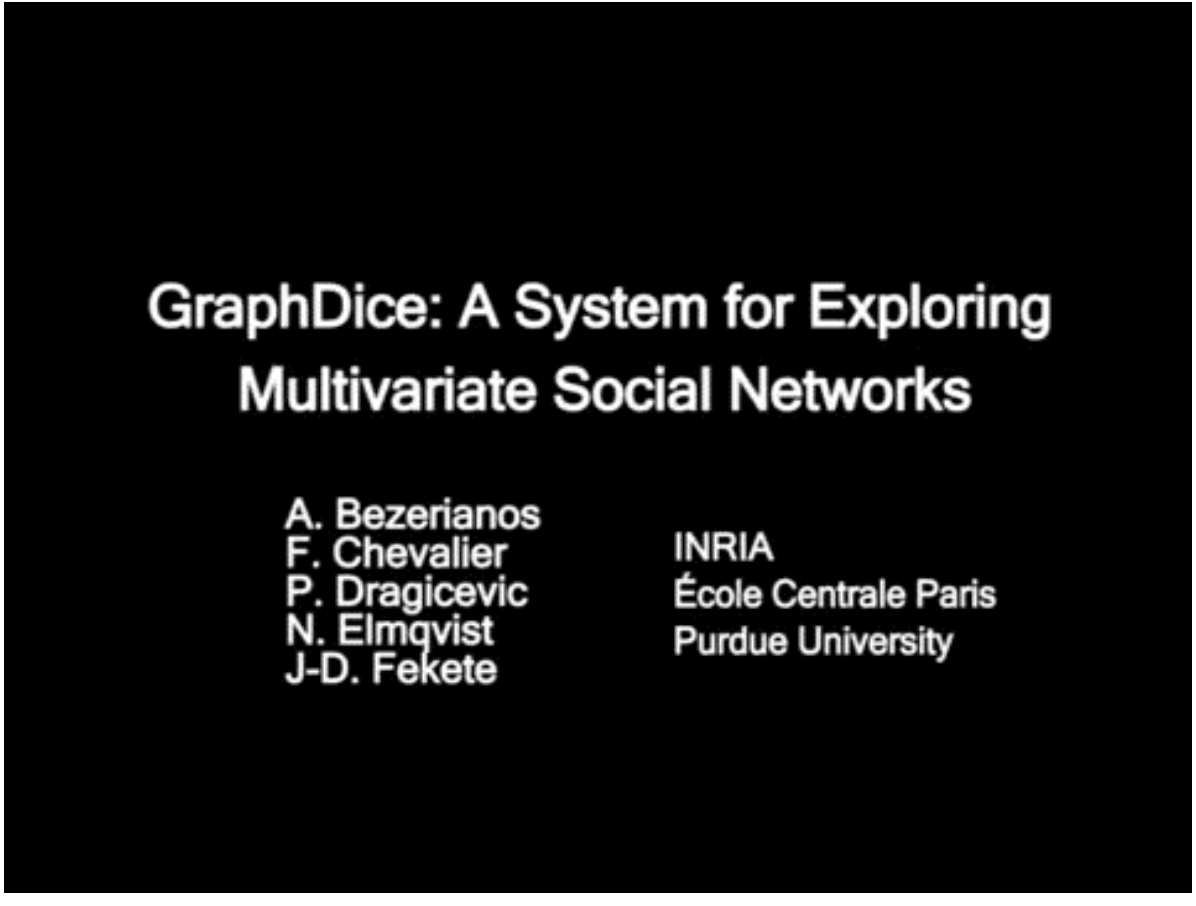


# MULTIVARIATE NETWORKS




**Jigsaw,**
Stasko et al., 2008

# MULTIVARIATE NETWORKS



# ADDITIONAL CHALLENGES

- TIME
- INTERACTION
- EDGE DIRECTION

# NETWORK TOOLS

- **Gephi**
- **Cytoscape**
- **Pajek**
- **Java** Jung toolkit
- **D3** + Cola.js

# ACKNOWLEDGEMENTS

Slides in were inspired and adapted from slides by

- Sheelagh Carpendale (University of Calgary)
- Pat Hanrahan (Stanford University)
- Benjamin Bach (University of Edinburgh)
- Jean-Daniel Fekete (Inria)