CSC 544
Data Visualization

Joshua Levine
josh@arizona.edu

Lecture 14
Graphs

Mar. 1, 2023

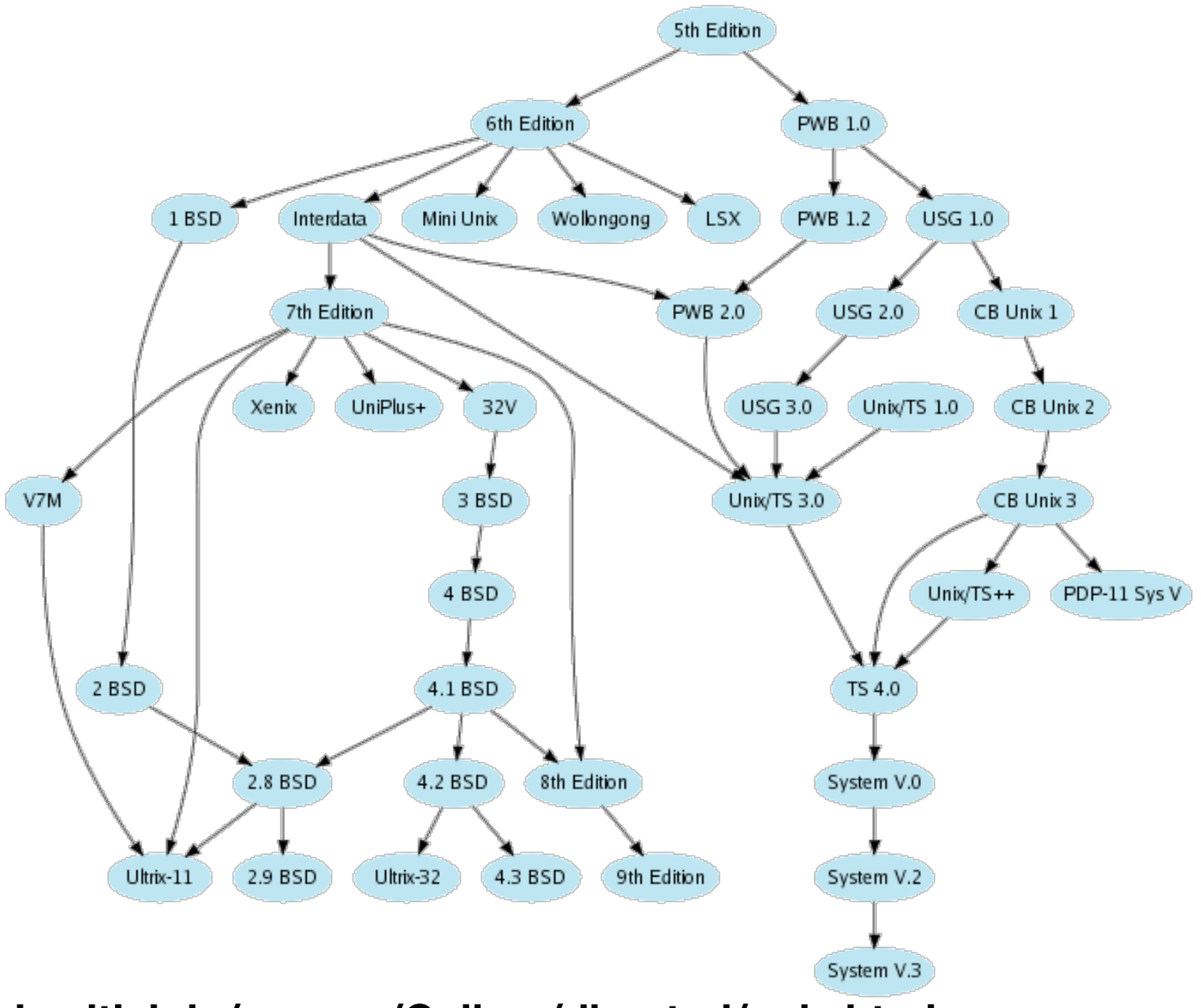
Today's Agenda

- Reminders:
  - A03 questions?, P02 questions?
- Goals for today:
  - Discuss visualization techniques for graphs

From Trees to DAGs

Not all Hierarchies are Trees

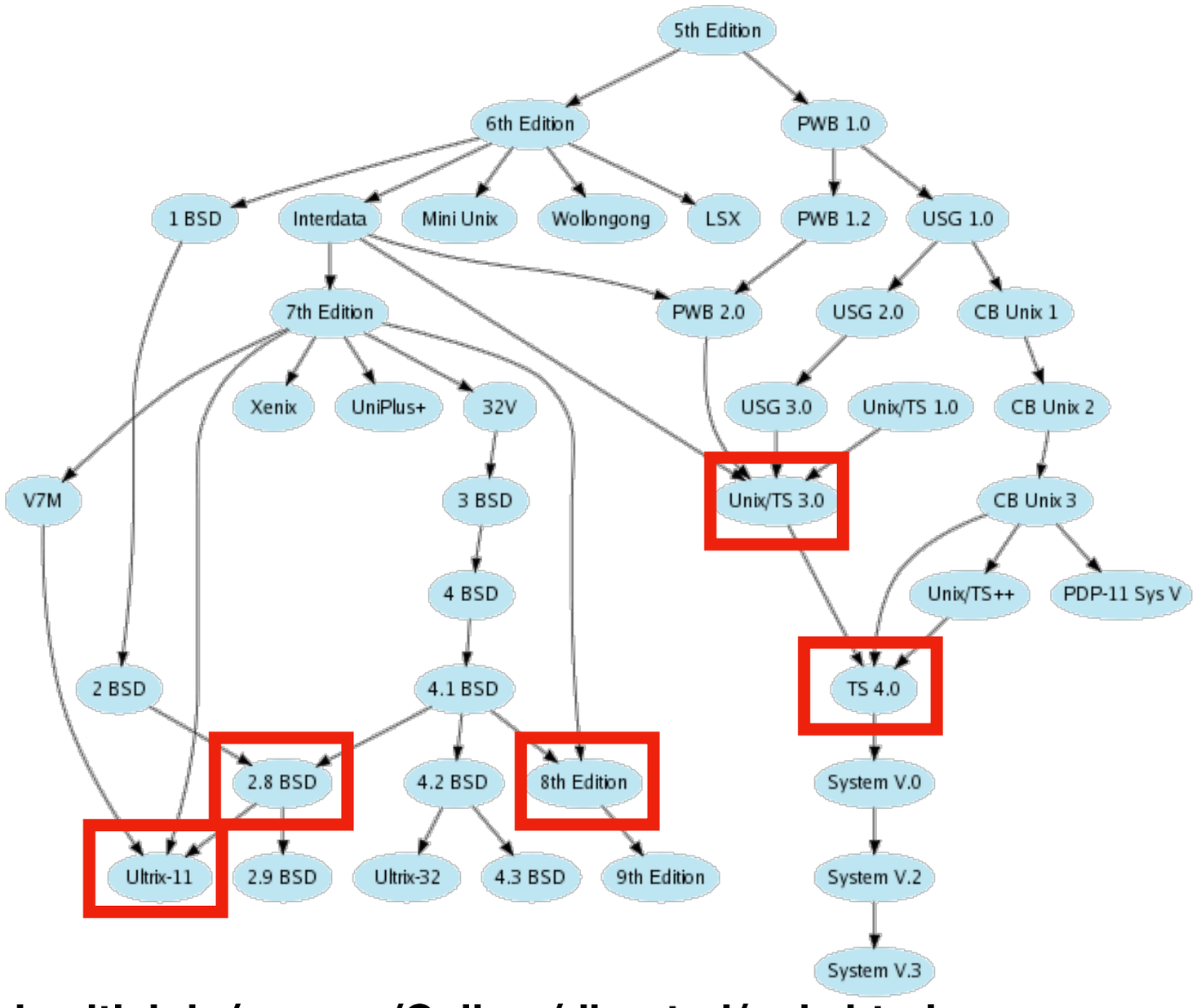
- In some hierarchies, the direct ancestor is not unique



Unix ancestry: https://graphviz.gitlab.io/_pages/Gallery/directed/unix.html

Not all Hierarchies are Trees

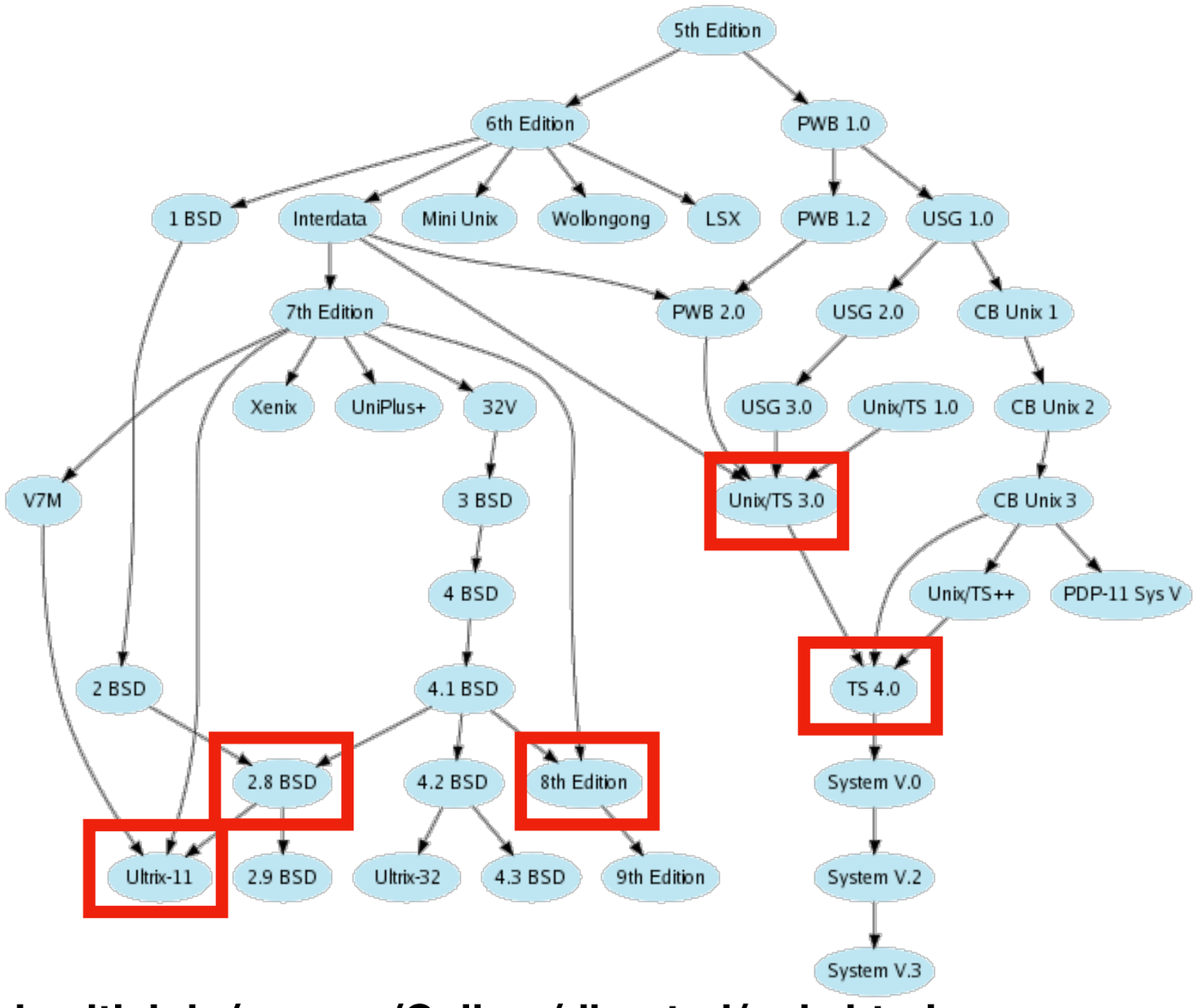
- In some hierarchies, the direct ancestor is not unique



Unix ancestry: https://graphviz.gitlab.io/_pages/Gallery/directed/unix.html

Not all Hierarchies are Trees

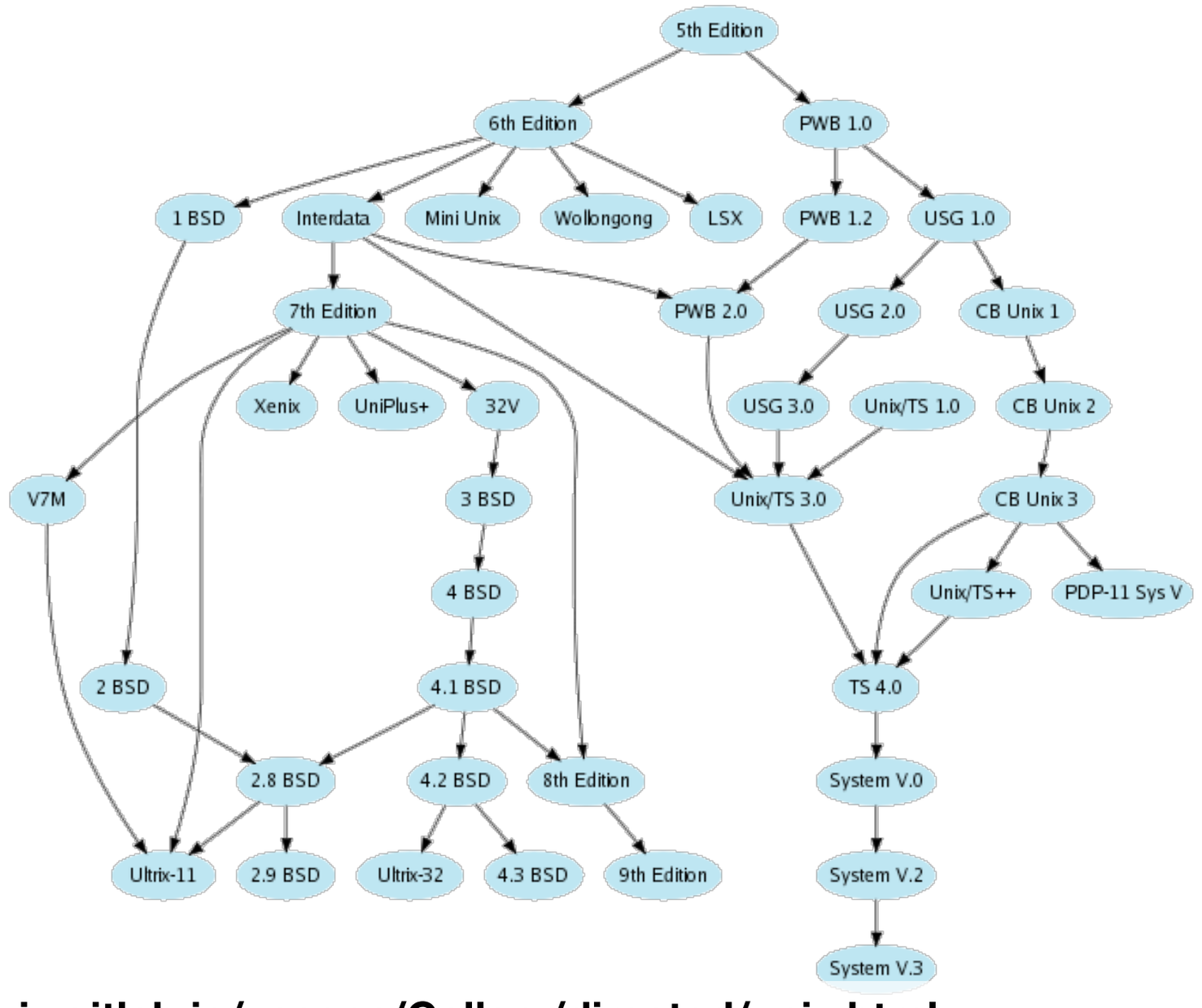
- In some hierarchies, the direct ancestor is not unique
- Can we apply similar approaches to directed, acyclic graphs (DAGs)?



Unix ancestry: https://graphviz.gitlab.io/_pages/Gallery/directed/unix.html

Node-Link Sugiyama Layout

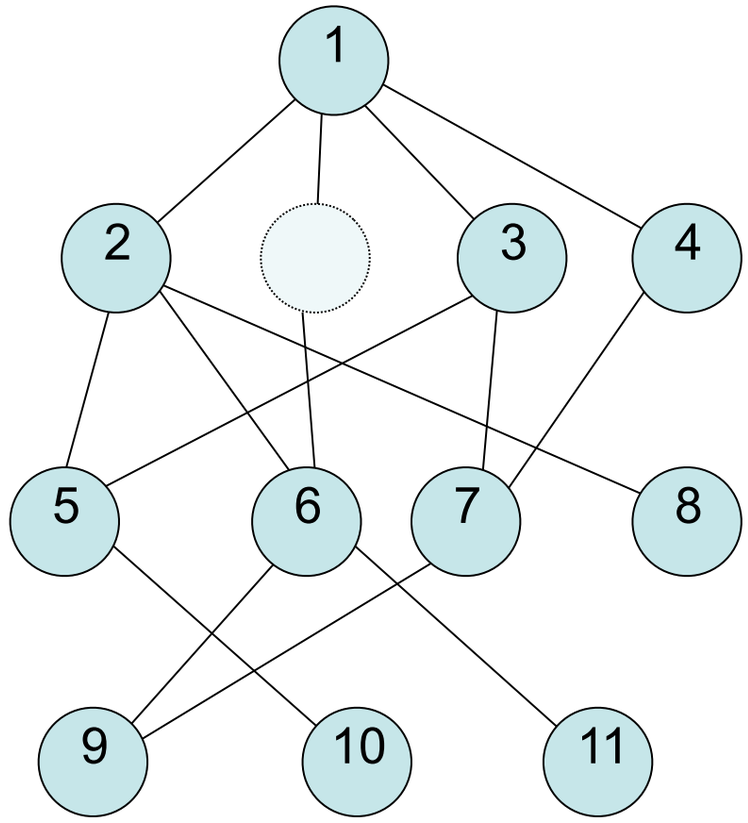
- Great for graphs that have an intrinsic ordering
- Layered approach where depth in graph mapped to one axis



Unix ancestry: https://graphviz.gitlab.io/_pages/Gallery/directed/unix.html

Sugiyama Step 1

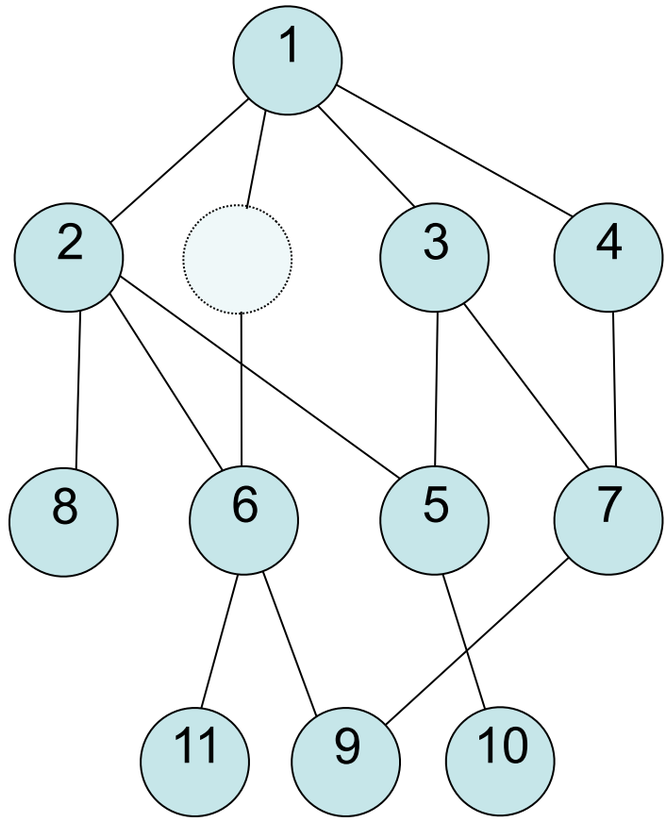
- Create layering of graph, potentially using:
  - Domain specific knowledge
  - Longest path from root
  - Algorithmically determine best layering (NP-Hard)
- Dummy nodes for long edges



Gansner et al., A Technique for Drawing Directed Graphs. IEEE Trans on Soft Engr (1993)

Sugiyama Step 2

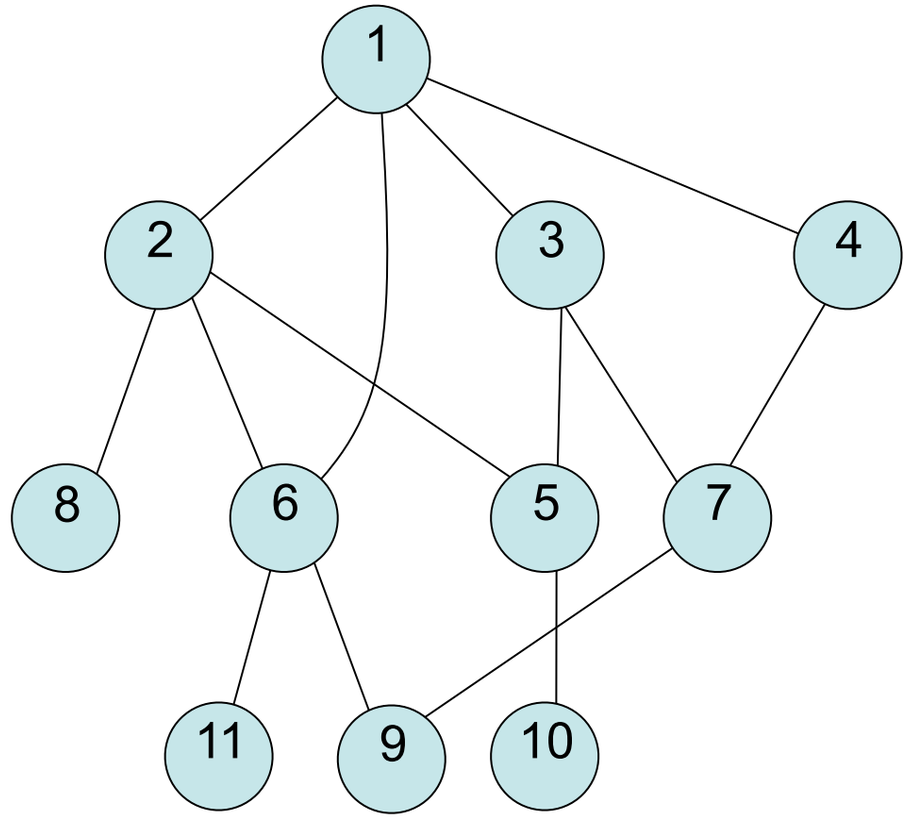
- Minimize crossings layer by layer (NP-hard)
- Numerous heuristics available



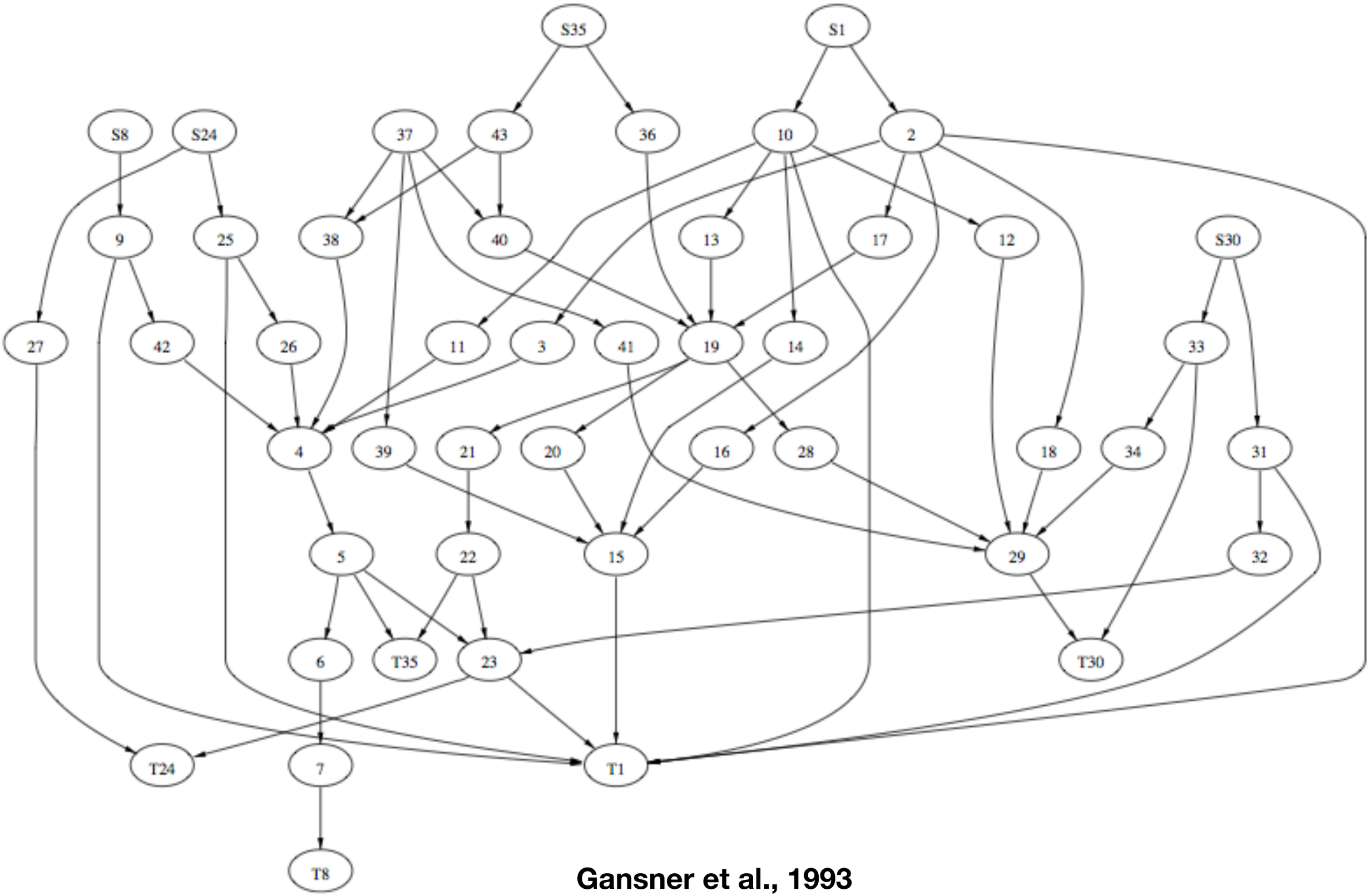
Gansner et al., A Technique for Drawing Directed Graphs. IEEE Trans on Soft Engr (1993)

Sugiyama Step 3

- Final assignment of x-coordinates
- Routing of edges



Gansner et al., A Technique for Drawing Directed Graphs. IEEE Trans on Soft Engr (1993)



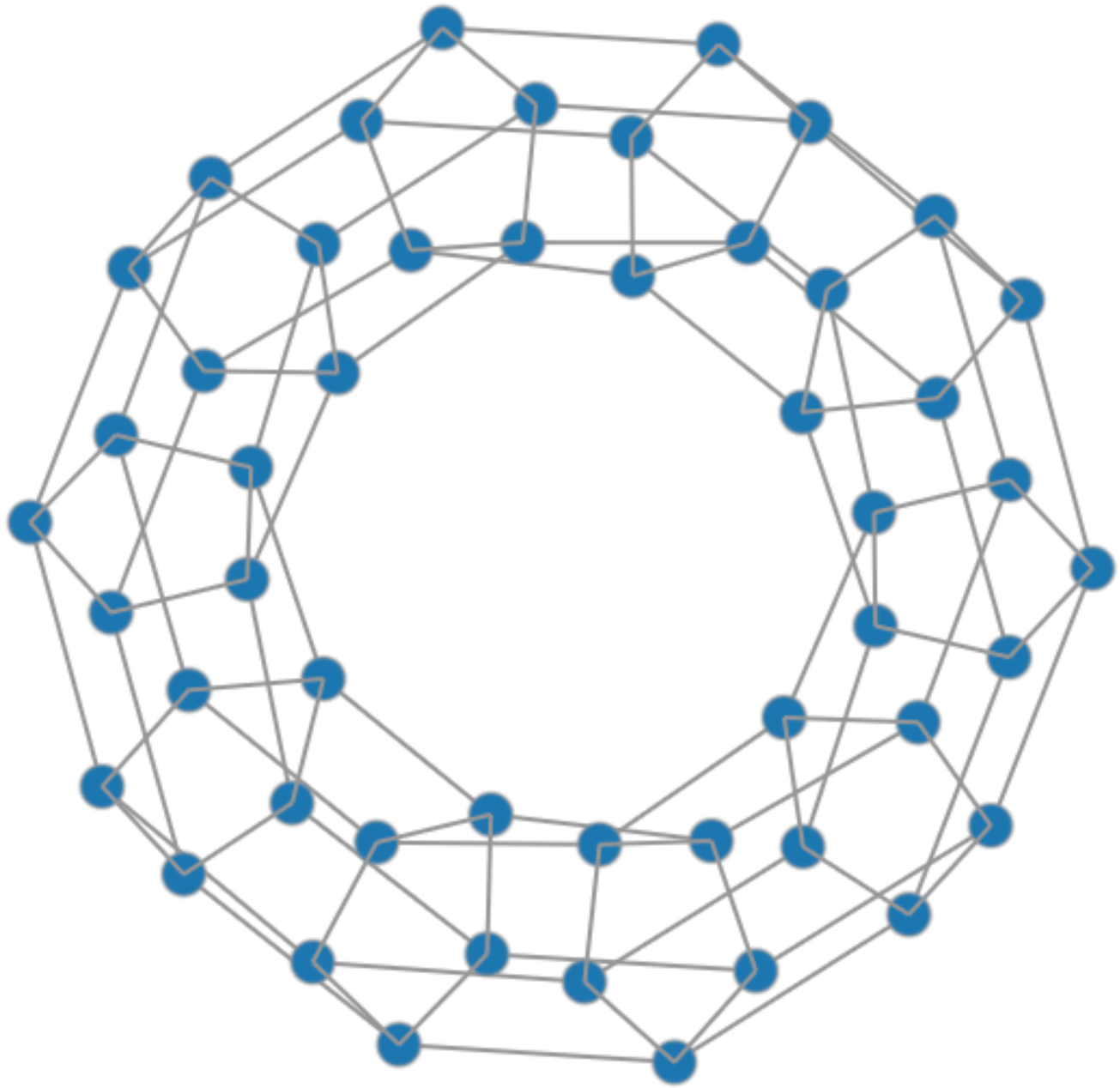
Gansner et al., 1993

Sugiyama Layout

- Strengths:
    - Nice, readable top down flow
    - Relatively fast (depending on heuristic used for crossing minimization)
- Limitations:
    - Not really suitable for graphs that don’t have an intrinsic top down structure
    - Can be hard to implement
        - Use free graphviz lib instead: http://www.graphviz.org

Visualizing Graphs as Node Link Diagrams

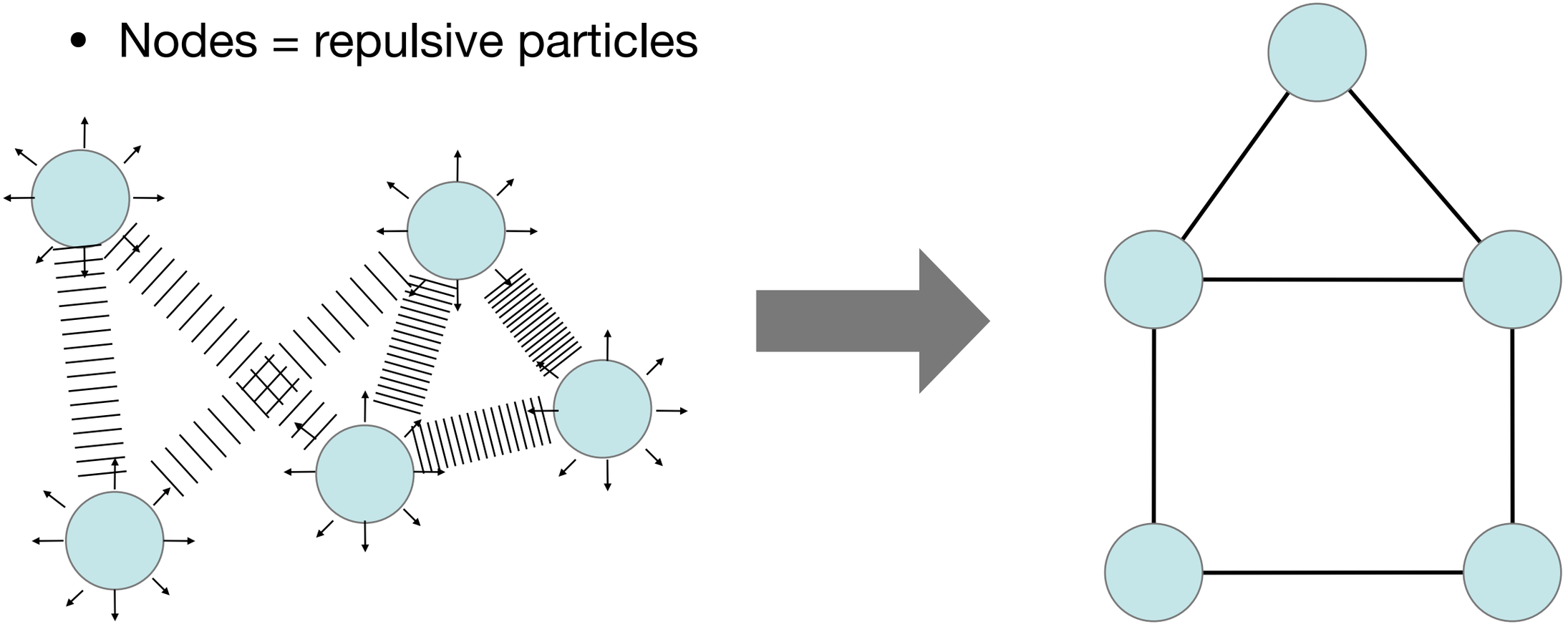
Node-Link Diagrams



https://blocks.roadtolarissa.com/christophermanning/1703449

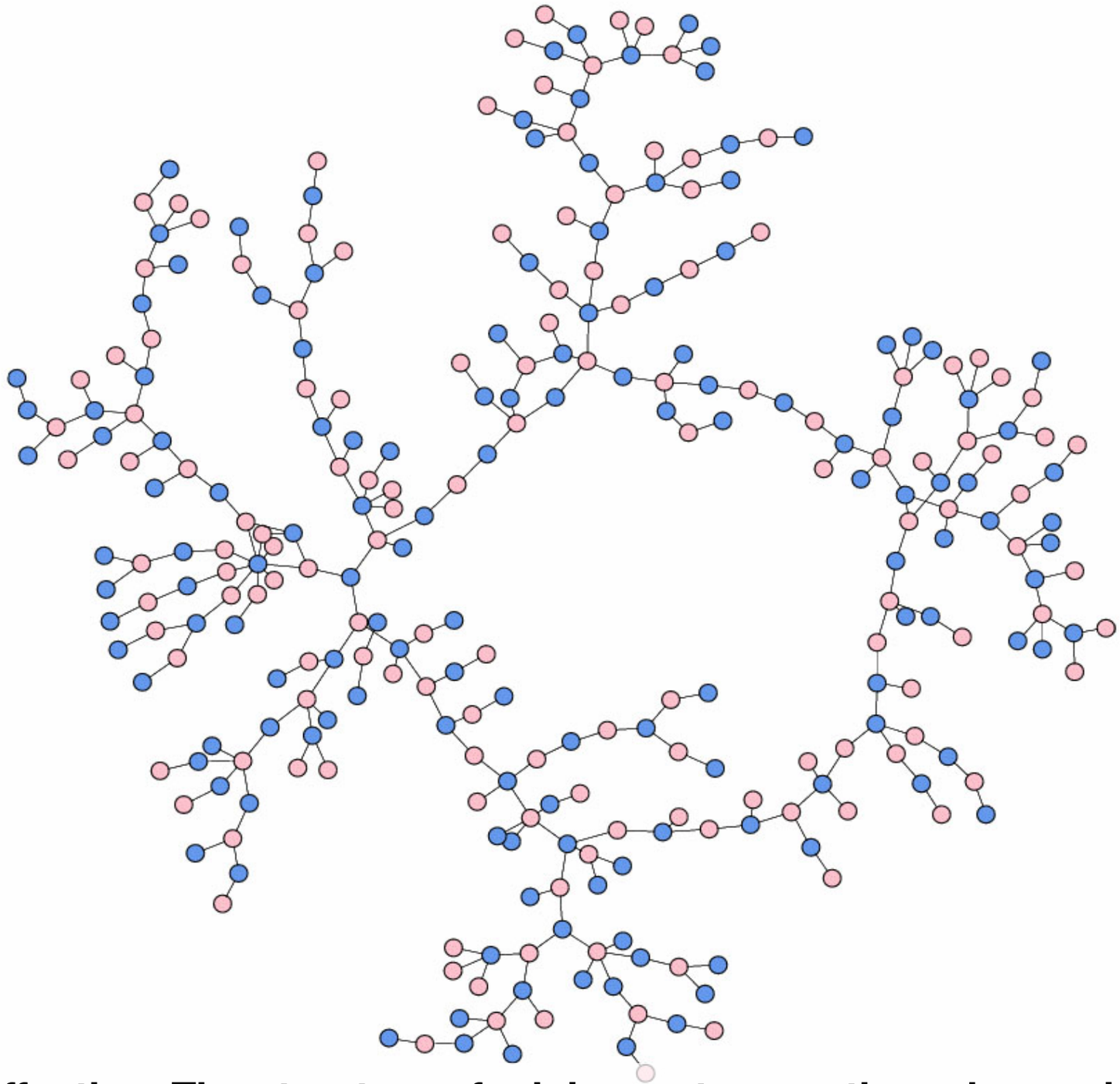
Force Directed Layout

- No intrinsic layering, what other options?
- Physics model:
  - Edges = springs
  - Nodes = repulsive particles



Aesthetically Pleasing?

high school dating network



Data from Bearman et al., Chains of affection: The structure of adolescent romantic and sexual networks, 2004. Image from http://www-personal.umich.edu/~mejn/networks/

Force Model

- Many variations, but usually physical analogy:
  - Repulsion between any pair of vertices v_i and v_j:

$$f_V(d_{ij}) = C_V \times \frac{m_i m_j}{d_{ij}}$$

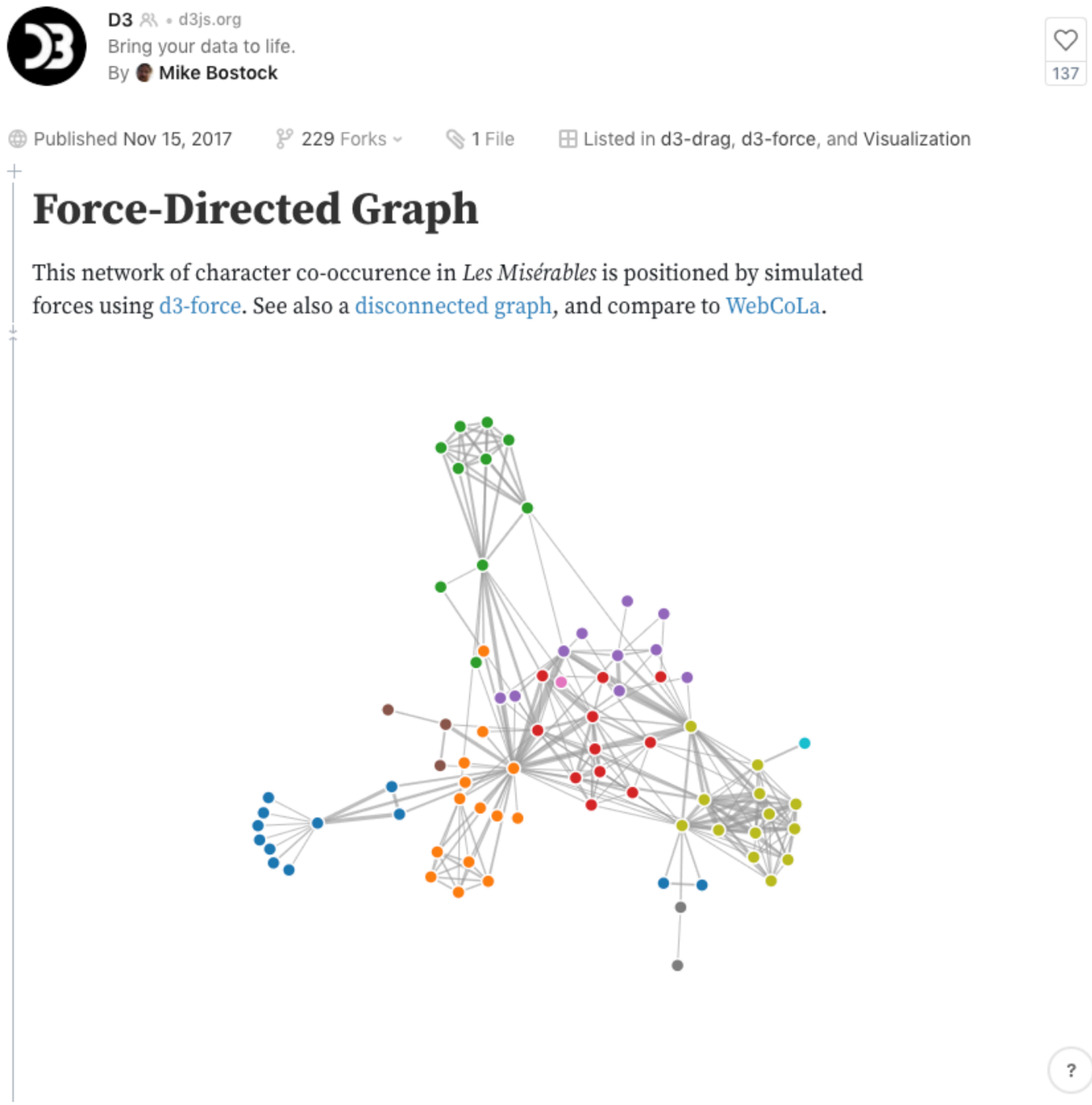
  - Attraction (i.e. Hooke's Law) for any edge e_{ij} between vertices v_i and v_j:

$$f_E(d_{ij}) = C_E \times (d_{ij} - L)$$

- Total force on a node v_i:

$$\sum_{v_j \in V, i \neq j} f_V(d_{ij}) + \sum_{e_{ij} \in E} f_E(d_{ij})$$

Force-Directed Layout



- Use forces to compute a direction of movement and then move a small amount in that direction
- Iterate until convergence
- Repulsion loop is $O(n^2)$ per iteration
    - Faster algorithms exist, Barnes-Hut, multipole methods, etc.

https://observablehq.com/@d3/force-directed-graph

vizster

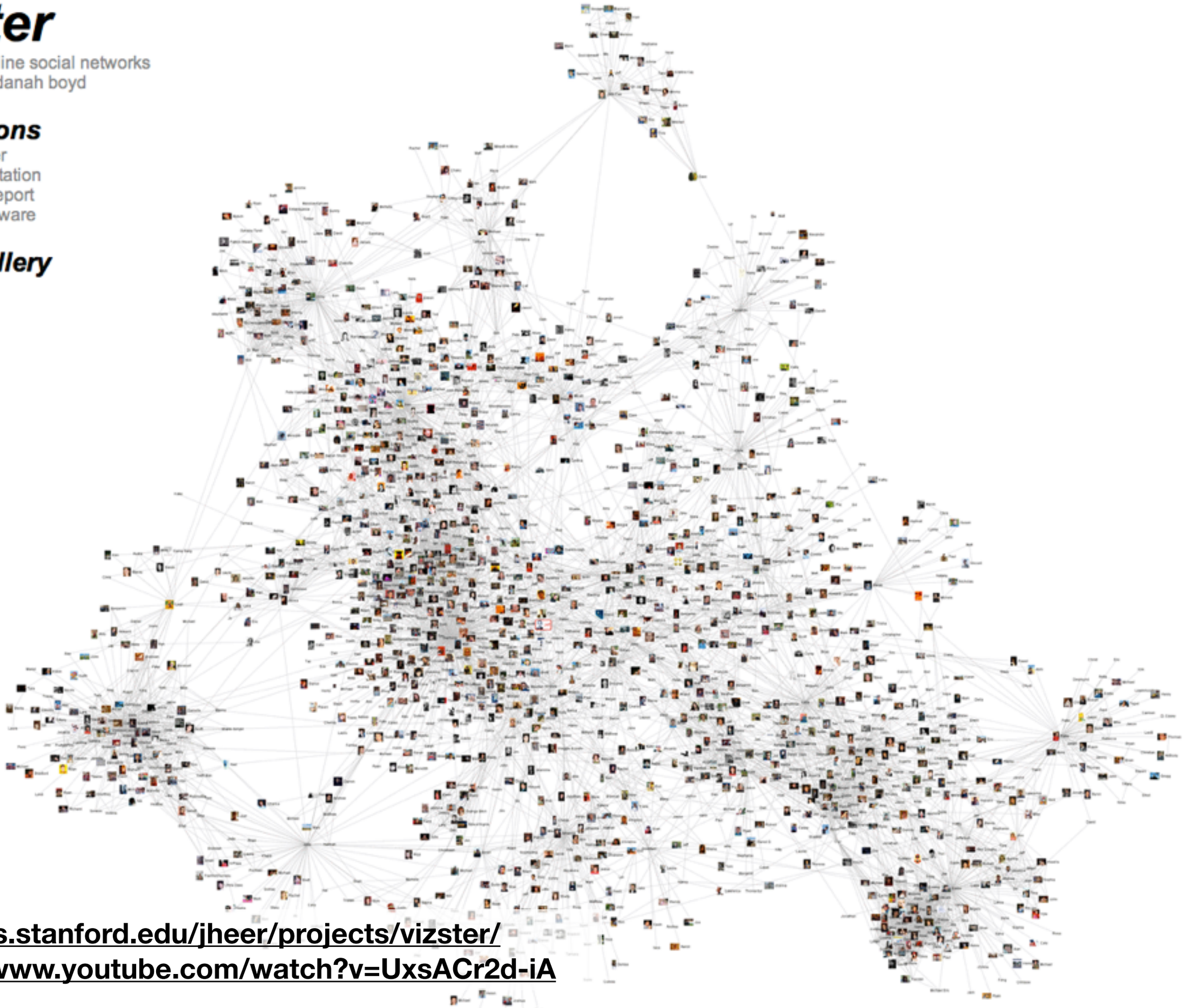
visualizing online social networks
jeffrey heer + danah boyd

publications

research paper
video demonstation
early design report
download software

photo gallery

egocentric
community
linkage
search
x-ray 1
x-ray 2



http://vis.stanford.edu/jheer/projects/vizster/
https://www.youtube.com/watch?v=UxsACr2d-iA

HOLA: Human-like Orthogonal Network Layout

Steve Kieffer, Tim Dwyer, Kim Marriott, and Michael Wybrow

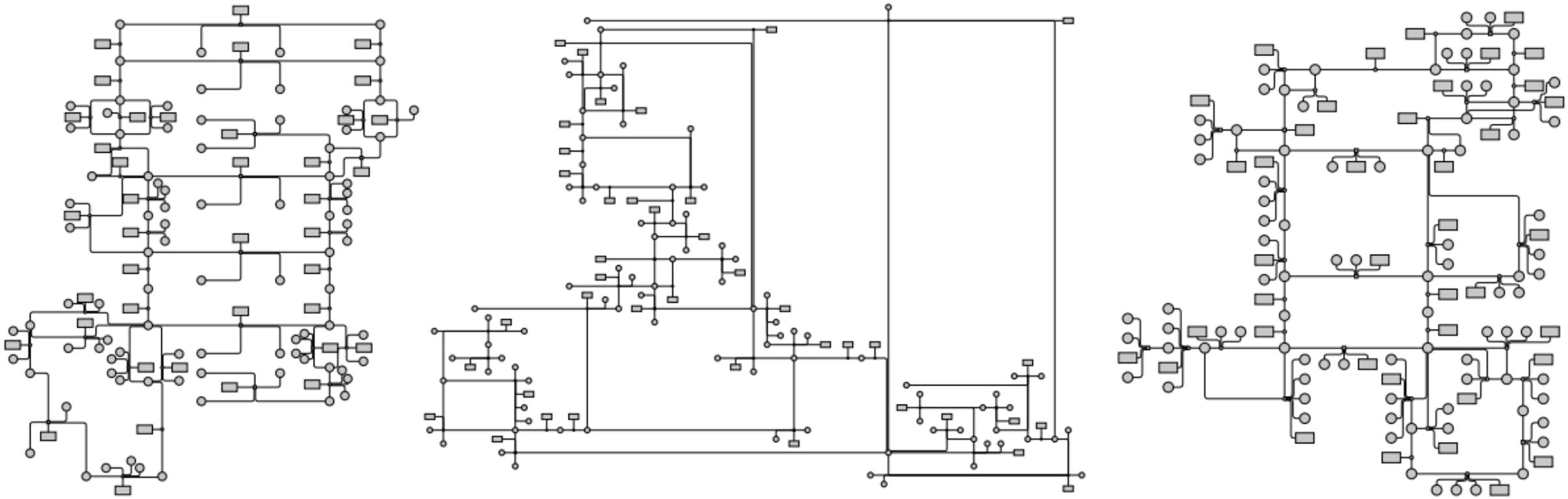
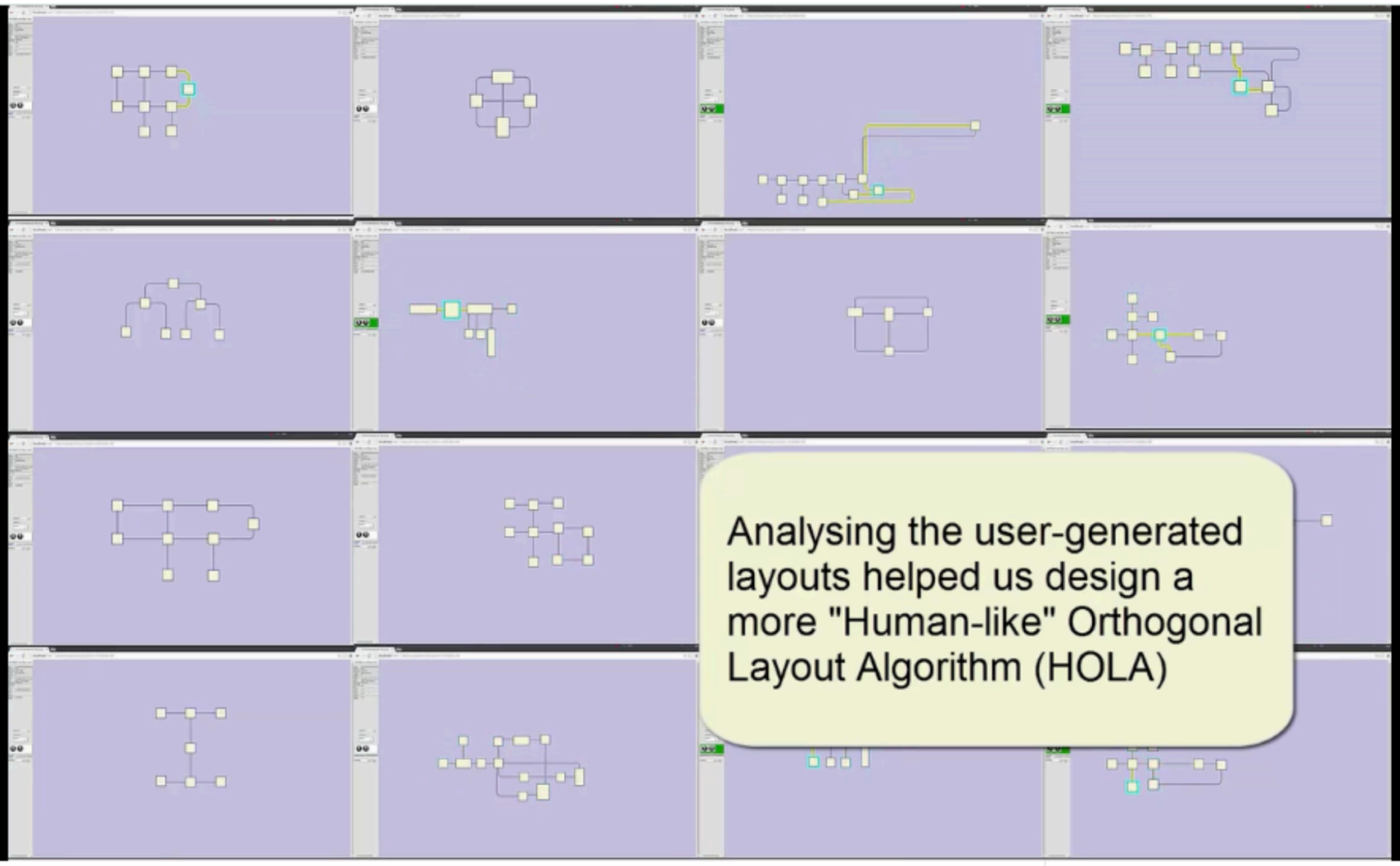


Fig. 1: Human, yFiles, and HOLA layouts of SBGN Glycolysis-Glygoneogensis pathway. It is clear that the human and the state-of-the-art layout algorithm from yFiles produce structurally quite different layouts. In this paper we explore the reasons why humans arrange such orthogonal network diagrams differently to standard algorithms and use this to inform the design of a new algorithm, HOLA (output shown right-most), that aims to produce more "human-like" layout.



HOLA: Human-like Orthogonal Layout Algorithm

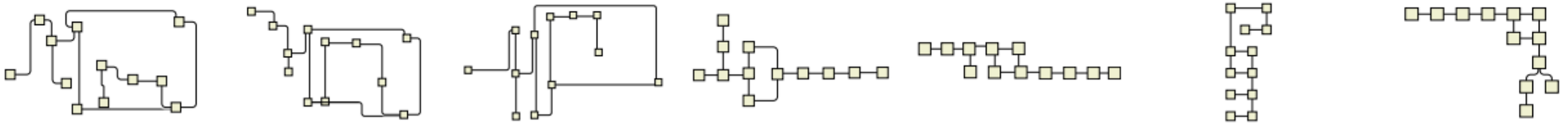
7 years ago

Steve Kieffer + Follow

549 0 0 0

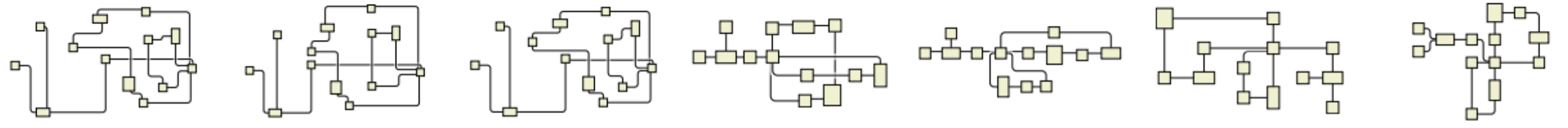
Download Share

Graph 4



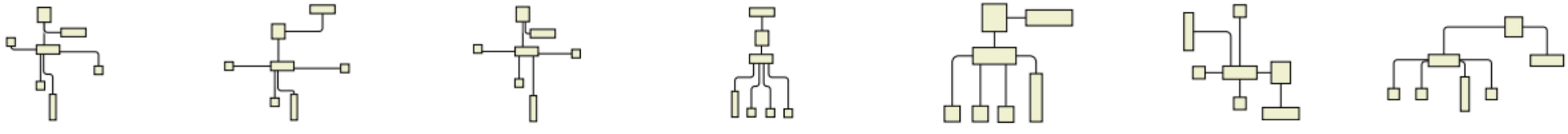
$\mu_1 = 0.00$ | $\mu_1 = 0.00$ | $\mu_1 = 0.00$ | $\mu_1 = 0.58$ | $\boxed{\mu_1 = 0.59}$, $\mu_2 = 0.59$ | $\mu_1 = 0.21$, $\mu_2 = 0.11$ | $\boxed{\mu_2 = 0.80}$

Graph 5



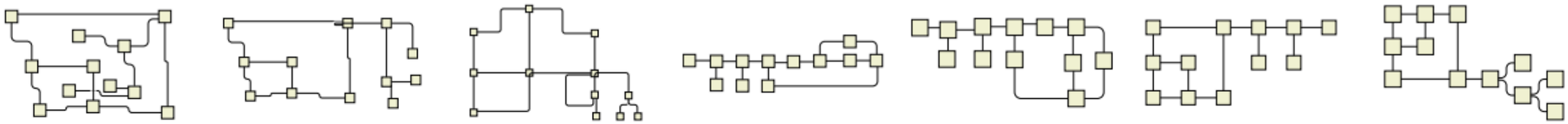
$\mu_1 = 0.00$ | $\mu_1 = 0.00$ | $\mu_1 = 0.07$ | $\mu_1 = 0.60$ | $\boxed{\mu_1 = 0.63}$, $\boxed{\mu_2 = 0.70}$ | $\boxed{\mu_1 = 0.63}$, $\mu_2 = 0.38$ | $\mu_2 = 0.42$

Graph 6



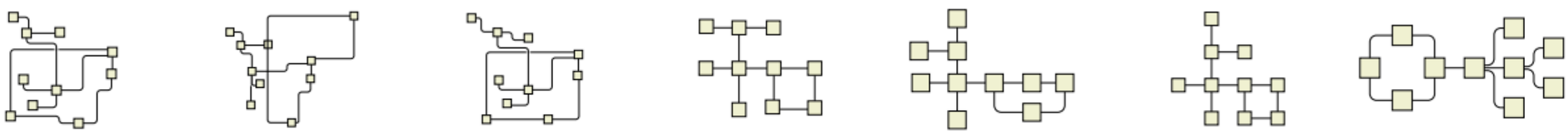
$\bar{\mu}_1 = 0.02$ | $\bar{\mu}_1 = 0.05$ | $\bar{\mu}_1 = 0.02$ | $\bar{\mu}_1 = 0.55$ | $\boxed{\bar{\mu}_1 = 0.66}$, $\boxed{\bar{\mu}_2 = 0.83}$ | $\bar{\mu}_1 = 0.15$, $\bar{\mu}_2 = 0.14$ | $\bar{\mu}_2 = 0.53$

Graph 7



$\bar{\mu}_1 = 0.00$ | $\bar{\mu}_1 = 0.00$ | $\bar{\mu}_1 = 0.03$ | $\bar{\mu}_1 = 0.63$ | $\boxed{\bar{\mu}_1 = 0.64}$, $\bar{\mu}_2 = 0.52$ | $\bar{\mu}_1 = 0.63$, $\bar{\mu}_2 = 0.42$ | $\boxed{\bar{\mu}_2 = 0.56}$

Graph 8

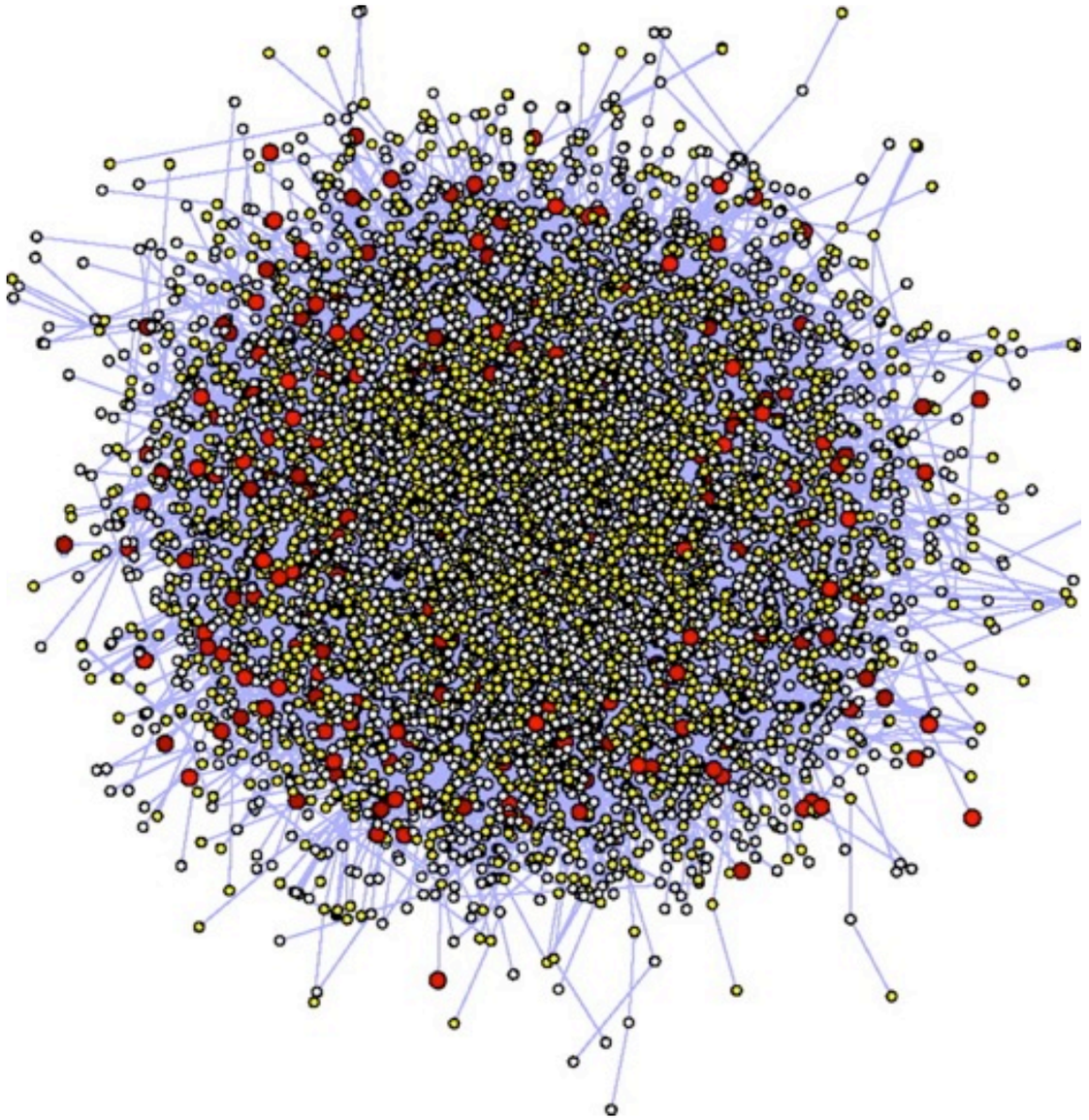


$\bar{\mu}_1 = 0.00$ | $\bar{\mu}_1 = 0.00$ | $\bar{\mu}_1 = 0.02$ | $\bar{\mu}_1 = 0.61$ | $\boxed{\bar{\mu}_1 = 0.68}$, $\bar{\mu}_2 = 0.45$ | $\bar{\mu}_1 = 0.61$, $\bar{\mu}_2 = 0.42$ | $\boxed{\bar{\mu}_2 = 0.64}$

Fig. 3: The 8 graphs and some of their layouts from the manual layout of orthogonal graphs study. At left is the initial layout. The next 4 columns show the two worst and the two best manual layout. The two final columns show automatic layout from yFiles and our proposed algorithm HOLA. $\bar{\mu}_1$ = normalised inverted mean rank for first study, $\bar{\mu}_2$ = that for second study. Best possible value is 1, worst possible 0. Means in boxes indicate best actual mean rank for that contest.

Node-Link Layouts

- Strengths:
  - Understandable visual mapping
  - Can show overall structure, clusters, paths
  - Flexible, many variations
- Limitations:
  - All but the most trivial algorithms are > $O(N^2)$
  - Not good for dense graphs: Hairball problem!
  - Small changes in the graph can cause dramatic changes to the layout
    - See Frishman and Tal. Online Dynamic Graph Drawing. Proc EuroVis 2007



Gephi, an open source graph visualization and manipulation software

http://gephi.org/ RSS hookes law

Apple Yahoo! Google Maps YouTube Wikipedia News (1,949) Popular Google Scholar

Gephi
makes graphs handy

Download Blog Store Wiki Forum Support Bug tracker

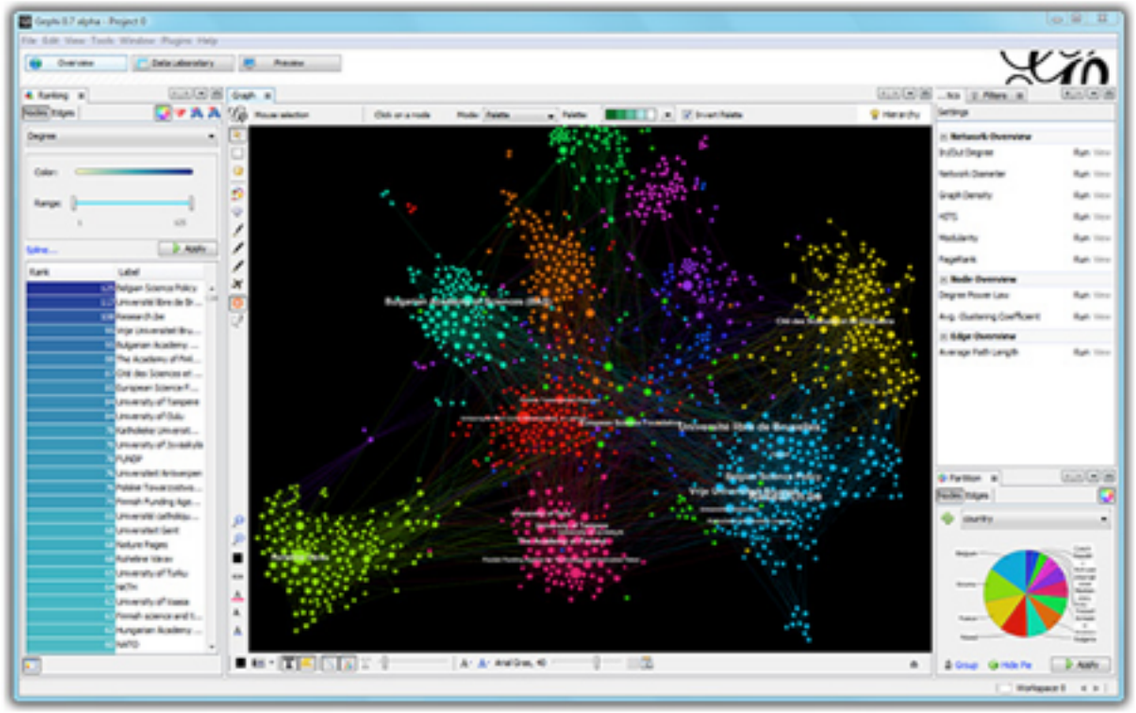
Home Features Learn Develop Plugins Consortium

The Open Graph Viz Platform

Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs.

Runs on Windows, Linux and Mac OS X. Gephi is open-source and free.

Learn More on Gephi Platform »



Download FREE
Gephi 0.8 beta

Release Notes | System Requirements

- Features
- Quick start
- Screenshots
- Videos

Gephi 0.8 beta has been released! Discover a new Preview and dynamic features, start building commercial applications with the new open source license.

Learn More »

APPLICATIONS

- **Exploratory Data Analysis**: intuition-oriented analysis by networks manipulations in real time.
- **Link Analysis**: revealing the underlying structures of associations between objects, in particular in scale-free networks.
- **Social Network Analysis**: easy creation of social data connectors to map community organizations and small-world networks.
- **Biological Network analysis**: representing patterns of biological data.
- **Poster creation**: scientific work promotion with hi-quality printable maps.

Learn More »

“Like Photoshop™ for graphs.

— the Community

LATEST NEWS

- Weekly news
  February 27, 2012
- Annual report 2011
  February 25, 2012
- Gephi-Neo4j presentation at FOSDEM
  February 20, 2012
- Gephi meet-up #4 in Berlin
  February 2, 2012
- Introducing the Gephi Plugins Bootcamp
  January 12, 2012

See All »

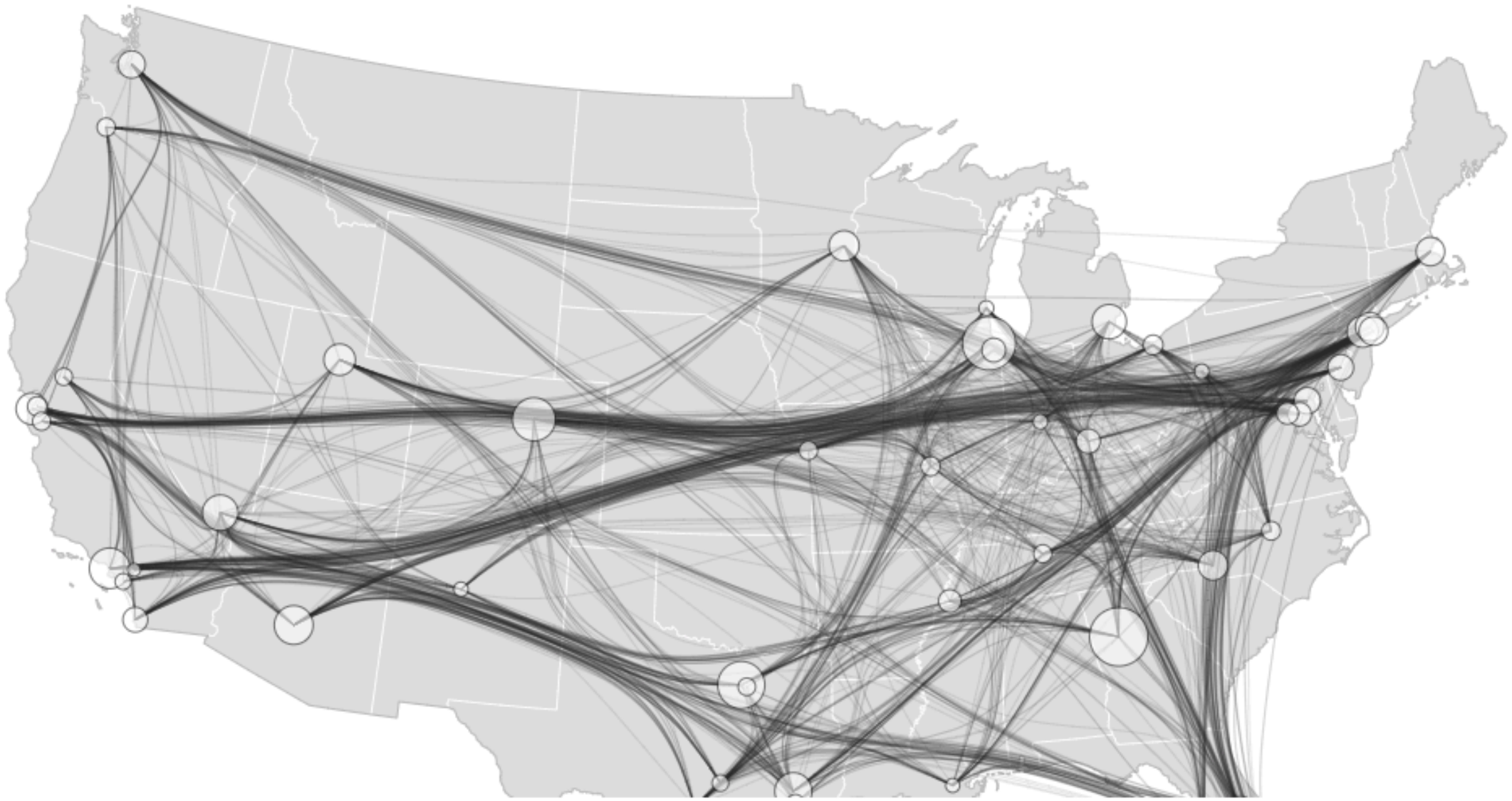
PAPERS

Alternatives for Visualizing Graphs

Edge Aggregation

- Problem with node-link: many crossing edges make it difficult to see the graph structure
    - May be impossible to remove unless the graph is planar
- Proposed solution: allow groups of edges to be merged and drawn together, decreasing clutter

Flight Paths Edge Bundling



Visualizes flights between airports in the continental United States using edge bundling. The code can be easily modified to either show the top 50 airports by degree or the highest degree airport in each state.

Open

This example combines our map and graph visualizations together in a single visualization. It demonstrates map projections, TopoJSON, Voronoi diagrams, force-directed layouts, and edge bundling.

IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 12, NO. 5, SEPTEMBER/OCTOBER 2006

Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data

Danny Holten

Abstract—A compound graph is a frequently encountered type of data set. Relations are given between items, and a hierarchy is defined on the items as well. We present a new method for visualizing such compound graphs. Our approach is based on visually bundling the adjacency edges, i.e., non-hierarchical edges, together. We realize this as follows. We assume that the hierarchy is shown via a standard tree visualization method. Next, we bend each adjacency edge, modeled as a B-spline curve, toward the polyline defined by the path via the inclusion edges from one node to another. This hierarchical bundling reduces visual clutter and also visualizes implicit adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes. Furthermore, hierarchical edge bundling is a generic method which can be used in conjunction with existing tree visualization techniques. We illustrate our technique by providing example visualizations and discuss the results based on an informal evaluation provided by potential users of such visualizations.

Index Terms—Network visualization, edge bundling, edge aggregation, edge concentration, curves, graph visualization, tree visualization, node-link diagrams, hierarchies, treemaps.

1 Introduction

There is a large class of data sets that contain both hierarchical components, i.e., parent-child relations between data items, as well as non-hierarchical components representing additional relations between data items. Parent-child relations are henceforth called *inclusion* relations, whereas additional, non-hierarchical relations are henceforth called *adjacency* relations. Some examples of such data sets are:

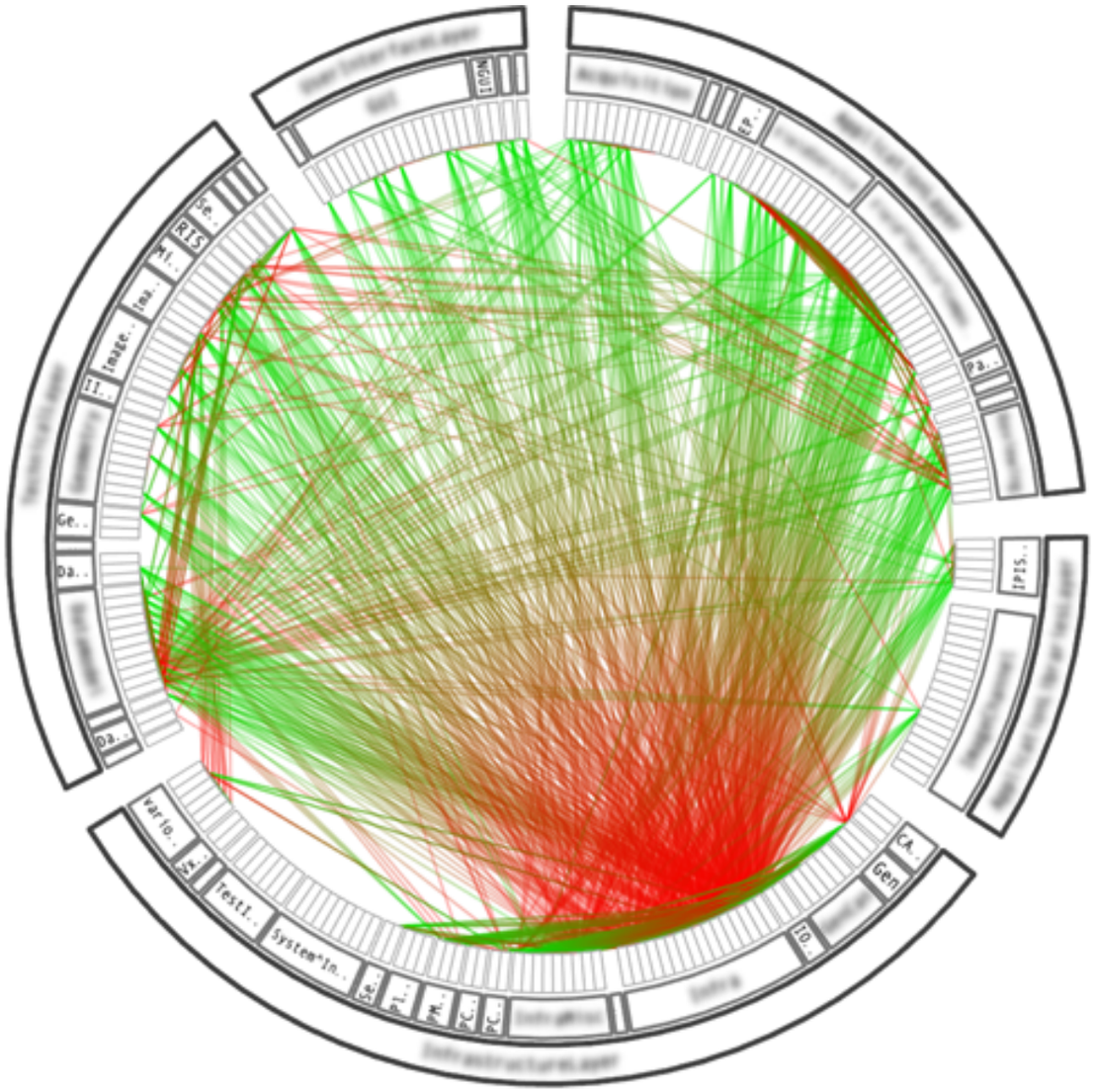
- A hierarchically organized software system, e.g., source code divided into directories, files, and classes, and the relations between these elements, for instance, dependency relations;
- Social networks comprised of individuals at the lowest level of the hierarchy and groups of individuals at higher levels of the hierarchy. Relations could indicate if (groups of) people are ac-

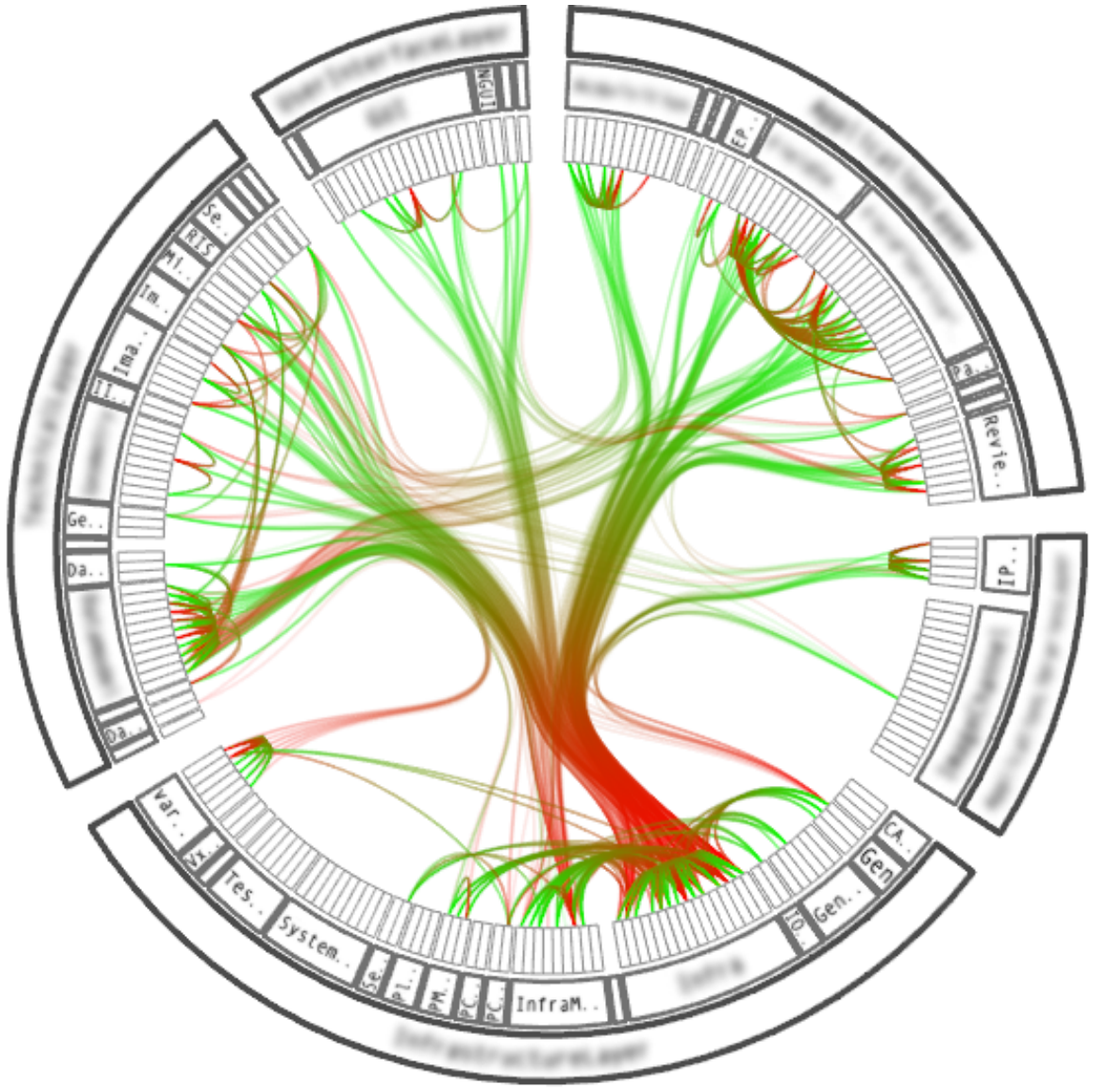
and adjacency edges become intertwined, which can make it difficult to visually separate both types of edges from each other.

At present, only few techniques are available that are specifically designed to display adjacency relations on top of a tree structure, as is also mentioned by Neumann et al. [20]. Hence, the focus of this paper is on the construction of a generic technique for the visualization of compound graphs and compound directed graphs (digraphs) comprised of a tree and an additional (directed) adjacency graph.

We present hierarchical edge bundles, as described below, for the visualization of compound (di)graphs. Hierarchical edge bundling is based on the principle of visually bundling adjacency edges together analogous to the way electrical wires and network cables are merged into bundles along their joint paths and fanned out again at the end, in order to make an otherwise tangled web of wires and cables more manageable. The main features of the proposed technique are as fol-

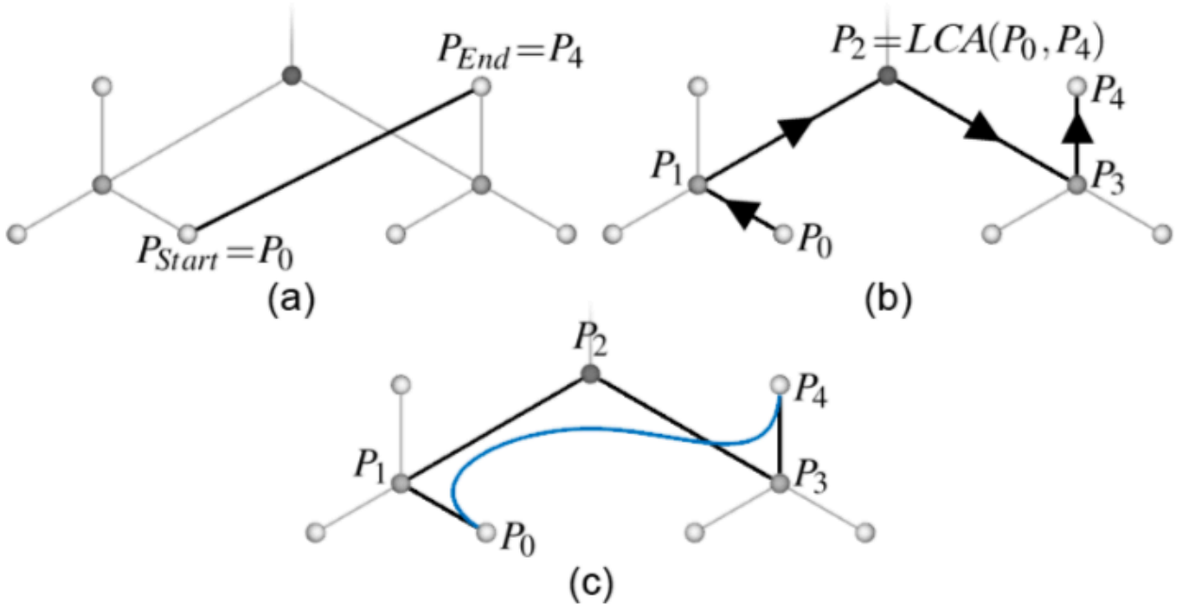
Reducing Clutter

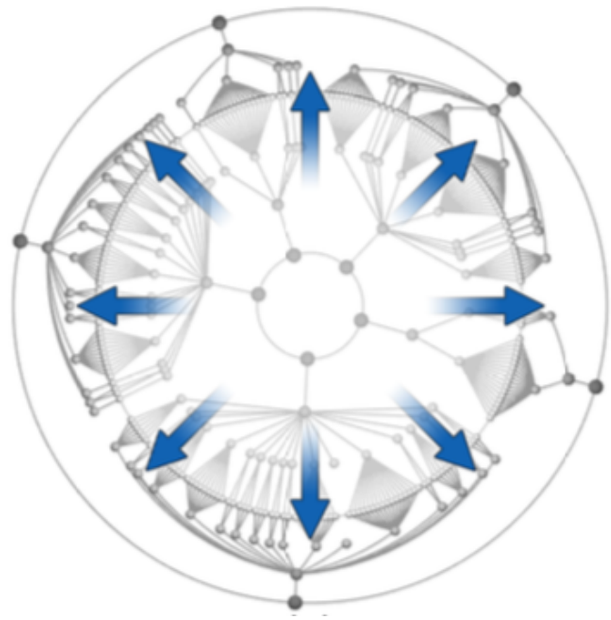


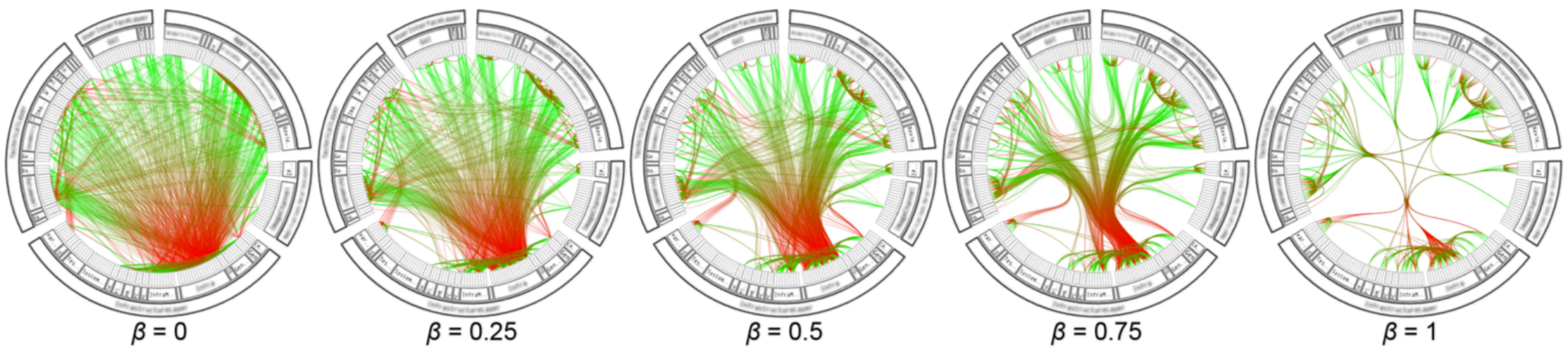


Holten, 2006

Hierarchical Edge Bundling

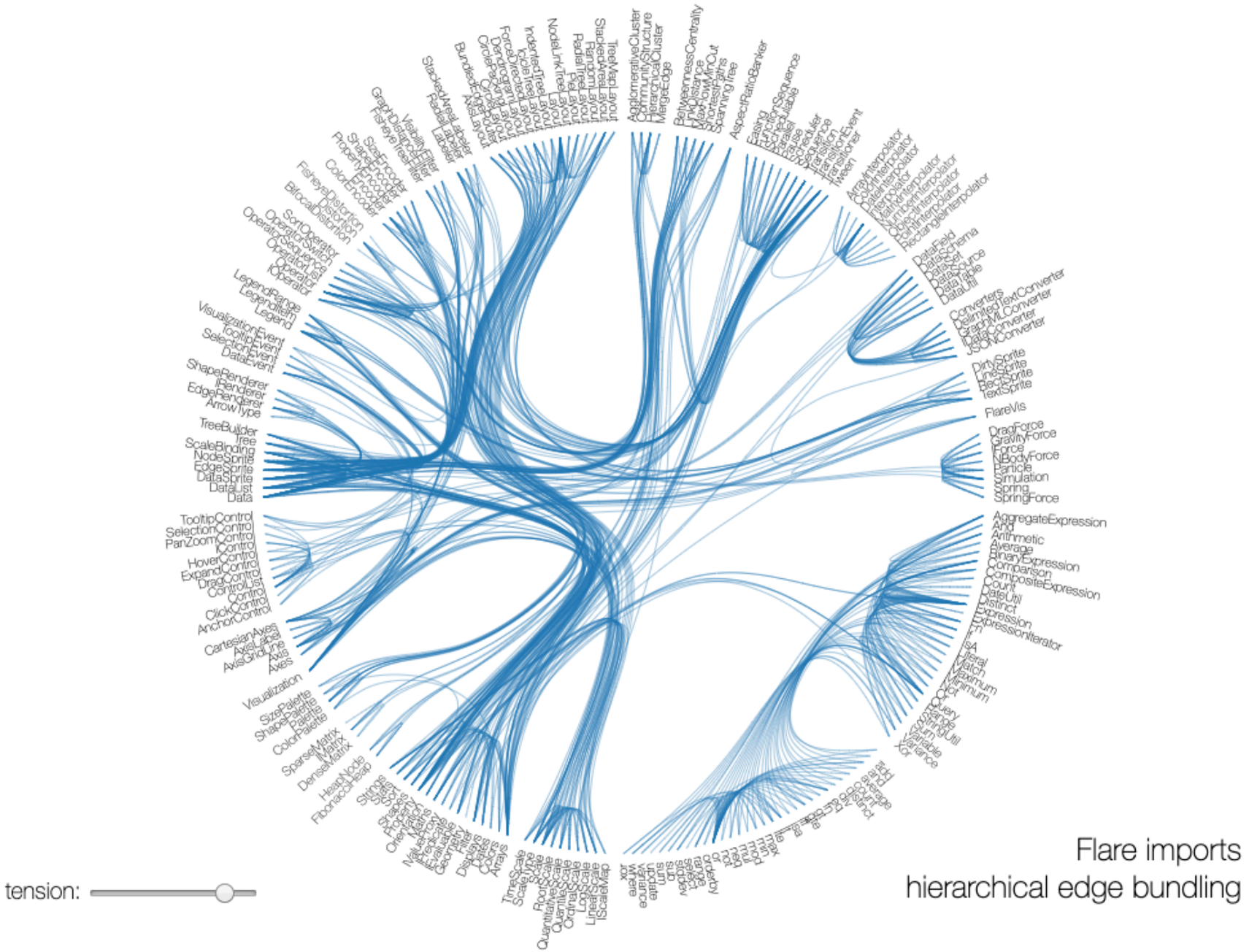






Holten, 2006

Bundling Strength Demo



http://mbostock.github.io/d3/talk/20111116/bundle.html

Critique

d3-graph-gallery.com/bundle.html

← D3.js Graph Gallery

CHART TYPES QUICK ALL R PYTHON DATA TO VIZ WHO AM I ABOUT

Hierarchical edge bundling

Hierarchical edge bundling allows to visualize adjacency relations between entities organized in a hierarchy. The idea is to bundle the adjacency edges together to decrease the clutter usually observed in complex networks. Learn more about this kind of chart in data-to-viz.com, or visit the examples below to implement it in d3.js.

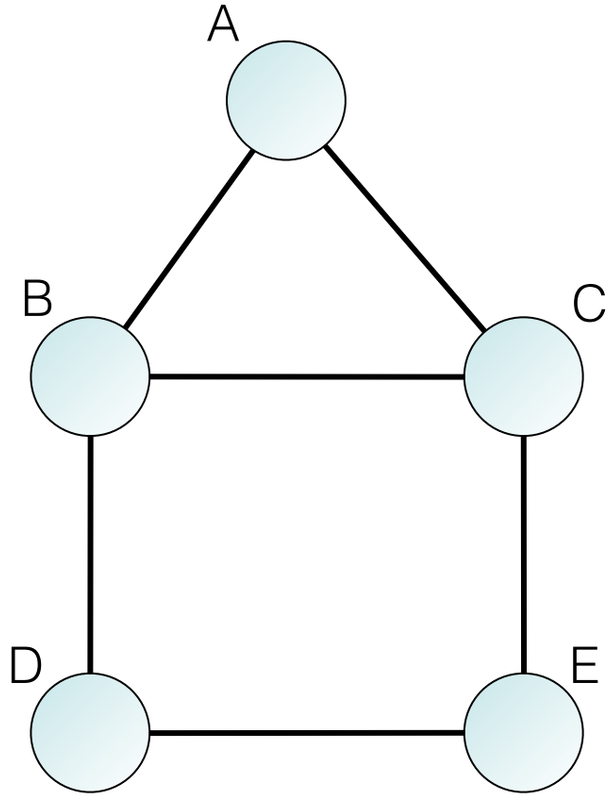
STEP BY STEP

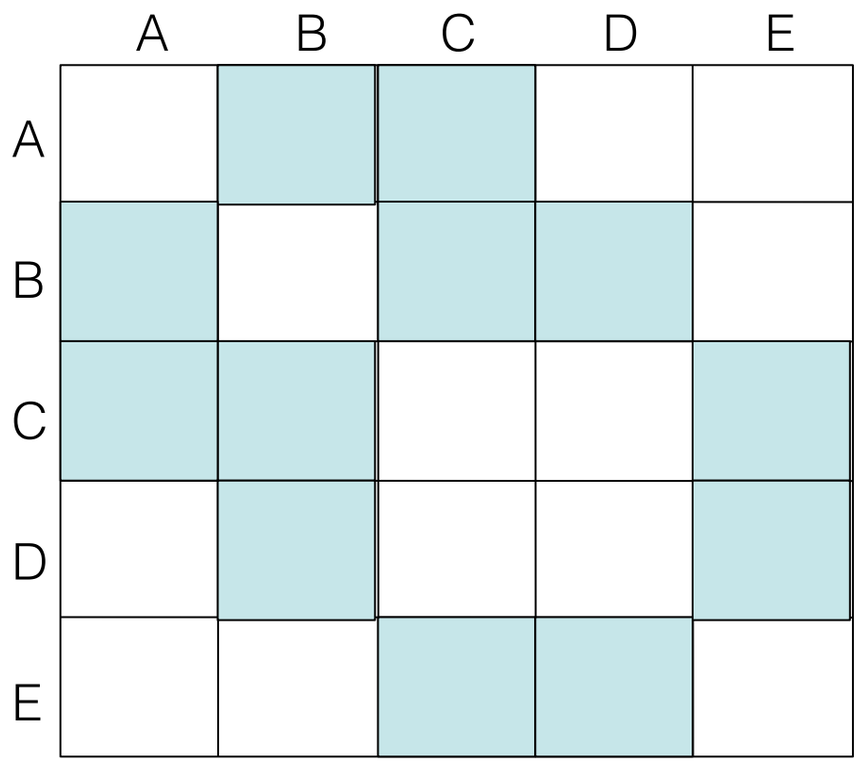
Before trying to implement hierarchical edge bundling, it is crucial to know how this kind of chart works. It is crucial to understand that input data is composed by i/ a hierarchy and ii/ a list of links between elements of the hierarchy.

https://www.d3-graph-gallery.com/bundle.html

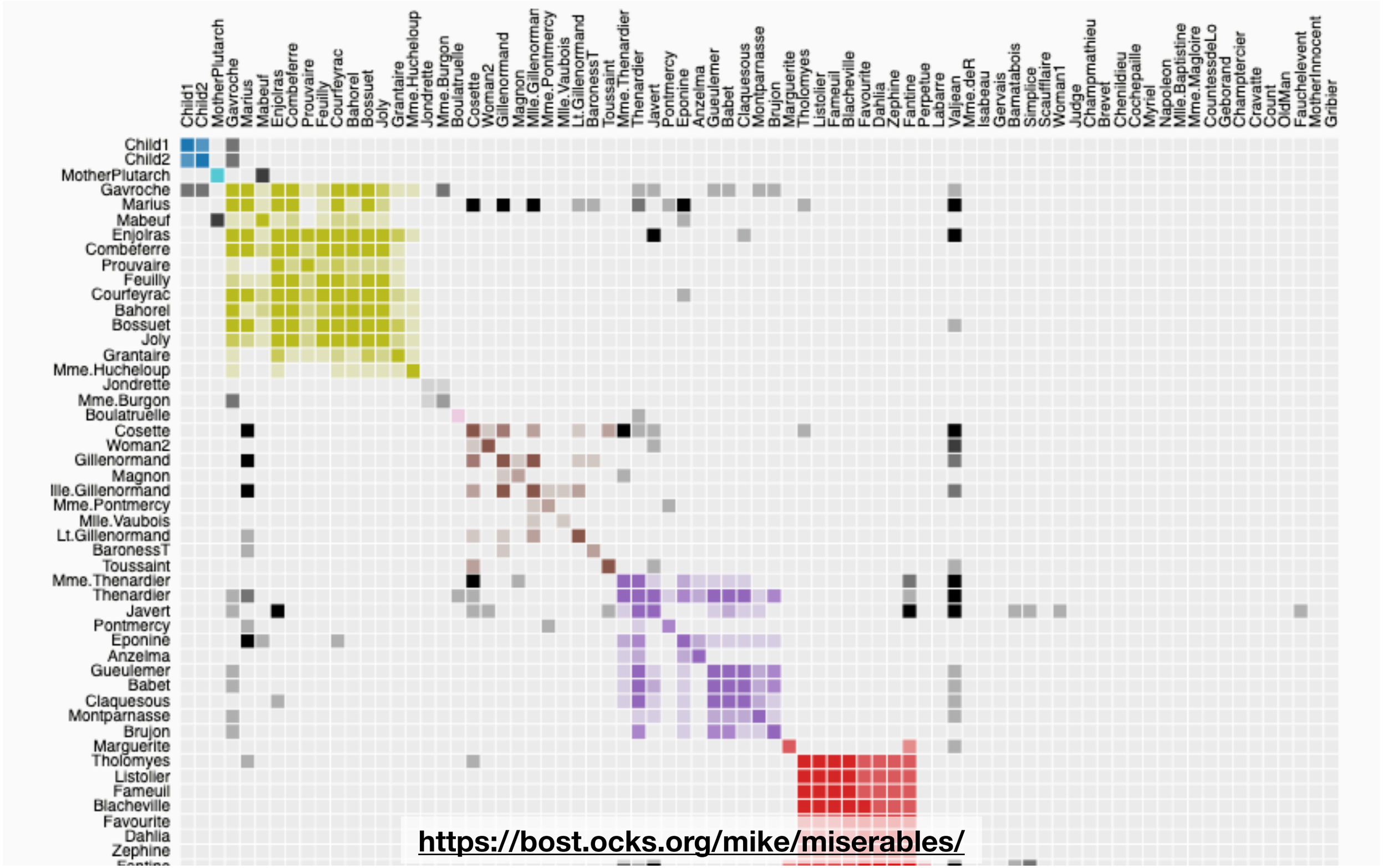
Matrix Layouts

- Instead of node-link diagrams, use the adjacency matrix to represent



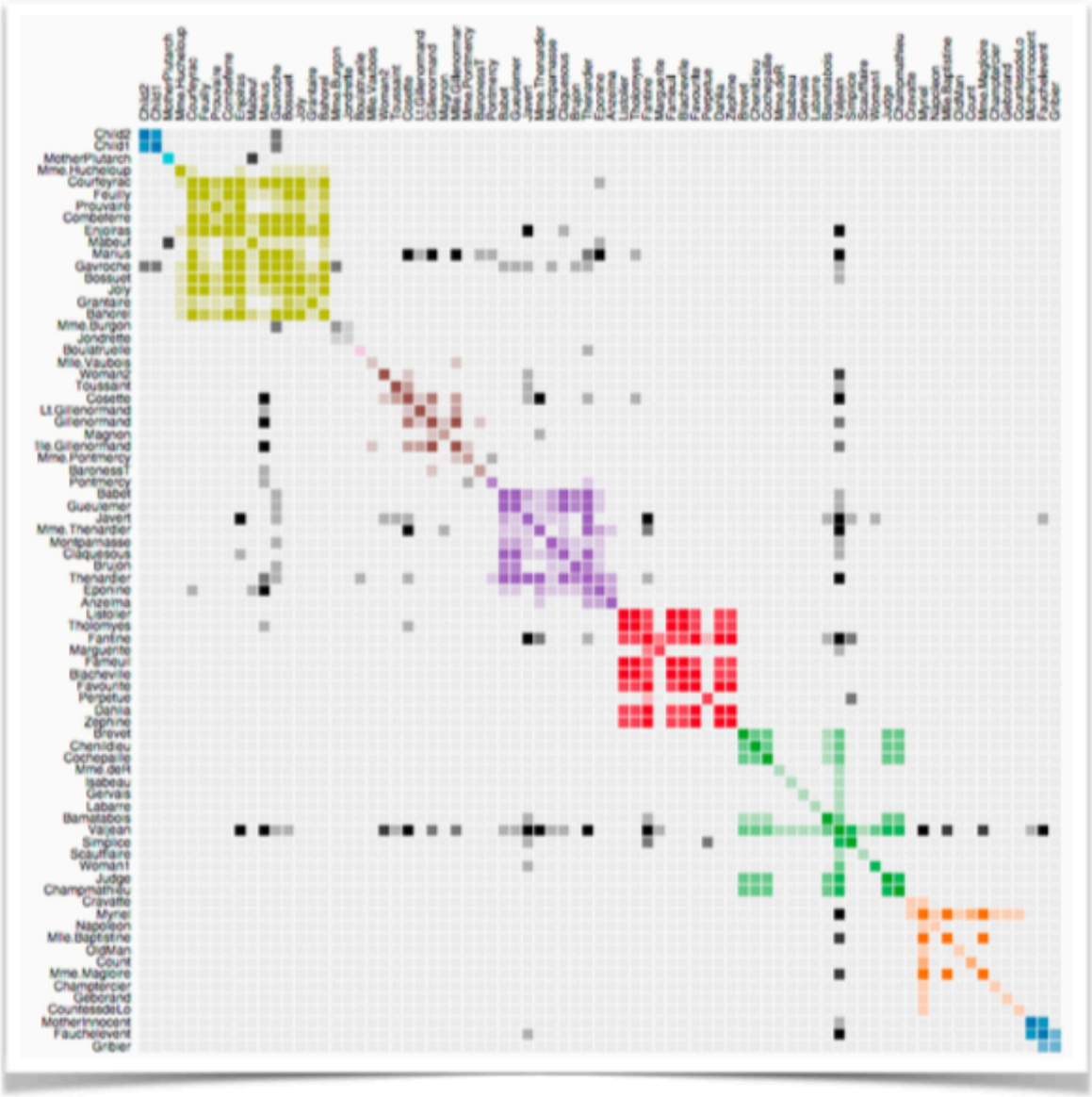


Matrix Diagrams

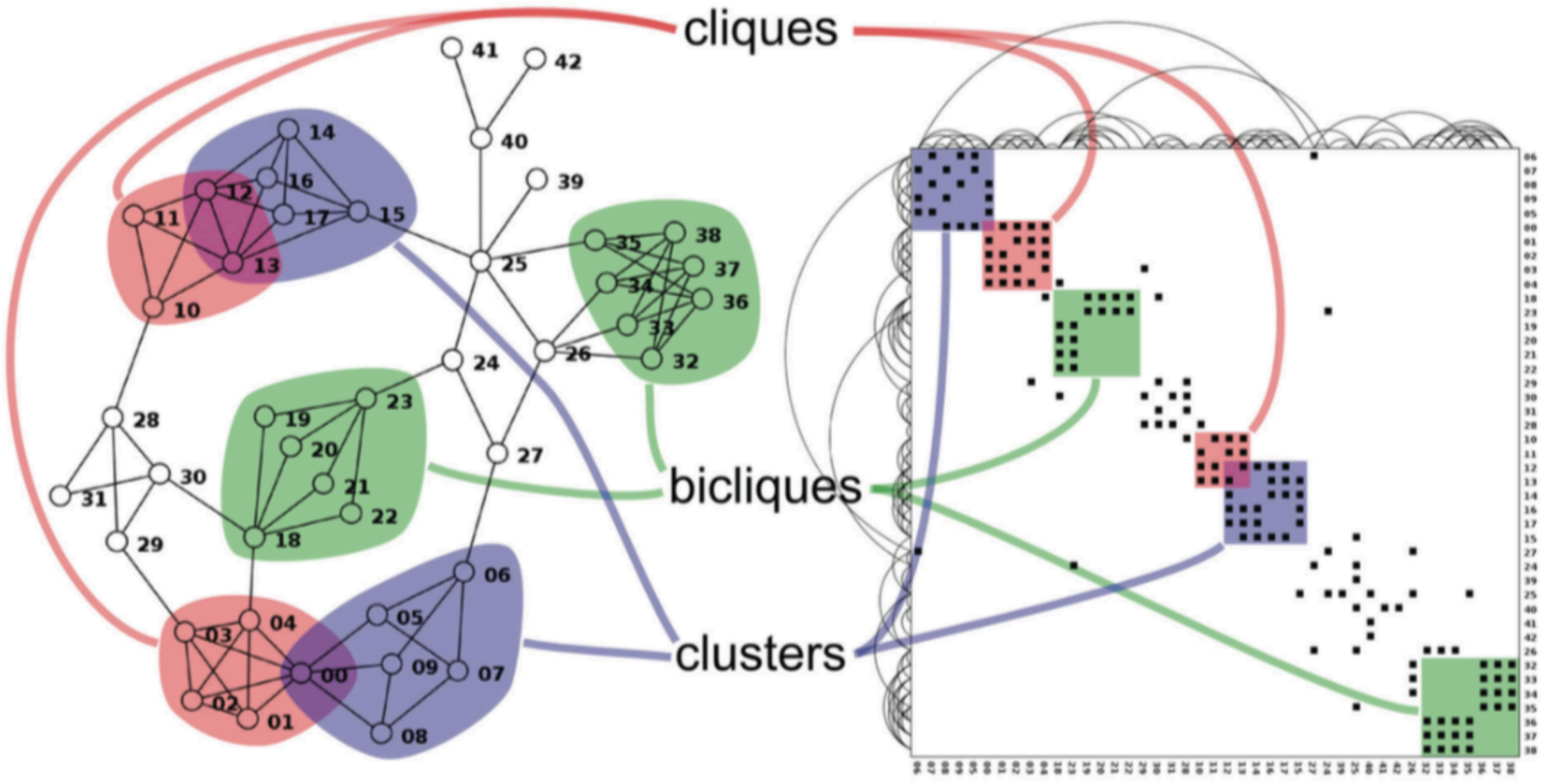


https://bost.ocks.org/mike/miserables/

Row Ordering Has a Dramatic Effect



Spotting Patterns in Matrices



McGuffin 2012

Treemap | Tree Node-Link | Table Matrix | Table Scatter Plot | Table Time Series | Table Parallel Coordinates

About | Graph Matrix | Graph Node-Link | Tree Icicle

Fisheyes | Rulers

Visual | Excentric

Detail | Filters

Label

Column

count (Integer)

Label all items

Size

Column

(None)

Default

0=Fit Labels 50

Background

Color | gradier

Sort

Column

(None)

Inverse Order

Label Vertex by

author (Ben Shneiderman, S. K. Card, J. Mackinlay, Allison Woodruff, S. F. Roth, Lucy Nowell, enjamin B. Bederson, Roberto Tamassia, Marti Hearst, Michael Stonebraker, Harpreet Sawhney, Alexander Aiken, Daniel A. Keim, Robert K. France, Edward A. Fox, D. T. Lee, Peter Pirolli, Alan M. MacEachren, Deborah Hix, R. Rao, Jim Hafner, Monika Gorkani, Byron Dom, Qian Huang, Jonathan Ashley, Wayne Niblack, Peter Yanker, David Steele, Dragutin Petkovic, Myron Flickner, Ed Huai-hsin Chi, Michael T. Goodrich, J. T. Stasko, Dennis J. Brueni, James D. Hollan, Andries van Dam, Kaushai Dalal, S. G. Eick, George Otto, William C. Wake, Scott A. Guyer, Eric Labow, uillermo A. Averboch, Lenwood S. Heath, Galina Shubina, Robert Cohen, Ulrick Barandes, Luca Vismara, Mark Handy, Benoît Hudson, P. Lucas, Chris Olston, Daniel Haug, Ryan S. Baker, Natasha Gelfand, David S. Ebert, Beth Hetzler, George G. Robertson, Catherine Plaisant, Elizabeth Jurrus, Michael Christensen, Preben Mogensen, J. A. Senn, K. Palaniappan, M. Chalmers, Mac Holden, Dan Shapiro)

Sort Row by

degree (Integer)

Inverse Sort Row

Sort Column by

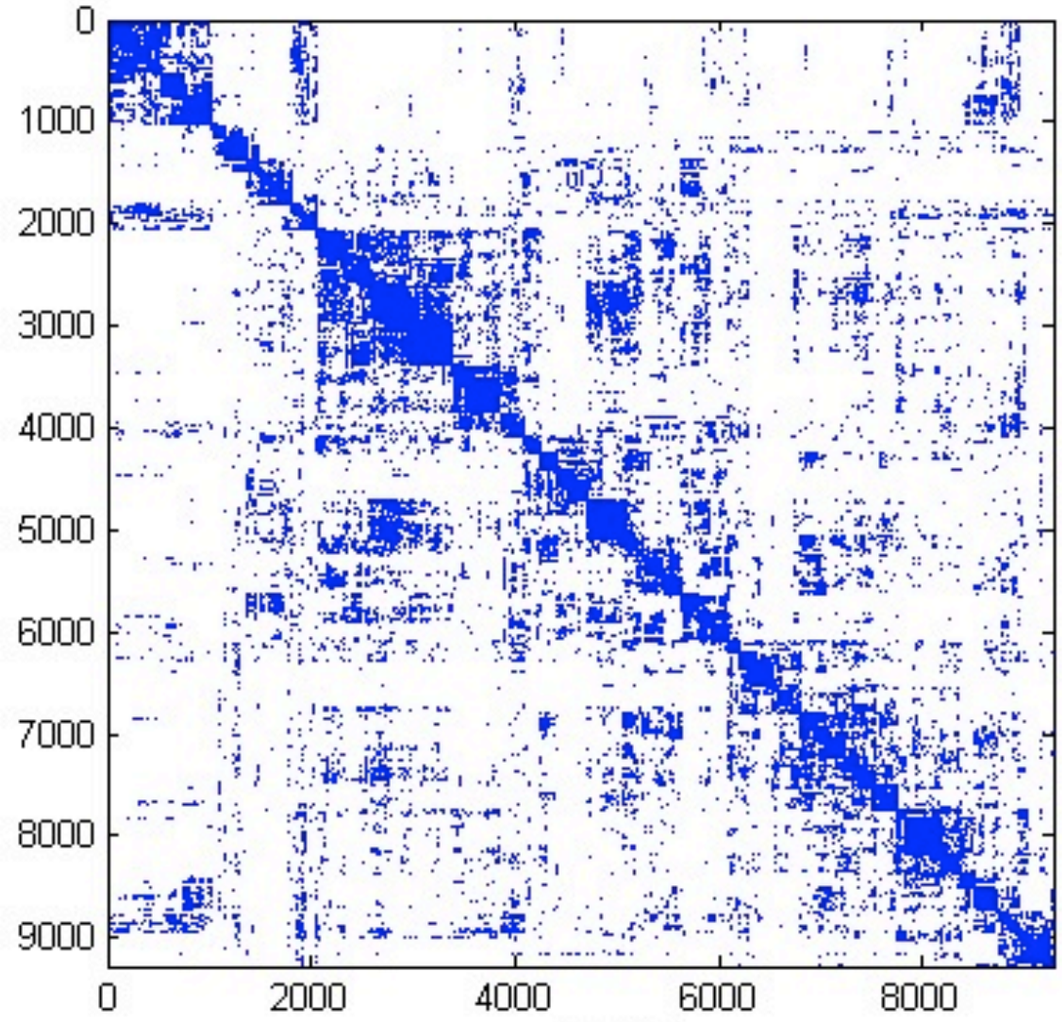
degree (Integer)

Inverse Sort Column

Adjacency Matrices

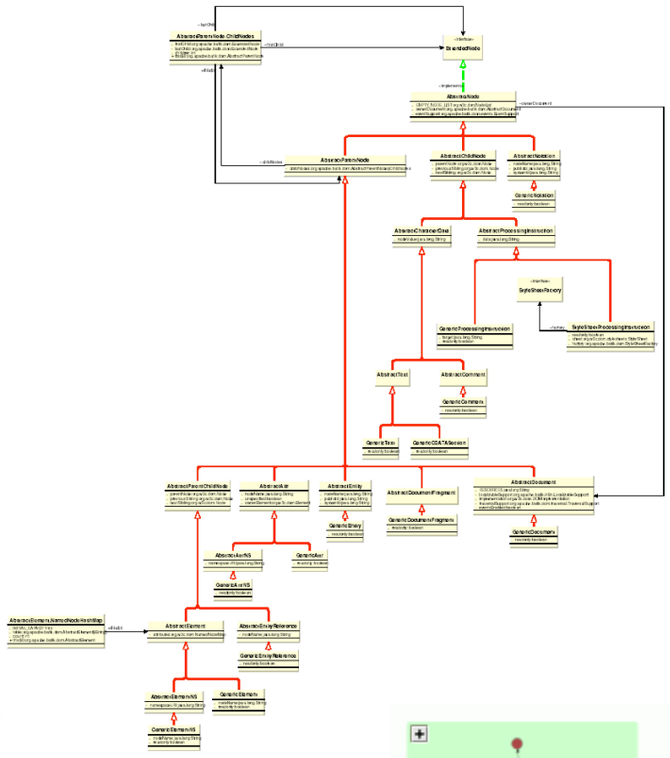
Matrix Representations

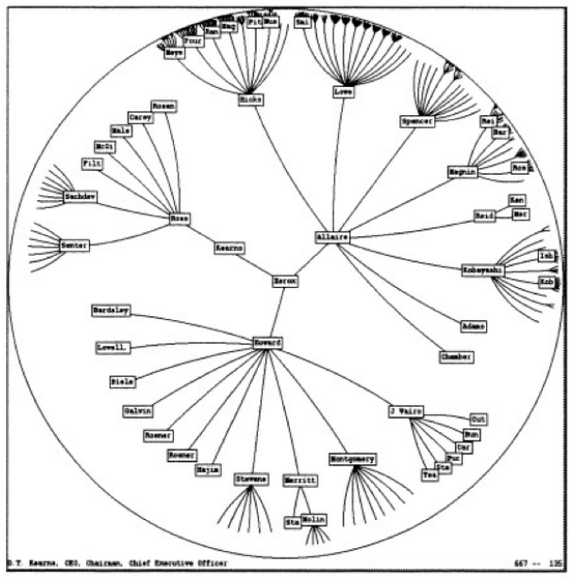
- Strengths
  - Great for dense graphs
  - Easy to compute
  - Easy to incorporate edge attributes
  - Visually scalable
  - Can spot clusters
- Limitations
  - Abstract visualization
  - Hard to follow paths
  - Row order has a big impact

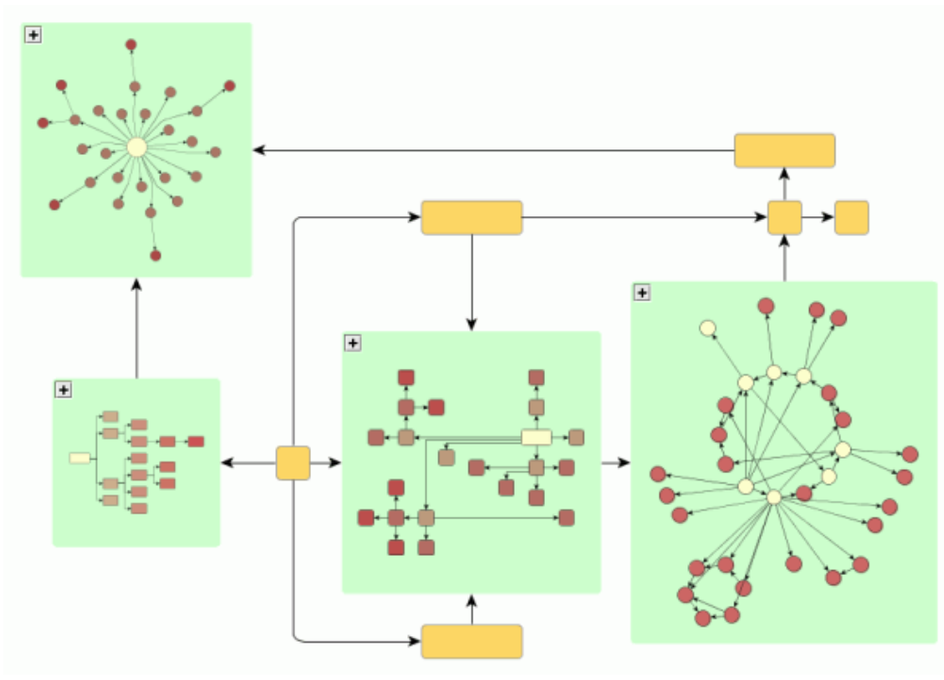


Other Layouts

- Orthogonal
  - Great for UML diagrams
  - Algorithmically complex
- Circular layouts
  - Emphasizes ring topologies
  - Used in social network diagrams
- Nested layouts
  - Recursively apply layout algorithms
  - Great for graphs with hierarchical structure







visualcomplexity.com | A visual exploration on mapping complex networks

http://www.visualcomplexity.com/vc/

Google

Apple Yahoo! Google Maps YouTube Wikipedia News (1,950) Popular Google Scholar

VC book is now available for purchase!

Buy at Amazon.com See other retailers

Home About VC Book Stats Blog Books Links Contact

Subscribe to latest projects: Follow on:

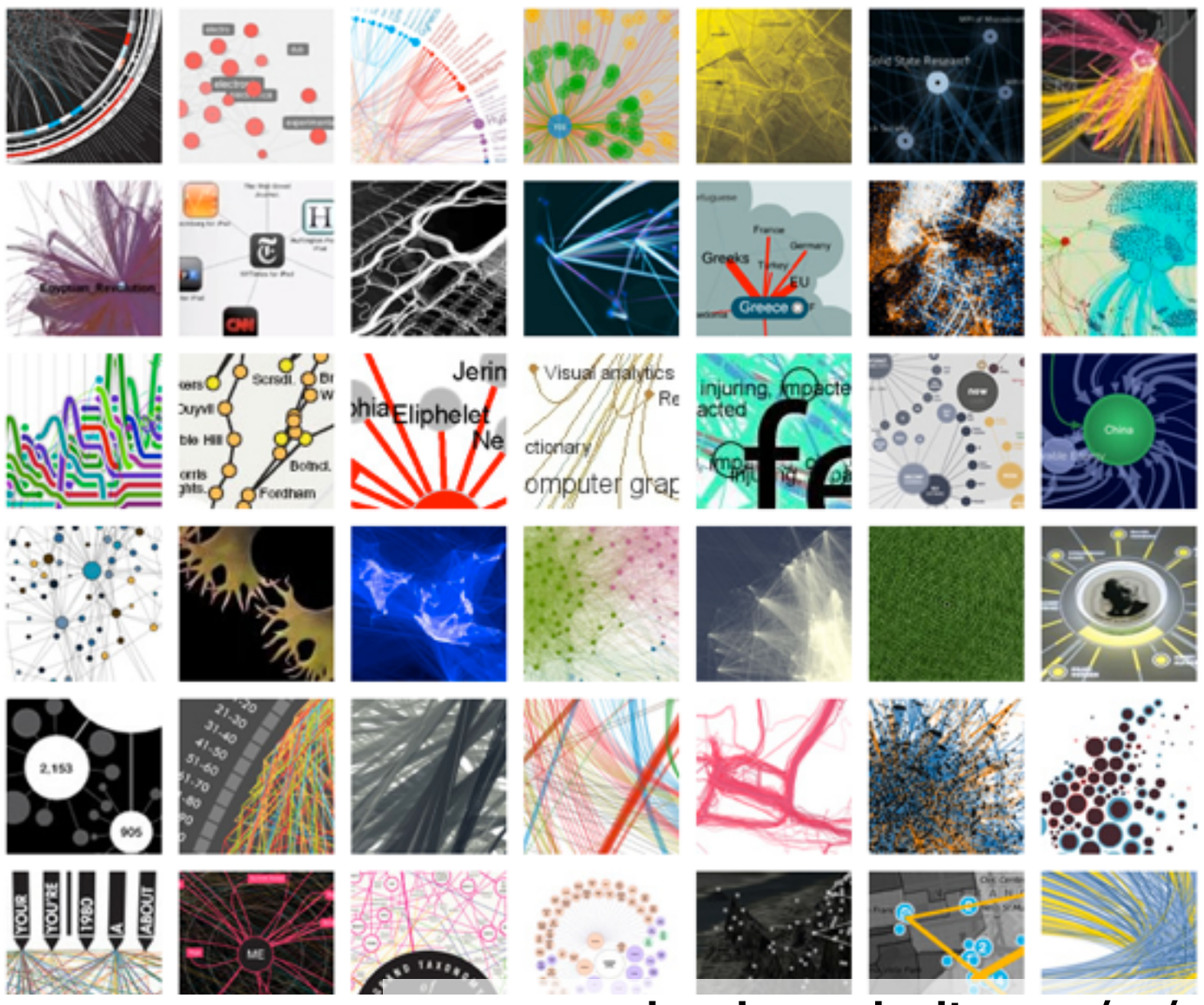
visual complexity

Search the VC database:

GO

Latest Projects:

Indexing **772** projects



Filter by: SUBJECT

- Art (62)
- Biology (52)
- Business Networks (29)
- Computer Systems (33)
- Food Webs (8)
- Internet (30)
- Knowledge Networks (110)
- Multi-Domain Representation (62)
- Music (39)
- Others (63)
- Pattern Recognition (28)
- Political Networks (22)
- Semantic Networks (30)
- Social Networks (101)
- Transportation Networks (49)
- World Wide Web (54)

See All (772)

www.visualcomplexity.com/vc/

Attribute-Driven Layout


facebook

Paul Butler, 2010

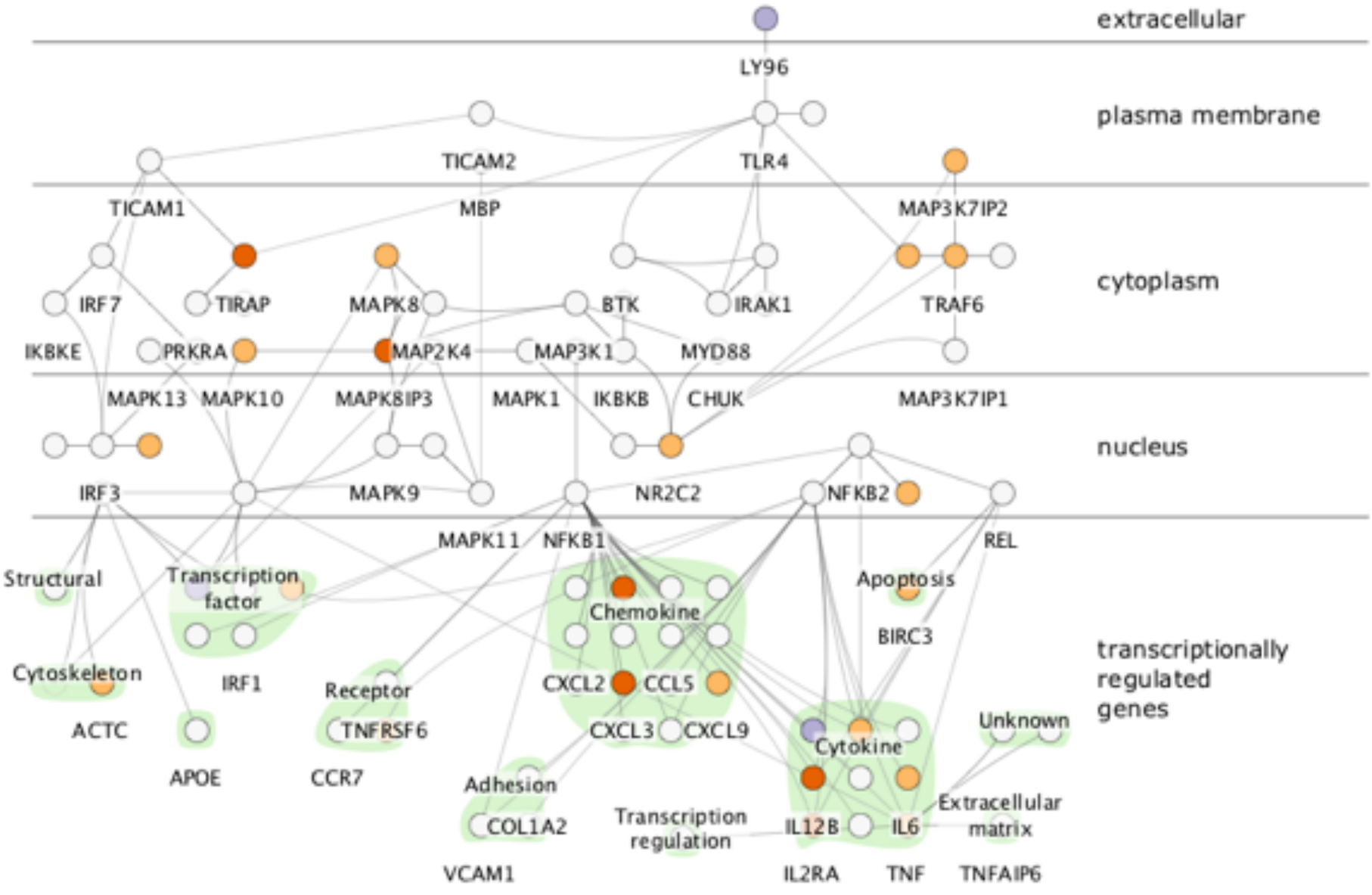
December 2010

https://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919

Attribute-Driven Layout

- Large node-link diagrams get messy!
- Are there additional structures we can exploit?
- Idea: use data attributes to perform layout
  - e.g., scatterplot based on node values
- Dynamic queries and/or brushing can be used to enhance perception of connectivity

Cerebral



Barsky 2008

Visual Exploration of Multivariate Graphs

Martin Wattenberg
Visual Communication Lab, IBM Research
1 Rogers St., Cambridge MA 02142
mwatten@us.ibm.com

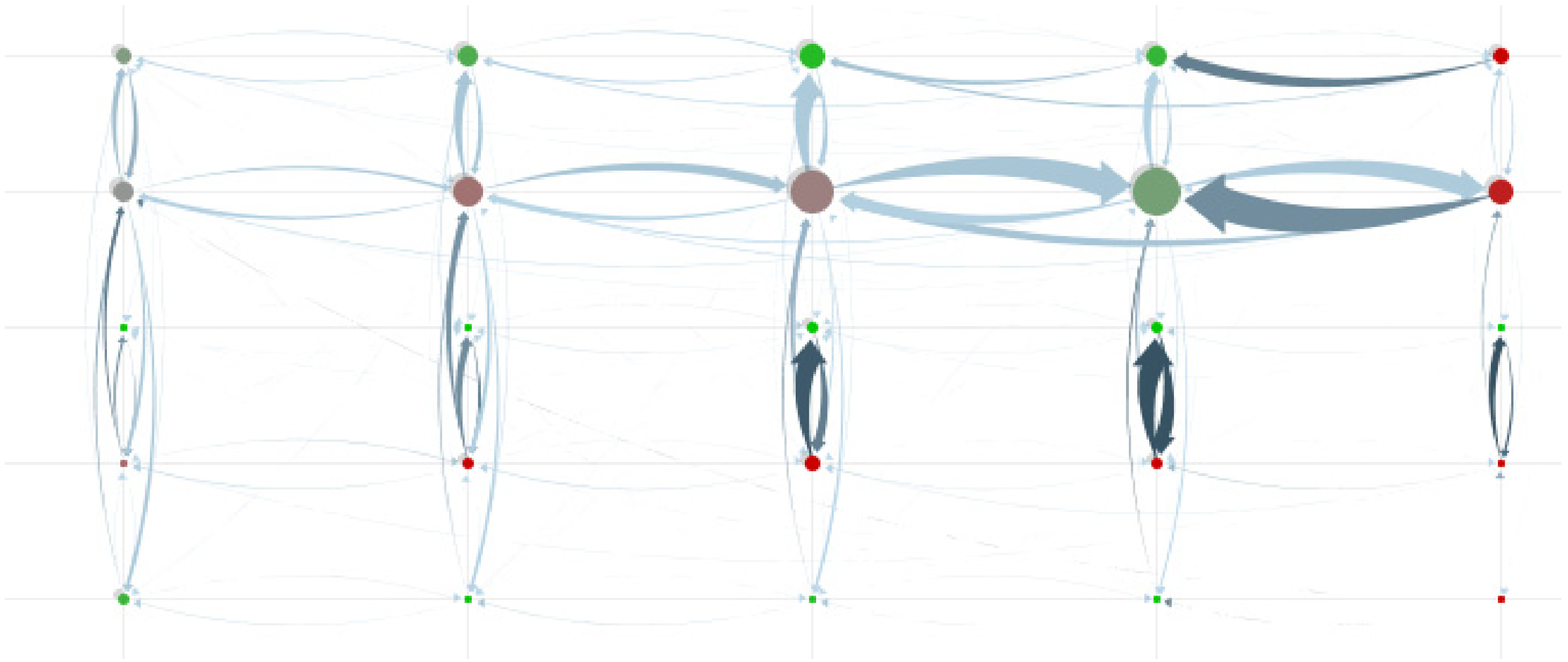


Figure 1. *A PivotGraph visualization of a large graph rolled up onto two categorical dimensions*

ABSTRACT

Author Keywords
information visualization, graph drawing

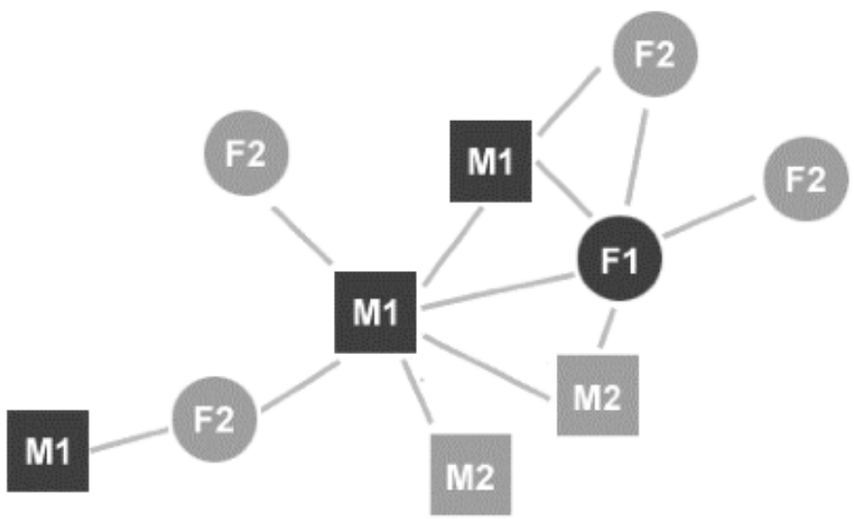
ACM Classification Keywords

Pivot Graph

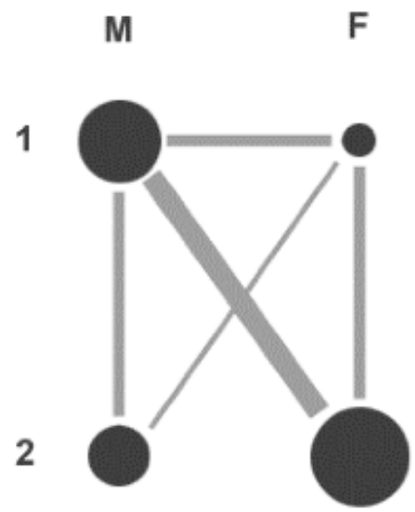
- Task abstraction
  - Show relationship between node attributes and connections in a multi-attribute graph
- Data abstraction
  - Relational dataset
  - Nodes (and edges) have multiple discrete attributes
  - Rollup and selection transformations

Visual Encoding

- Line (1D) or grid (2D) layout
    - Area subdivided by number of values for an attribute
- Number of nodes based on attribute count, not original graph node count
- Size of nodes and edges related to number of aggregated original nodes and edges
- Scalability through abstraction, not layout algorithm



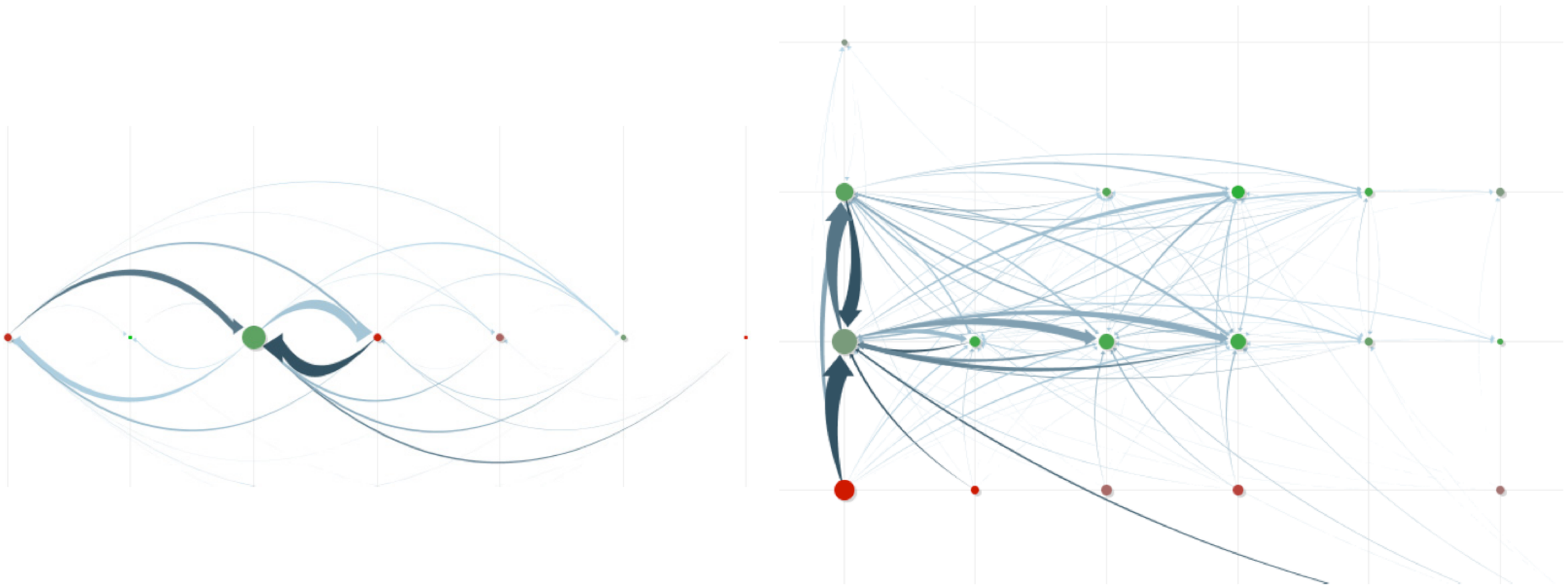
Node and Link Diagram



PivotGraph Roll-up

Visual Encoding

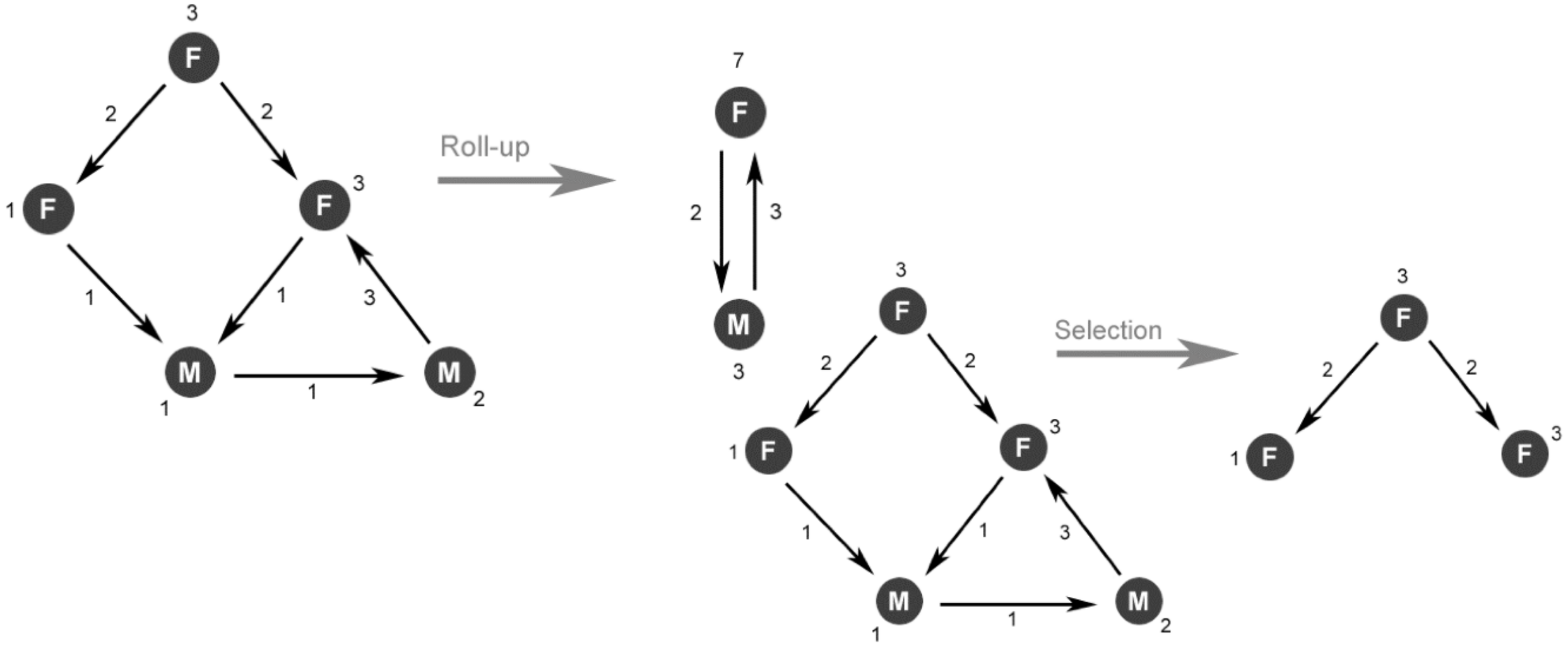
- Line for 1D rollup, or grid for 2D case



Wattenberg 2006

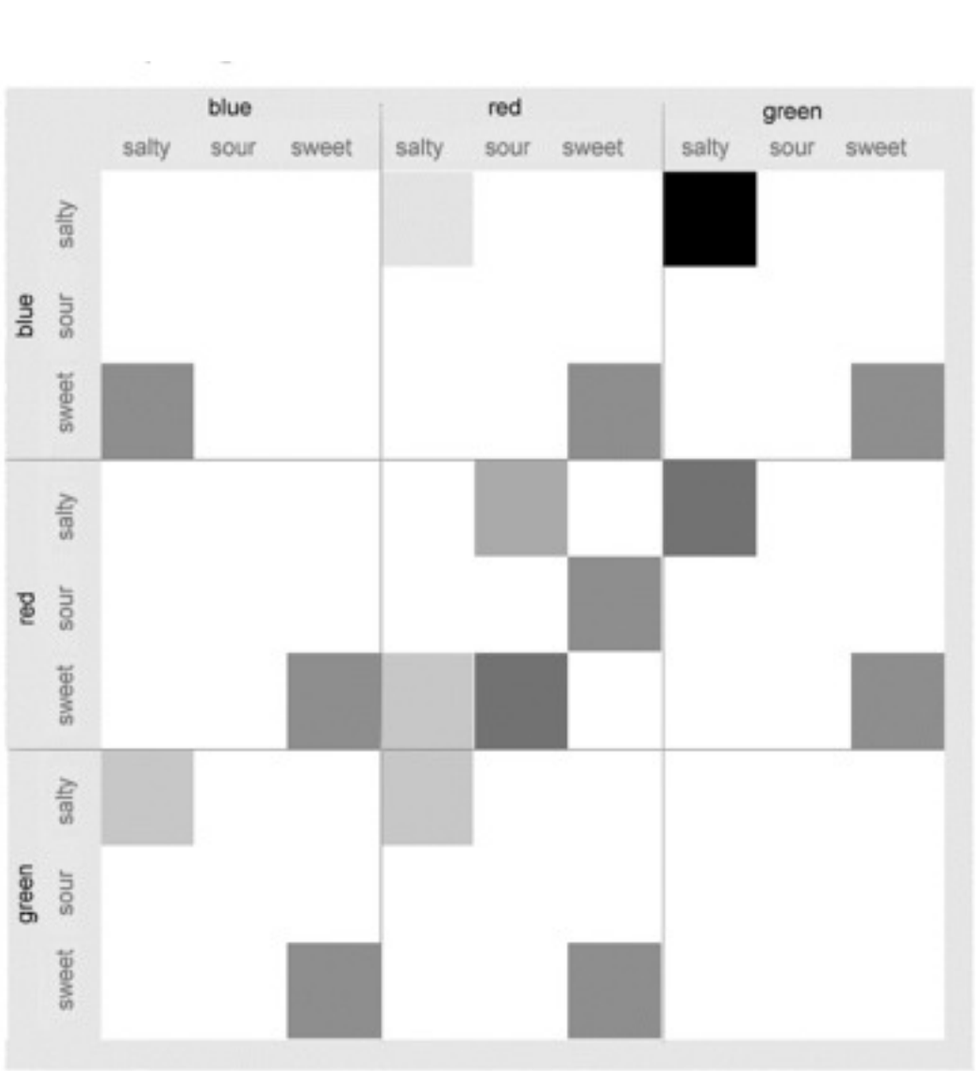
Interaction

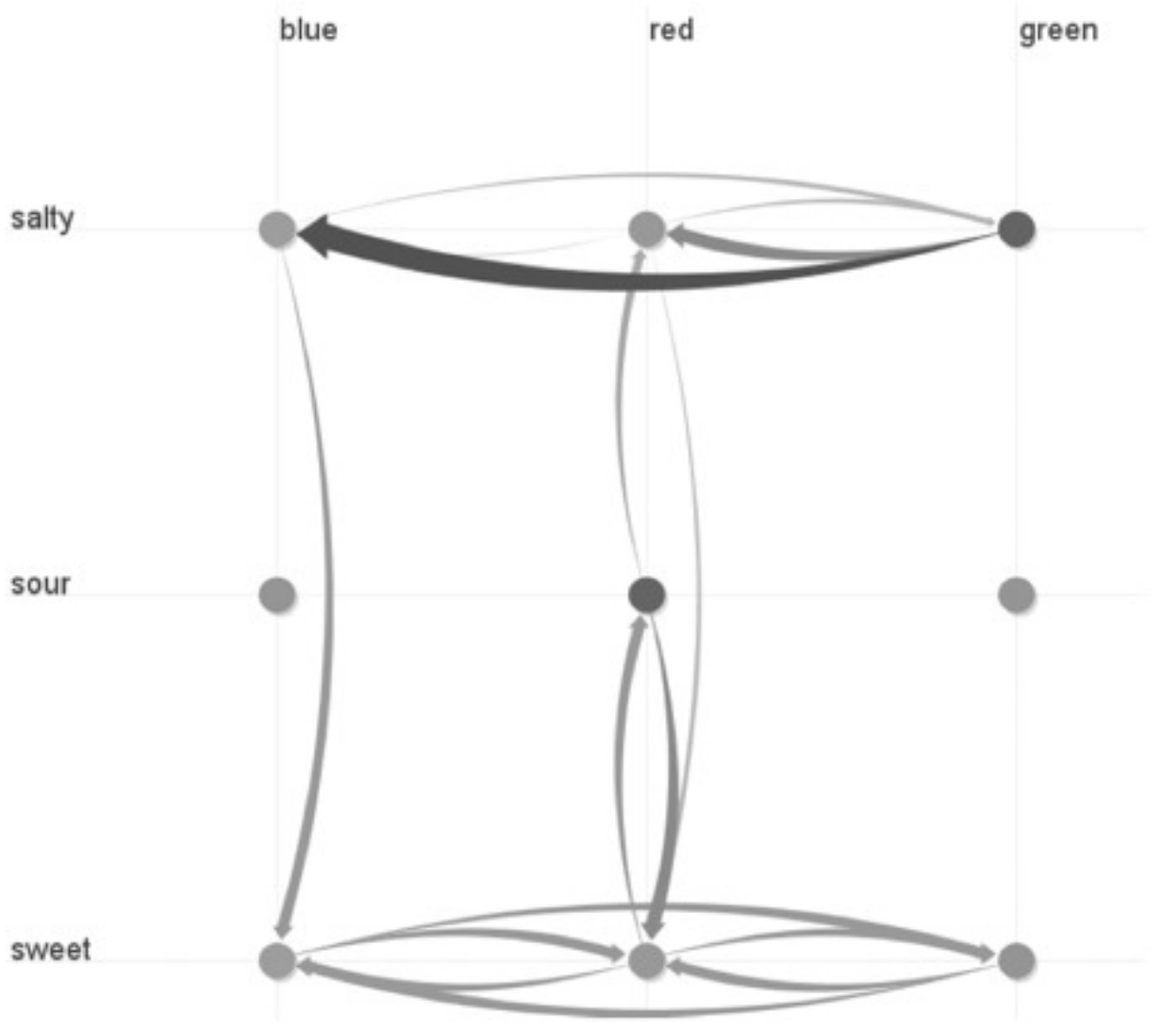
- Changing rollup/selection choices
- Animated transitions between states



Pivot Graph

- In general, more compact than matrix representation





Summary

- Trees
  - Indentation: simple, effective for small trees
  - Node-link and layered: look good but needs exponential space
  - Enclosure (treemaps): great for size related tasks but suffer in structure related tasks
- Graphs
  - Node-link: familiar, but problematic for dense graphs
  - Adjacency matrices: abstract, hard to follow paths
  - Attribute-driven: not always possible

Takeaway: No Best Solution!

Lec15 Reading

- Munzner, Chapter 8.1-8.3

- Unfolding the Earth: Myriahedral Projections. Jarke J. van Wijk, The Cartographic Journal, Vol. 45, No. 1, pp.32-42, February 2008. See also Additional Media.

Reminder
Assignment 03

Assigned: Monday, February 20
Due: Monday, March 13, 4:59:59 pm

Reminder
Project Milestone 02

Assigned: Wednesday, February 22
Due: Wednesday, March 29, 4:59:59 pm