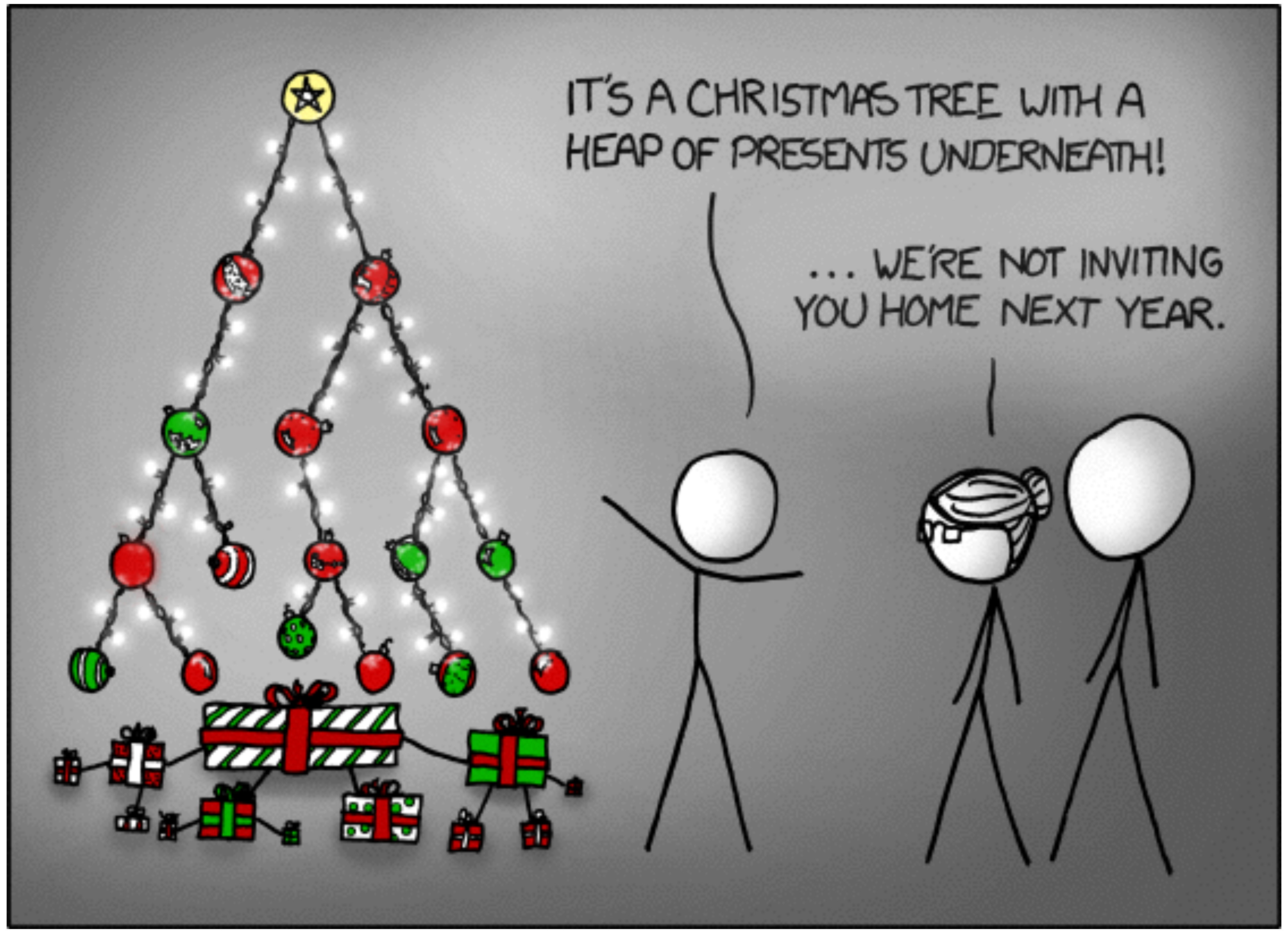
CS171 Visualization

Alexander Lex
alex@seas.harvard.edu

Graphs



[xkcd]

HARVARD
School of Engineering and Applied Sciences

This Week

Reading: VAD, Chapters 9

Lecture 12: Text & Documents

Sections: D3 and JS Design Guidelines. HW1 Review.

Updates

- Design Studio moved to Tuesday after Spring-Break
- HW 4 consists of "only" the project proposal

Design Exercise

Data & Use Case by Augusto Sandoval

Student question: How to show this data?

ID

Gender

High School Type

Degree

Year of Admission

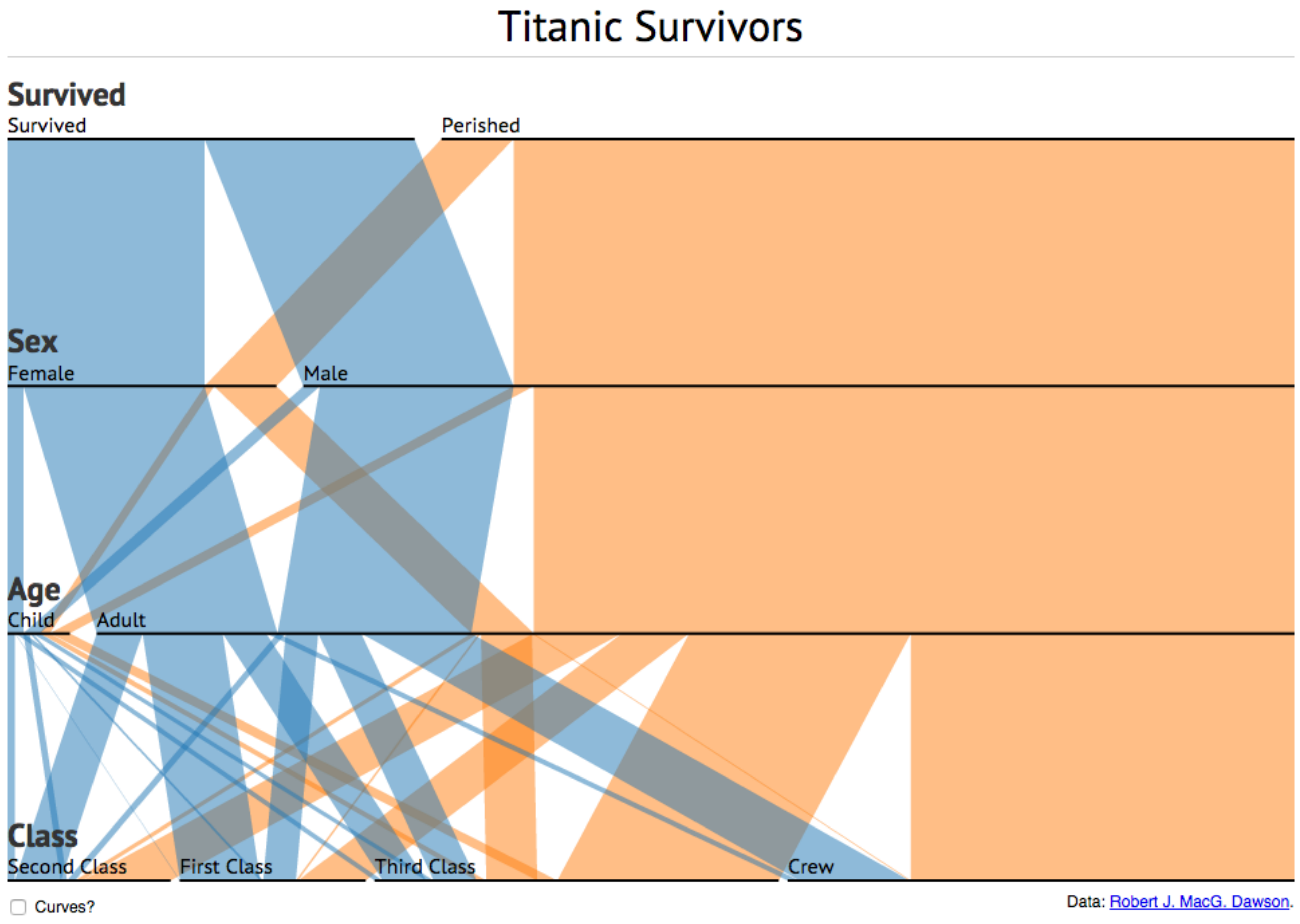
GPA

GPA z-score

Id	gender	e of High Sch	Degree	Year of admissior	GPA	GPA zcore by Degree
nNYliG0HwFTjdFNks1	M	Private	journalism	2012	0	-6.384275
eT+GiwpOBm4QJ3cZ	M	Public	engineering	2008	0.18888888	-5.895198
ryP0ztCYW26aMdUq	M	Private	Laws	2004	0	-5.730906
S034U/Z7g9IQXTuU9	M	Private	Arts	2012	0.26666666	-5.028417
9Z16bxKDeVDbGiiD8	F	Private	Arts	2012	1.666666667	-2.331488
4EI03gz2P0m1wkwA	M	Private	engineering	2013	0.6	-4.865675
Ic+5vQGCtRba7j/vTx	M	Private	engineering	2012	0.84444444	-4.363712
9Rqf6BCleCiywUVdG	F	Private	history	2012	0.76	-4.033634
iJtFP+NRF/yVVW2yLt	M	Private	agronomy	2012	0.84	-3.64938
nKiM2cmiLt5hXdGQ	M	Private	engineering	2009	1	-4.000308
2s9GWUyTNMwyI8al	M	Public	theology	2008	0.91111111	-3.627317
ug3OgnYirUcEUImXtJ	F	Private	nursing	2011	1.733333333	-4.0063
Sm+XS3+8amJFzowV	M	Private	theology	2013	0.98333333	-3.507009
jrwfkQmu9YDTzWdE	F	Public	nursing	2010	1.85	-3.495274
Jv6VtB+mIVY30ZYR4	M	Private	business	2012	1.316666667	-3.428196
szP2BH1uaYrsk3w9J	F	Private	COLLEGE	2012	1	-3.345934
elIKHcnNQ0fW9jxR9	F	Private	Laws	2013	1.253333333	-3.256467
qdY8dwFW0CxGdqUI	F	Private	nursing	2008	1.933333333	-3.250402

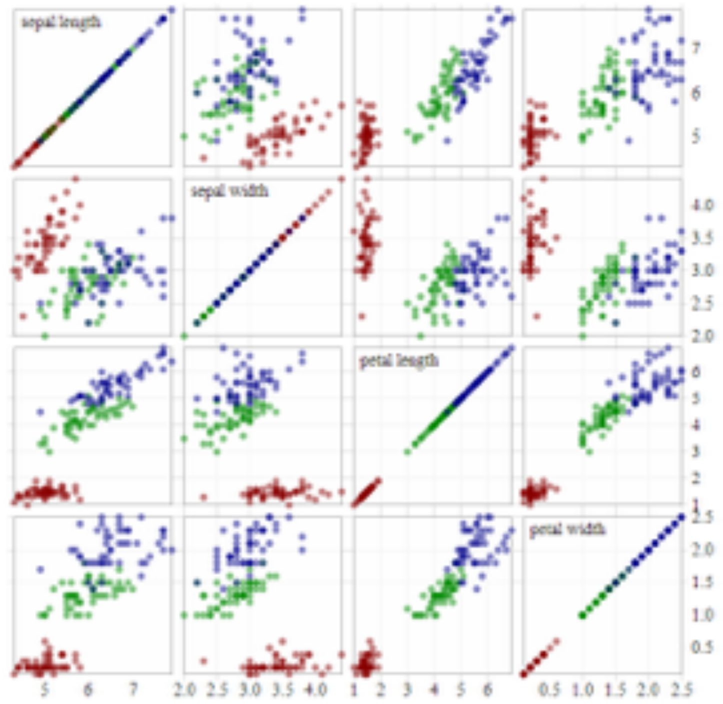
Visualizing Categorical Data

Example:
Parallel Sets

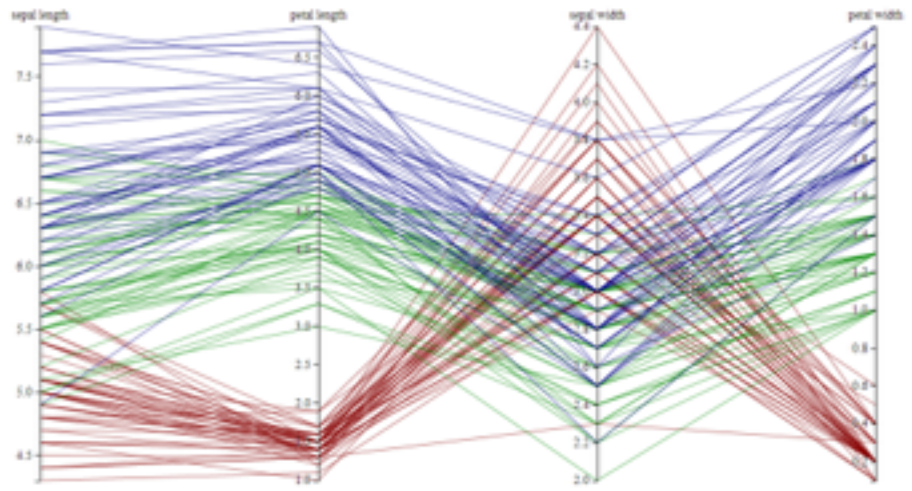


Last Week:
Highdimensional Data

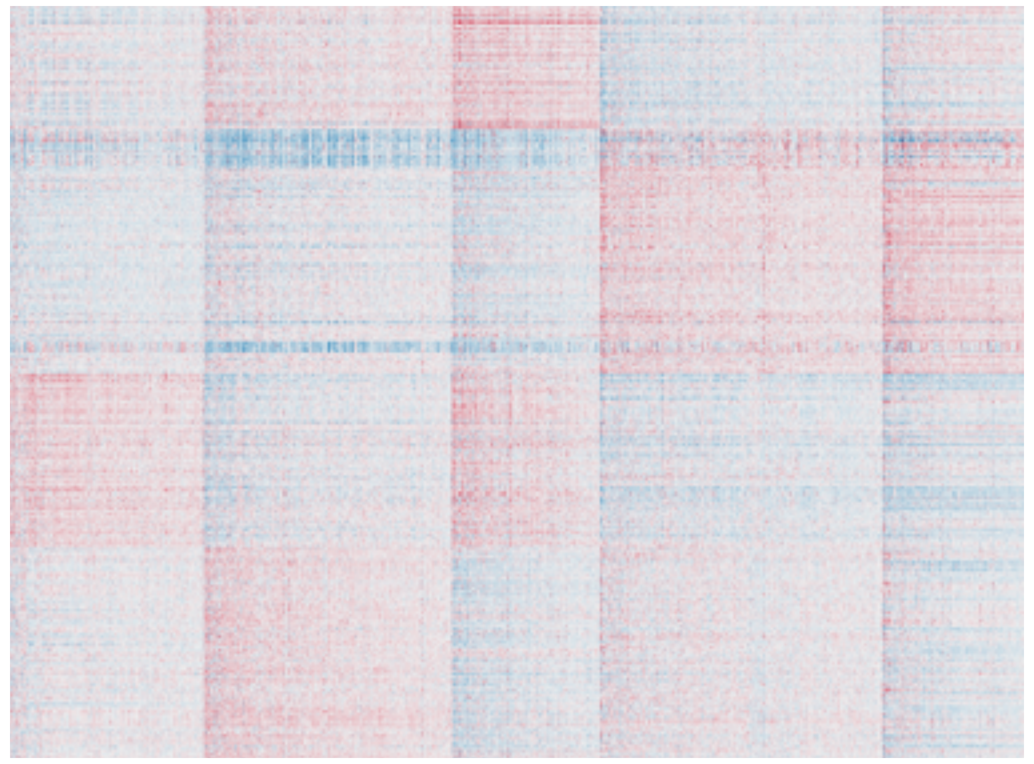
Analytic Component



Scatterplot Matrices
[Bostock]

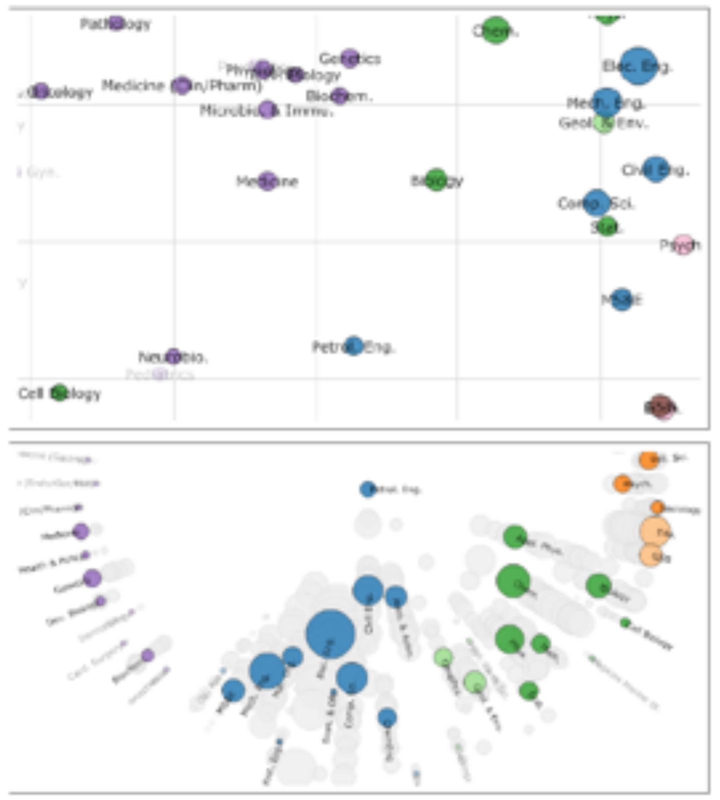


Parallel Coordinates
[Bostock]



Pixel-based visualizations / heat maps

Multidimensional Scaling
[Doerk 2011]



[Chuang 2012]

no / little analytics

strong analytics component

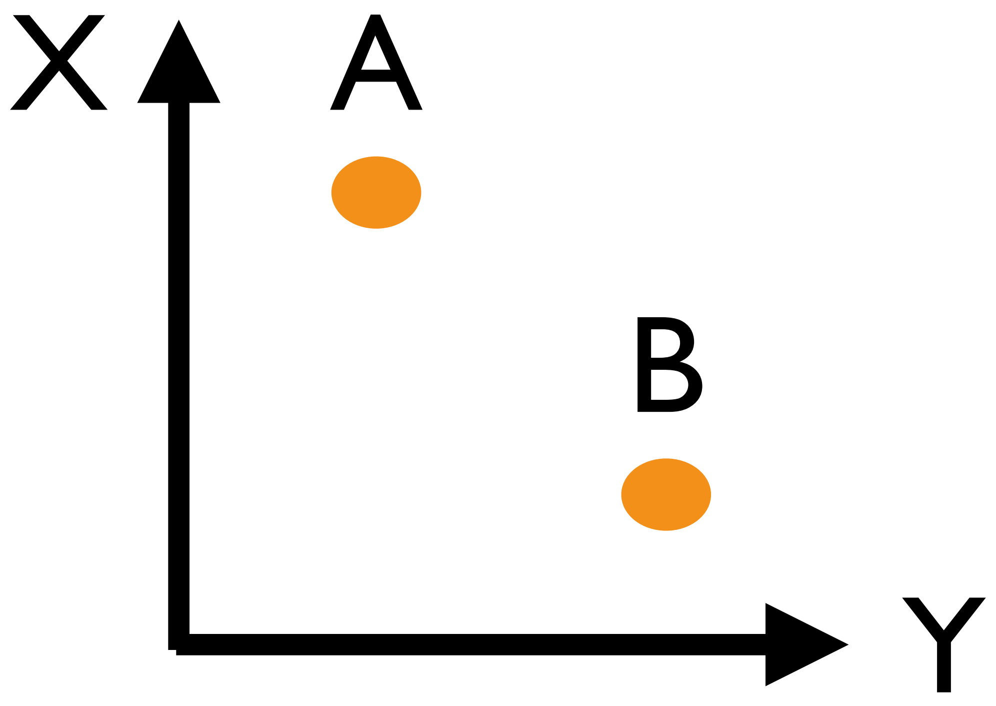
Geometric Methods

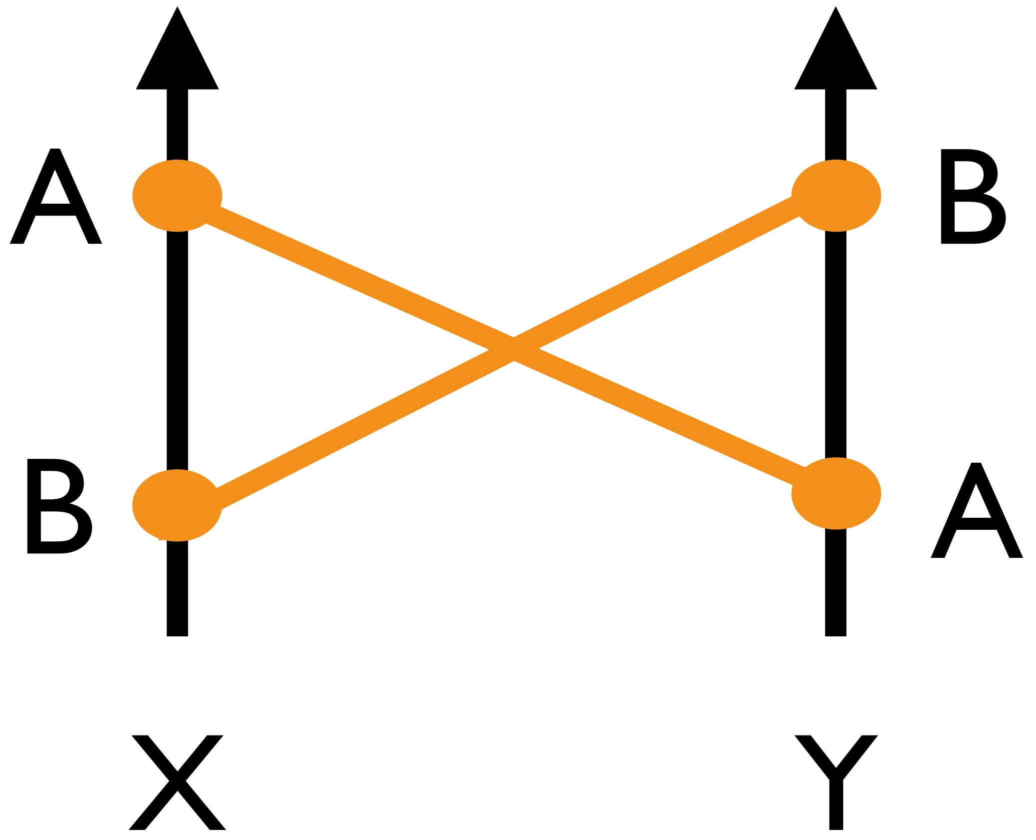
Parallel Coordinates (PC)

Inselberg 1985

Axes represent attributes

Lines connecting axes represent items





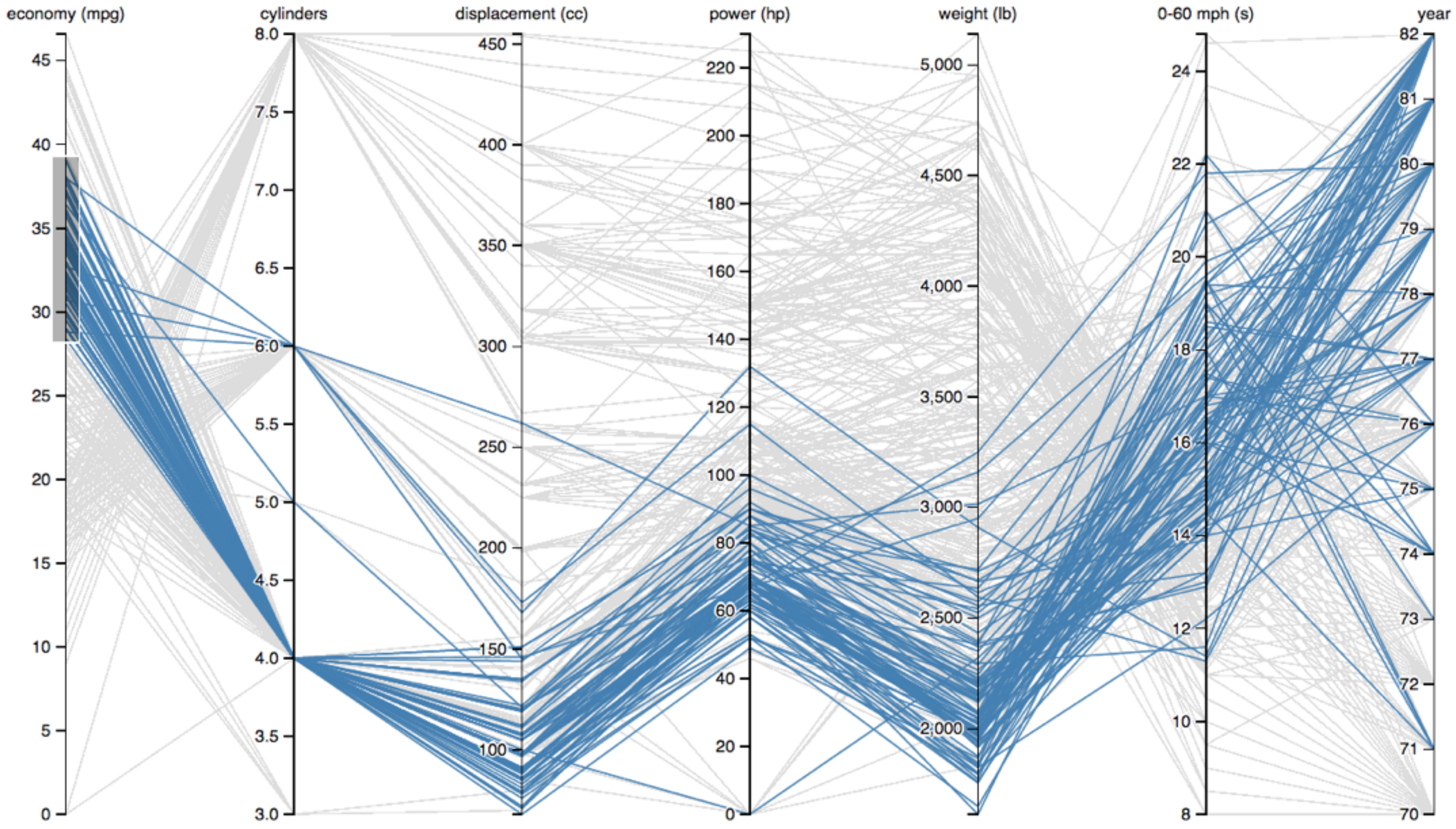
Parallel Coordinates

Each axis represents dimension

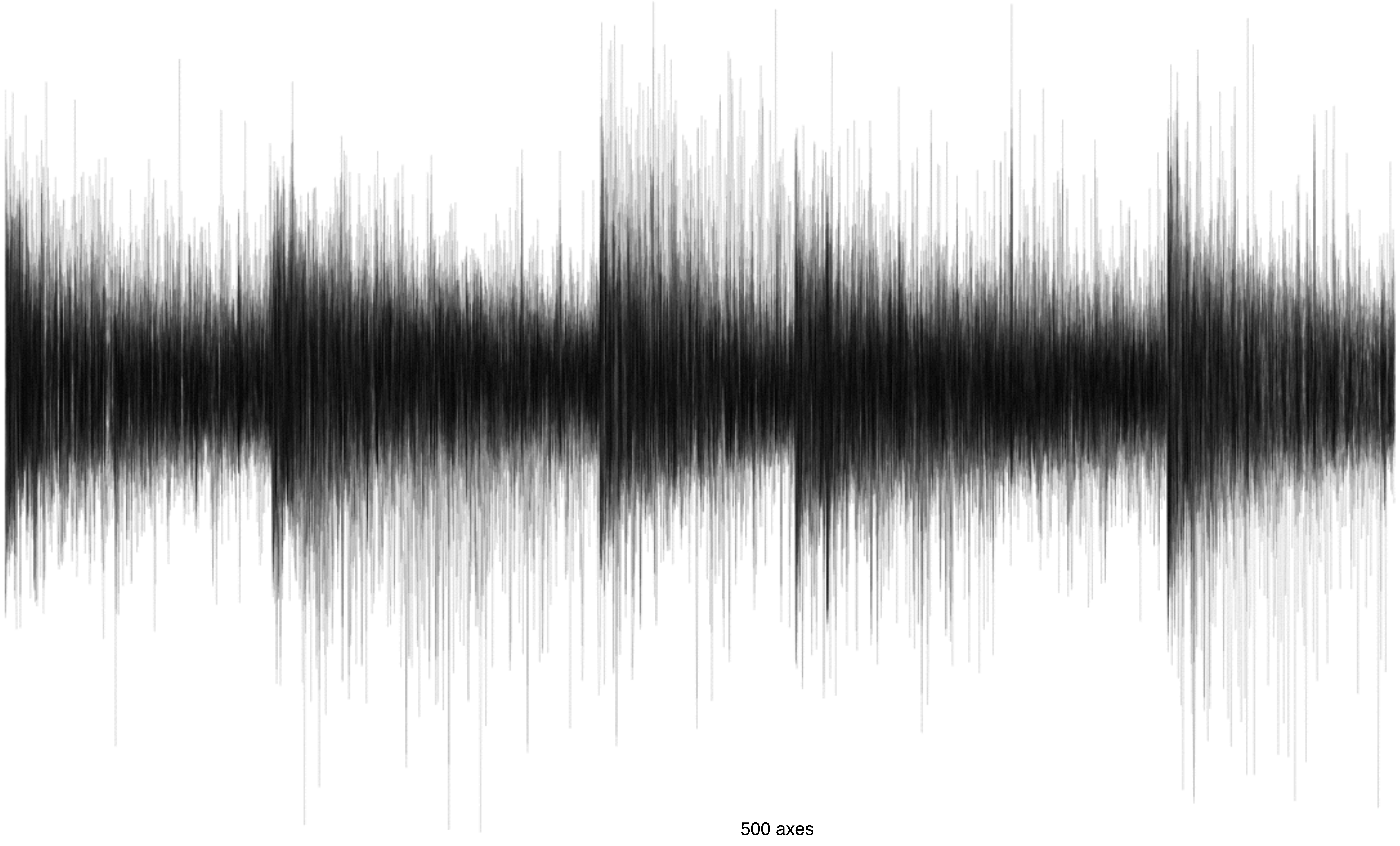
Lines connecting axis represent records

Suitable for

- all tabular data types
- heterogeneous data



PC Limitation: Scalability to Many Dimensions

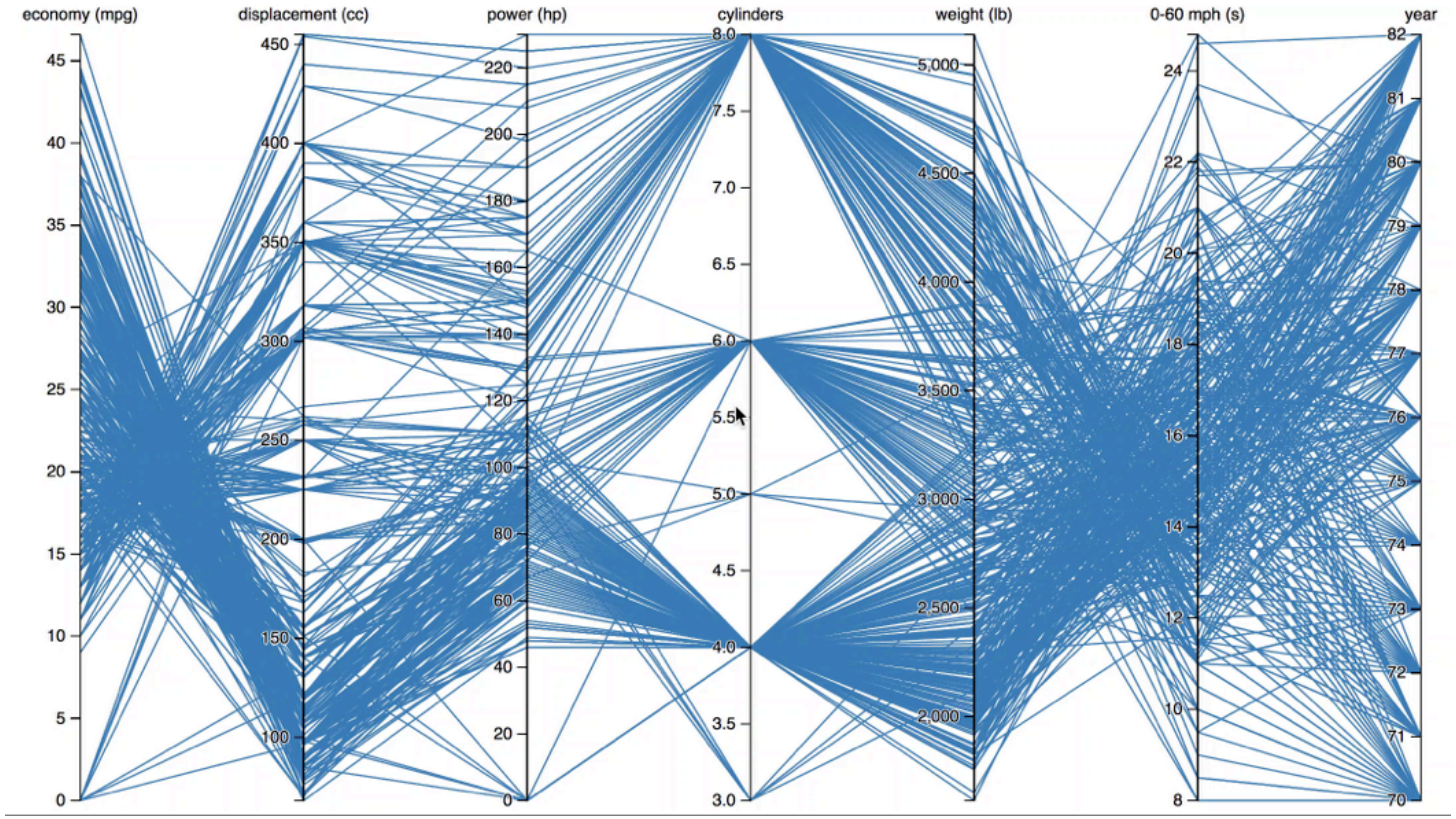


PC Limitations

Correlations only between adjacent axes

Solution: Interaction

- Brushing
- Let user change order



Parallel Coordinates

Shows primarily relationships between adjacent axis

Limited scalability (~50 dimensions, ~1-5k records)

- Transparency of lines

Interaction is crucial

- Axis reordering
- Brushing
- Filtering

Algorithmic support:

- Choosing dimensions
- Choosing order
- Clustering & aggregating records

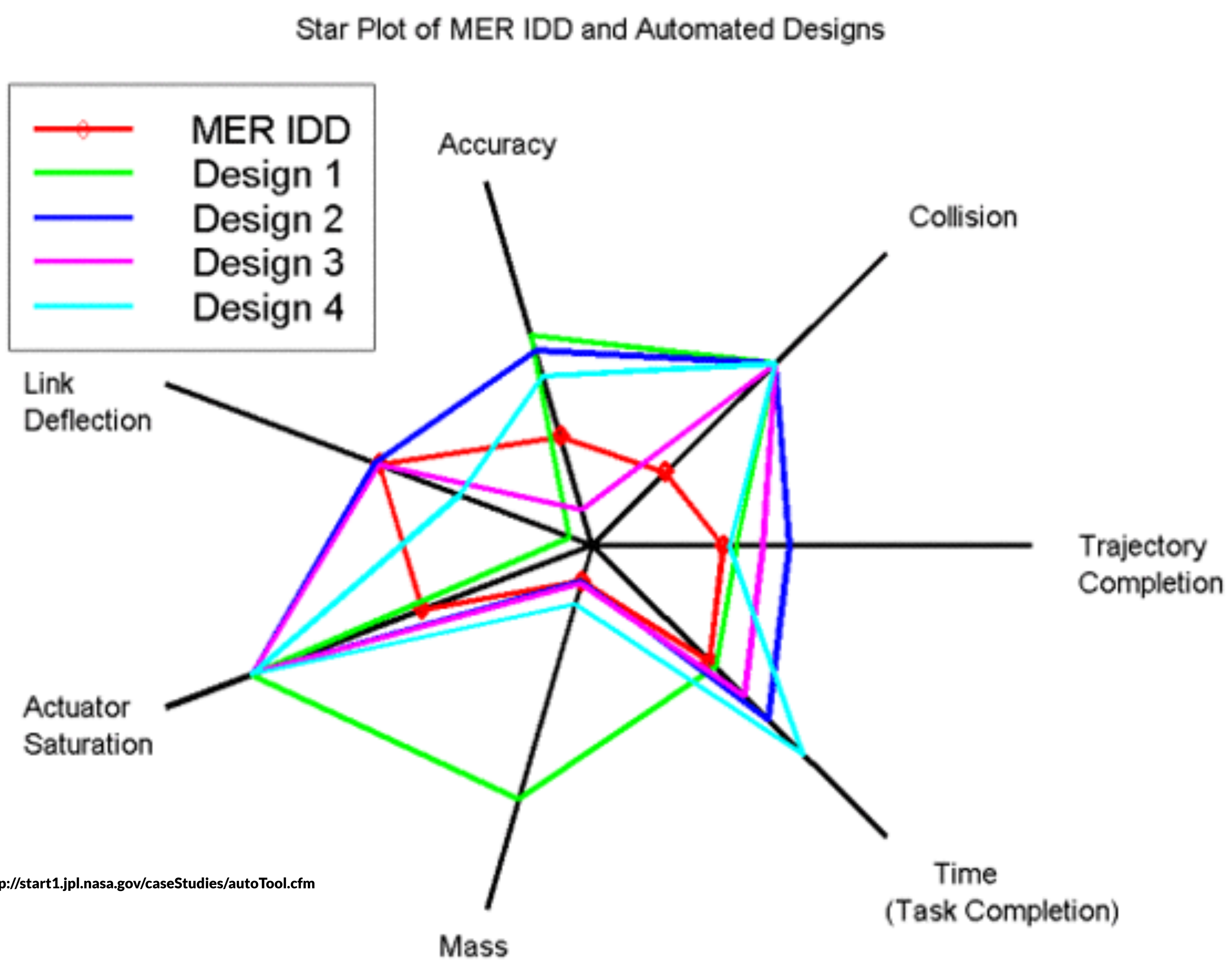
http://bl.ocks.org/jasondavies/1341281

Star Plot

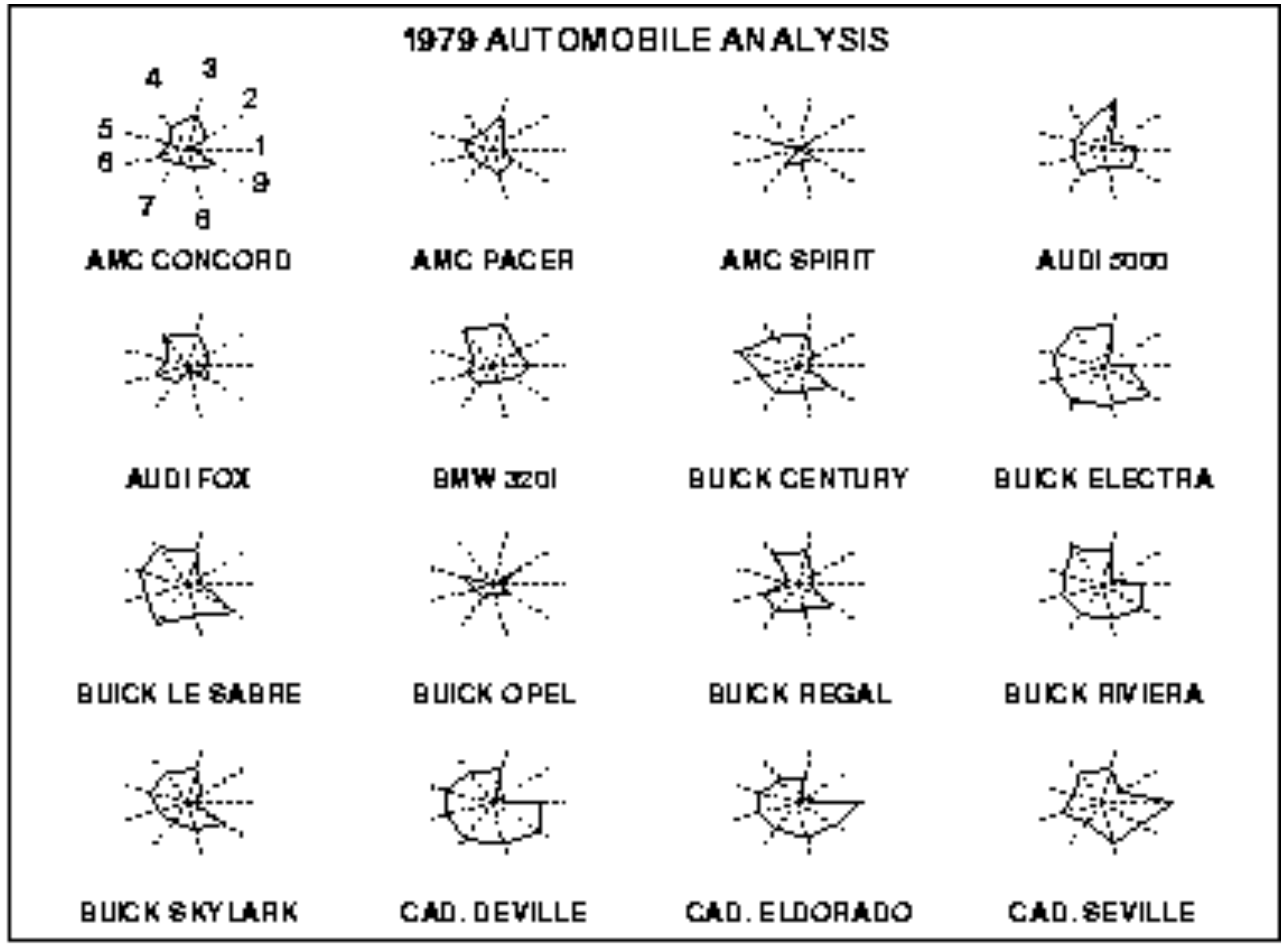
[Coekin1969]

Similar to parallel coordinates

Radiate from a common origin



http://start1.jpl.nasa.gov/caseStudies/autoTool.cfm



http://www.itl.nist.gov/div898/handbook/eda/section3/starplot.htm

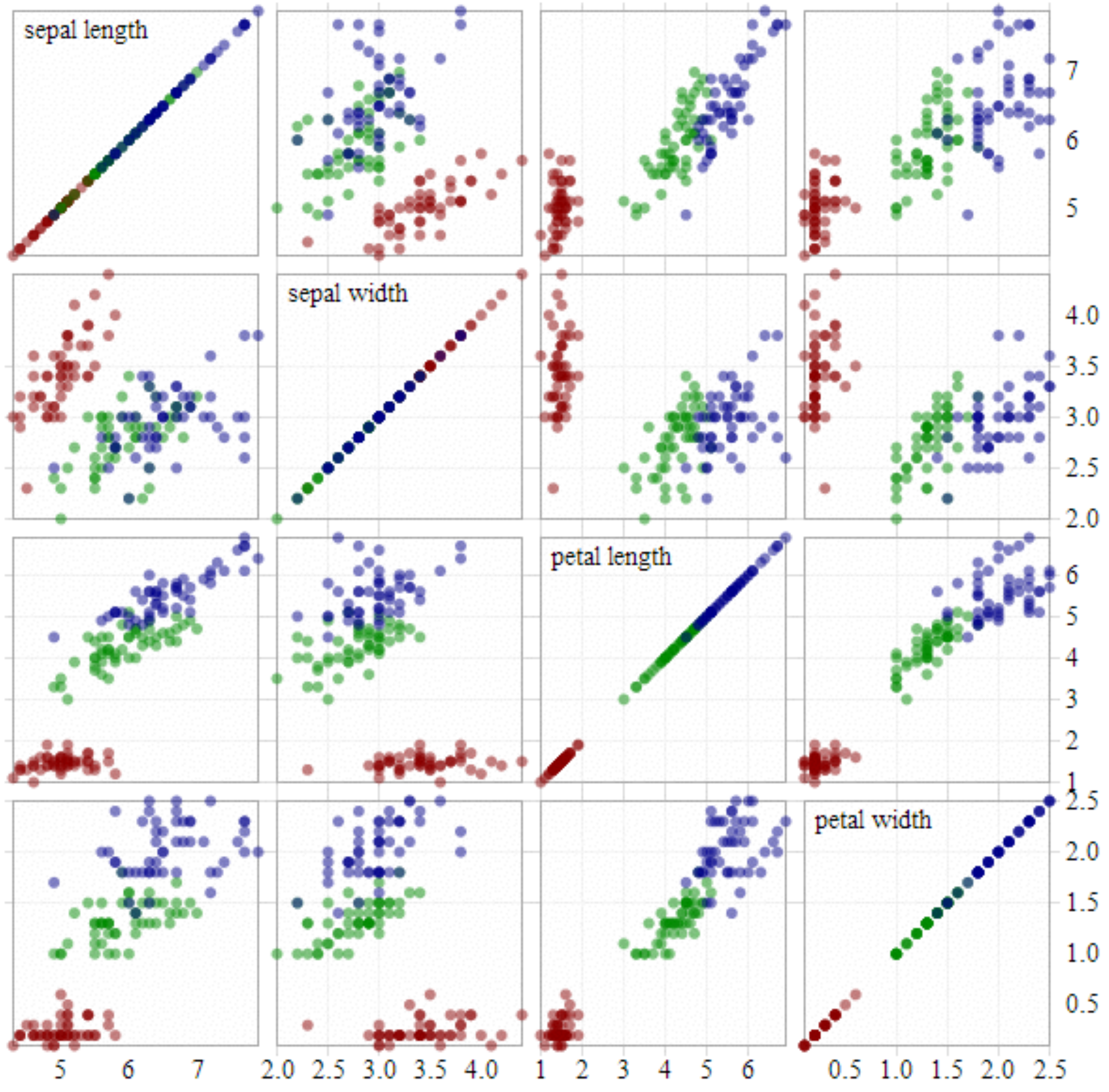
http://bl.ocks.org/kevinschaul/raw/8833989/

Scatterplot Matrices (SPLOM)

Matrix of size d*d

Each row/column is one dimension

Each cell plots a scatterplot of two dimensions



Scatterplot Matrices

Limited scalability (~20 dimensions, ~500-1k records)

Brushing is important

Often combined with “Focus Scatterplot” as F+C technique

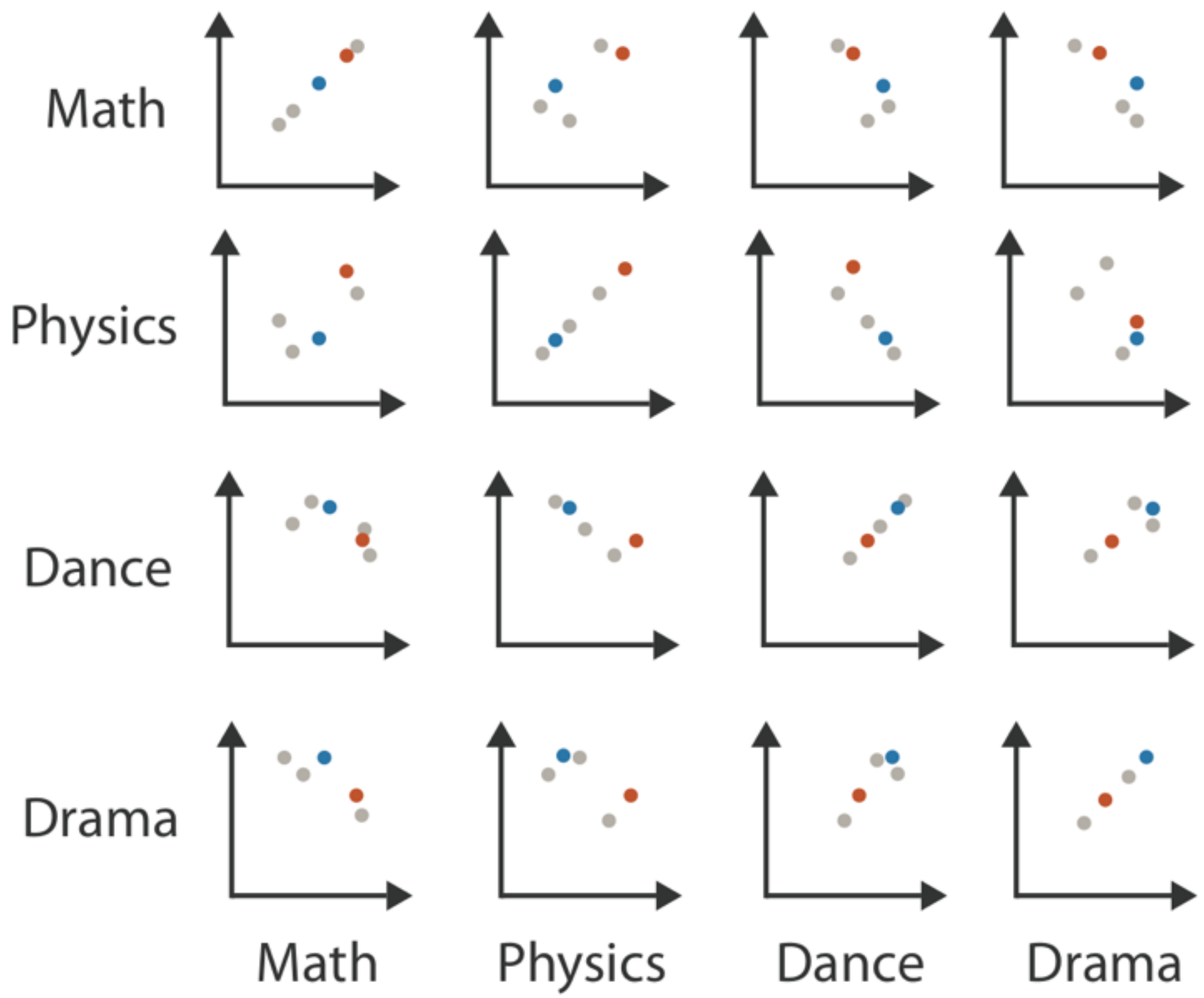
Algorithmic approaches:

- Clustering & aggregating records
- Choosing dimensions
- Choosing order

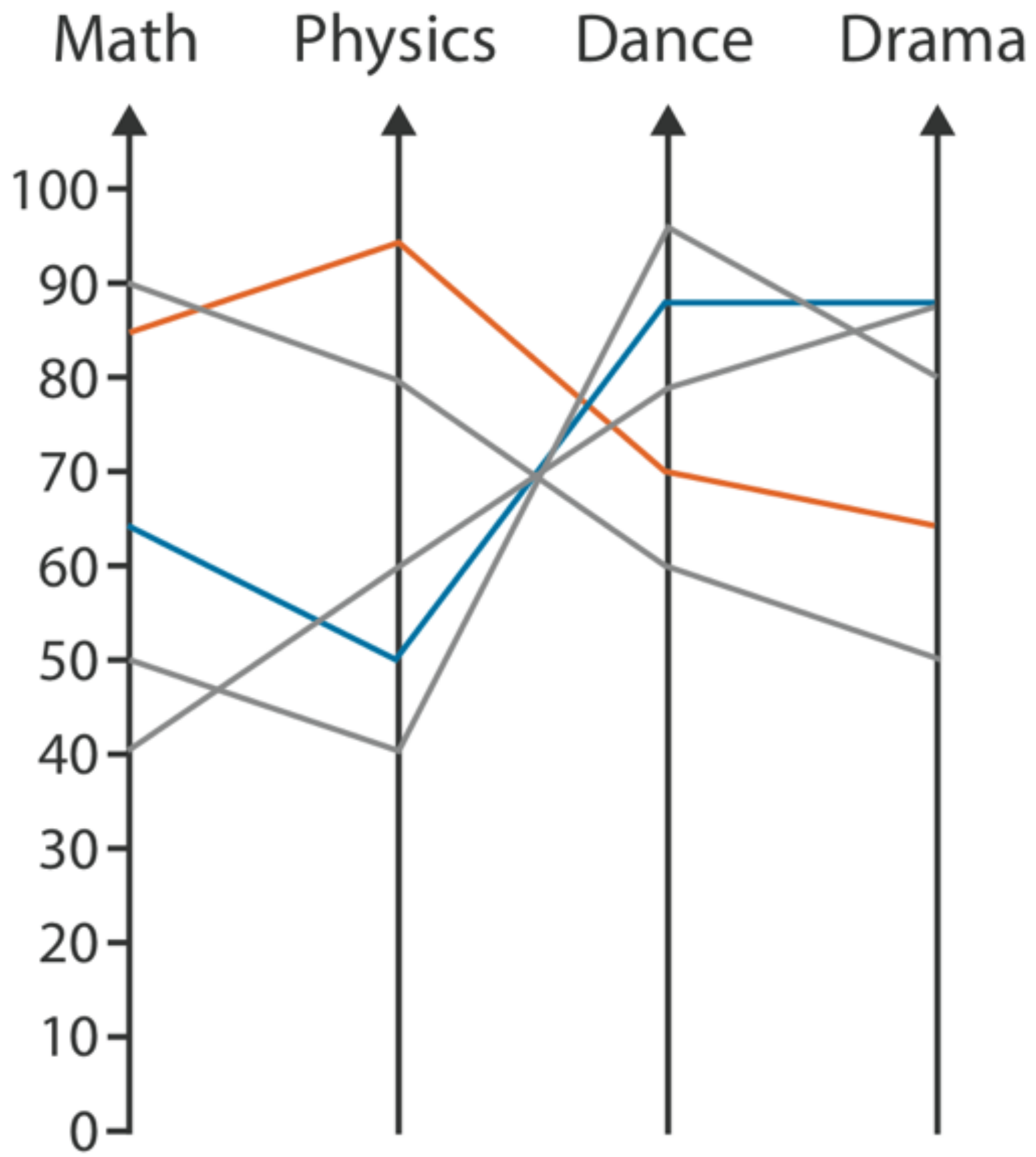
Table

Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90

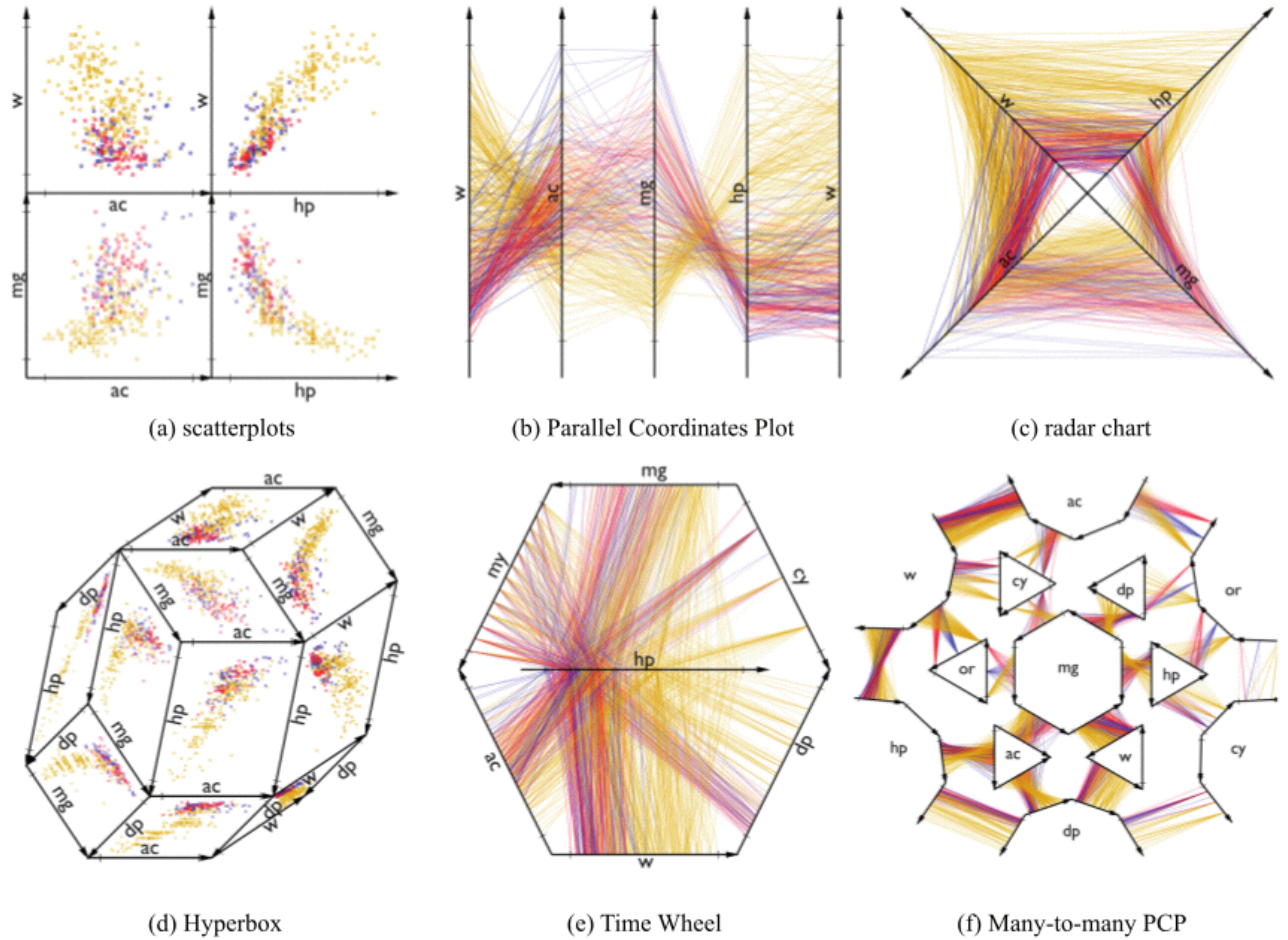
Scatterplot Matrix



Parallel Coordinates



Flexible Linked Axes (FLINA)



Claessen & van Wijk 2011

Data Reduction

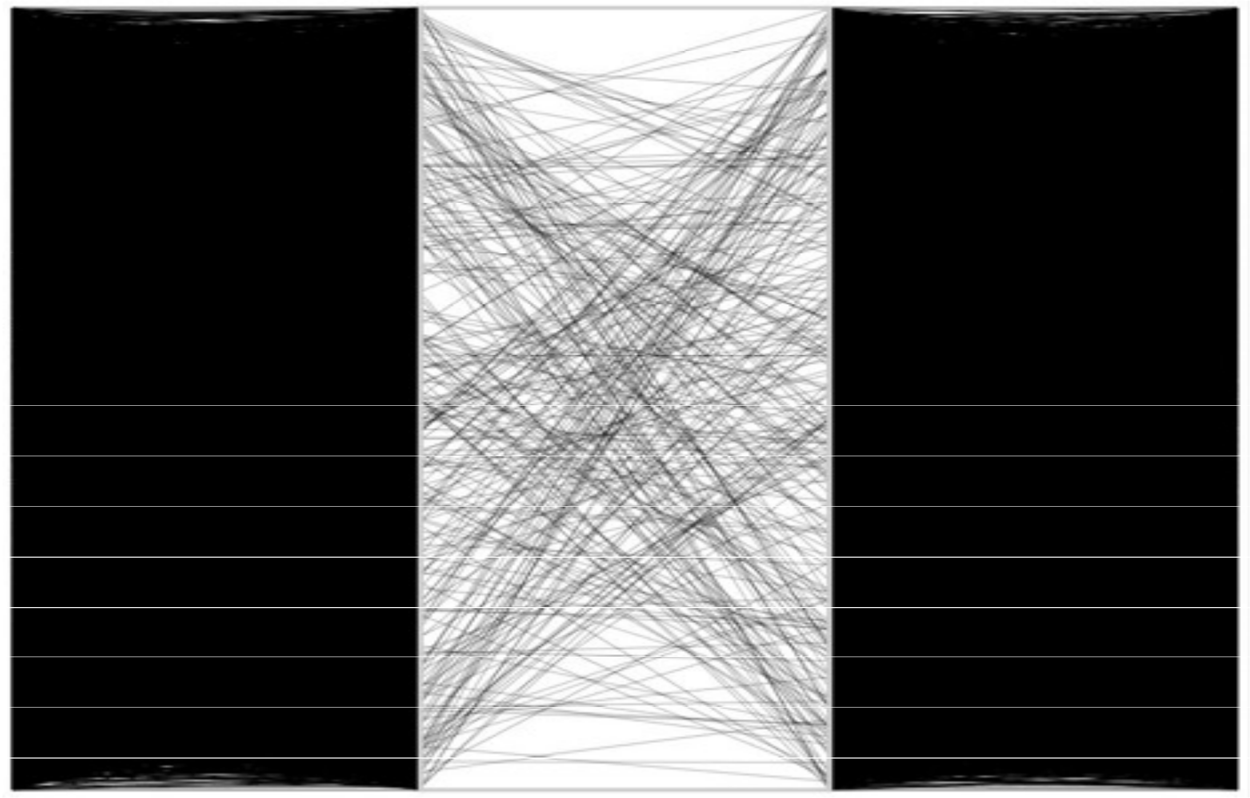
Sampling

Don’t show every element, show a (random) subset

Efficient for large dataset

Apply only for display purposes

Outlier-preserving approaches



[Ellis & Dix, 2006]

Filtering

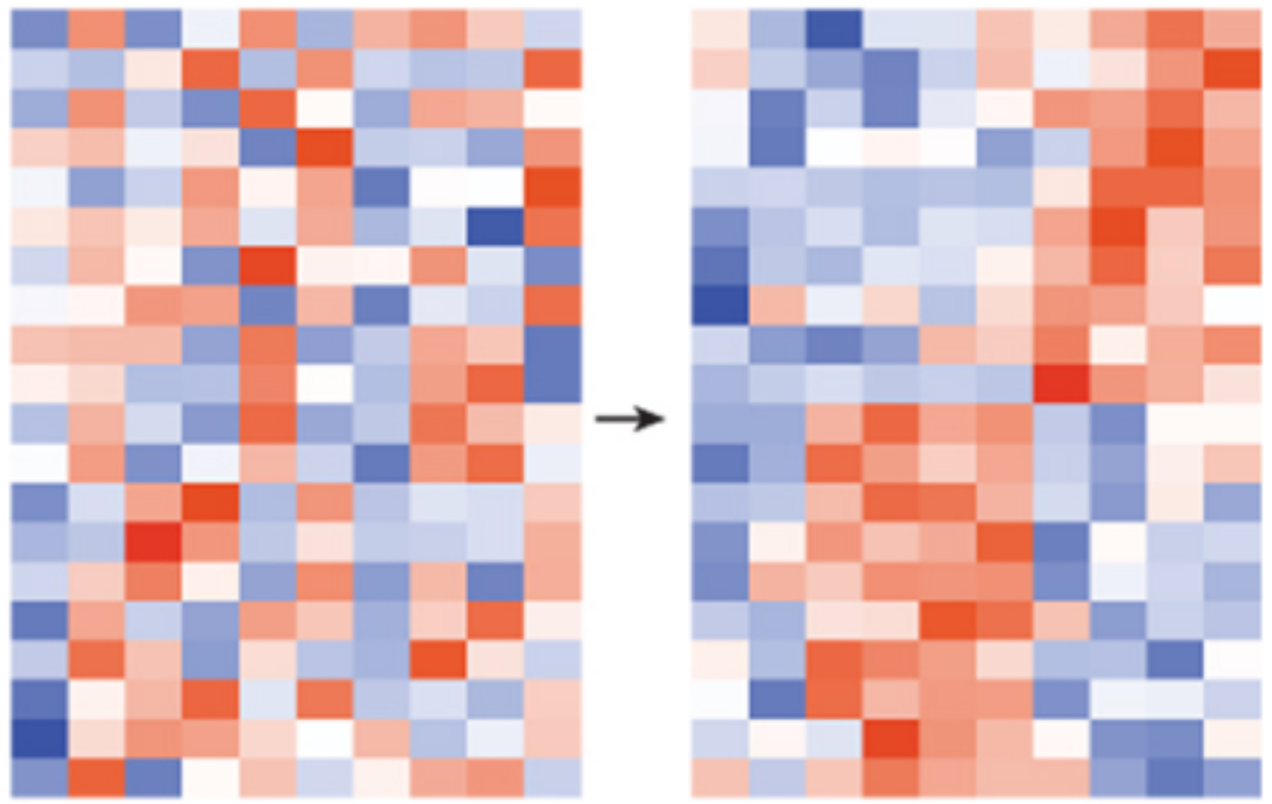
Define criteria to remove data, e.g.,

- minimum variability
- > / < / = specific value for one dimension
- consistency in replicates, …

Can be interactive, combined with sampling

Pixel Based Methods

Pixel Based Displays



Each cell is a “pixel”, value encoded in color / value

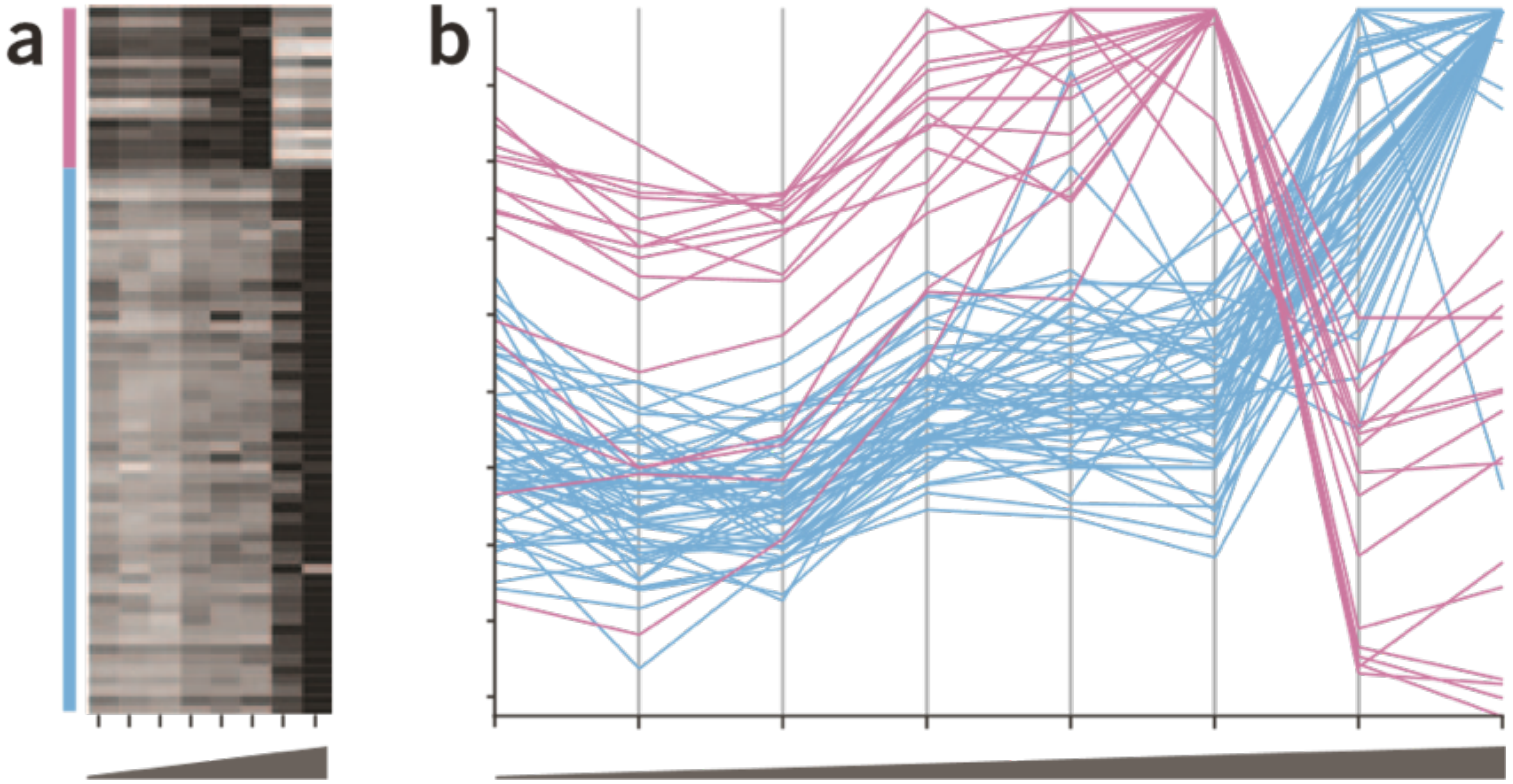
Meaning derived from ordering

If no ordering inherent, clustering is used

Scalable – 1 px per item

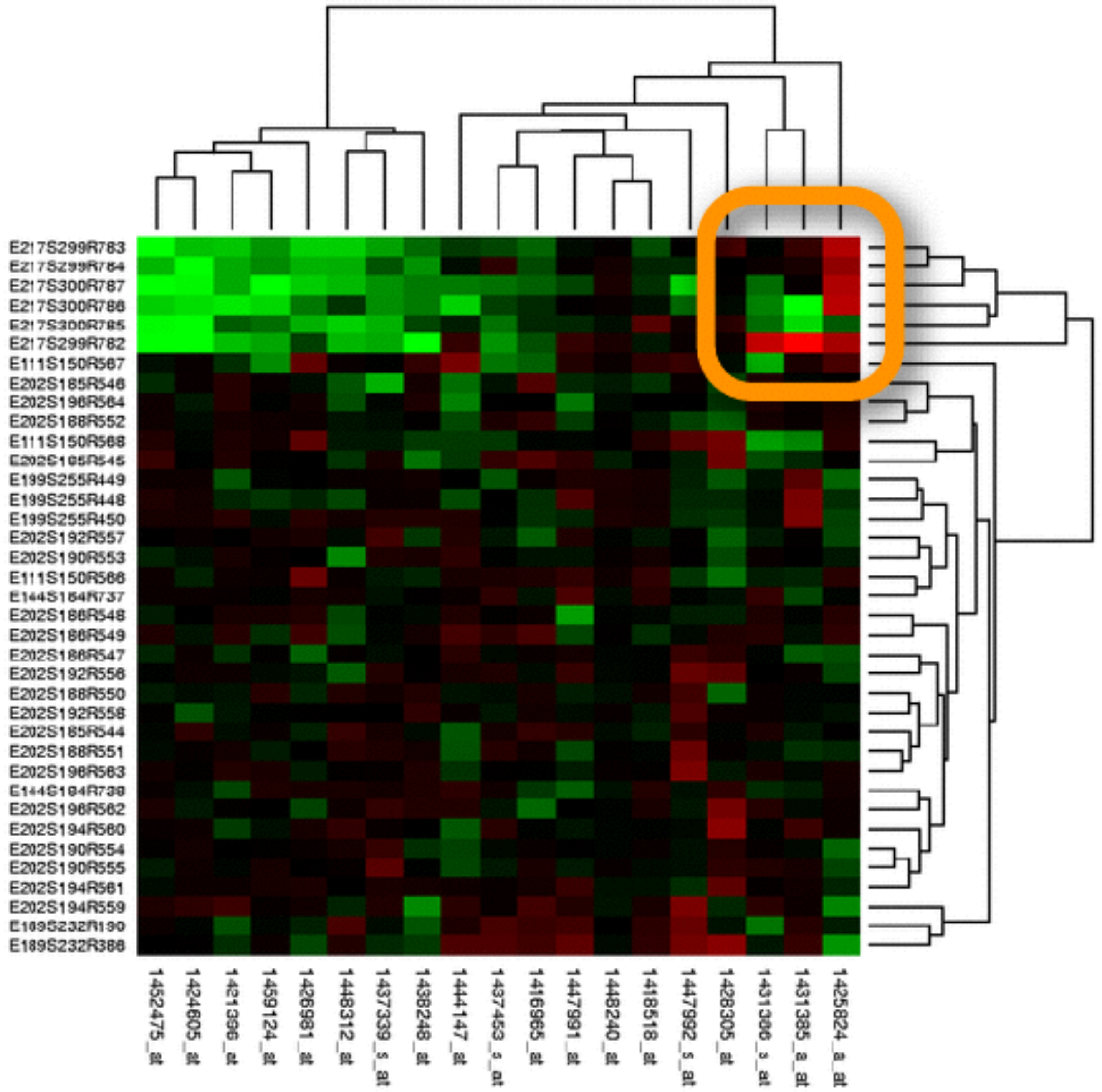
Good for homogeneous data

same scale & type

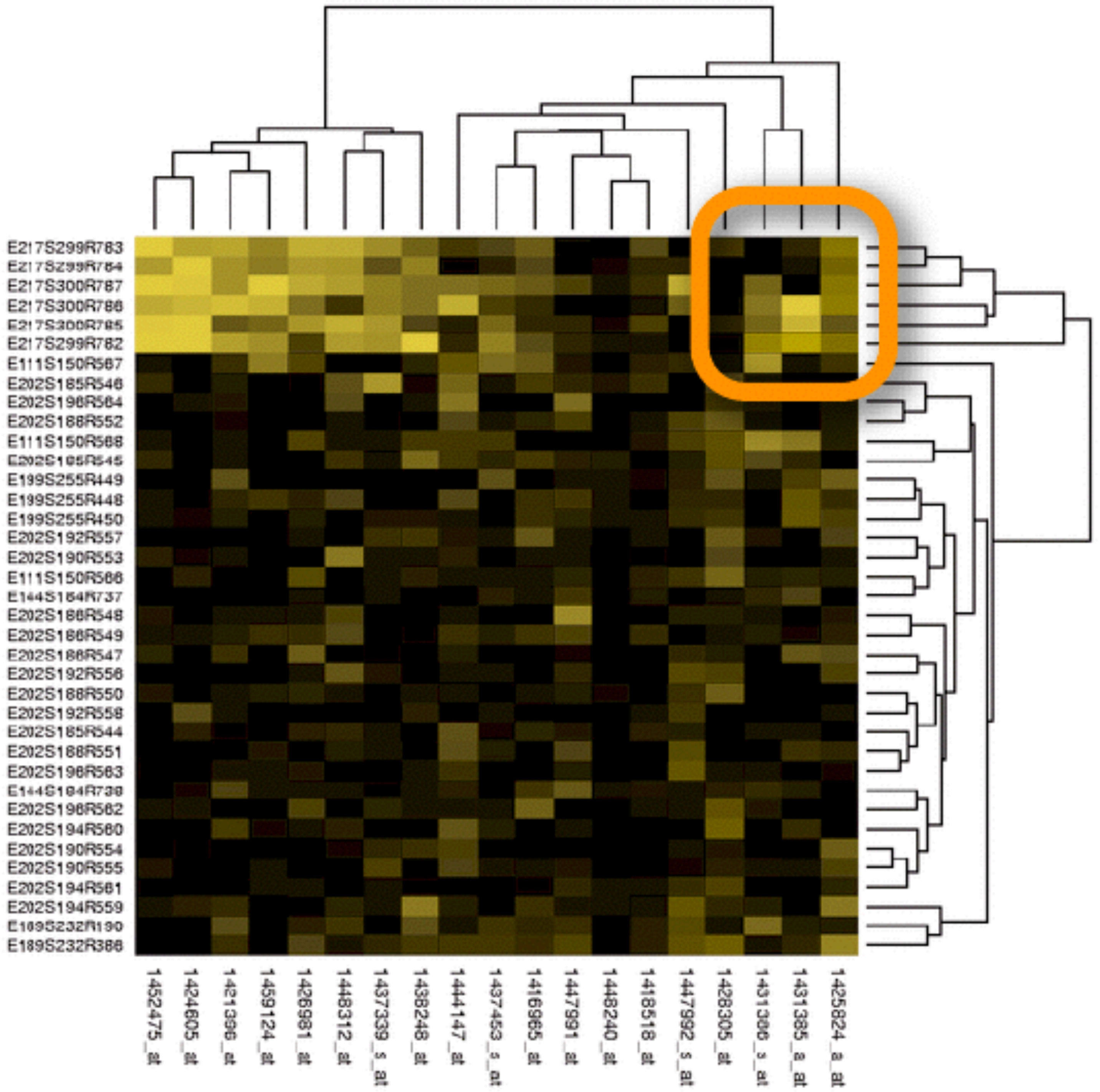


[Gehlenborg & Wong 2012]

Bad Color Mapping

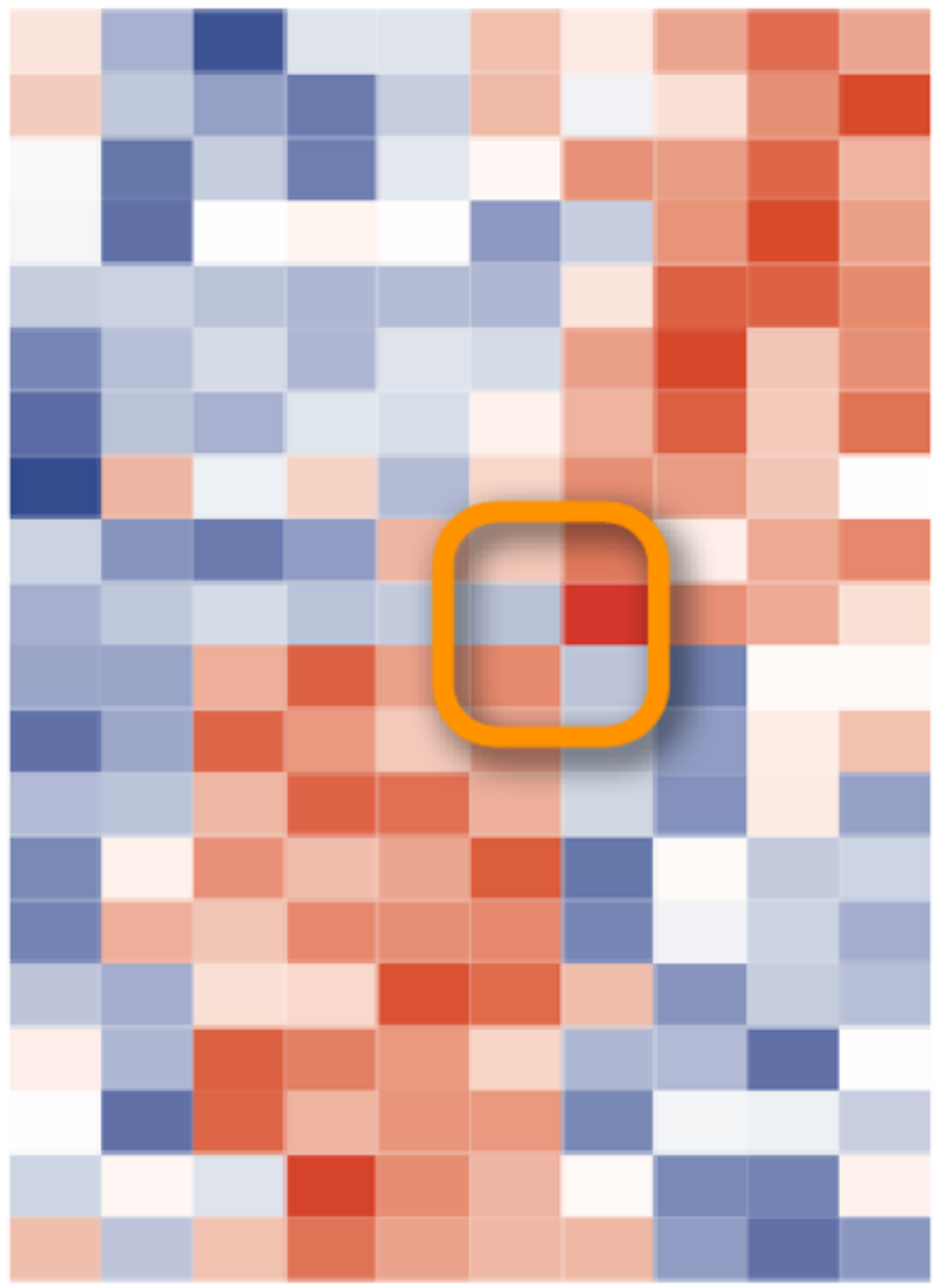


Normal Vision

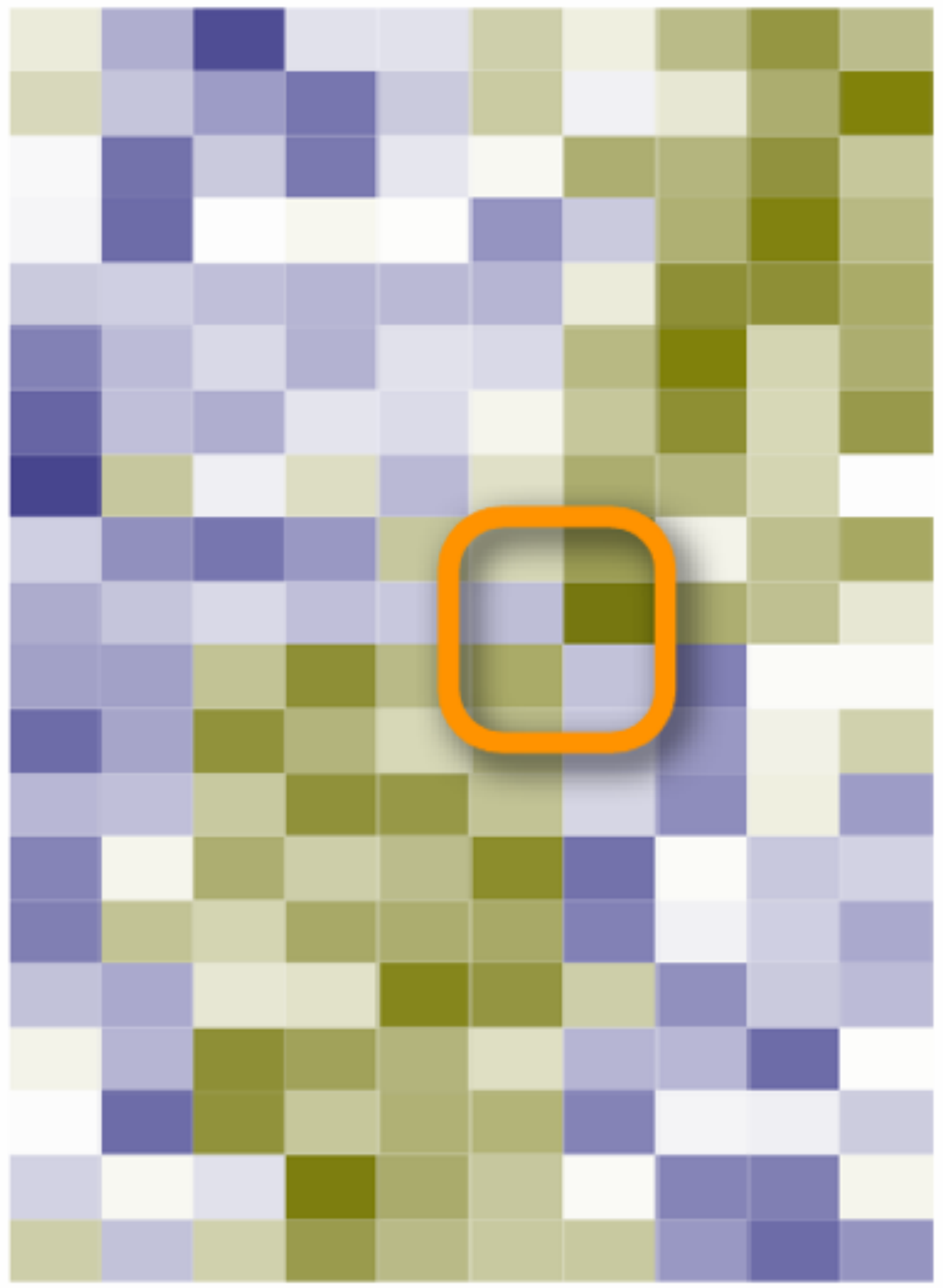


Deuteranope Vision
("Red-Green Blindness")

Good Color Mapping

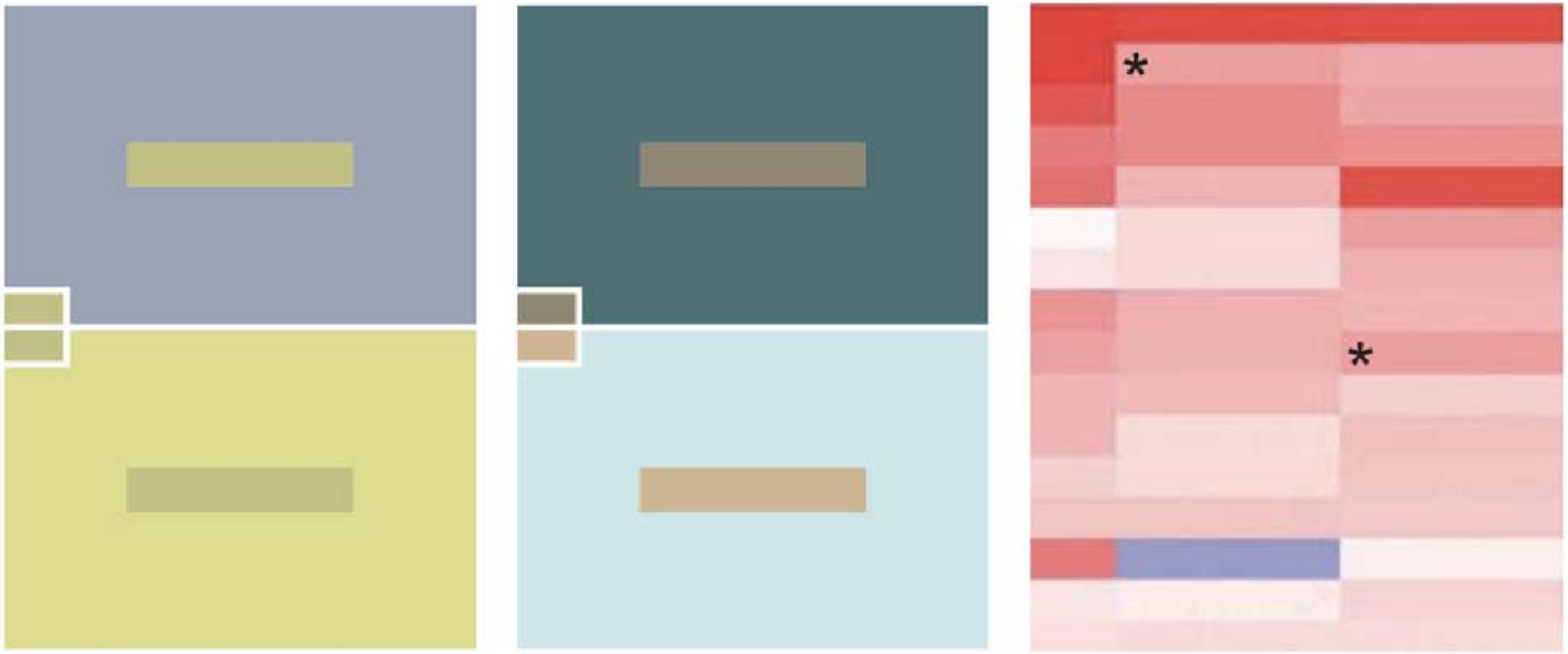


Normal Vision



Deuteranope Vision
(“Red-Green Blindness”)

Color is relative!



Clustering

Classification of items into "similar" bins

Based on similarity measures

- Euclidean distance, Pearson correlation, ...

Partitional Algorithms

- divide data into set of bins
- # bins either manually set (e.g., k-means) or automatically determined (e.g., affinity propagation)

Hierarchical Algorithms

- Produce "similarity tree" – dendrogram

Bi-Clustering

- Clusters dimensions & records

Fuzzy clustering

- allows occurrence of elements in multiples clusters

Clustering Applications

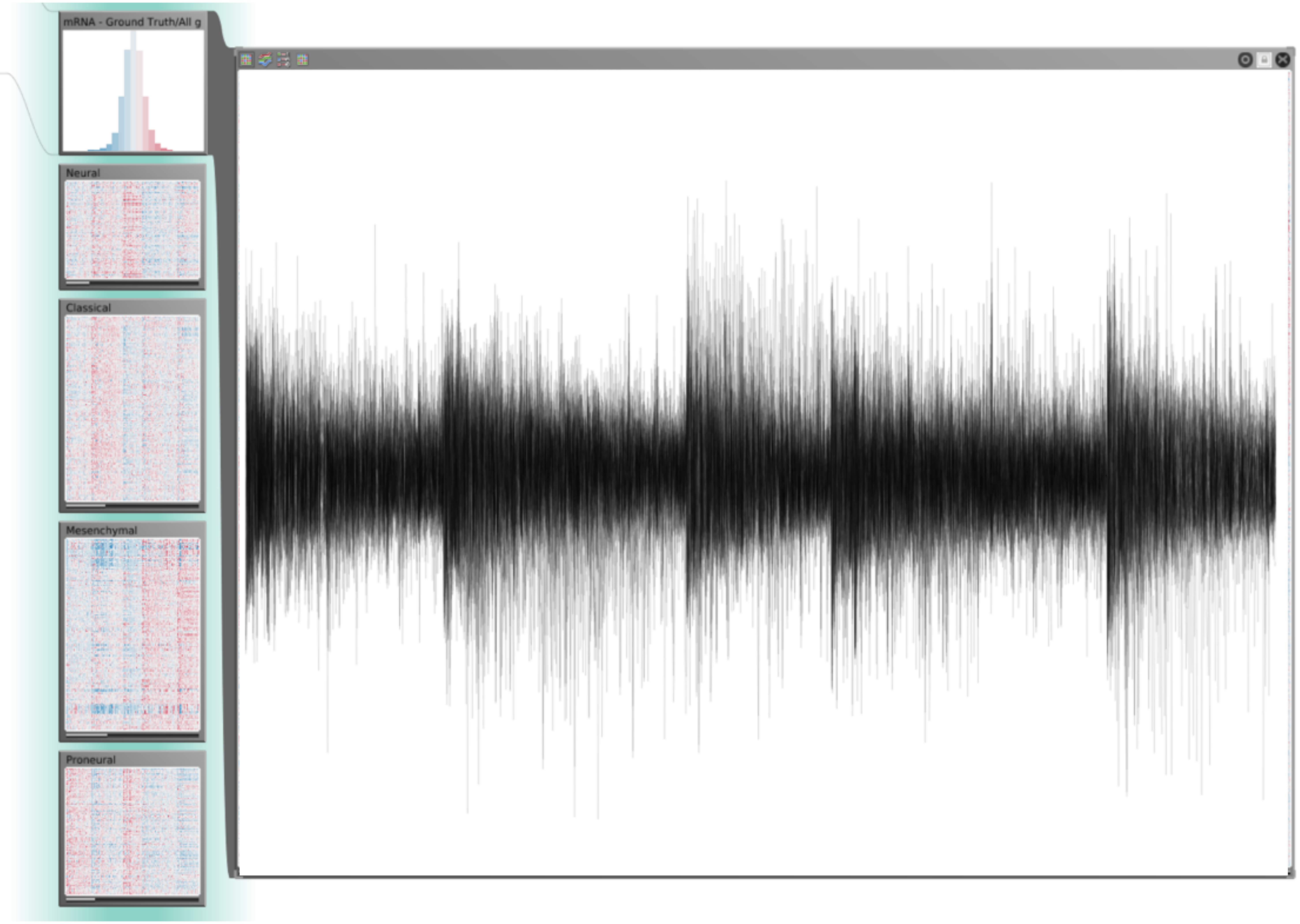
Clusters can be used to

- order (pixel based techniques)
- brush (geometric techniques)
- aggregate

Aggregation

- cluster more homogeneous than whole dataset
- statistical measures, distributions, etc. more meaningful

Clustered Heat Map



Dimensionality Reduction

Dimensionality Reduction

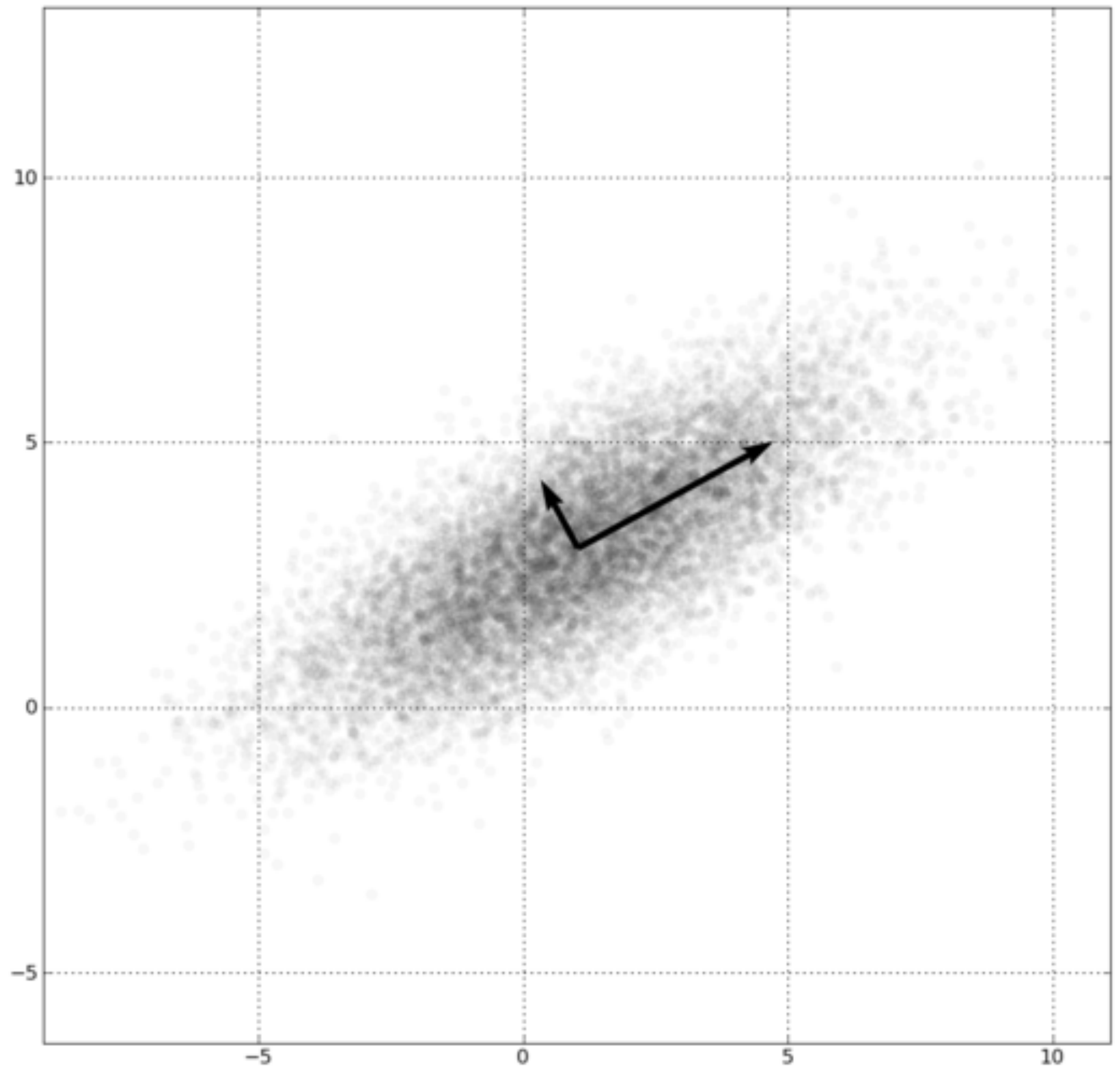
Reduce high dimensional to lower dimensional space

Preserve as much of variation as possible

Plot lower dimensional space

Principal Component Analysis (PCA)

linear mapping, by order of variance



Multidimensional Scaling

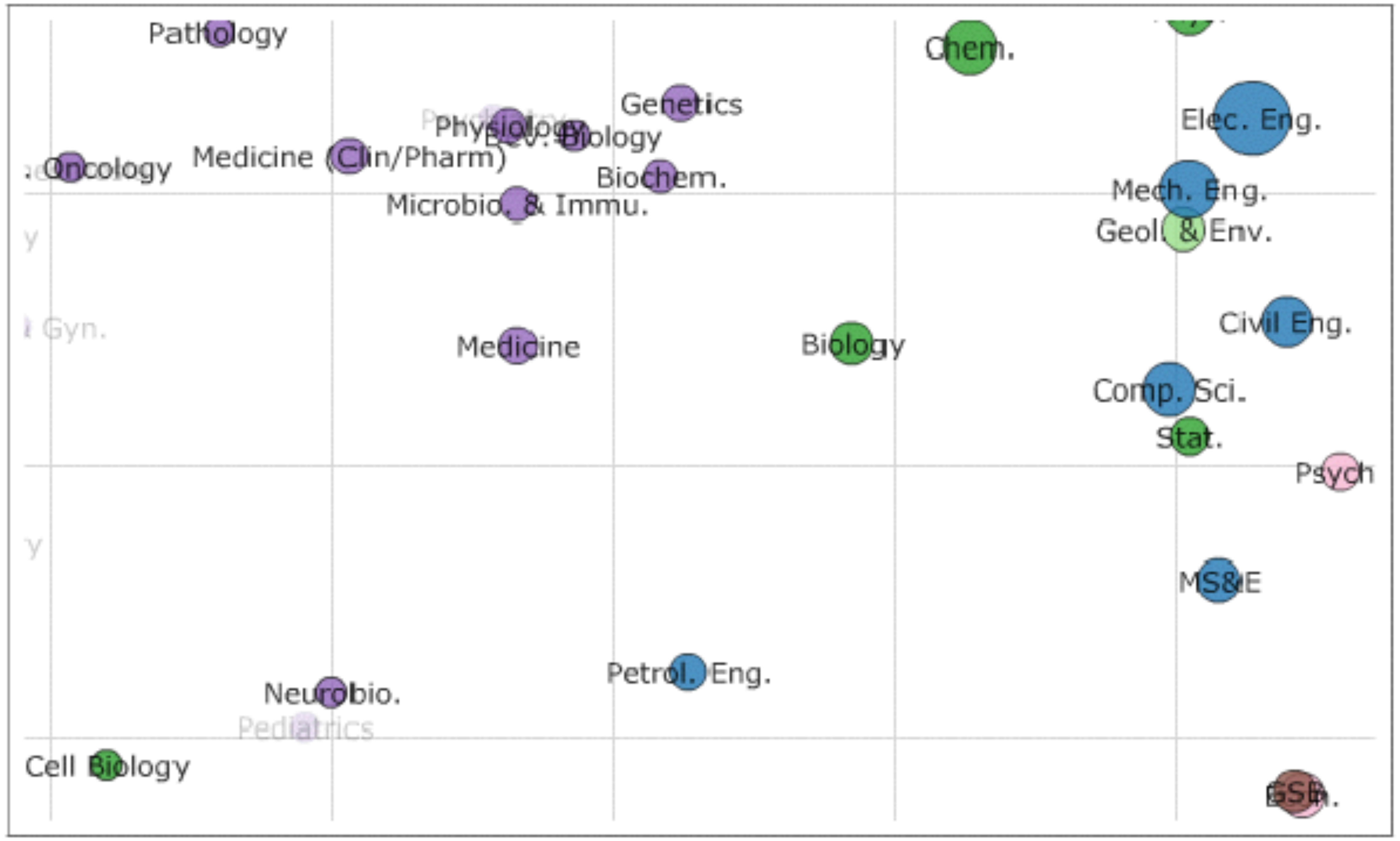
Nonlinear, better suited for some DS

Popular for text analysis

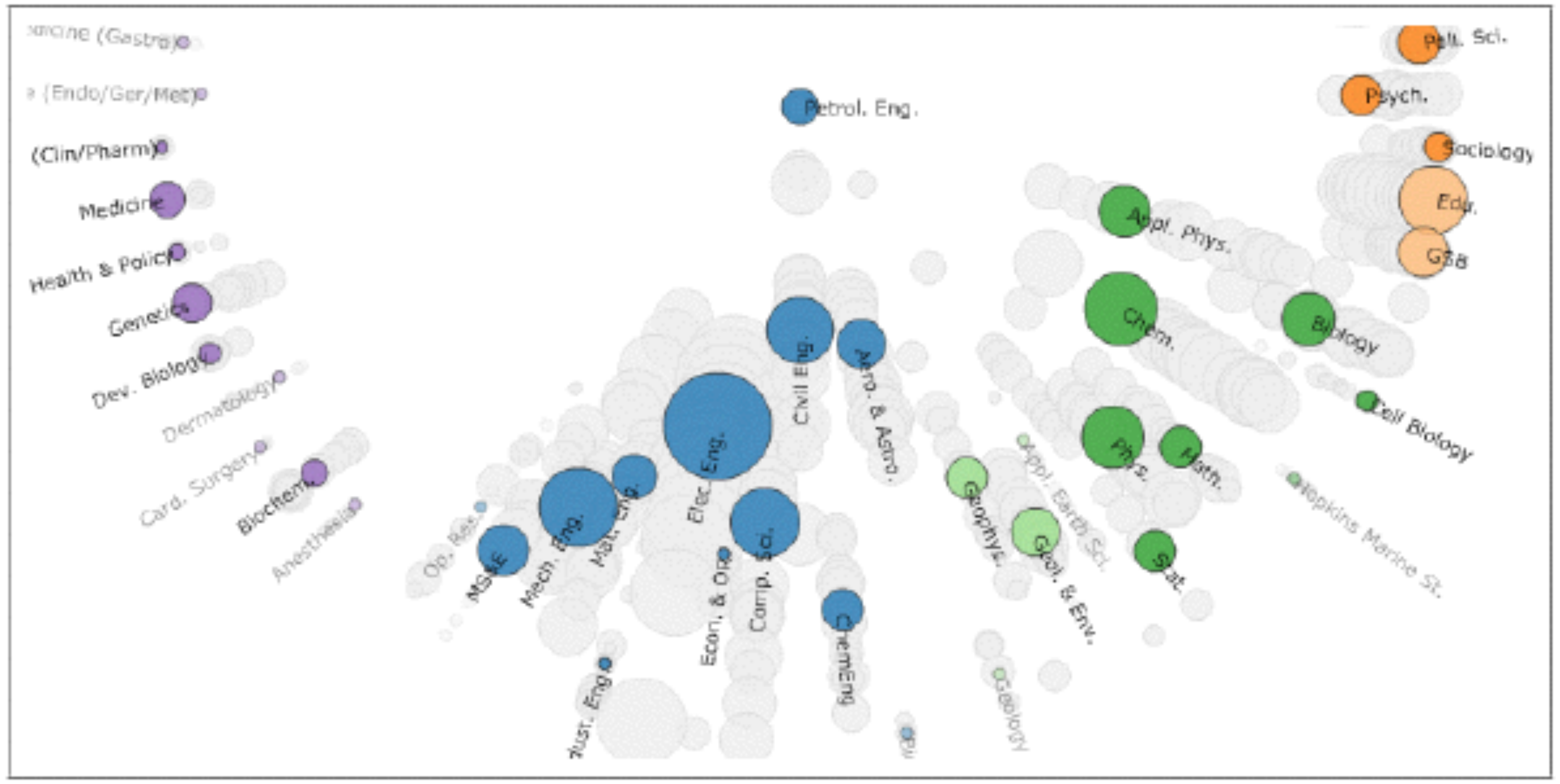
[Doerk 2011]

Can we Trust Dimensionality Reduction?

Topical distances between departments in a 2D projection



Topical distances between the selected Petroleum Engineering and the others.

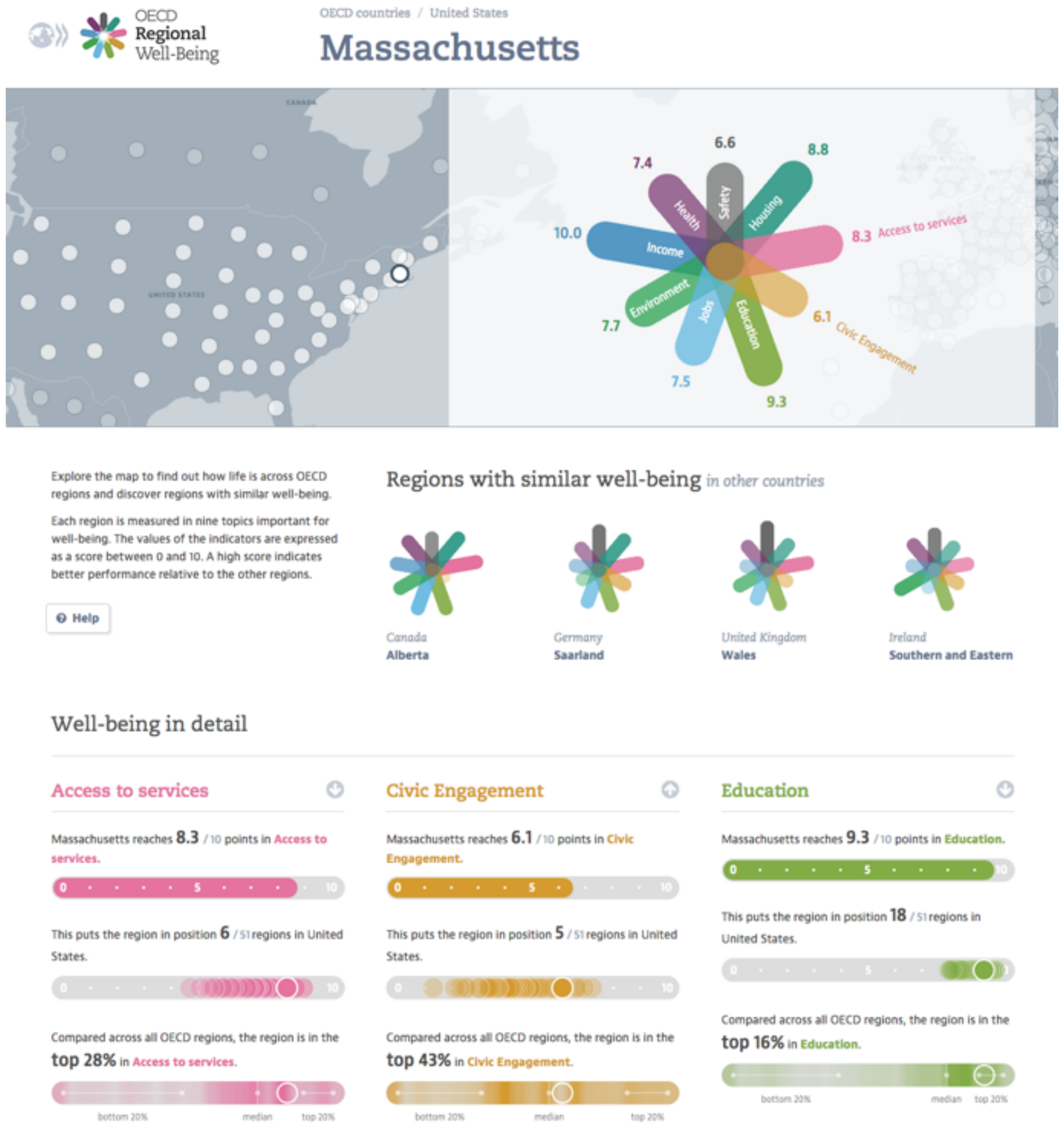


[Chuang et al., 2012]

http://www-nlp.stanford.edu/projects/dissertations/browser.html

Design Critique

OECD: http://goo.gl/QfxHfv



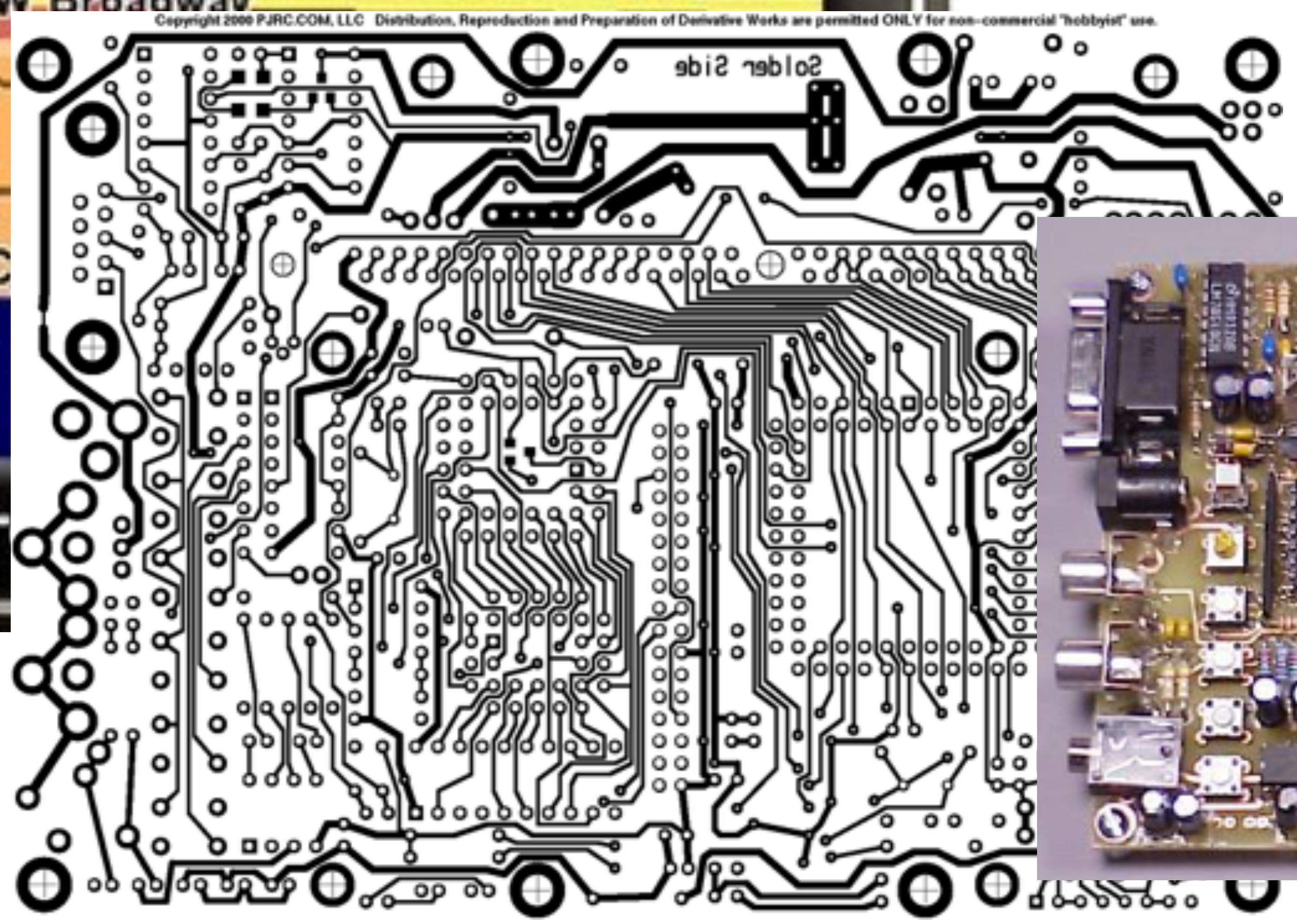
http://www.oecdregionalwellbeing.org/

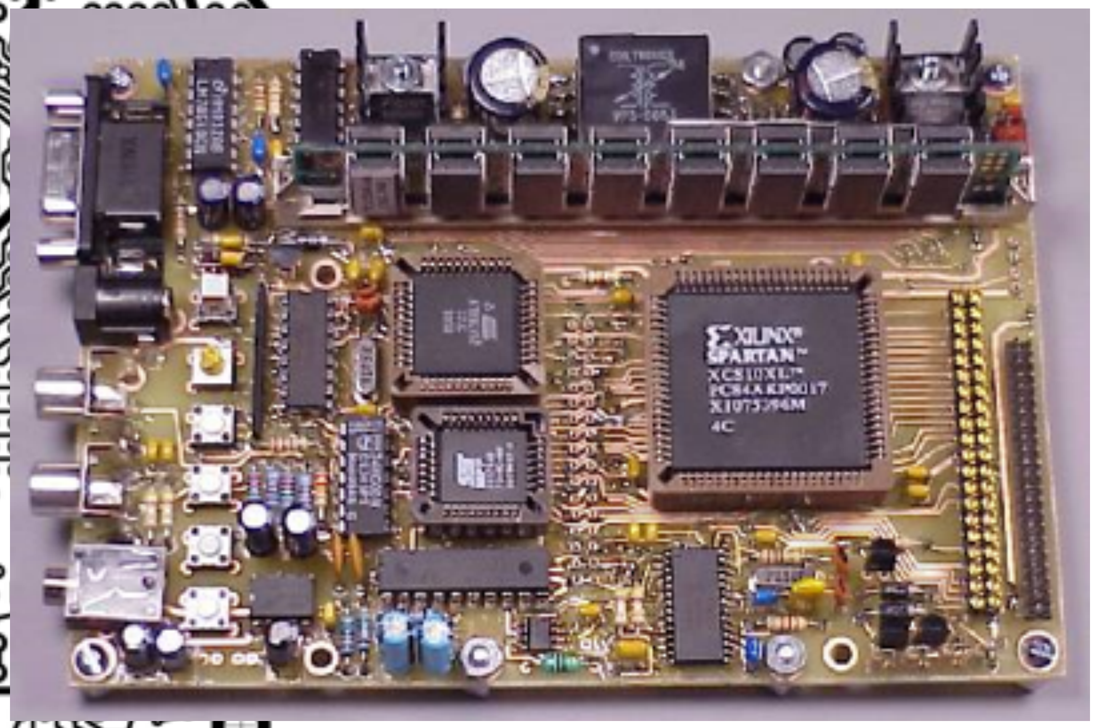
Graph Visualization

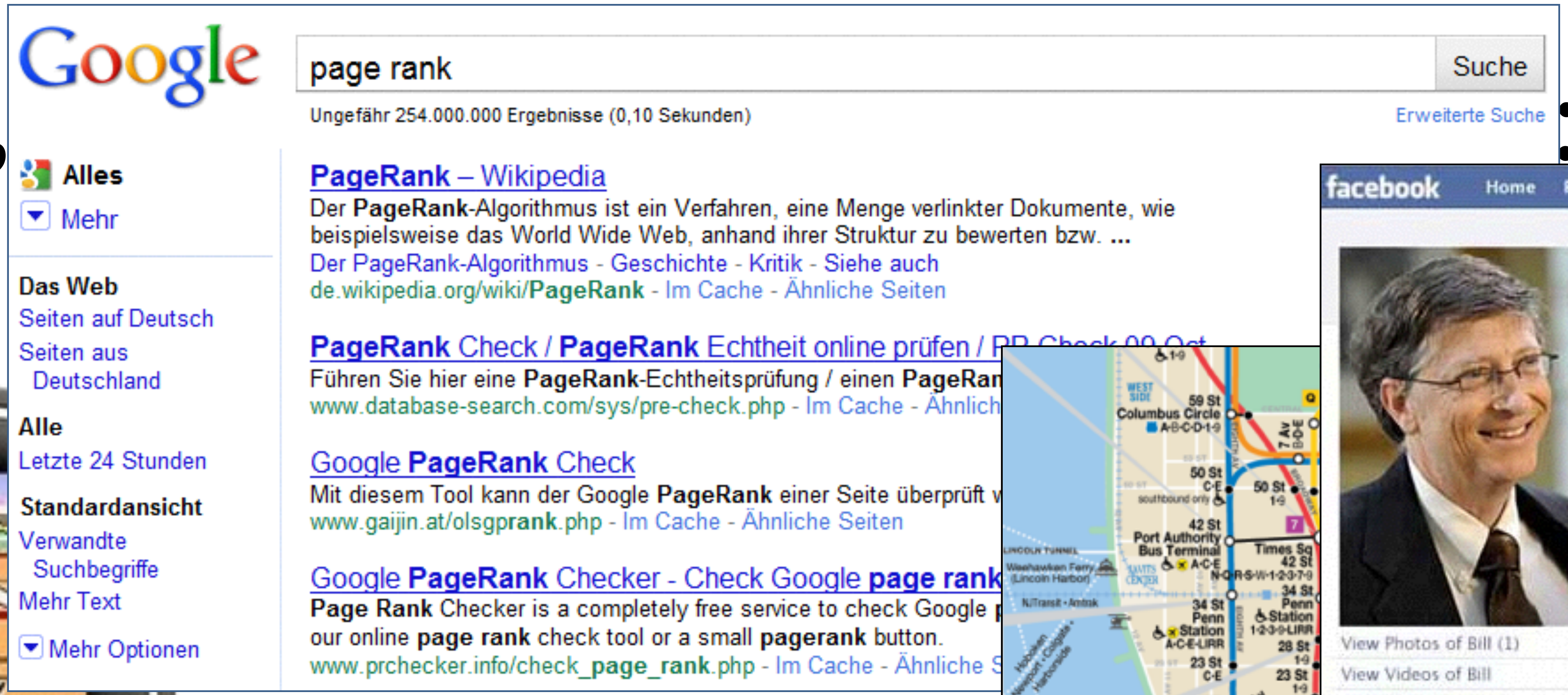
Based on Slides by HJ Schulz and M Streit

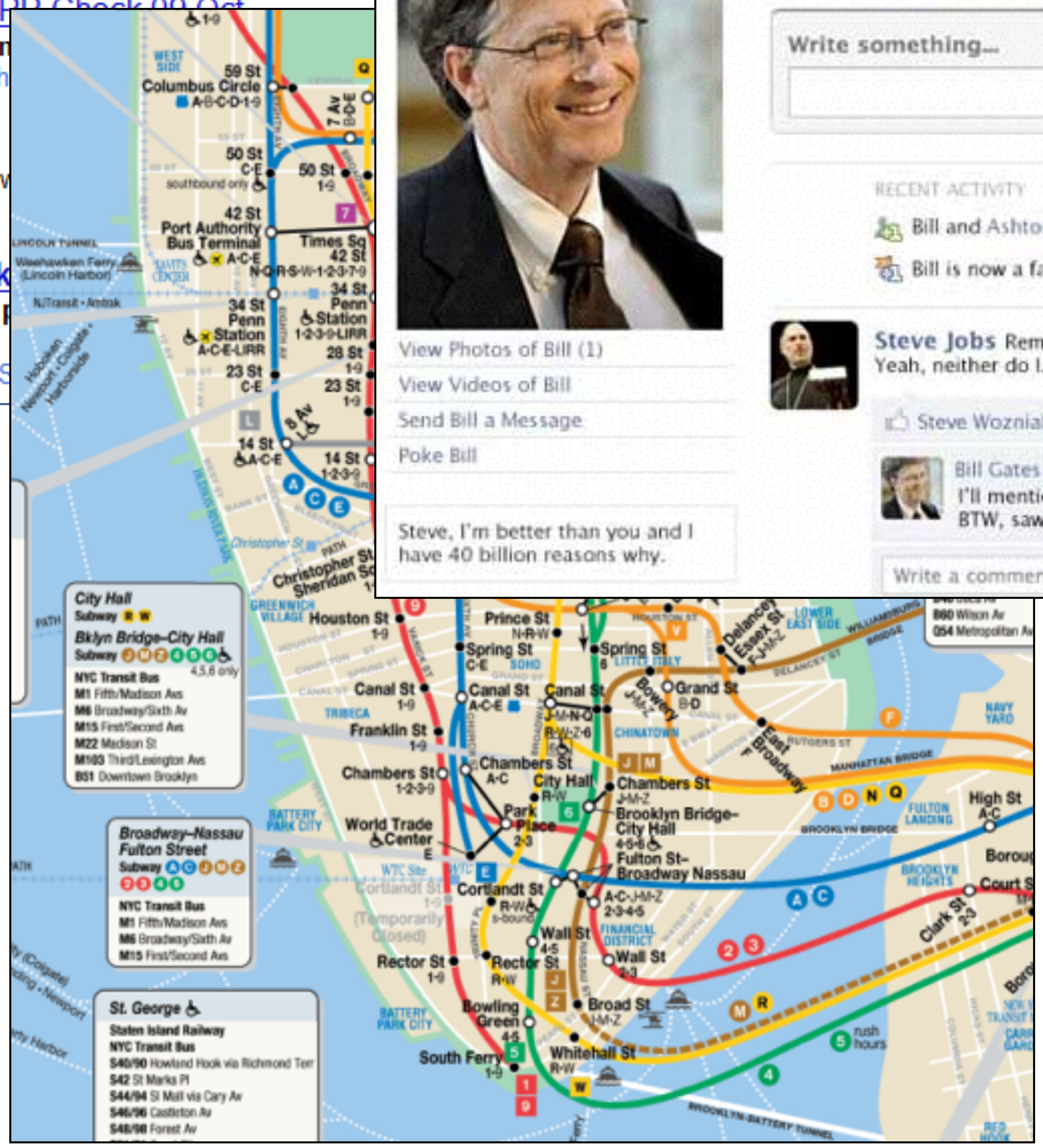
Applications of Graphs

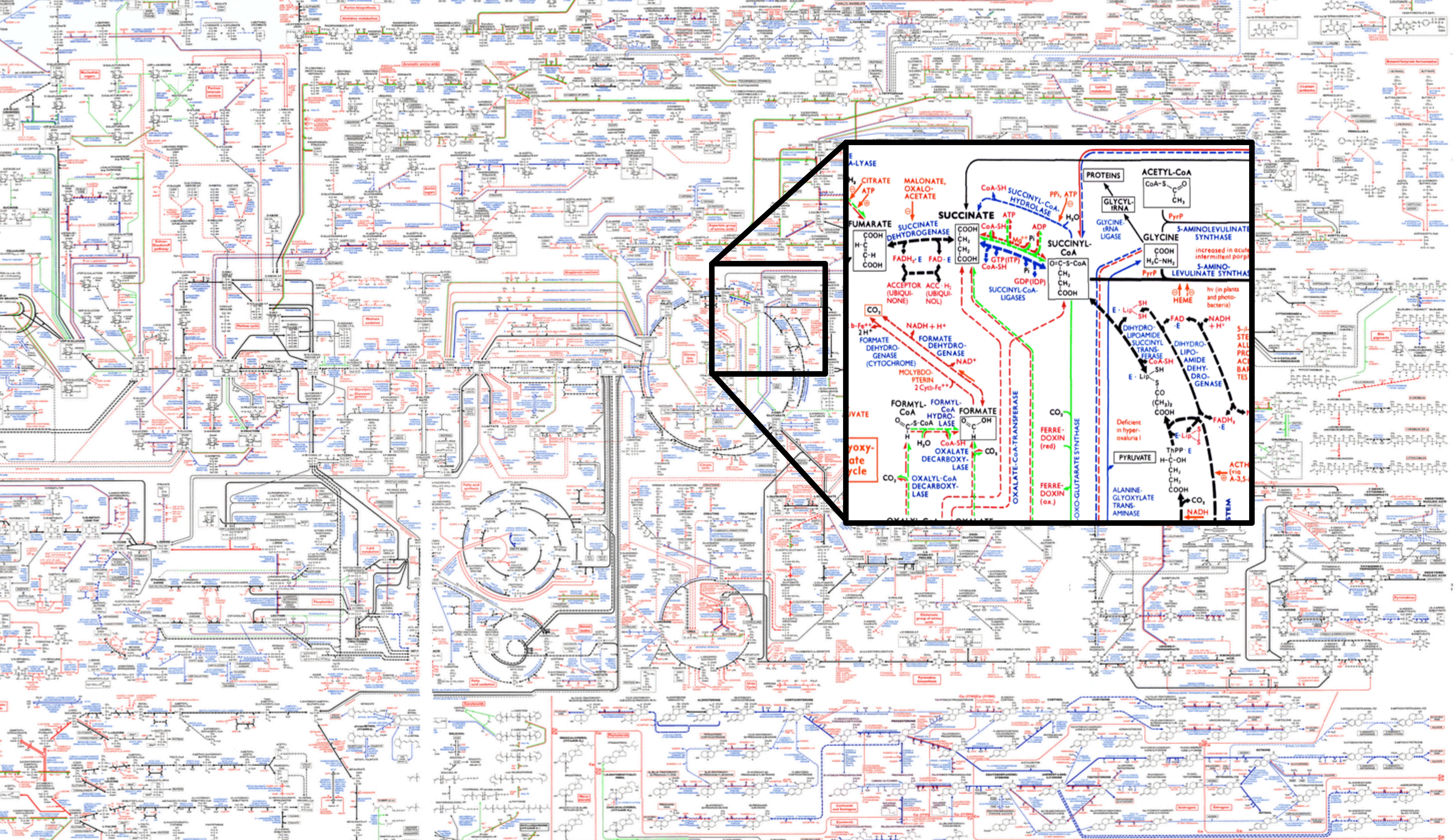
Without graphs,





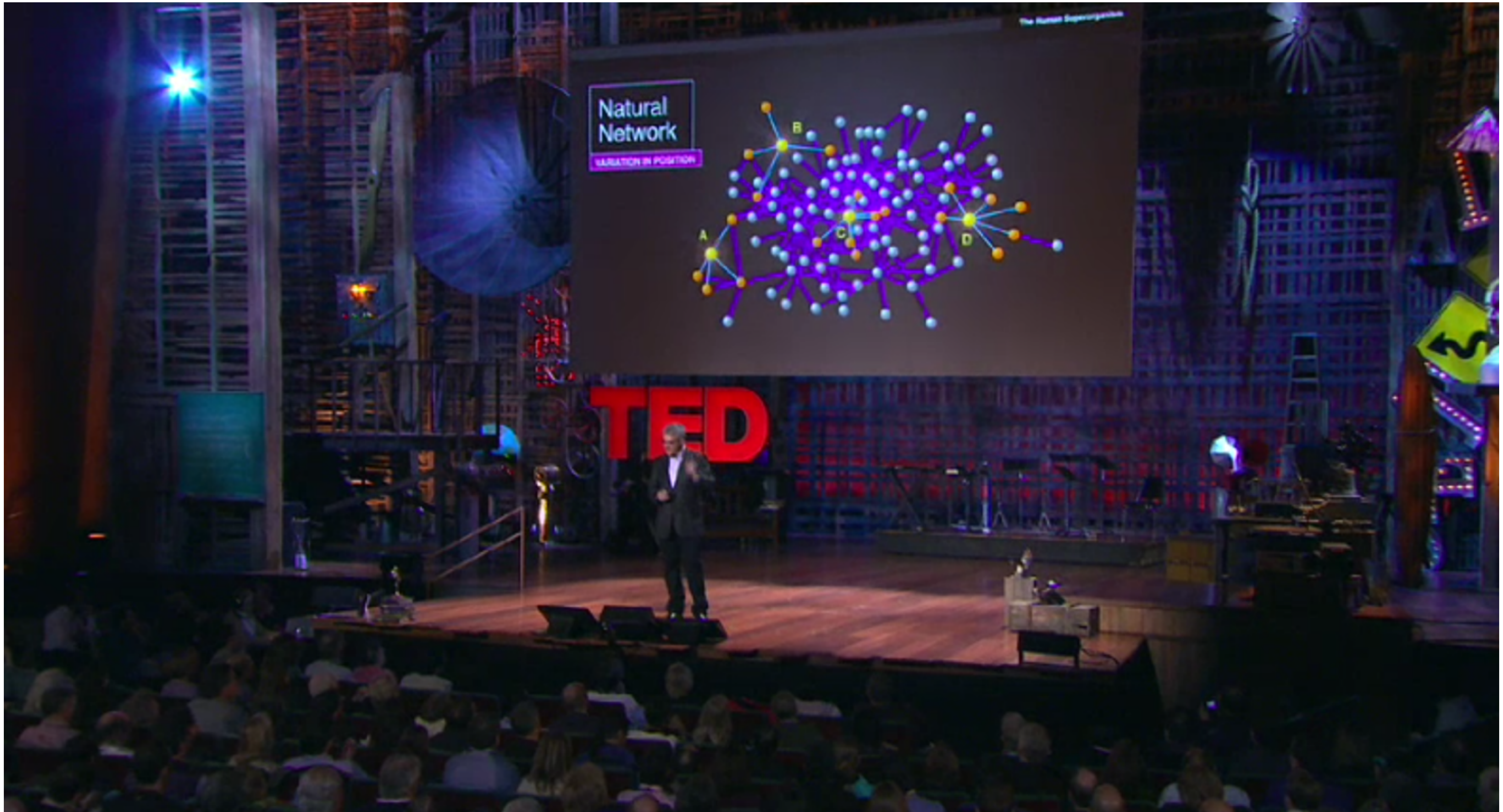





CITRATE
ATP
MALONATE, OXALO-ACETATE
SUCCINATE
CoA-SH
SUCCINYL-CoA-HYDROLASE
PPi, ATP
H2O
PROTEINS
ACETYL-CoA
GLYCYL-tRNA
FUMARATE
SUCCINATE DEHYDROGENASE
ATP
ADP
Pi
SUCCINYL-CoA
GLYCINE-tRNA LIGASE
GLYCINE
PyrP
5-AMINOLEVULINATE SYNTHASE
FADH2 · E
FAD · E
GTP (ITP)
CoA-SH
GDP (IDP)
increased in acute intermittent porph
5-AMINO-LEVULINATE SYNTHAS
ACCEPTOR (UBIQUINONE)
ACC · H2 (UBIQUINOL)
SUCCINYL-CoA-LIGASES
HEME
hν (in plants and photo-bacteria)
CO2
NADH + H+
FORMATE DEHYDROGENASE (CYTOCHROME)
FORMATE DEHYDRO-GENASE
NAD+
MOLYBDO-PTERIN
DIHYDRO-LIPOAMIDE SUCCINYL TRANS-FERASE
DIHYDRO-LIPO-AMIDE DEHY-DRO-GENASE
FAD · E
NADH + H+
FORMYL-CoA
FORMYL-CoA HYDRO-LASE
FORMATE
H2O
CoA-SH
OXALATE DECARBOXY-LASE
CO2
OXALYL-CoA DECARBOXY-LASE
OXALATE-CoA-TRANSFERASE
FERRE-DOXIN (red)
FERRE-DOXIN (ox.)
OXO-GLUTARATE SYNTHASE
Deficient in hyper-oxaluria I
PYRUVATE
ALANINE-GLYOXYLATE TRANS-AMINASE
FADH2 · E
ThPP · E
CO2
NADH
Michal 2000

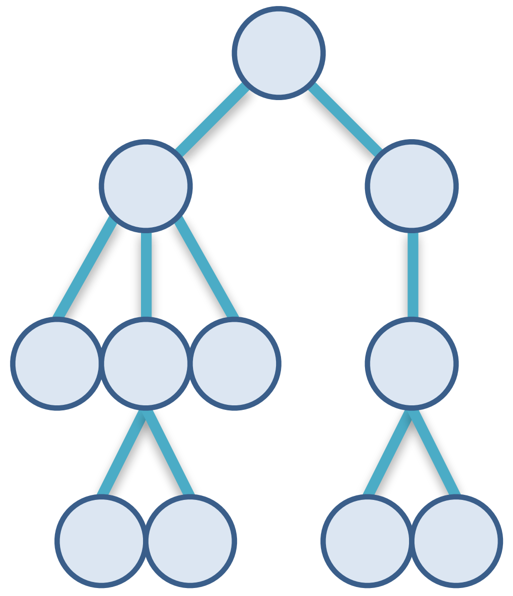

facebook
December 2010

Graph Visualization Case Study

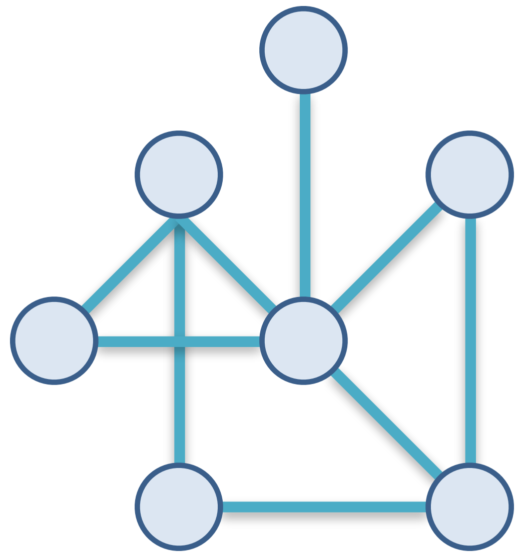


Graph Theory Fundamentals

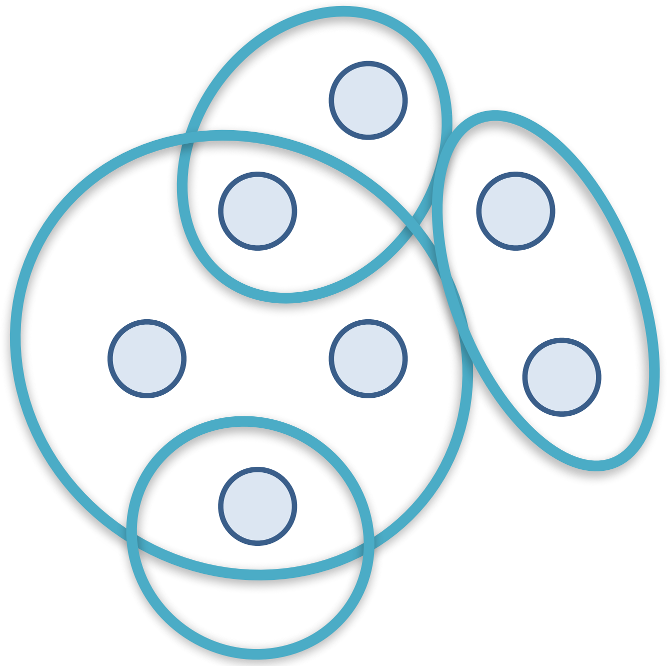
Tree



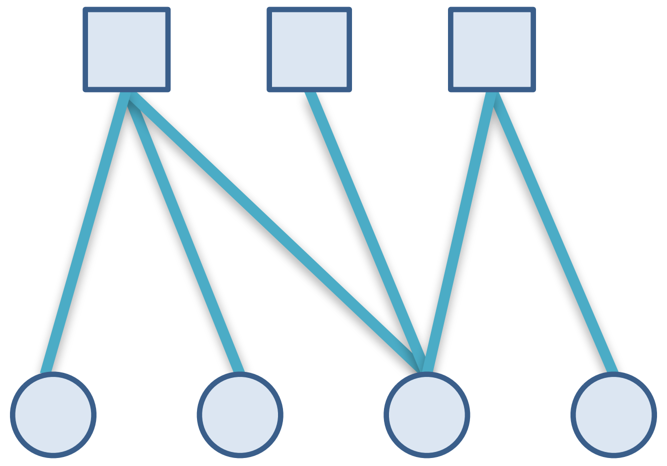
Network



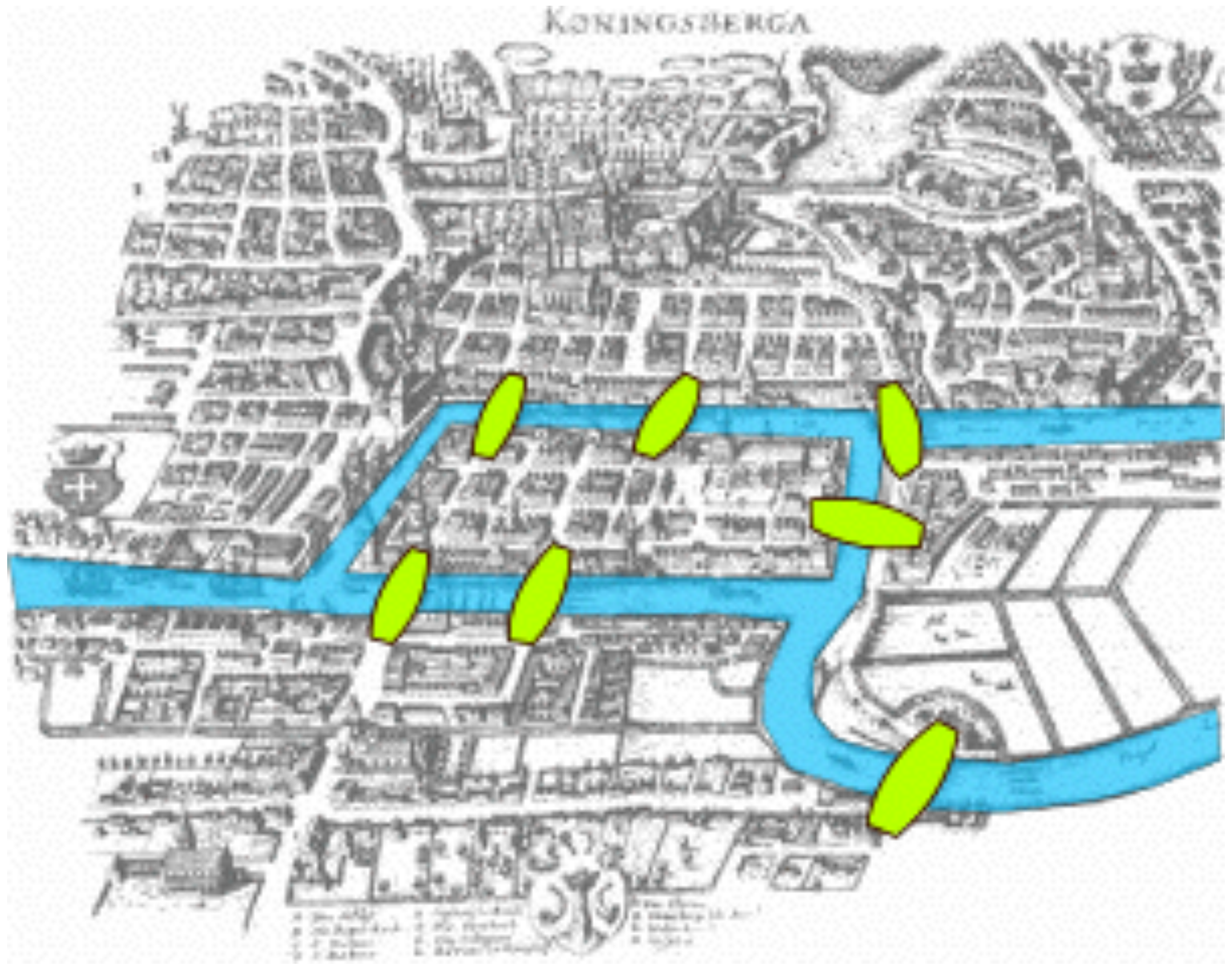
Hypergraph

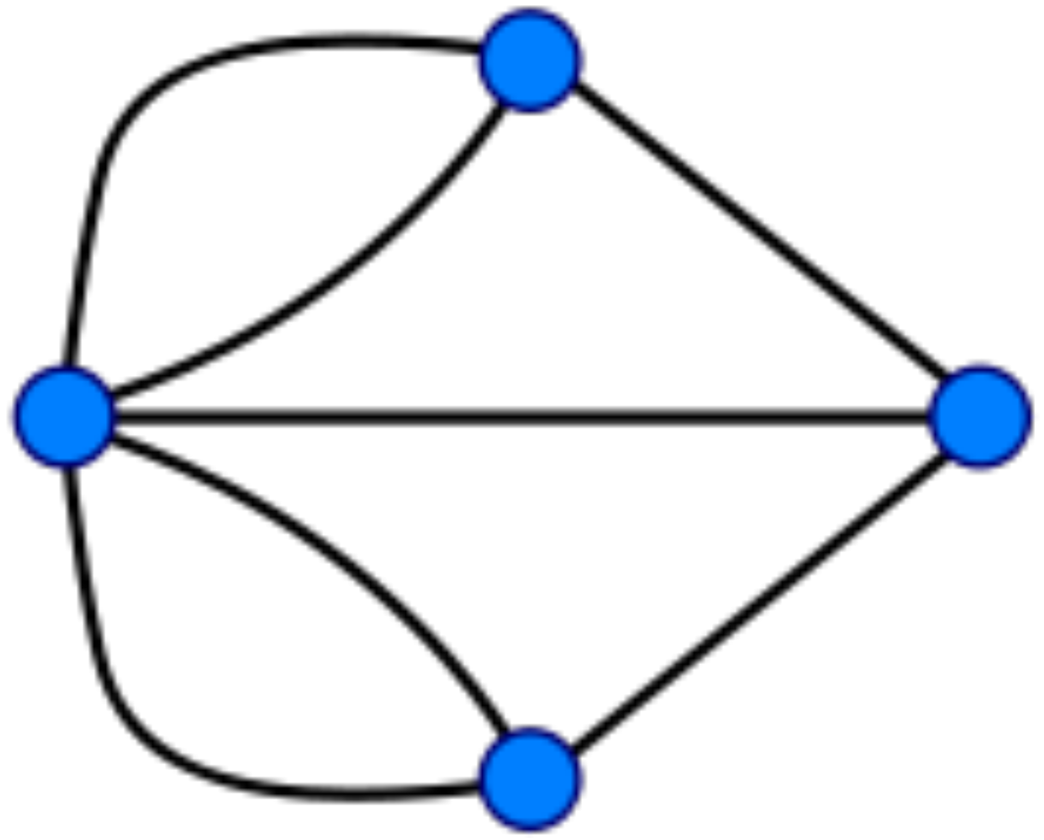


Bipartite Graph



Königsberg Bridge Problem (1736)



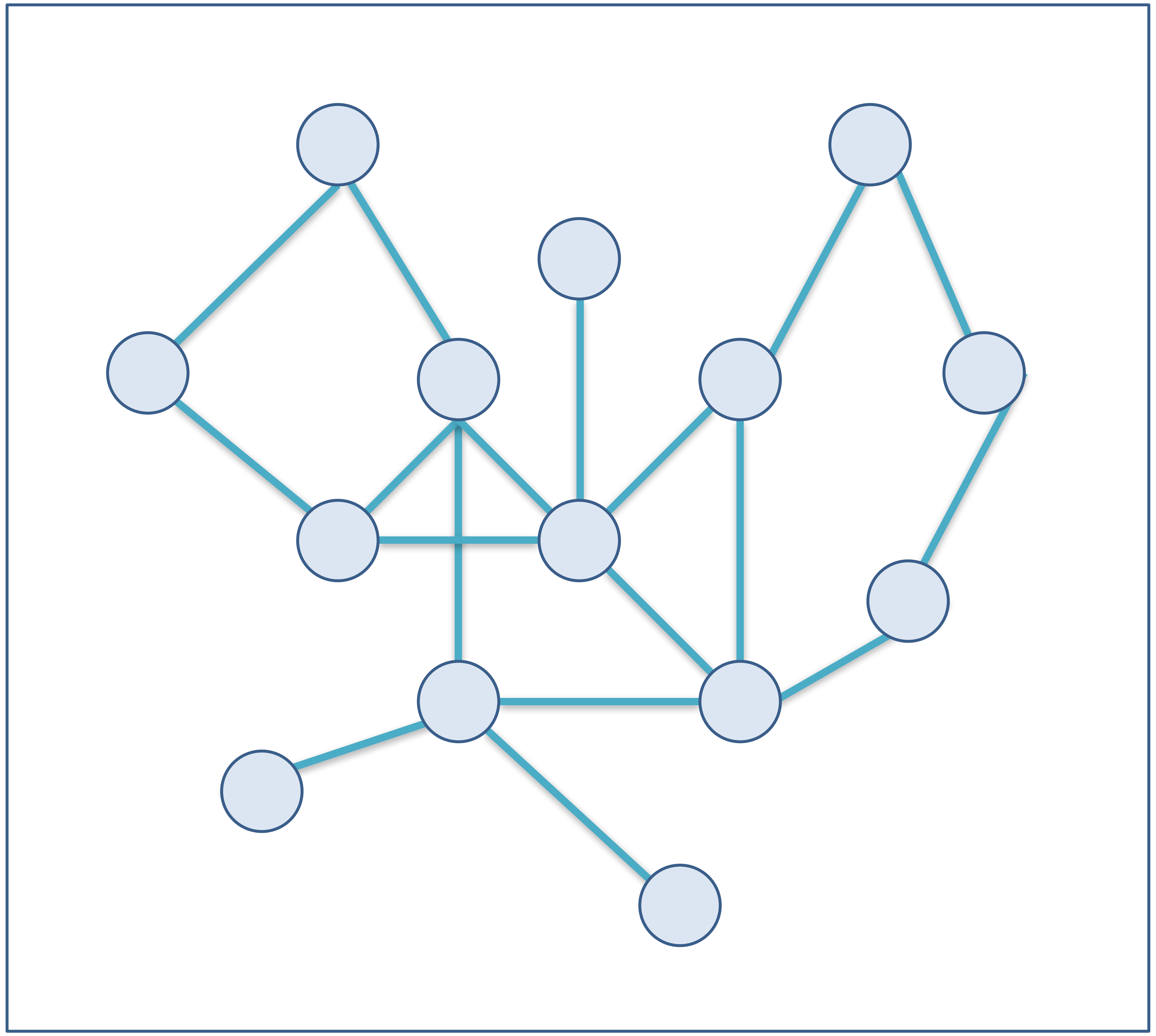


Find a Hamiltonian Path (path that visits each vertex exactly once).
Want to make 1 million $? Develop O(n^k) algorithm.

Graph Terms (1)

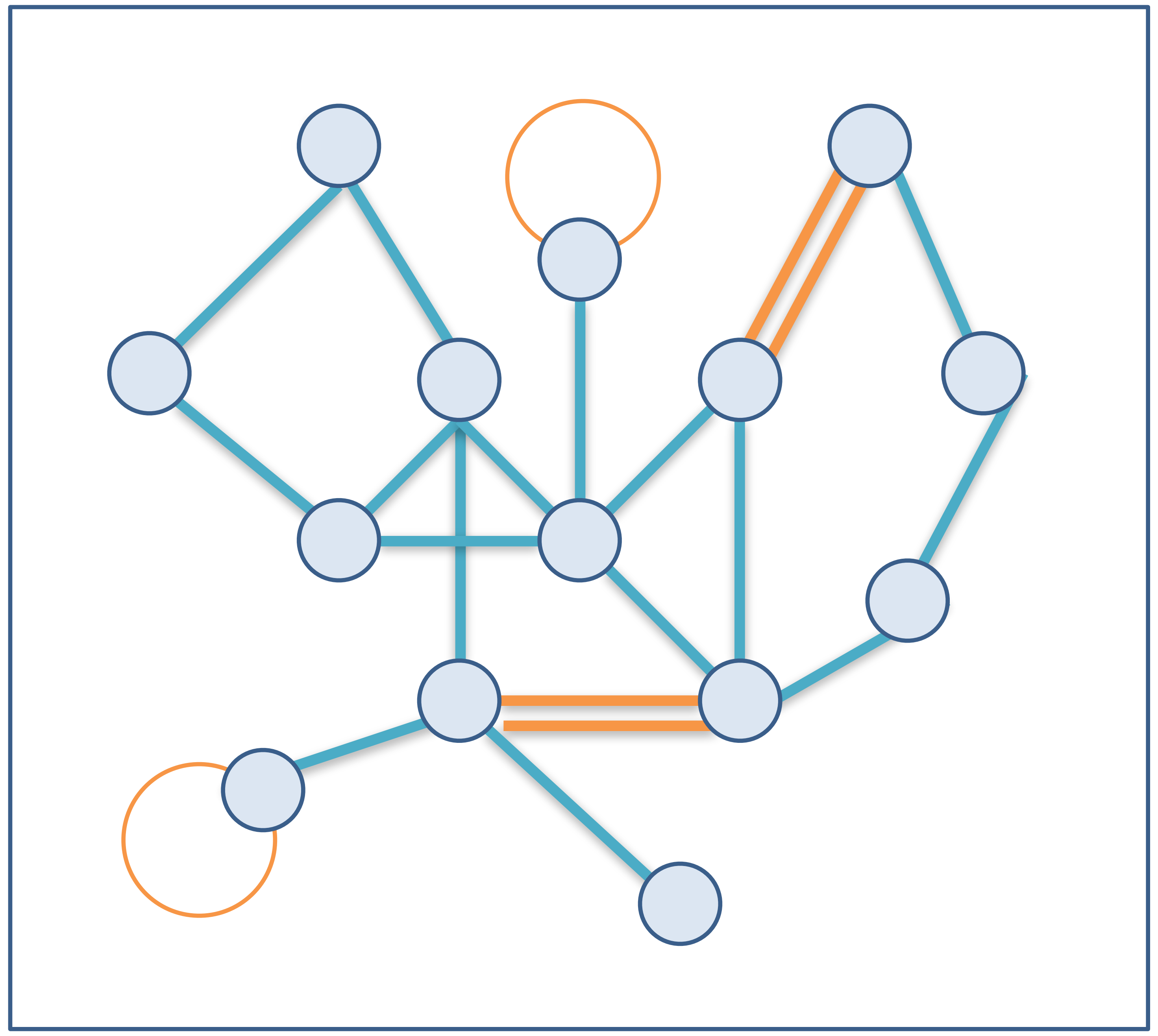
A graph **G(V,E)** consists of a set of **vertices V** (also called nodes) and a

set of **edges E** connecting these vertices.



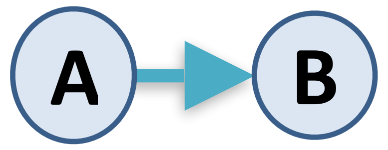
Graph Terms (2)

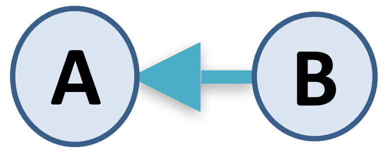
A simple graph G(V,E) is a graph which contains **no multi-edges** and **no loops**



Not a simple graph!
→ A ***general graph***

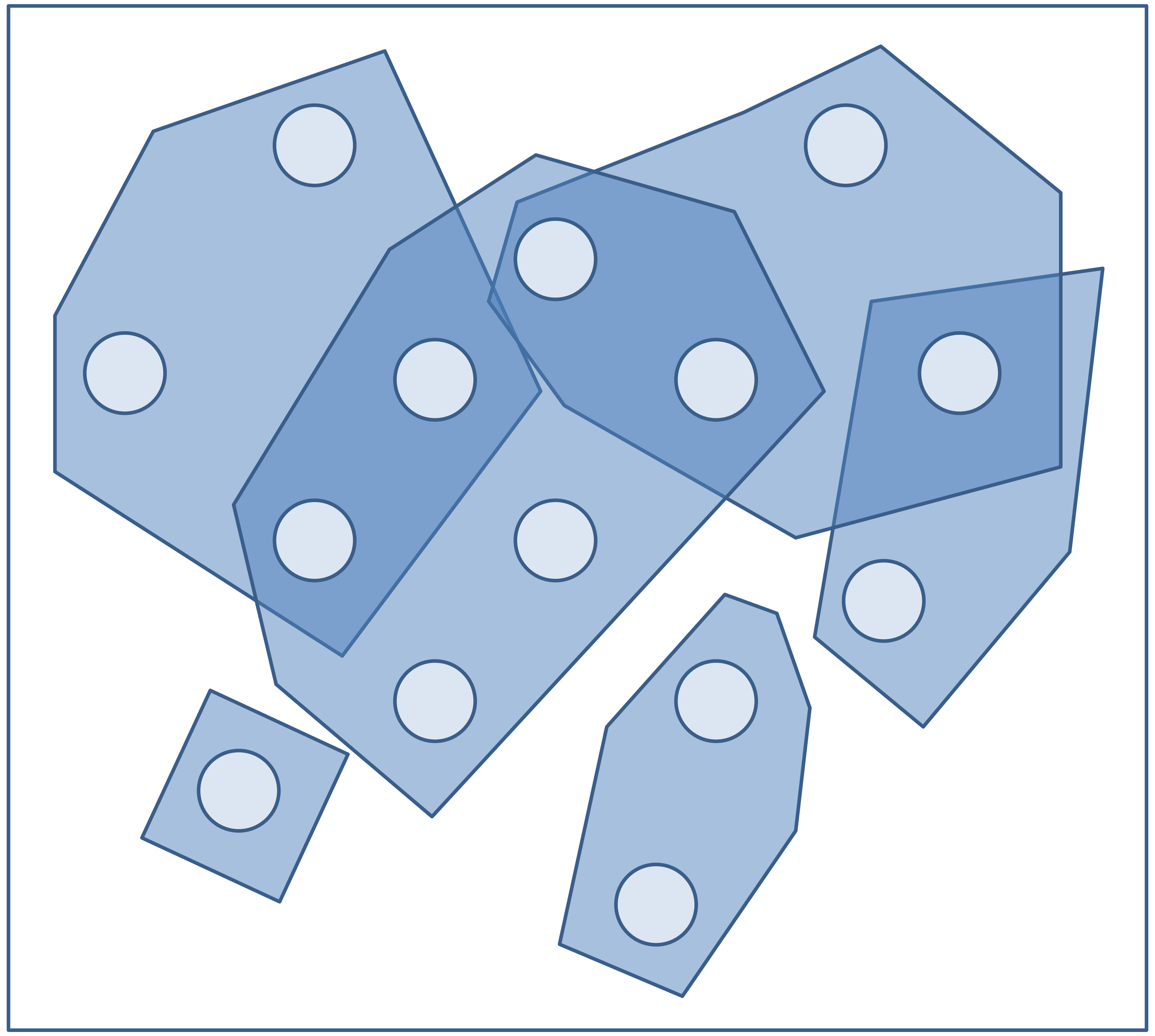
Graph Terms (3)

A directed graph (digraph) is a graph that discerns between the edges 

and 

.

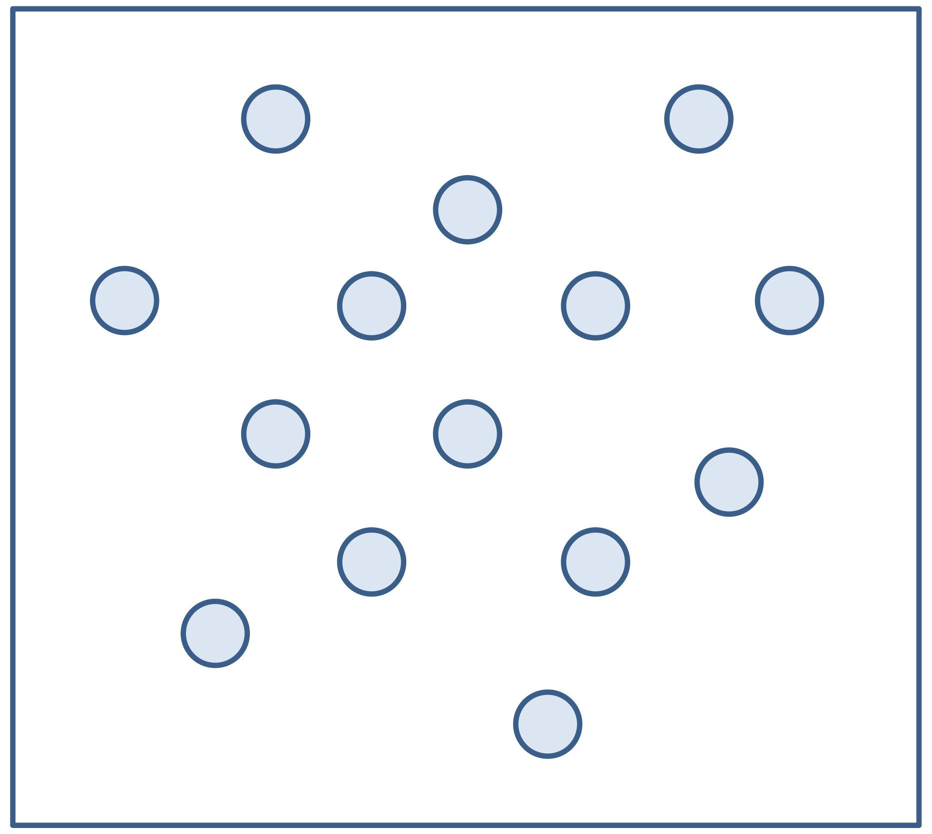
A hypergraph is a graph with edges connecting any number of vertices.



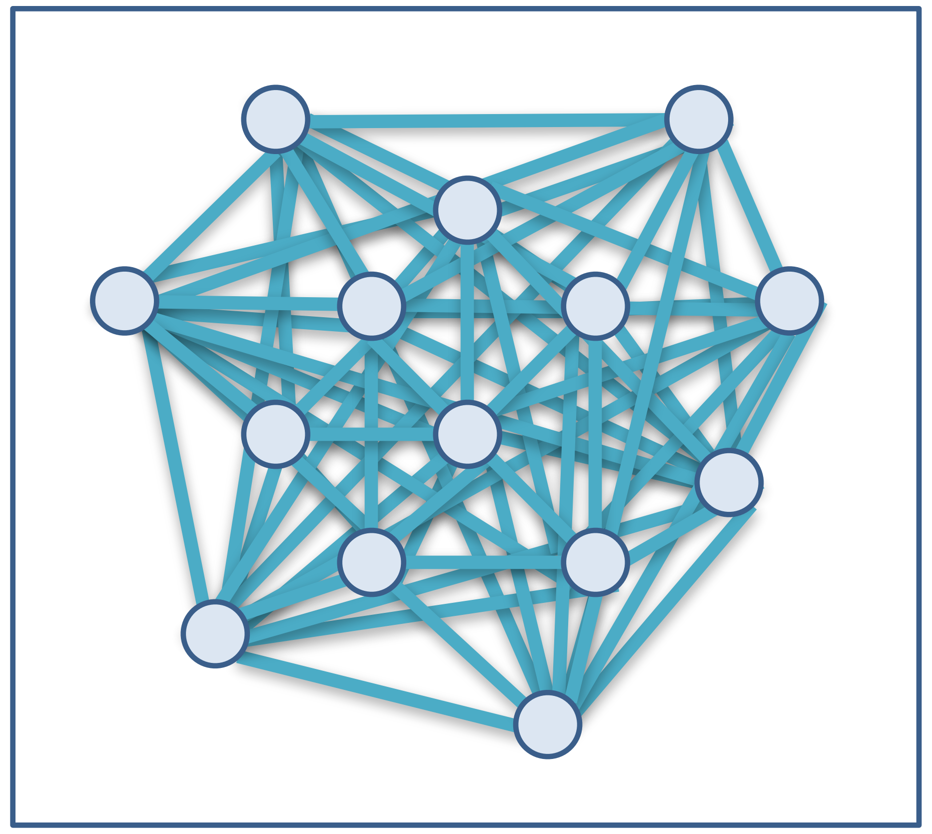
Hypergraph Example

Graph Terms (4)

Independent Set
G contains no edges


Independent Set

Clique
G contains all possible edges

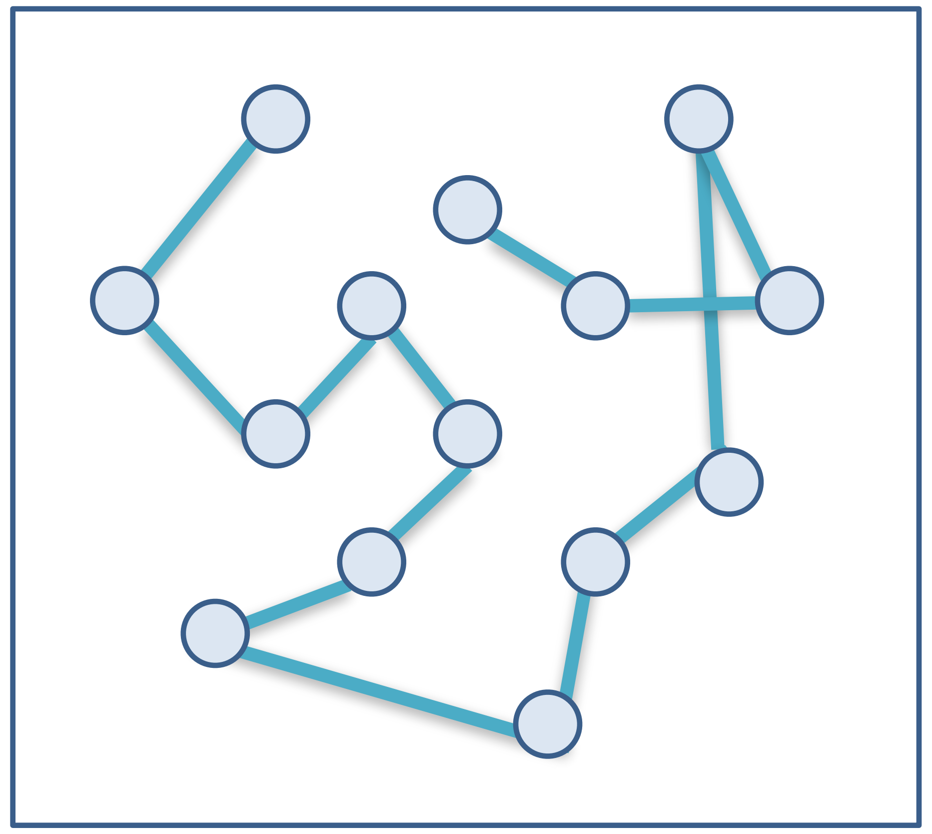

Clique

Graph Terms (5)

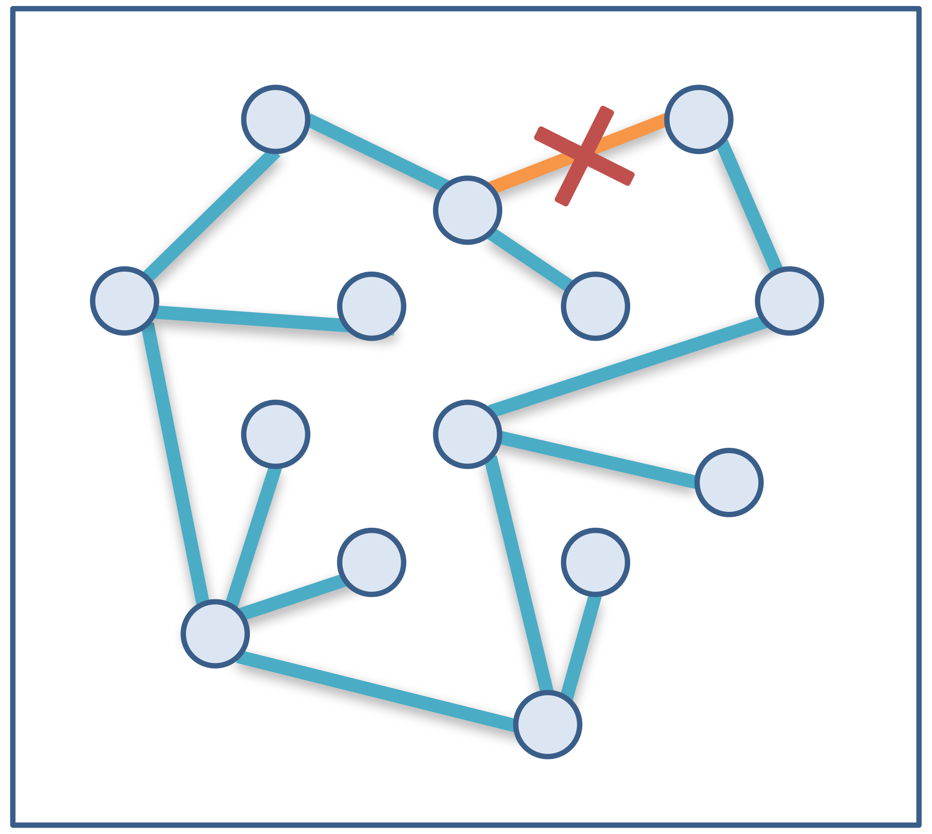
Path
G contains only edges that can be consecutively traversed

Tree
G contains no cycles

Network
G contains cycles



Path

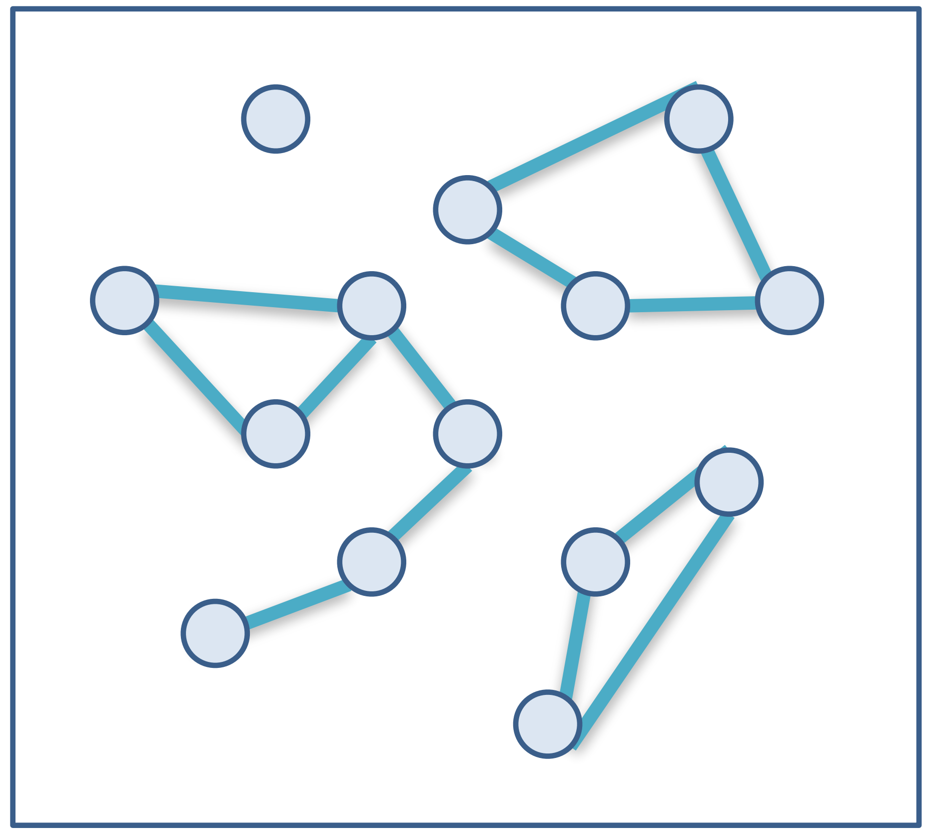


Tree

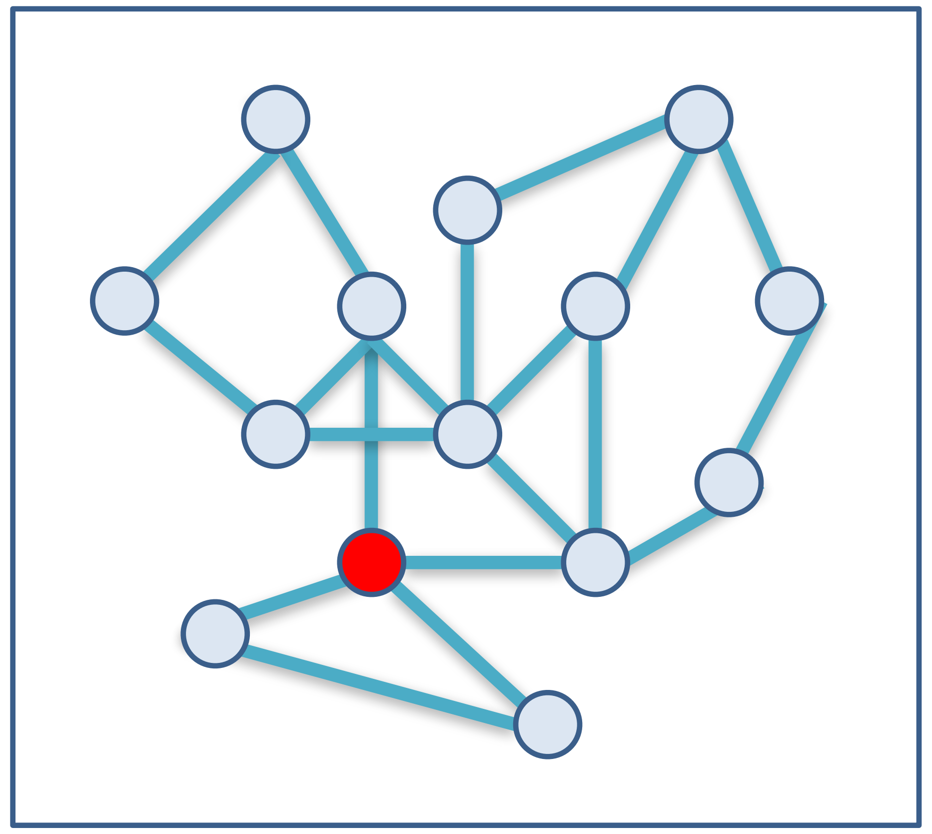
Graph Terms (6)

Unconnected graph
An edge traversal starting from a given vertex cannot reach any other vertex.

Articulation point
Vertices, which if deleted from the graph, would break up the graph in multiple sub-graphs.



Unconnected Graph

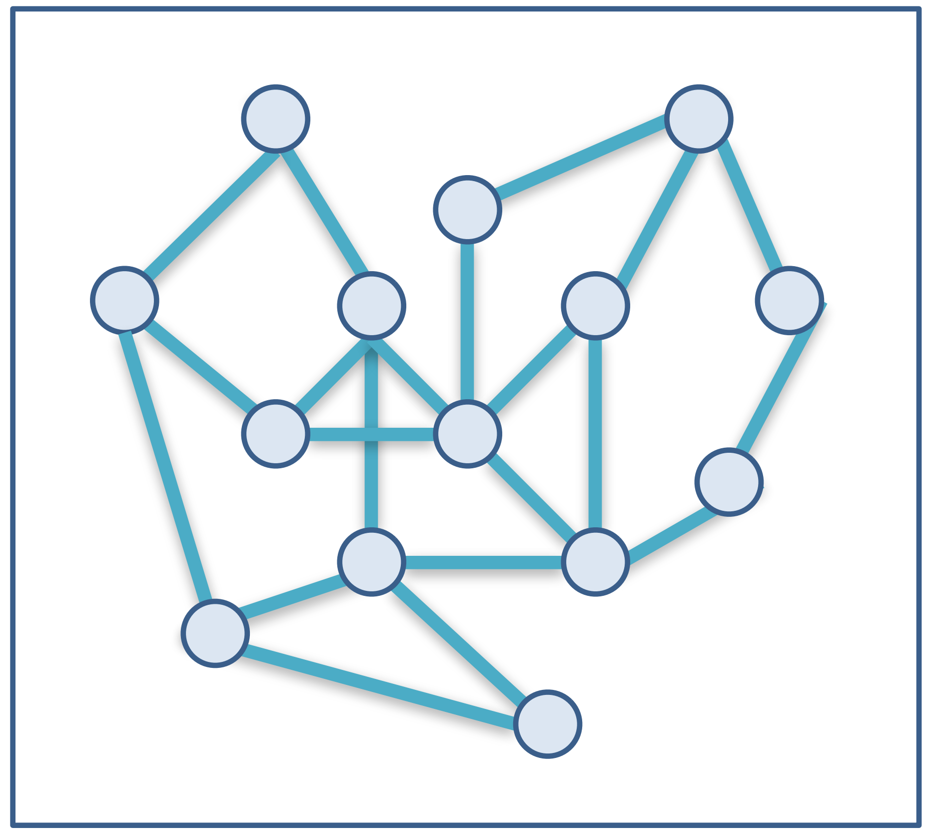


Articulation Point (red)

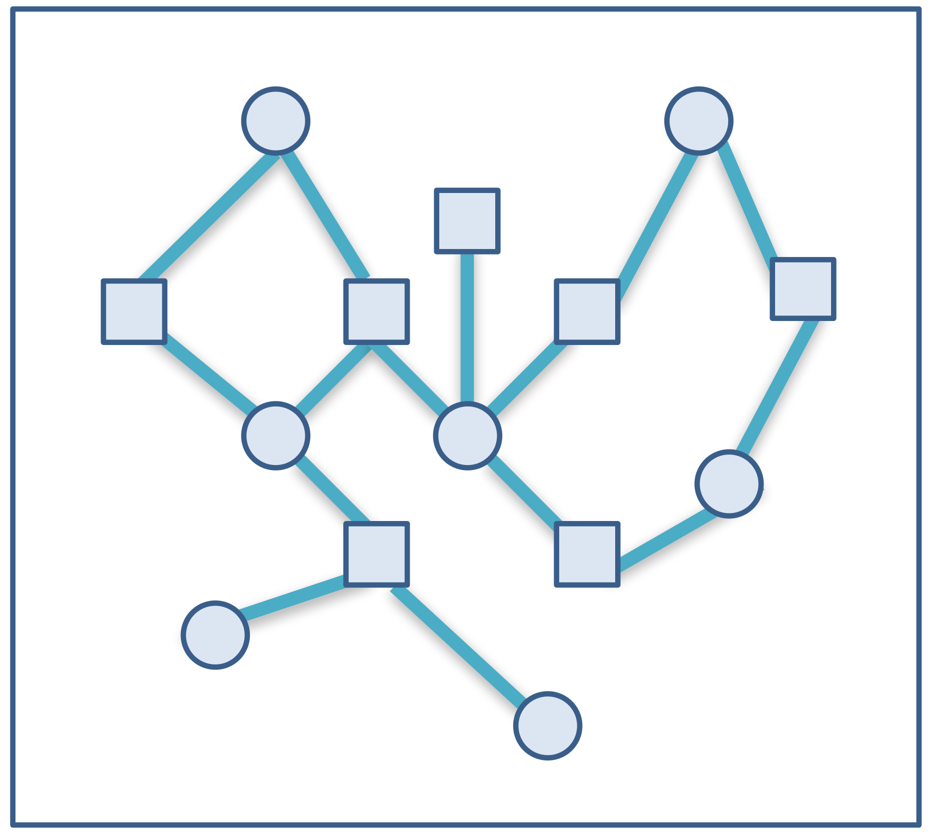
Graph Terms (7)

Biconnected graph
A graph without articulation points.

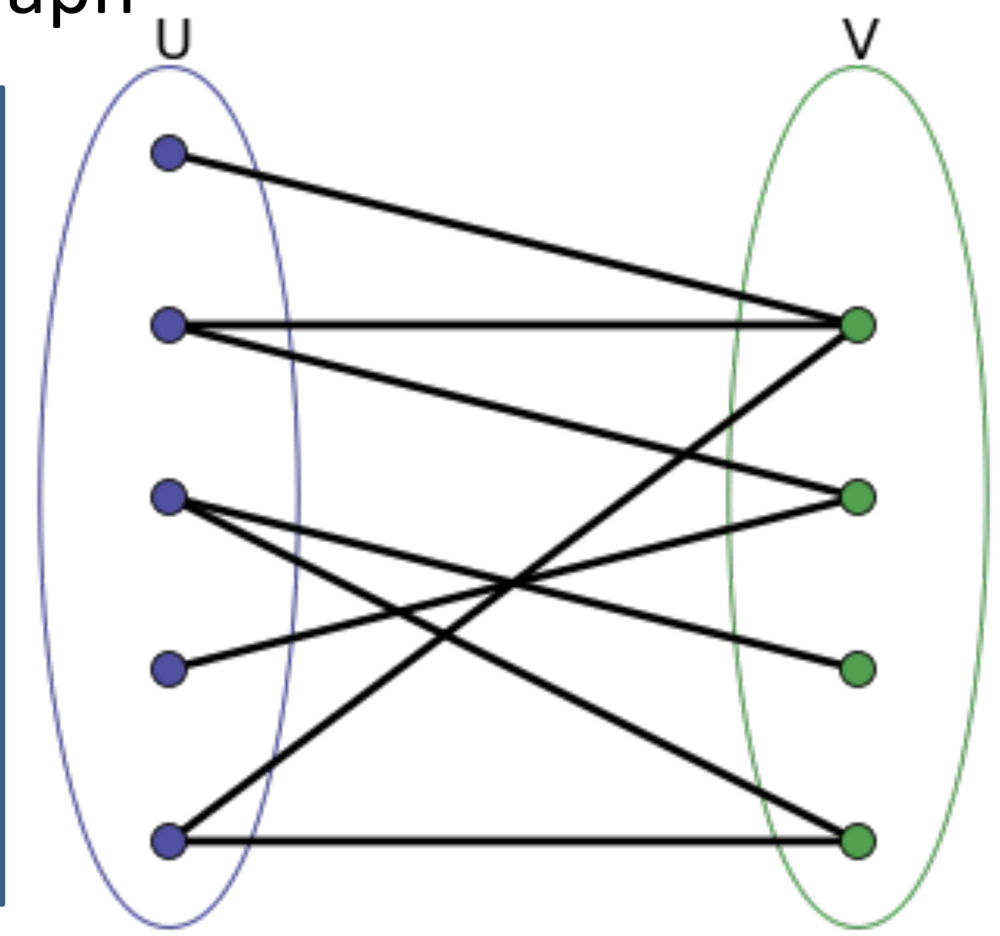
Bipartite graph
The vertices can be partitioned in two independent sets.



Biconnected Graph



Bipartite Graph

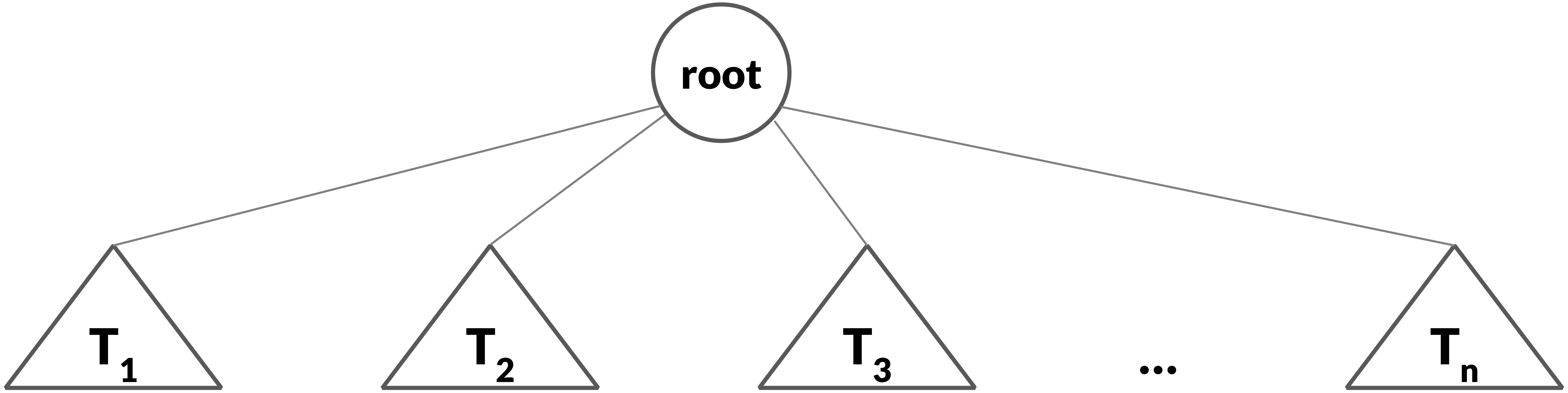


Tree

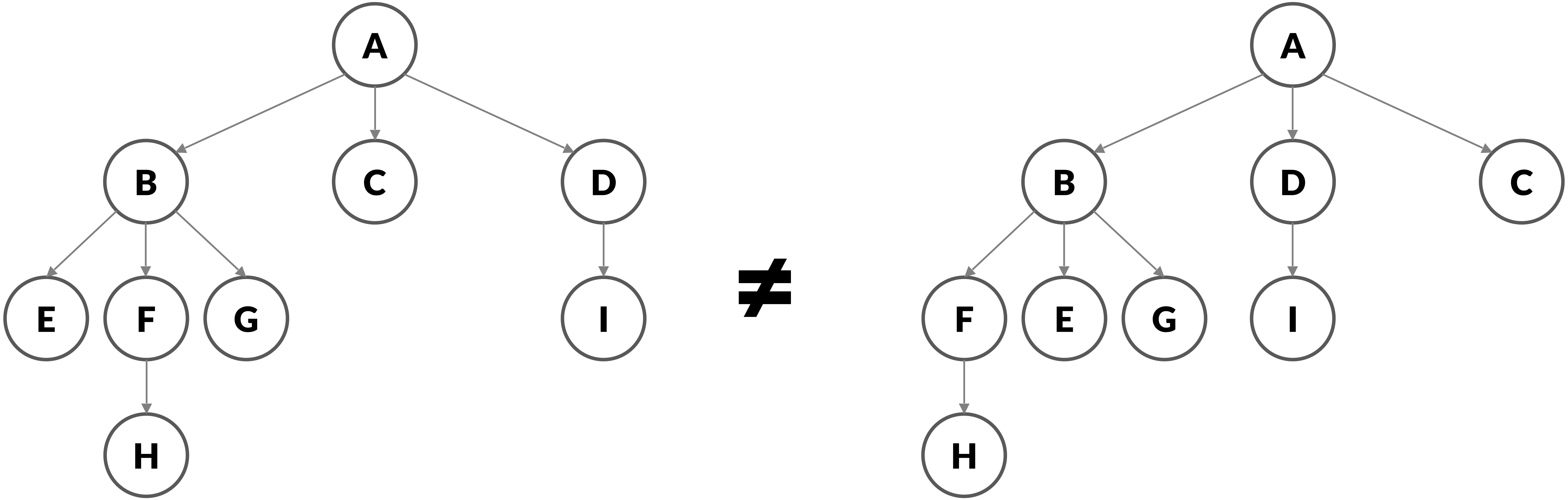
A graph with no cycles - or:

A collection of nodes

- **contains a root node and 0-n subtrees**
- **subtrees are connected to root by an edge**



Ordered Tree

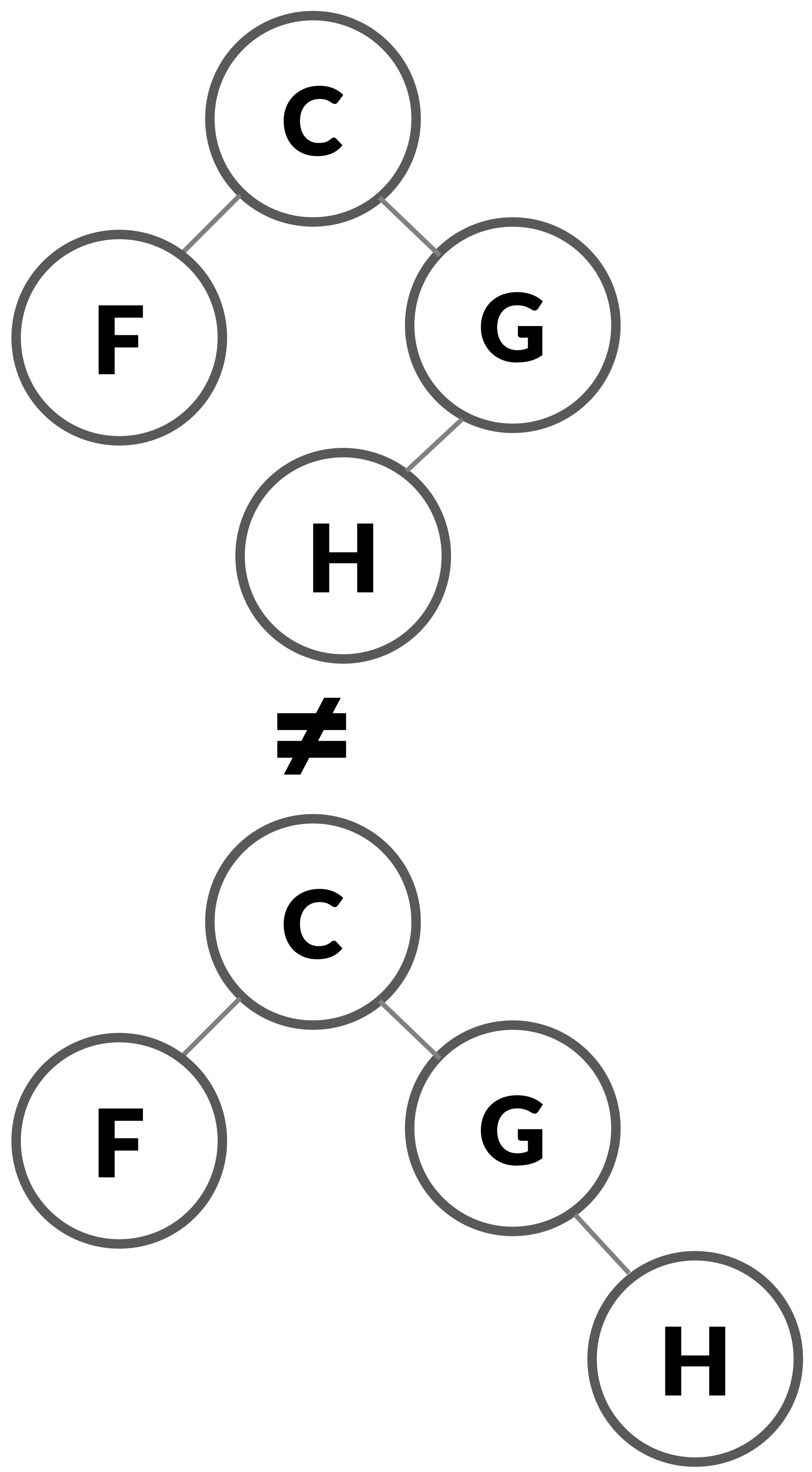


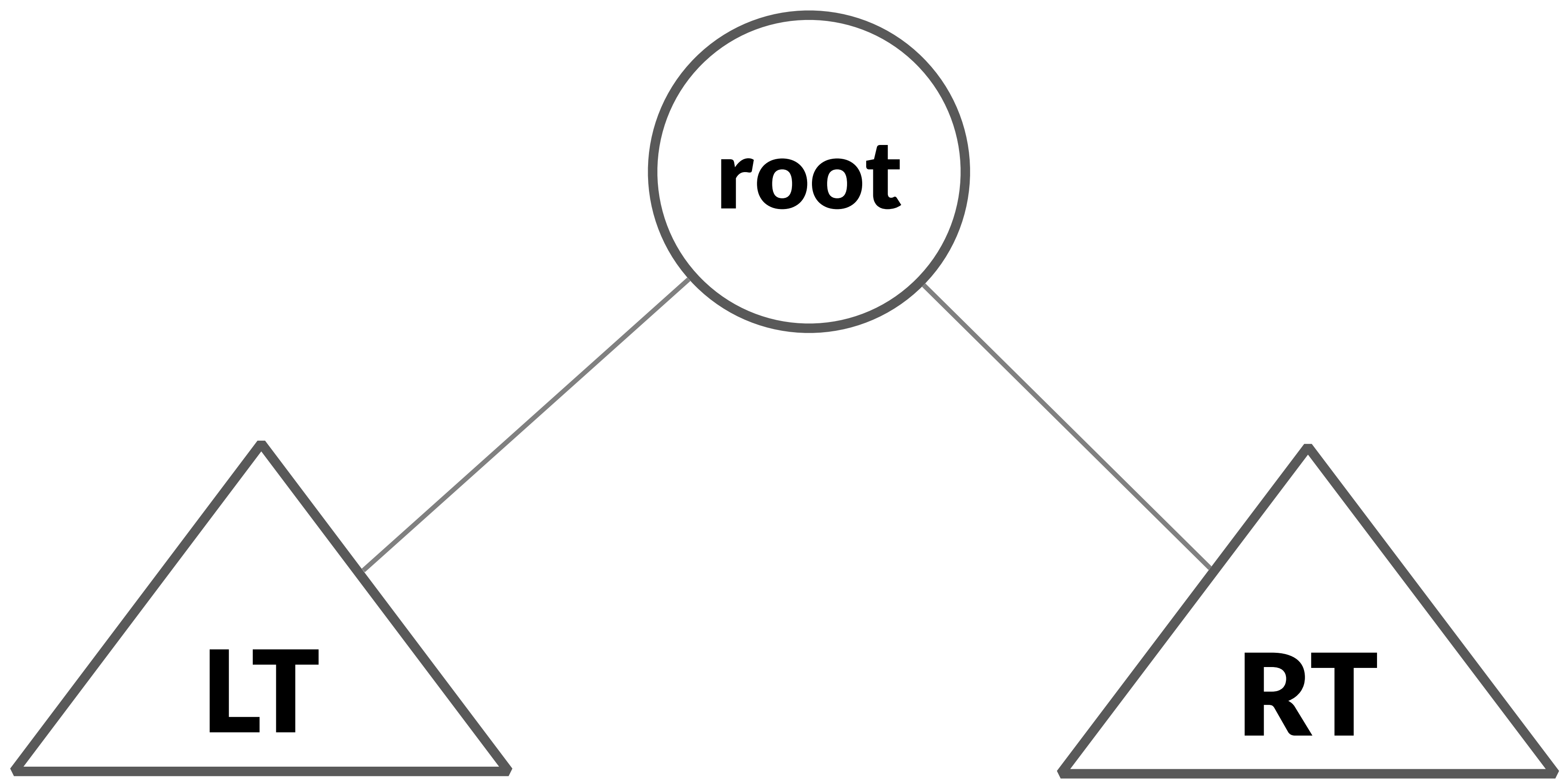
Binary Trees

Contains no nodes, or

Is comprised of three disjoint sets of nodes:

- a root node,
- a binary tree called its left subtree, and
- a binary tree called its right subtree

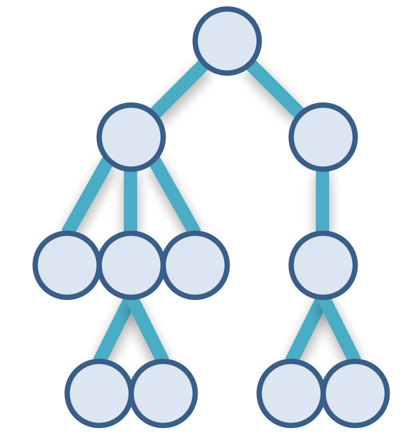




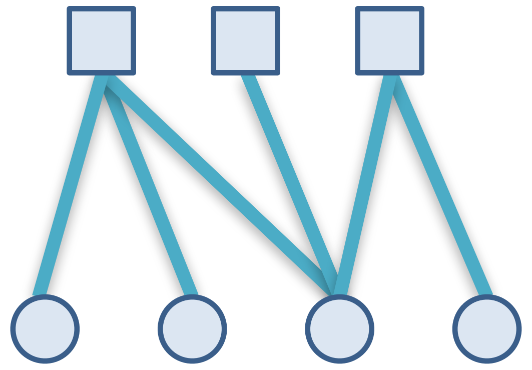
Different Kinds of Graphs

Over 1000 different graph classes

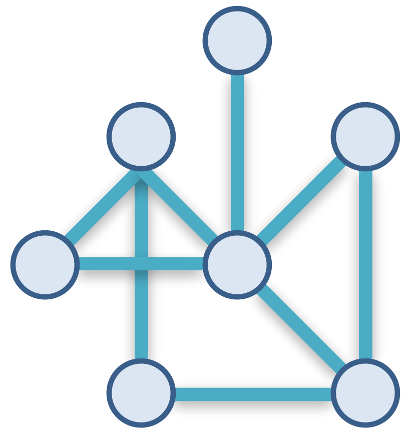
Tree



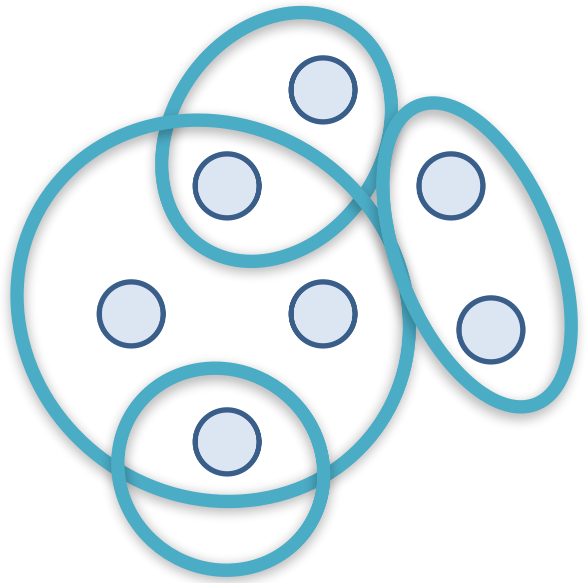
Bipartite Graph

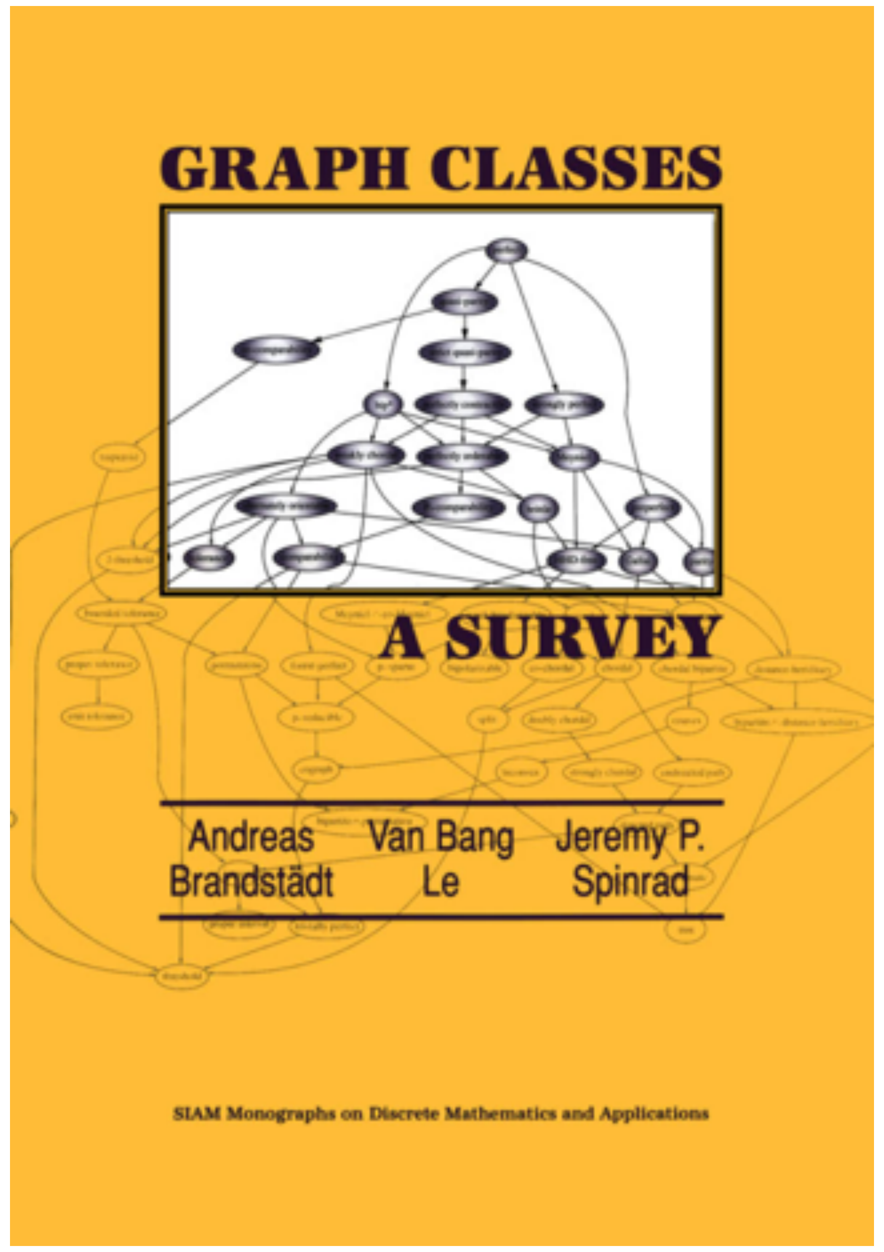


Network



Hypergraph





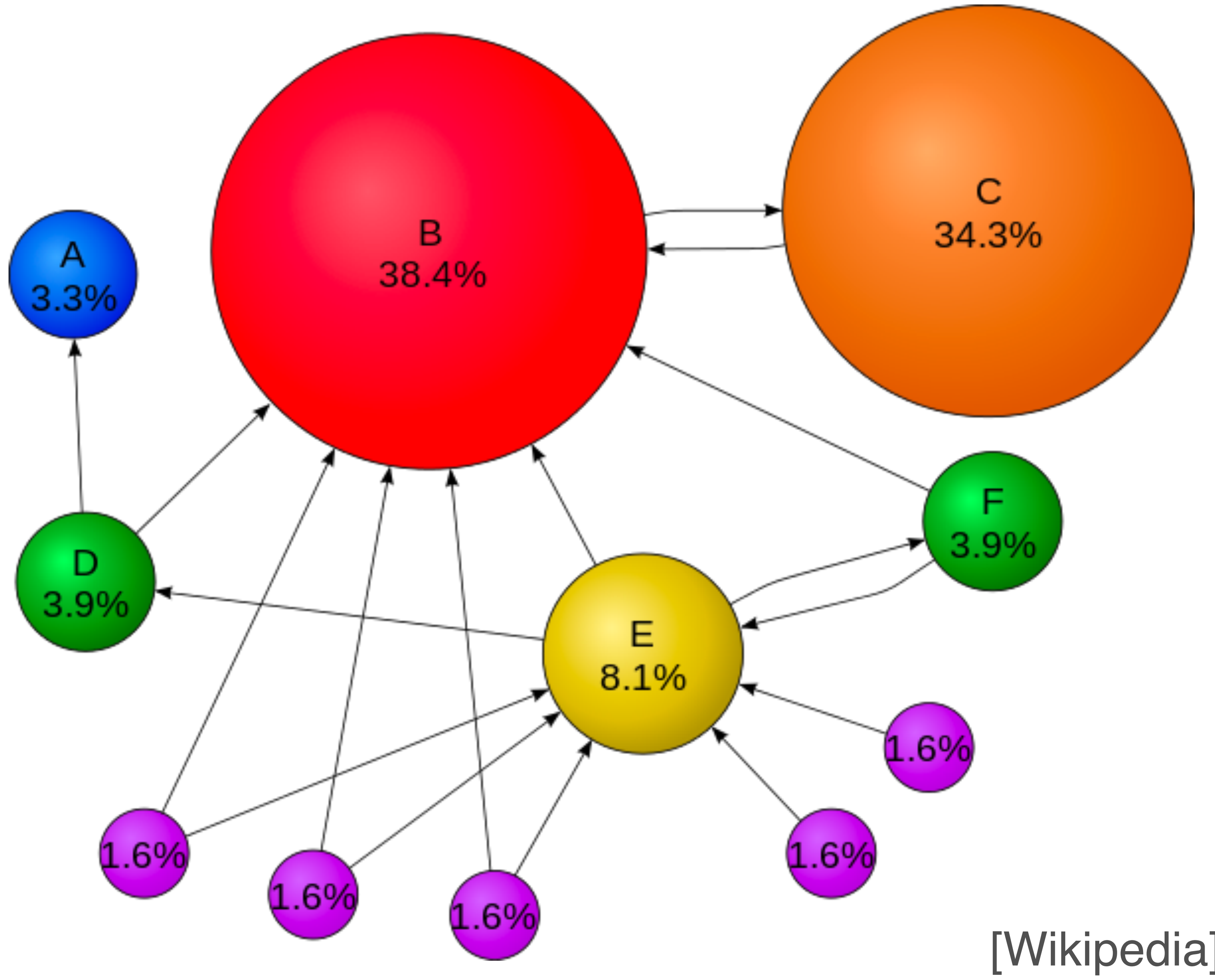
A. Brandstädt et al. 1999

Graph Measures

Node degree deg(x)
The number of edges being incident to this node. For directed graphs indeg/outdeg are considered separately.

Diameter of graph G
The longest shortest path within G.

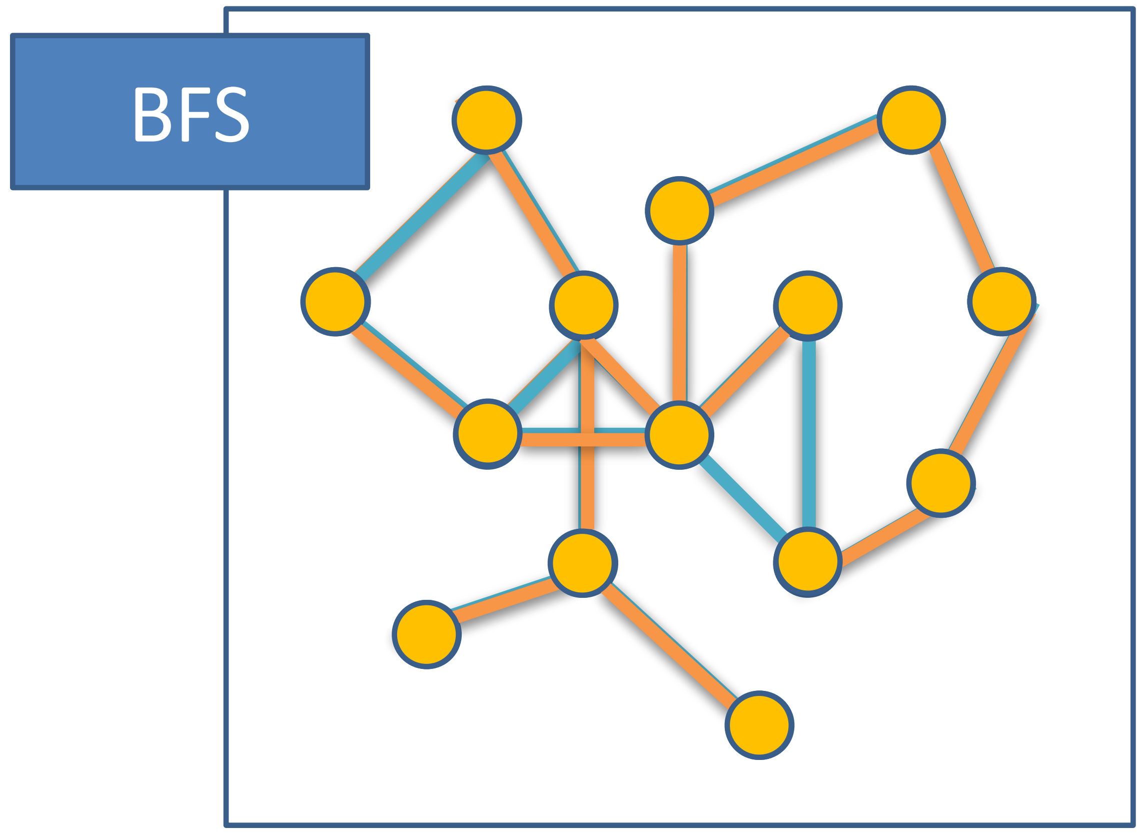
Pagerank
count number & quality of links



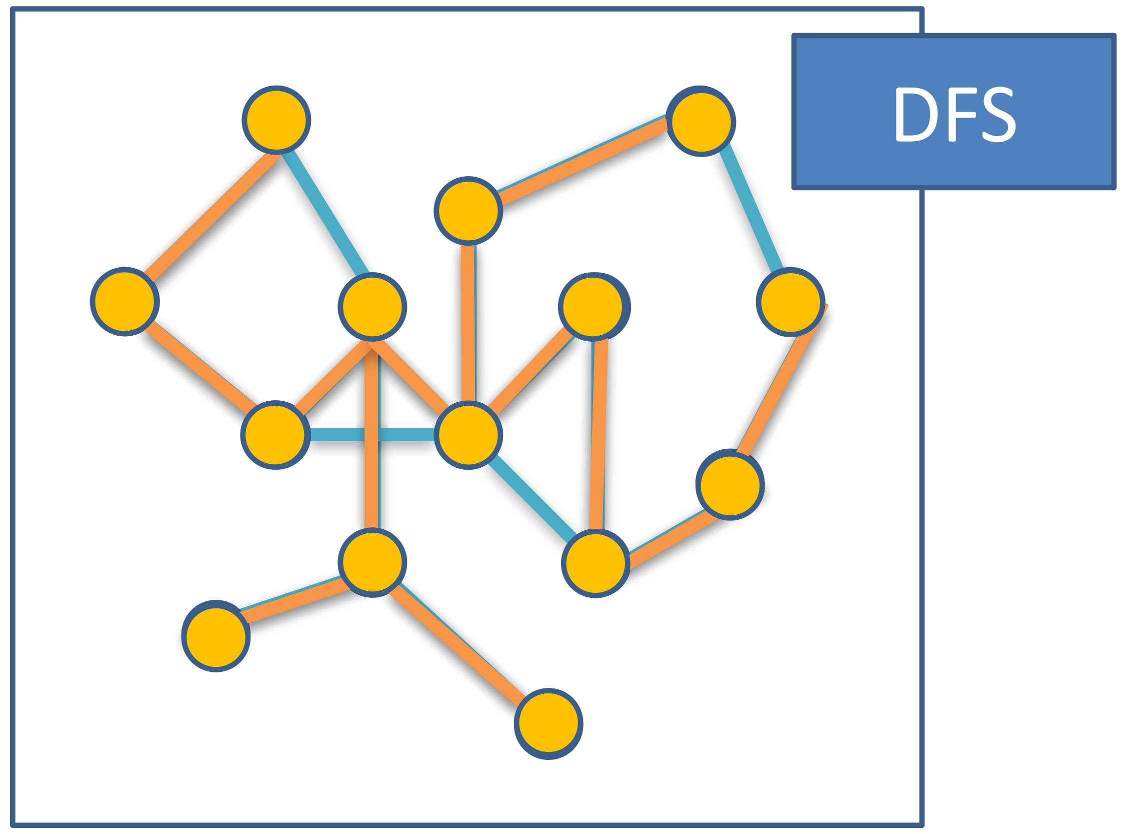
[Wikipedia]

Graph Algorithms (1)

Traversal: Breadth First Search, Depth First Search



- generates neighborhoods
- hierarchy gets rather wide than deep
- solves single-source shortest paths (SSSP)



- classical way-finding/back-tracking strategy
- tree serialization
- topological ordering

Hard Graph Algorithms (NP-Complete)

Longest path

Largest clique

Maximum independent set (set of vertices in a graph, no two of which are adjacent)

Maximum cut (separation of vertices in two sets that cuts most edges)

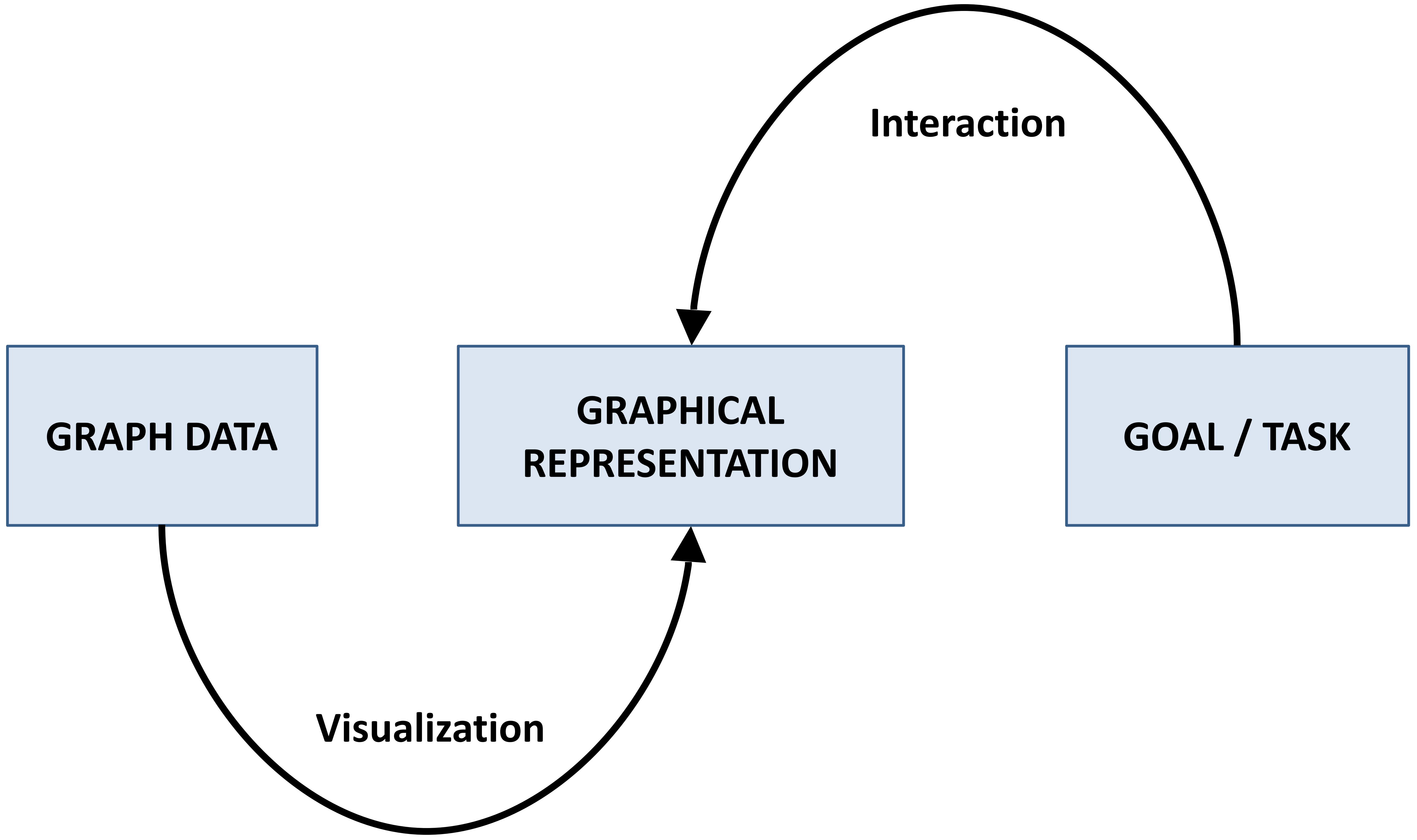
Hamiltonian path/cycle (path that visits all vertexes once)

Coloring / chromatic number (colors for vertices where no adjacent v. have same color)

Minimum degree spanning tree

Graph and Tree Visualization

Setting the Stage



How to decide which **representation** to use for which **type of graph** in order to achieve which kind of **goal**?

Different Kinds of Tasks/Goals

Two principal types of tasks: **attribute-based (ABT)** and **topology-based (TBT)**

Localize – find a single or multiple nodes/edges that fulfill a given property

- ABT: Find the edge(s) with the maximum edge weight.
- TBT: Find all adjacent nodes of a given node.

Quantify – count or estimate a numerical property of the graph

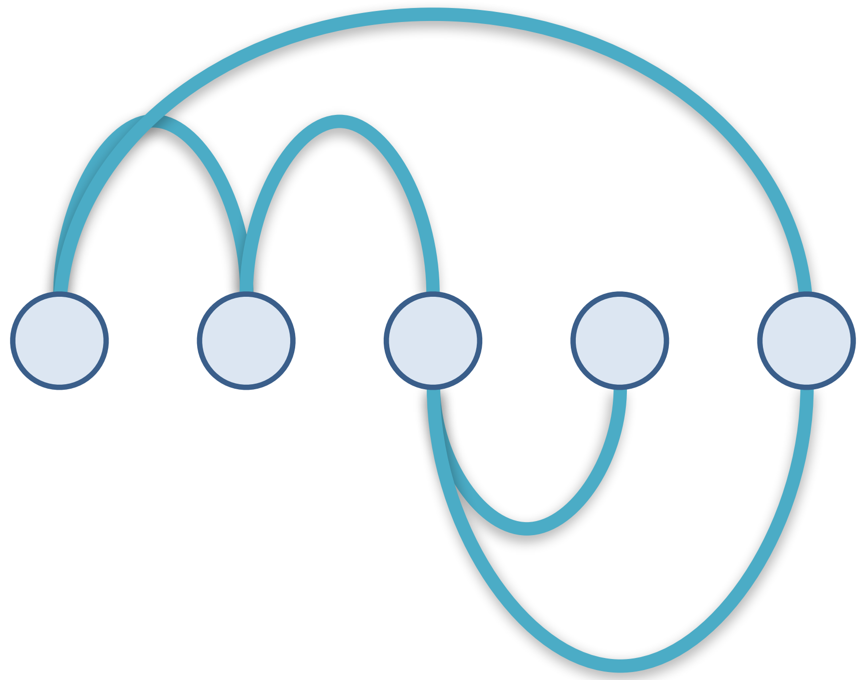
- ABT: Give the number of all nodes.
- TBT: Give the indegree (the number of incoming edges) of a node.

Sort/Order – enumerate the nodes/edges according to a given criterion

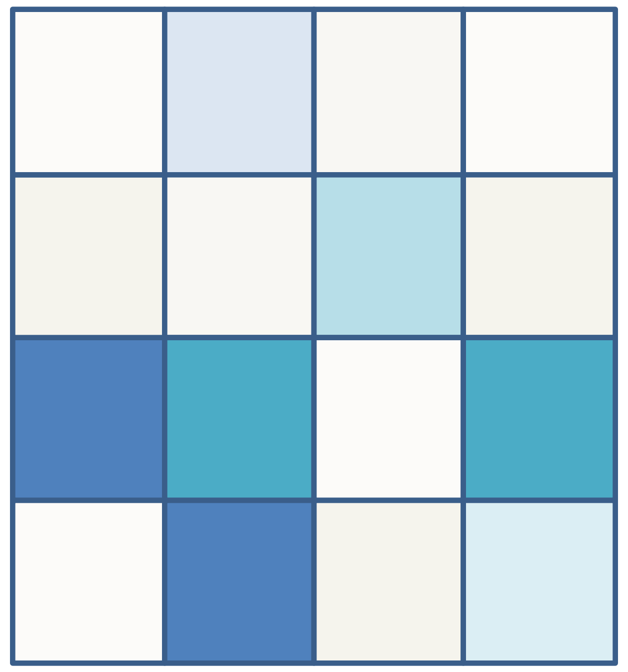
- ABT: Sort all edges according to their weight.
- TBT: Traverse the graph starting from a given node.

list adapted from Schulz 2010

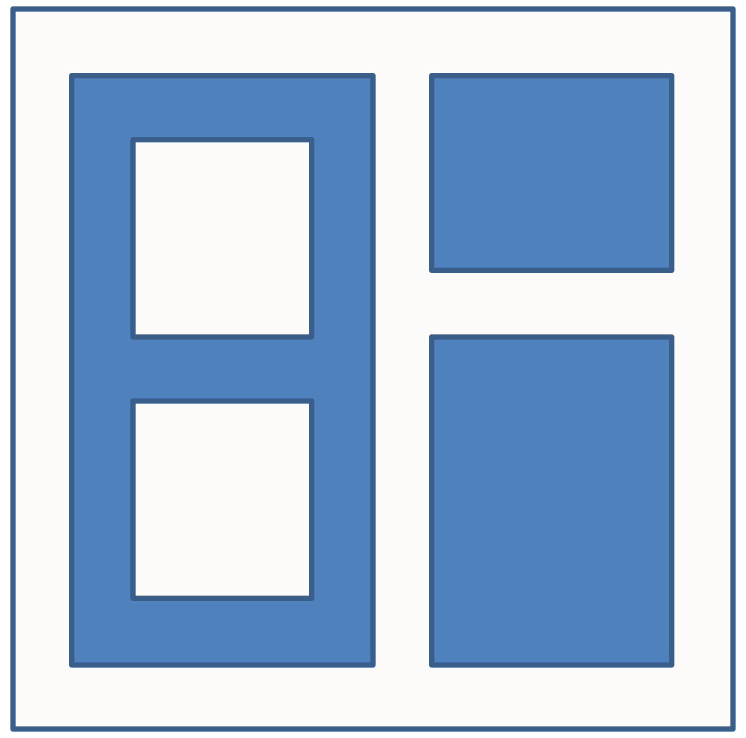
Three Types of Graph Representations



Explicit
(Node-Link)



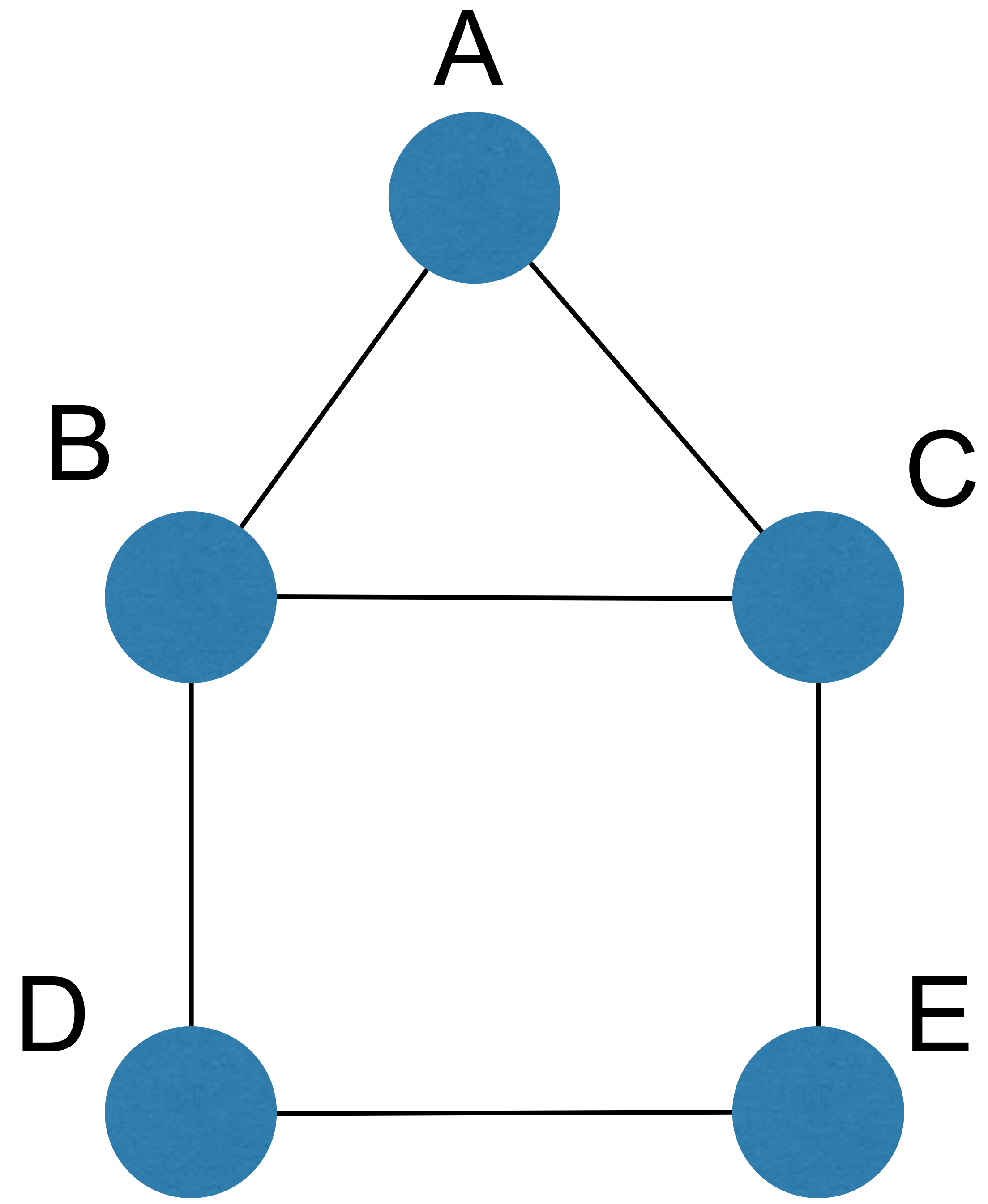
Matrix

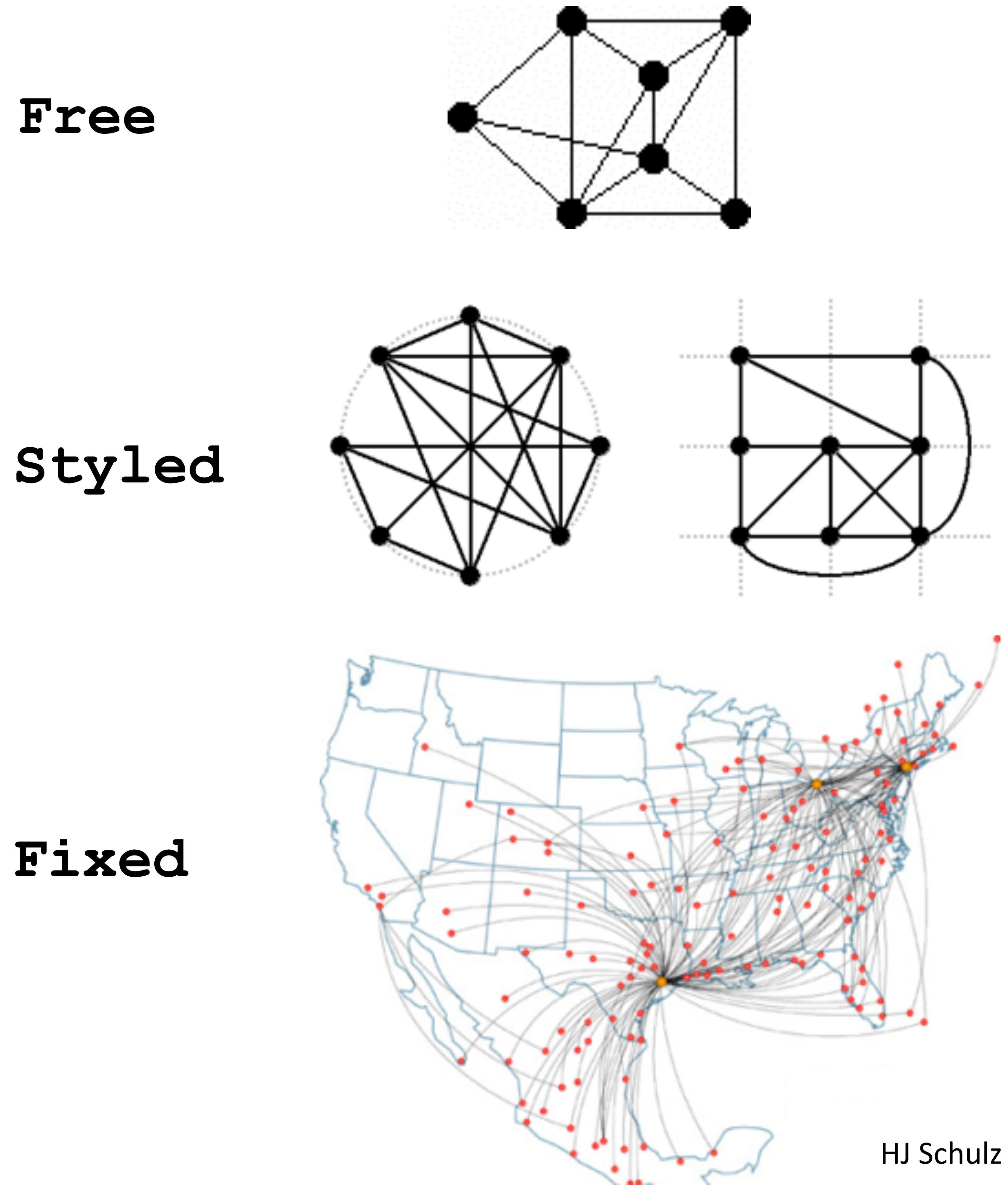


Implicit

Explicit Graph Representations

Node-link diagrams: vertex = point, edge = line/arc





HJ Schulz 2006

Criteria for Good Node-Link Layout

Minimized **edge crossings**

Minimized **distance** of neighboring nodes

Minimized **drawing area**

Uniform edge **length**

Minimized edge **bends**

Maximized **angular distance** between different edges

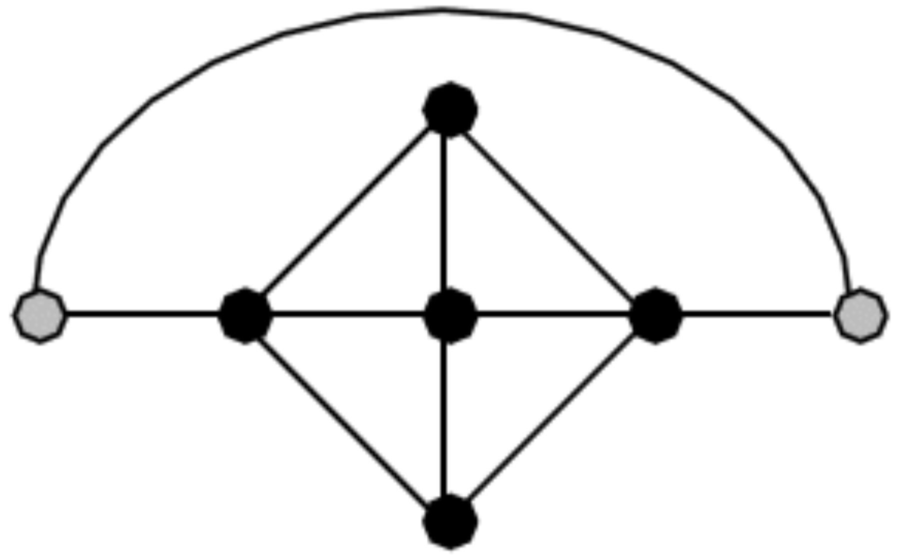
Aspect ratio about 1 (not too long and not too wide)

Symmetry: similar graph structures should look similar

list adapted from Battista et al. 1999

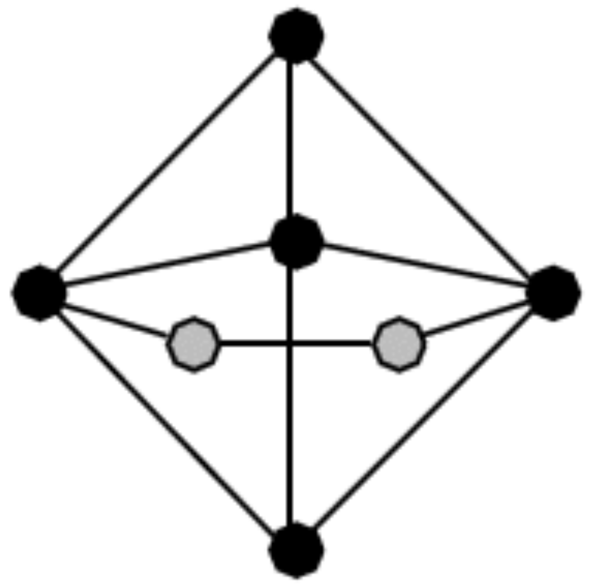
Conflicting Criteria

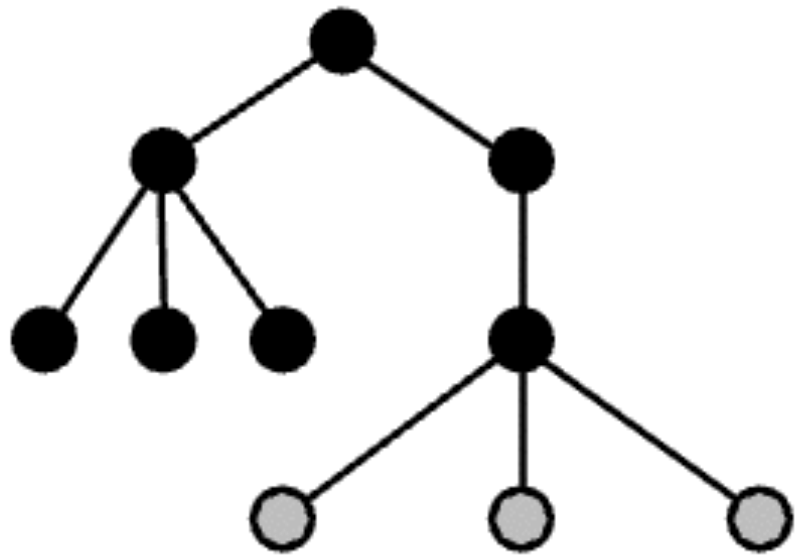
Minimum number of edge crossings



vs.

Uniform edge length

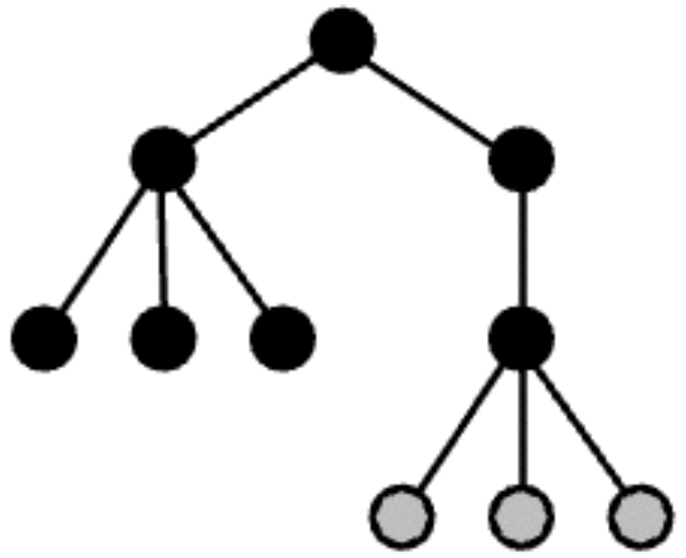




Space utilization

vs.

Symmetry



Schulz 2004

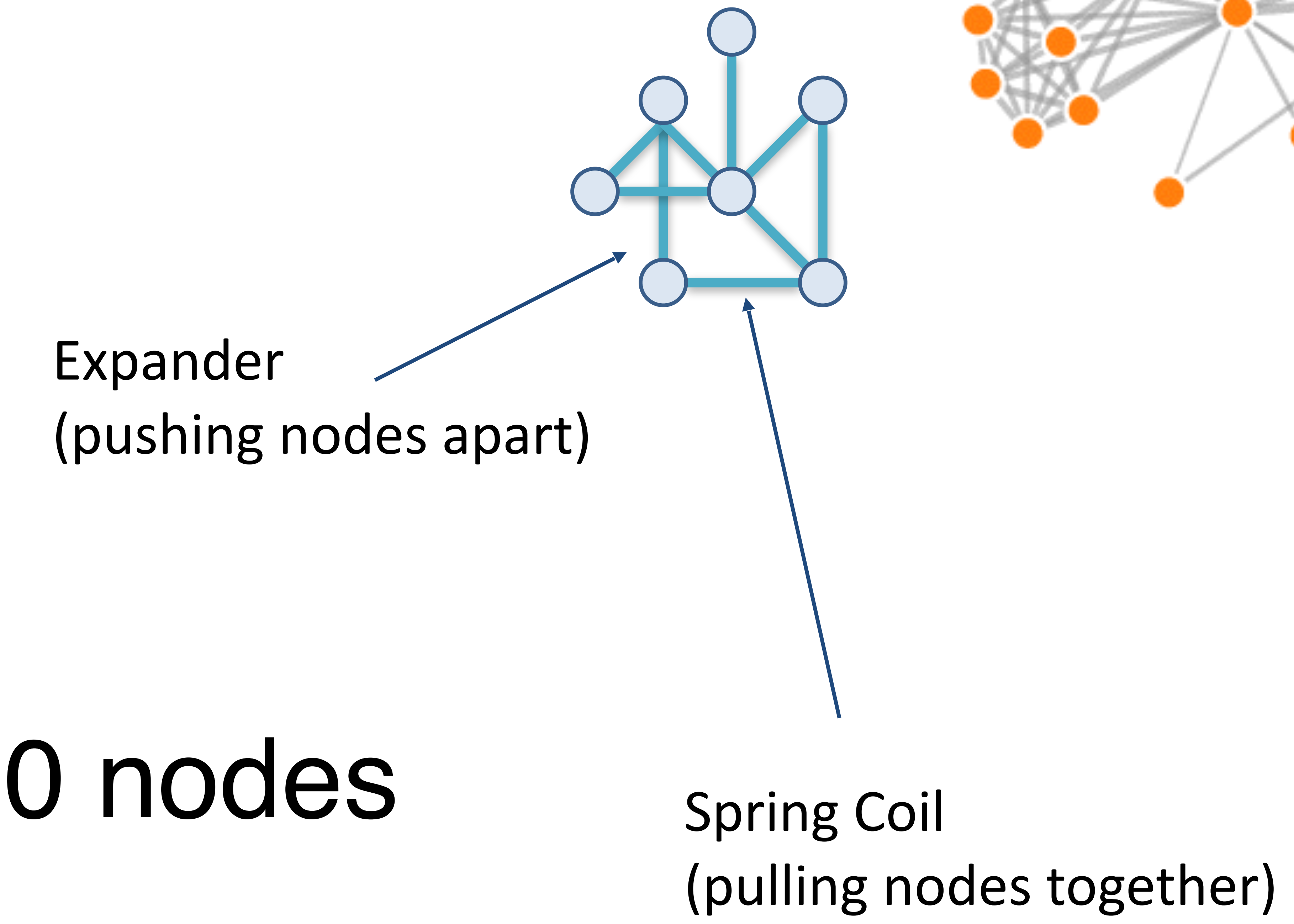
Force Directed Layouts

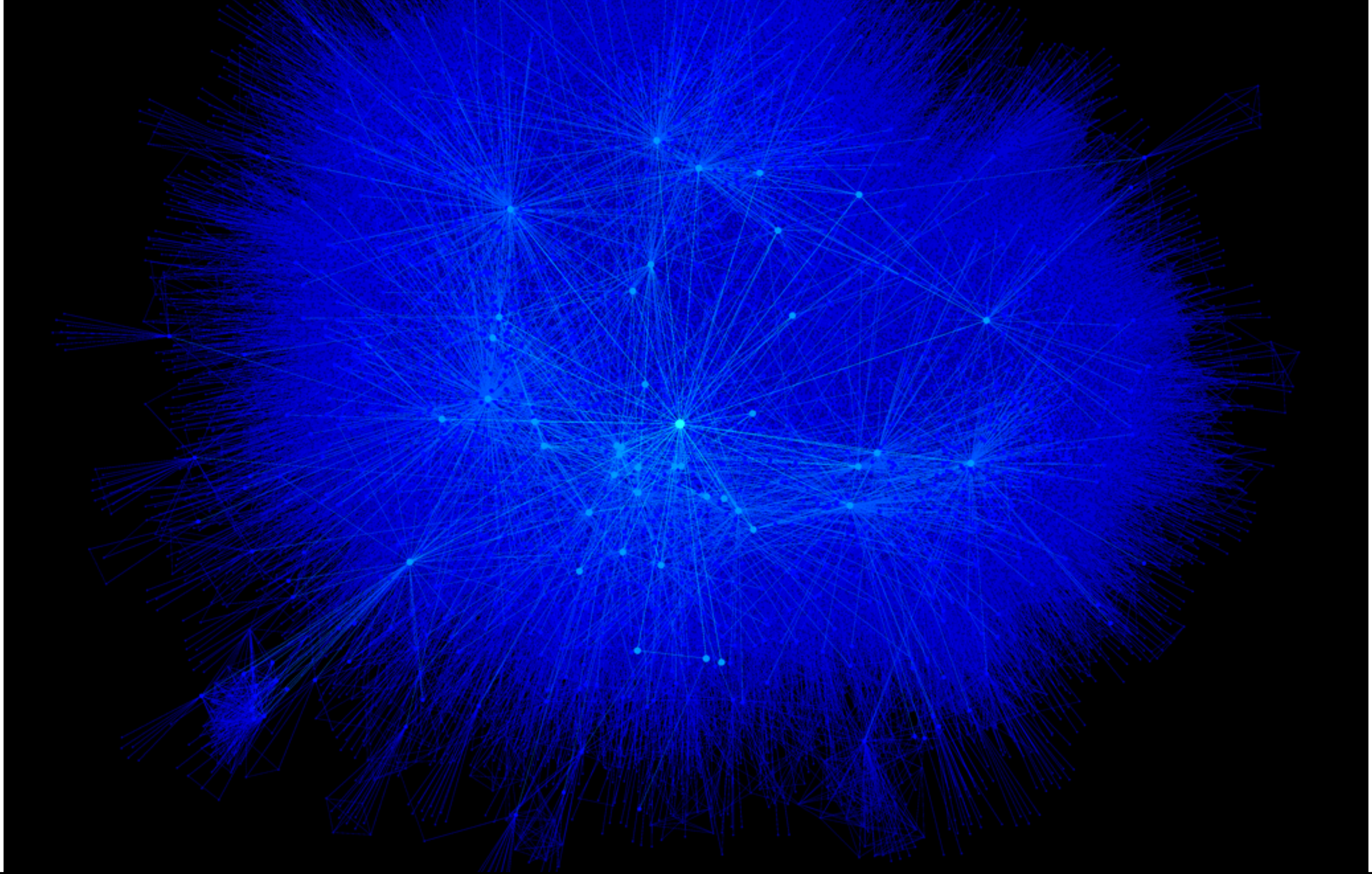
Physics model:
edges = springs,
vertices = repulsive magnets

in practice: damping

Computationally expensive: $O(n^3)$

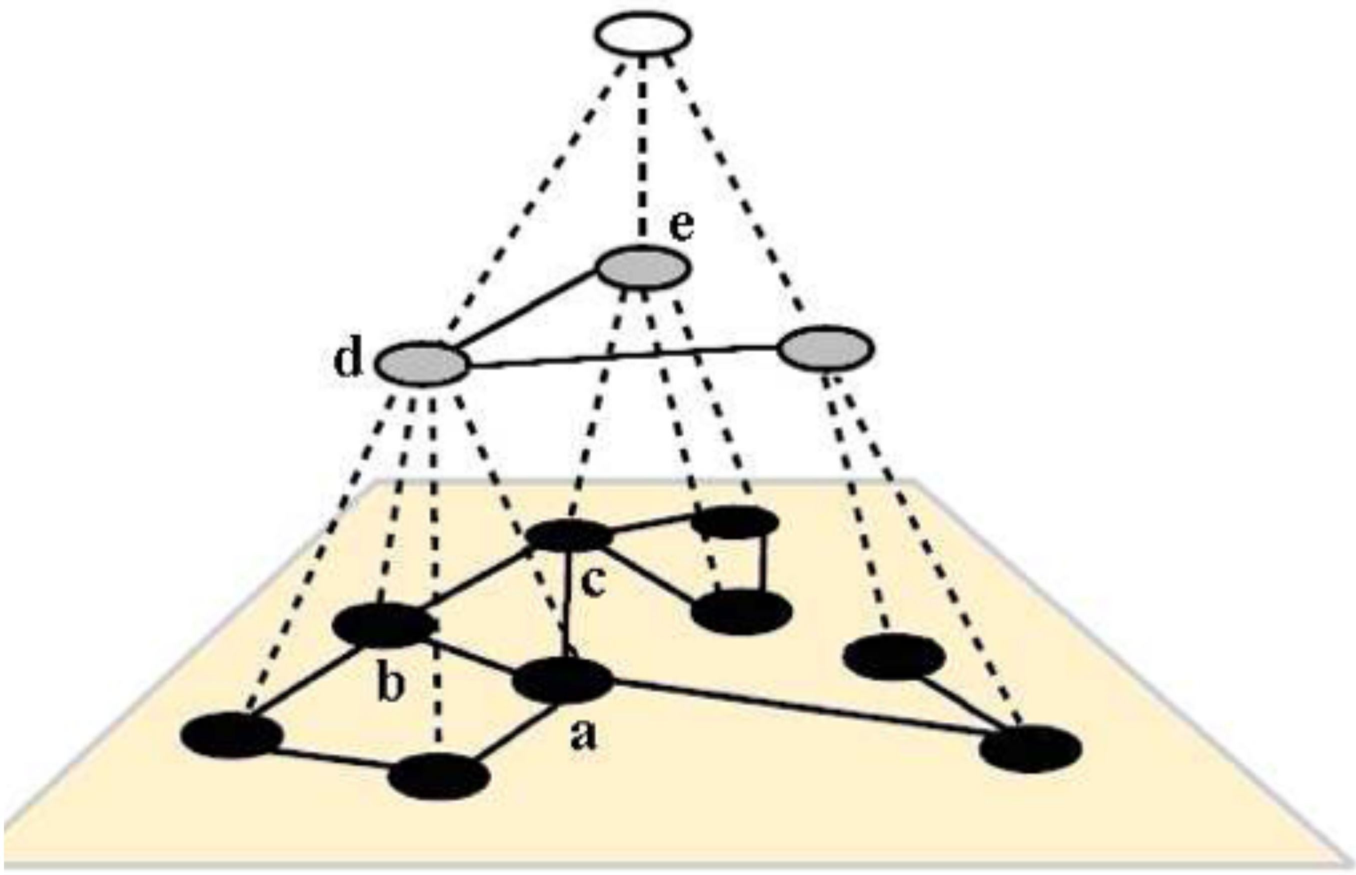
Limit (interactive): ~1000 nodes

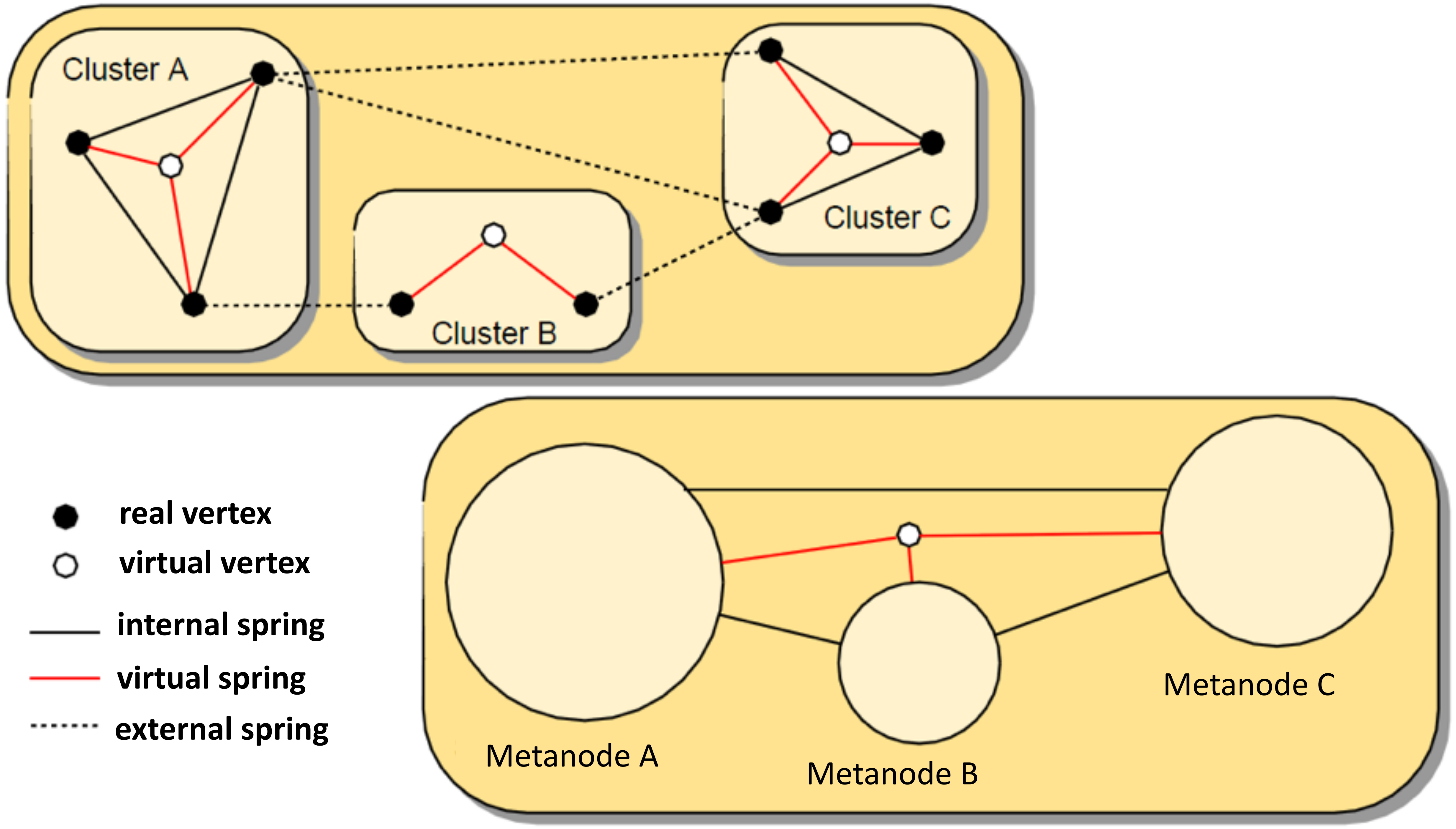




Giant Hairball

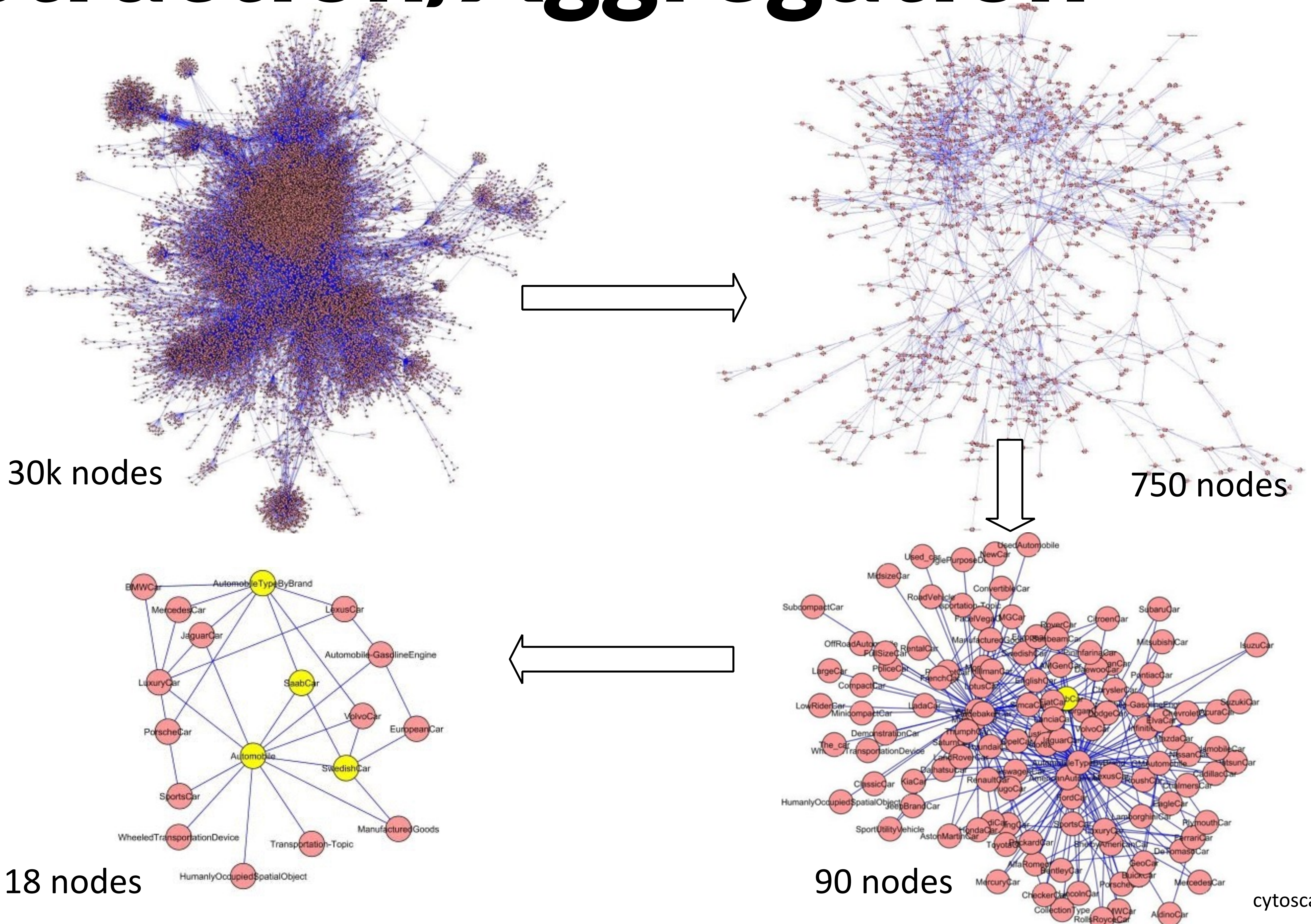
Adress Computational Scalability: Multilevel Approaches





[Schulz 2004]

Abstraction/Aggregation



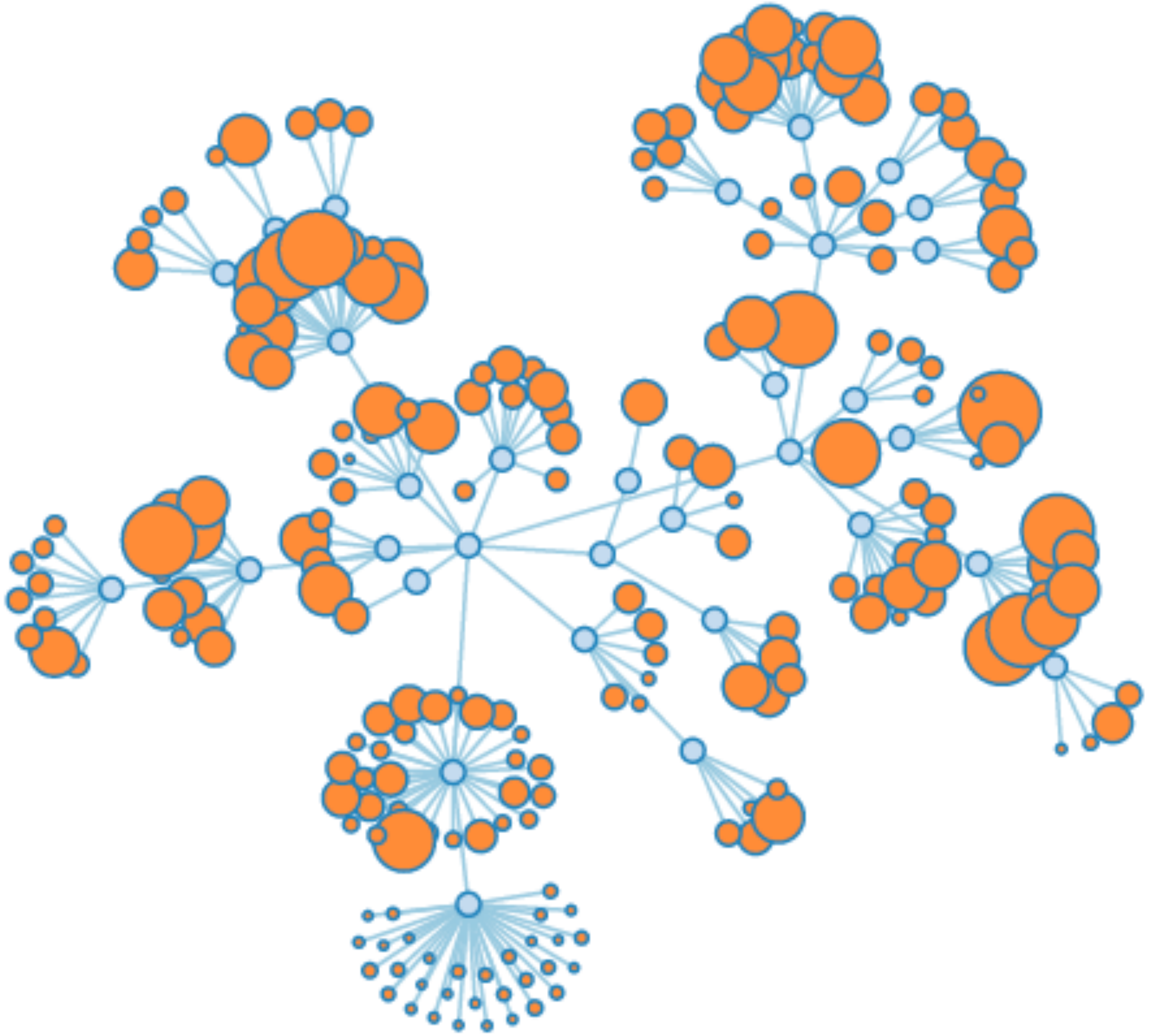
cytoscape.org

Collapsible Force Layout

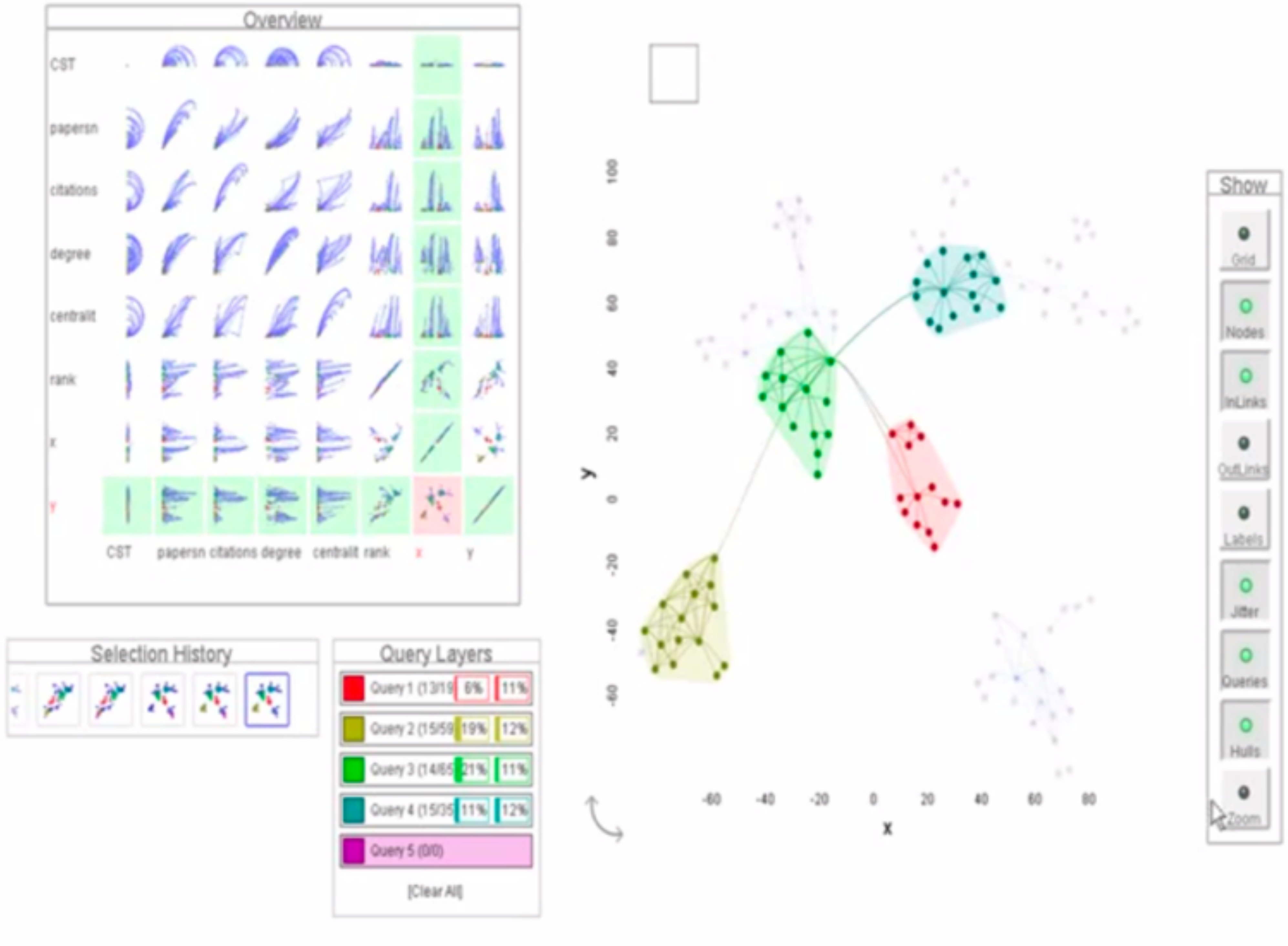
Supernodes: aggregate of nodes

manual or algorithmic

- clustering



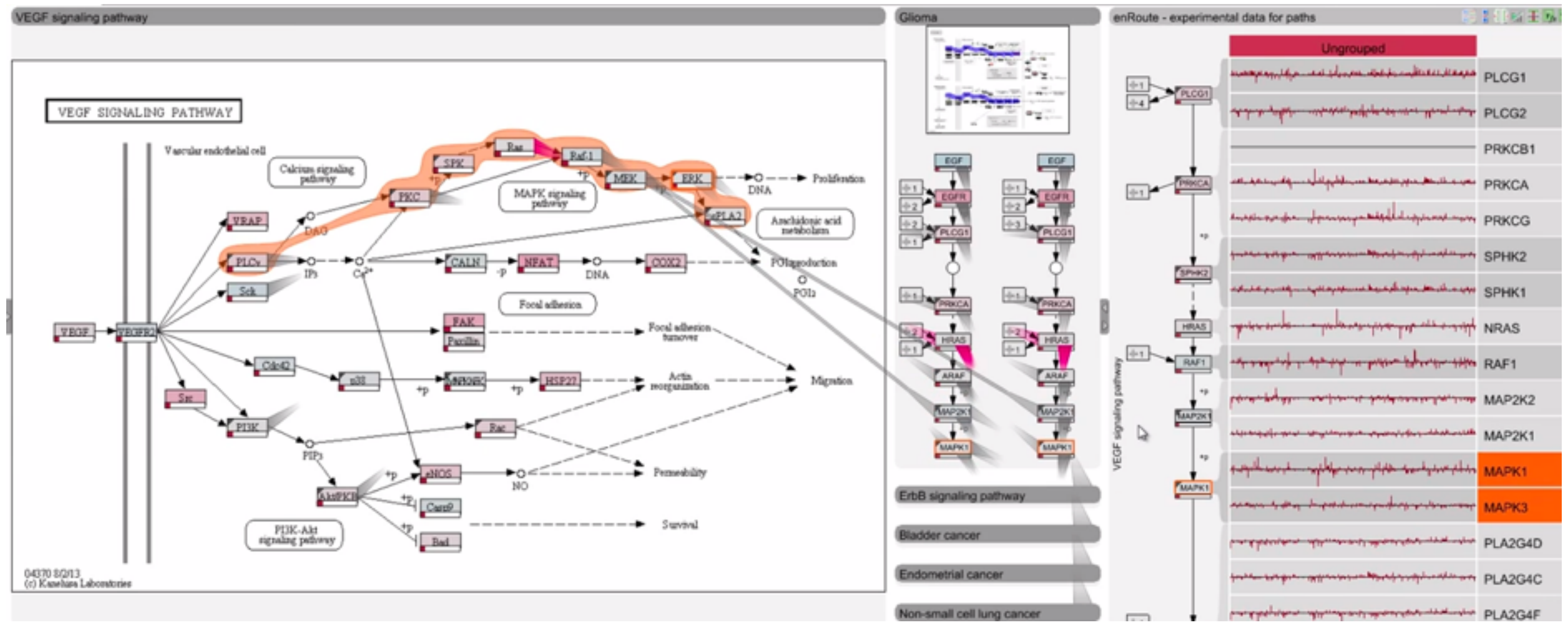
Node Attributes

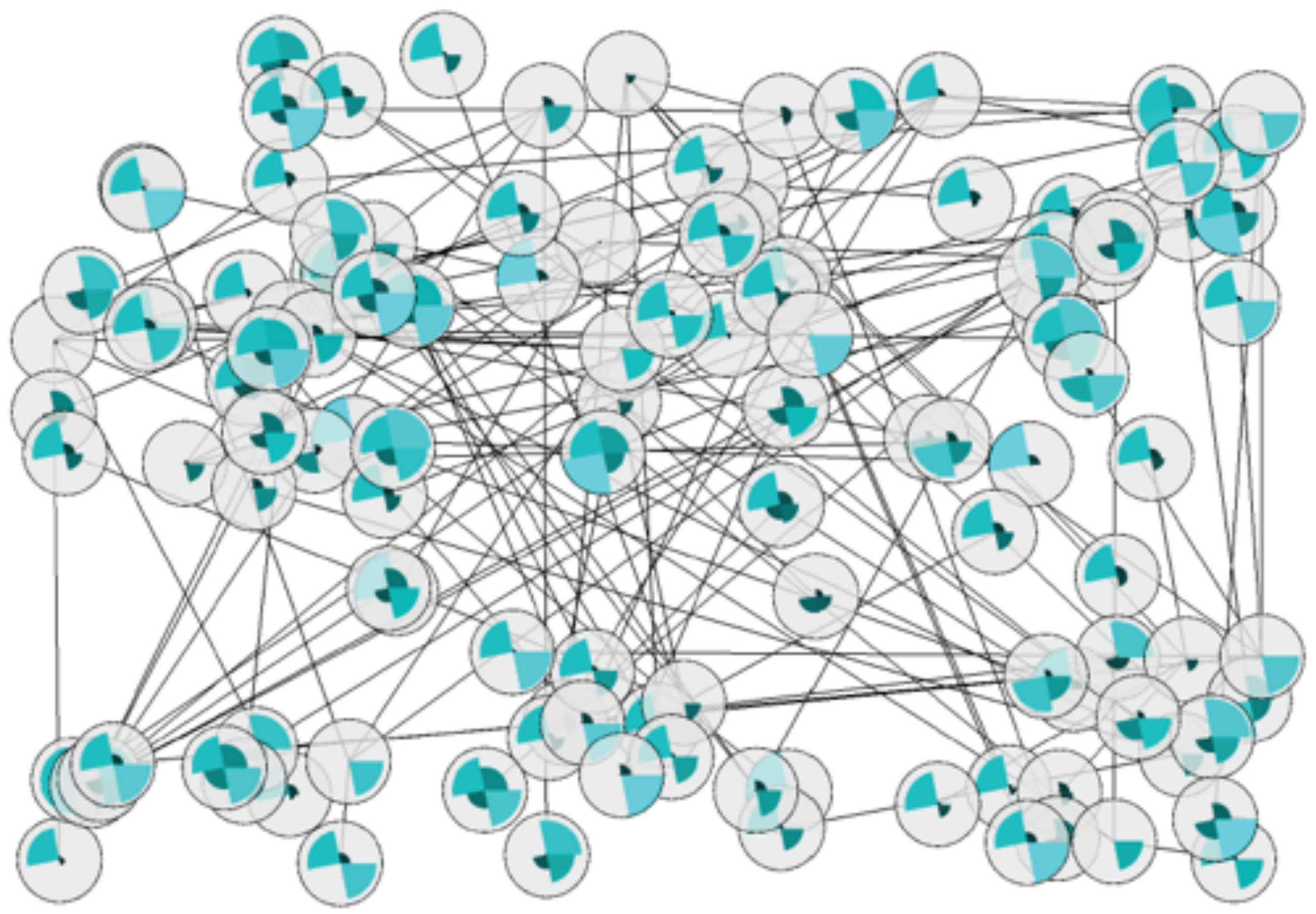


Coloring

Position

Multiple Views /
Path extraction



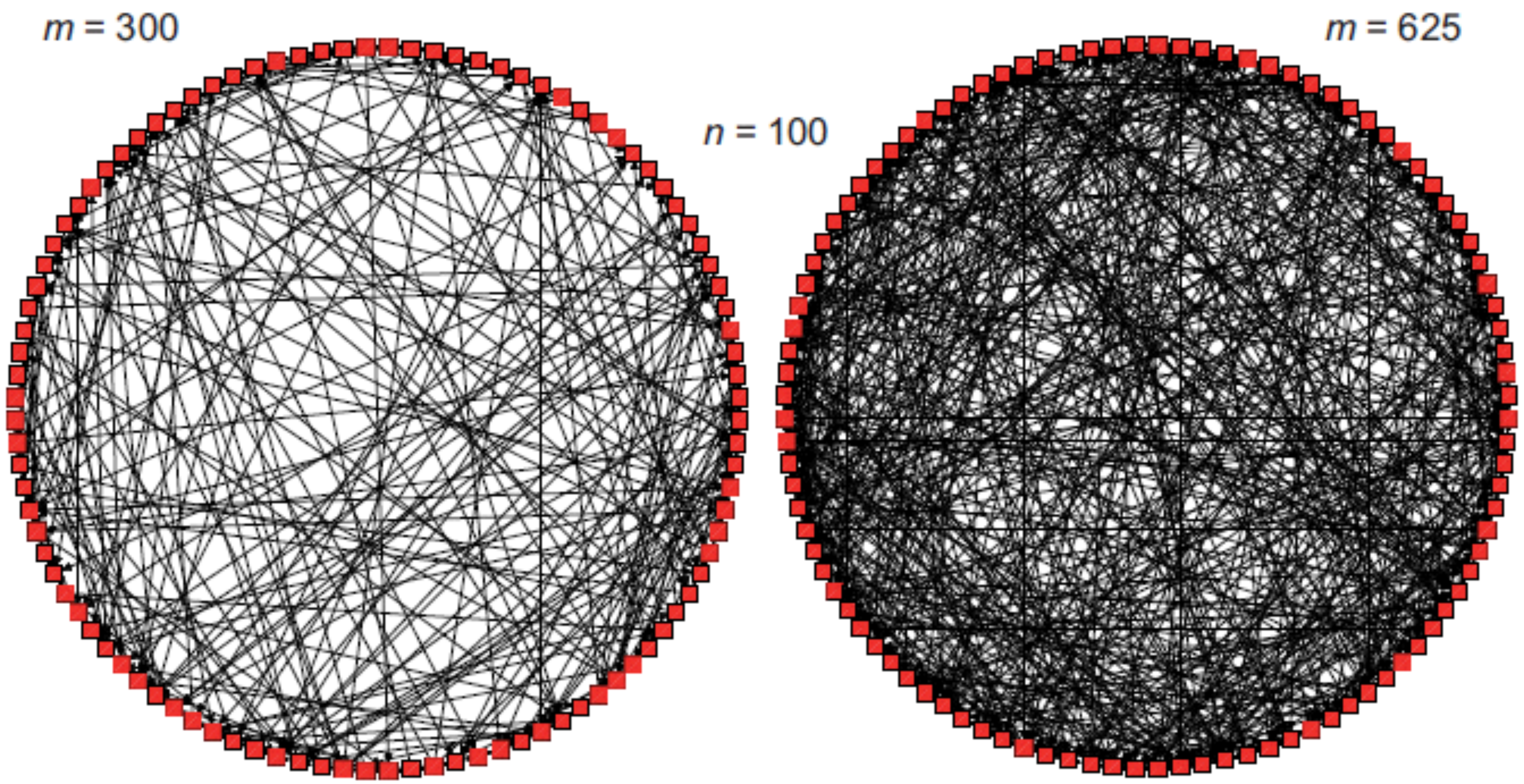


Styled / Restricted Layouts

Circular Layout

Node ordering

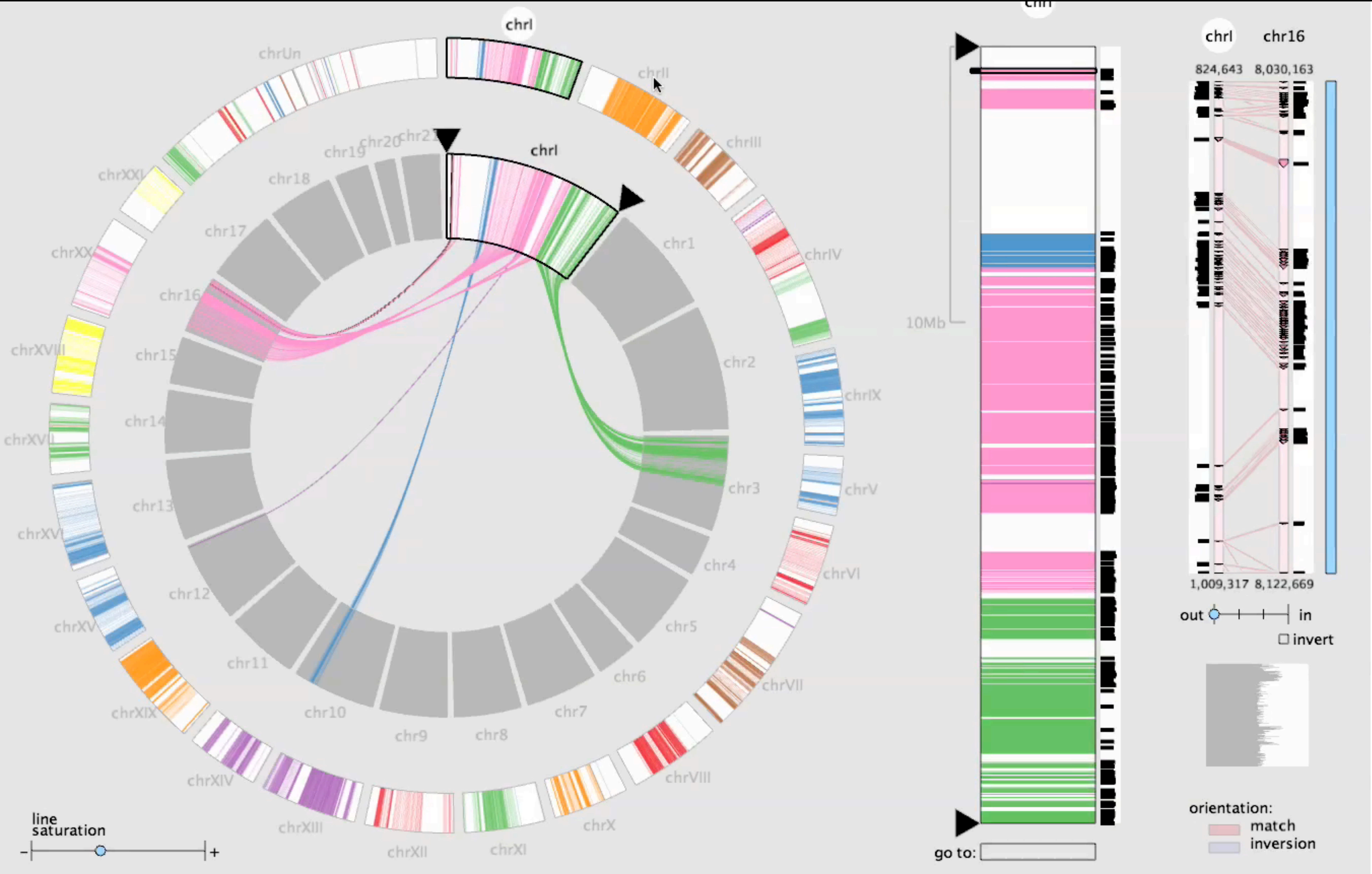
Edge Clutter



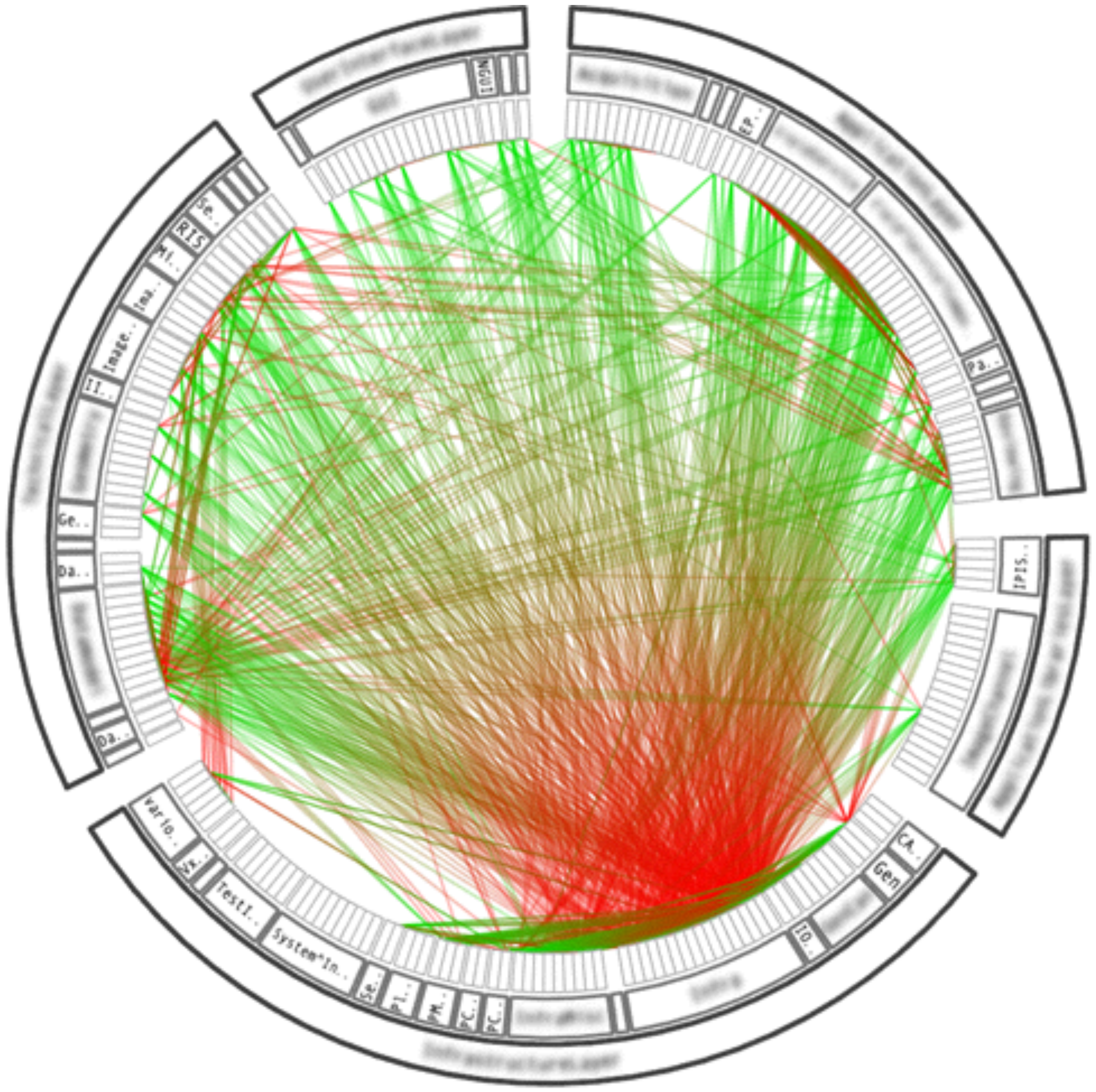
ca. 3% of all possible edges

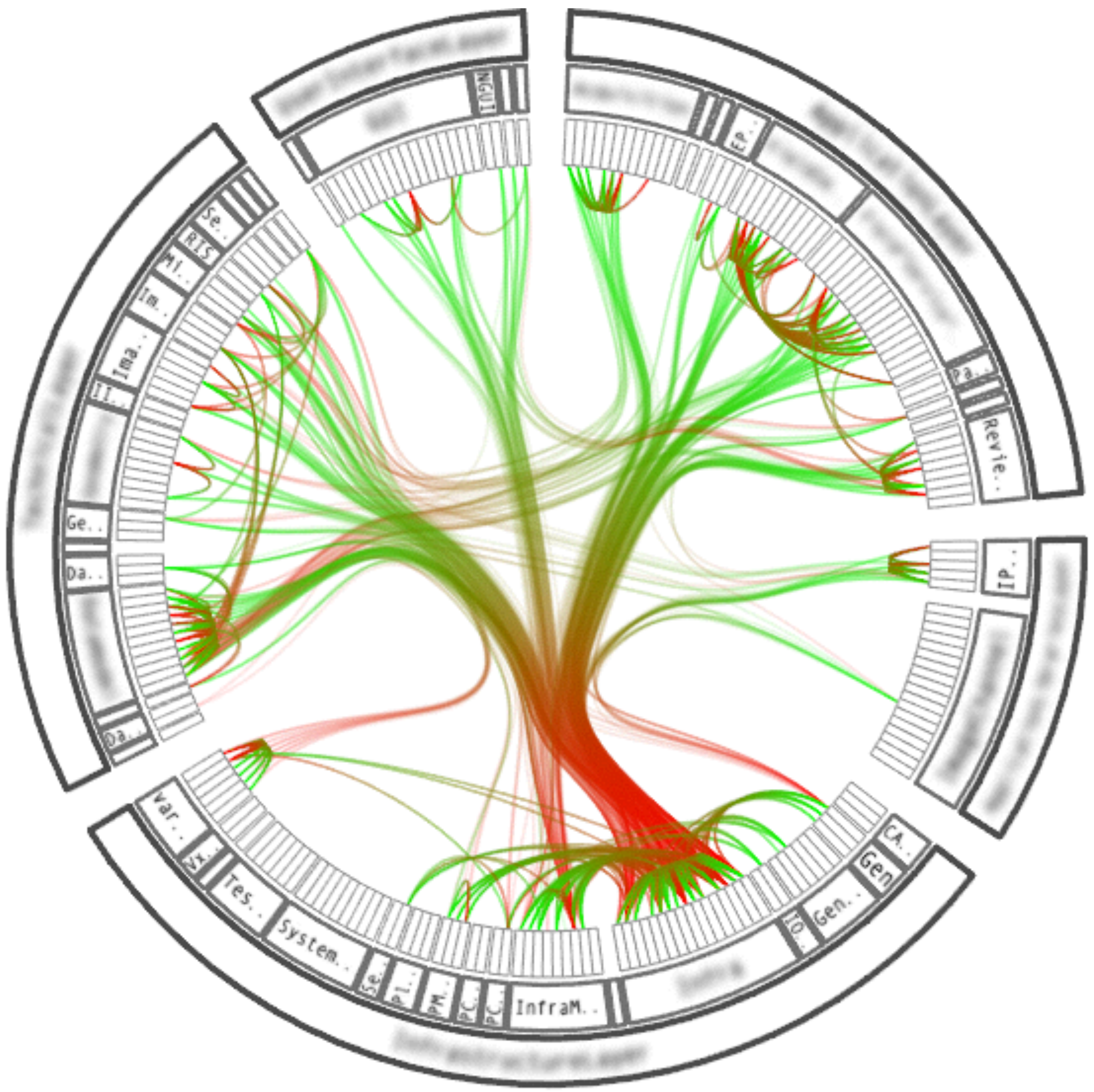
ca. 6,3% of all possible edges

Example: MizBee [Meyer et al. 2009]



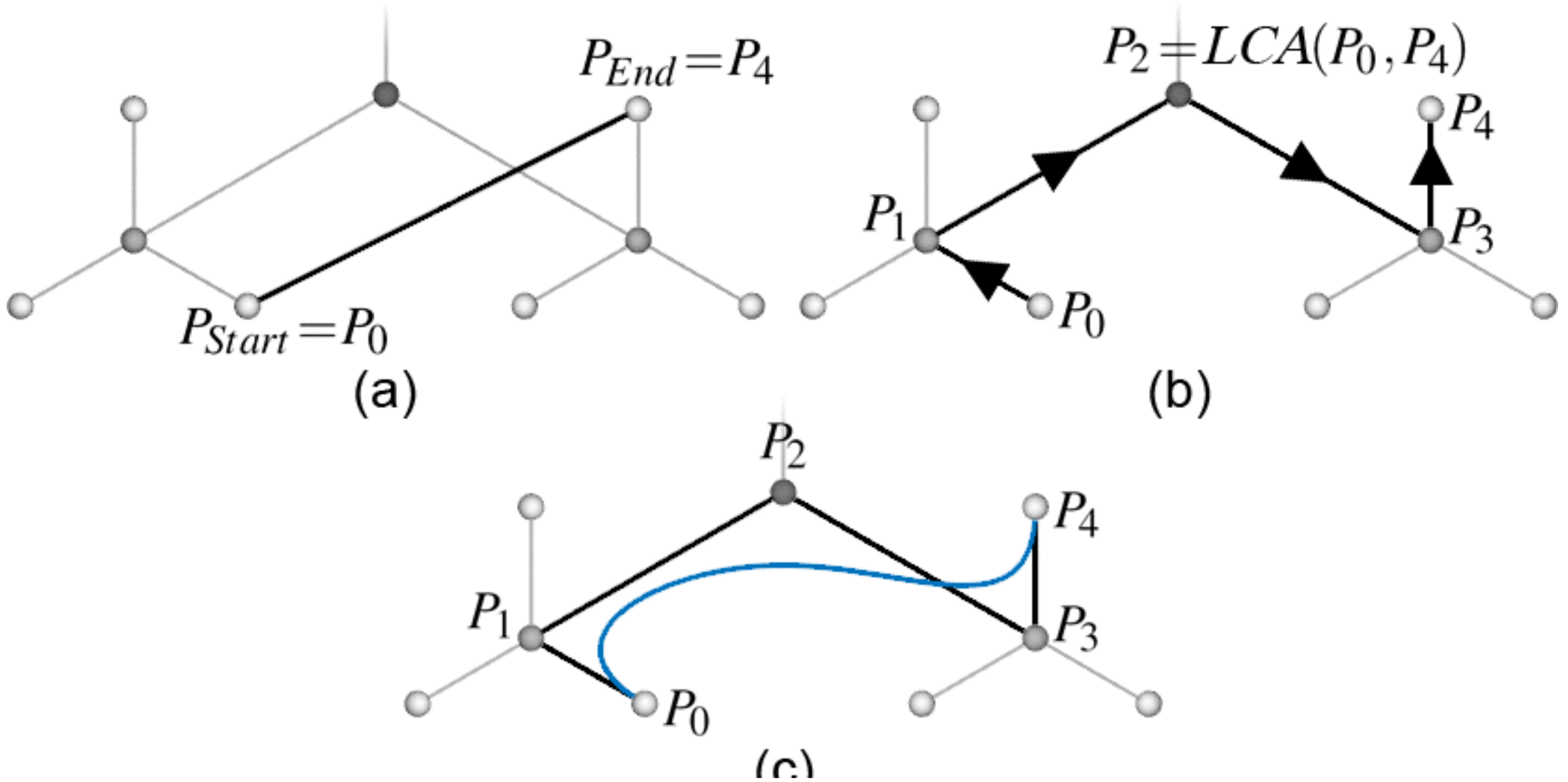
Reduce Clutter: Edge Bundling

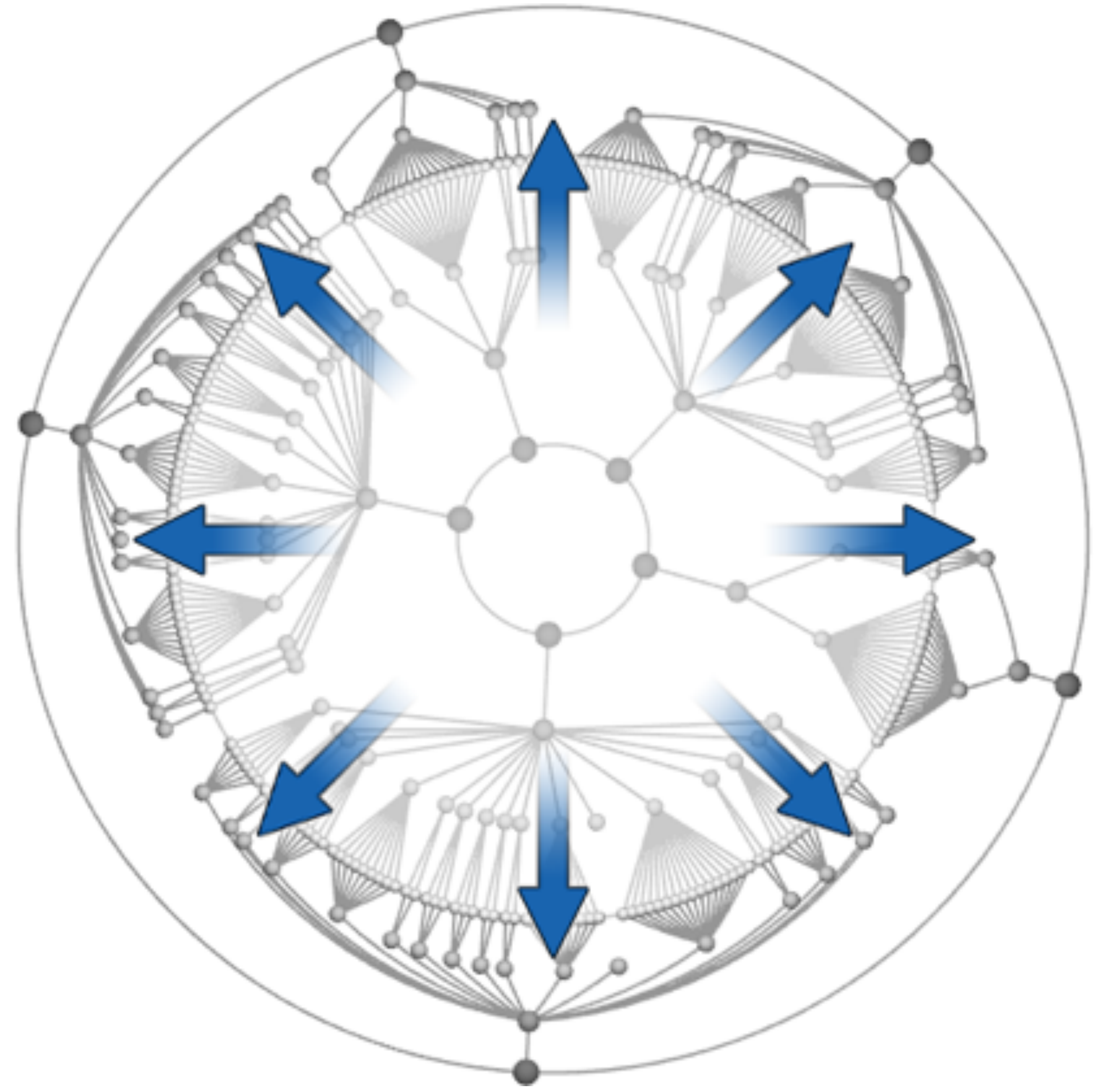


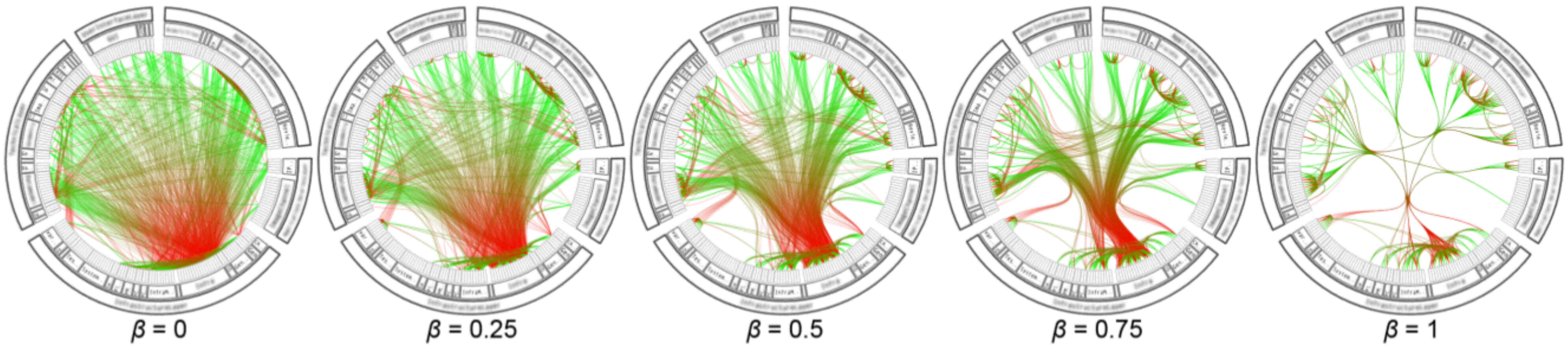


Holten et al. 2006

Hierarchical Edge Bundling







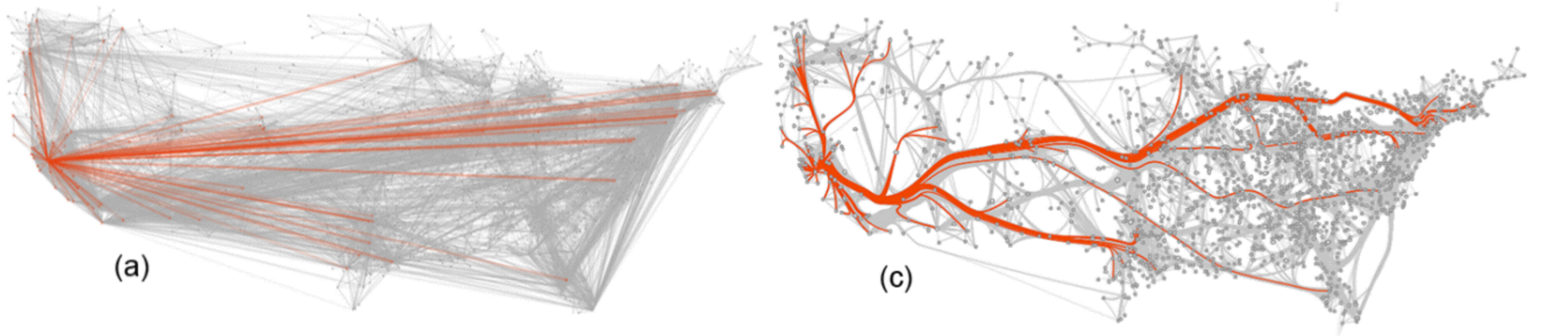
Bundling Strength

Holten et al. 2006

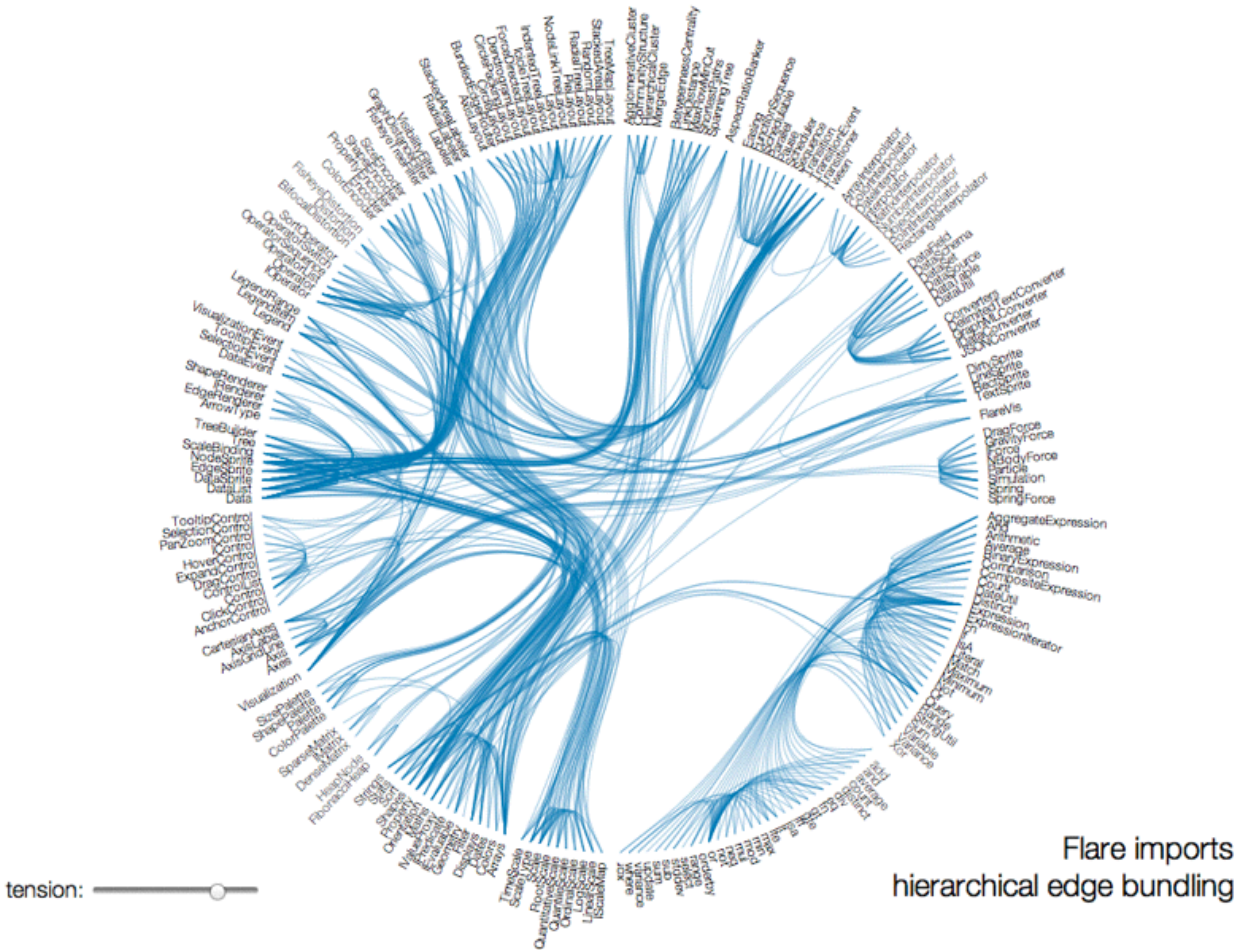
Fixed Layouts

Can’t vary position of nodes

Edge routing important



Bundling Strength



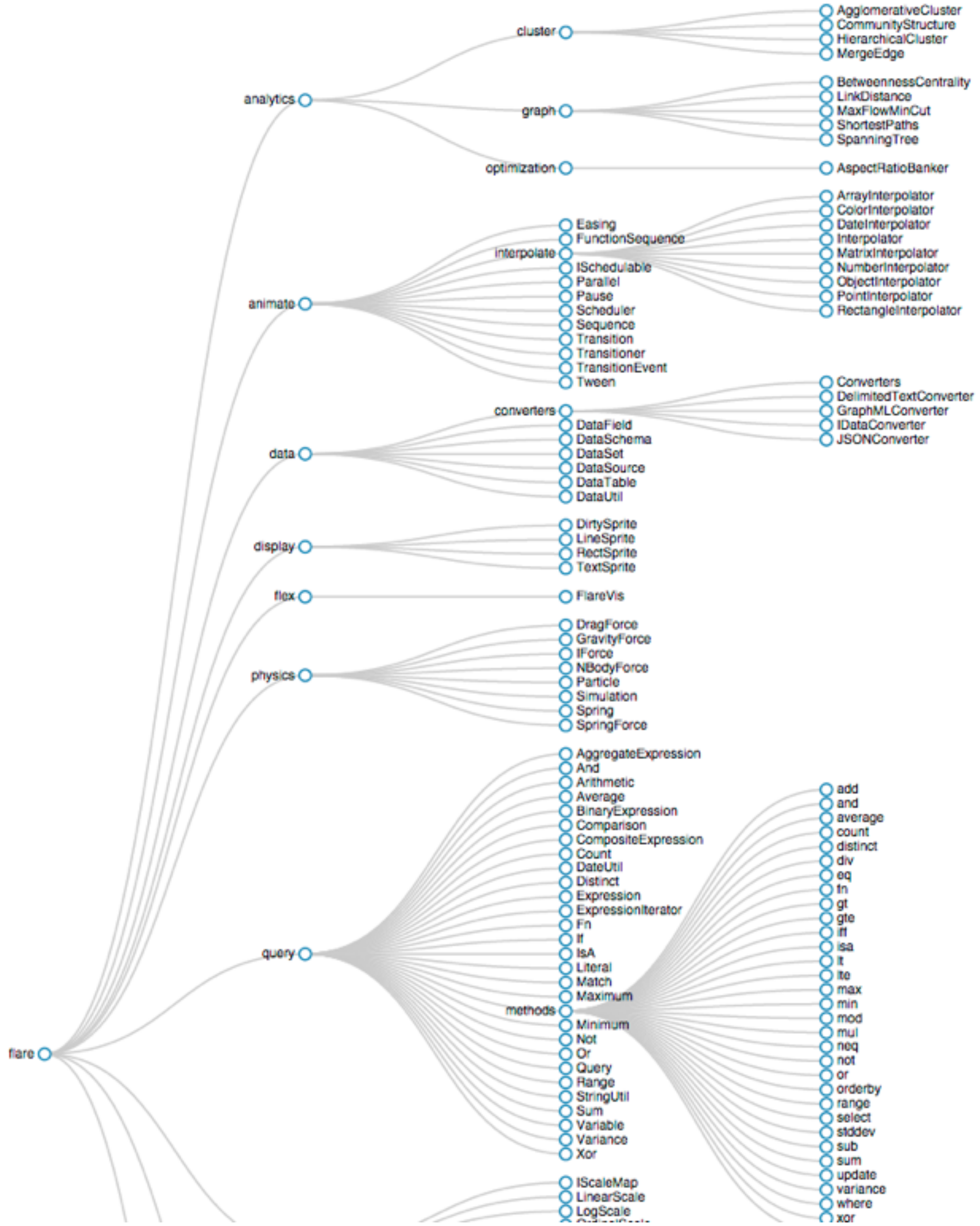
mbostock.github.com/d3/talk/20111116/bundle.html

Michael Bostock

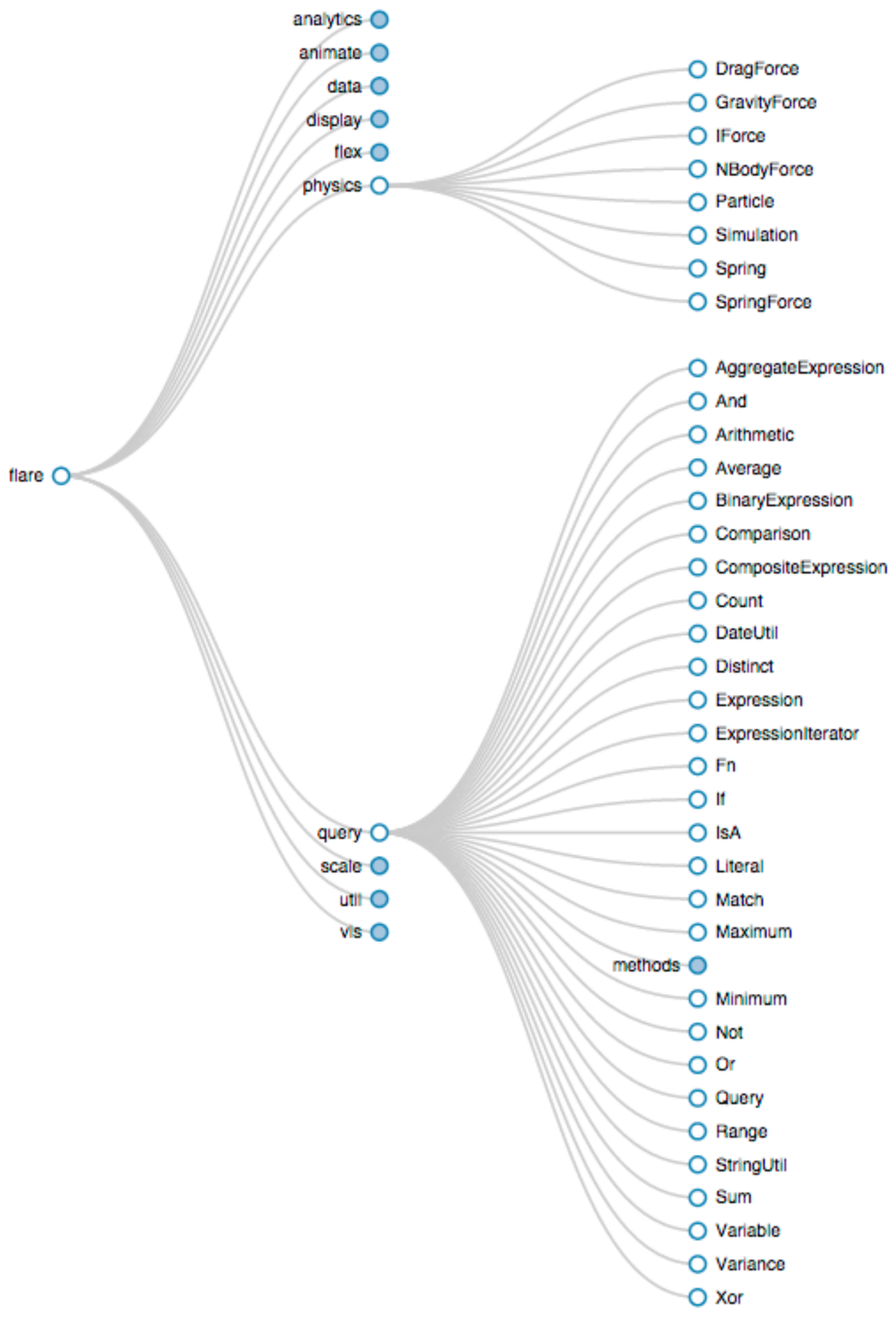
Explicit Tree Visualization

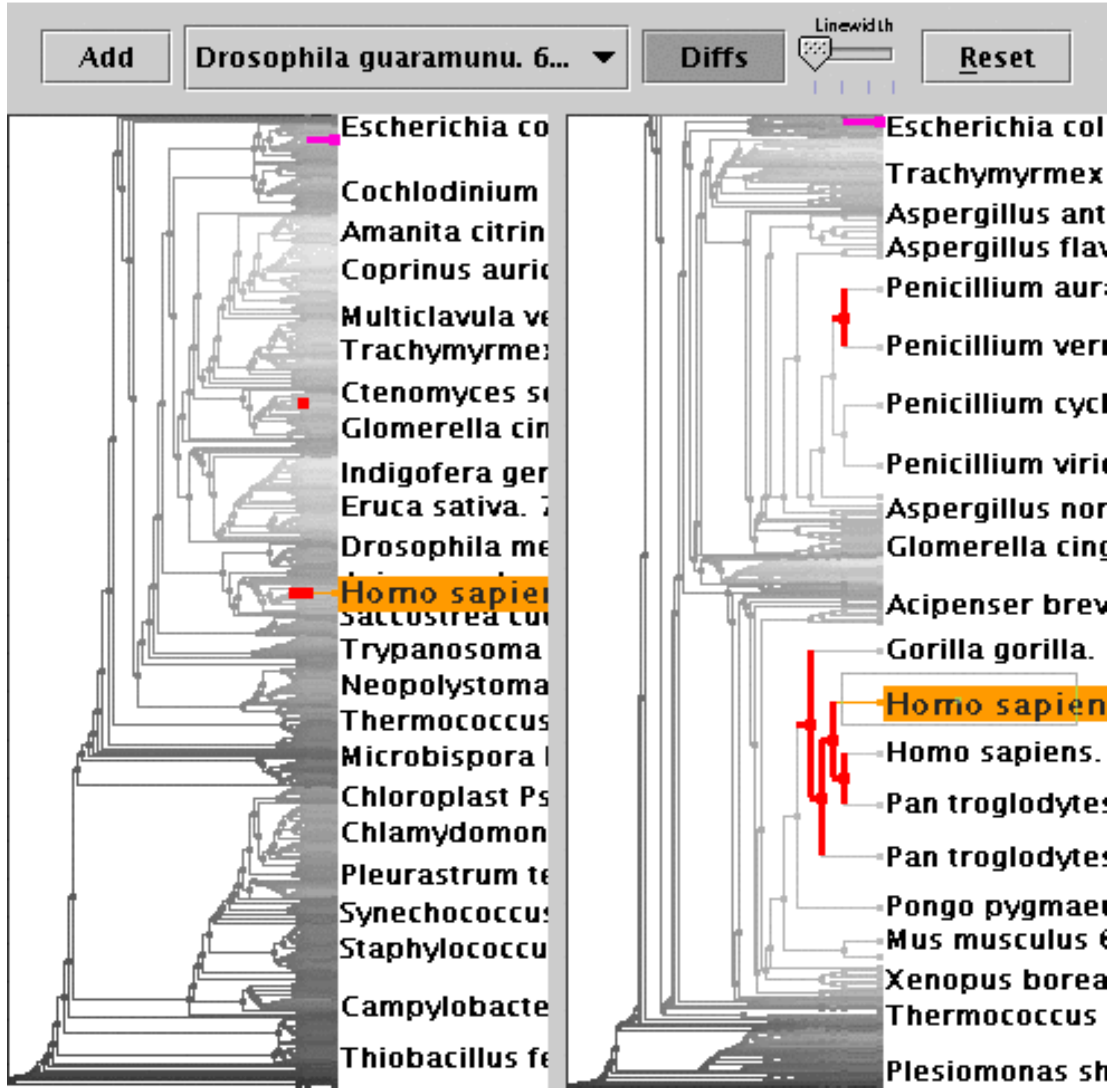
Reingold–Tilford layout

http://billmill.org/pymag-trees/

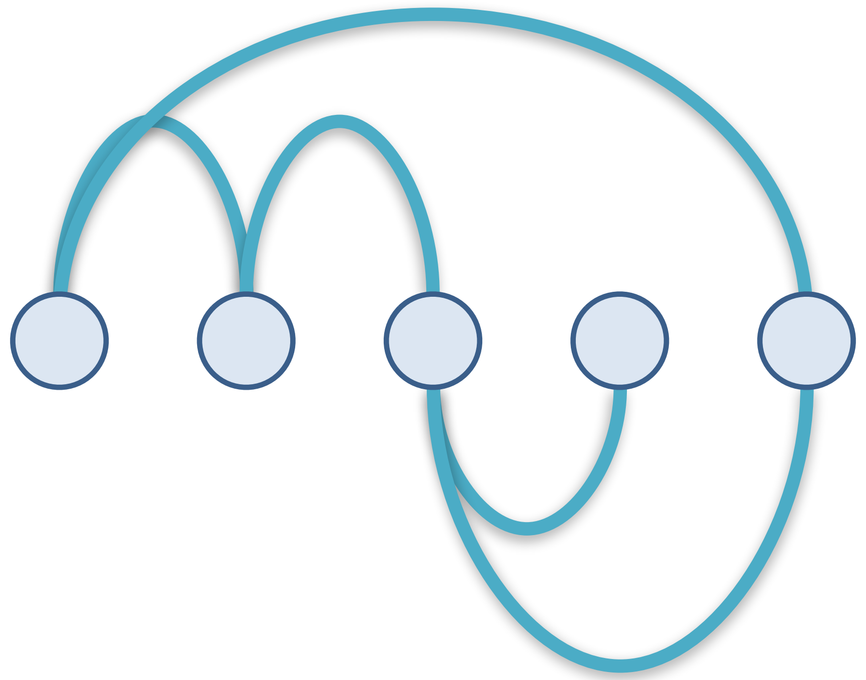


Tree Interaction, Tree Comparison

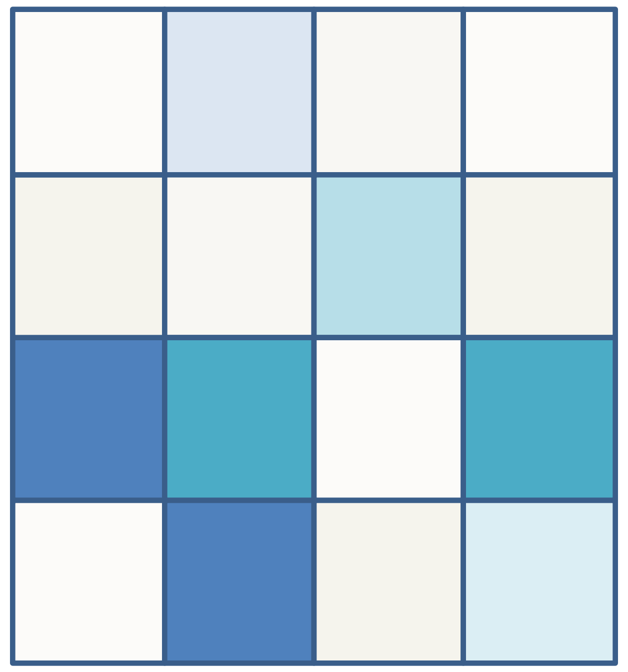




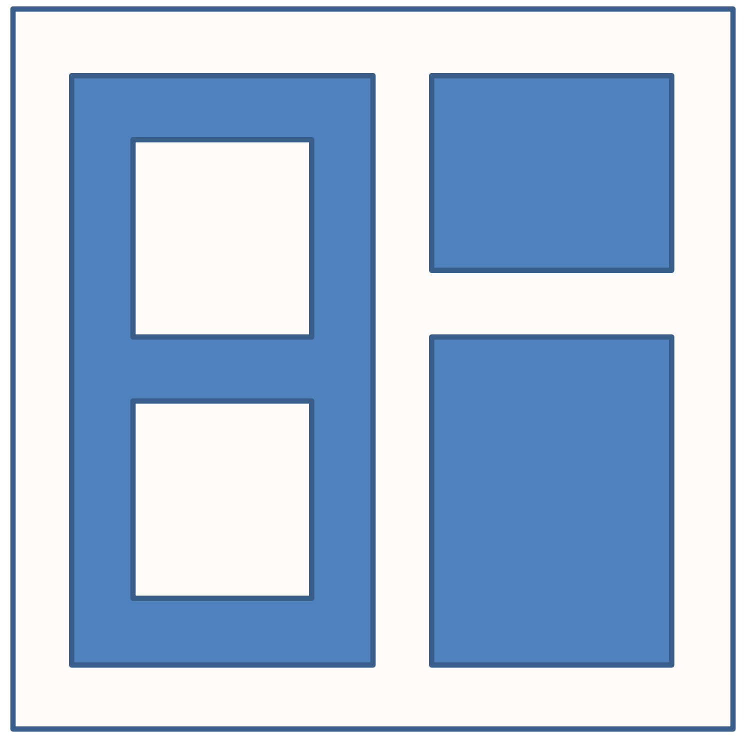
Matrix Representations



Explicit
(Node-Link)



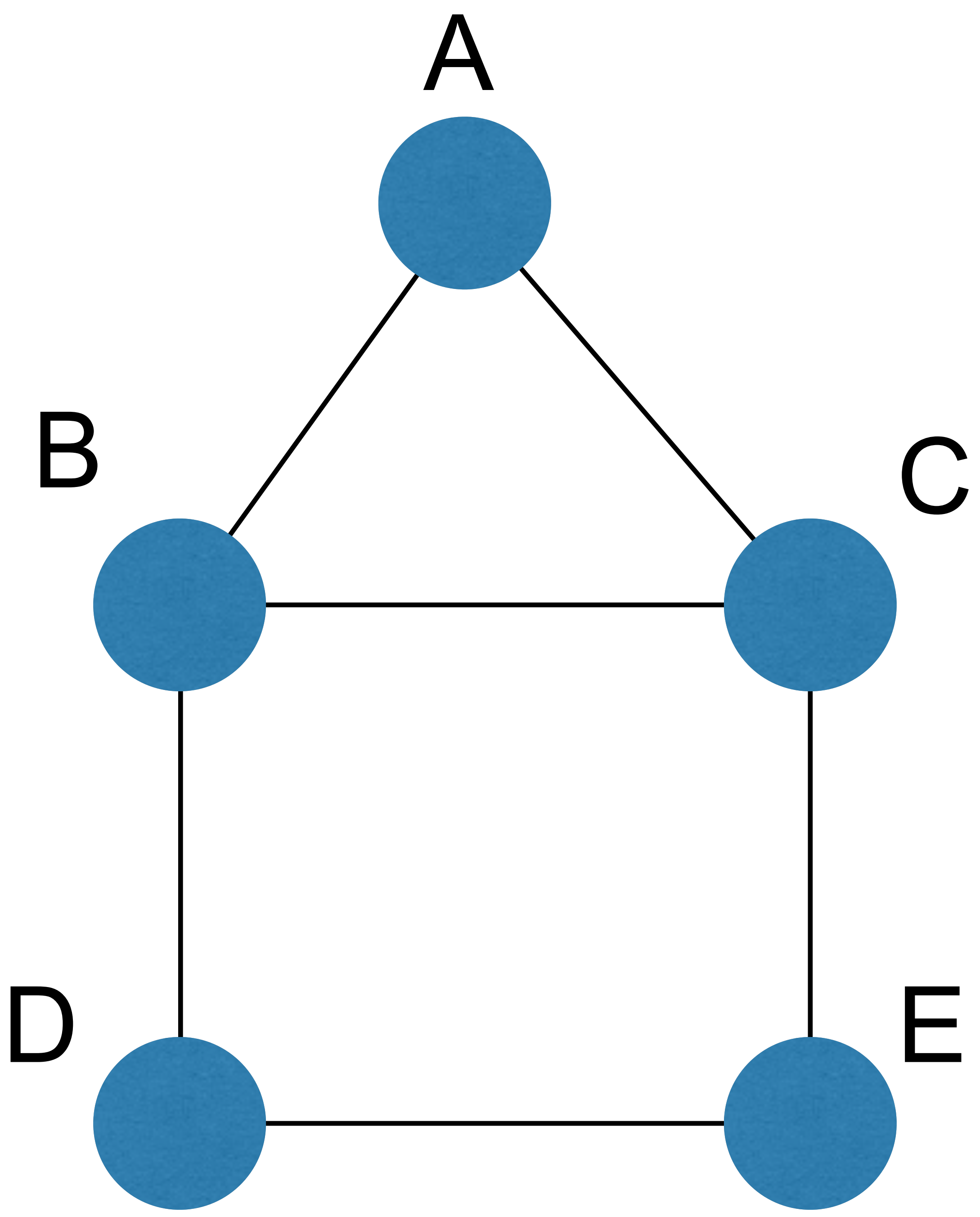
Matrix

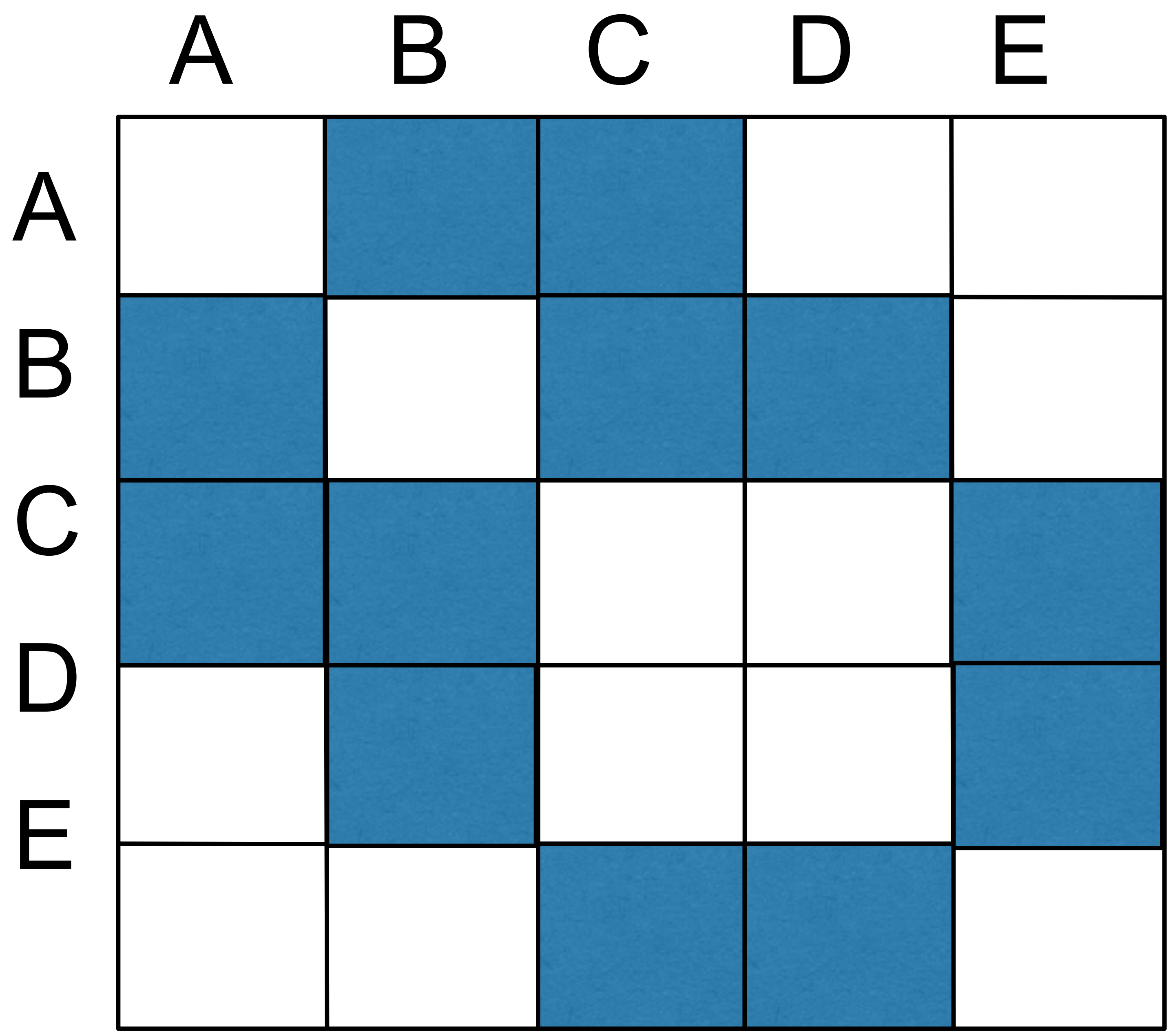


Implicit

Matrix Representations

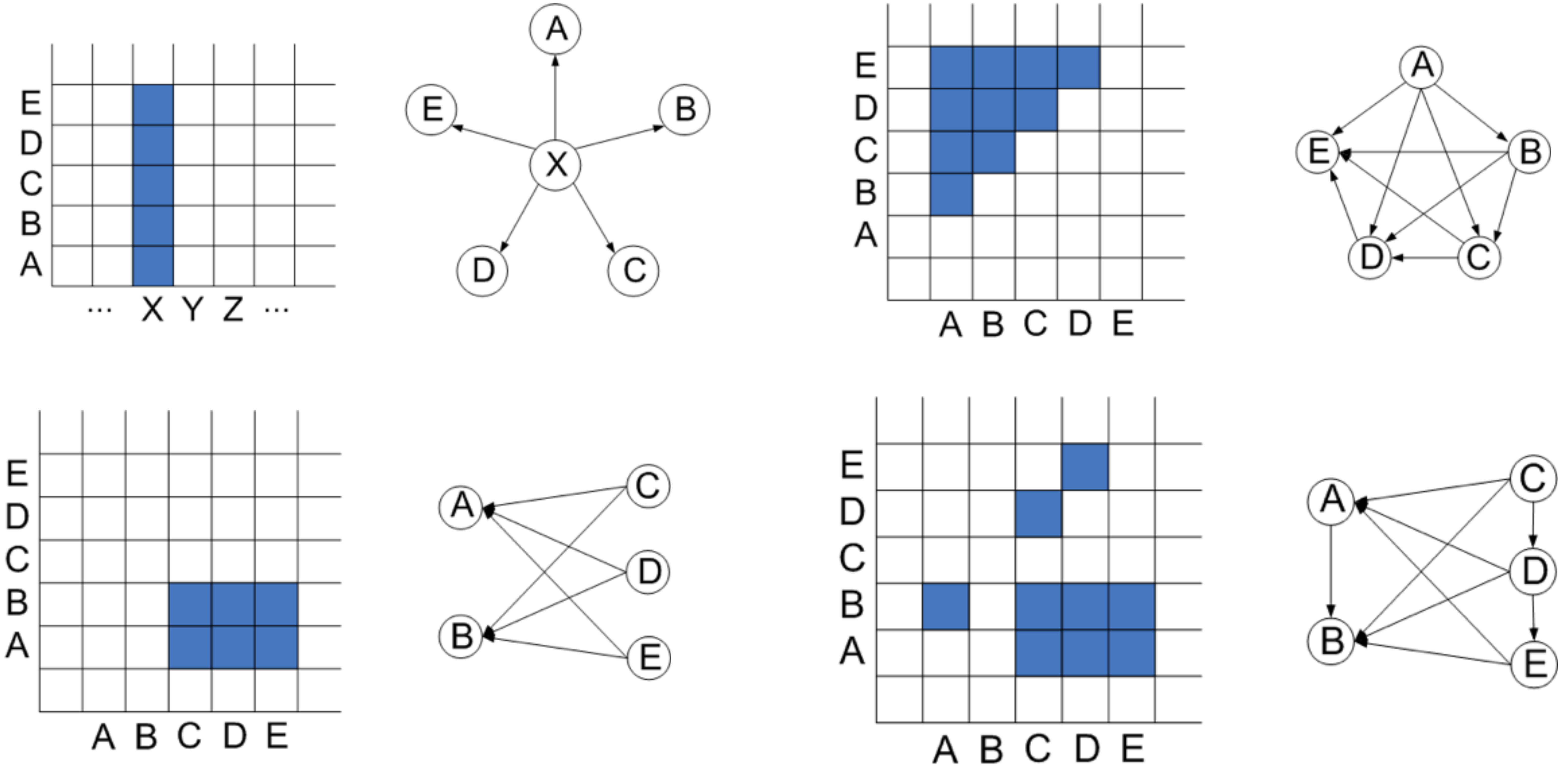
Instead of node link diagram, use adjacency matrix





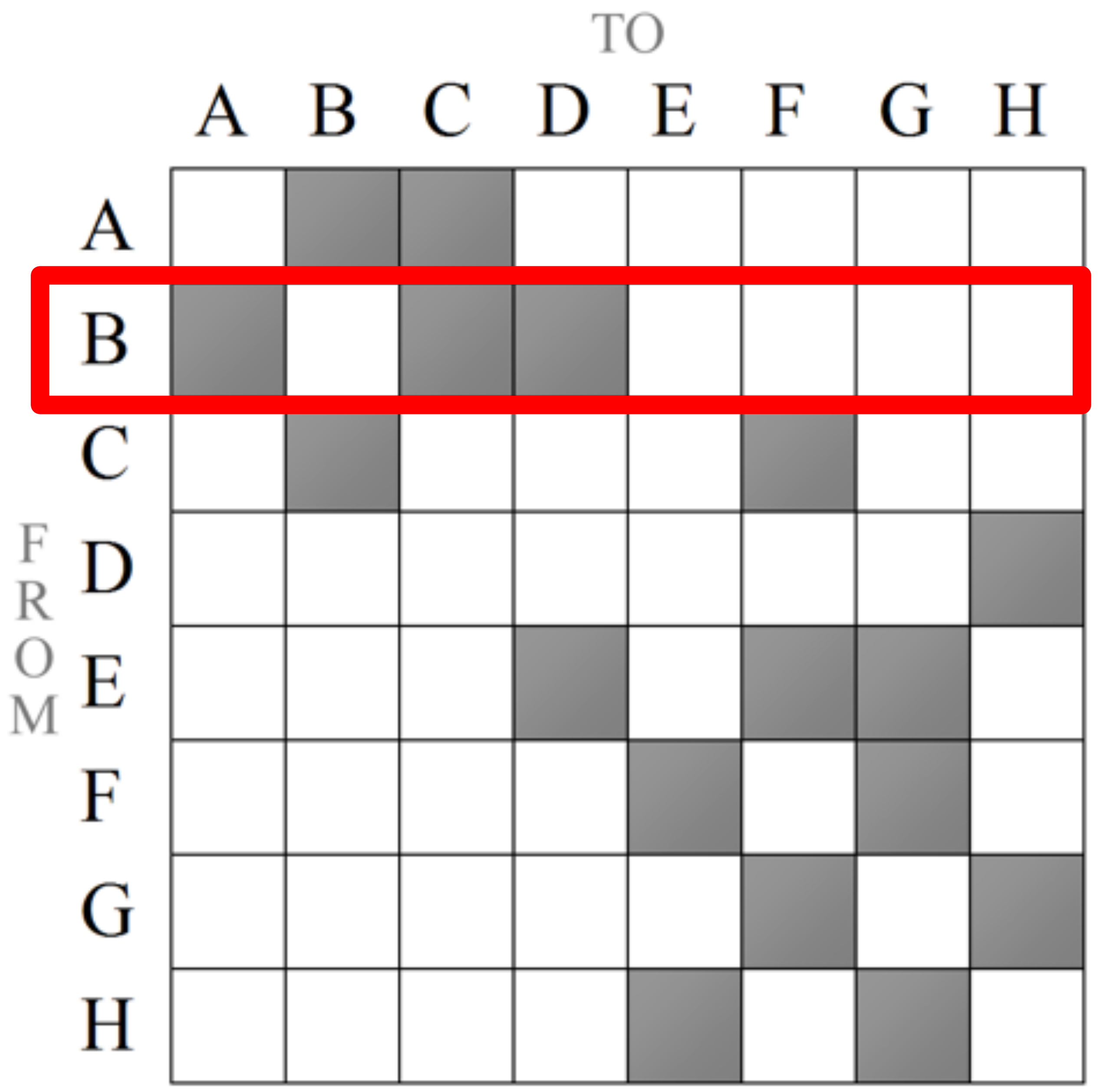
Matrix Representations

Examples:

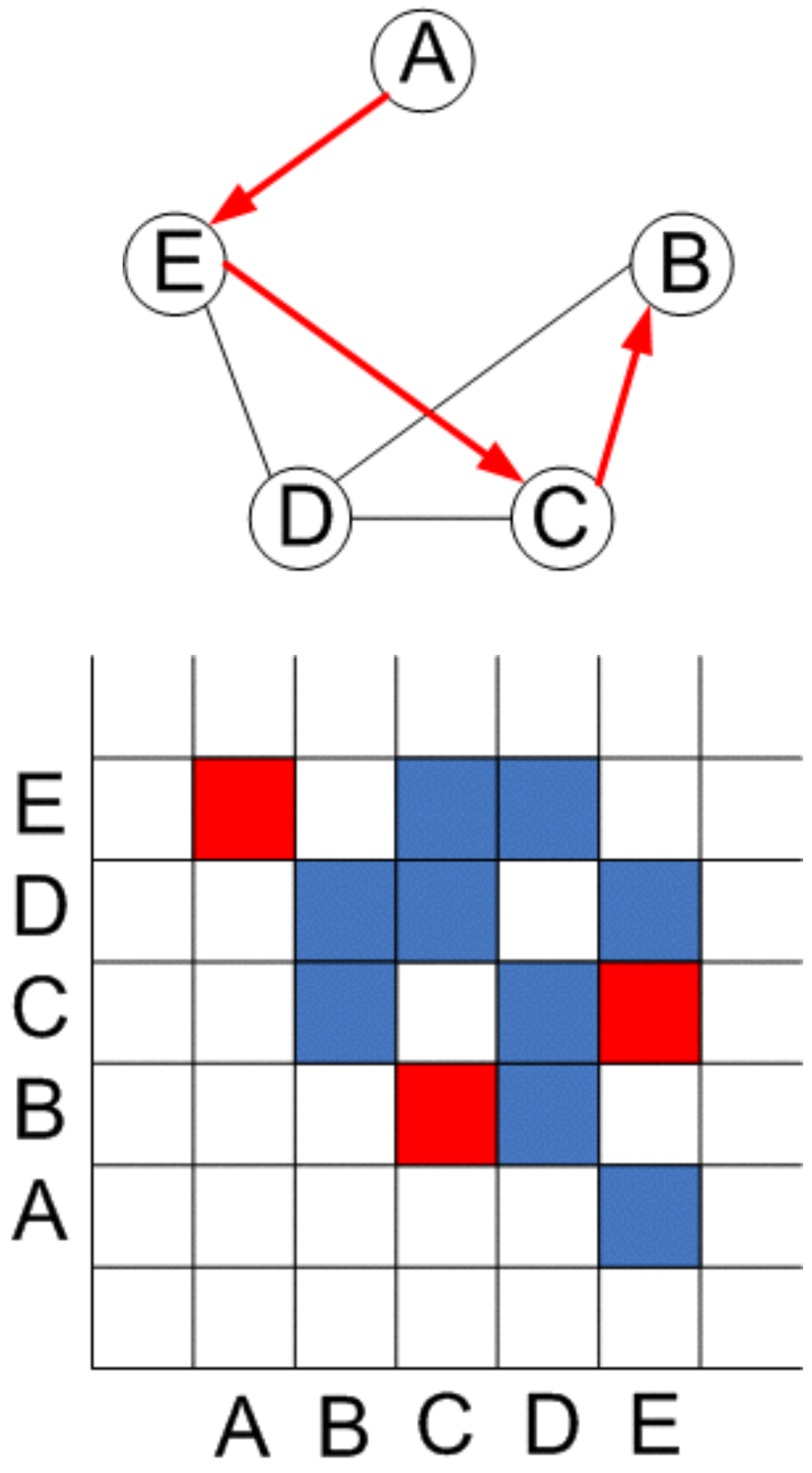


HJ Schulz 2007

Matrix Representations

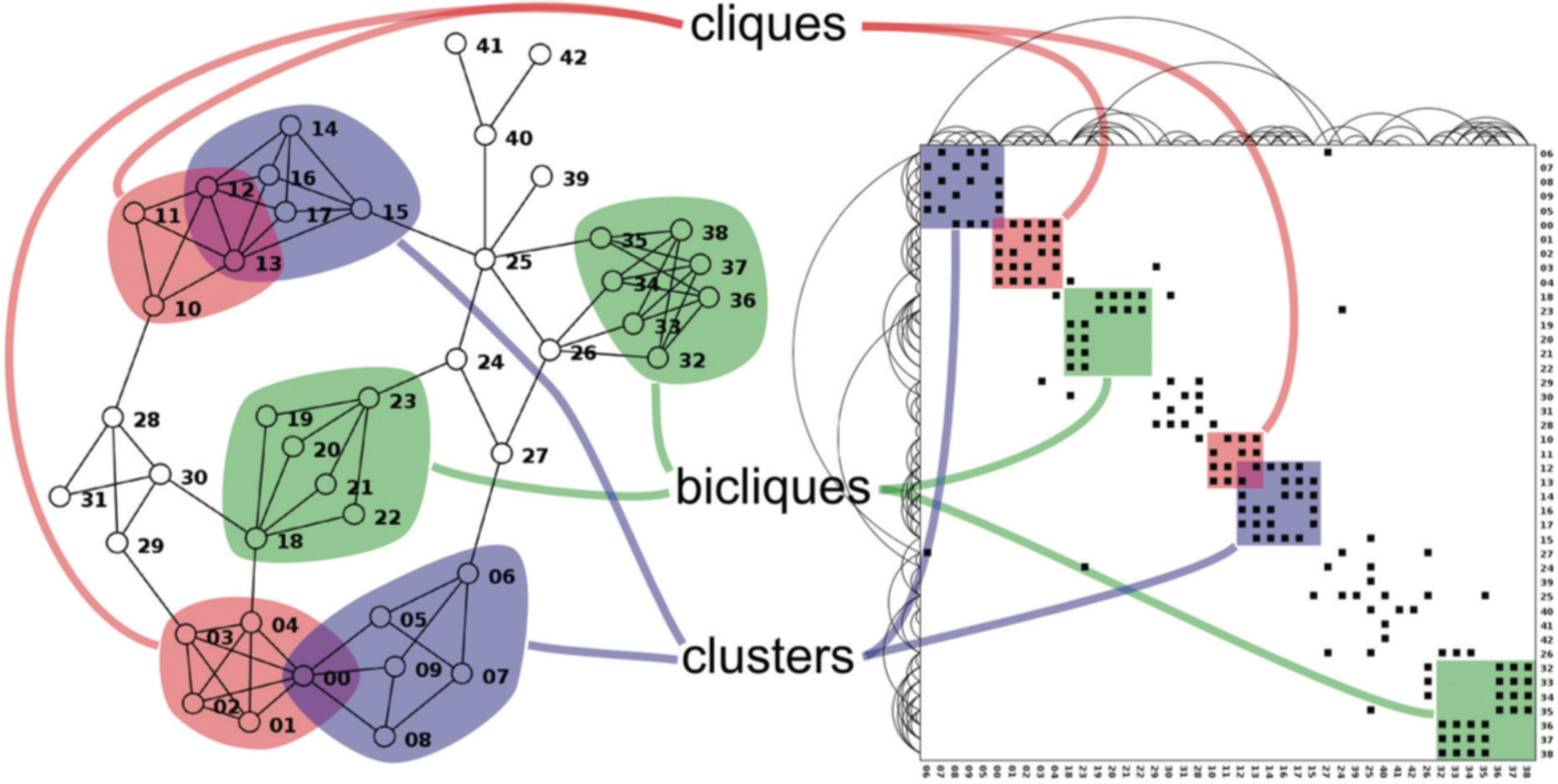


Well suited for
neighborhood-related TBTs



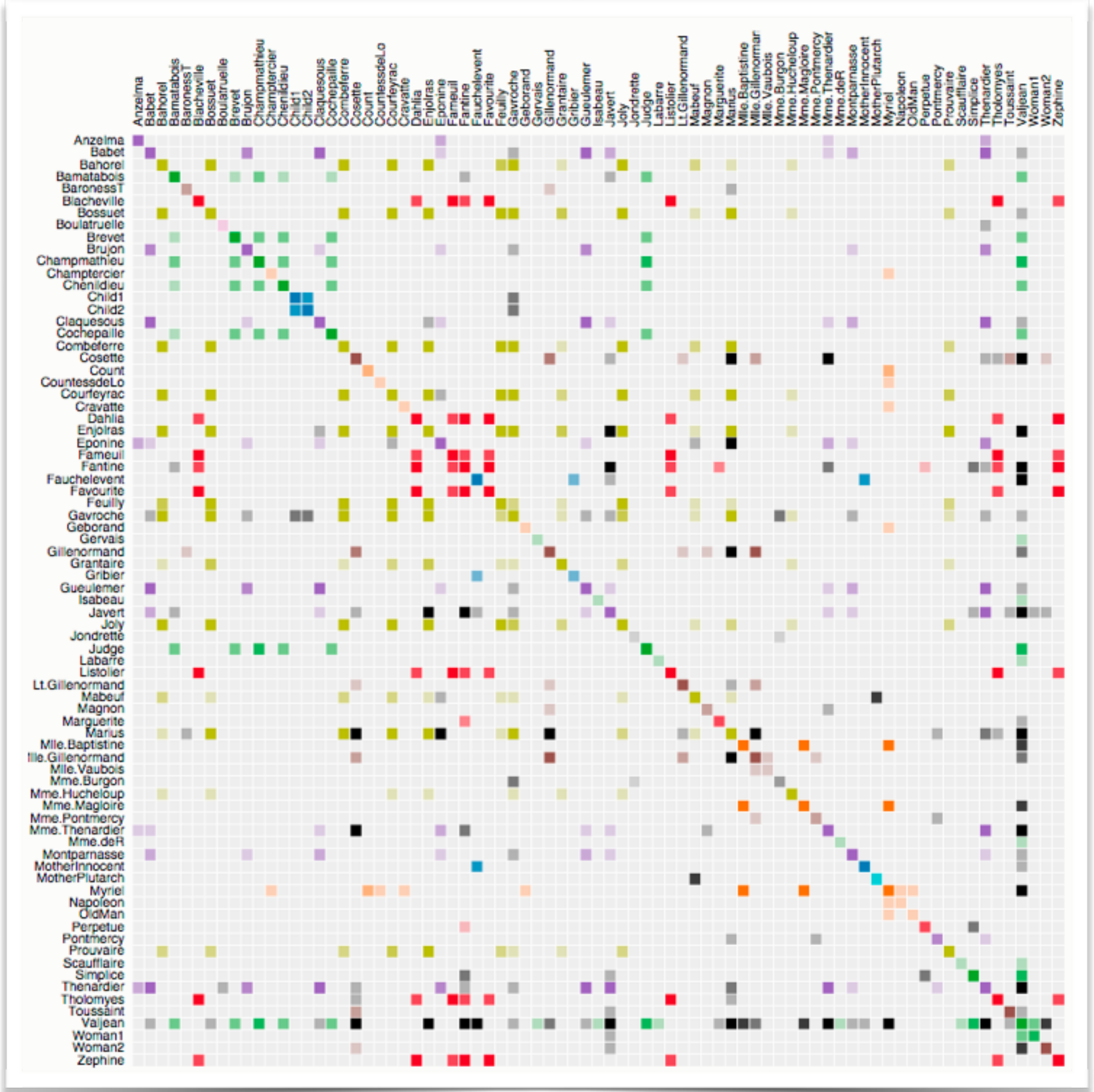
Not suited for
path-related TBTs

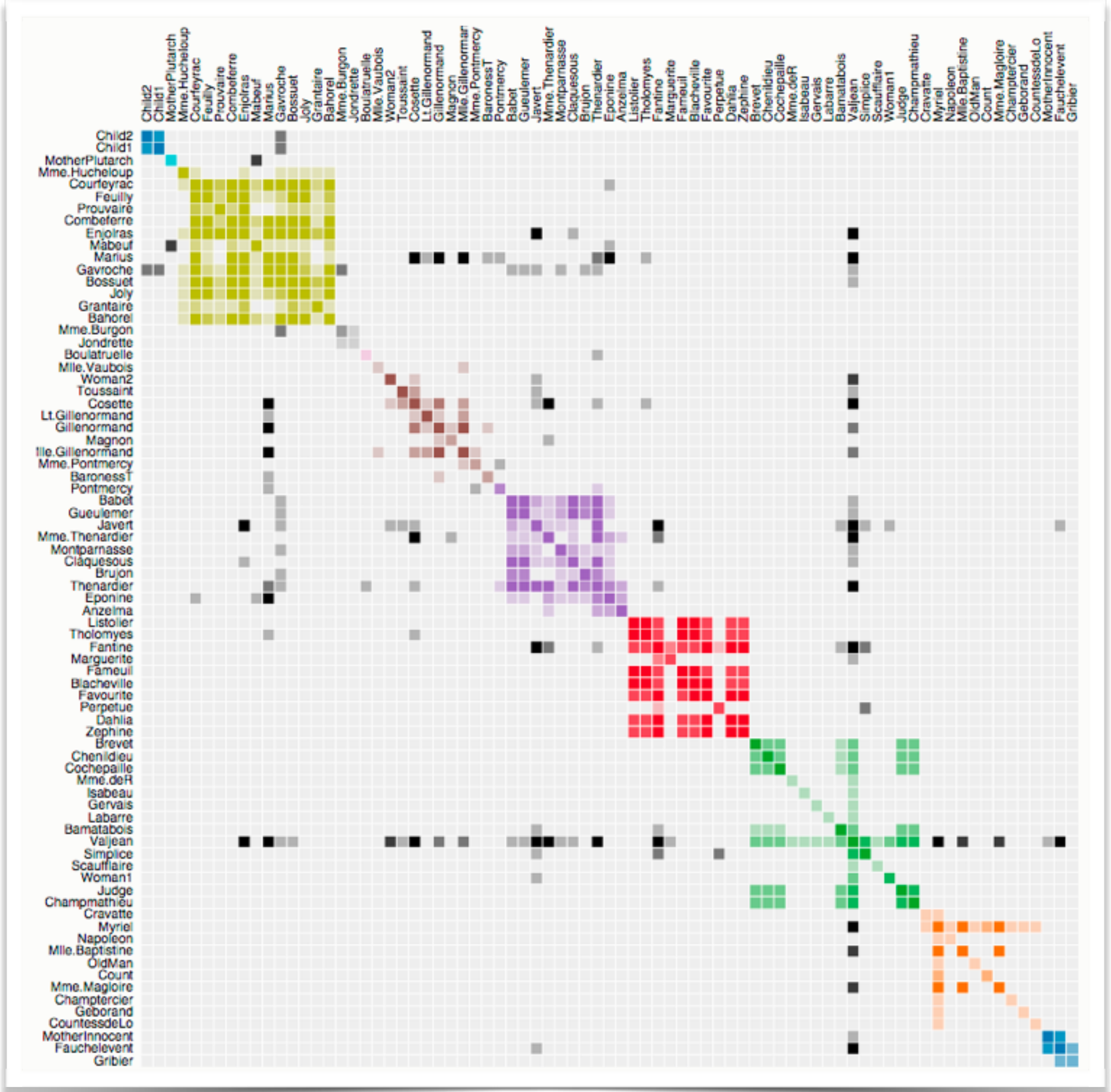
van Ham et al. 2009
Shen et al. 2007



McGuffin 2012

Order Critical!





Matrix Representations

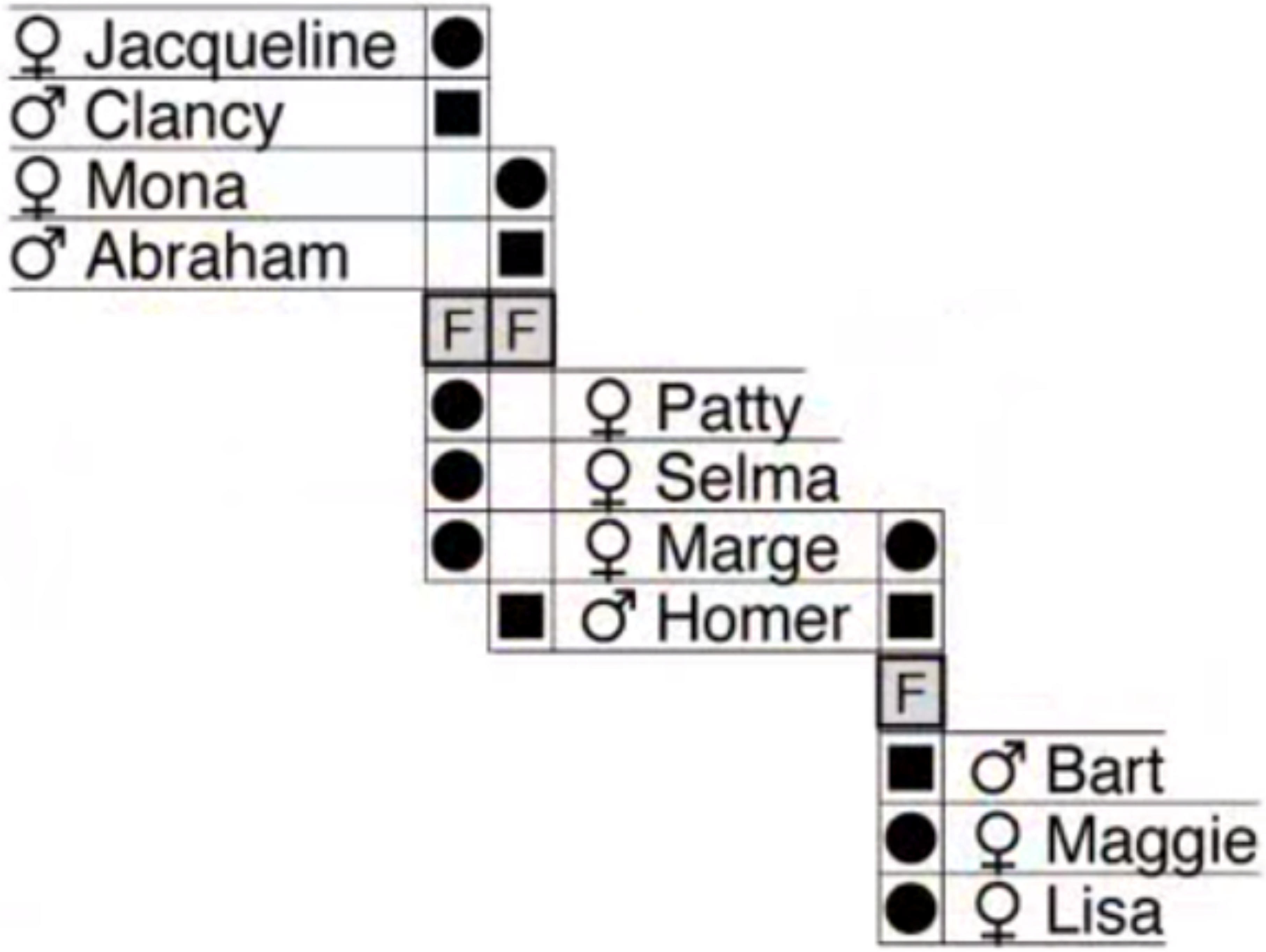
Pros:

- can represent **all graph classes** except for hypergraphs
- puts **focus on the edge set**, not so much on the node set
- simple grid -> **no elaborate layout** or rendering needed
- well suited for **ABT on edges** via coloring of the matrix cells
- well suited for **neighborhood-related TBTs** via traversing rows/columns

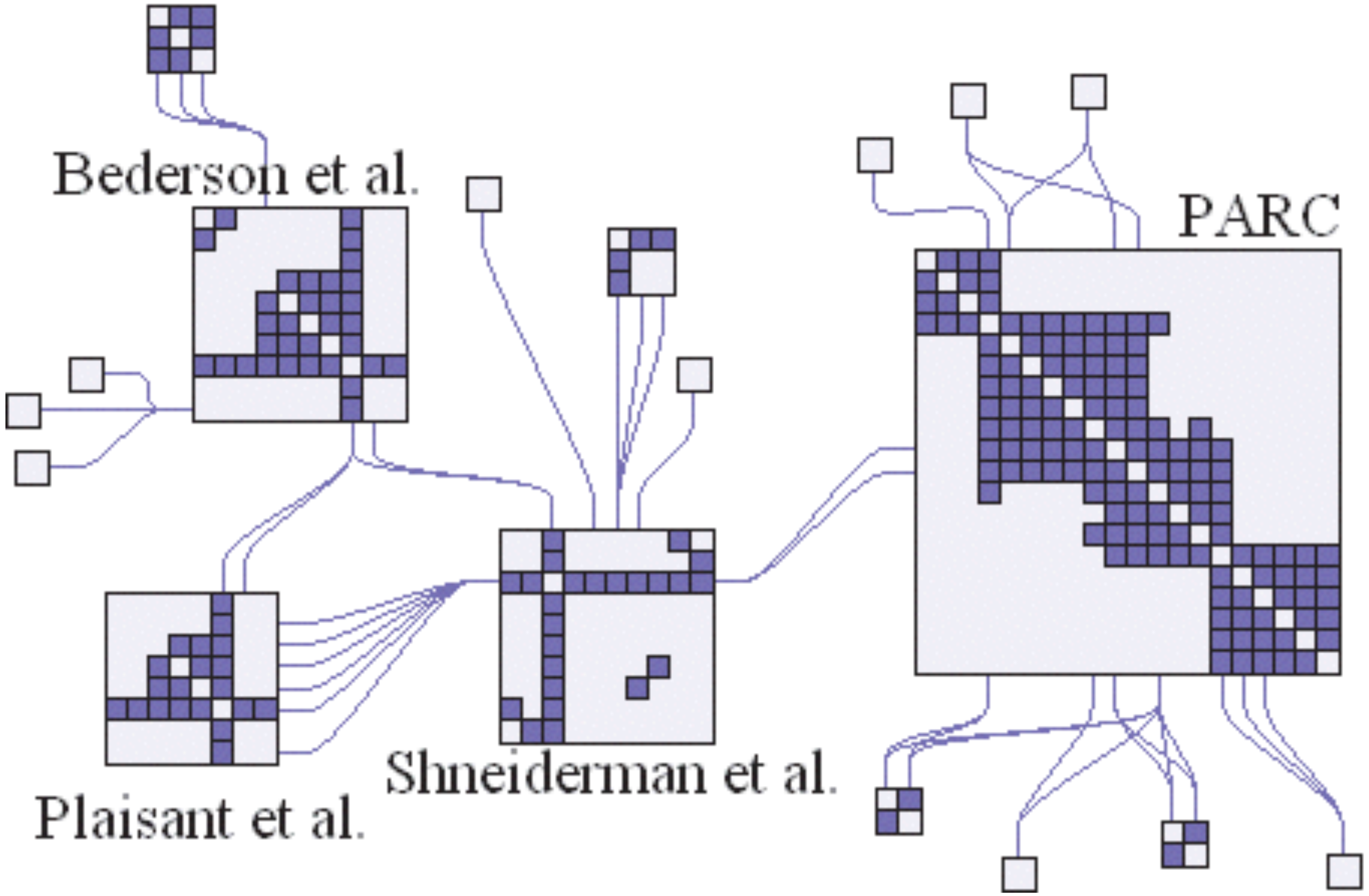
Cons:

- quadratic screen space requirement (any possible edge takes up space)
- not suited for path-related TBTs

Special Case: Genealogy

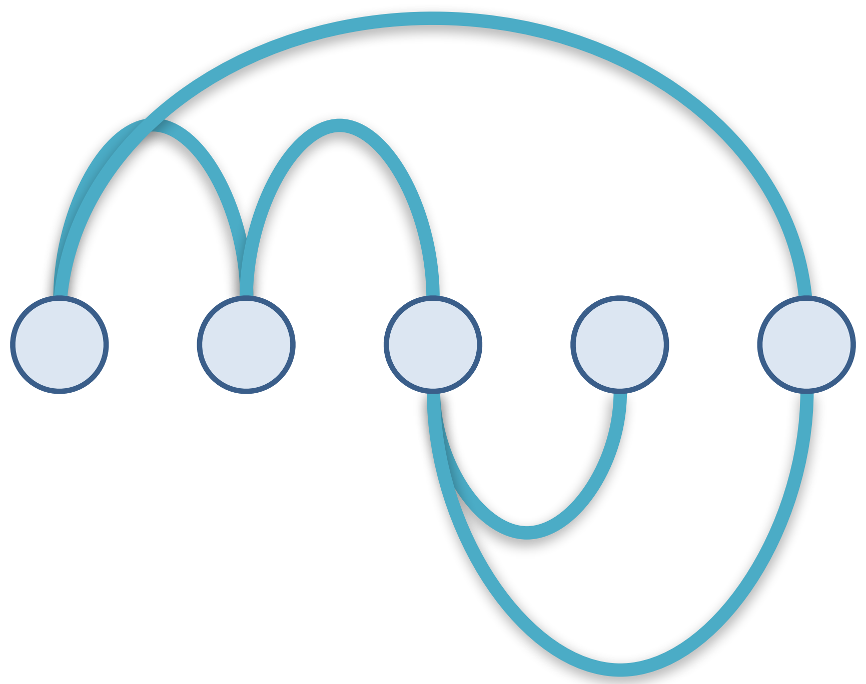


Hybrid Explicit/Matrix

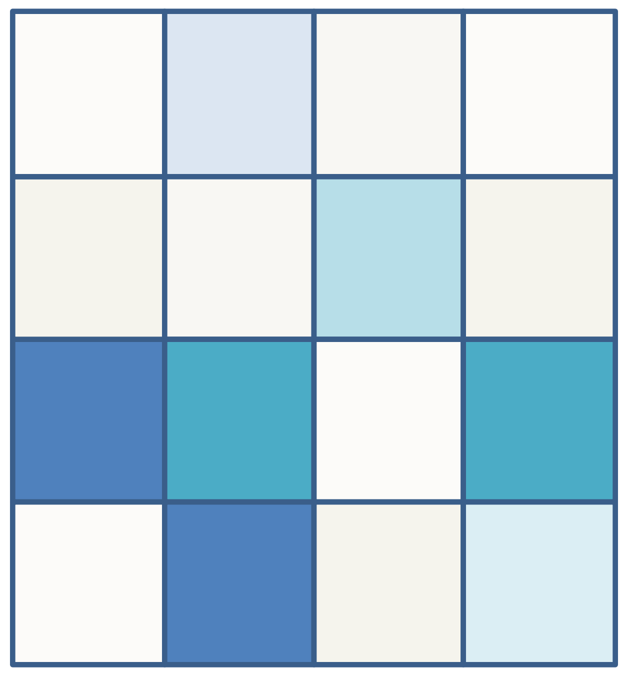


NodeTrix
[Henry et al. 2007]

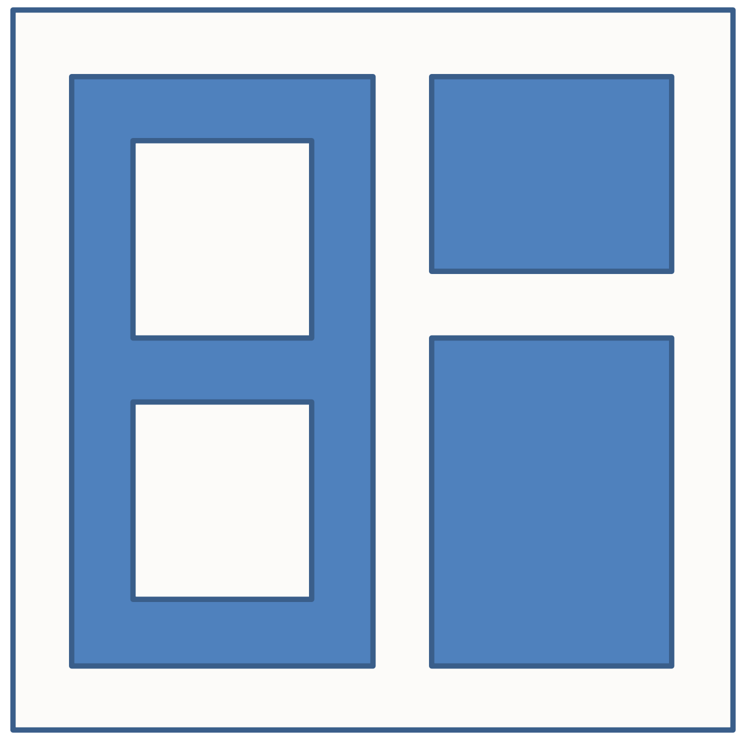
Implicit Layouts



Explicit
(Node-Link)



Matrix



Implicit

Explicit vs. Implicit Tree Vis

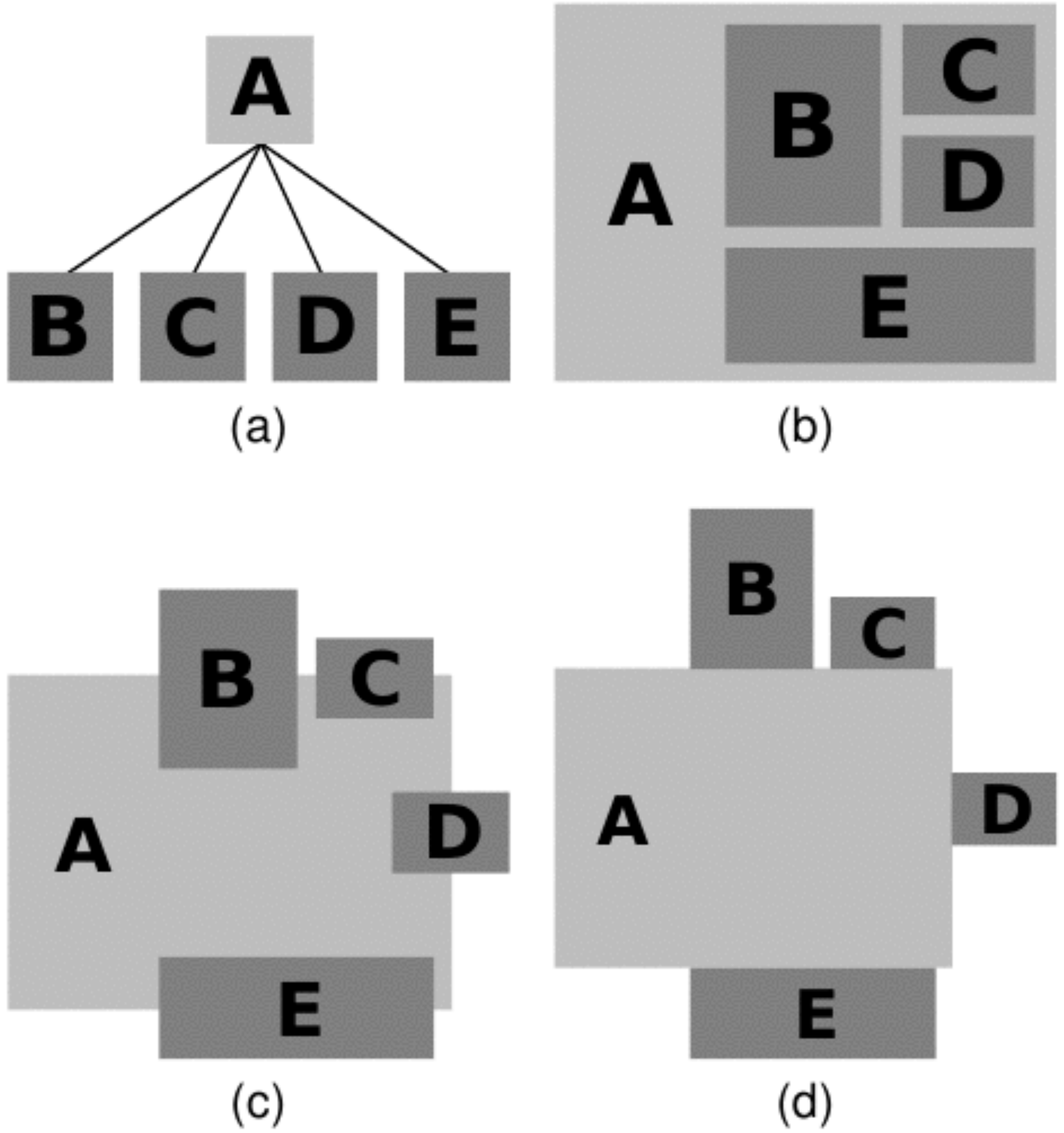
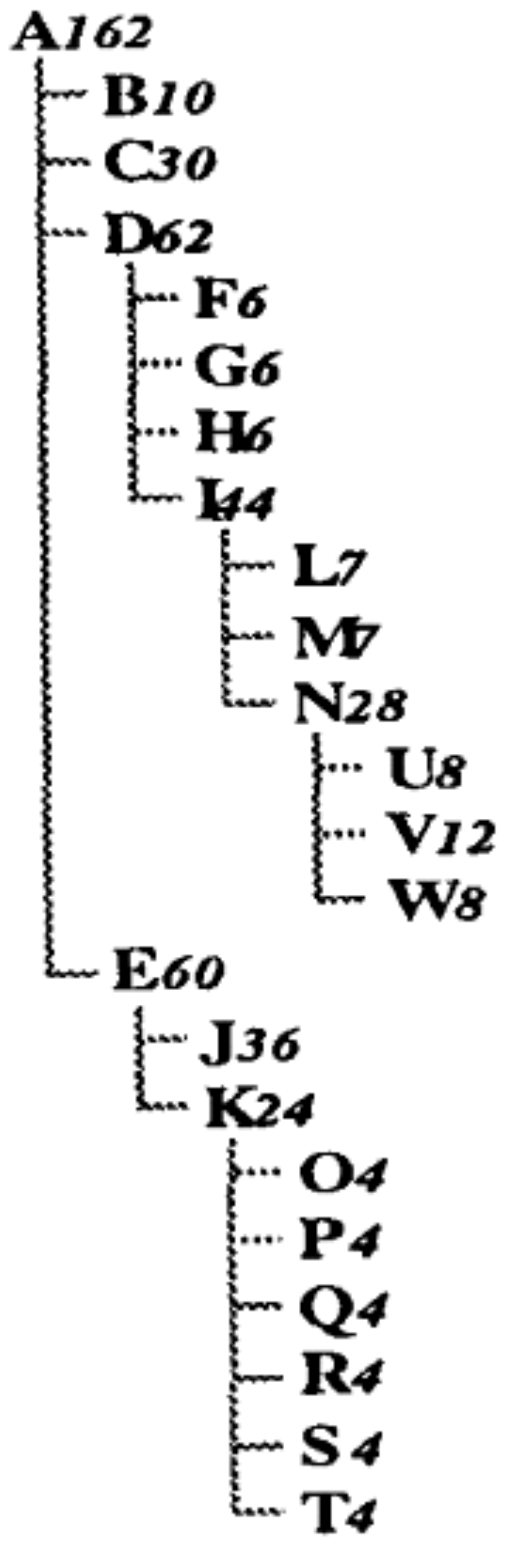
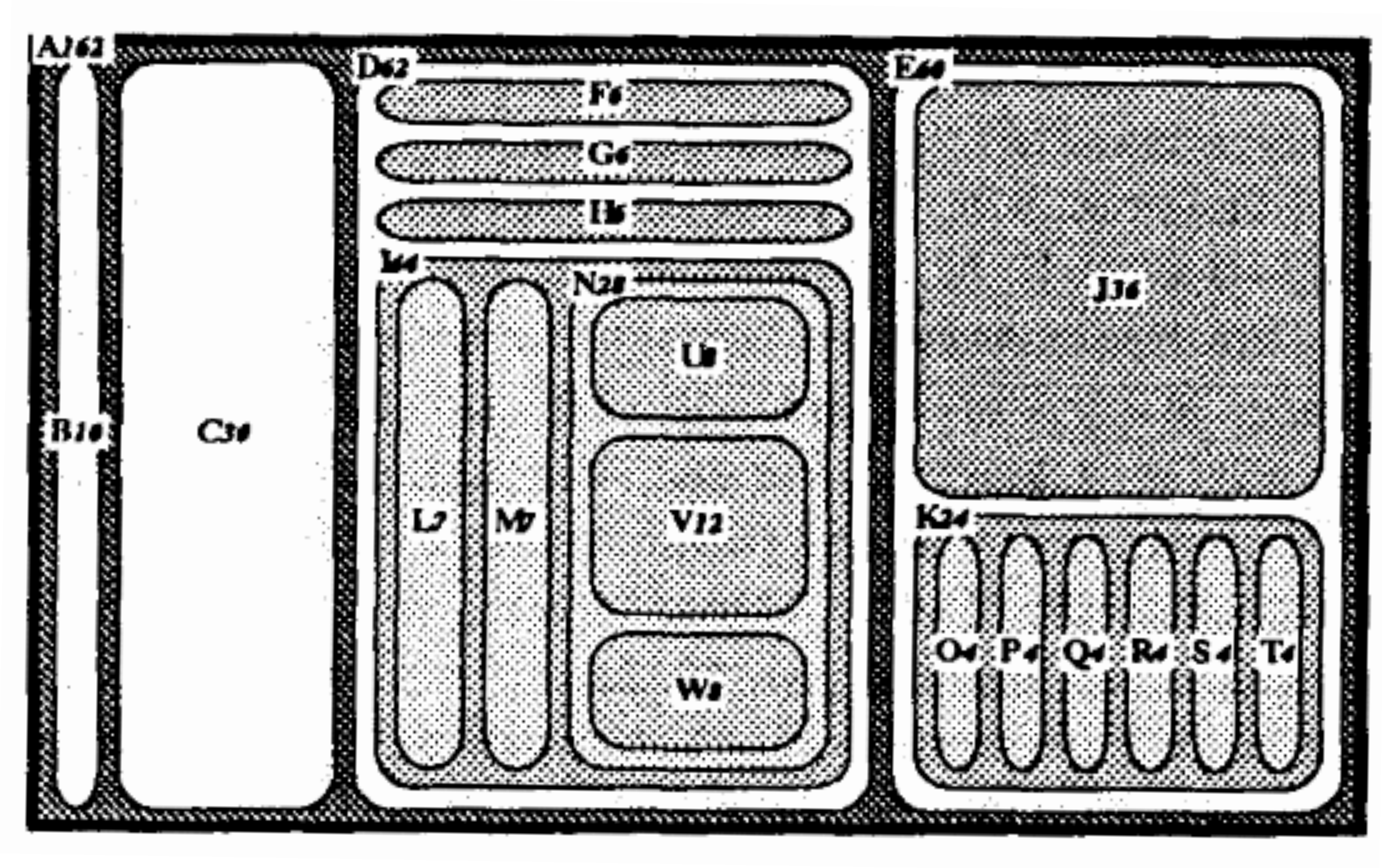


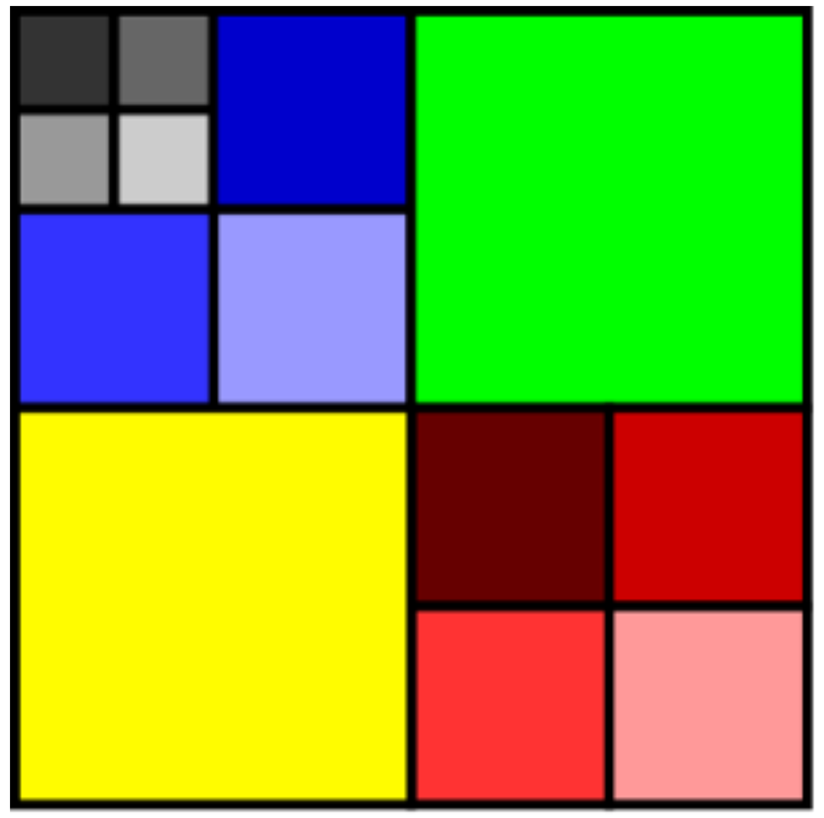
Fig. 2. (a) Explicit, node-link layout, (b) Implicit layout by inclusion, (c) Implicit Layout by overlap, (d) Implicit layout by adjacency.

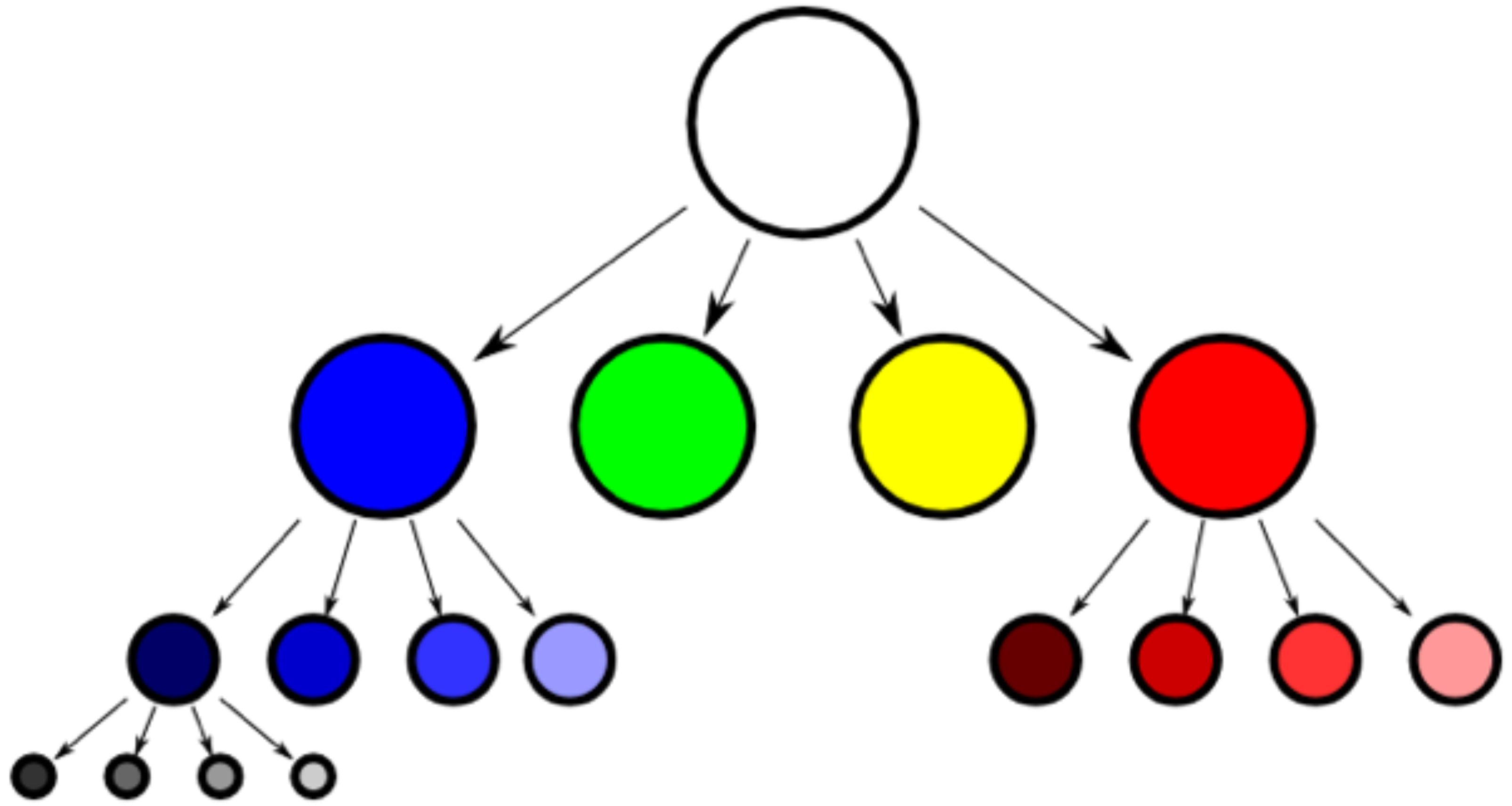
Schulz 2011

Tree Maps



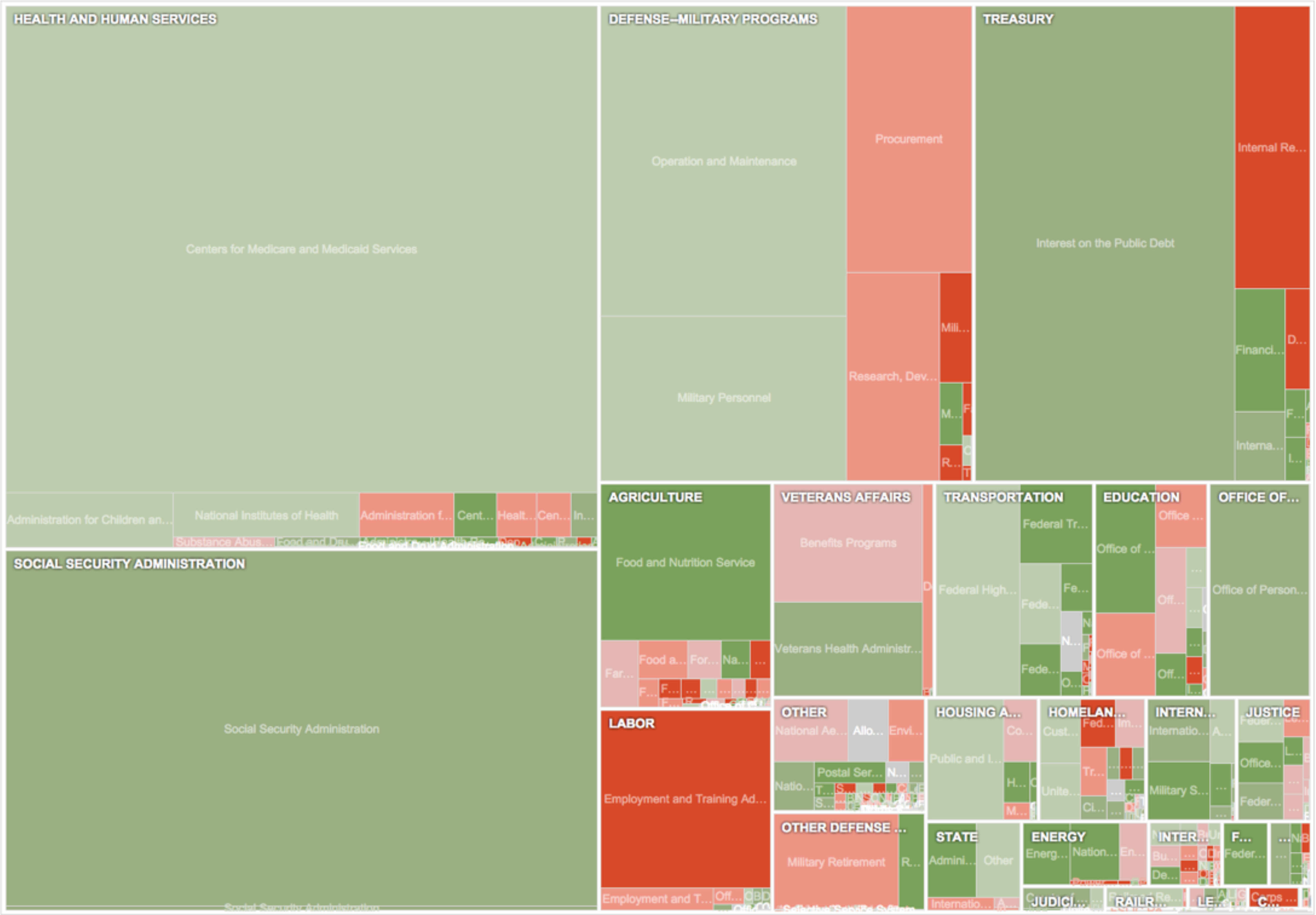




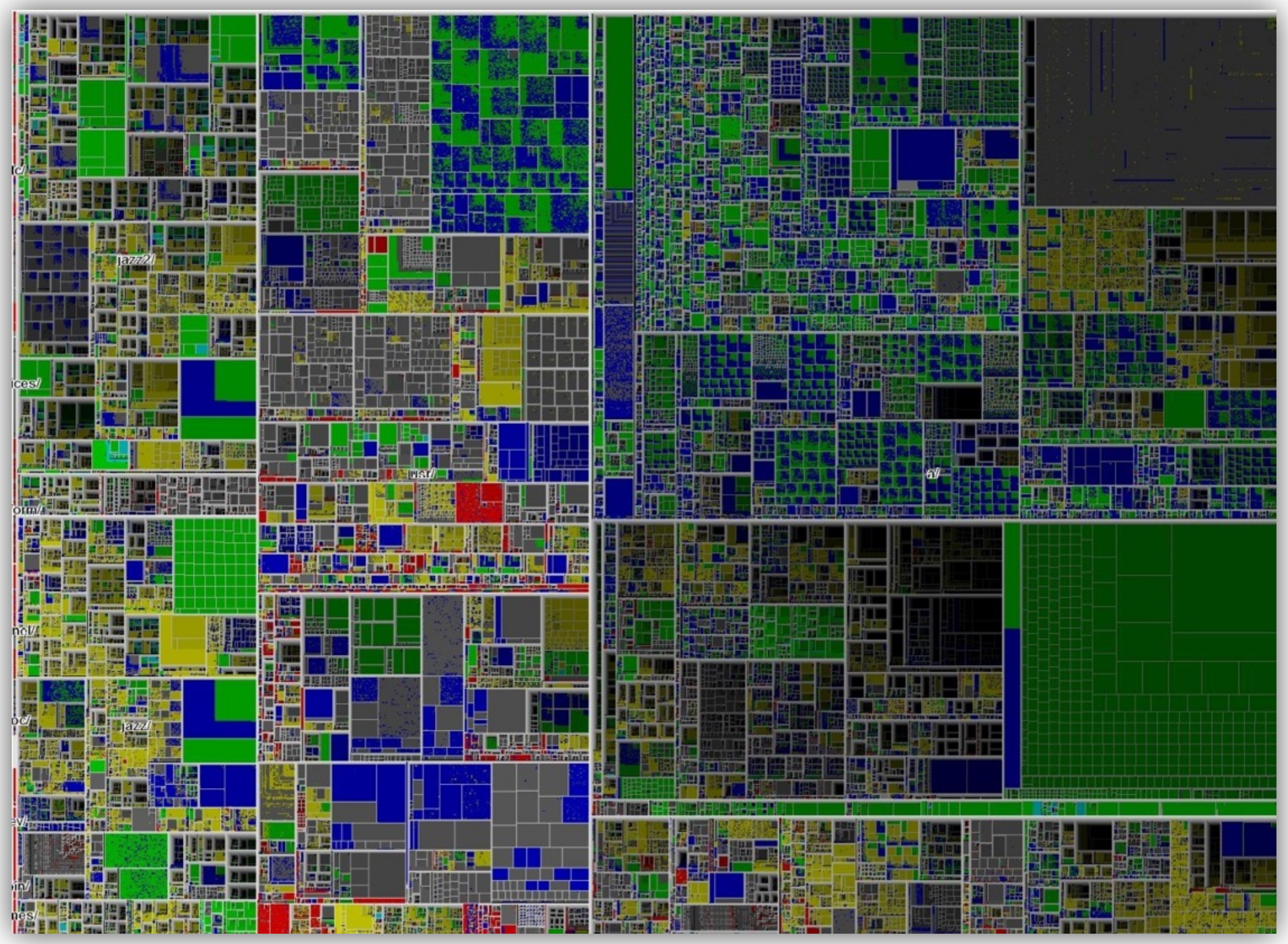


Johnson and Shneiderman 1991

Zoomable Treemap



Example: Interactive TreeMap of a Million Items

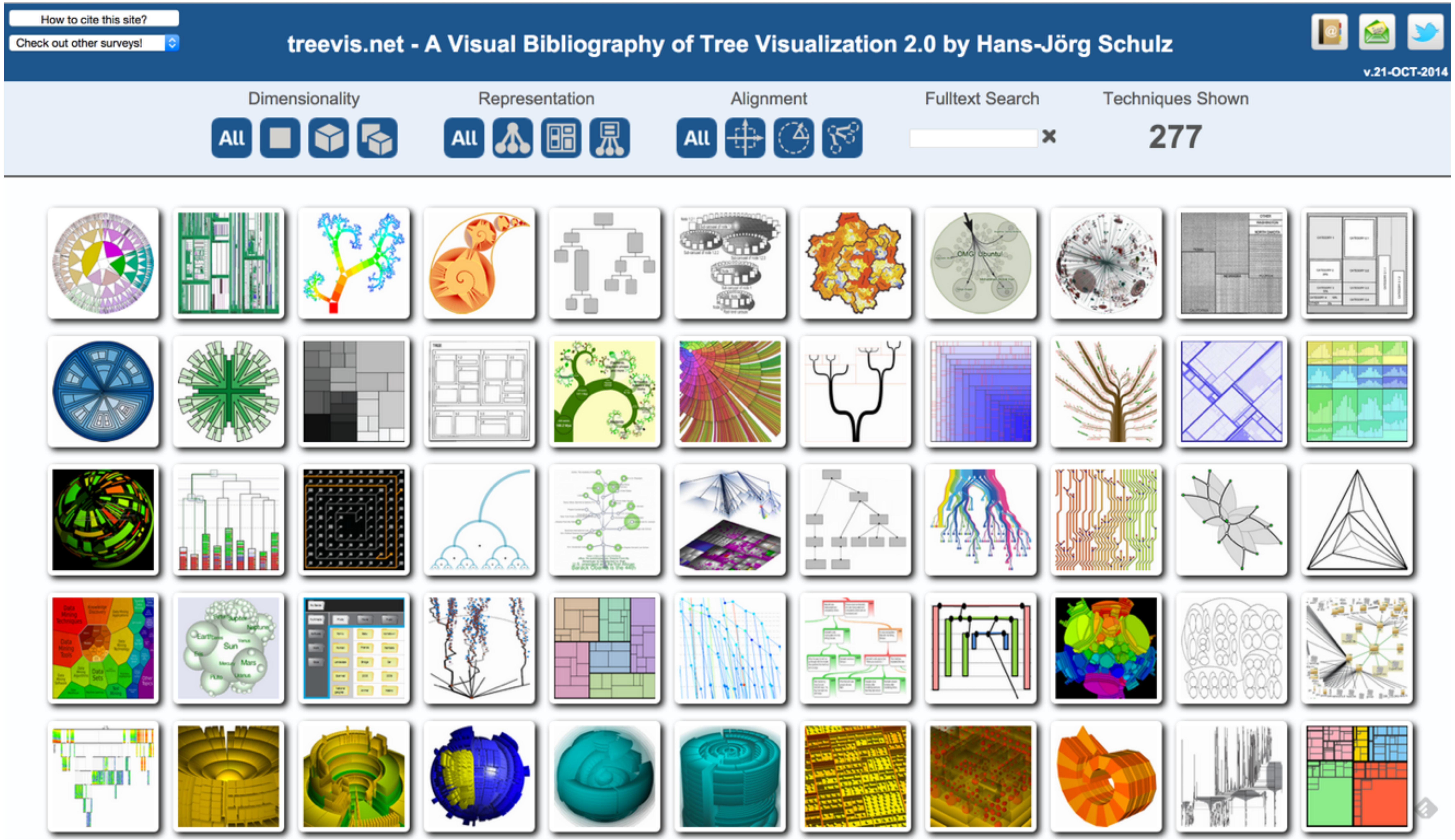


Fekete et al. 2002

Sunburst: Radial Layout

[Sunburst by John Stasko, Implementation in Caleydo by Christian Partl]

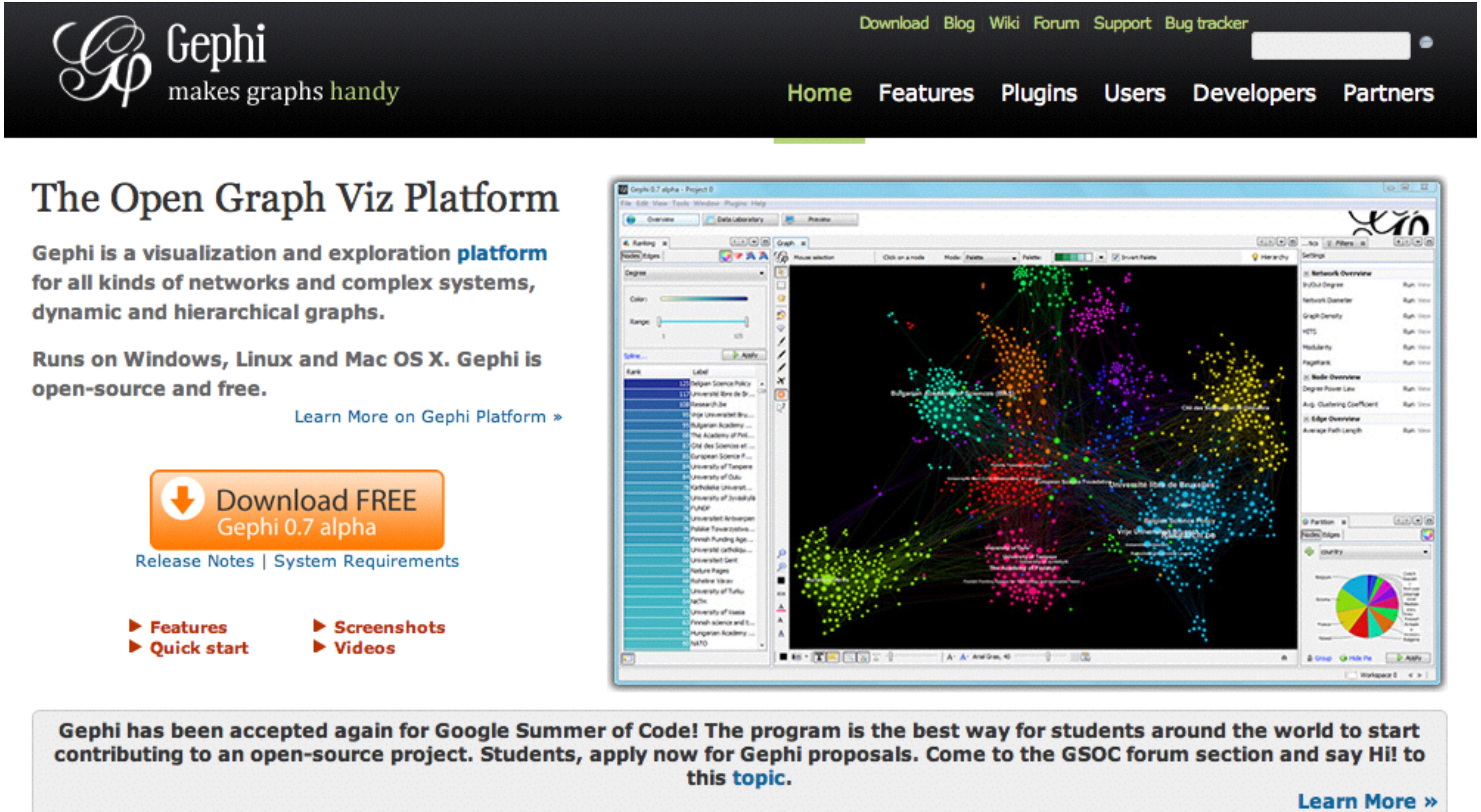
Tree Visualization Reference



Graph Tools & Applications

Gephi

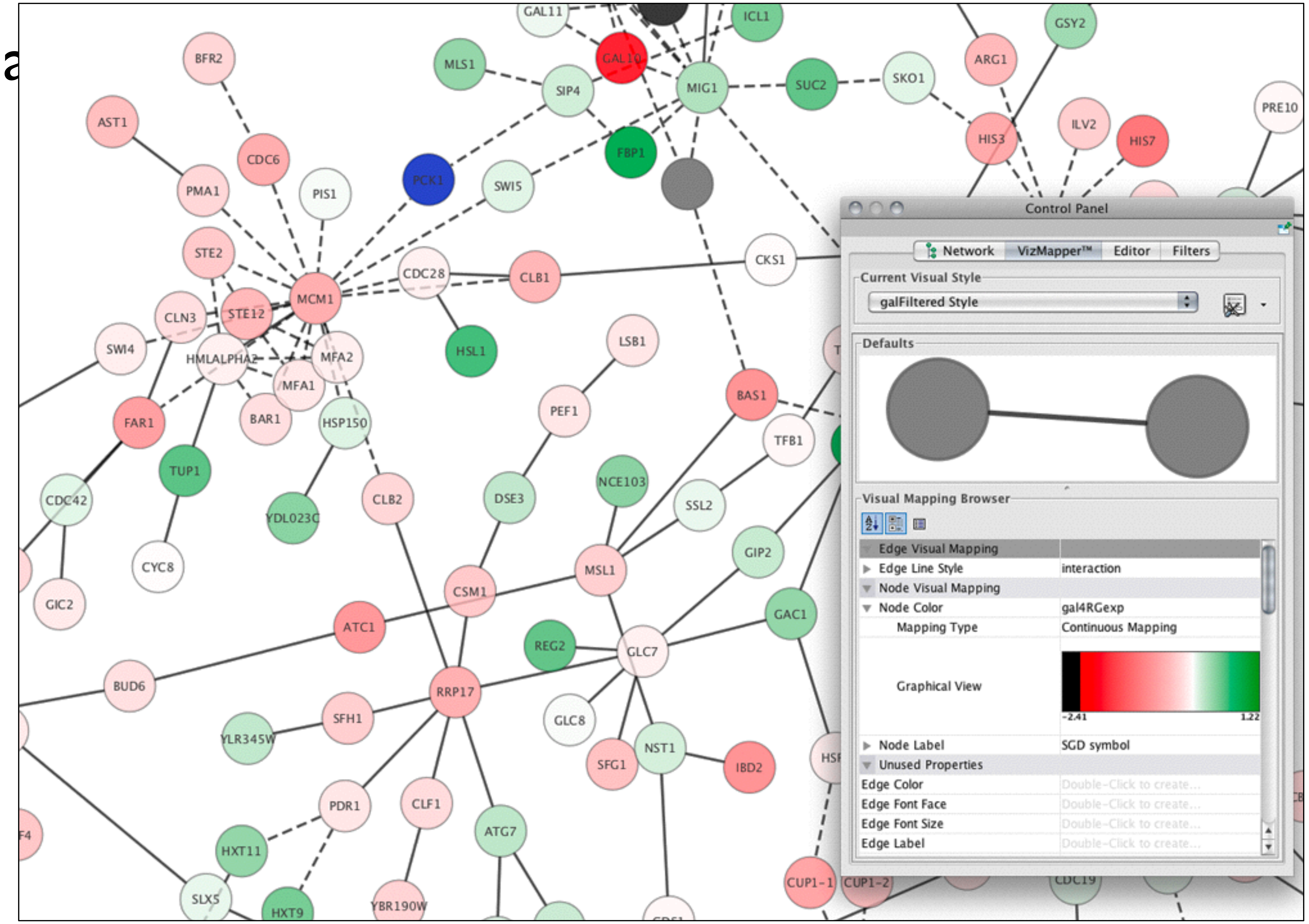
http://gephi.org



Cytoscape

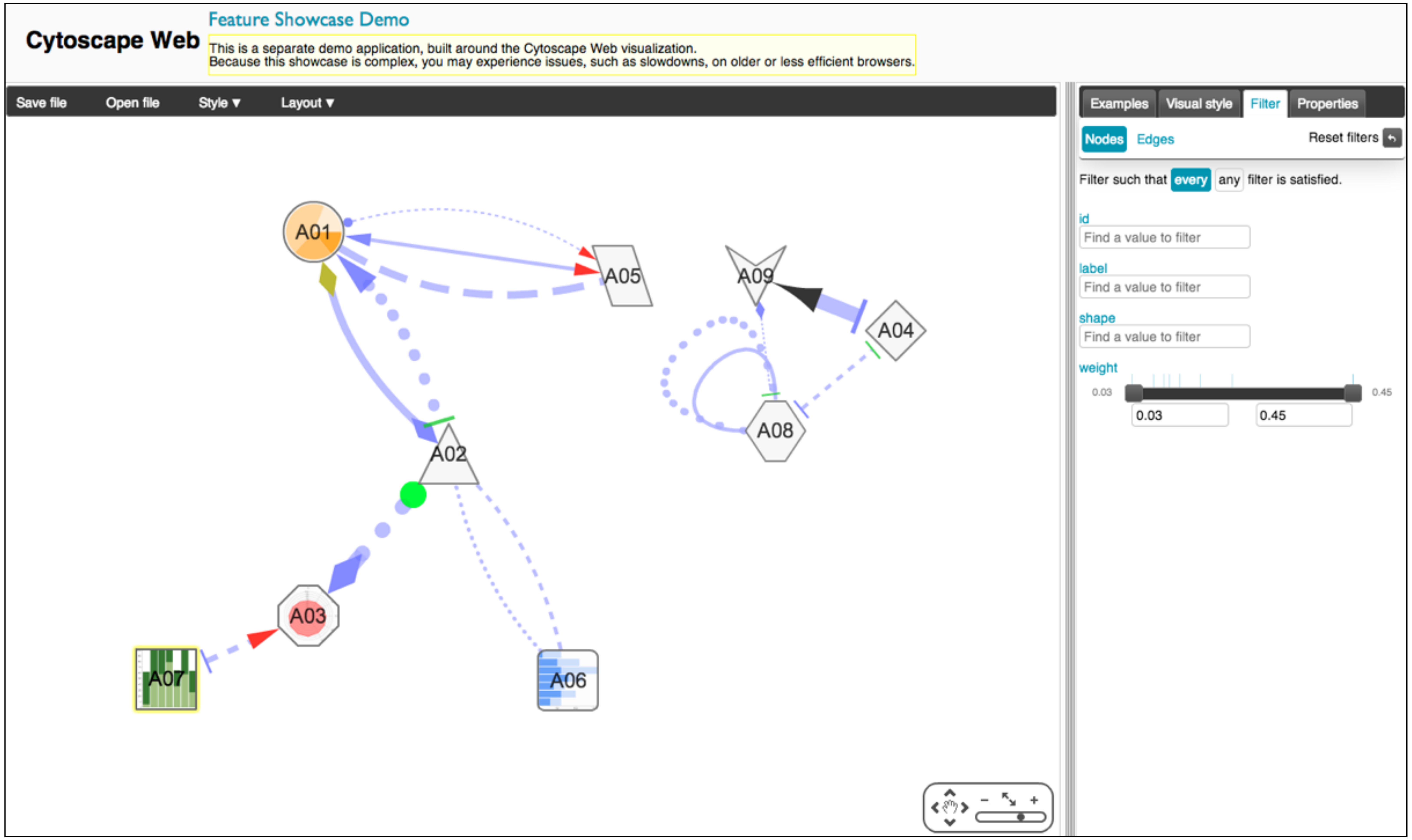
http://www.cytoscape.org/

Open source pla



Cytoscape Web

http://cytoscapeweb.cytoscape.org/



NetworkX

https://networkx.github.io/

NetworkX

NetworkX Home | Documentation | Download | Developer (Github)

High-productivity software for complex networks

NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

Documentation
all documentation

Examples
using the library

Reference
all functions and methods

Features

- Python language data structures for graphs, digraphs, and multigraphs.
- Nodes can be "anything" (e.g. text, images, XML records)
- Edges can hold arbitrary data (e.g. weights, time-series)
- Generators for classic graphs, random graphs, and synthetic networks
- Standard graph algorithms
- Network structure and analysis measures
- Open source BSD license
- Well tested: more than 1800 unit tests, >90% code coverage
- Additional benefits from Python: fast prototyping, easy to teach, multi-platform

Versions

Latest Release

1.8.1 - 4 August 2013
downloads | docs | pdf

Development

1.9dev
github | docs | pdf
build passing
coverage 83%

Contact

Mailing list
Issue tracker
Developer guide