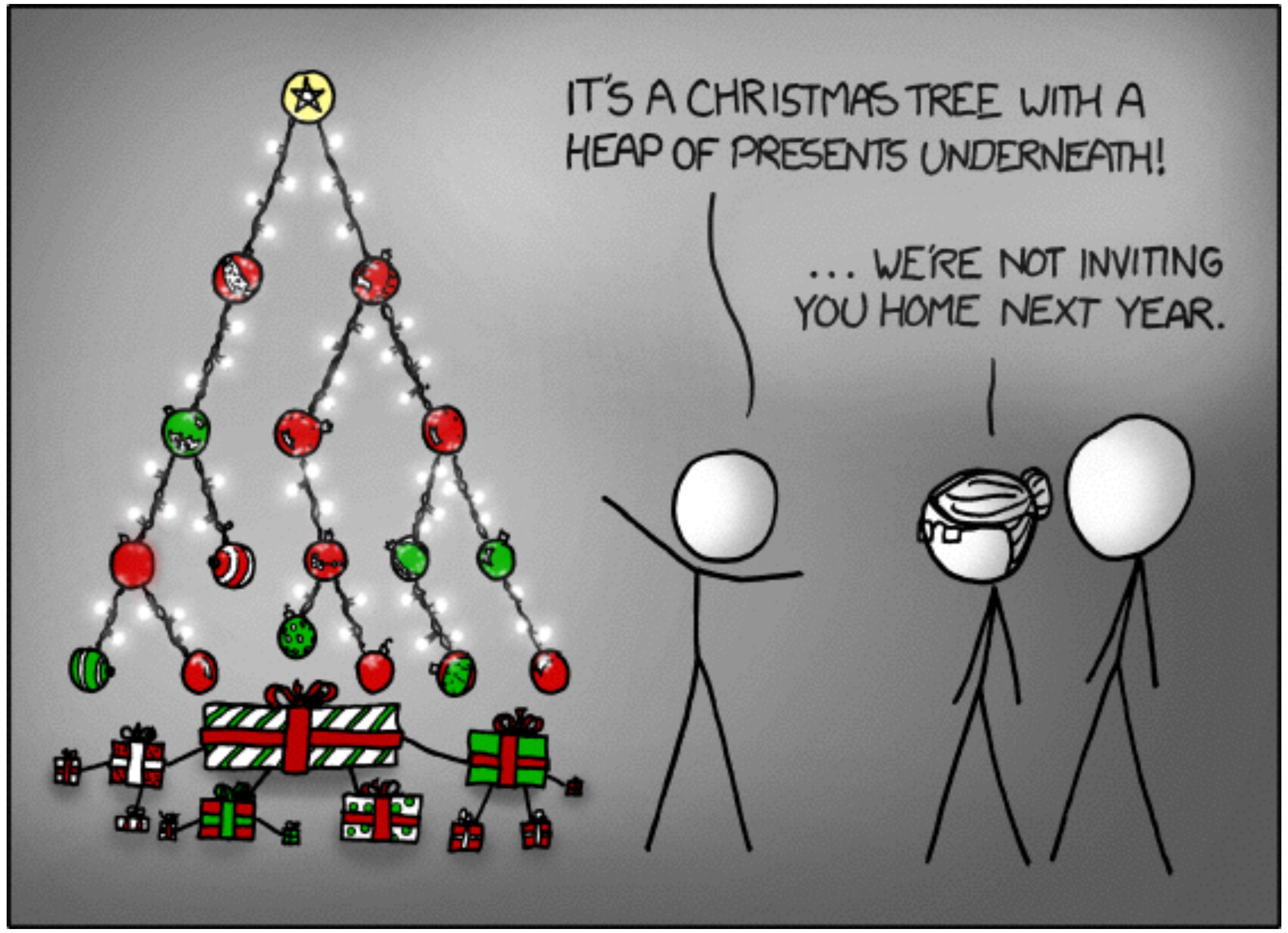
CS-5630 / CS-6630 Visualization
Graphs

Alexander Lex
alex@sci.utah.edu

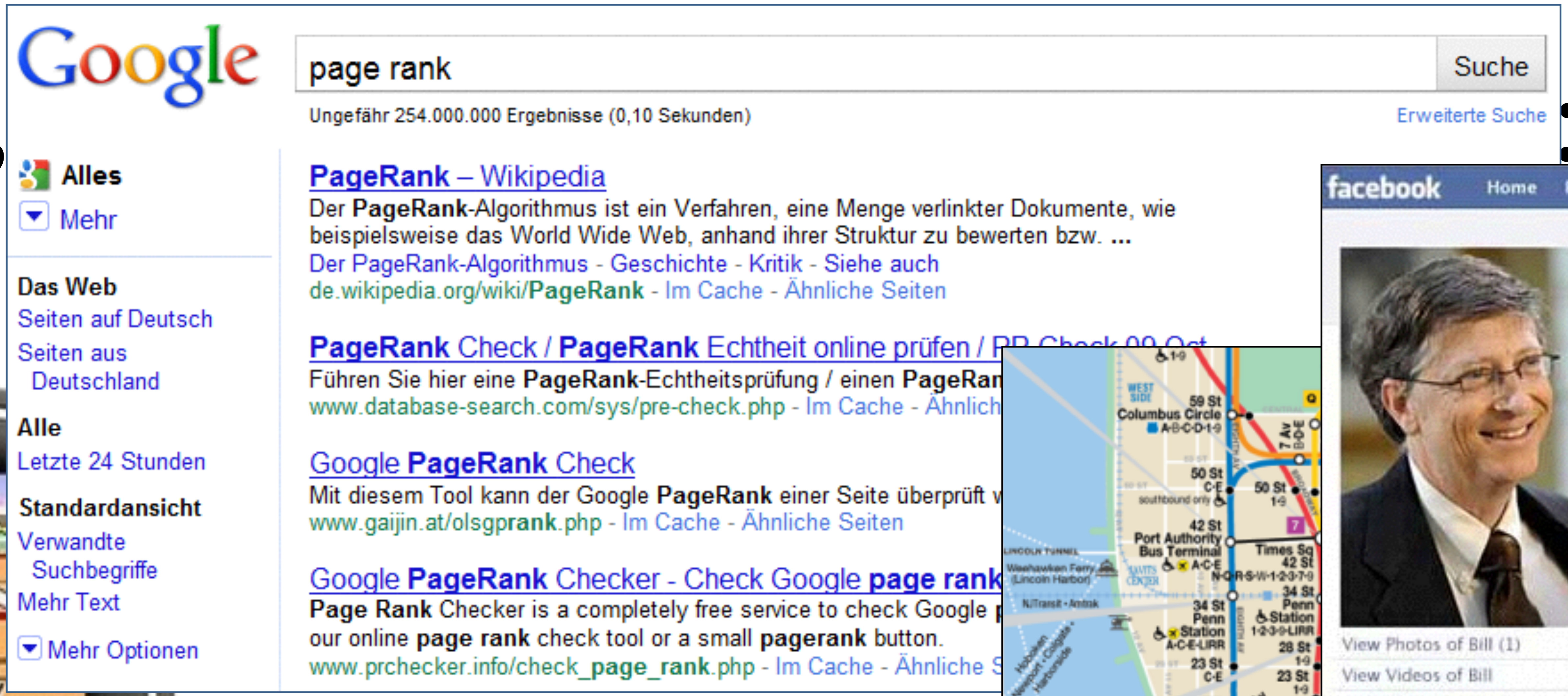
THE UNIVERSITY OF UTAH

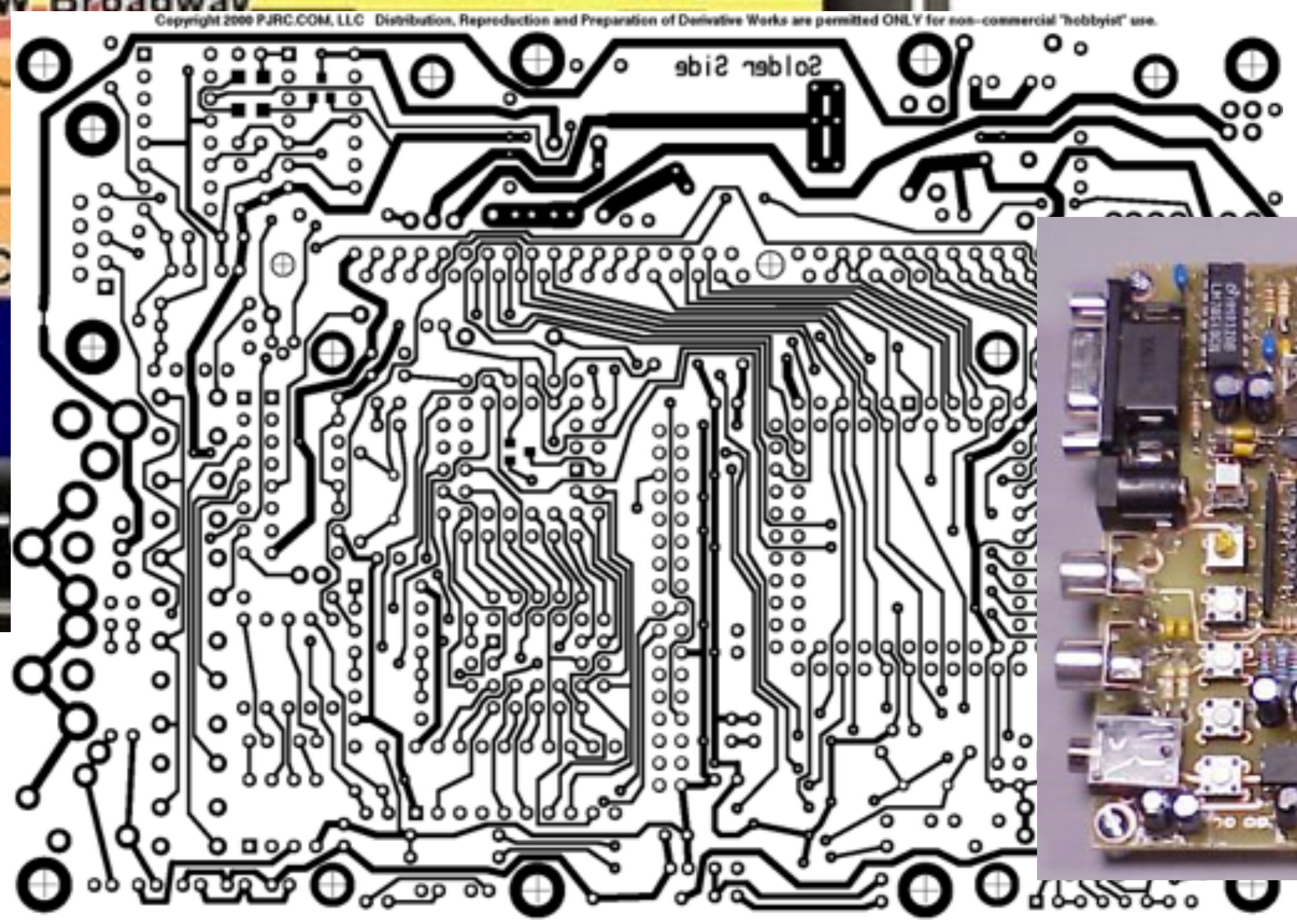


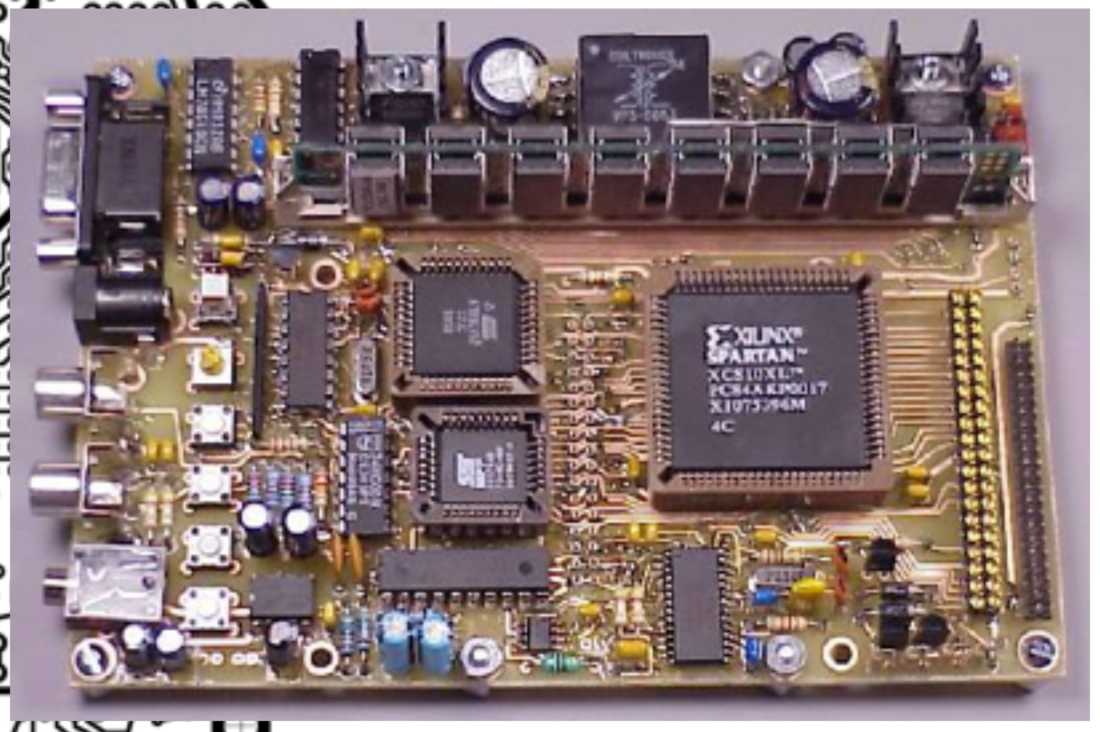
[xkcd]

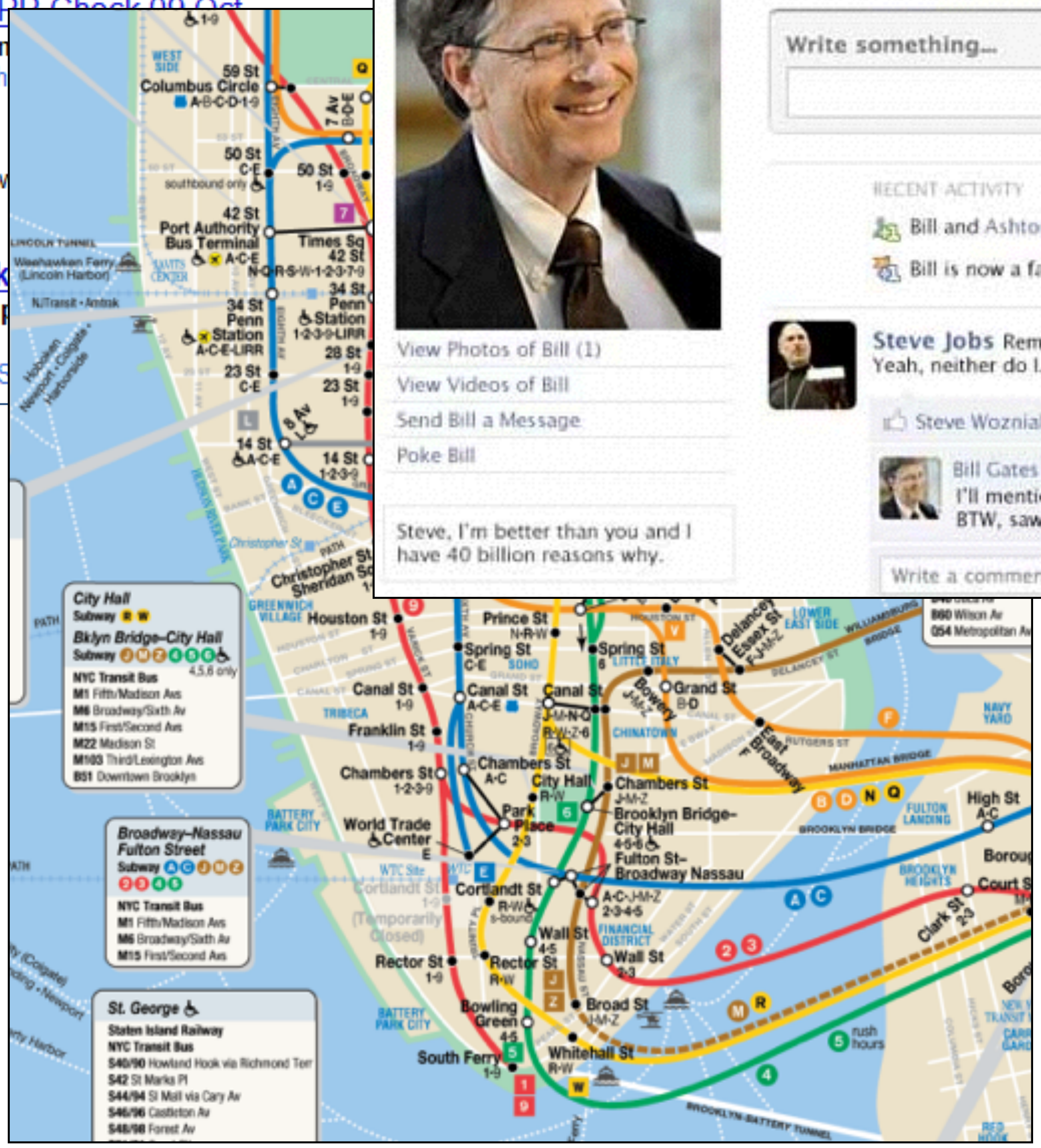
Applications of Graphs

Without graphs, :





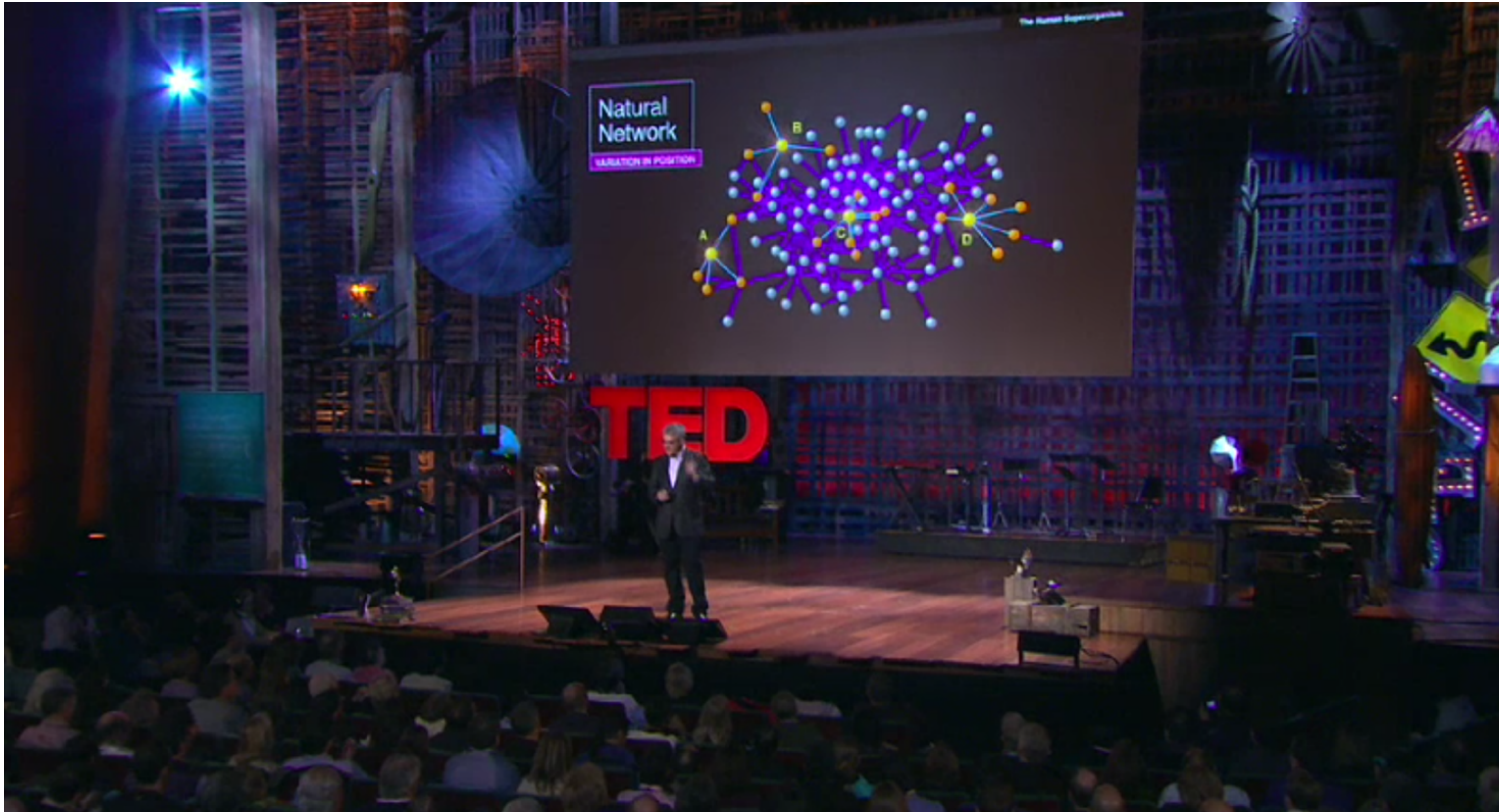





CITRATE
ATP
MALONATE, OXALO-ACETATE
FUMARATE
SUCCINATE DEHYDROGENASE
SUCCINATE
CoA-SH
SUCCINYL-CoA HYDROLASE
PPi, ATP
H_2O
PROTEINS
ACETYL-CoA
GLYCYL-tRNA
GLYCINE tRNA LIGASE
GLYCINE
5-AMINOLEVULINATE SYNTHASE
increased in acute intermittent porph
PyrP
ADP
Pi
SUCCINYL-CoA
GTP (ITP)
GDP (IDP)
SUCCINYL-CoA-LIGASES
$FADH_2 \cdot E$
FAD · E
ACCEPTOR (UBIQUINONE)
$ACC \cdot H_2$ (UBIQUINOL)
CO_2
NADH + H+
FORMATE DEHYDROGENASE
FORMATE DEHYDROGENASE (CYTOCHROME)
NAD+
MOLYBDOPTERIN
FORMYL-CoA
FORMYL-CoA HYDROLASE
FORMATE
OXALATE DECARBOXYLASE
OXALYL-CoA DECARBOXYLASE
OXALATE-CoA-TRANSFERASE
FERREDOXIN (red)
FERREDOXIN (ox.)
OXO-GLUTARATE SYNTHASE
HEME
hv (in plants and photobacteria)
DIHYDROLIPOAMIDE SUCCINYL TRANSFERASE
DIHYDROLIPOAMIDE DEHYDROGENASE
Deficient in hyperoxaluria I
PYRUVATE
ALANINE-GLYOXYLATE TRANSAMINASE
NADH
ThPP · E
Michal 2000

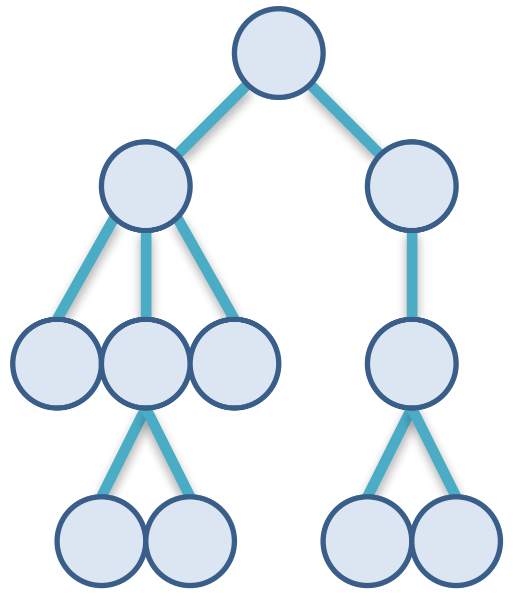

facebook
December 2010

Graph Visualization Case Study

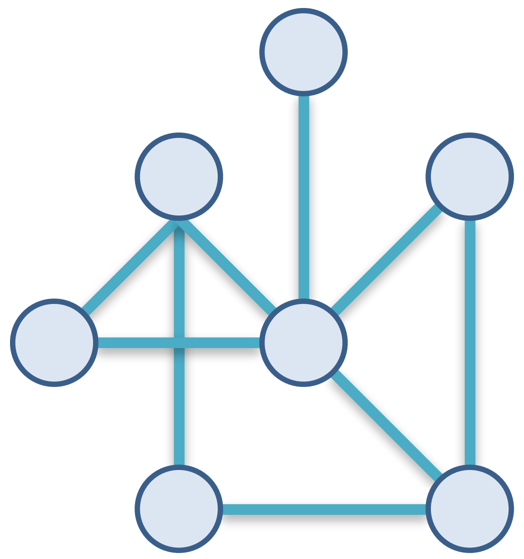


Graph Theory Fundamentals

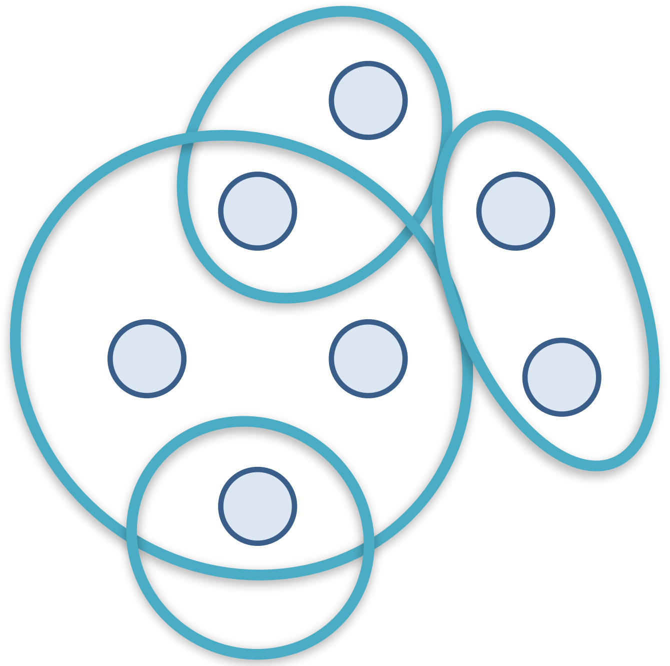
Tree



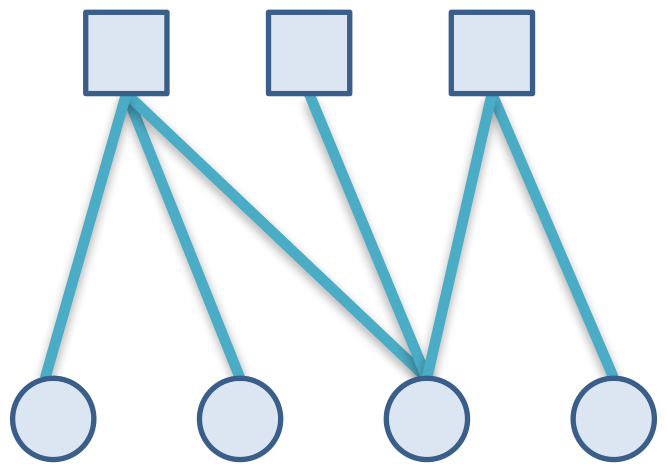
Network



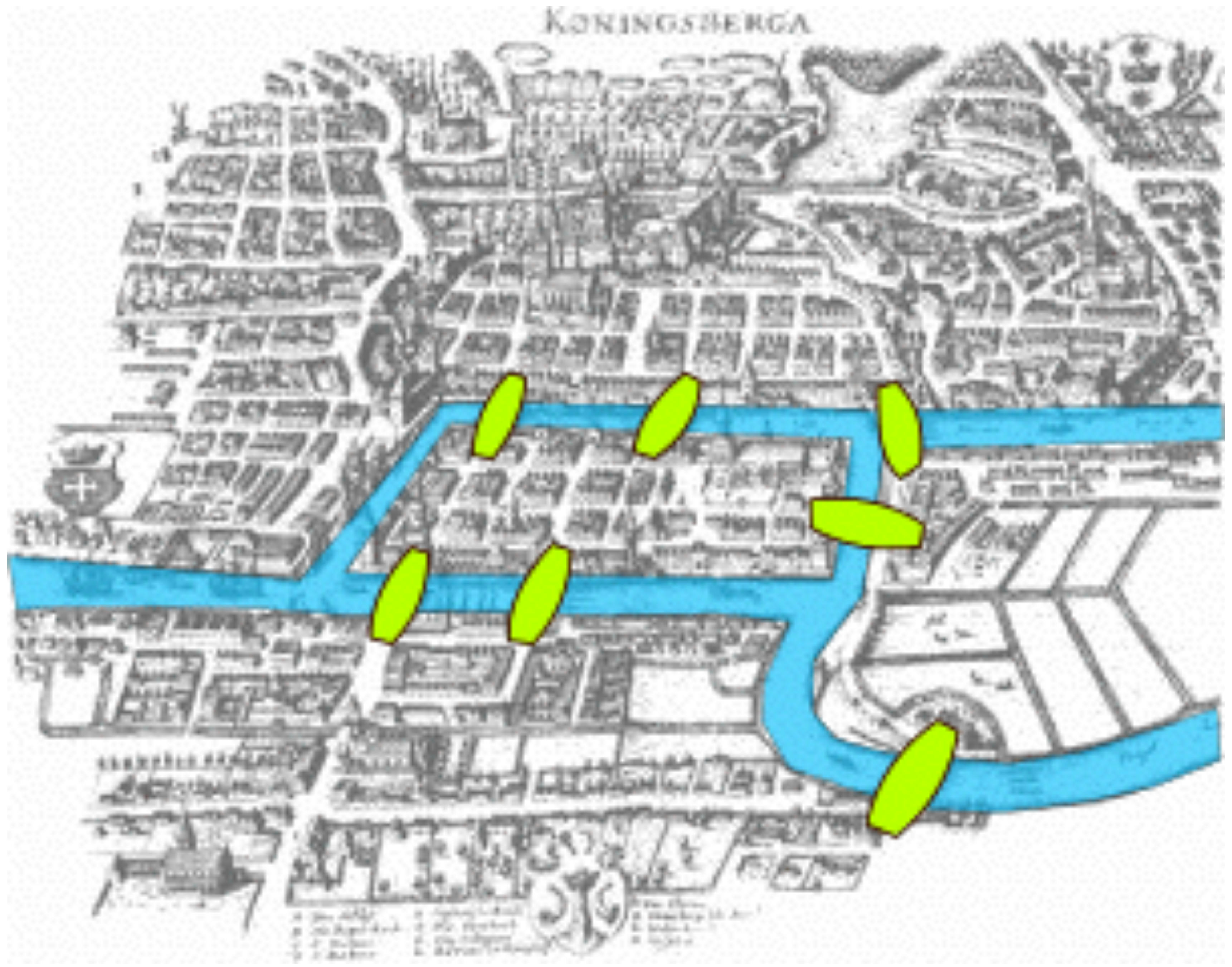
Hypergraph

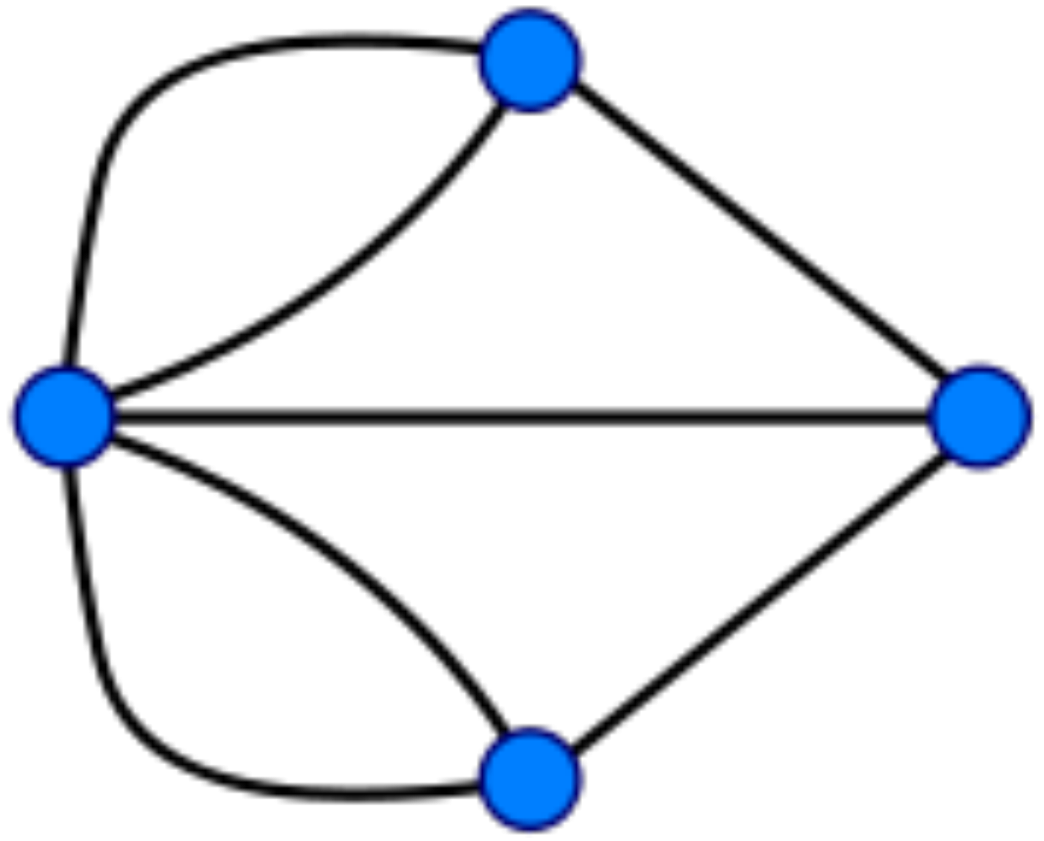


Bipartite Graph



Königsberg Bridge Problem (1736)



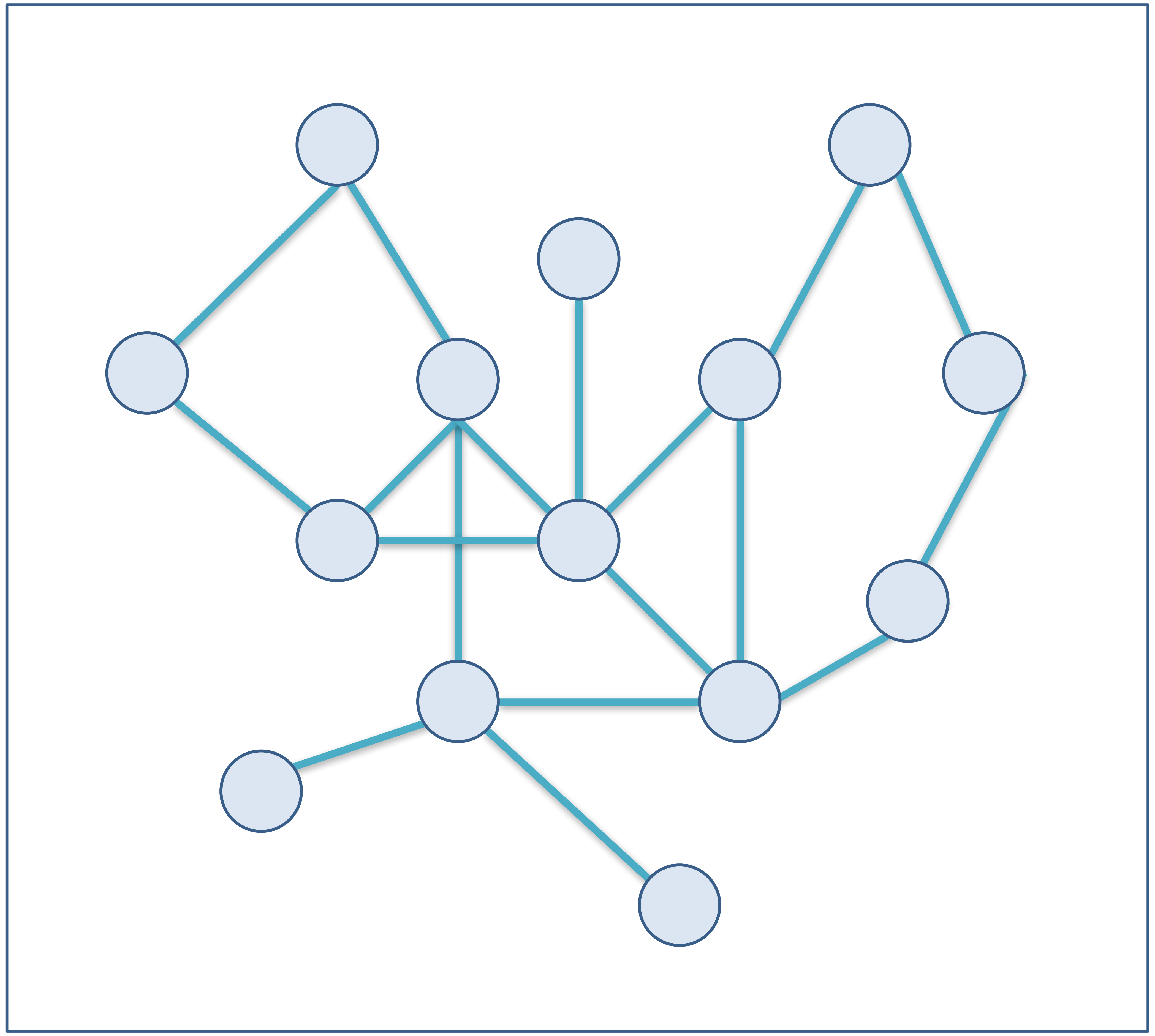


Want to make $1 million? Find an O(n^k) algorithm to find Hamiltonian Paths (path that visits each vertex exactly once) - example of P vs. NP problem.

Graph Terms (1)

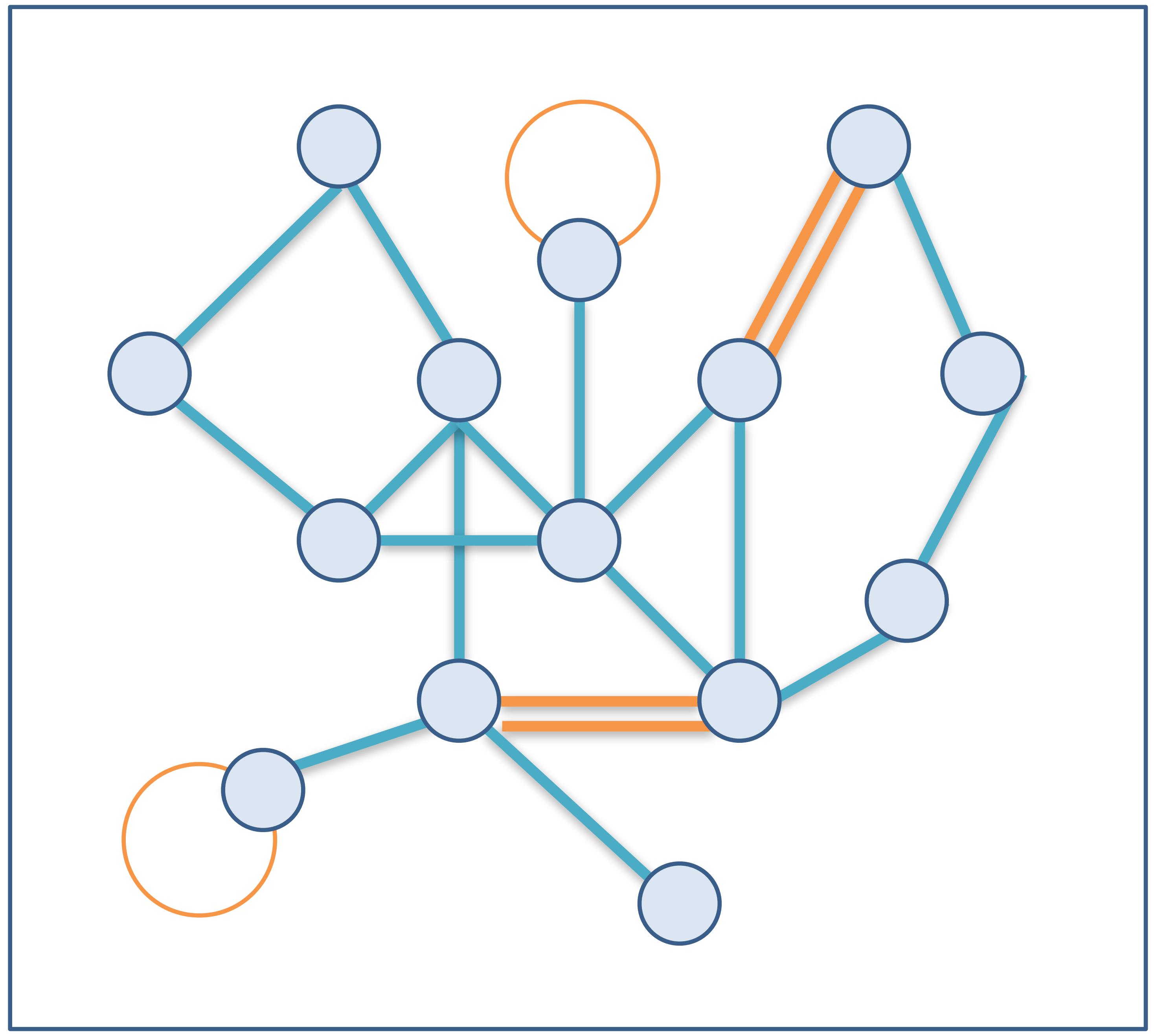
A graph **G(V,E)** consists of a set of **vertices V** (also called nodes) and a

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



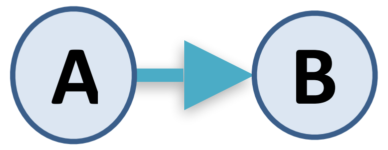
Graph Terms (2)

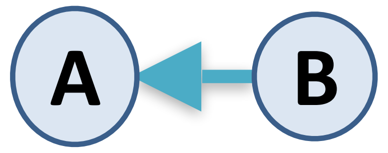
A simple graph G(V,E) is a graph which contains **no multi-edges** and **no loops**



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

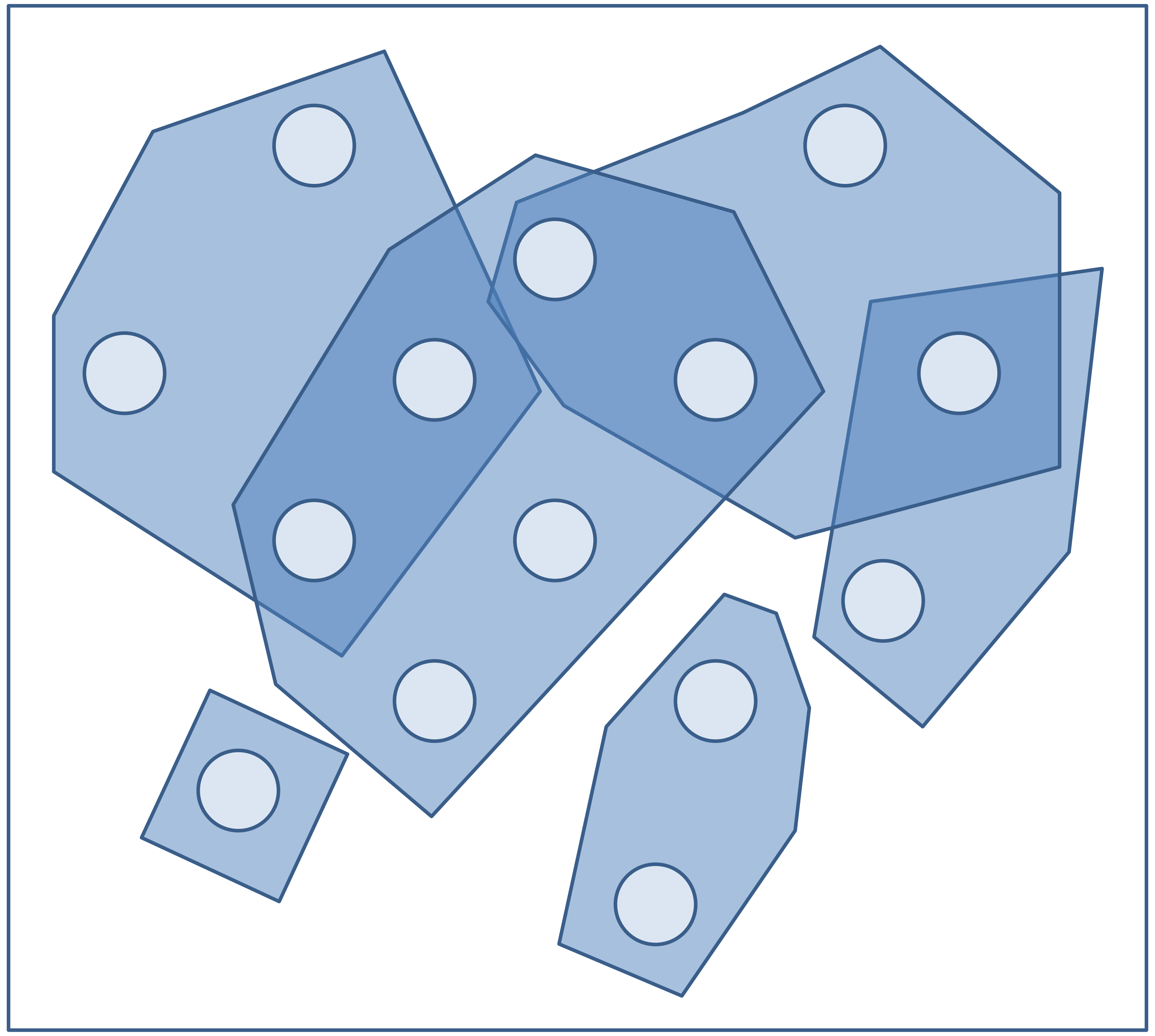
Graph Terms (3)

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.

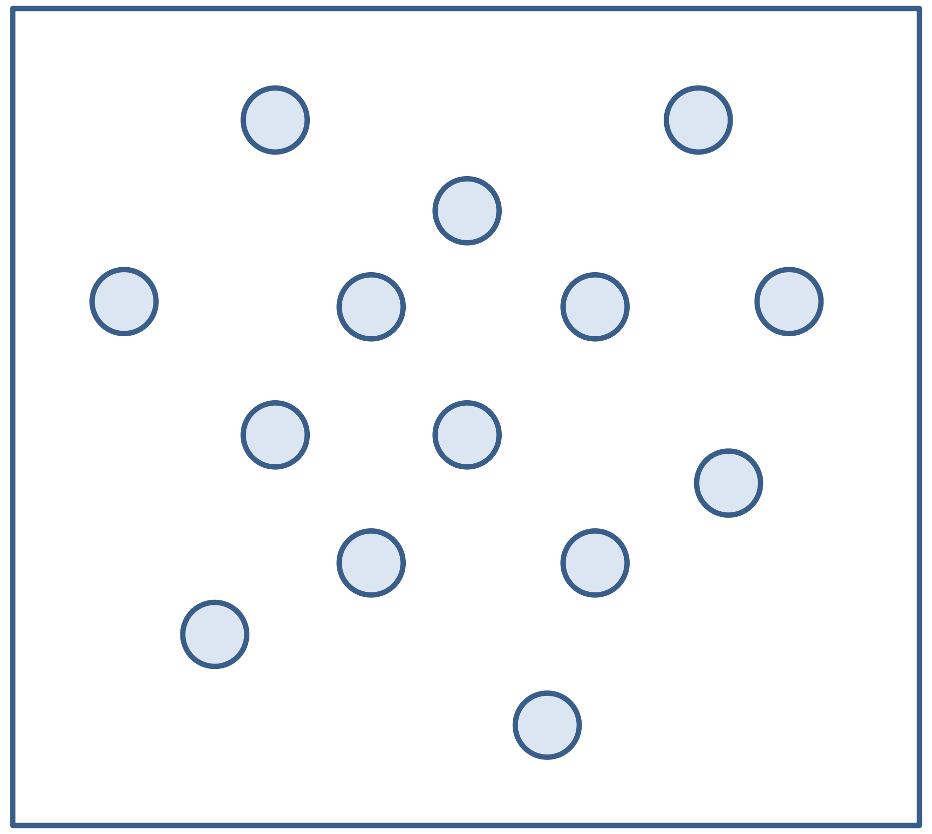


Hypergraph Example

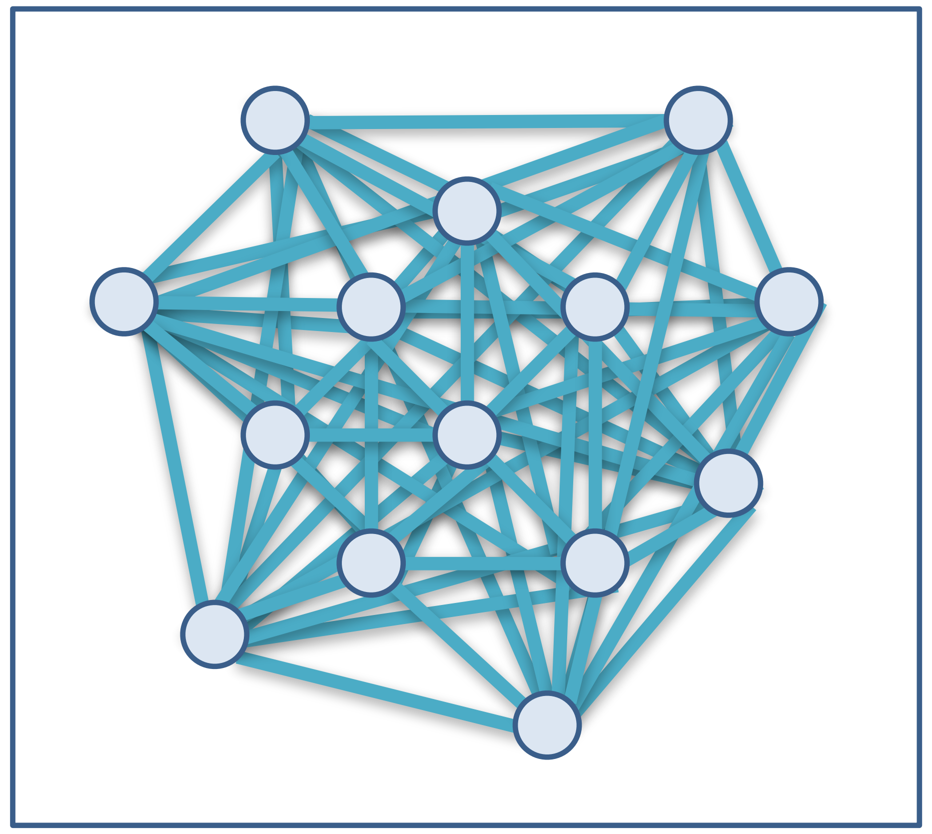
Graph Terms (4)

Independent Set
G contains no edges

Clique
G contains all possible edges



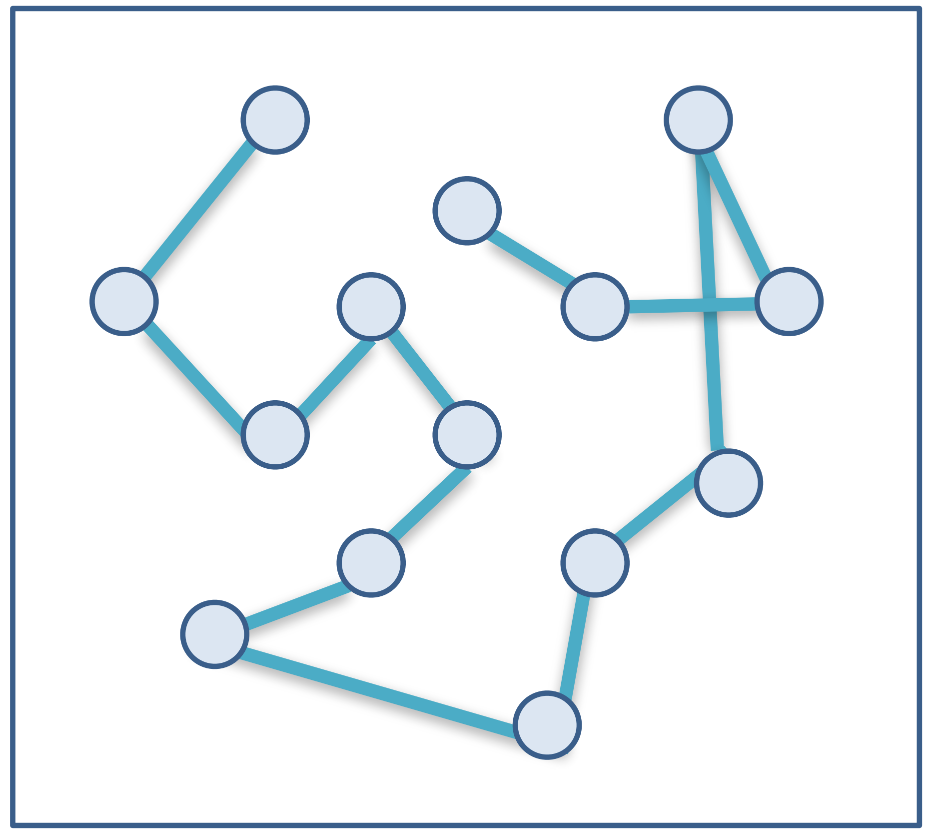
Independent Set



Clique

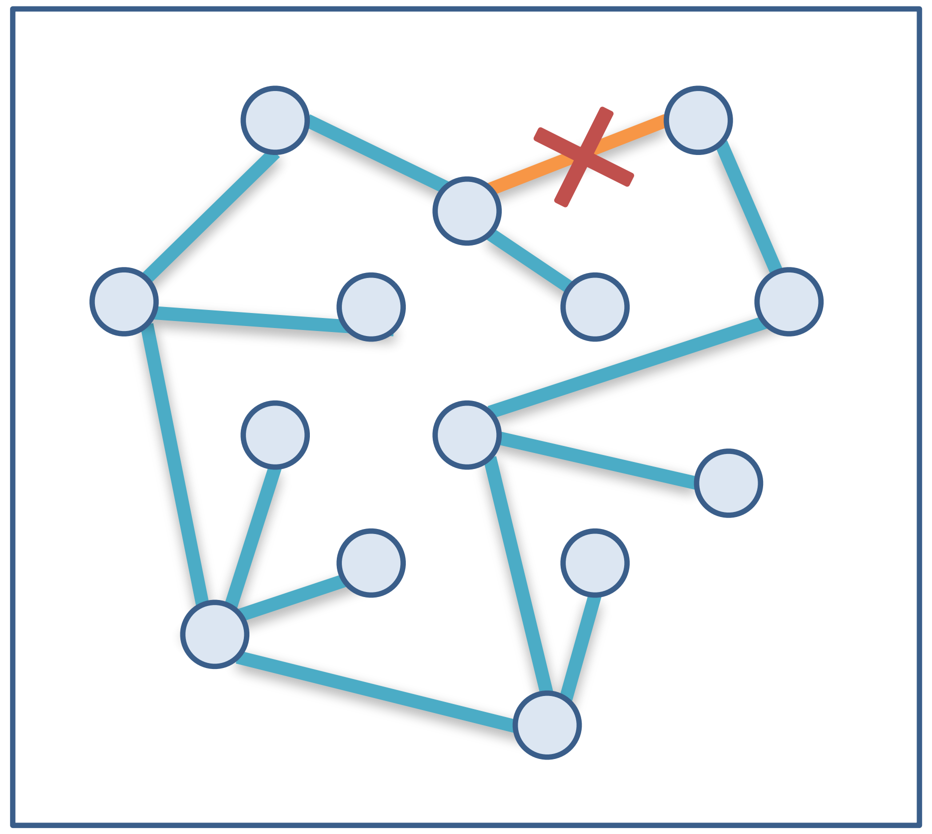
Graph Terms (5)

Path
G contains only edges that can be consecutively traversed



Path

Tree
G contains no cycles

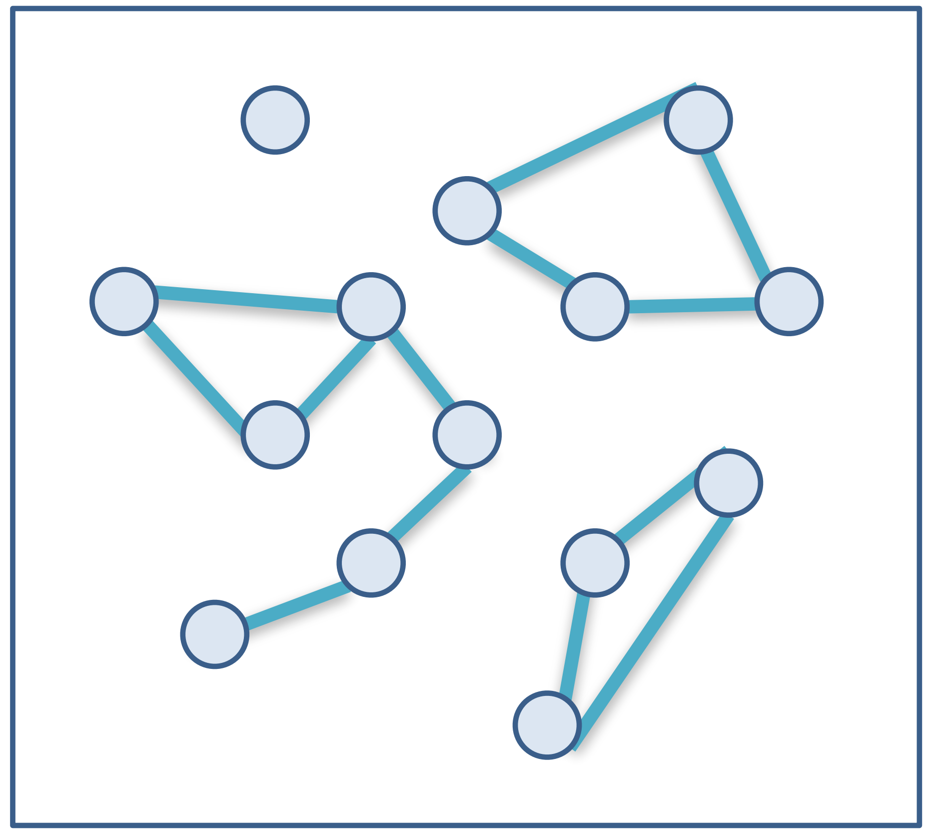


Tree

Network
G contains cycles

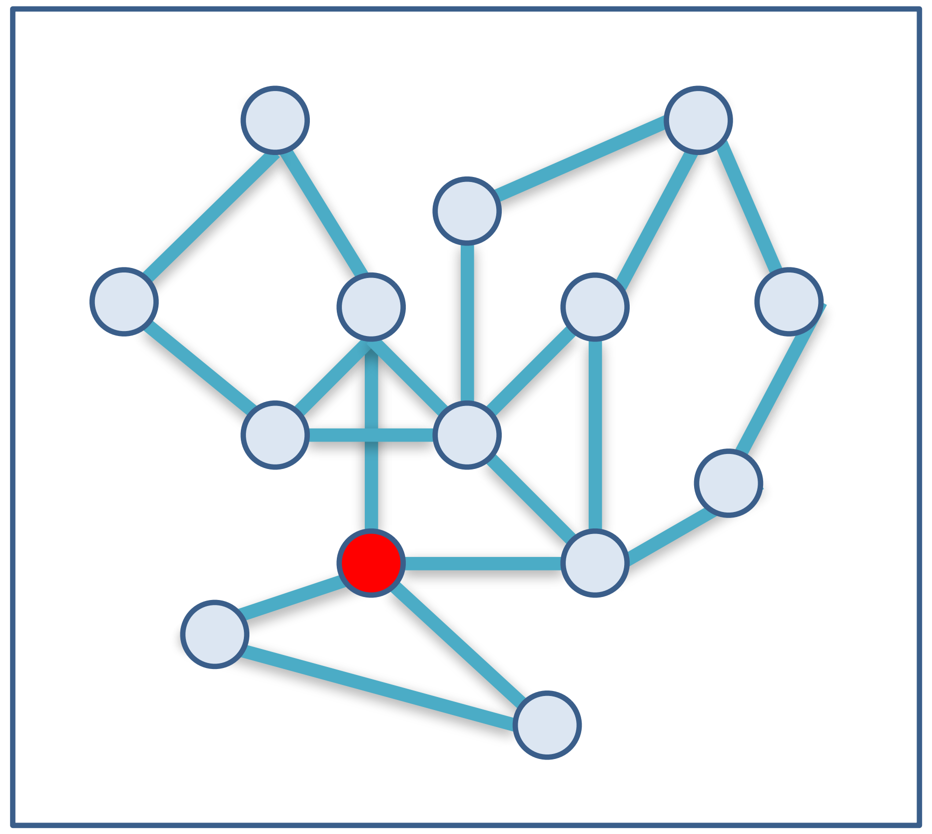
Graph Terms (6)

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



Unconnected Graph

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

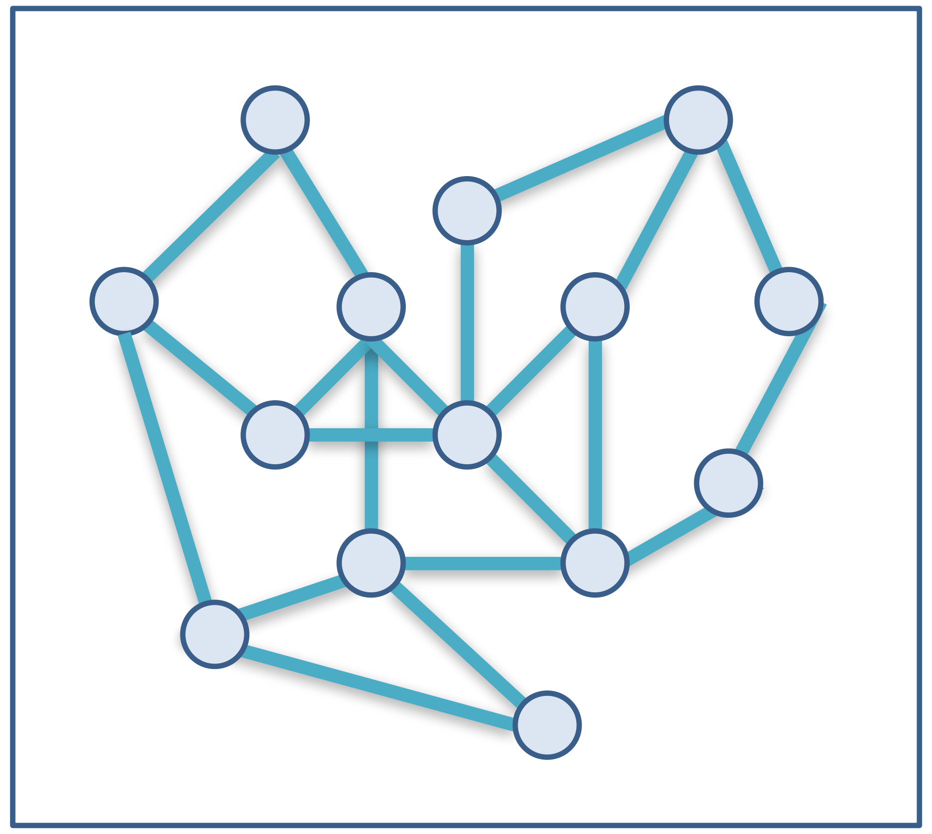


Articulation Point (red)

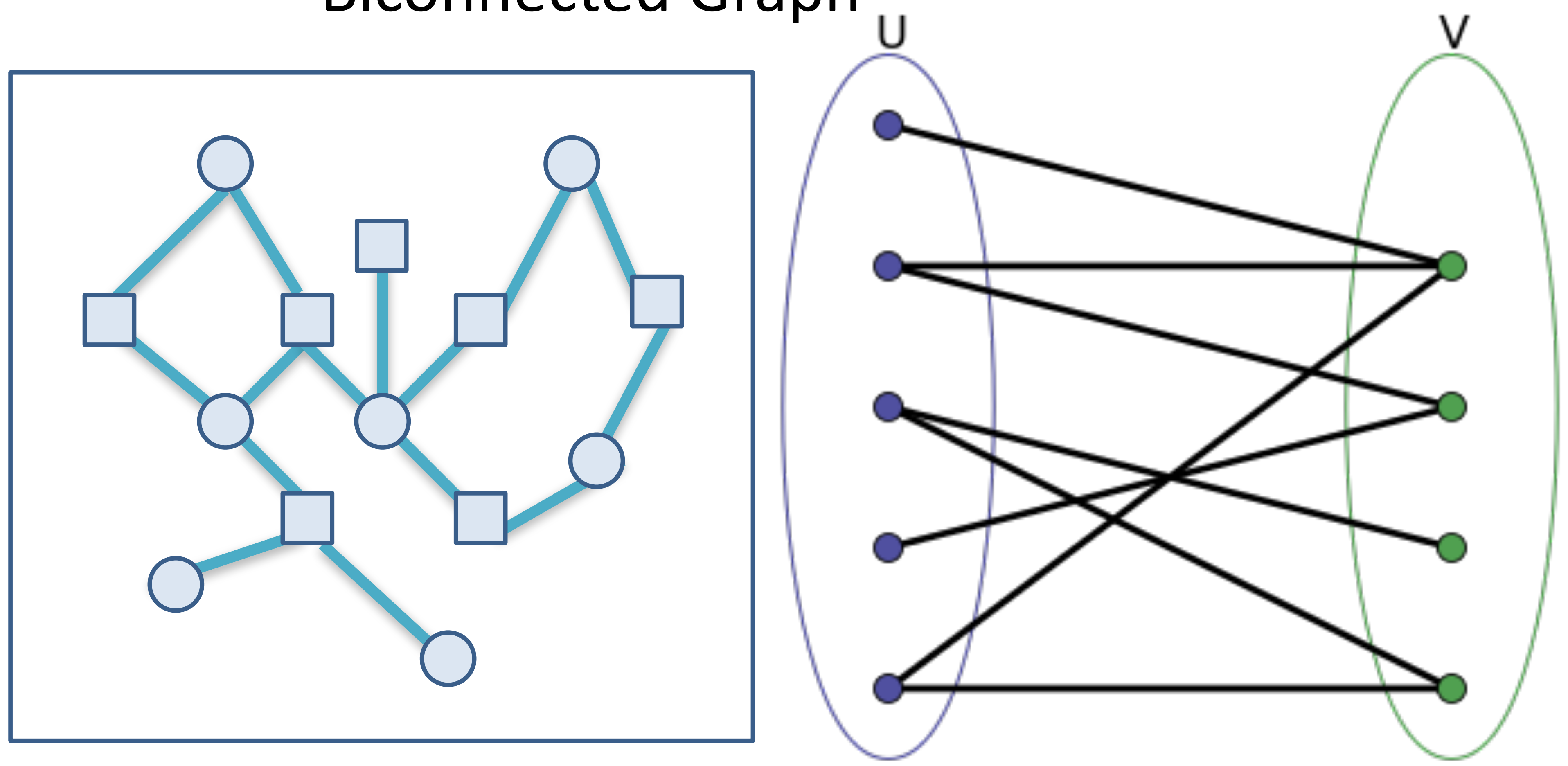
Graph Terms (7)

Biconnected graph
A graph without articulation points.

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



Biconnected Graph



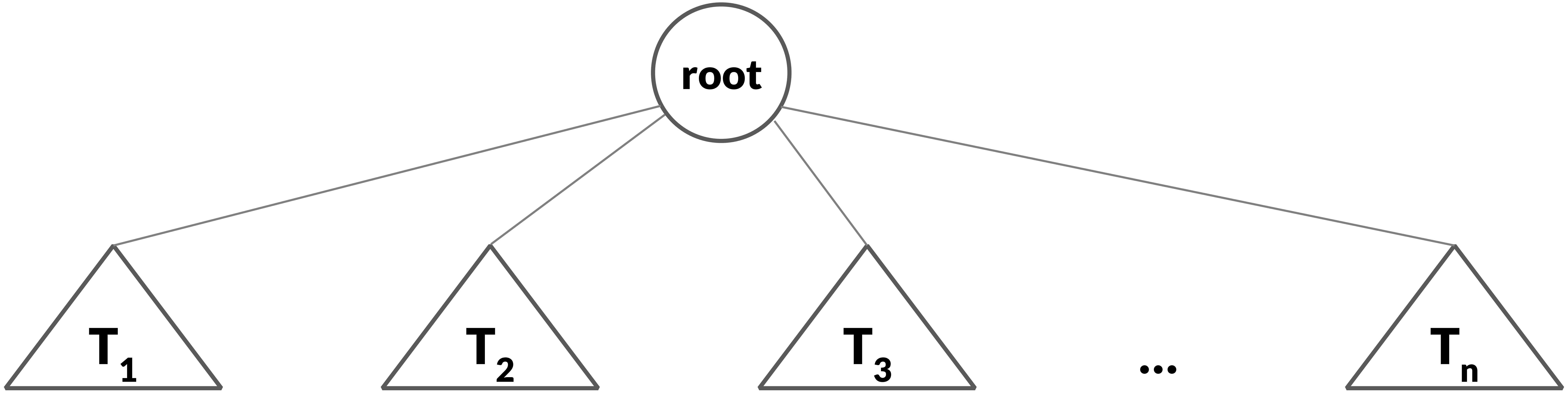
Bipartite Graph

Tree

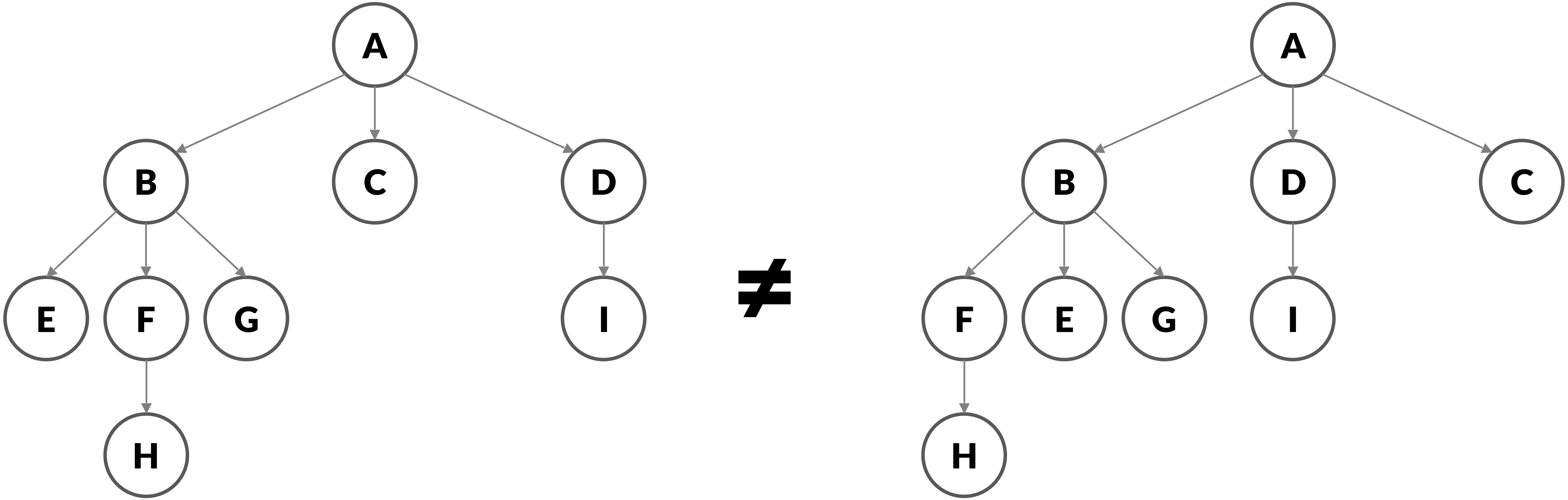
A graph with no cycles - or:

A collection of nodes

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



Ordered Tree

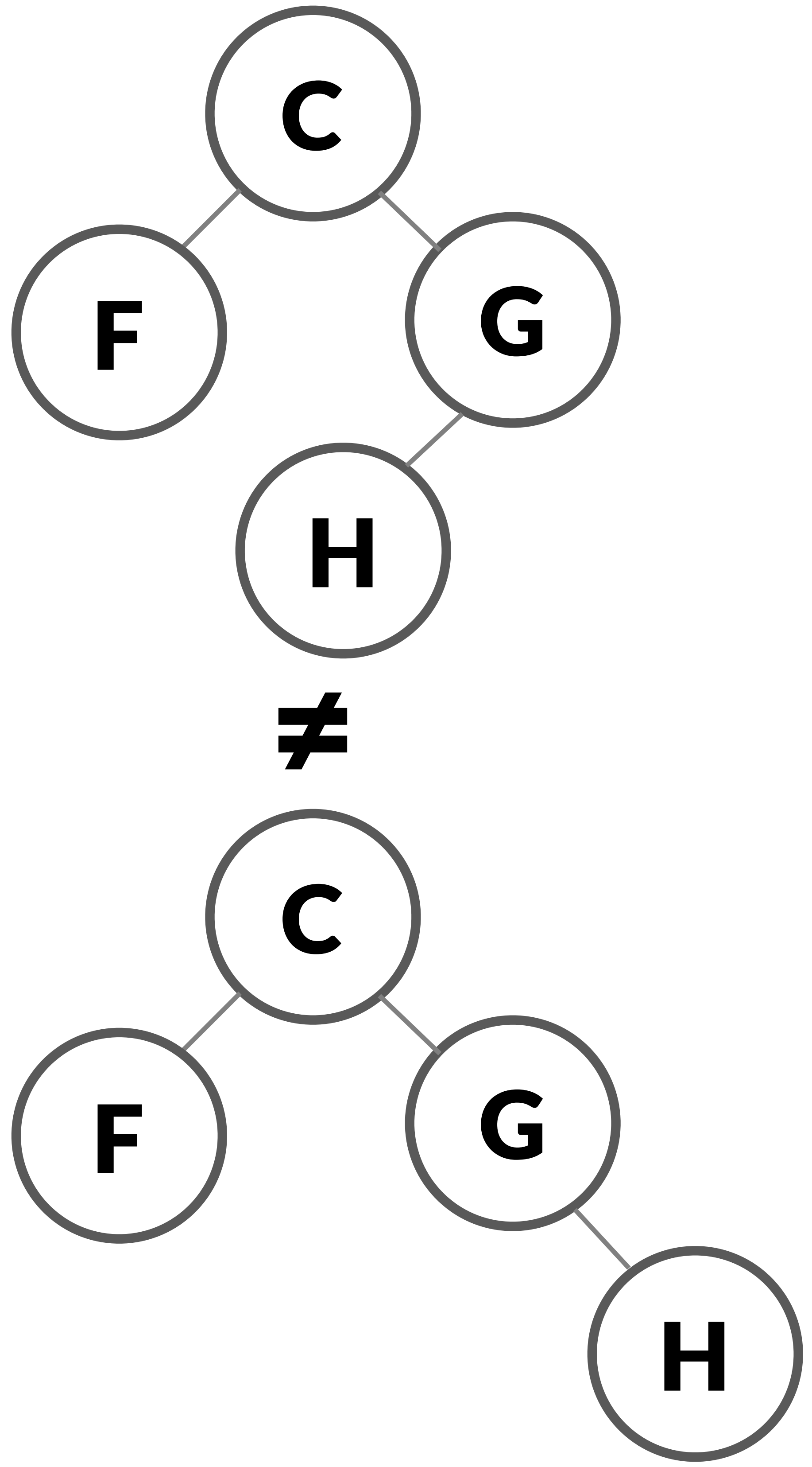


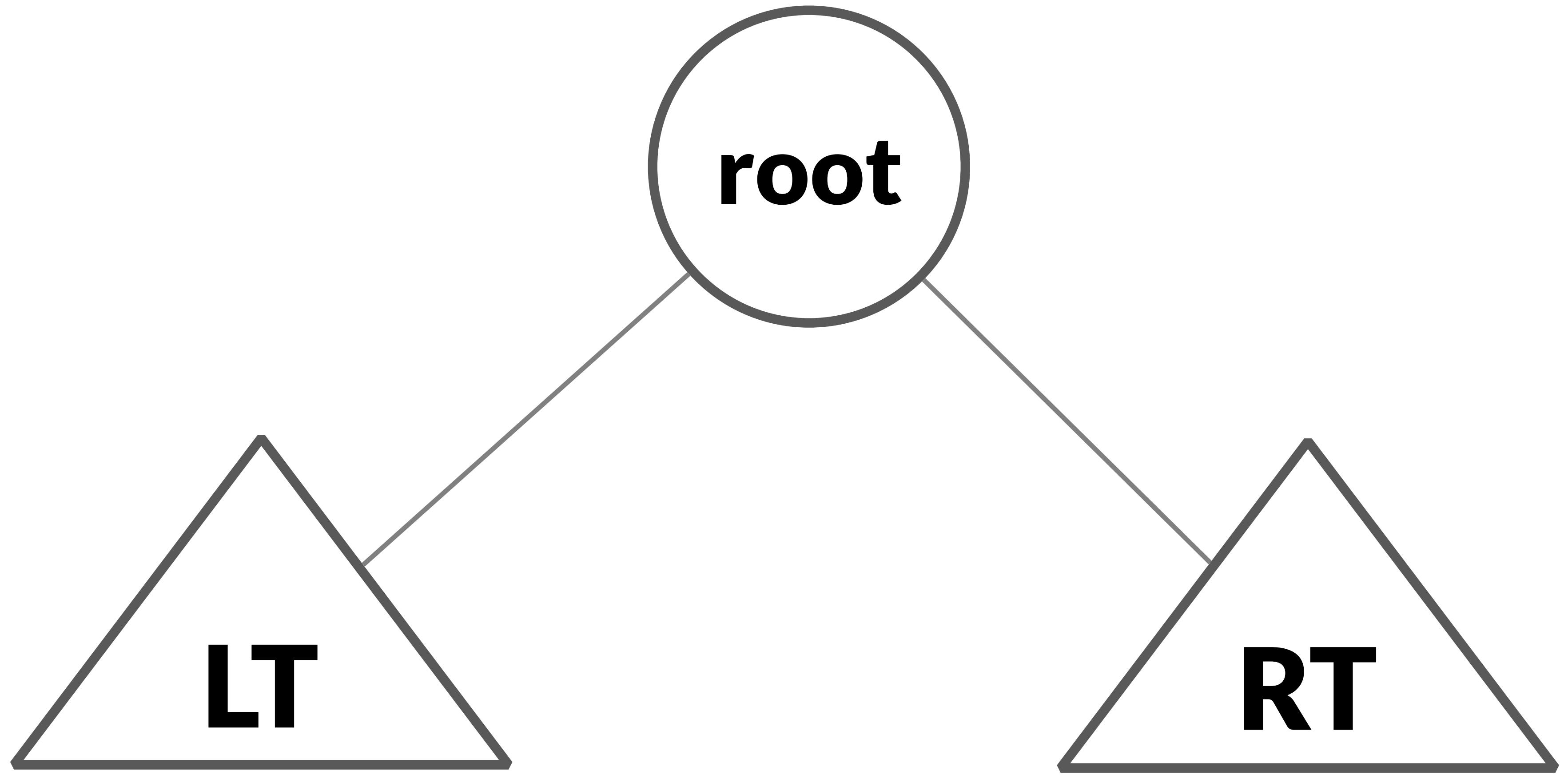
Binary Trees

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

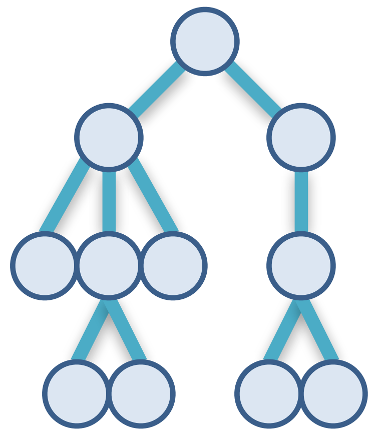




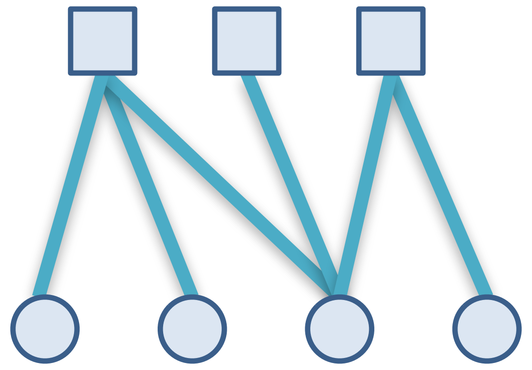
Different Kinds of Graphs

Over 1000 different graph classes

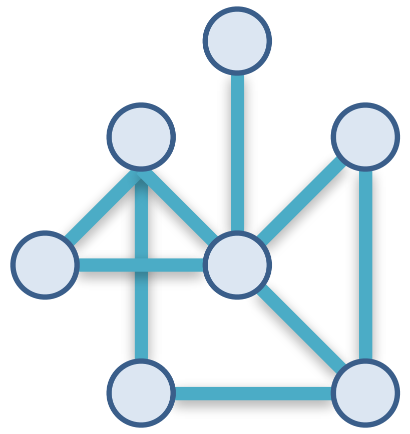
Tree



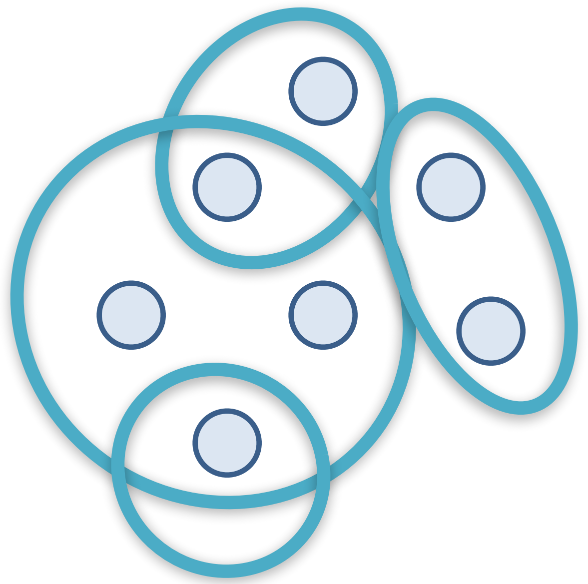
Bipartite Graph

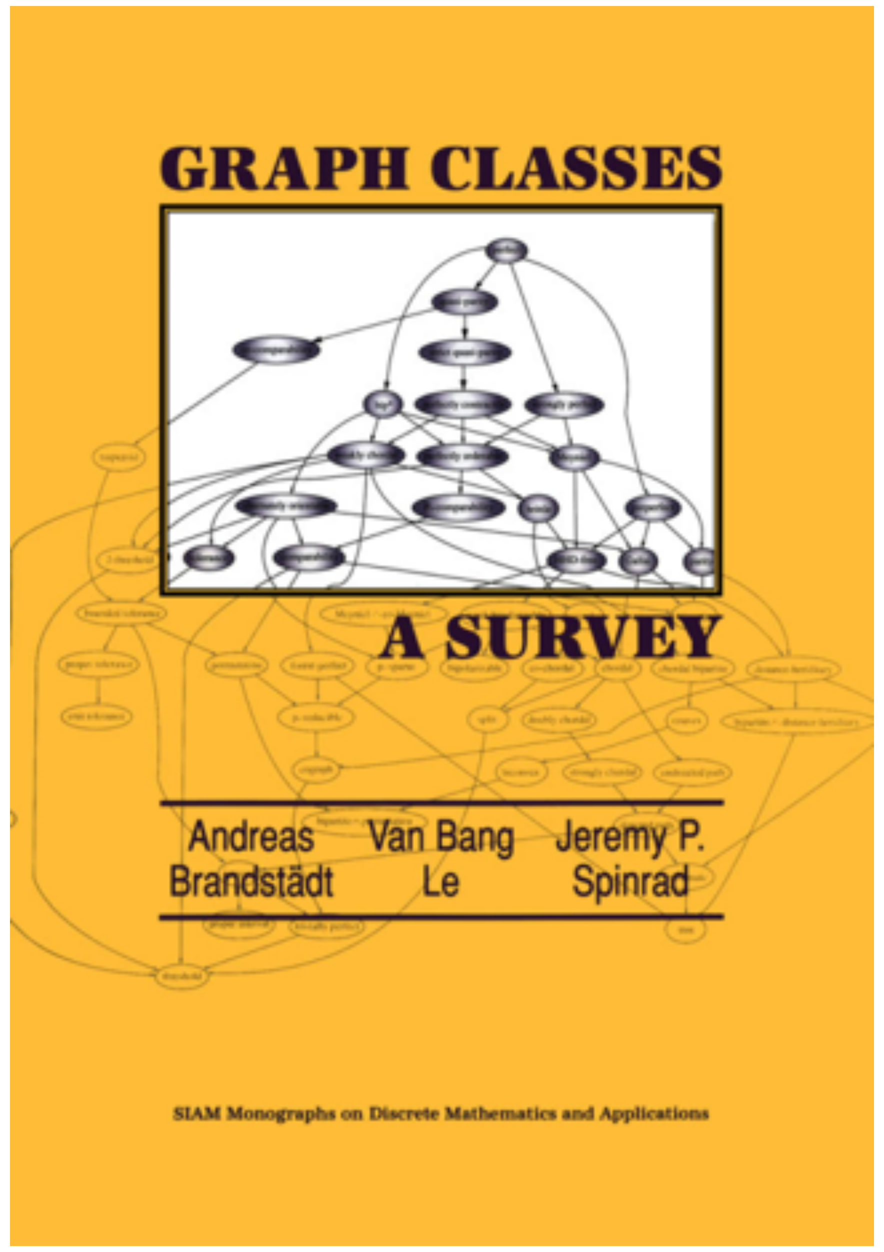


Network



Hypergraph





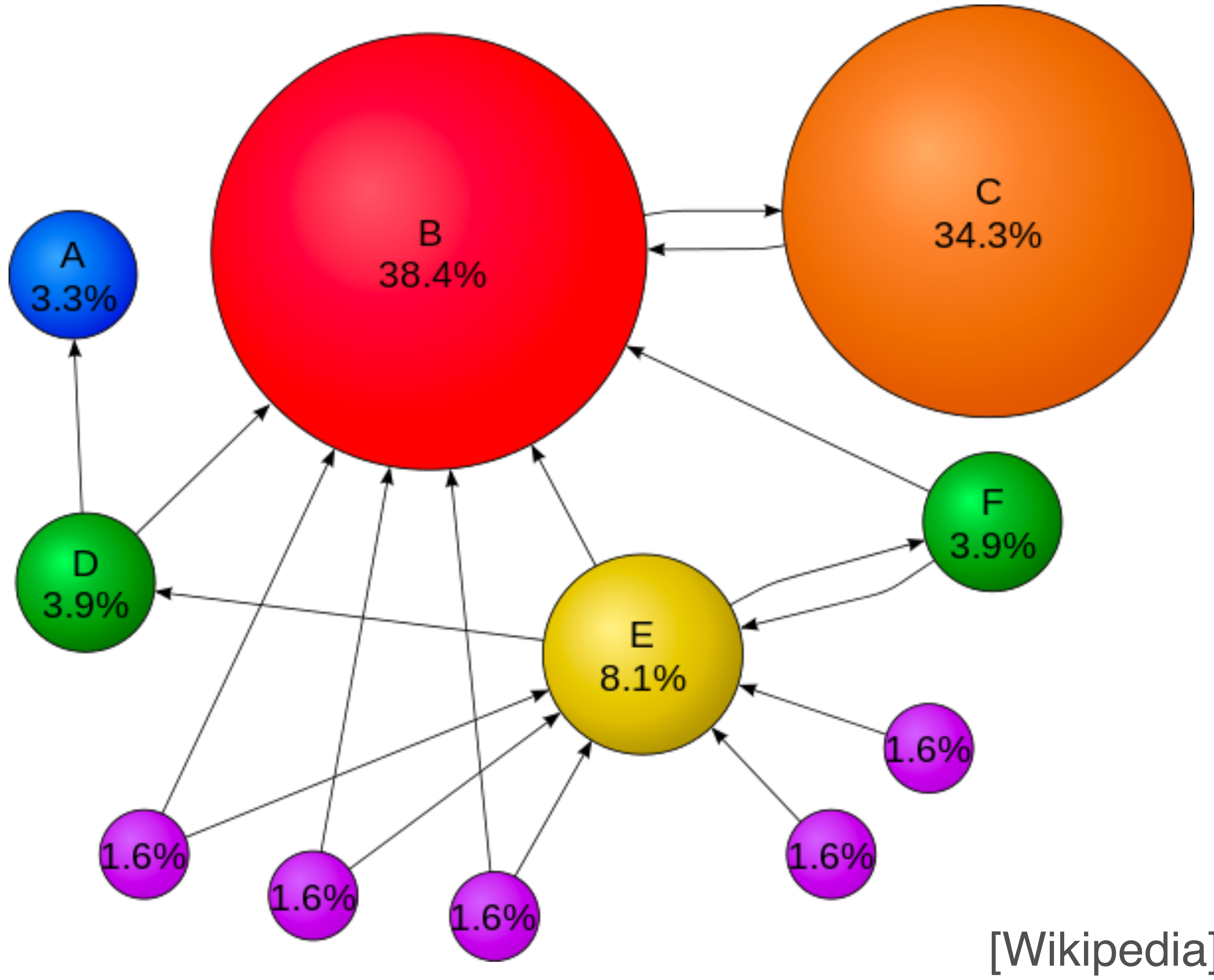
A. Brandstädt et al. 1999

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.

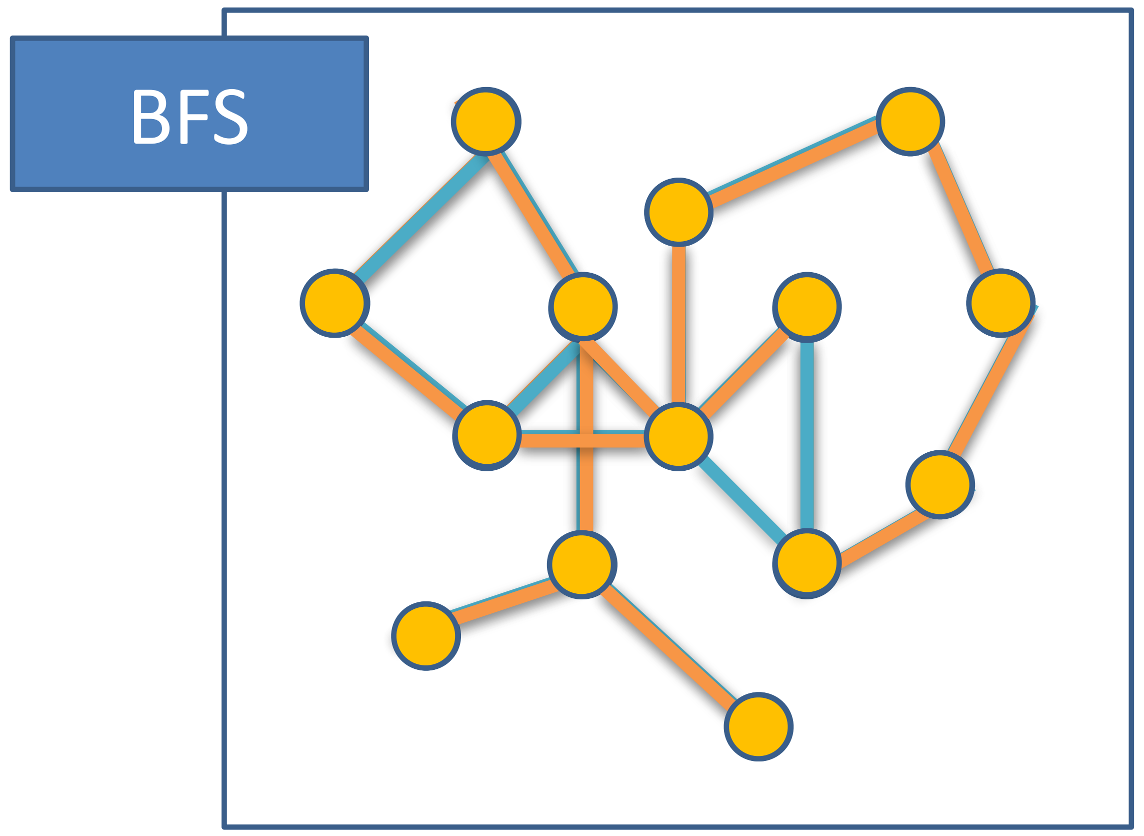
Pagerank
count number & quality of links



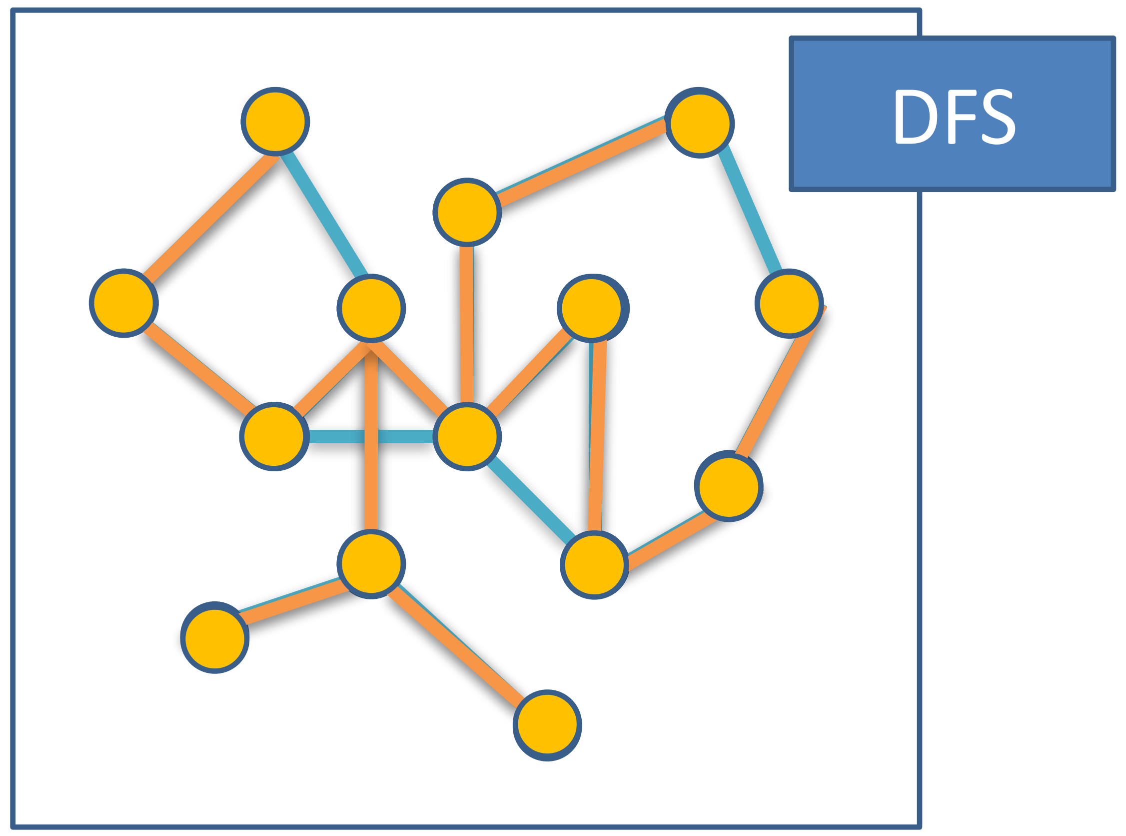
[Wikipedia]

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)

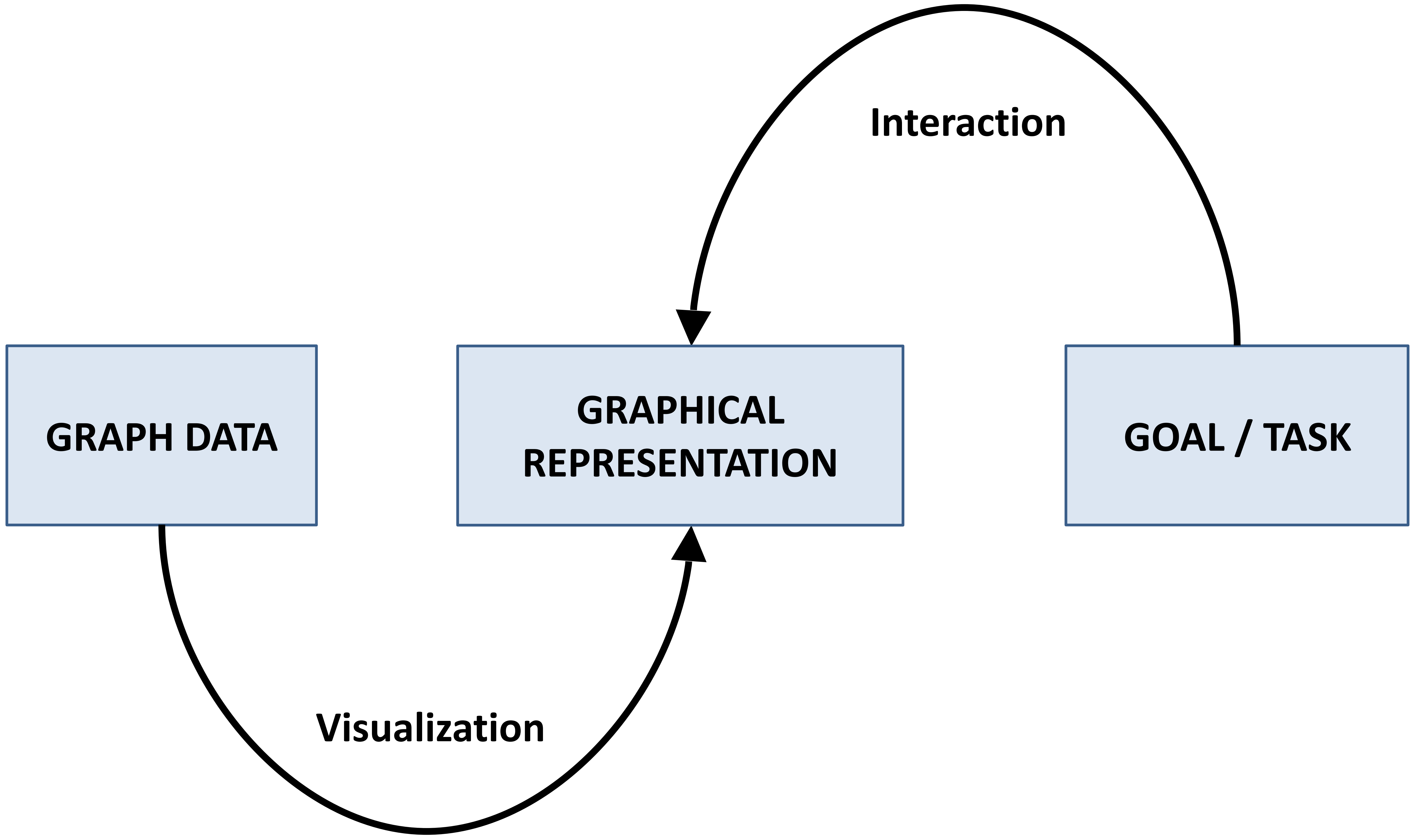
Hamiltonian path/cycle (path that visits all vertexes once)

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

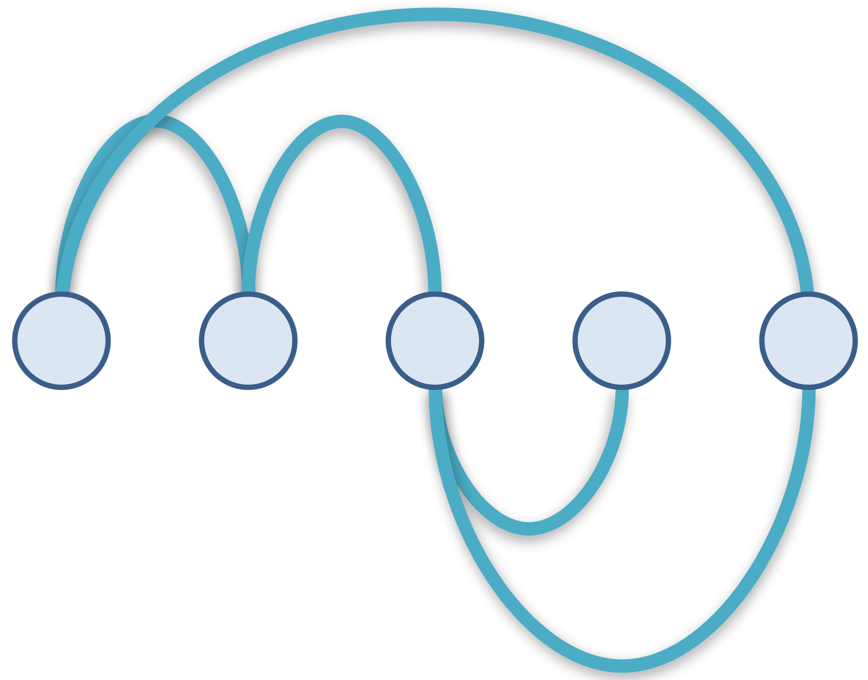
- ABT: Give the number of all nodes.
- TBT: Give the indegree (the number of incoming edges) of a node.

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

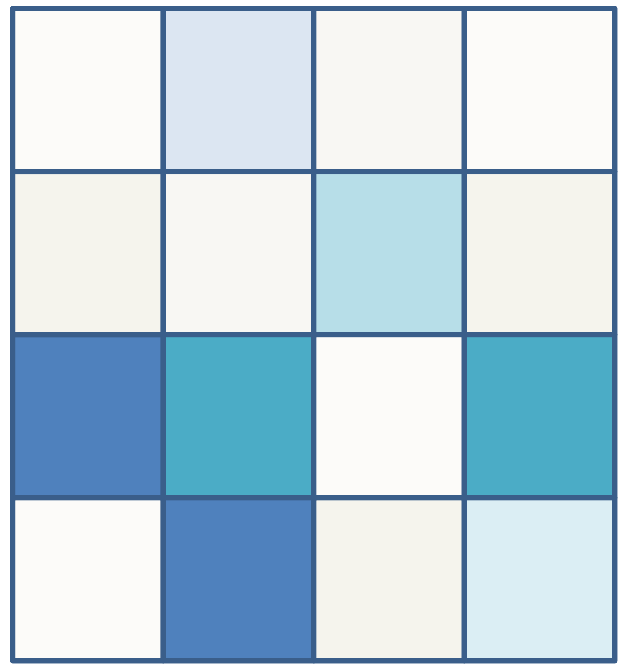
- ABT: Sort all edges according to their weight.
- TBT: Traverse the graph starting from a given node.

list adapted from Schulz 2010

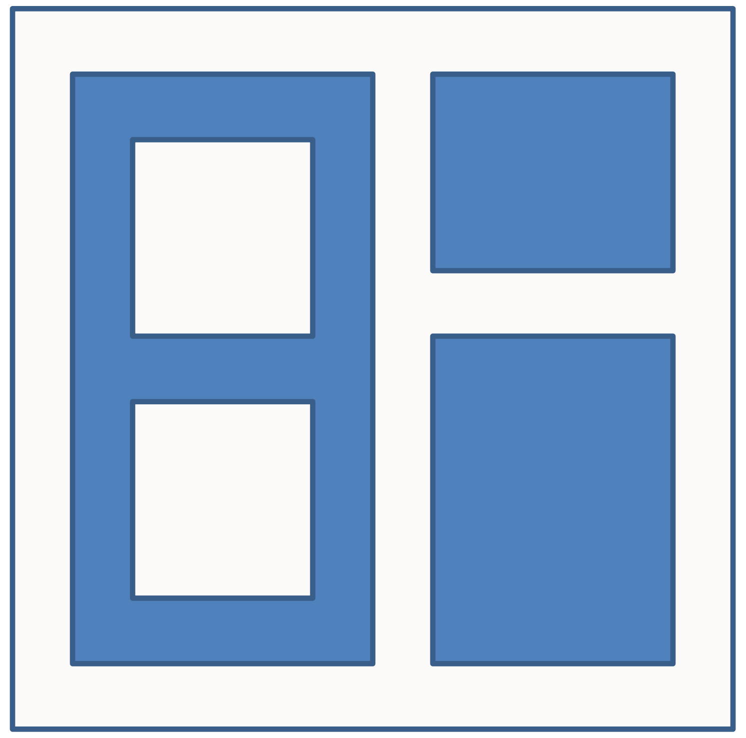
Three Types of Graph Representations



Explicit
(Node-Link)



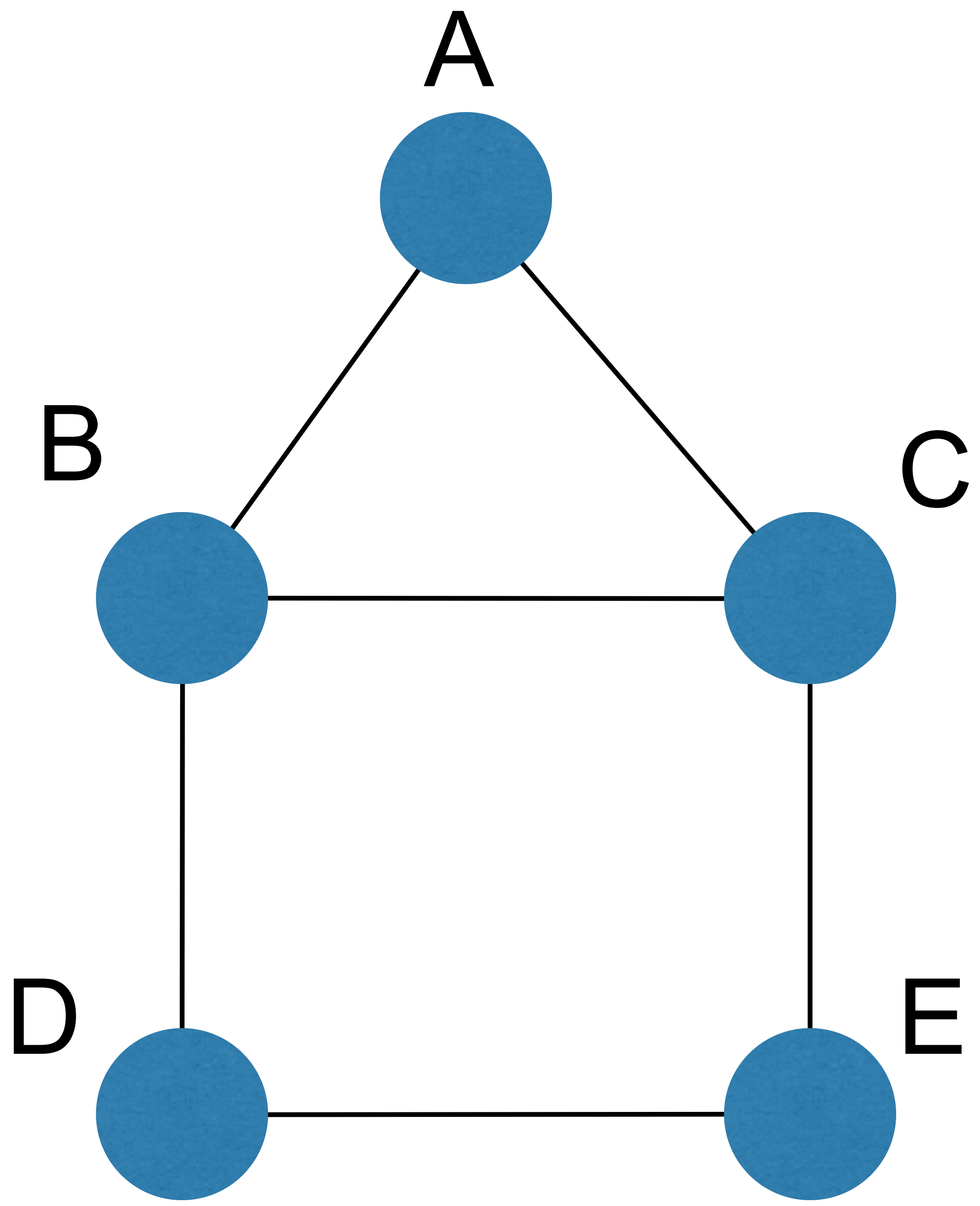
Matrix

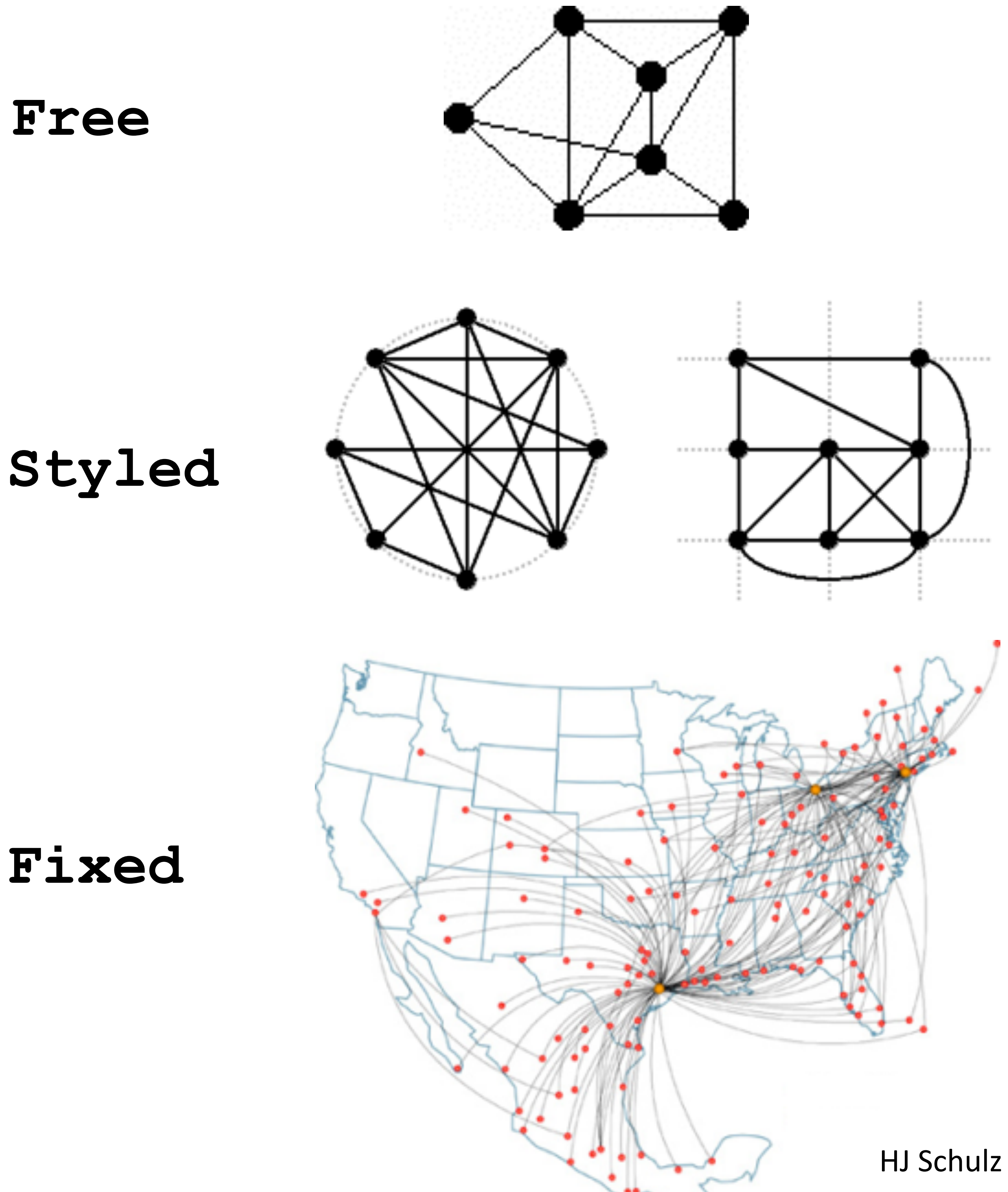


Implicit

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

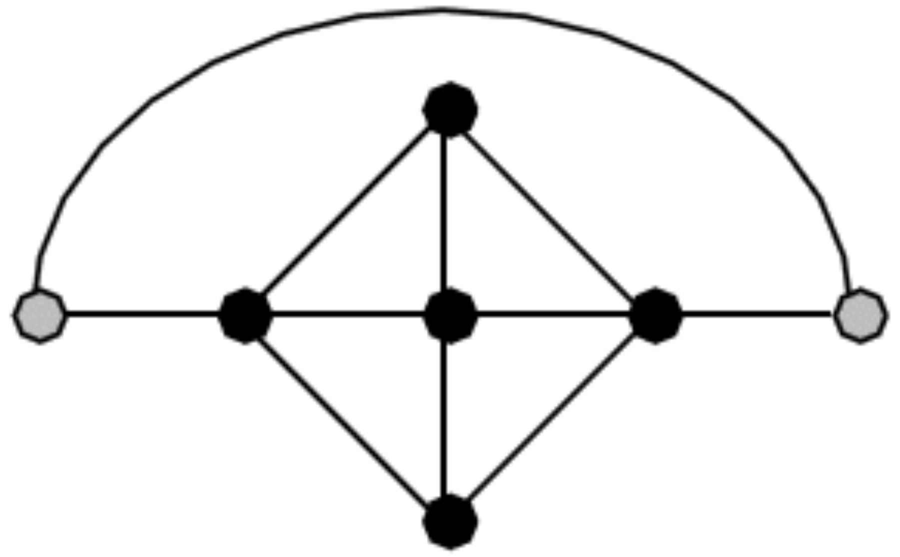
Aspect ratio about 1 (not too long and not too wide)

Symmetry: similar graph structures should look similar

list adapted from Battista et al. 1999

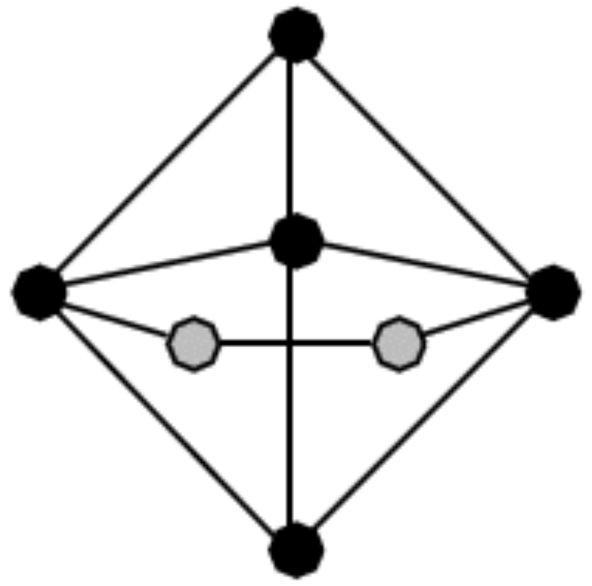
Conflicting Criteria

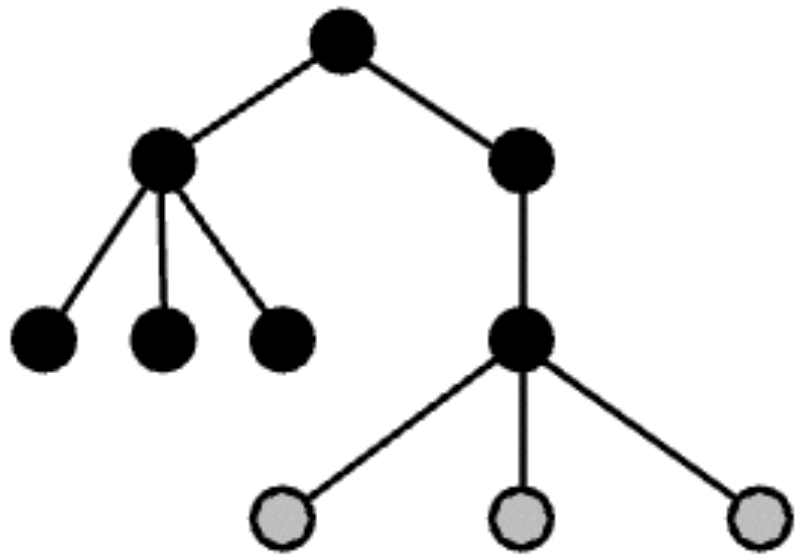
Minimum number of edge crossings



vs.

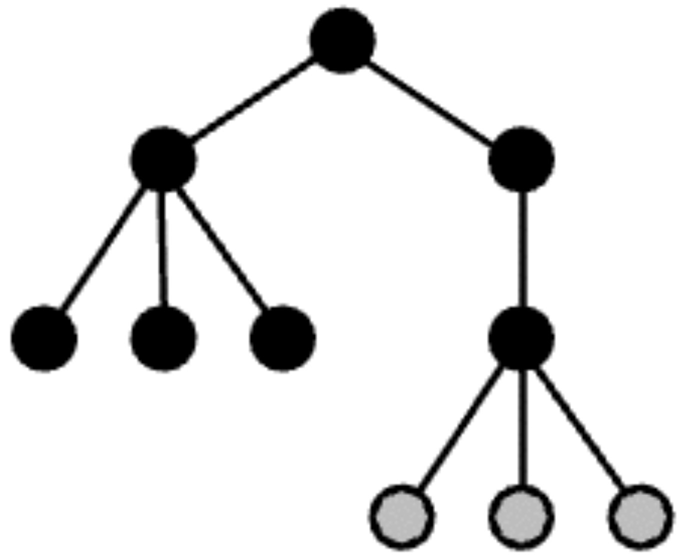
Uniform edge length





Space utilization

vs.



Symmetry

Schulz 2004

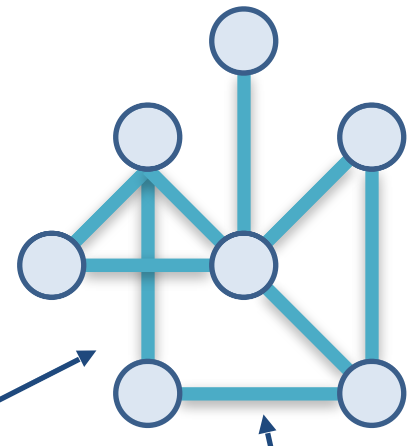
Force Directed Layouts

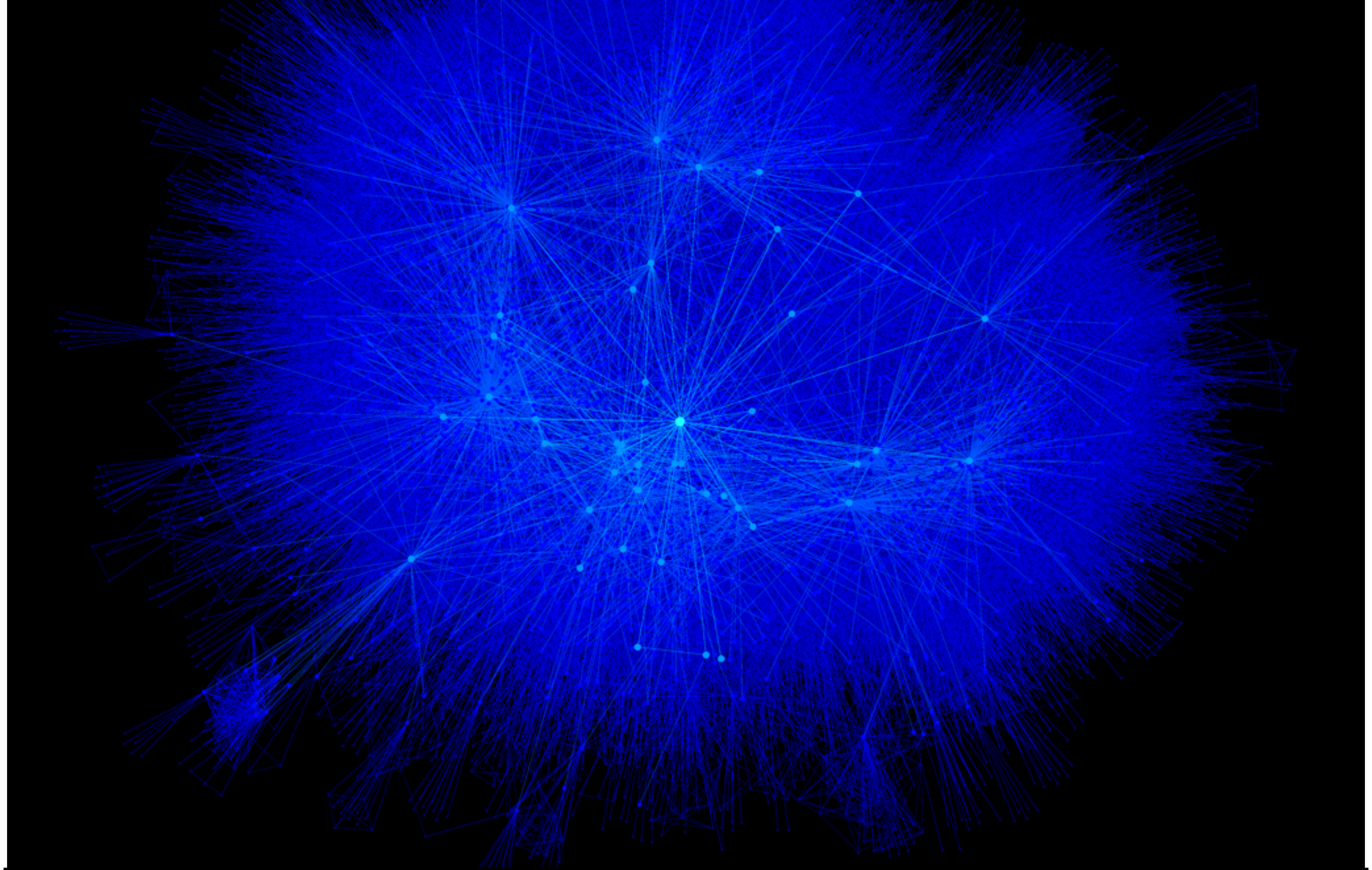
Physics model:
edges = springs,
vertices = repulsive magnets

in practice: damping

Computationally expensive: $O(n^3)$

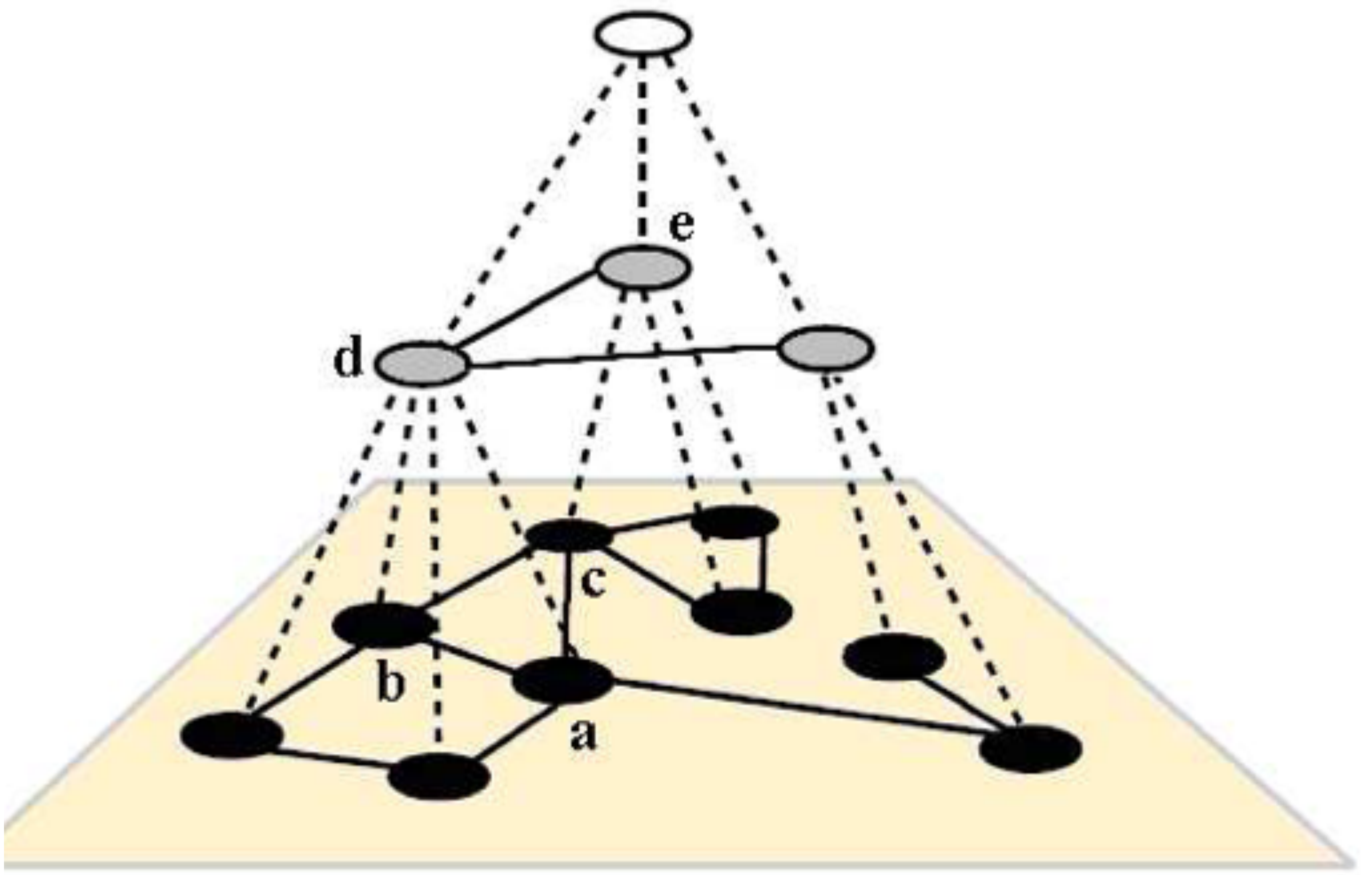
Limit (interactive): ~1000 nodes

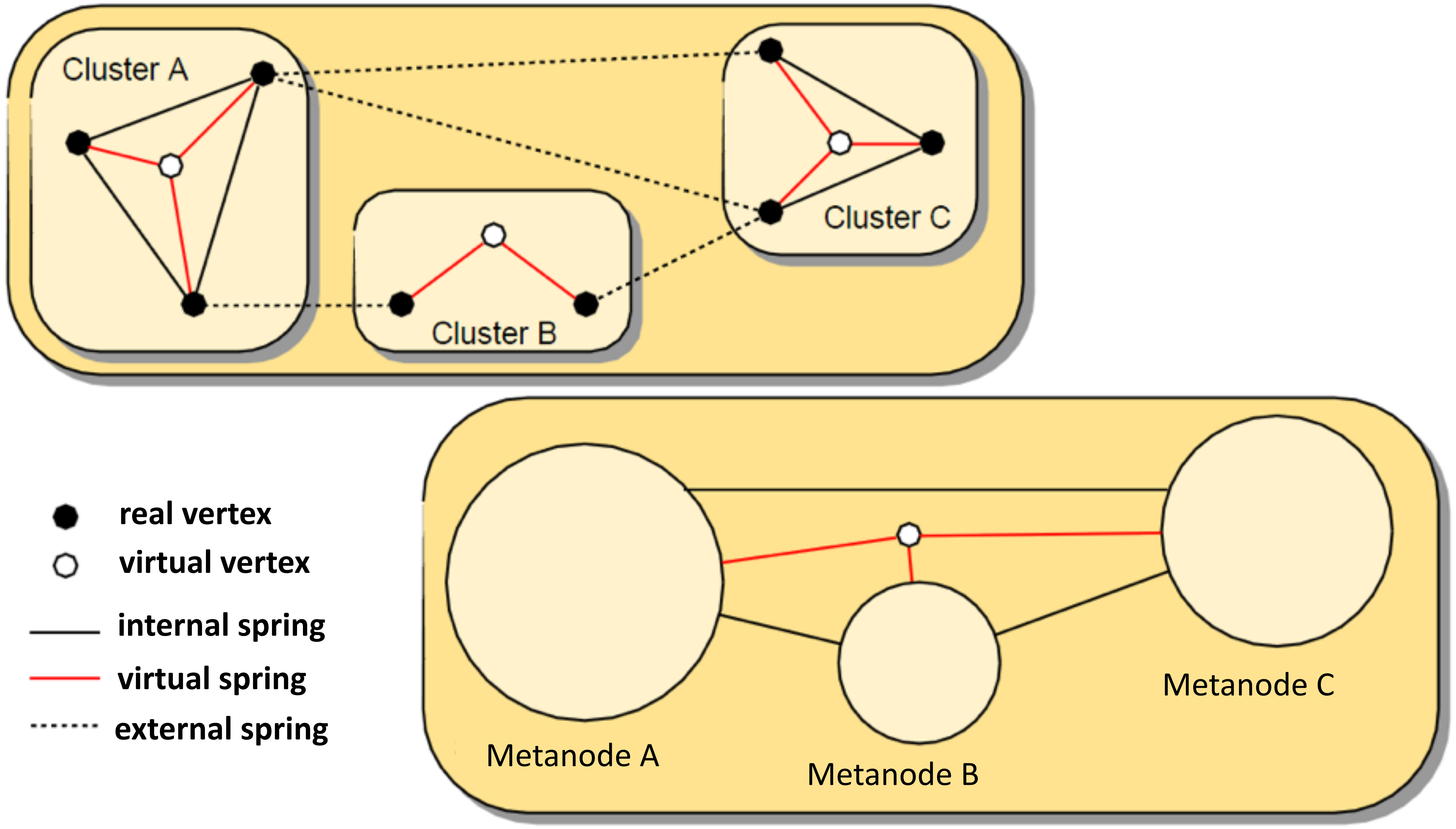




Giant Hairball

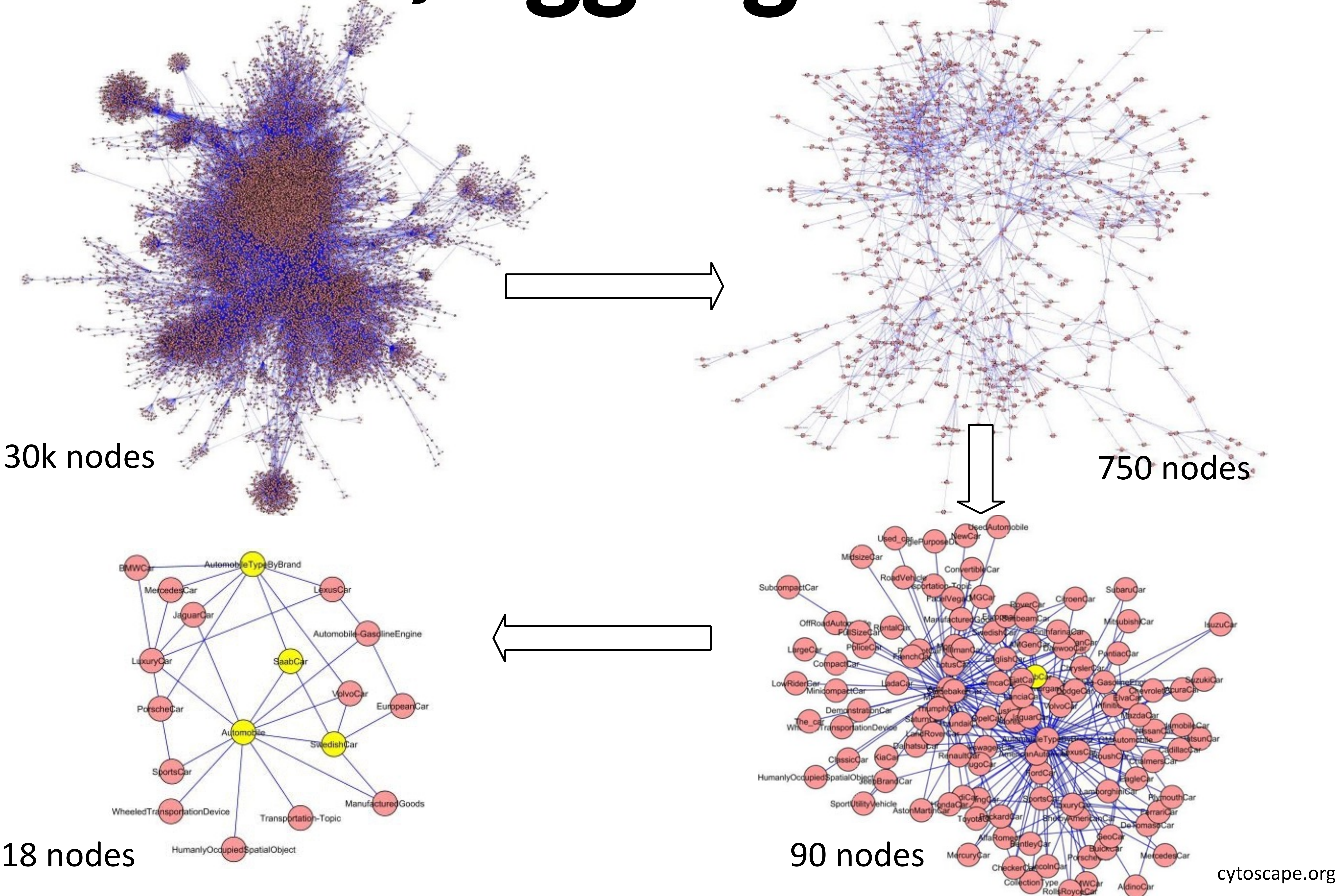
Adress Computational Scalability: Multilevel Approaches





[Schulz 2004]

Abstraction/Aggregation

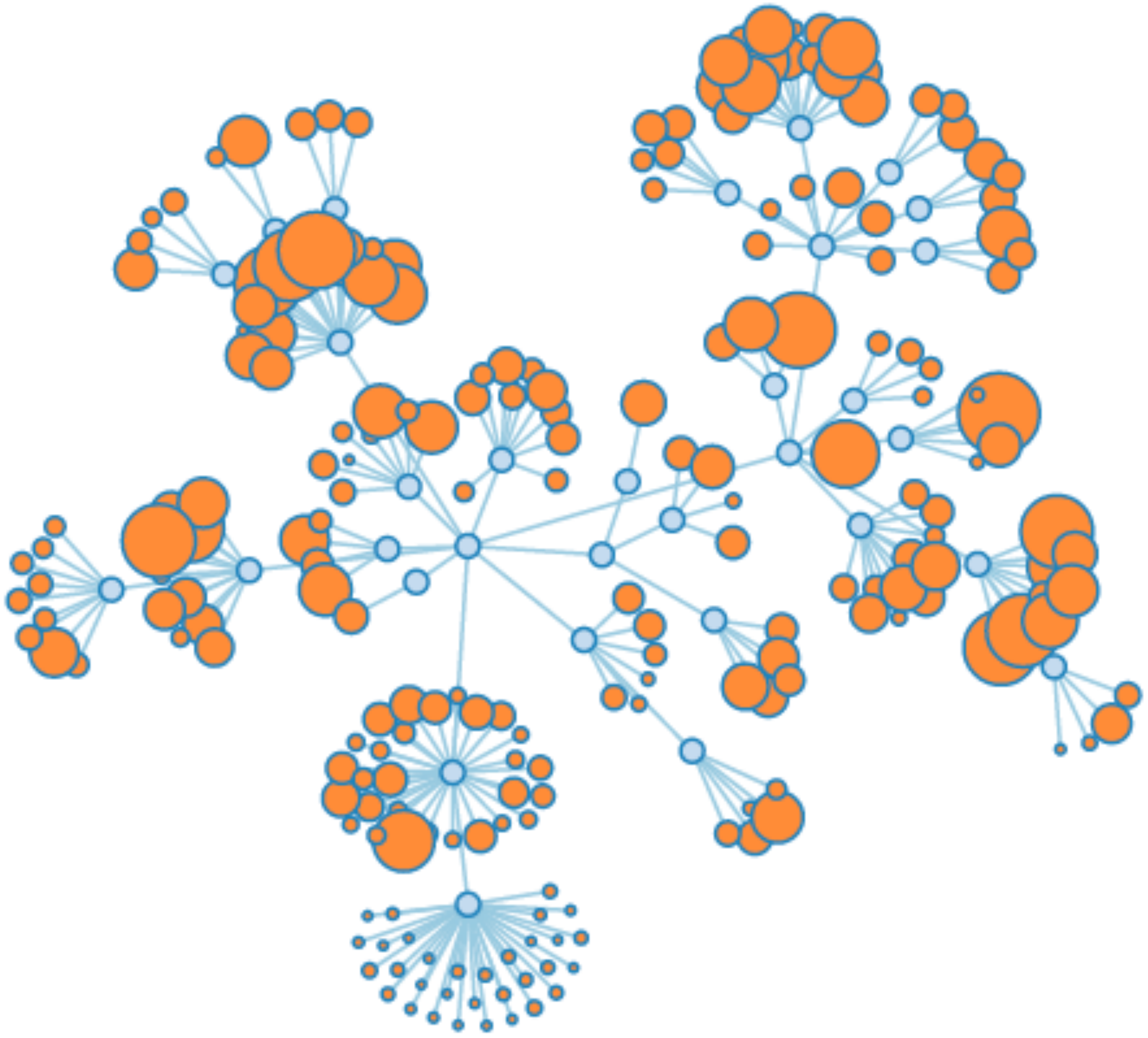


Collapsible Force Layout

Supernodes: aggregate of nodes

manual or algorithmic

- clustering

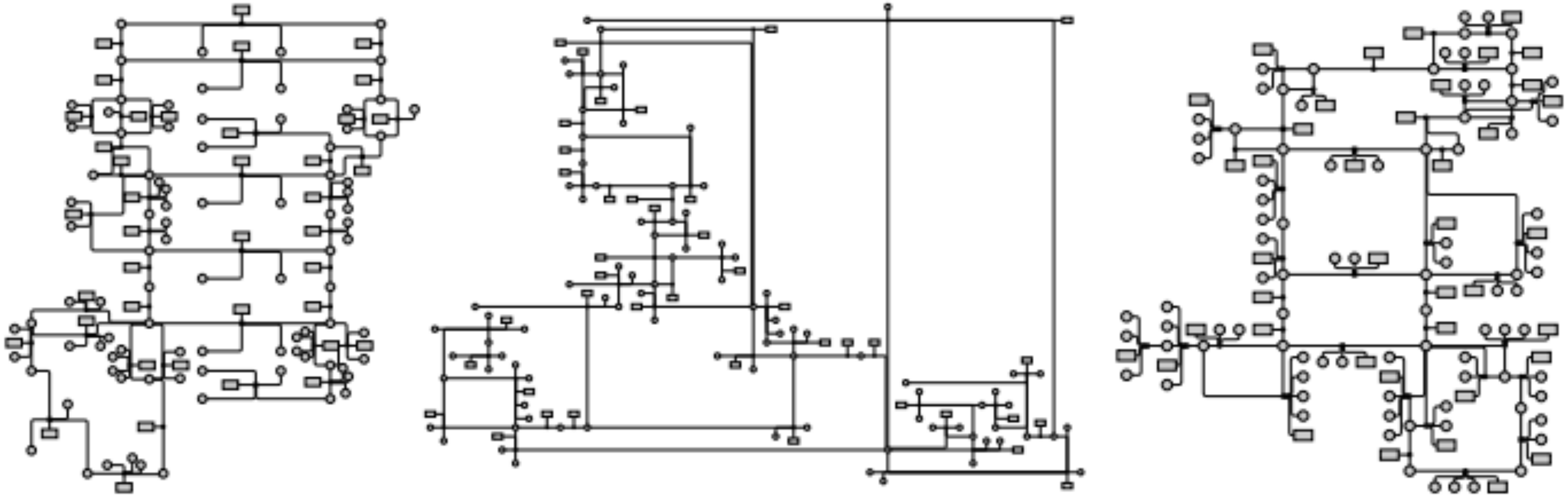


HOLA: Human-like Orthogonal Layout

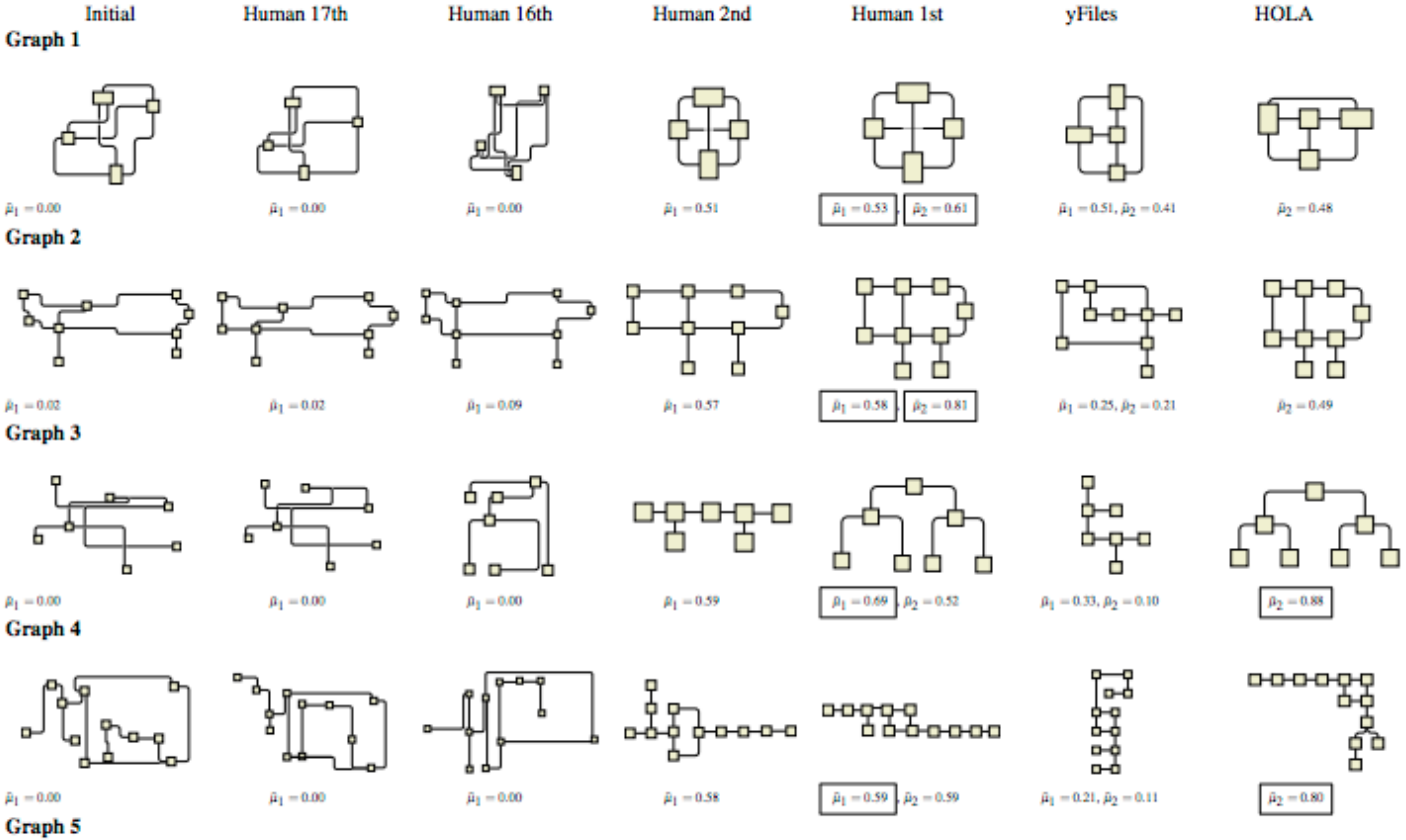
Study how humans lay-out a graph

Try to emulate layout

Left: human, middle: conventional algo, right new algo



[Kieffer et al, InfoVis 2015]

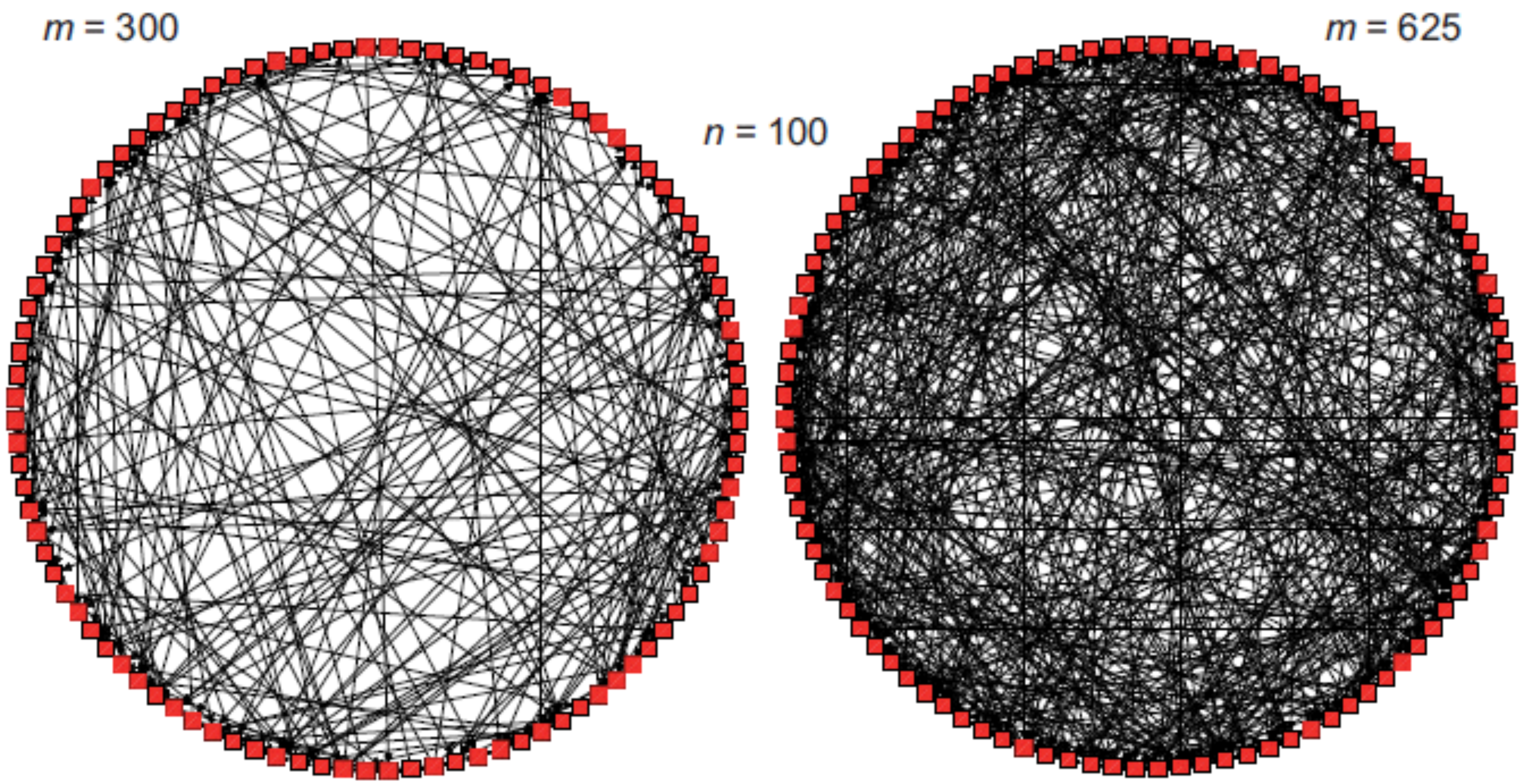

Initial
Human 17th
Human 16th
Human 2nd
Human 1st
yFiles
HOLA
Graph 1
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.51$
$\hat{\mu}_1 = 0.53$, $\hat{\mu}_2 = 0.61$
$\hat{\mu}_1 = 0.51$, $\hat{\mu}_2 = 0.41$
$\hat{\mu}_2 = 0.48$
Graph 2
$\hat{\mu}_1 = 0.02$
$\hat{\mu}_1 = 0.02$
$\hat{\mu}_1 = 0.09$
$\hat{\mu}_1 = 0.57$
$\hat{\mu}_1 = 0.58$, $\hat{\mu}_2 = 0.81$
$\hat{\mu}_1 = 0.25$, $\hat{\mu}_2 = 0.21$
$\hat{\mu}_2 = 0.49$
Graph 3
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.59$
$\hat{\mu}_1 = 0.69$, $\hat{\mu}_2 = 0.52$
$\hat{\mu}_1 = 0.33$, $\hat{\mu}_2 = 0.10$
$\hat{\mu}_2 = 0.88$
Graph 4
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.00$
$\hat{\mu}_1 = 0.58$
$\hat{\mu}_1 = 0.59$, $\hat{\mu}_2 = 0.59$
$\hat{\mu}_1 = 0.21$, $\hat{\mu}_2 = 0.11$
$\hat{\mu}_2 = 0.80$
Graph 5

Styled / Restricted Layouts

Circular Layout

Node ordering

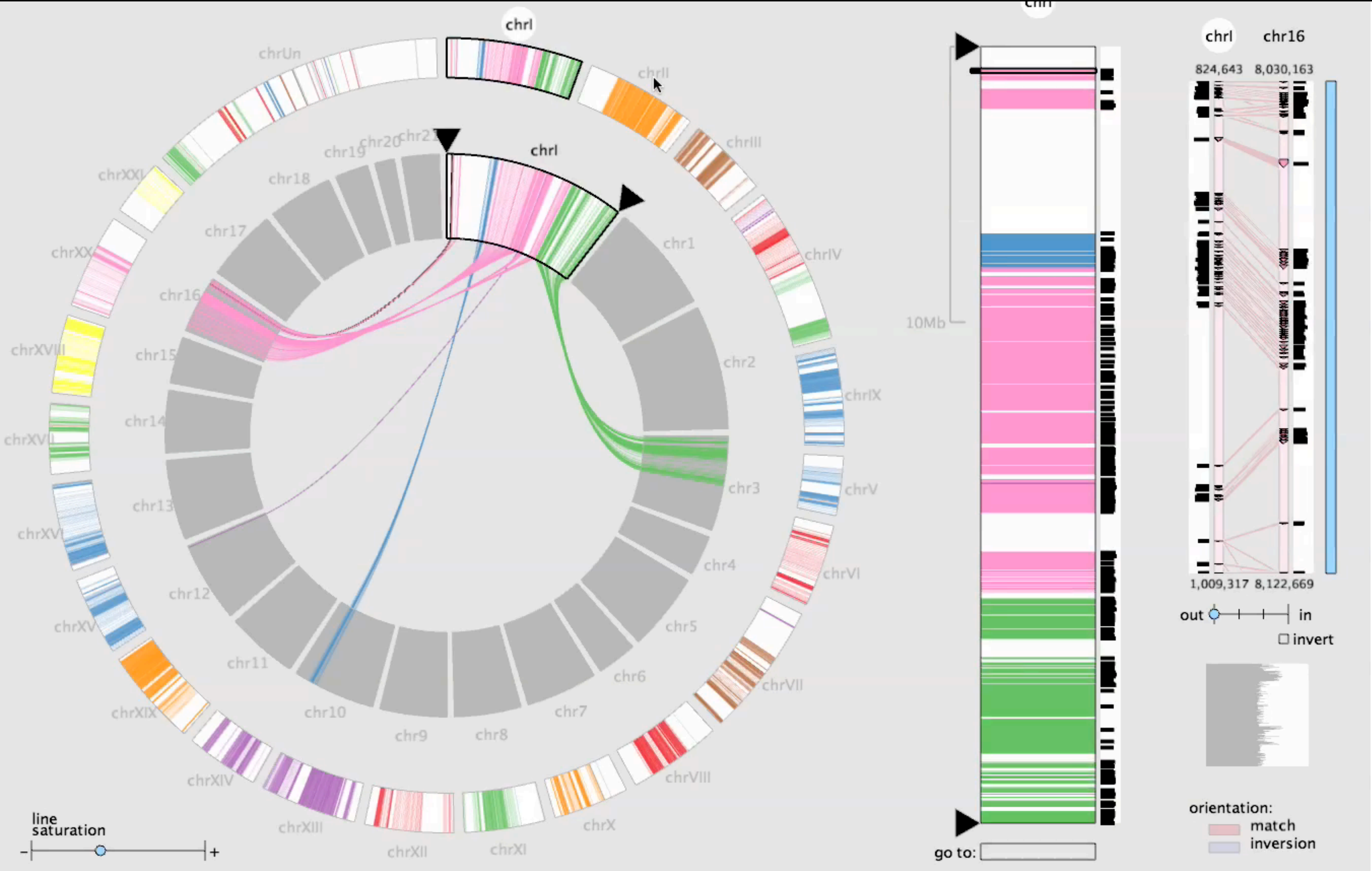
Edge Clutter



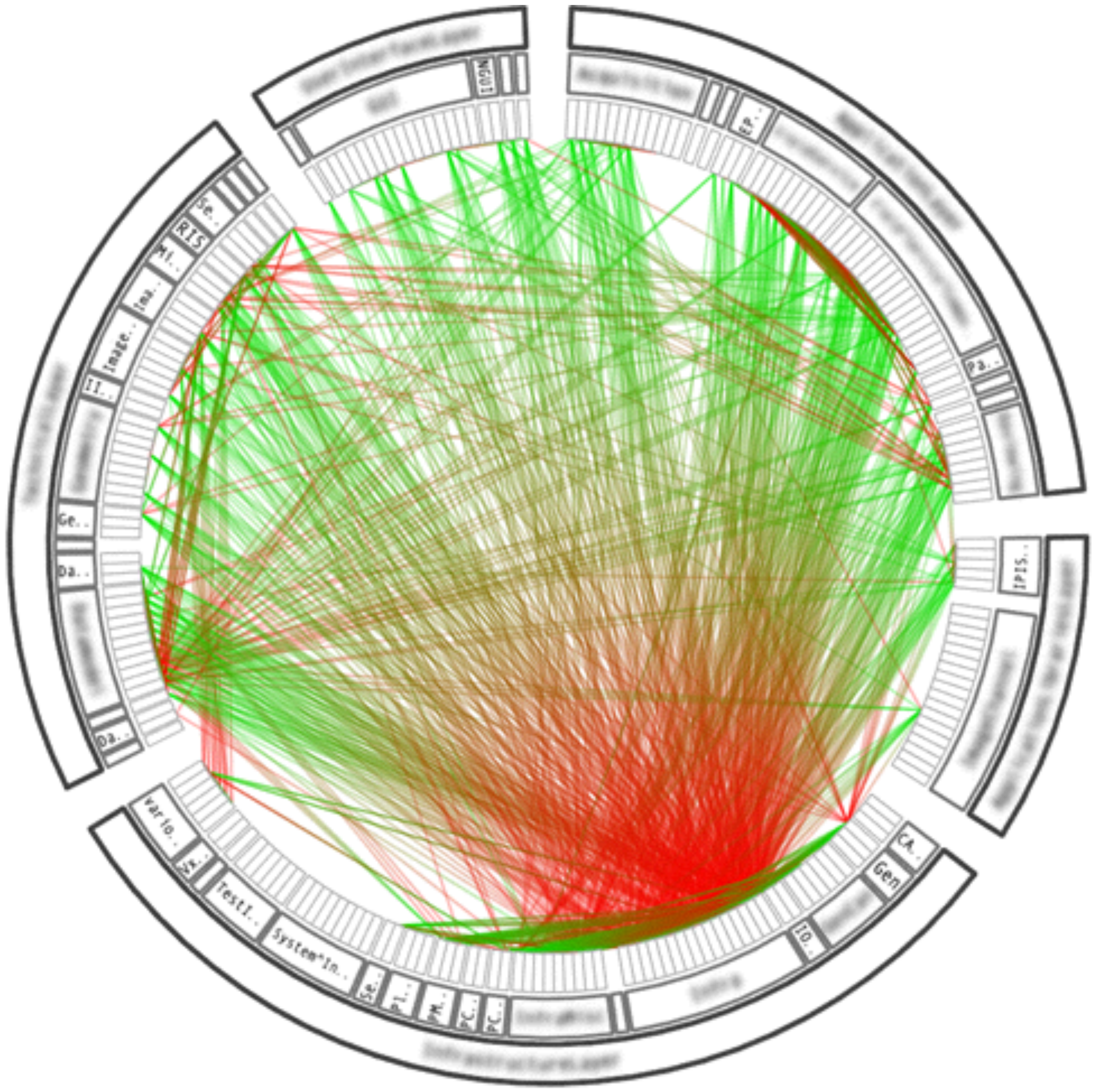
ca. 3% of all possible edges

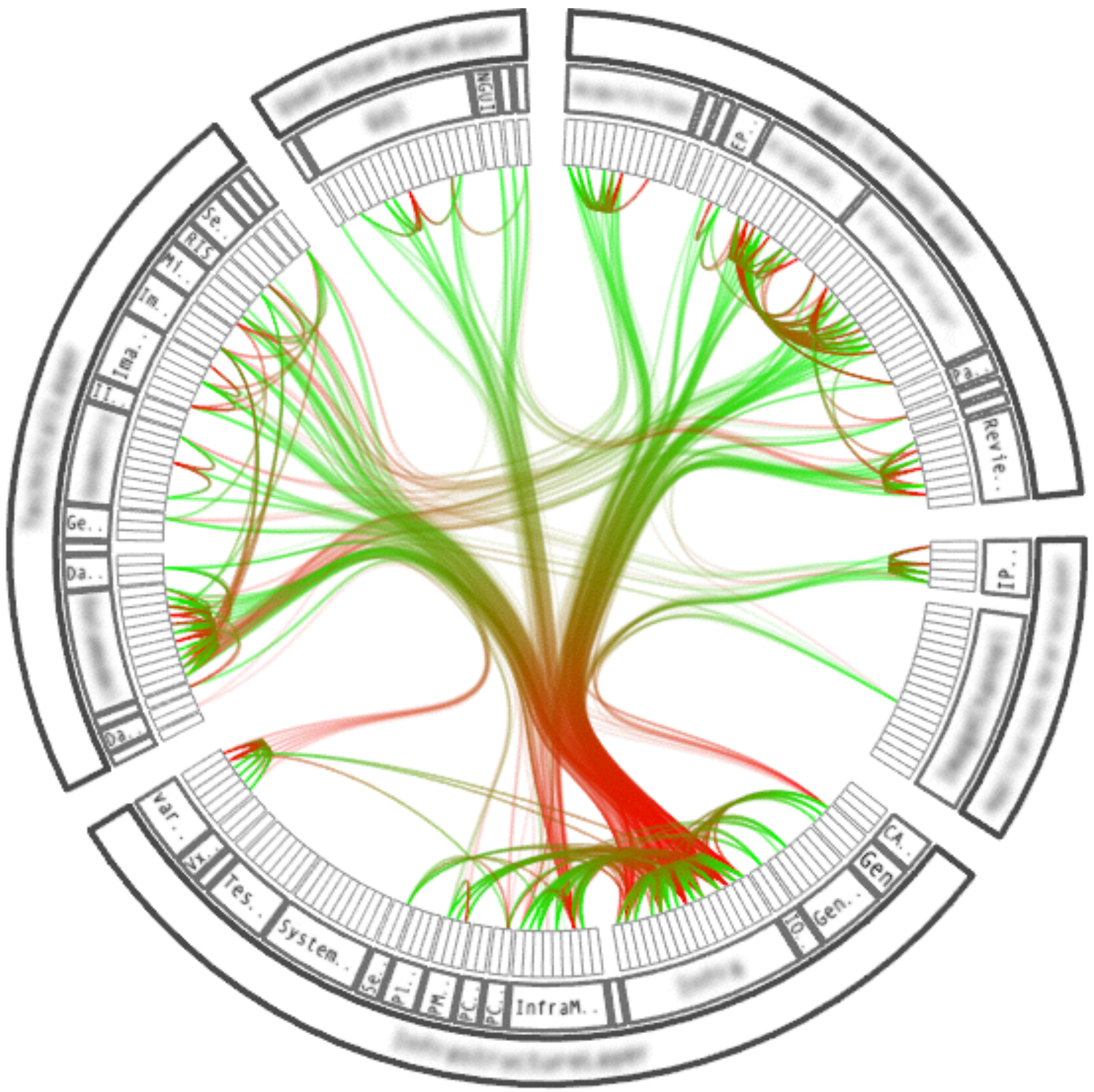
ca. 6,3% of all possible edges

Example: MizBee [Meyer et al. 2009]



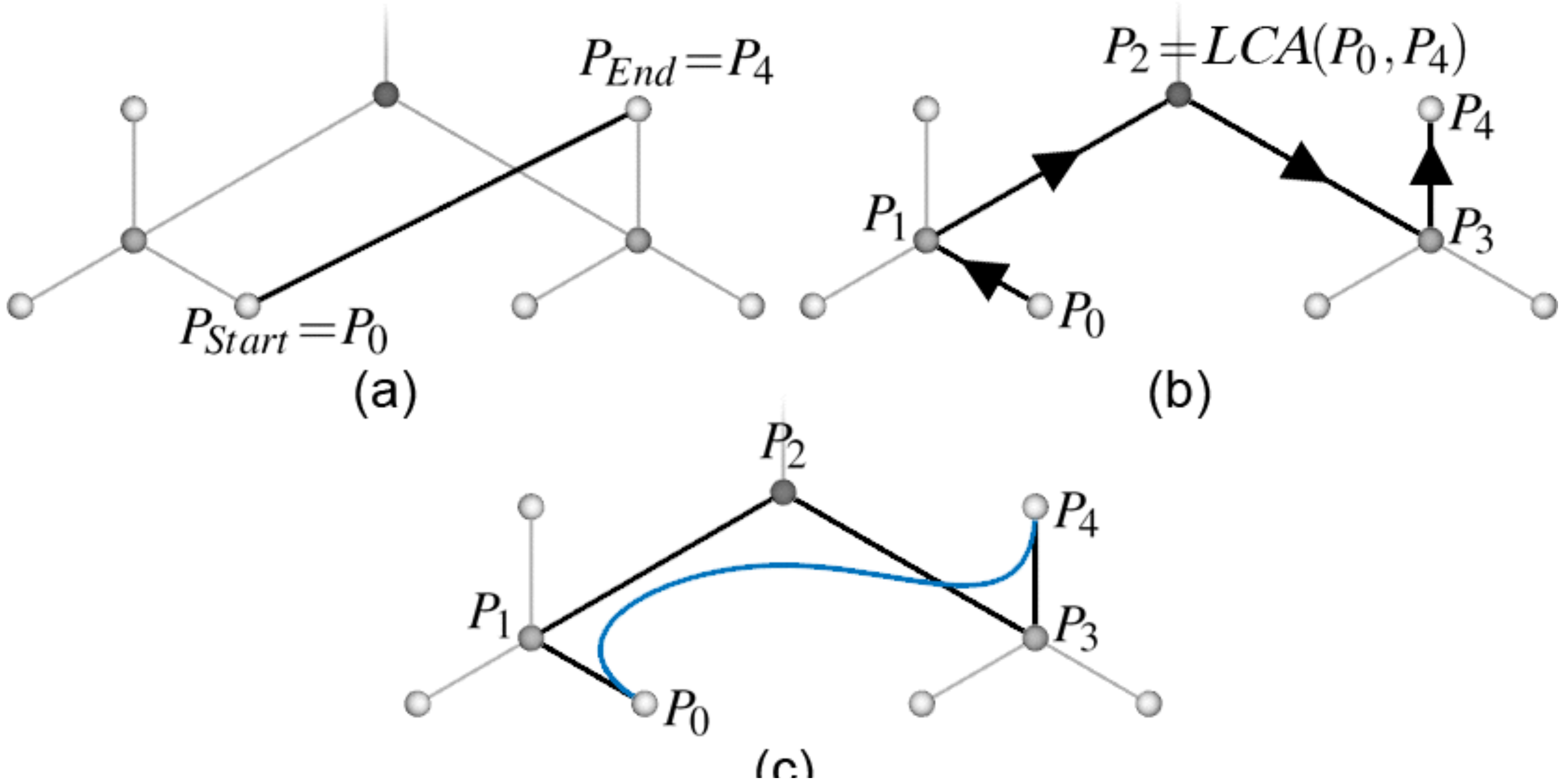
Reduce Clutter: Edge Bundling

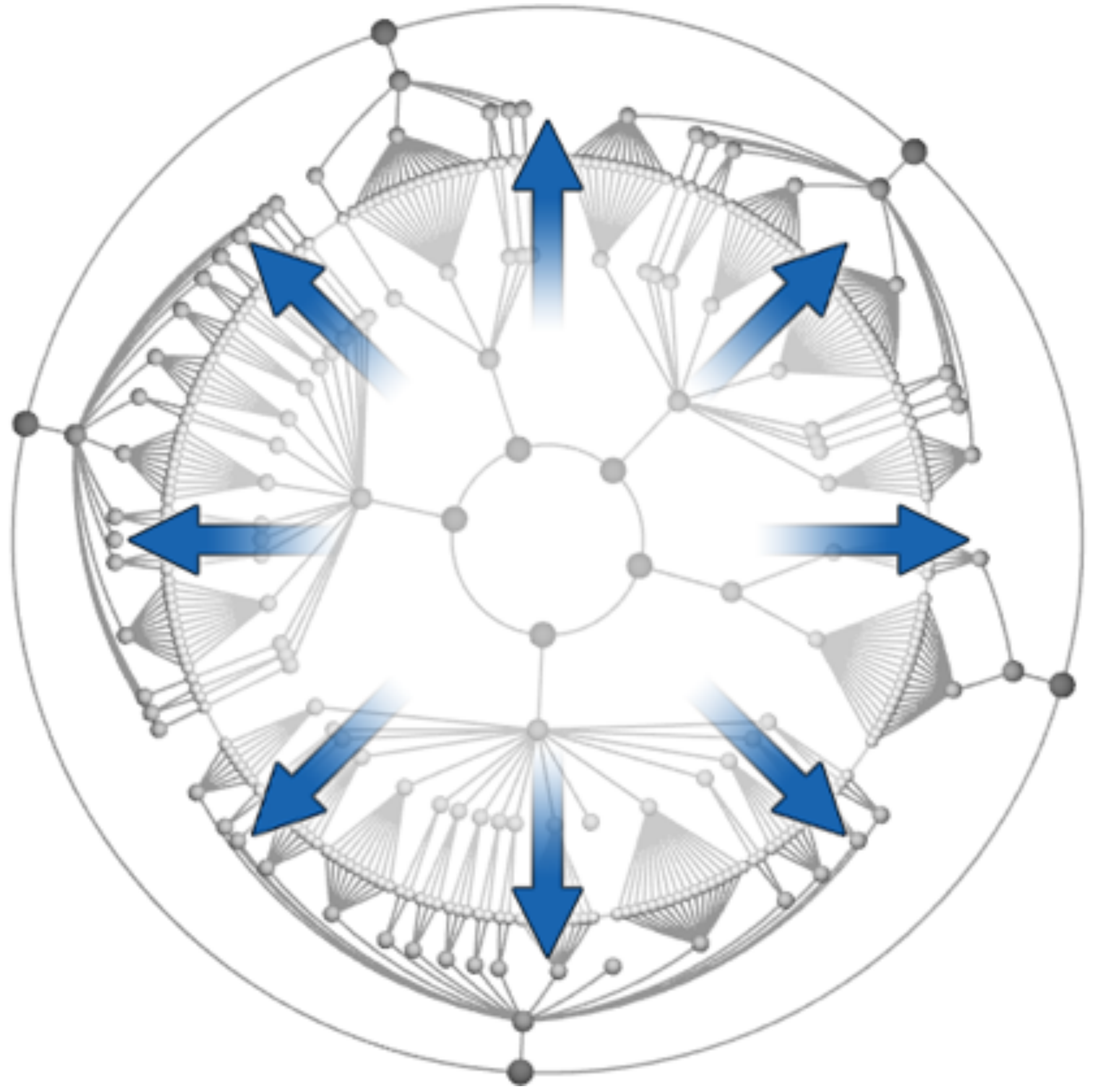


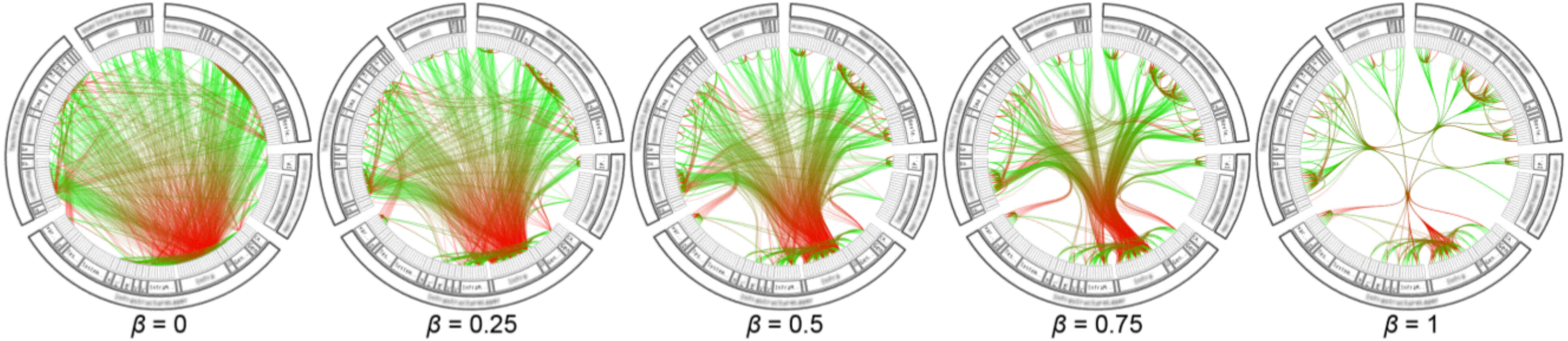


Holten et al. 2006

Hierarchical Edge Bundling







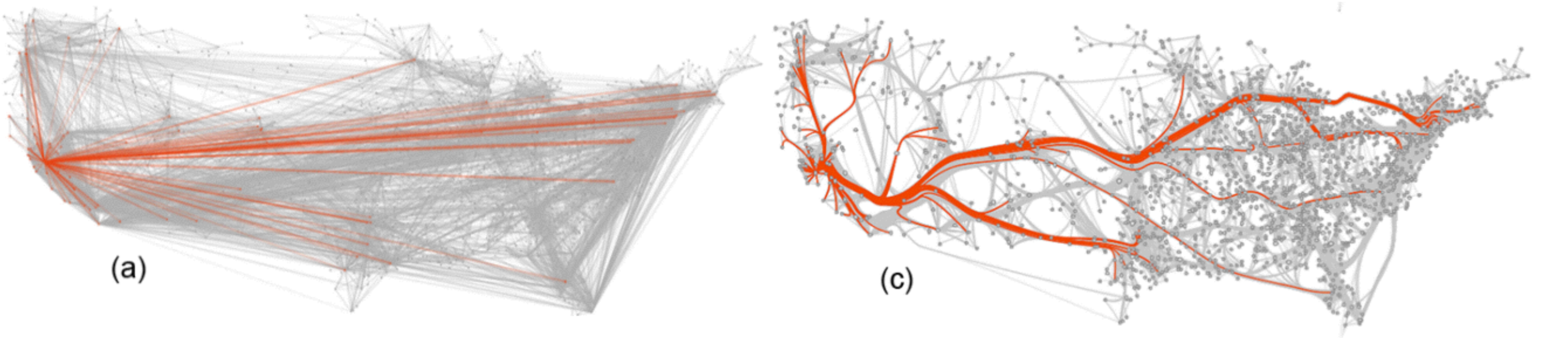
Bundling Strength

Holten et al. 2006

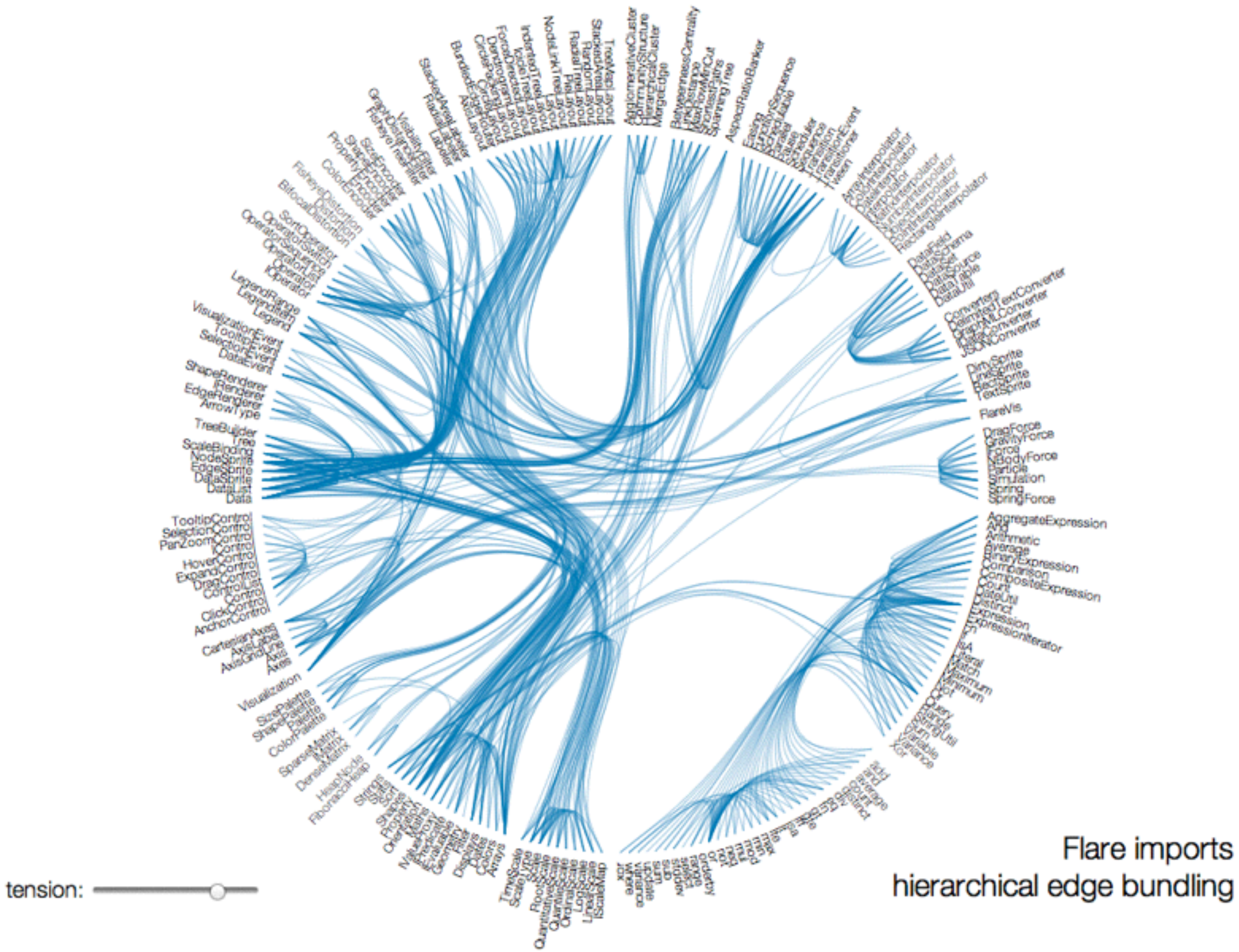
Fixed Layouts

Can’t vary position of nodes

Edge routing important



Bundling Strength



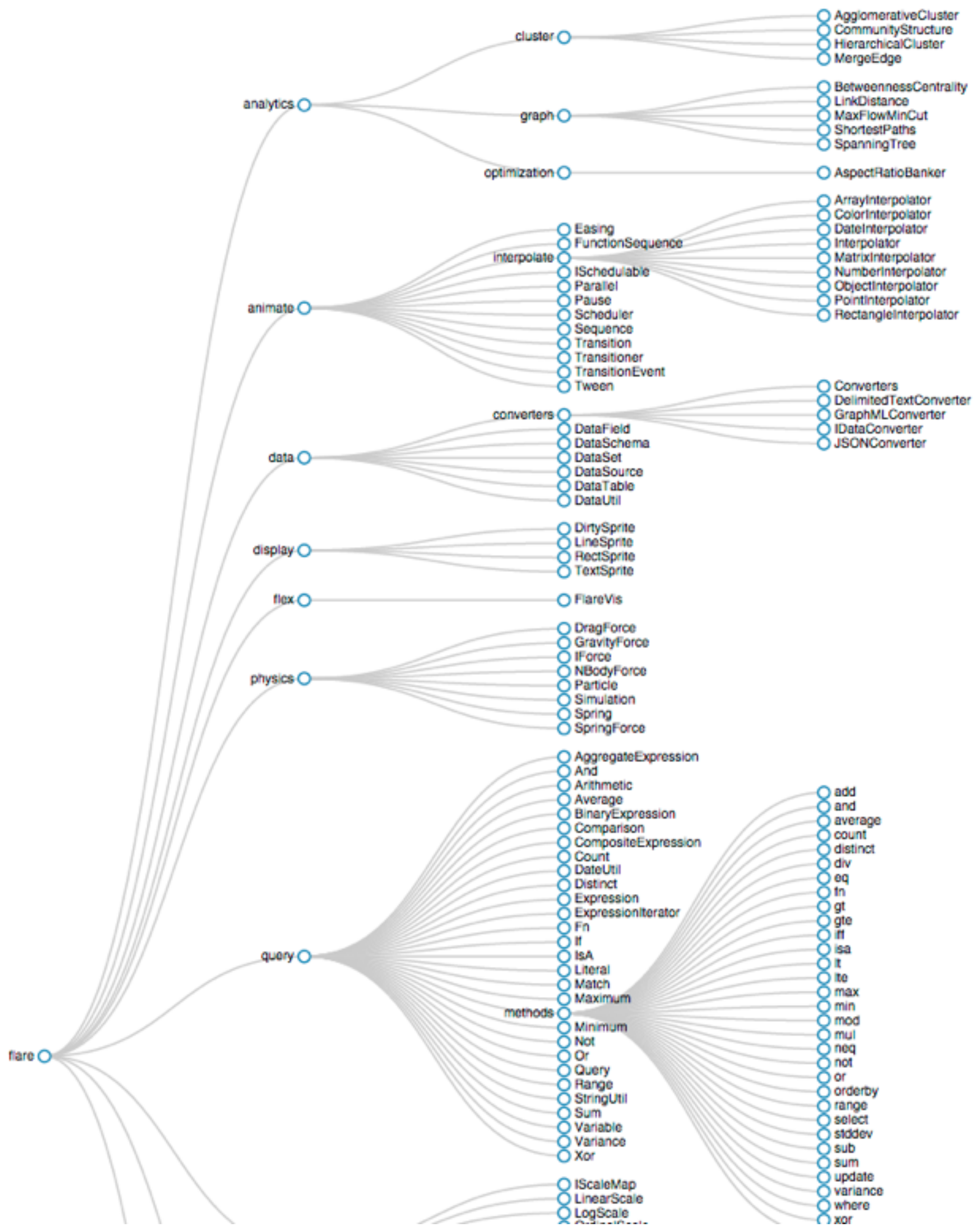
mbostock.github.com/d3/talk/20111116/bundle.html

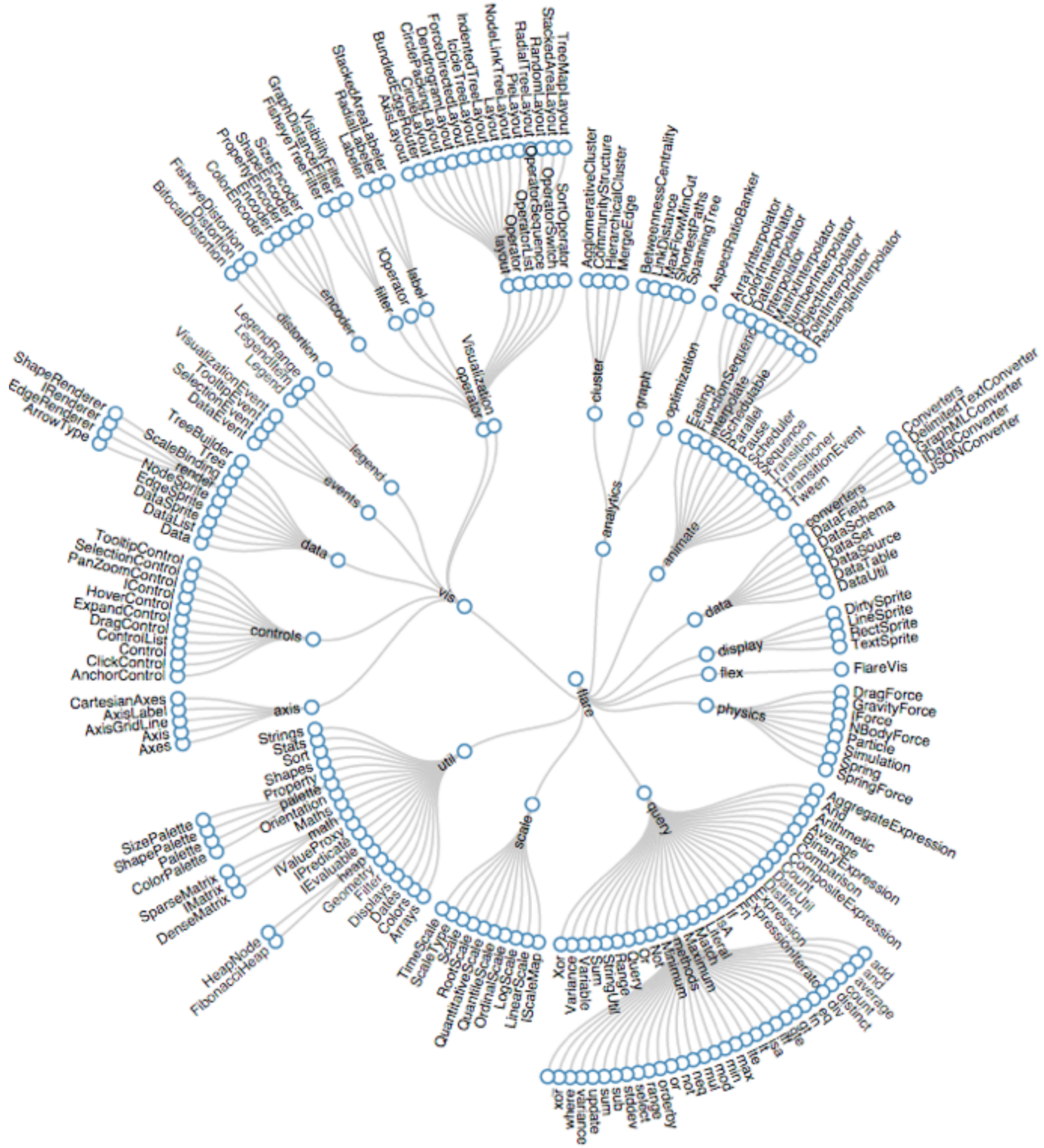
Michael Bostock

Explicit Tree Visualization

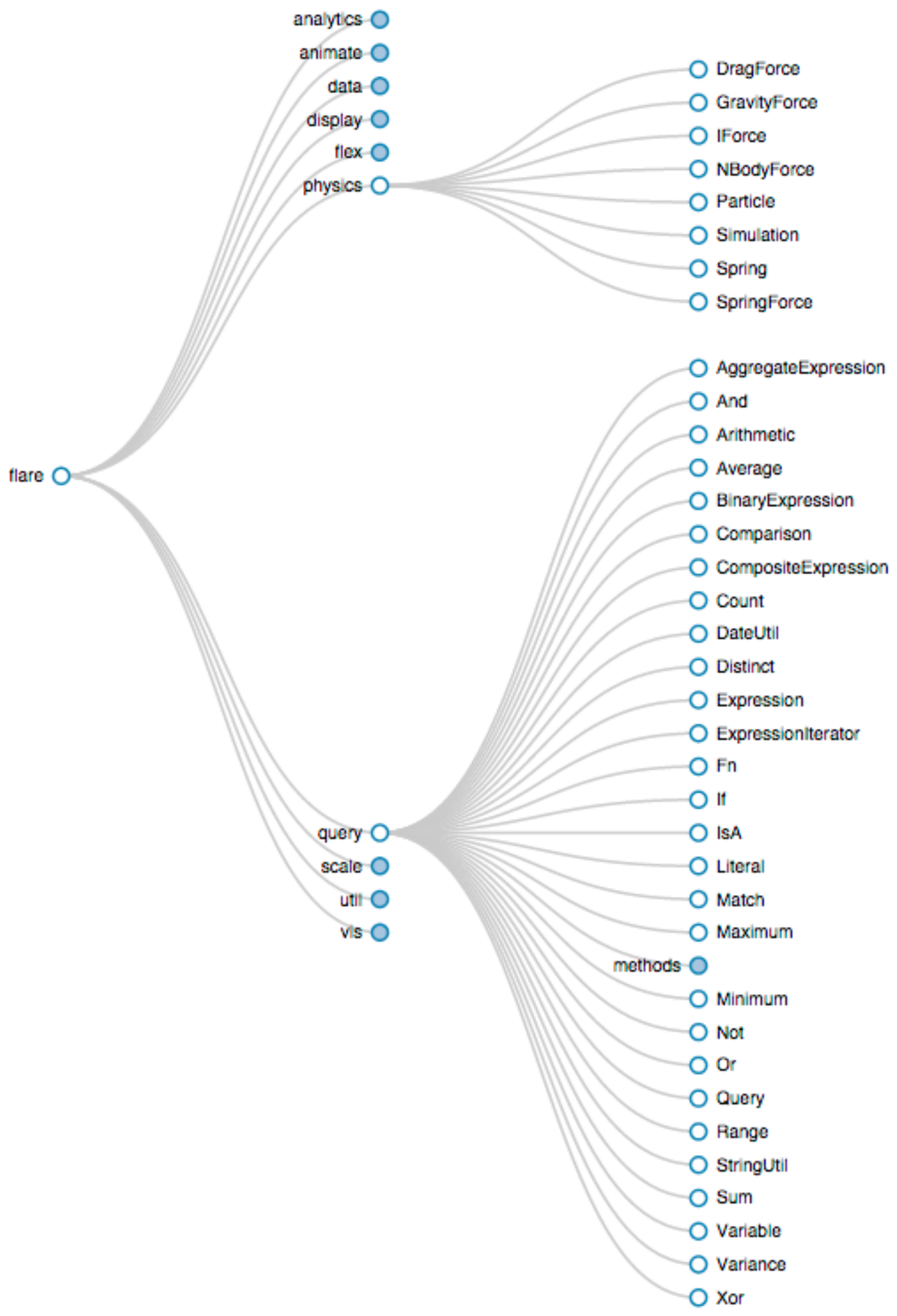
Reingold–
Tilford layout

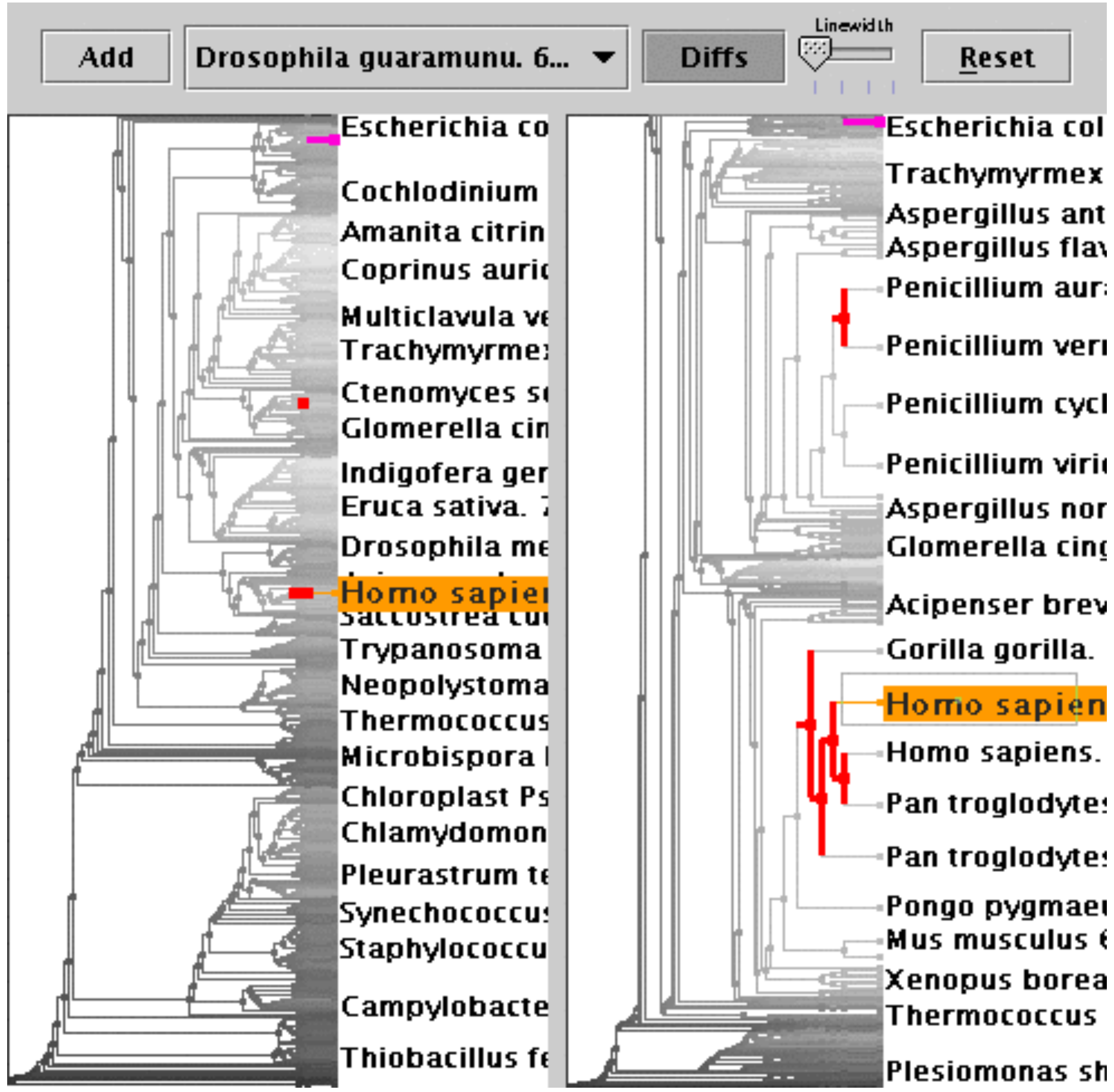
http://billmill.org/pymag-
trees/





Tree Interaction, Tree Comparison





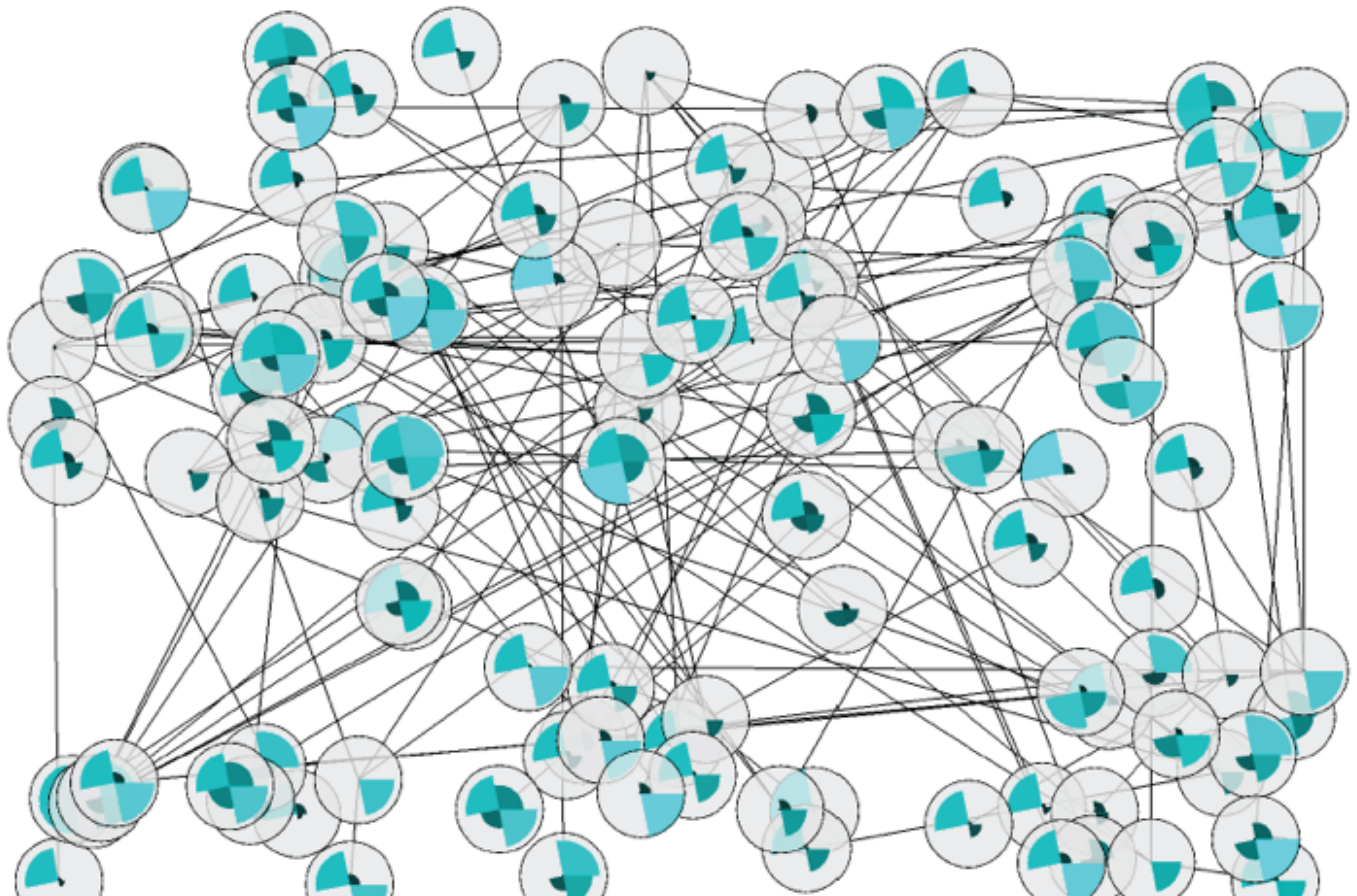
Multivariate Graphs

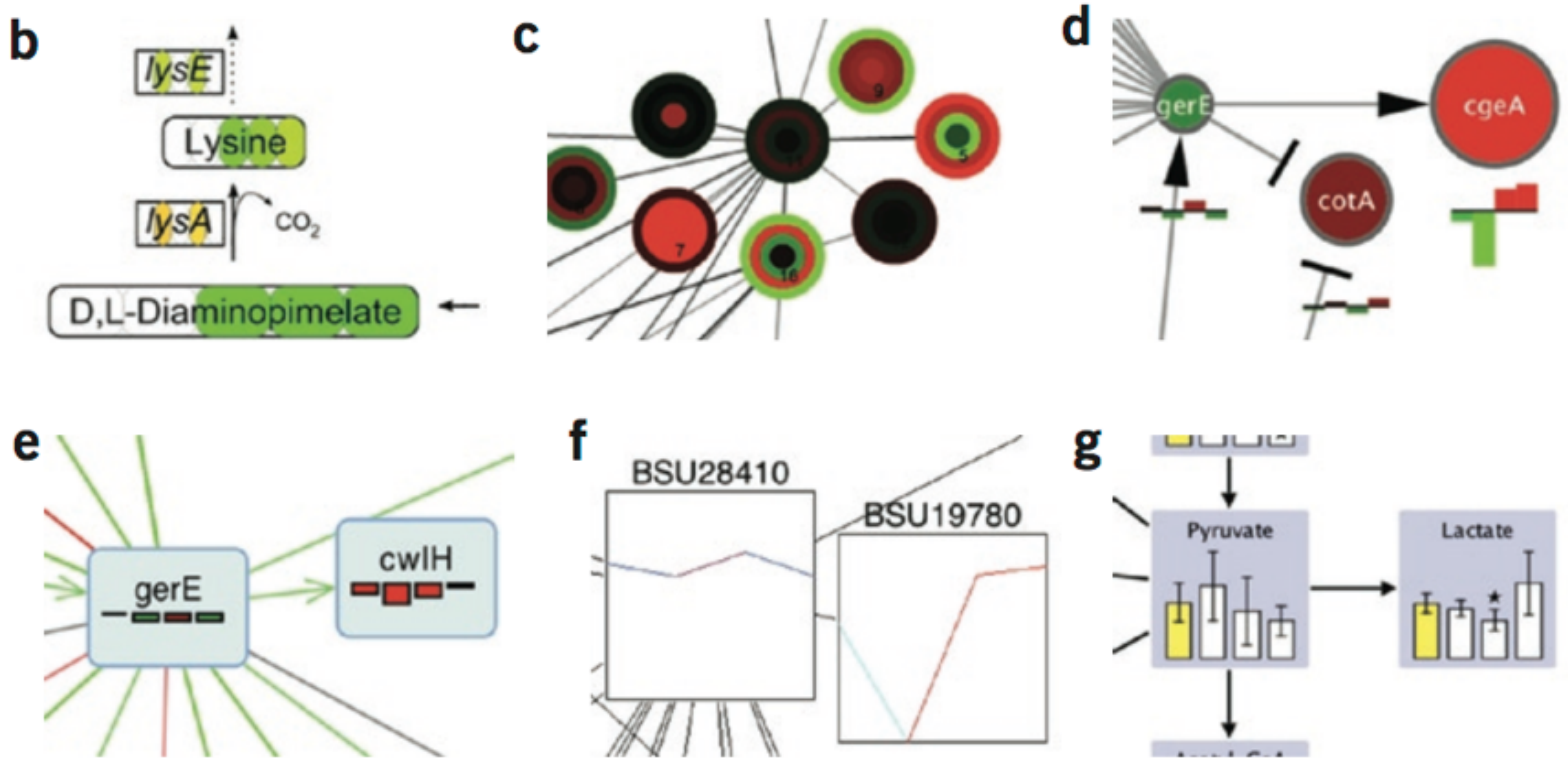
Node Attributes

Coloring

Glyphs

-> Limited in scalability

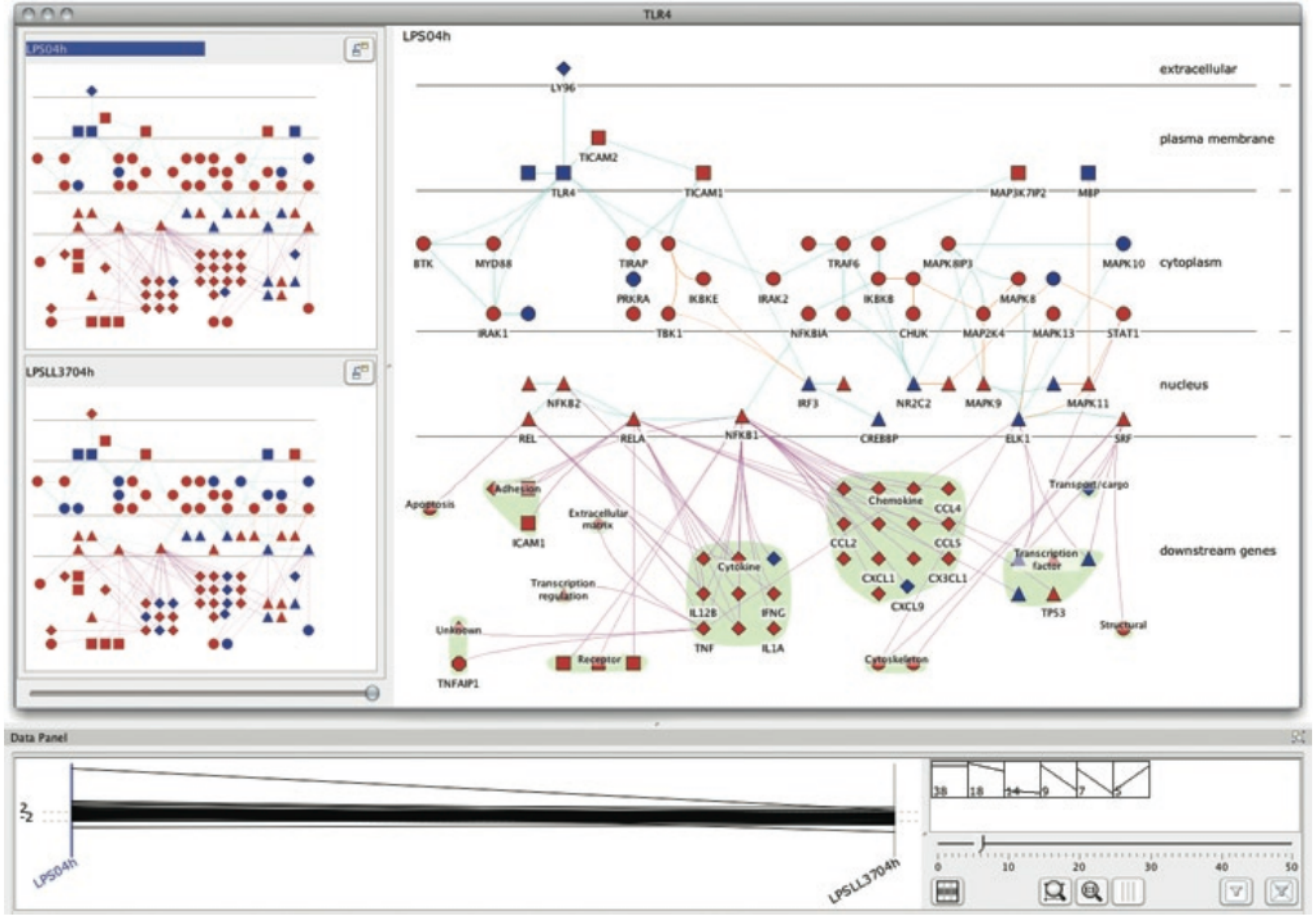




Small Multiples

Cerebral [Barsky, 2008]

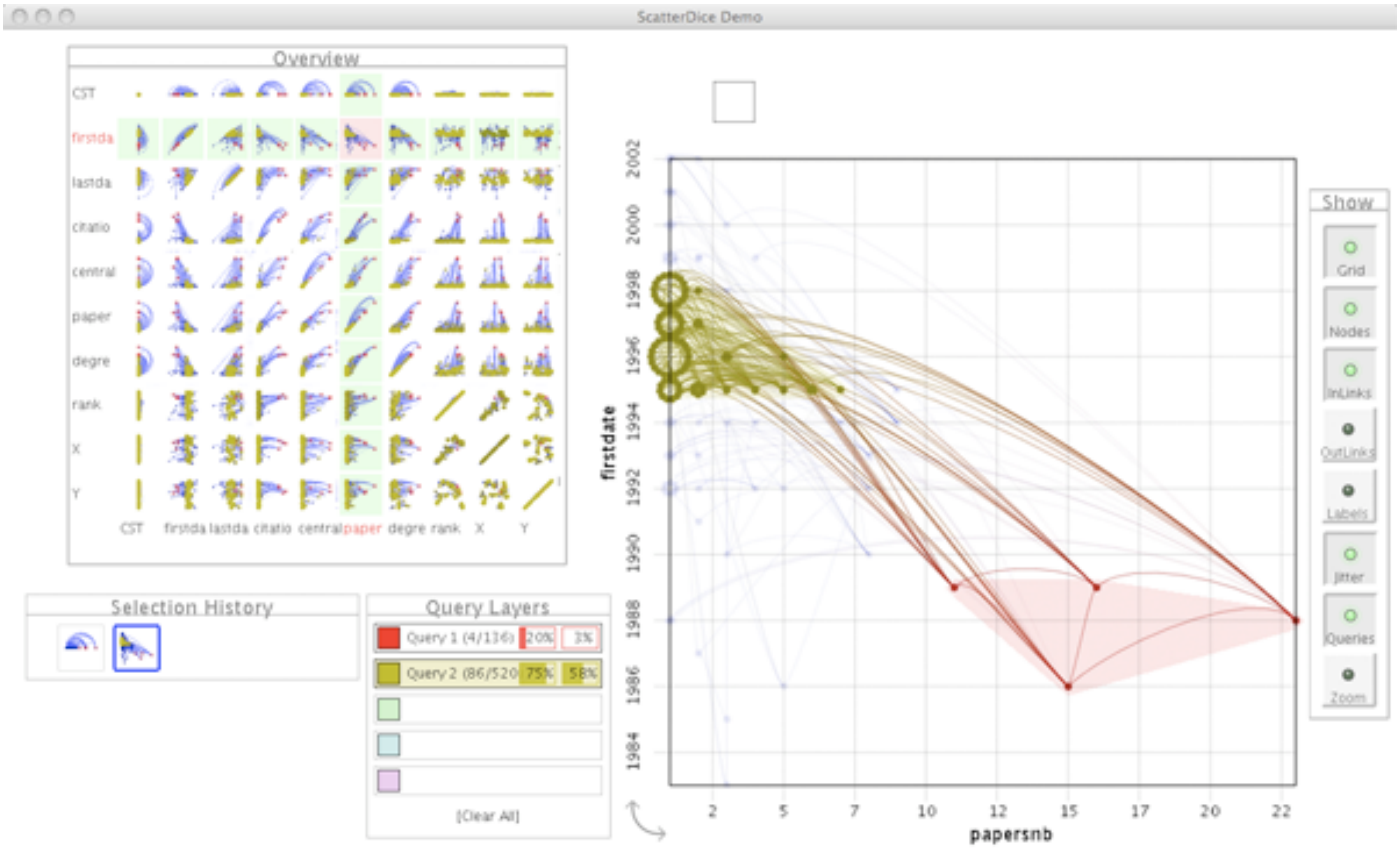
Each dimension in its own window



Data-driven node positioning

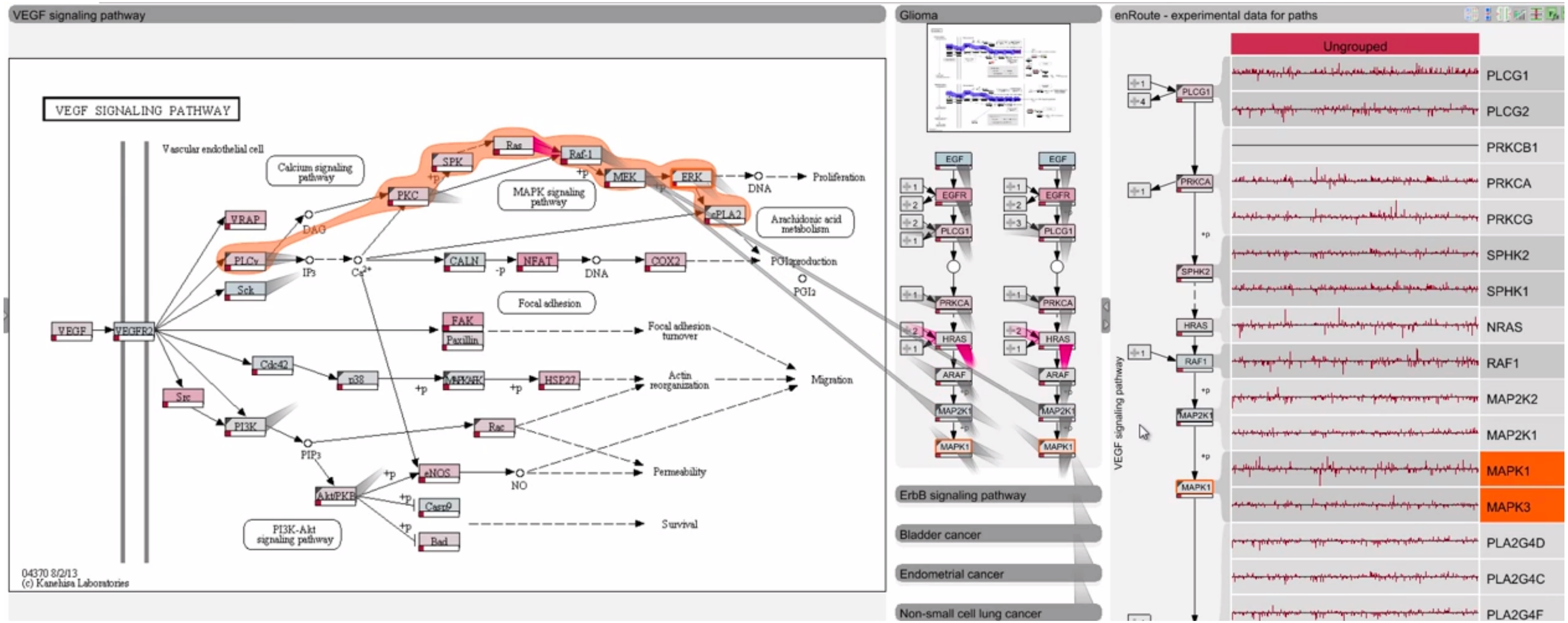
GraphDice

Nodes are laid out according to attribute values



[Bezerianos et al, 2010]

Path Extraction & Multiple Views



Experi-mental Data and Pathways

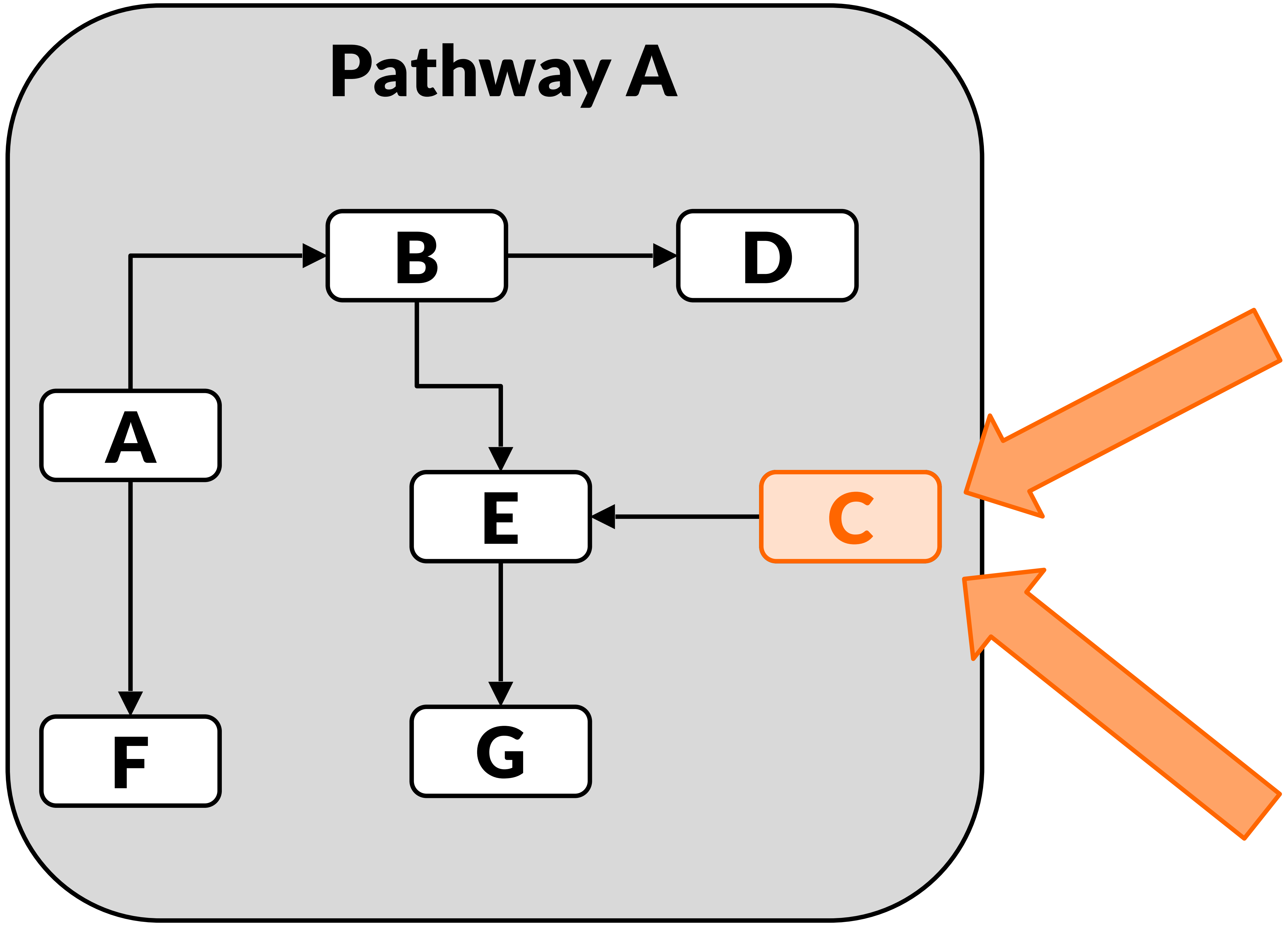
[Partl, BioVis ‘12]

Cannot account for variation found in real-world data

Branches can be (in)activated due to

- mutation,
- changed gene expression,
- modulation due to drug treatment,
- etc.

Many Node Attributes

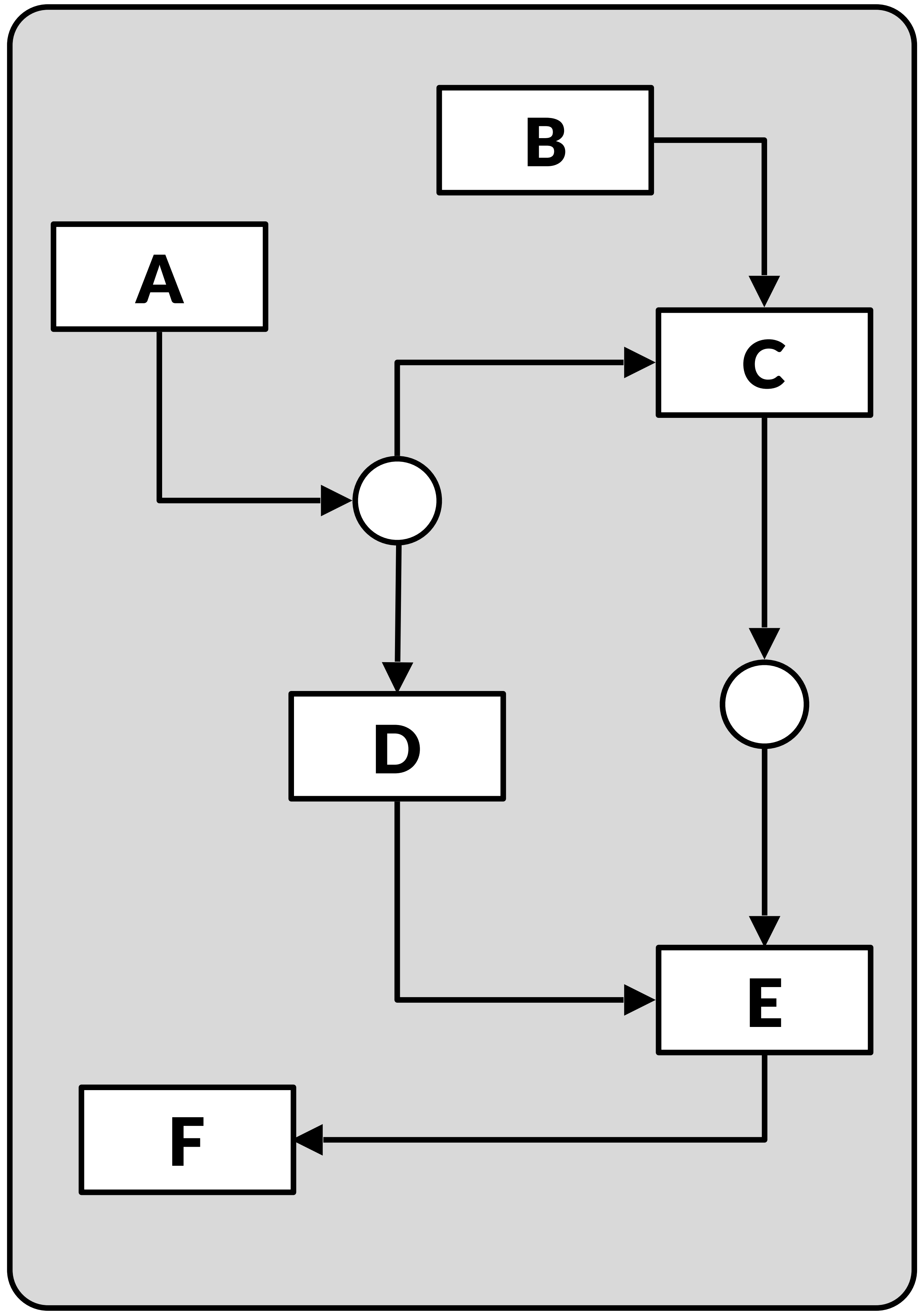


Node	Sample 1	Sample 2	Sample 3	...
A	0.55	0.95	0.83	...
B	0.12	0.42	0.16	...
C	0.33	0.65	0.38	...
...	...	...	...	

Node	Sample 1	Sample 2	Sample 3	...
A	low	low	very high	...
B	normal	low	high	...
C	high	very low	normal	...
...	...	...	...	

How to visualize experimental data on pathways?

Good Old Color Coding



A	-3.4	4.2	5.1	4.2
B	2.8	1.8	1.3	1.1
C	3.1	-2.2	2.4	2.2
D	-3	-2.8	1.6	1.0
E	0.5	0.3	-1.1	1.3
F	0.3	0.3	1.8	-0.3

[Lindroos2002]

Challenge: Data Scale & Heterogeneity

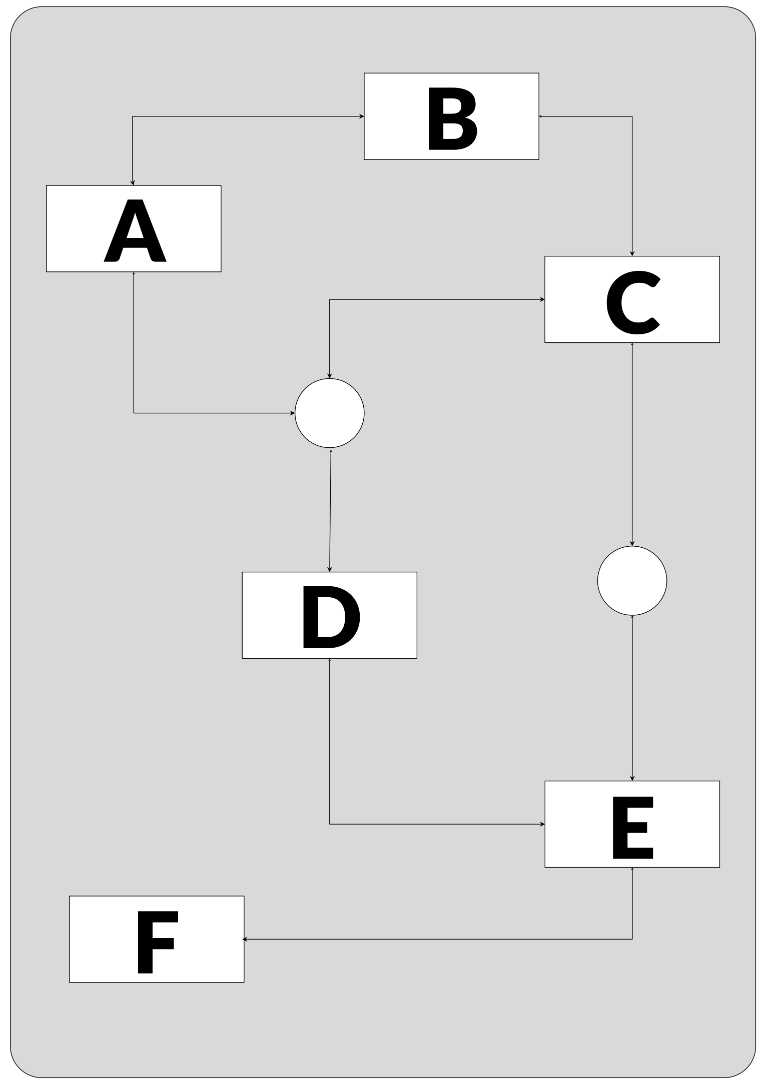
- Large number of experiments
  - Large datasets have more than 500 experiments
- Multiple groups/conditions
- Different types of data

Challenge: Supporting Multiple Tasks

Two central tasks:

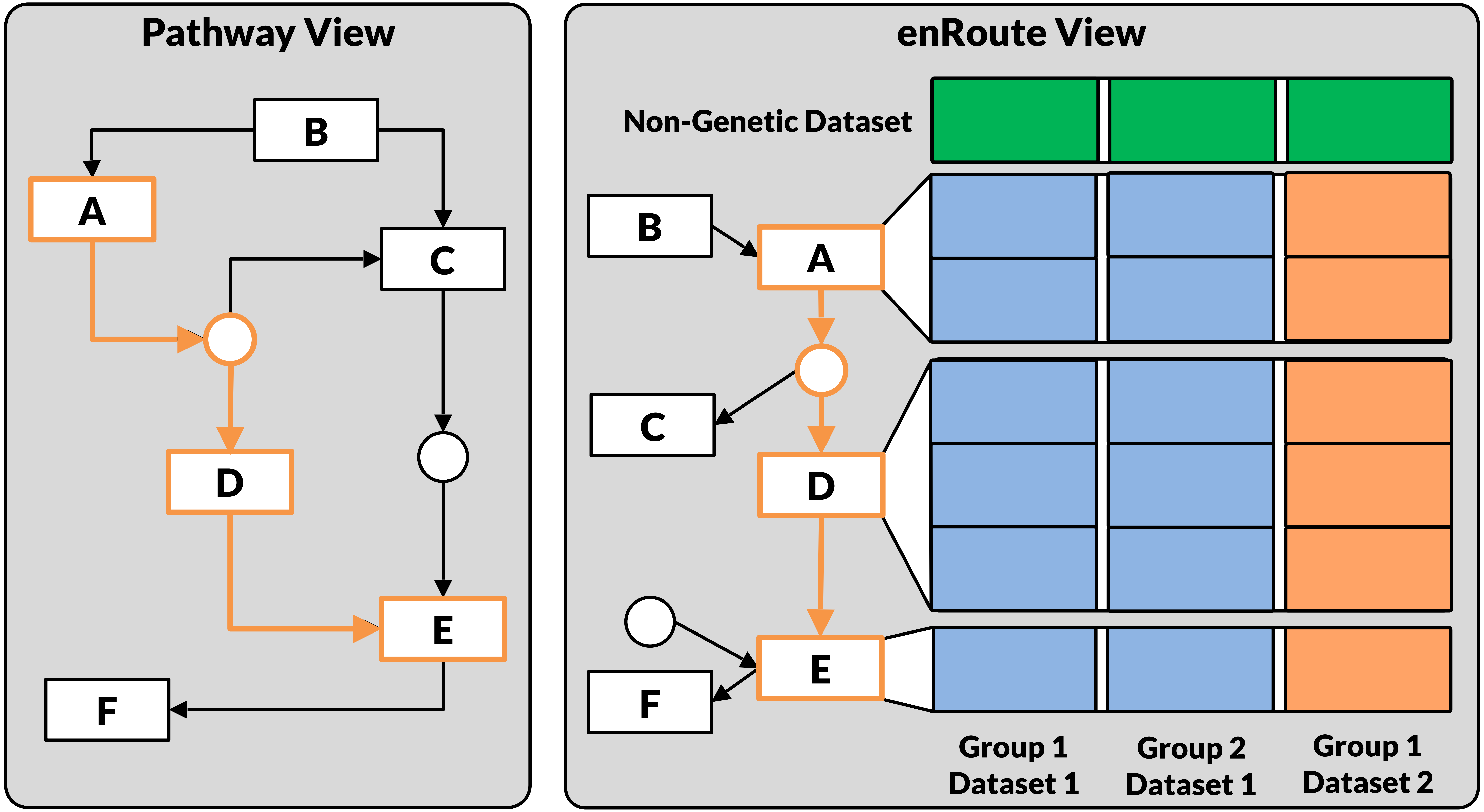
- Explore topology of pathway
- Explore the attributes of the nodes (experimental data)

Need to support both!



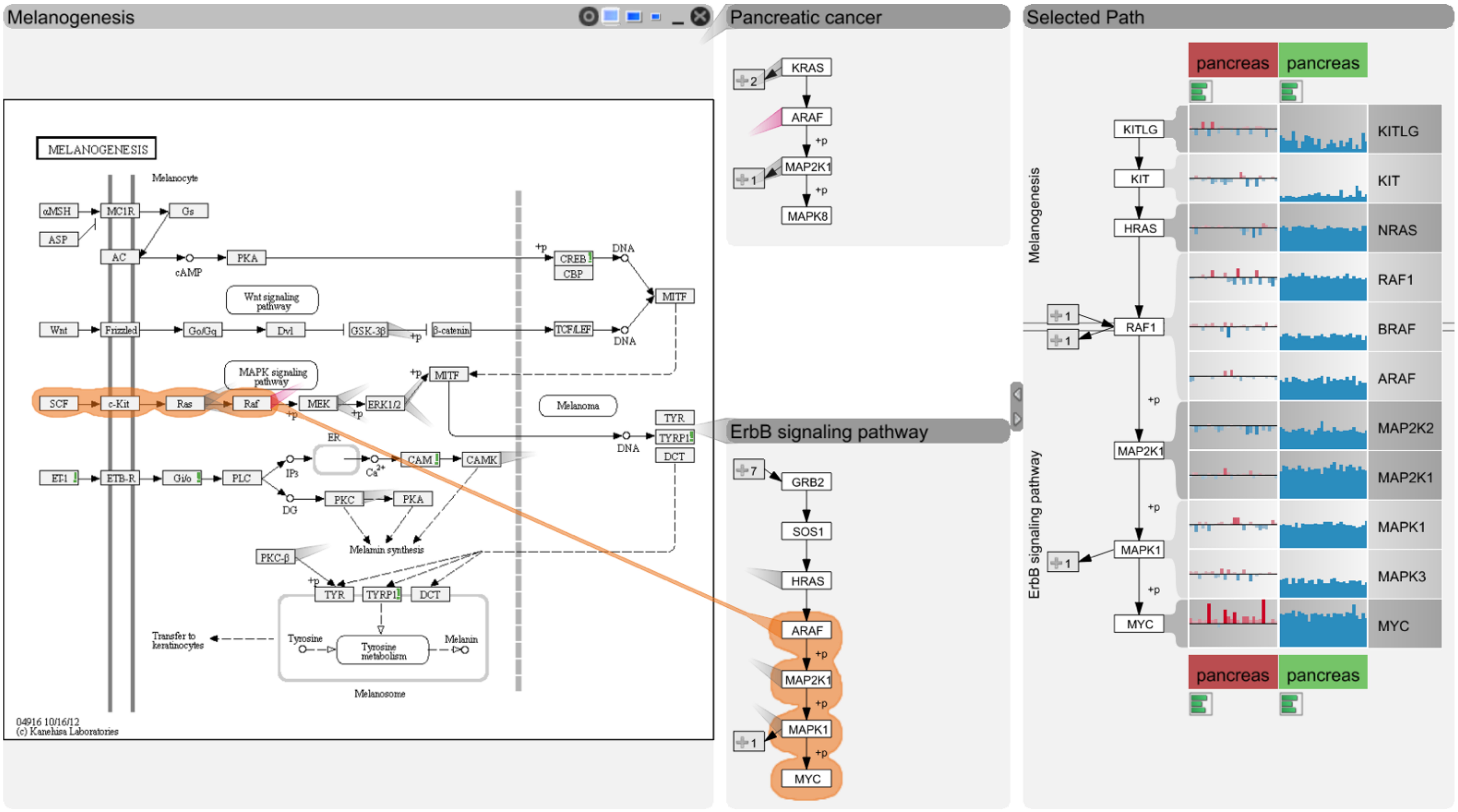
	Sample 1	Sample 2	Sample 3
Gene 1	1	1.1	0.4
Gene 2	2	0.5	1.2
Gene 3	1.4	0.2	0.5
Gene 4	0.3	0.5	0.7

Concept



Pathway View
B
A
C
D
E
F
enRoute View
Non-Genetic Dataset
B
A
C
D
E
F
Group 1
Dataset 1
Group 2
Dataset 1
Group 1
Dataset 2

enRoute



Melanogenesis
Pancreatic cancer
Selected Path
MELANOGENESIS
Melanocyte
αMSH
MC1R
Gs
ASP
AC
cAMP
PKA
CREB
CBP
DNA
MITF
Wnt signaling pathway
Wnt
Frizzled
Go/Gq
Dvl
GSK-3β
β-catenin
TCF/LEF
MAPK signaling pathway
SCF
c-Kit
Ras
Raf
MEK
ERK1/2
Melanoma
TYR
TYRP1
DCT
ER
ET-1
ETB-R
Gi/o
PLC
IP3
Ca2+
CAM
CAMK
DG
PKC
PKC-β
Melanin synthesis
Transfer to keratinocytes
Tyrosine
Tyrosine metabolism
Melanin
Melanosome
04916 10/16/12
(c) Kanehisa Laboratories
KRAS
ARAF
MAP2K1
MAPK8
ErbB signaling pathway
GRB2
SOS1
HRAS
MAPK1
MYC
pancreas
KITLG
KIT
NRAS
RAF1
BRAF
MAP2K2
MAPK3
ErbB signaling pathway

File Data Window View Help

Entourage

Pathways

Pathway

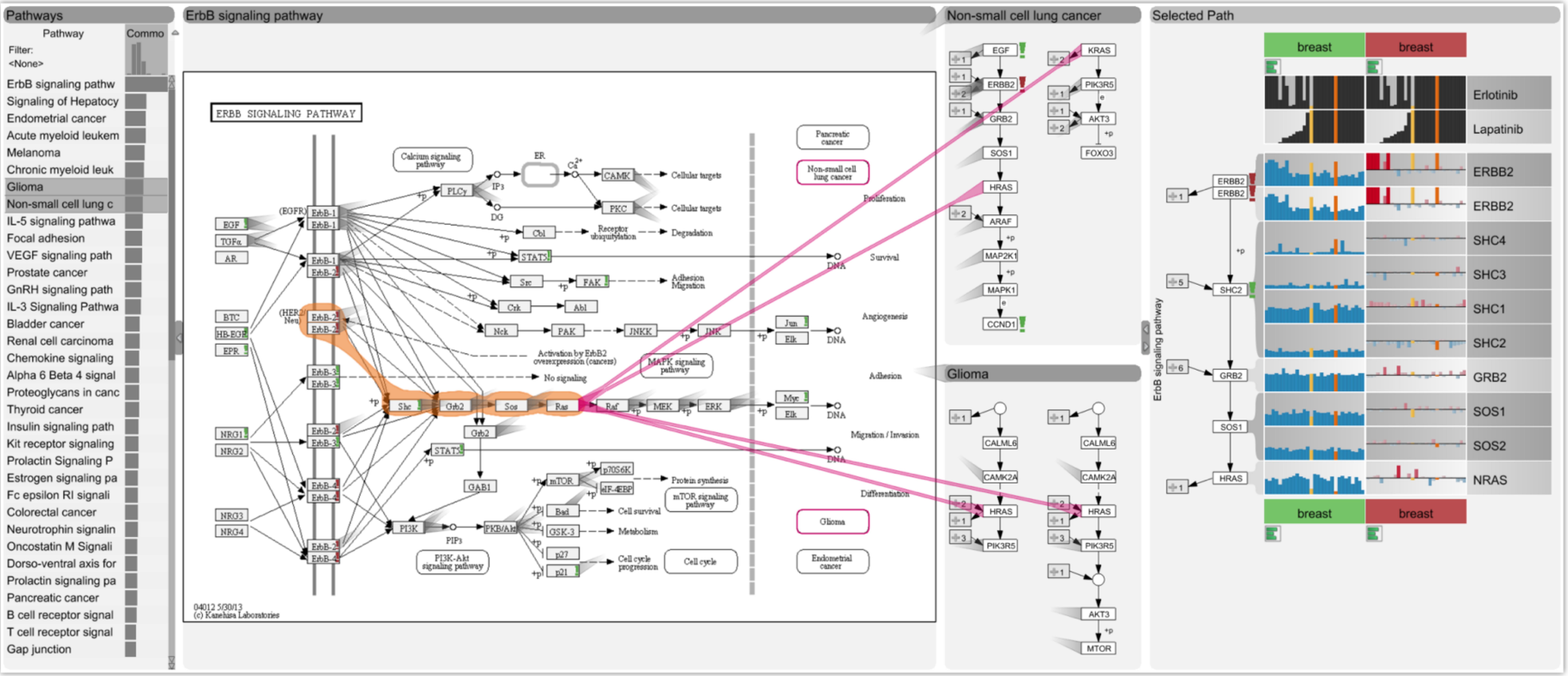
Filter:
<None>

1 C donor
2-Oxocarboxylic acid
ABC transporters
ABC-family proteins
ACE Inhibitor Pathwa
Acetylcholine Synthes
Acute myeloid leukem
Adherens junction
Adipocyte TarBase
Adipocytokine signali
Adipogenesis
Advanced glycosylatio
Aflatoxin B1 metaboli
African trypanosomias
AGE/RAGE pathway
AhR pathway
Alanine and aspartate
Alanine, aspartate an
Alcoholism
Aldosterone-regulated
Allograft rejection
Allograft rejection
Alpha 6 Beta 4 signal
alpha-Linolenic acid
Alzheimer's disease
Alzheimers Disease
amino acid conjugatio
amino acid conjugatio
Amino sugar and nucl
Aminoacyl-tRNA bios
Amoebiasis
Amphetamine addicti
AMPK signaling
Amyotrophic lateral sc
Androgen receptor si
Angiogenesis
Angiogenesis
angiogenesis overvie
Antigen processing an
APC/C-mediated degra
Apoptosis
Apoptosis
Apoptosis Meta Path
Apoptosis Modulation
Apoptosis Modulation
Apoptosis, anoikis an

Selected Path

Toggle summary button
Summary of all dimensions
All dimensions shown separately
Not Mutated
Mutated
Not Mutated
Mutated
Race
Days to Deat
Gender
Context (non-genetic) datasets
EGF
EGFR
PLCG1
PLCG2
PRKCB1
PRKCA
PRKCG
NRAS
Dimensions sorted by this row
Multi-mapping of gene family
Glioma
Collapsed incoming node
Expanded outgoing nodes
ARAF
PIK3R5
HRAS
Not Mutated
Mutated
Not Mutated
Mutated
Groups of same dataset
Different Datsets

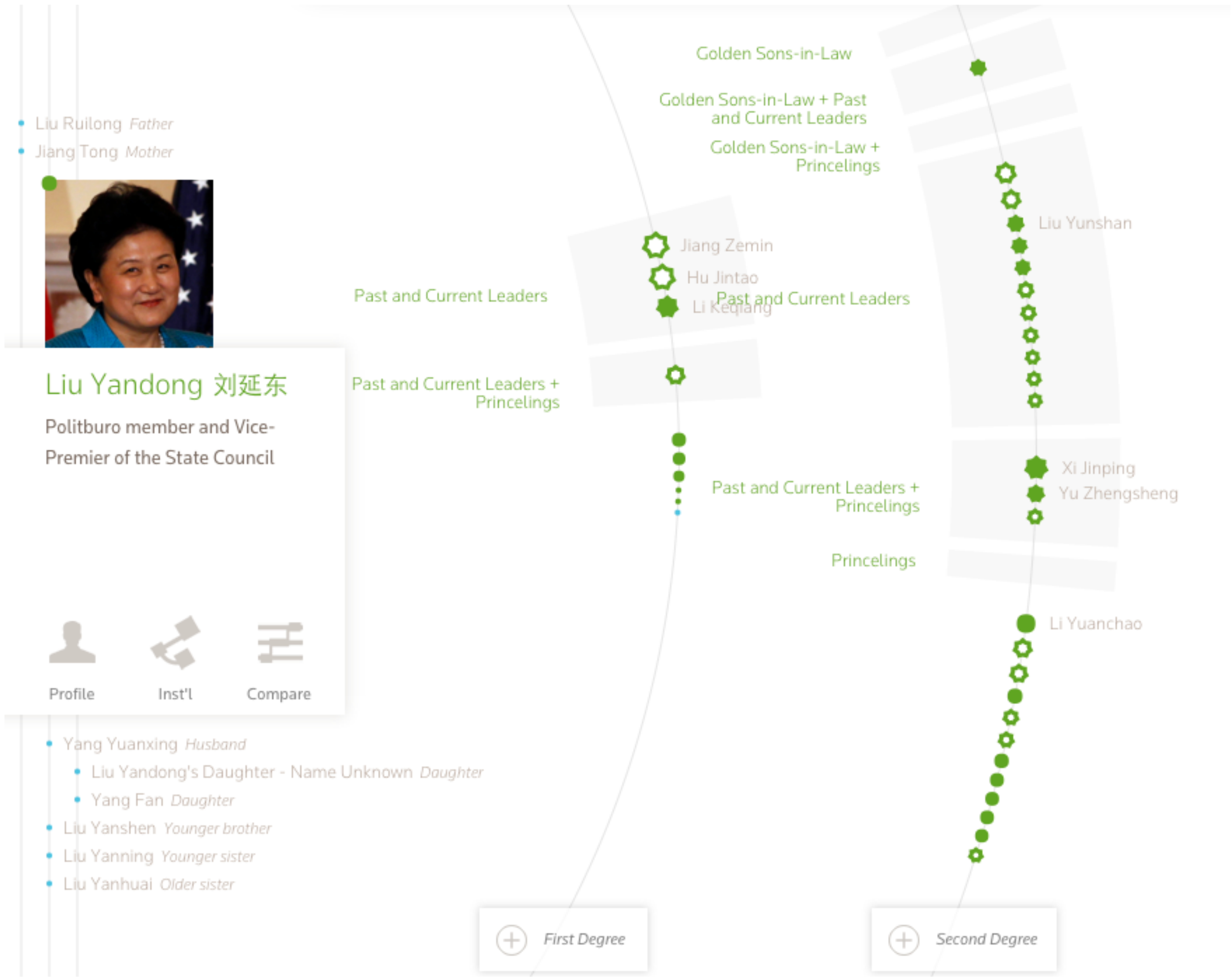
Case Study: CCLE Data



Pancreatic cancer
Non-small cell lung cancer
Proliferation
Survival
Angiogenesis
Adhesion
Migration / Invasion
Differentiation
Glioma
Jun
Elk
Myc
Elk
DNA
DNA
DNA
DNA
+p
+p
ERBB2
GRB2
SOS1
HRAS
ARAF
MAP2K1
MAPK1
CCND1
PIK3R5
AKT3
FOXO3
Glioma
CALML6
CAMK2A
HRAS
CALML6
CAMK2A
HRAS
ErbB signaling pathway
ERBB2
ERBB2
SHC2
GRB2
SOS1
HRAS
Erlotinib
Lapatinib
ERBB2
ERBB2
SHC4
SHC3
SHC1
SHC2
GRB2
SOS1
SOS2
NRAS
breast
breast

Design Critique

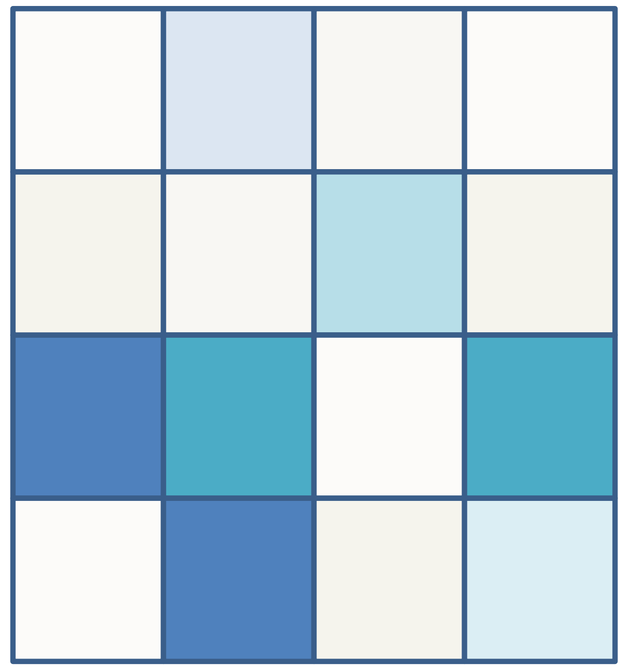
Connected China



https://goo.gl/YXkWYX

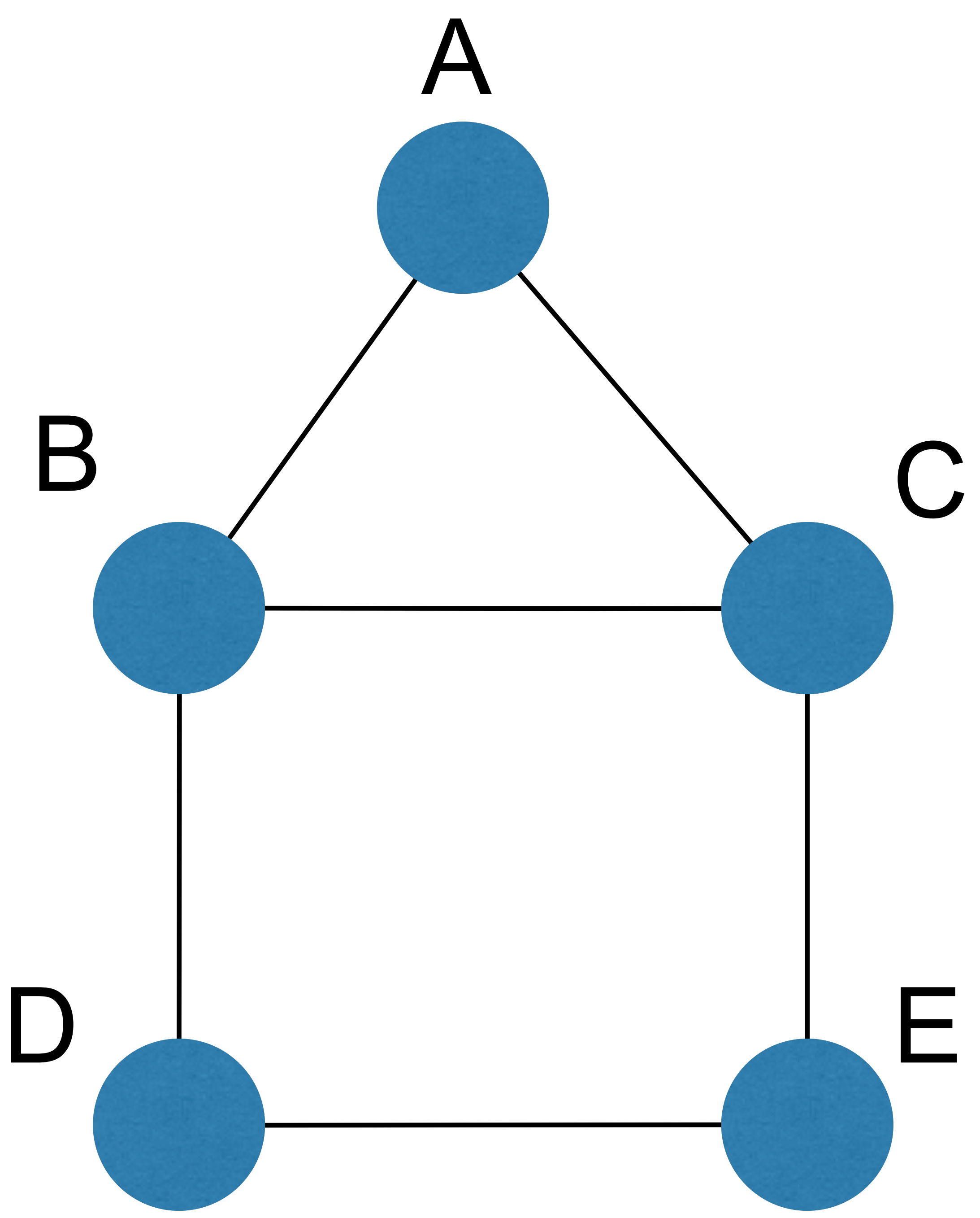
http://china.fathom.info/

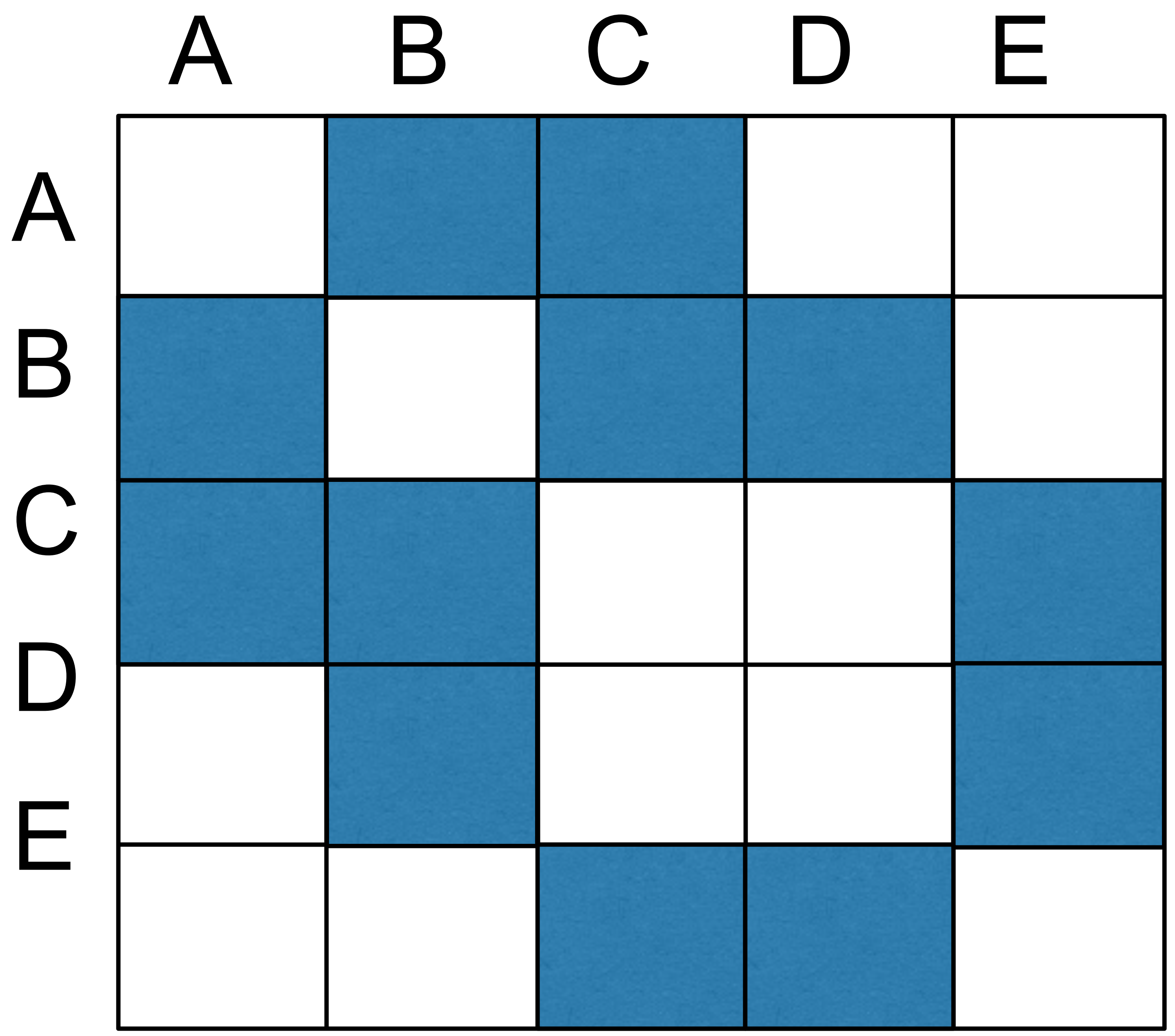
Matrix Representations



Matrix Representations

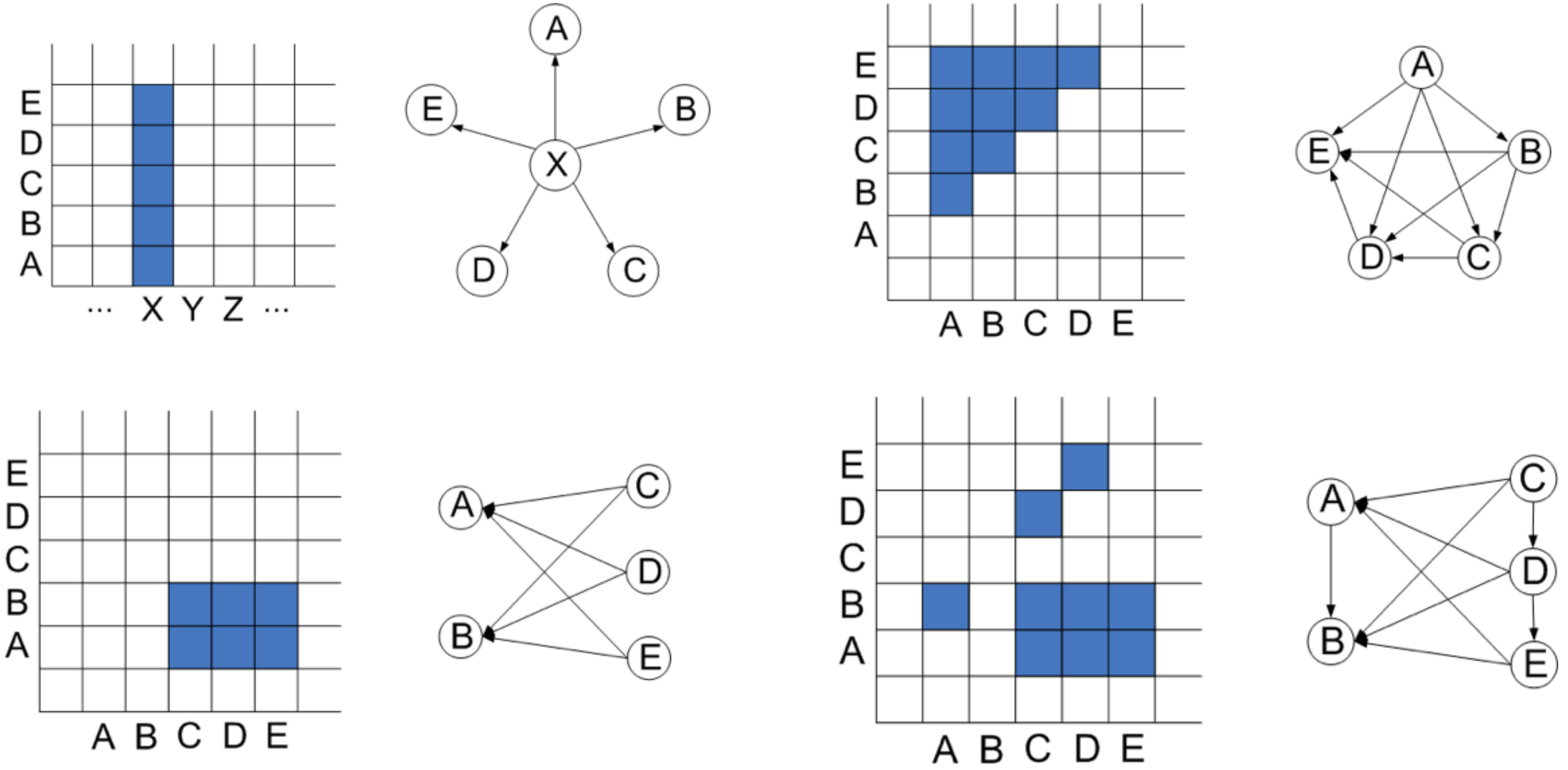
Instead of node link diagram, use adjacency matrix



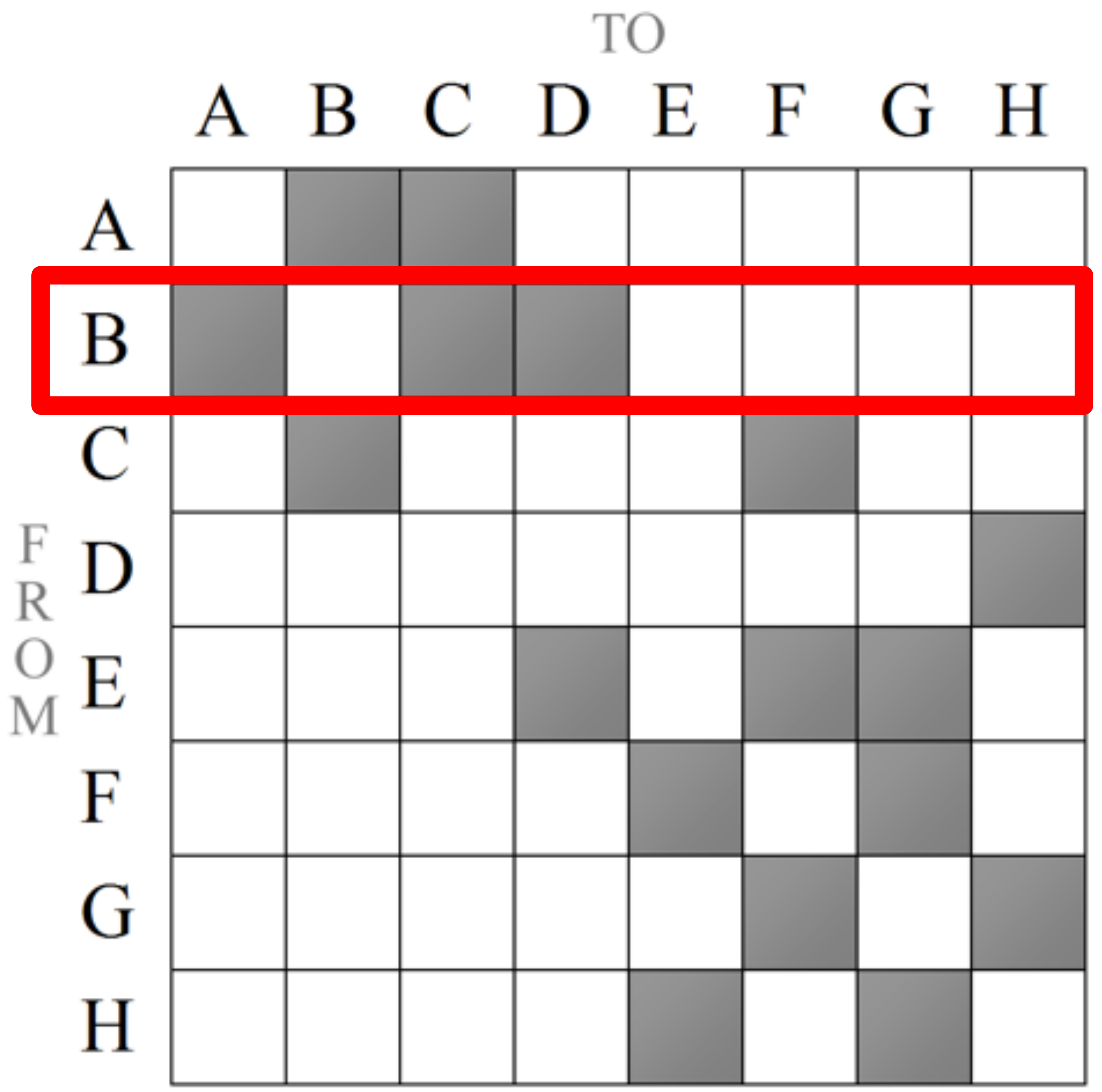


Matrix Representations

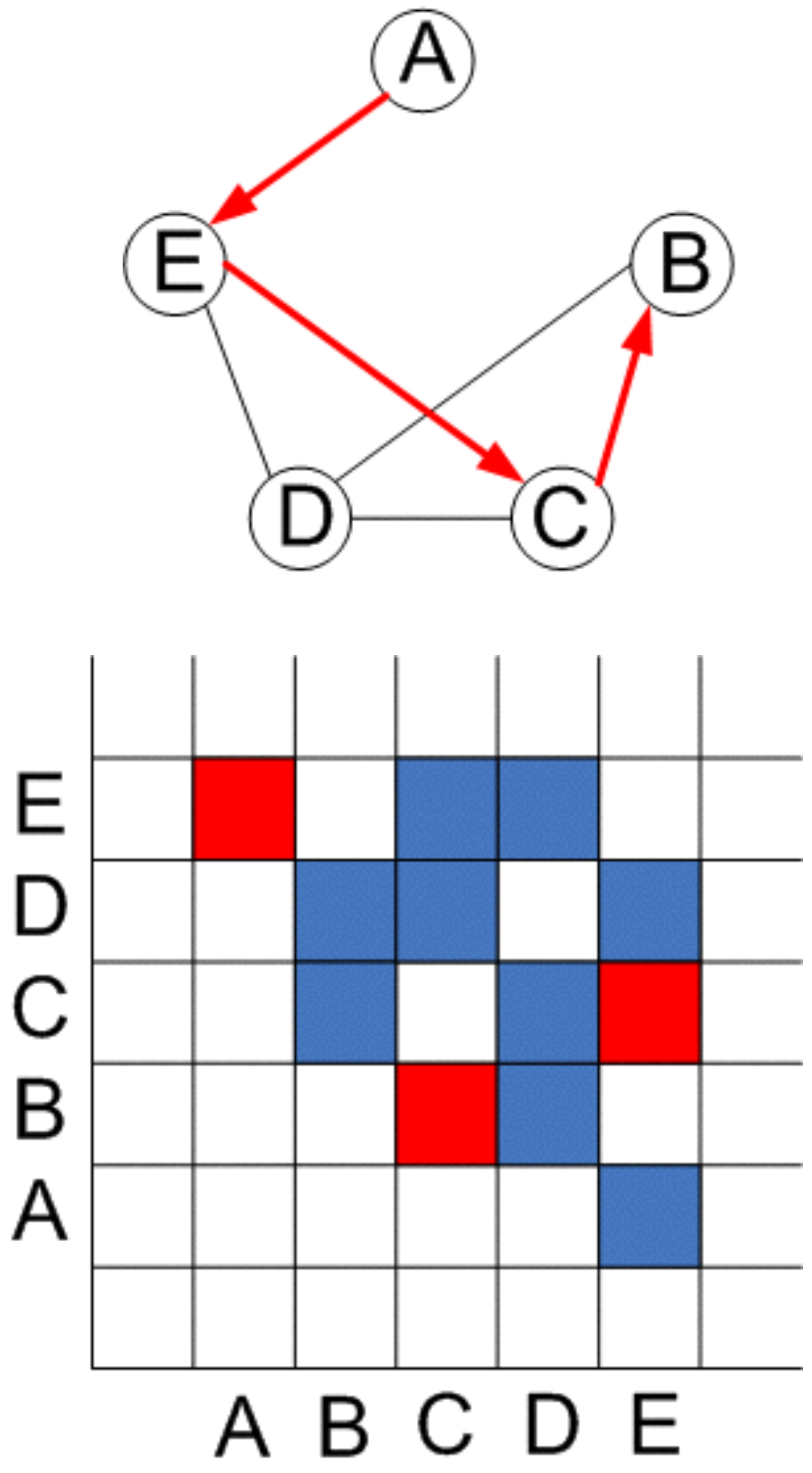
Examples:



Matrix Representations

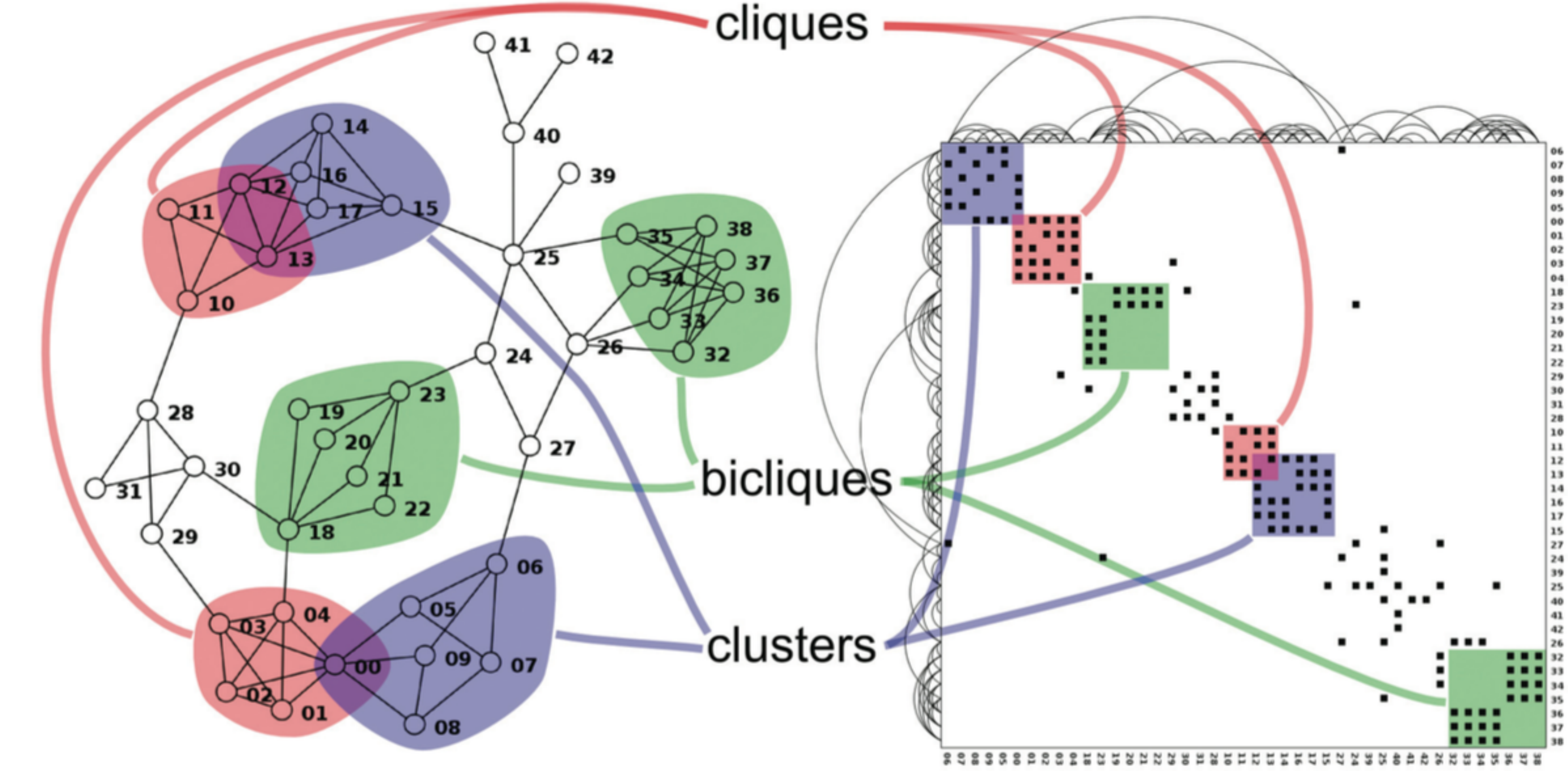


Well suited for
neighborhood-related TBTs



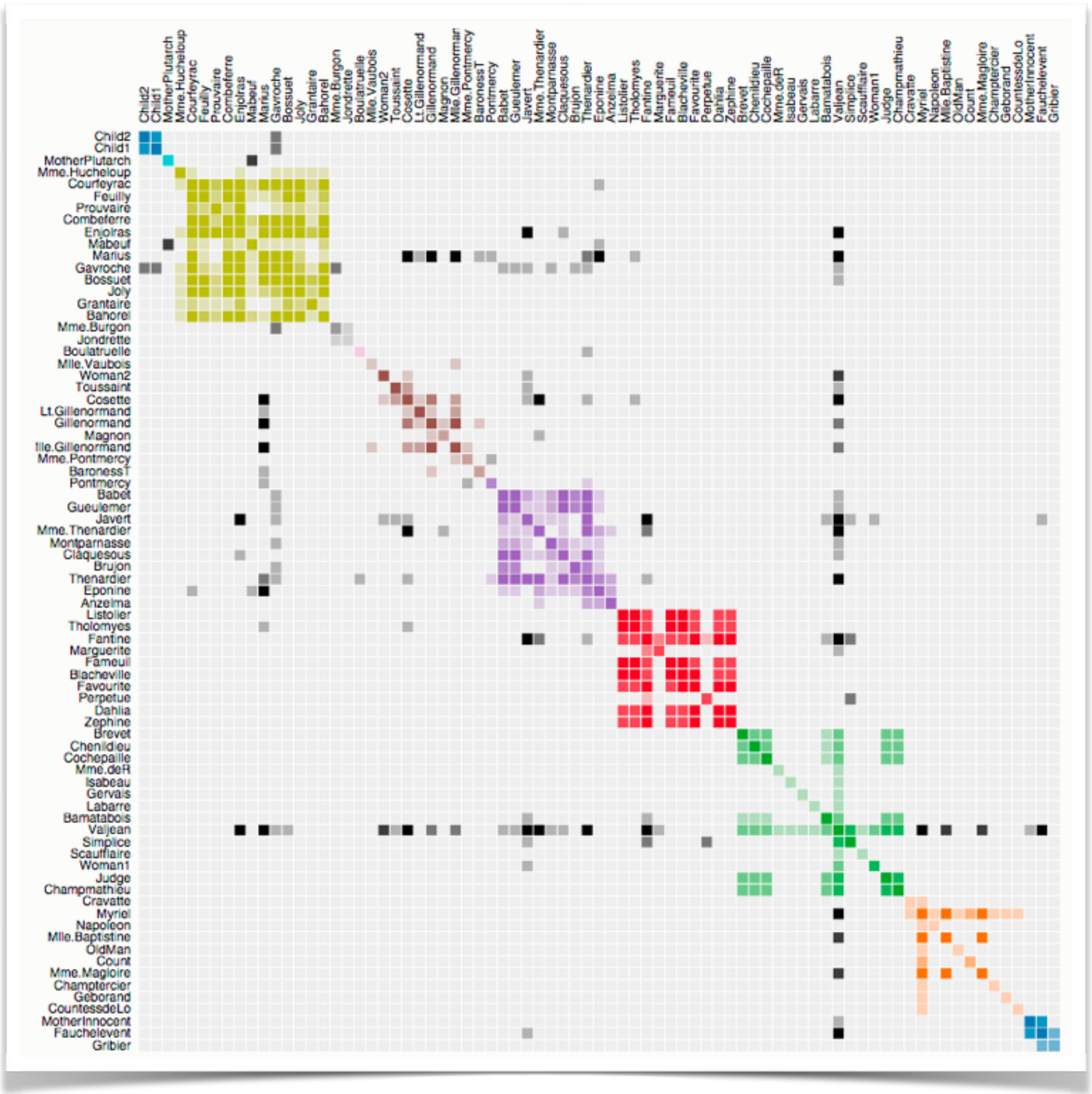
Not suited for
path-related TBTs

van Ham et al. 2009
Shen et al. 2007



McGuffin 2012

Order Critical!



Matrix Representations

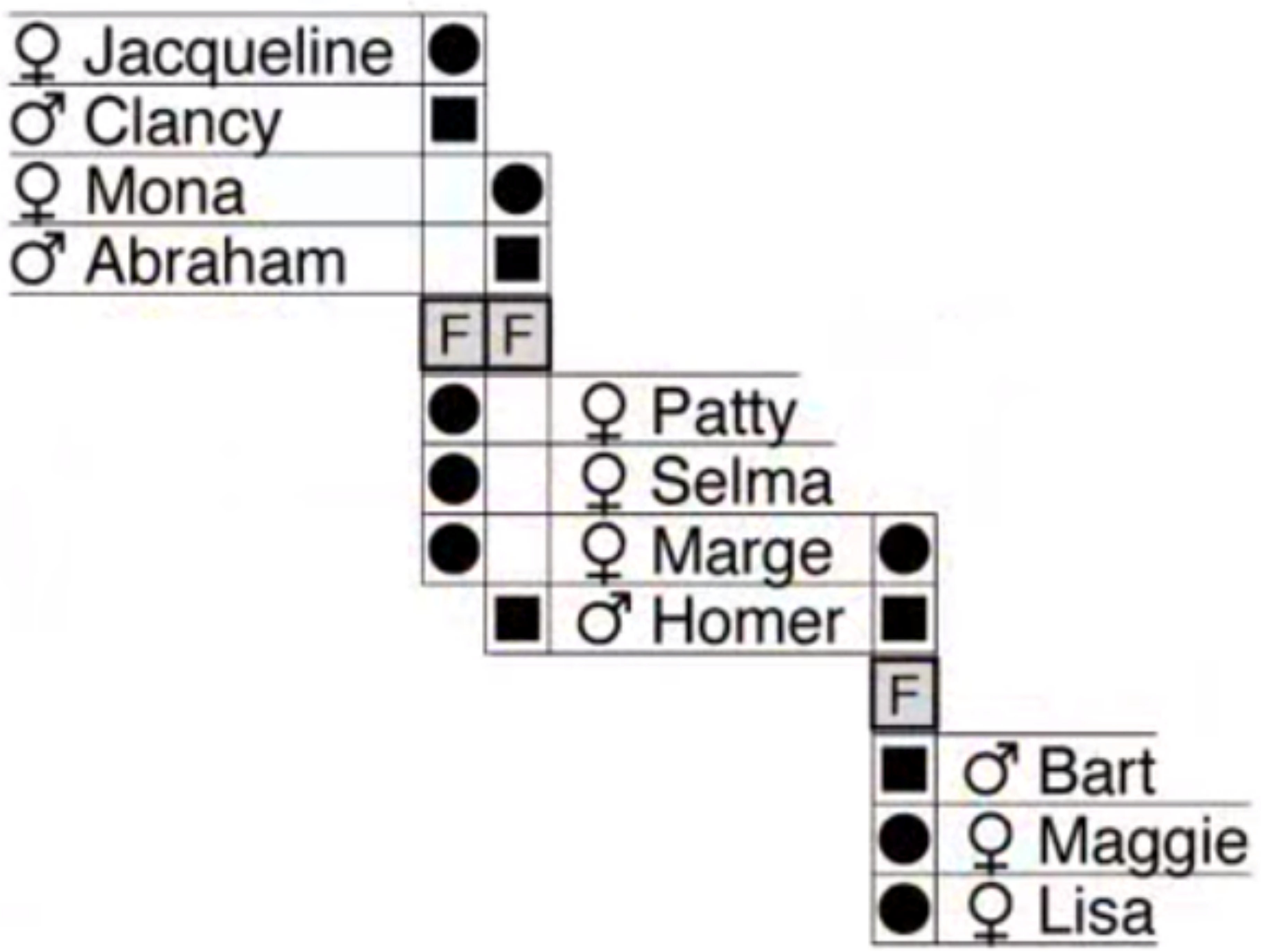
Pros:

- 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

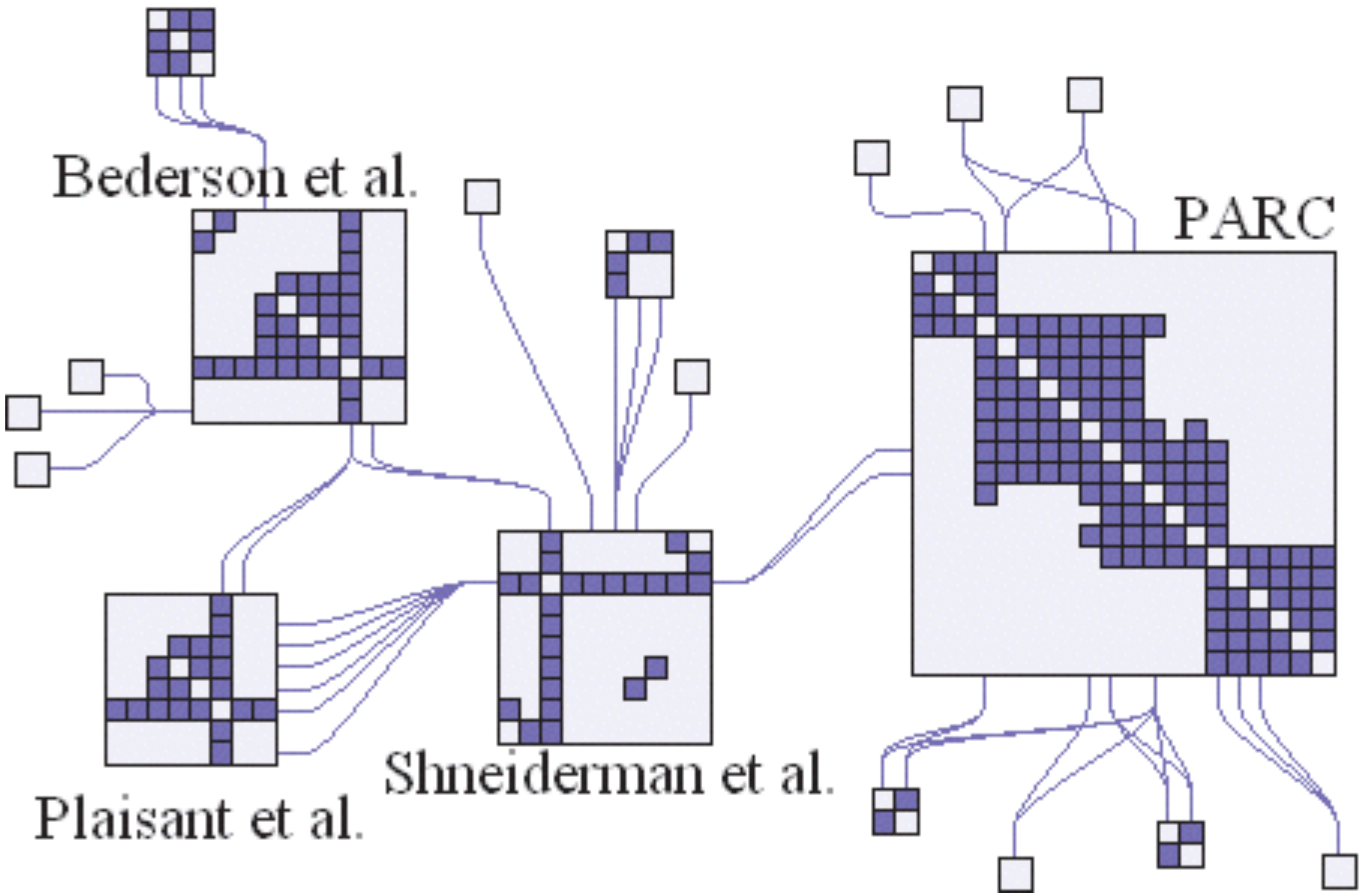
Cons:

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

Special Case: Genealogy

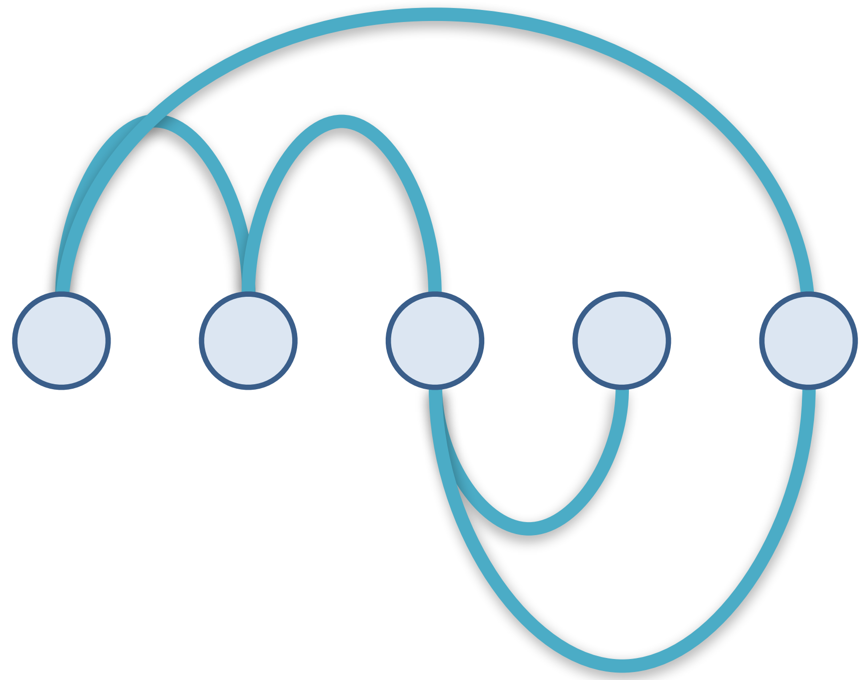


Hybrid Explicit/Matrix

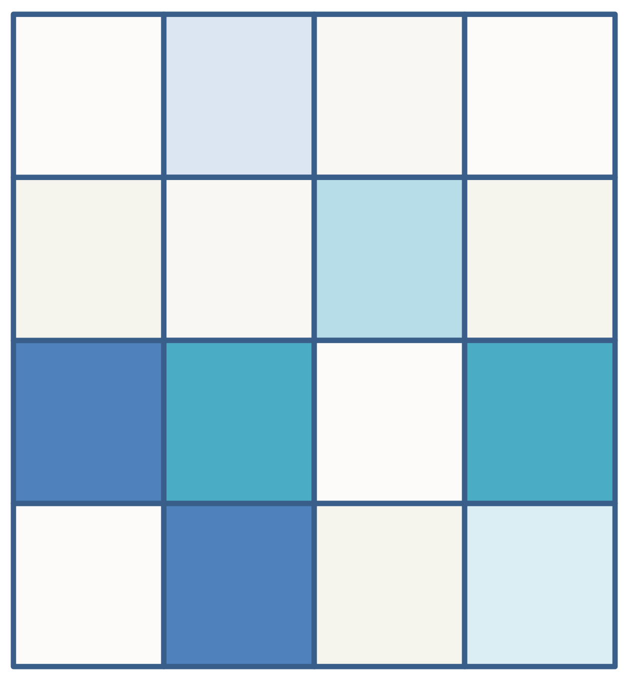


NodeTrix
[Henry et al. 2007]

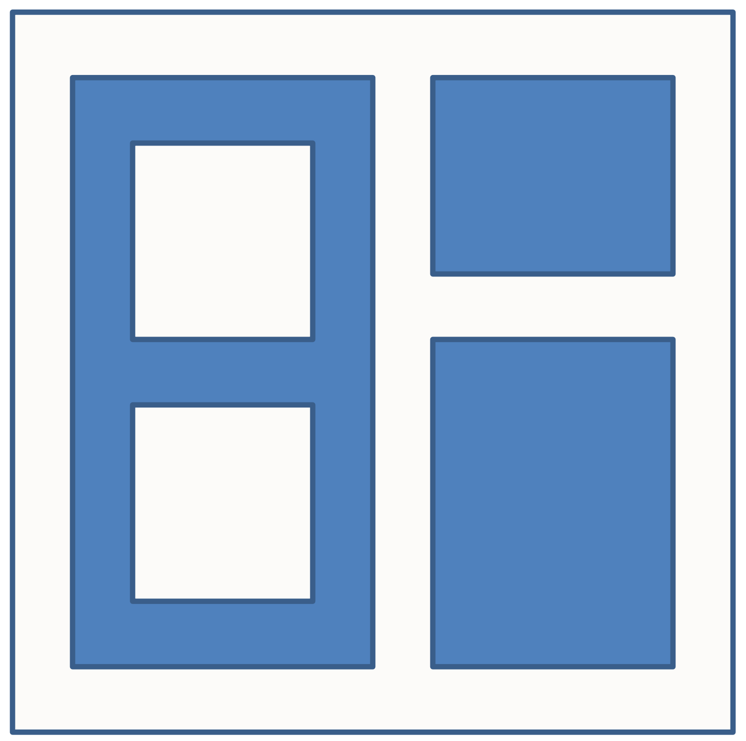
Implicit Layouts



Explicit
(Node-Link)



Matrix



Implicit

Explicit vs. Implicit Tree Vis

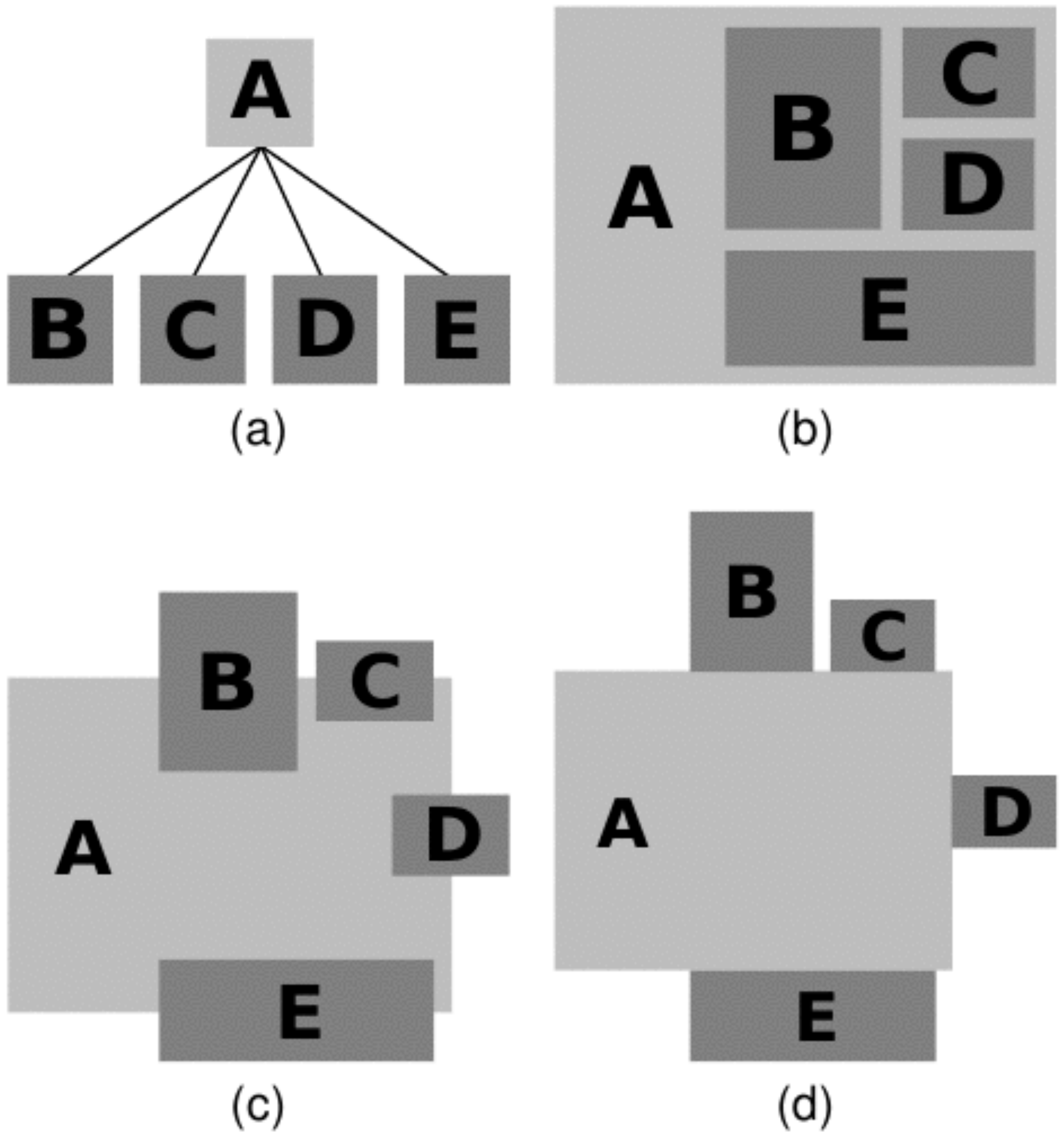
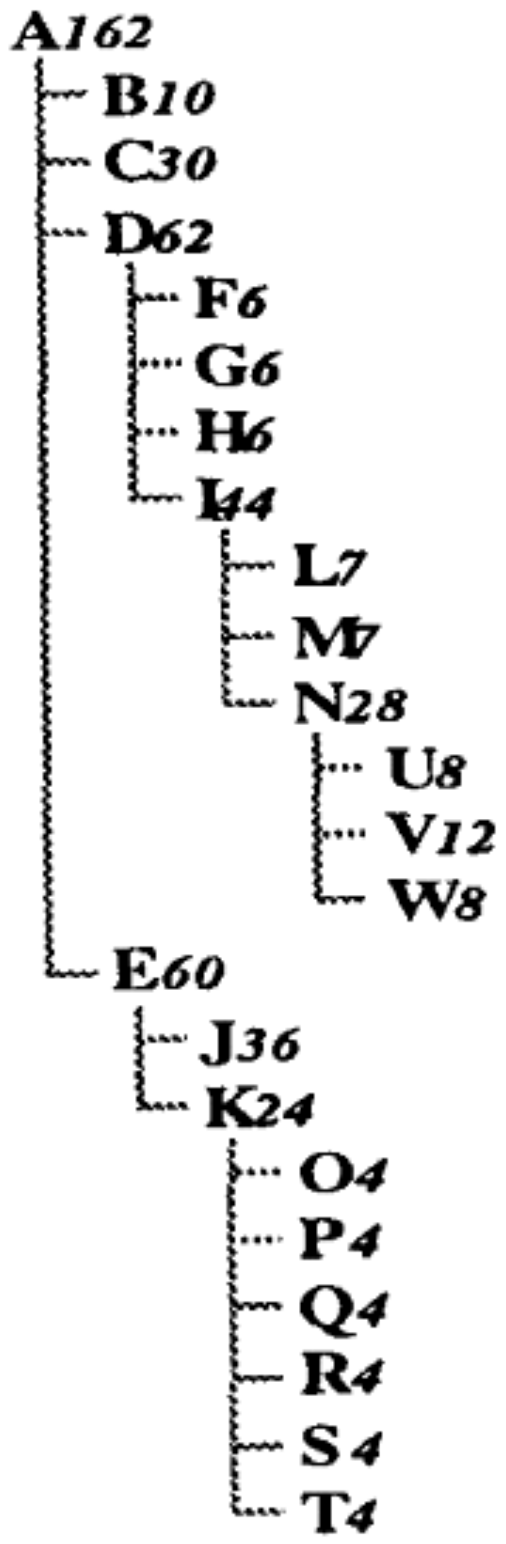
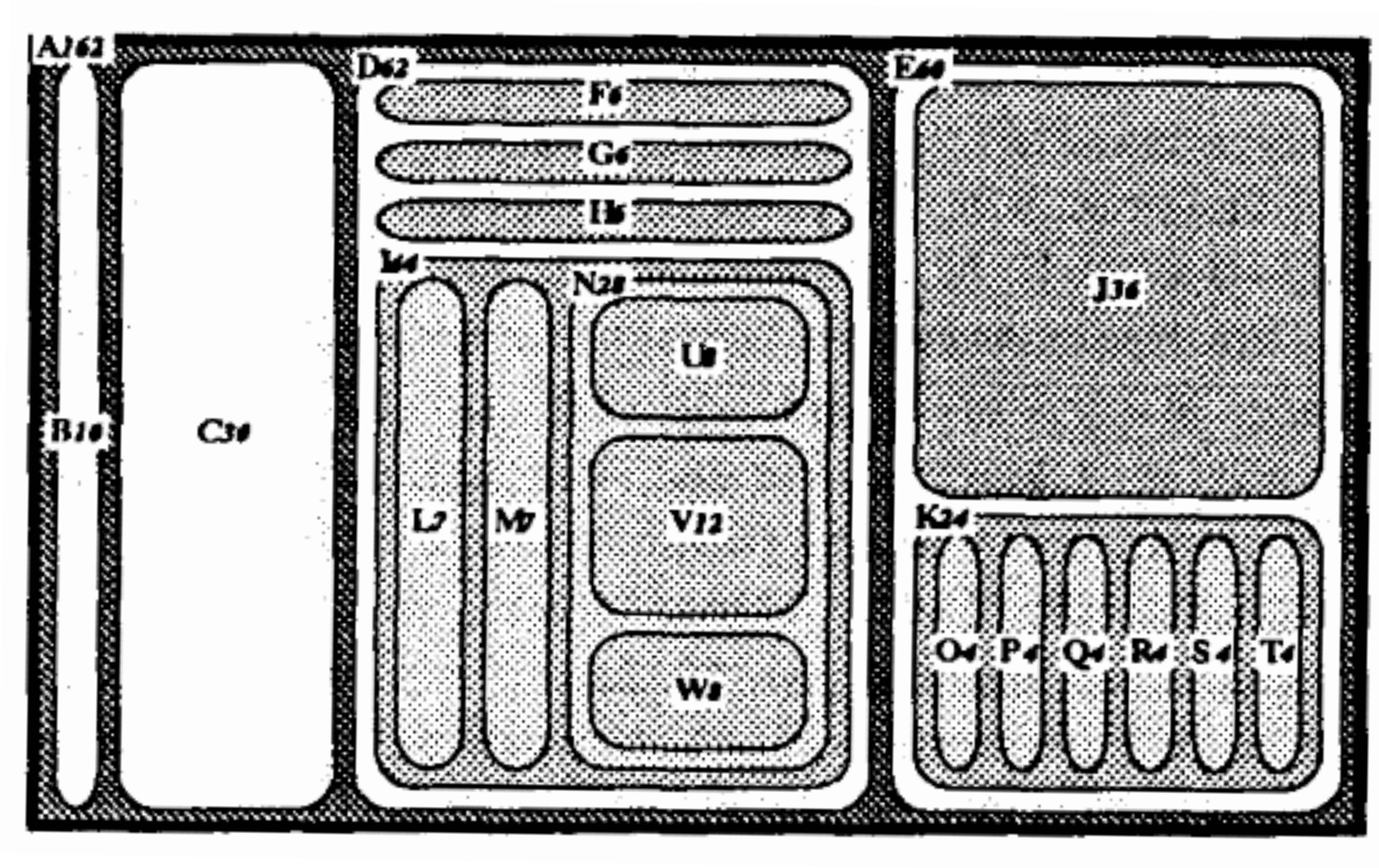


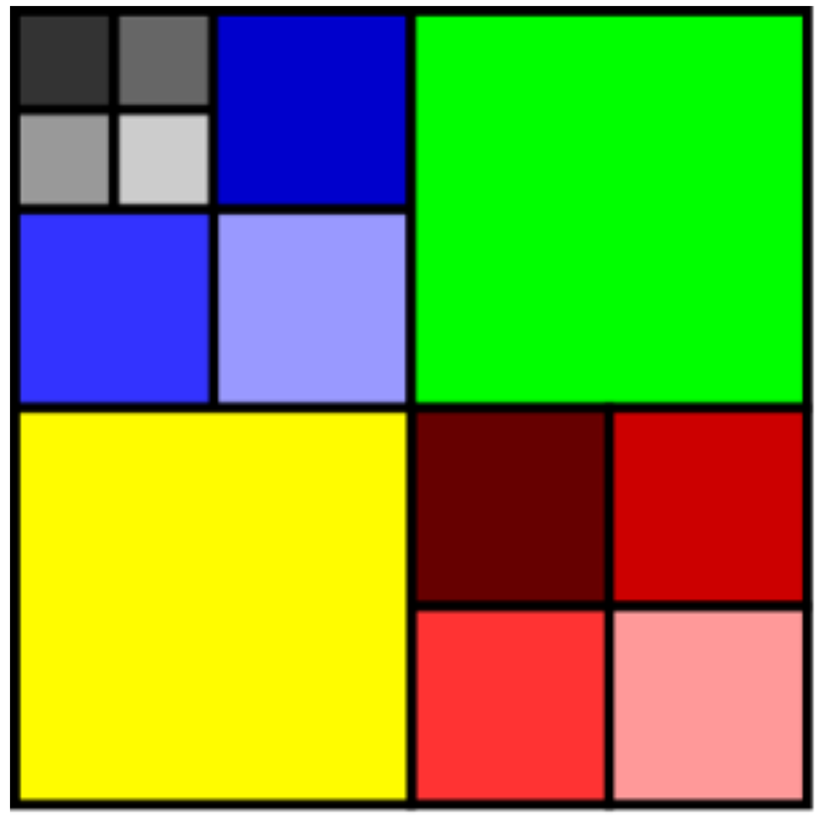
Fig. 2. (a) Explicit, node-link layout, (b) Implicit layout by inclusion, (c) Implicit Layout by overlap, (d) Implicit layout by adjacency.

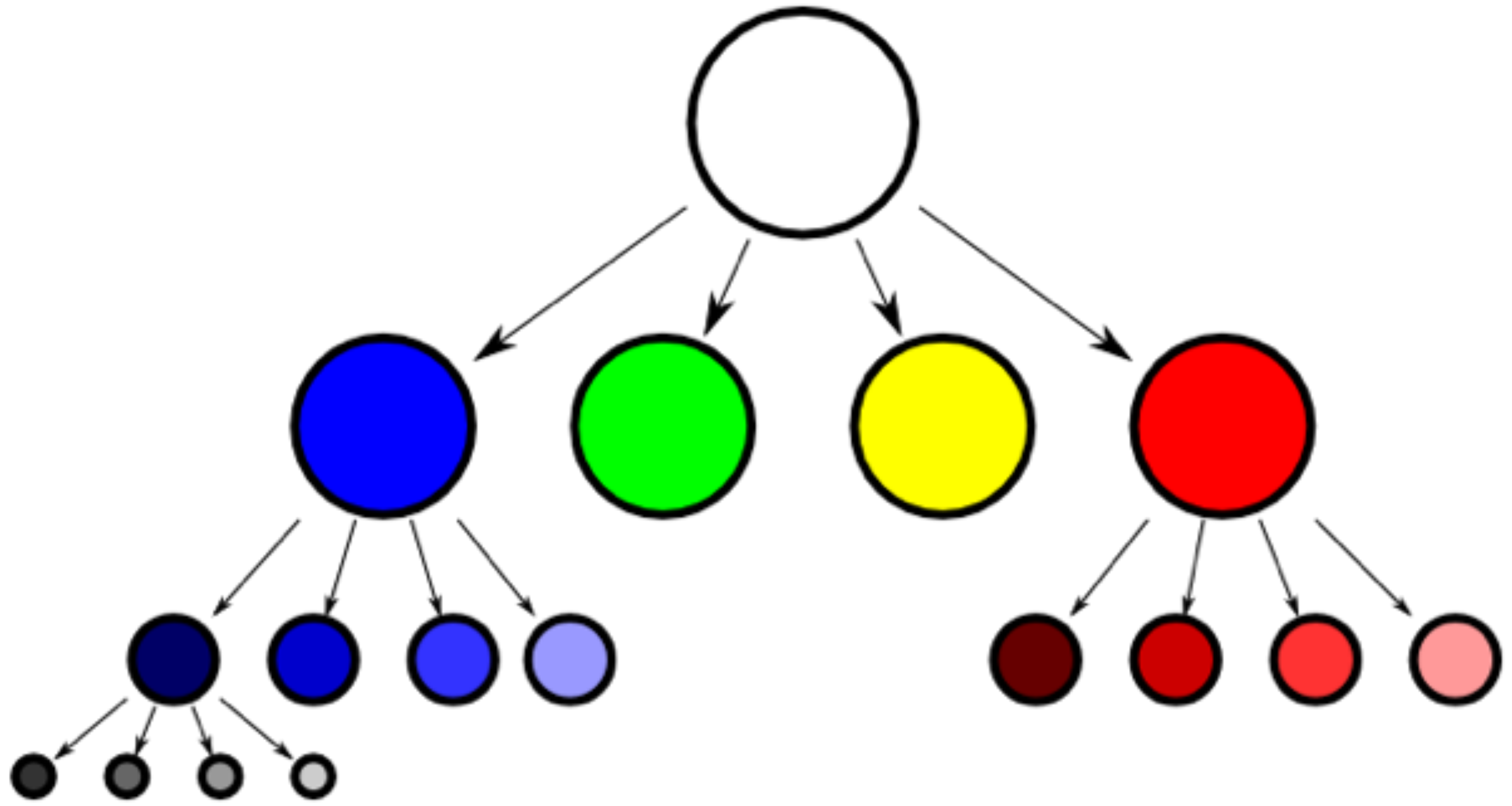
Schulz 2011

Tree Maps



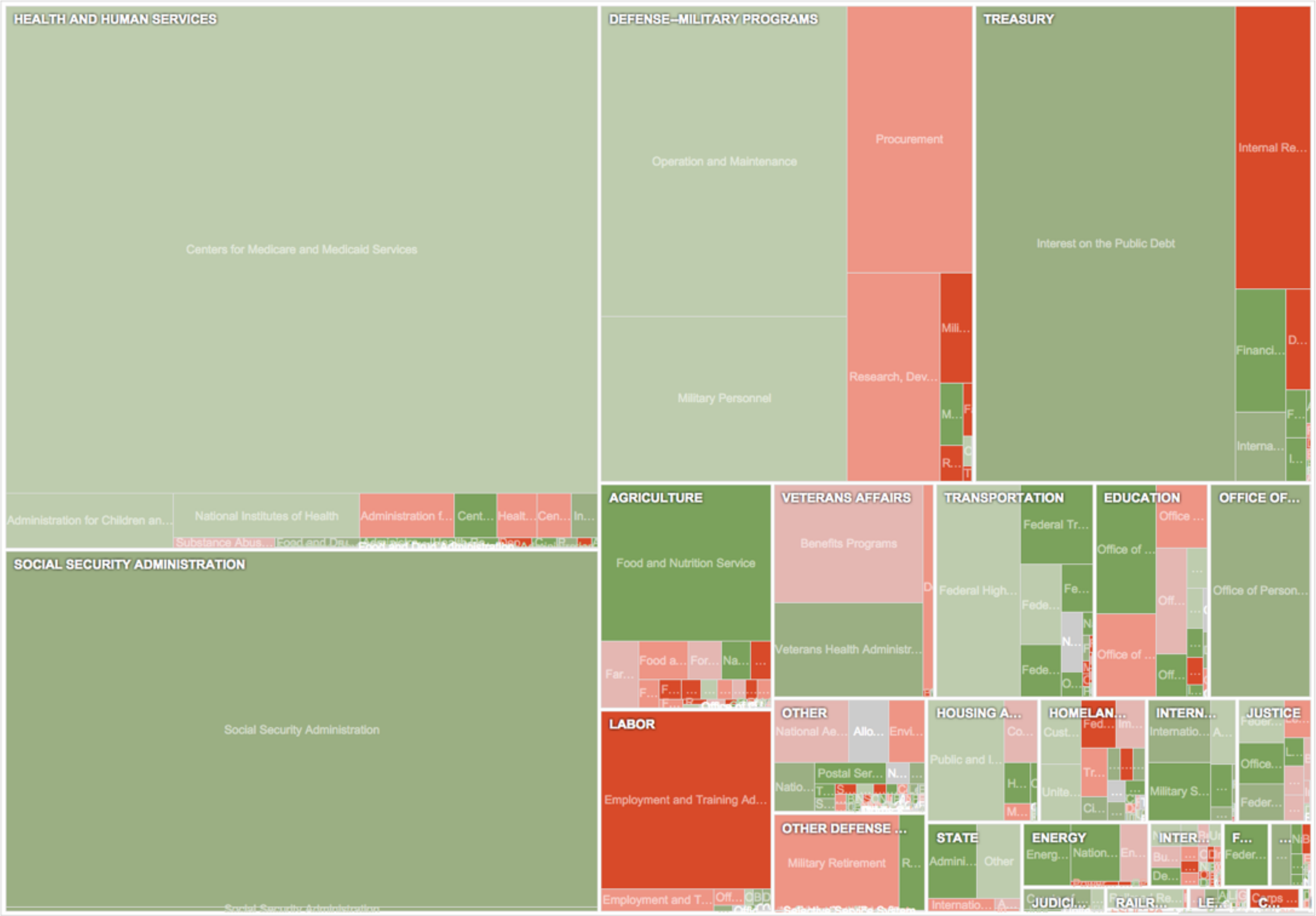




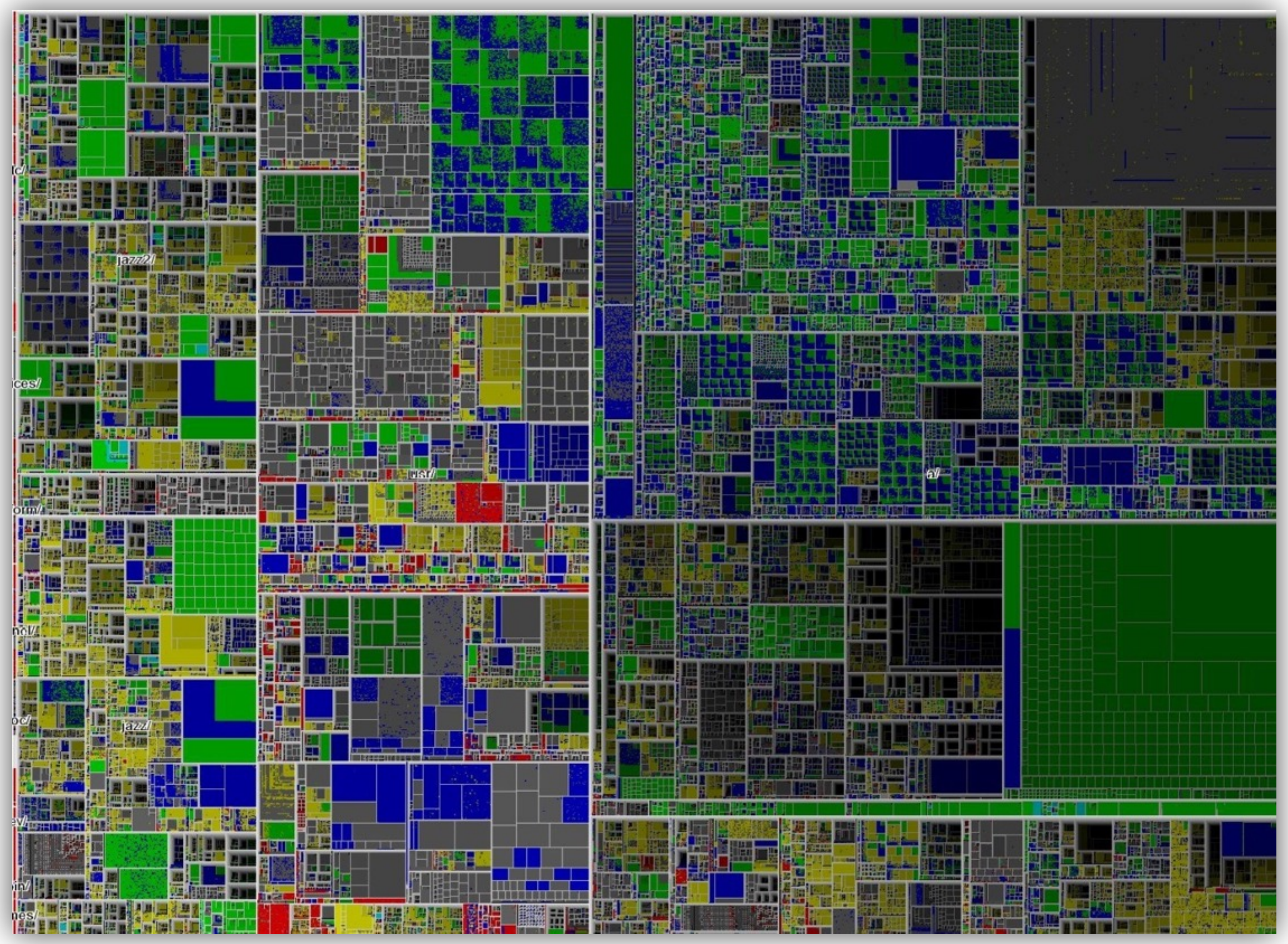


Johnson and Shneiderman 1991

Zoomable Treemap



Example: Interactive TreeMap of a Million Items

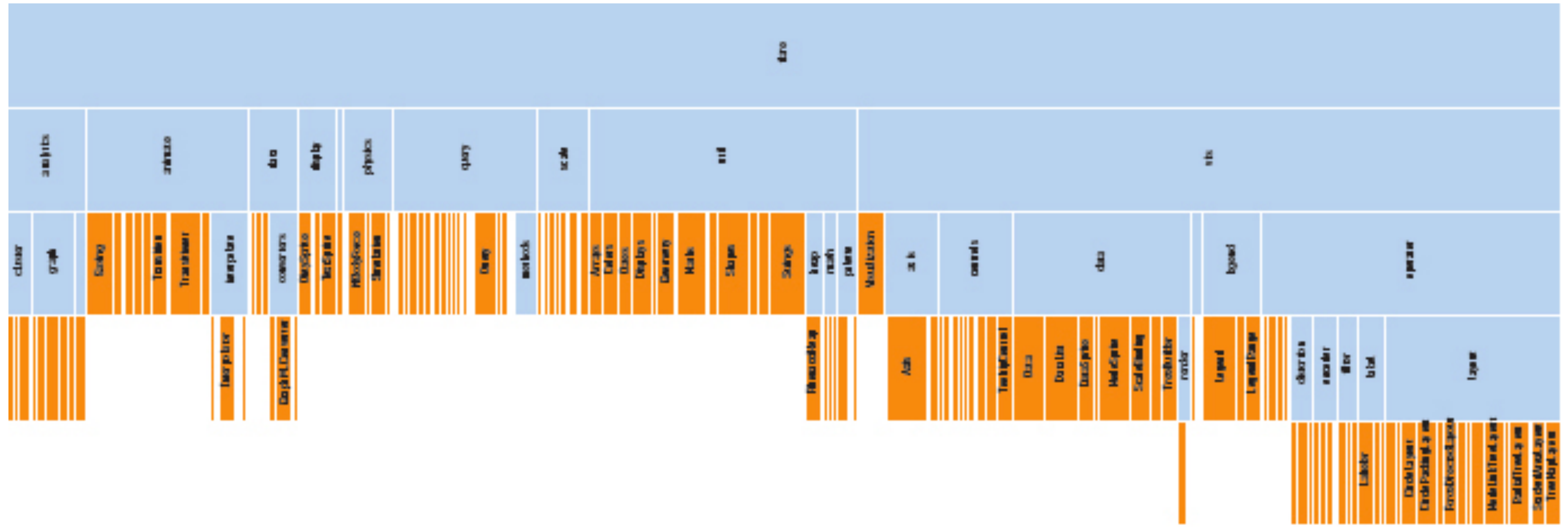


Fekete et al. 2002

Sunburst: Radial Layout

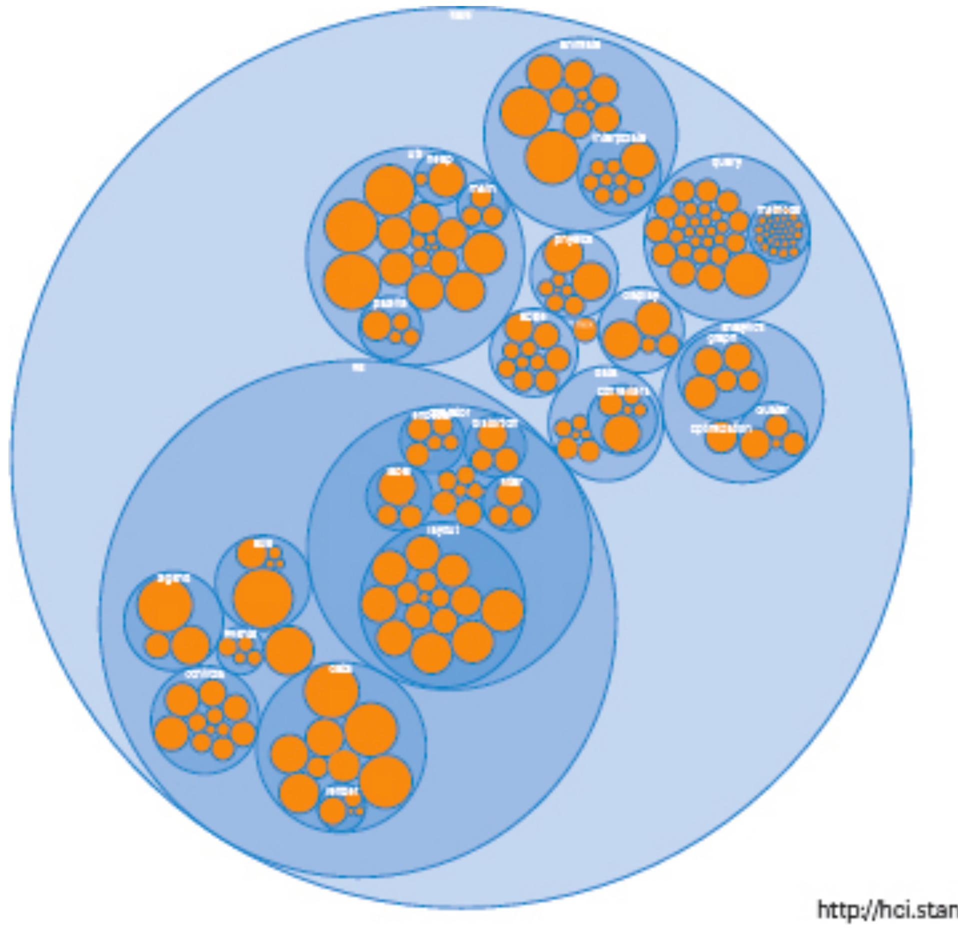
[Sunburst by John Stasko, Implementation in Caleydo by Christian Partl]

Others

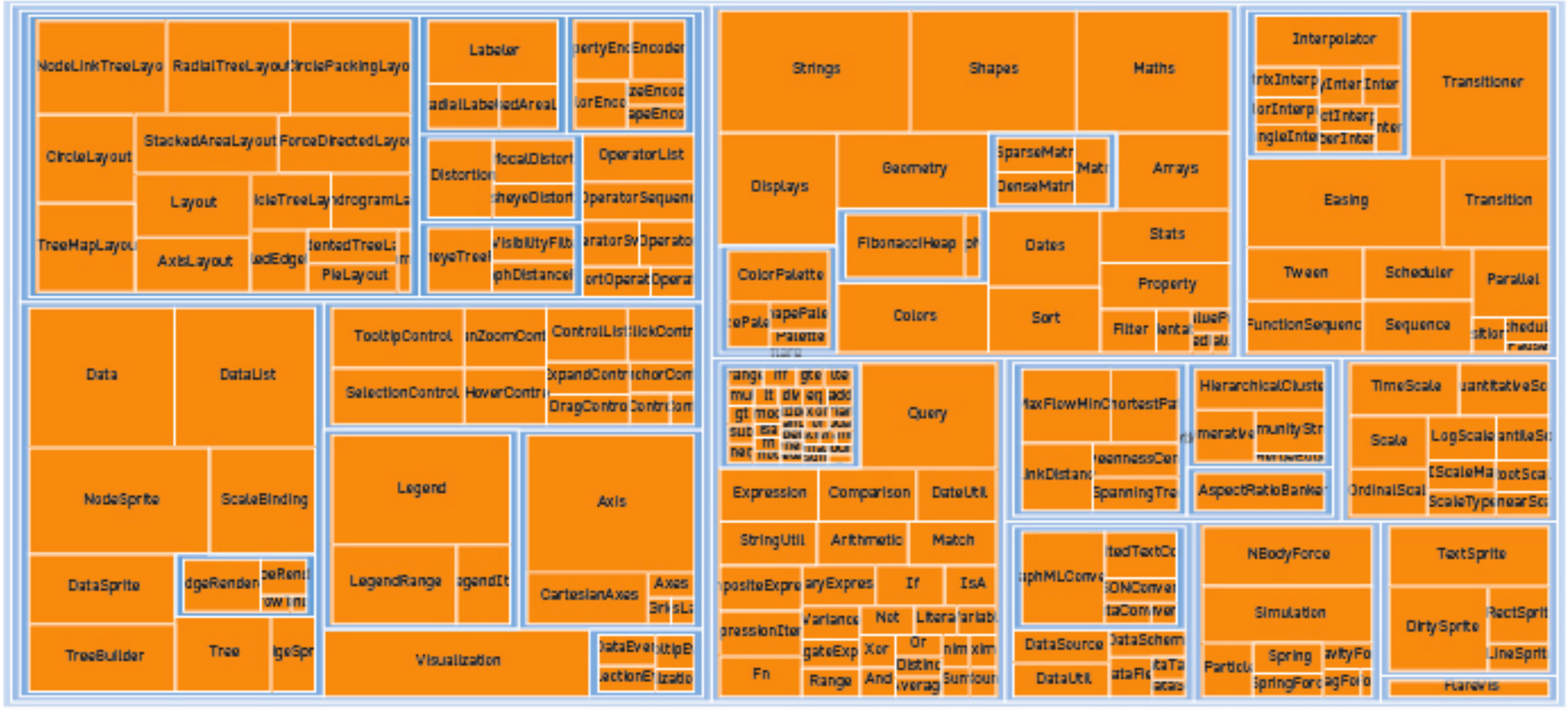


Icicle Plot

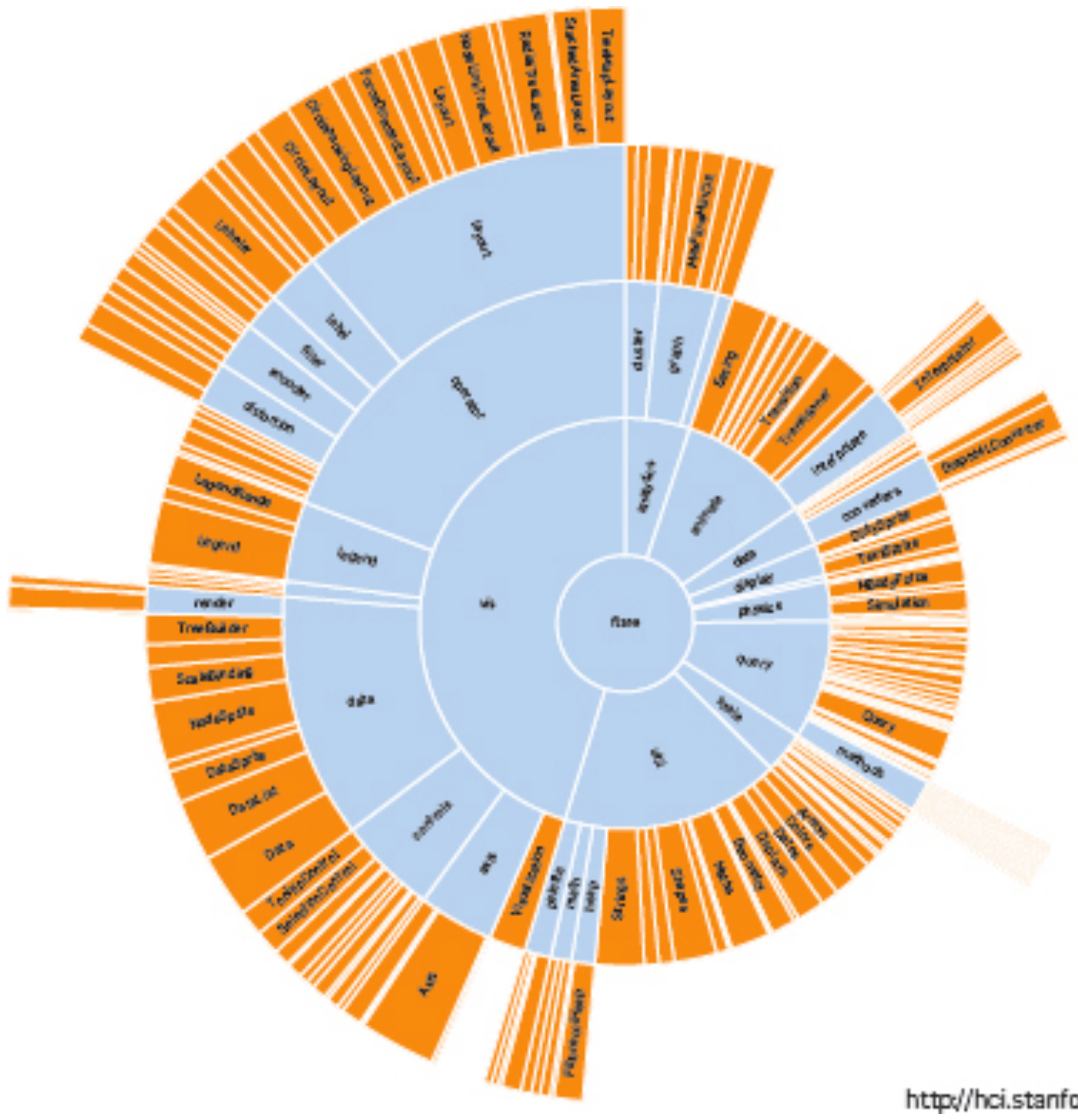
http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/icicle.html



http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/pack.html
Source: The Flare Toolkit http://flare.prefuse.org



http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/treemap.html



http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/sunburst.html

Implicit Representations

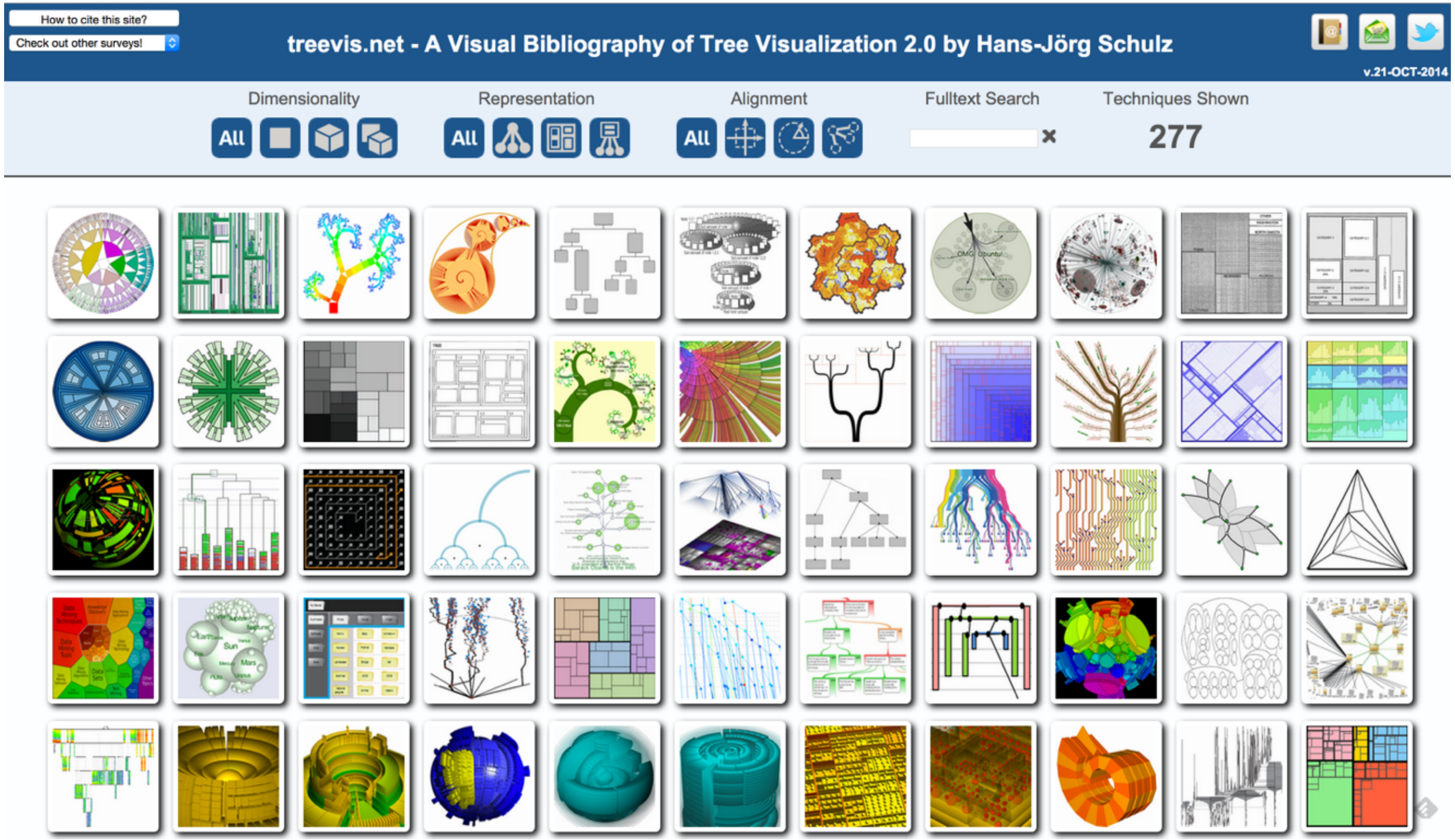
Pros:

- space-efficient because of the lack of explicitly drawn edges: scale well up to very large graphs
- in most cases well suited for ABTs on the node set
- depending on the spatial encoding also useful for TBTs

Cons:

- can only represent trees
- since the node positions are used to represent edges, they can no longer be freely arranged (e.g., to reflect geographical positions)
- useless to pursue any task on the edges
- spatial relations such as overlap or inclusion lead to occlusion

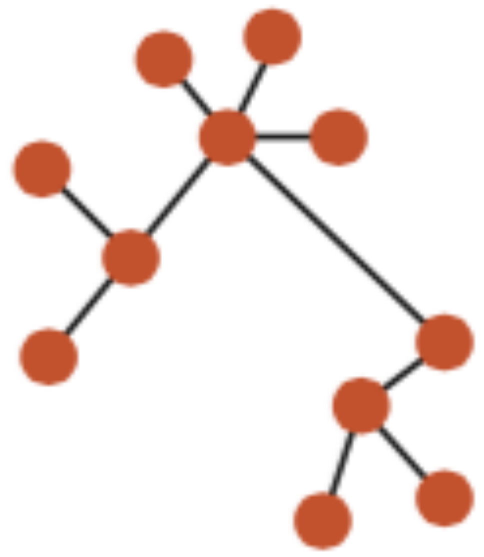
Tree Visualization Reference



Summary

Node–Link Diagrams
Connection Marks

✔ NETWORKS ✔ TREES



Adjacency Matrix
Derived Table

✔ NETWORKS ✔ TREES

Enclosure
Containment Marks

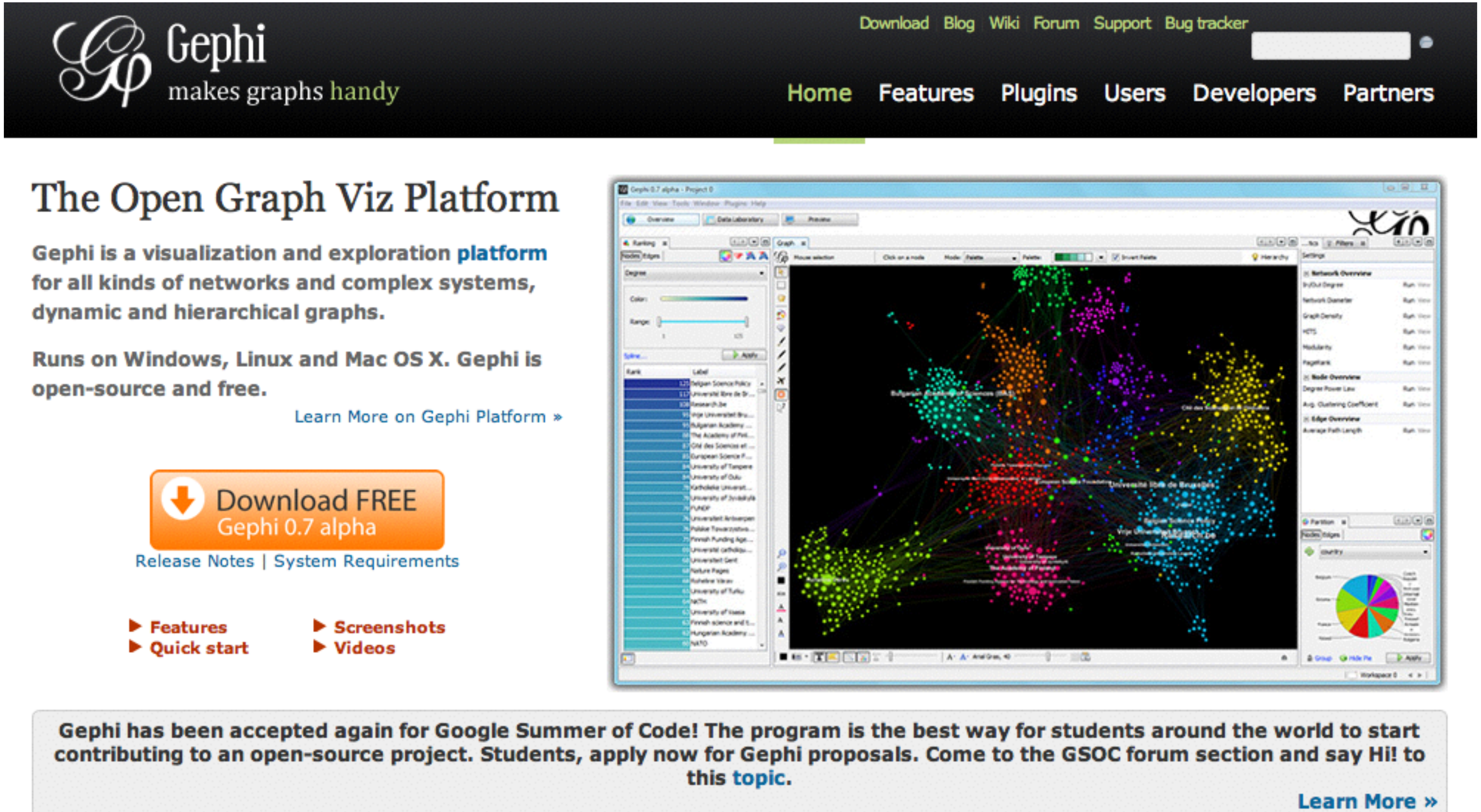
✘ NETWORKS ✔ TREES

Munzner 2014

Graph Tools & Applications

Gephi

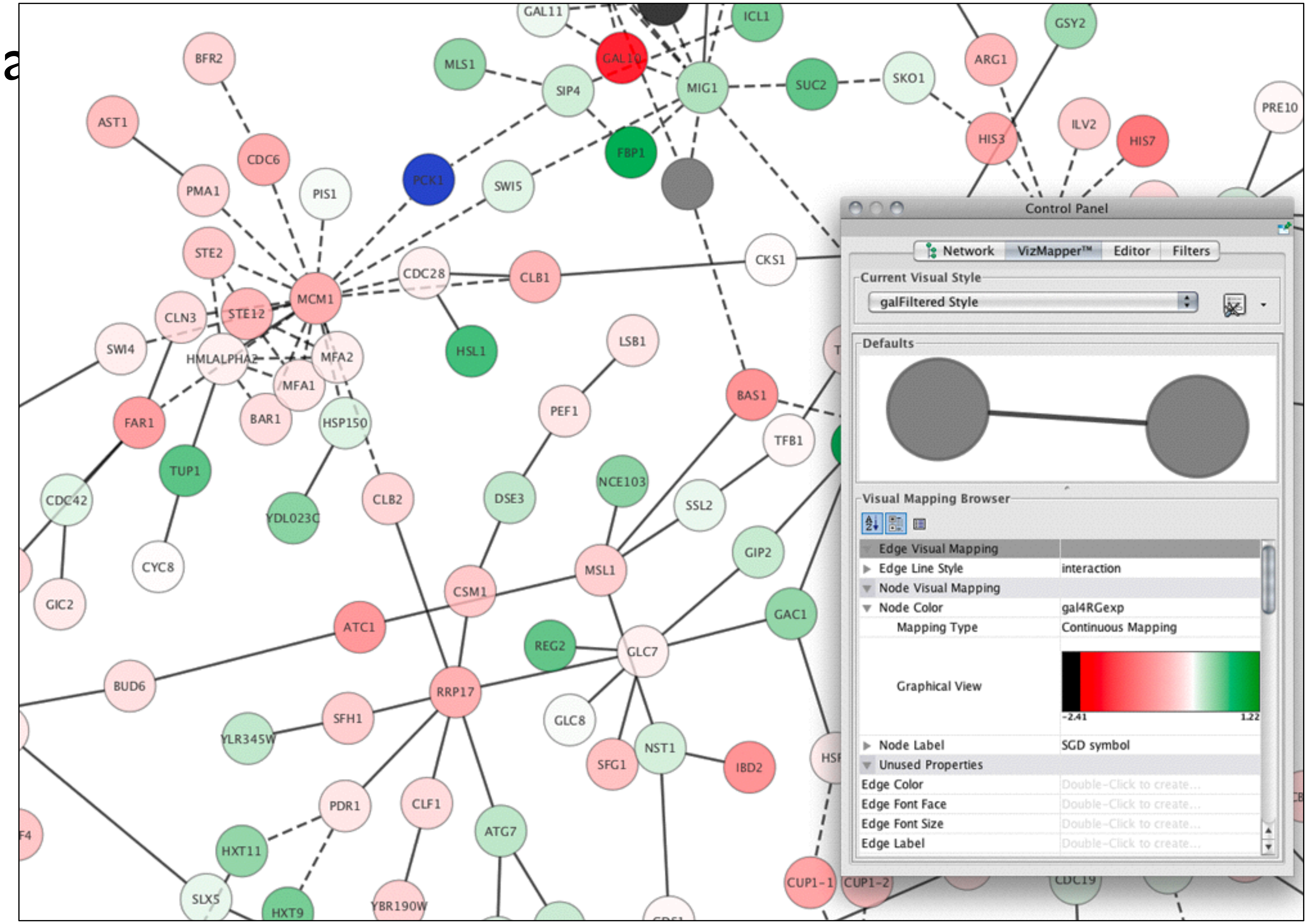
http://gephi.org



Cytoscape

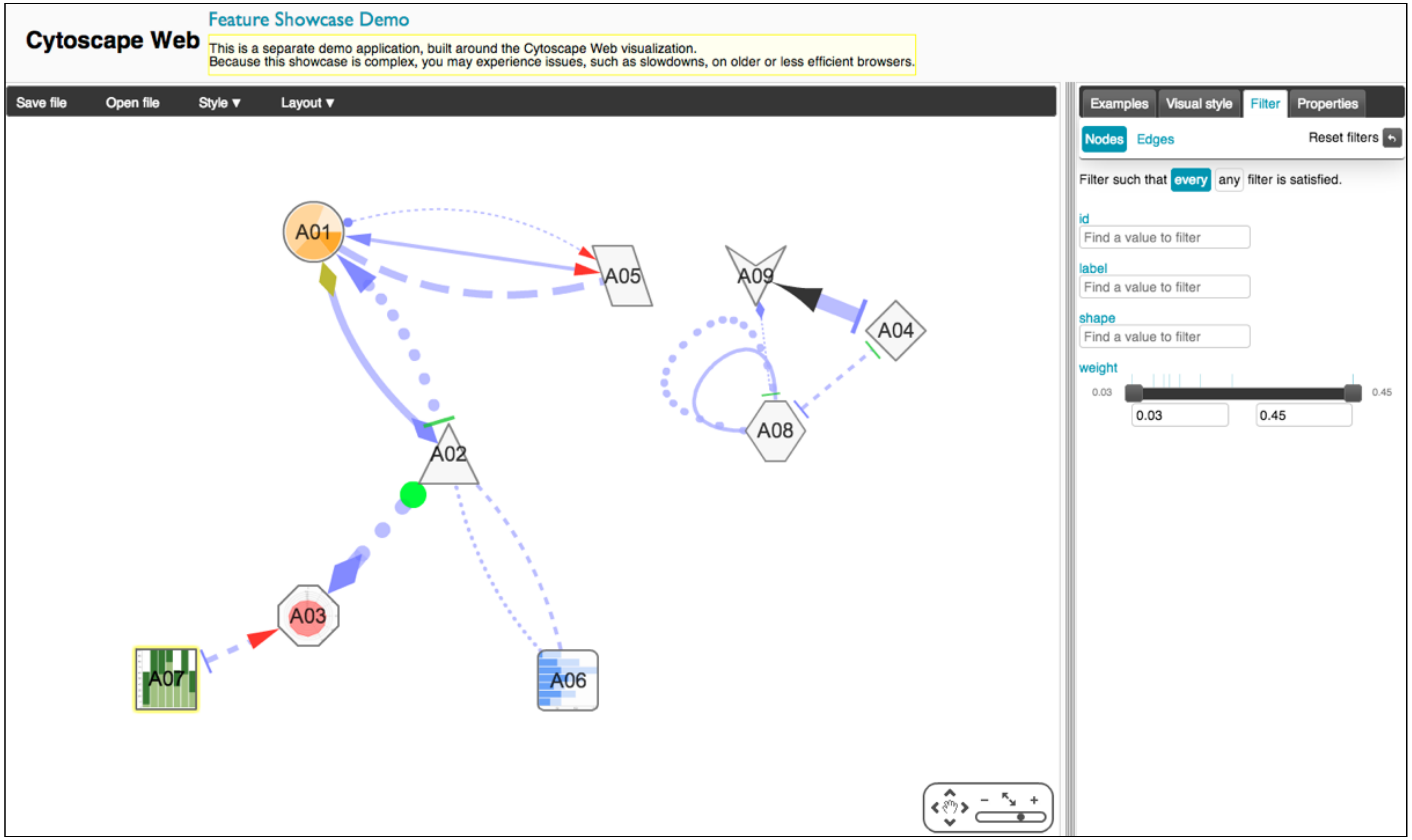
http://www.cytoscape.org/

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