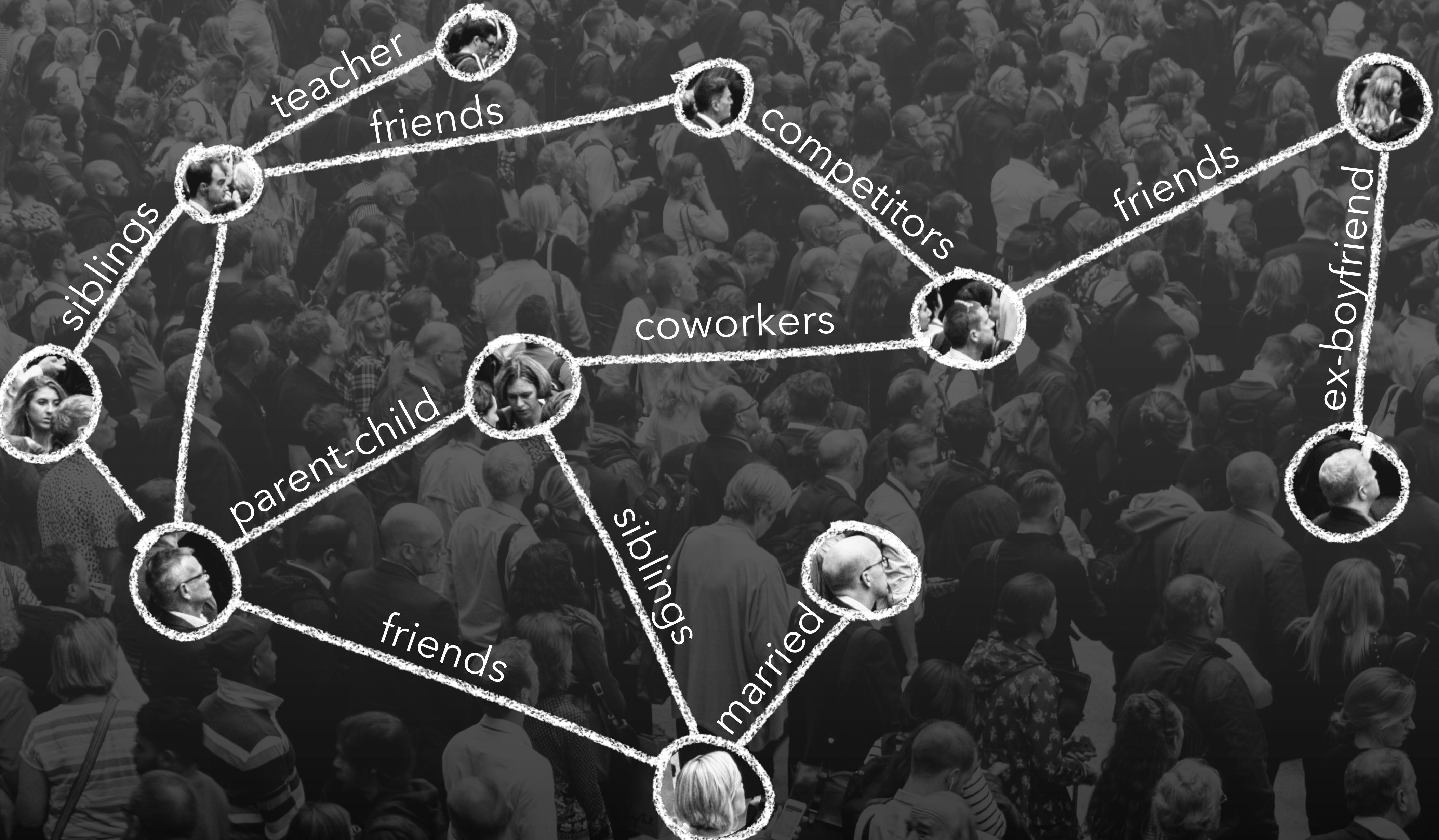


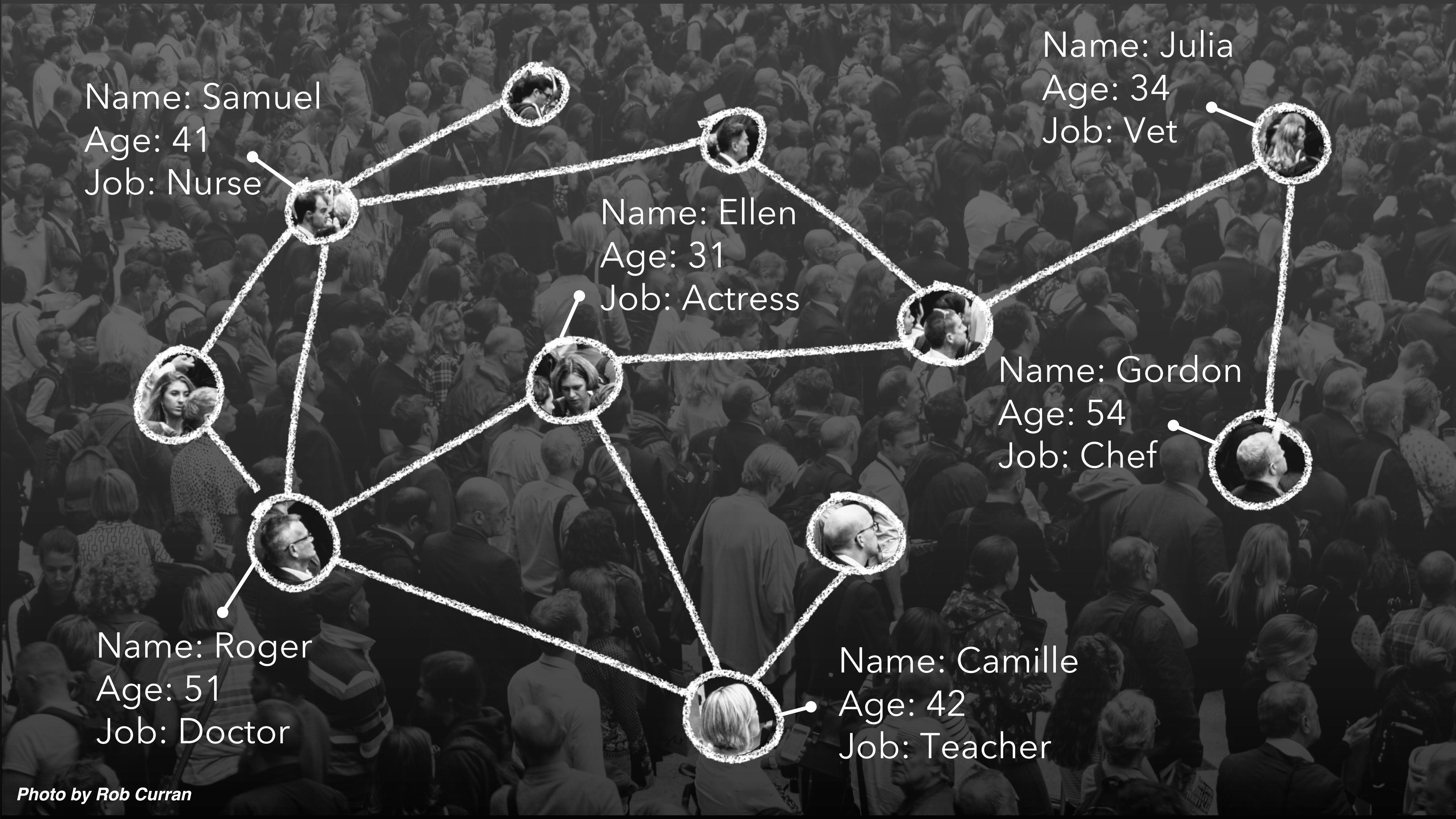
VIS EXAMPLES FOR MULTIVARIATE NETWORKS

CMPT 733

STEVEN BERGNER

SURVEY PAPER AND TUTORIAL BY
CAROLINA NOBRE, MARC STREIT, ALEXANDER LEX





Name: Samuel
Age: 41
Job: Nurse

Name: Roger
Age: 51
Job: Doctor

Name: Ellen
Age: 31
Job: Actress

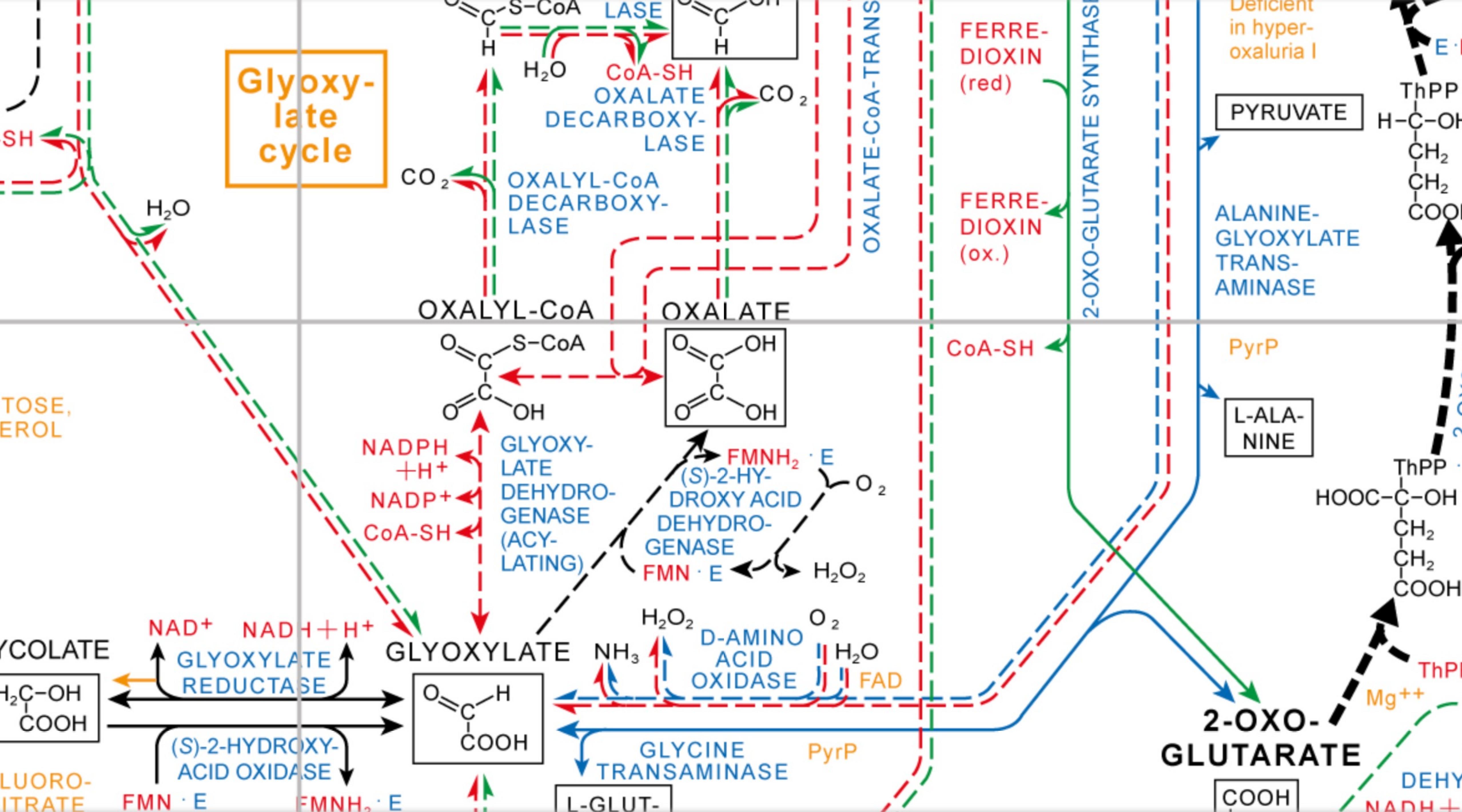
Name: Camille
Age: 42
Job: Teacher

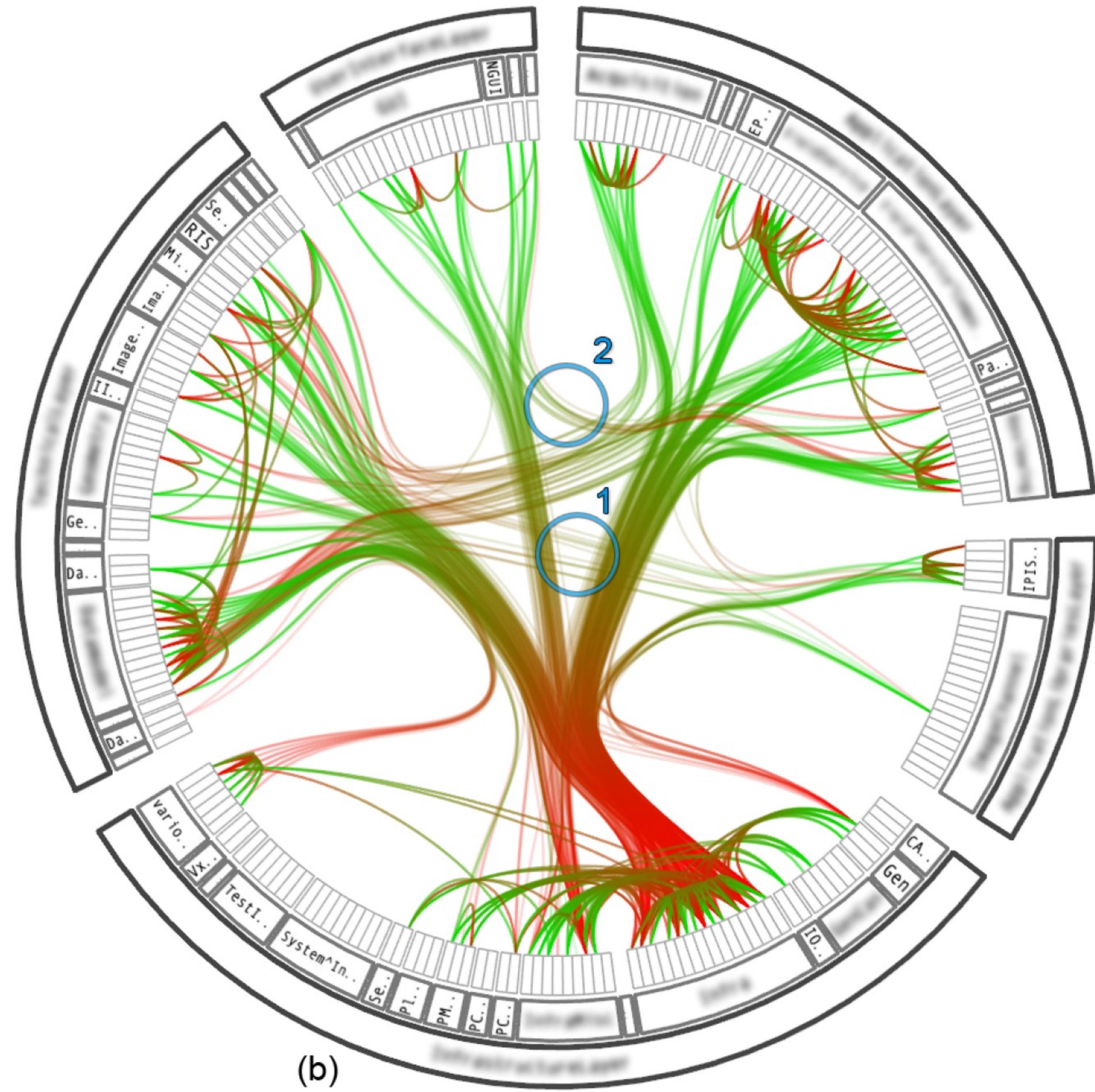
Name: Julia
Age: 34
Job: Vet

Name: Gordon
Age: 54
Job: Chef

A MULTIVARIATE NETWORK IS
NETWORK TOPOLOGY +
NODE AND EDGE ATTRIBUTES







(b)

Holten and Wijk, 2009

The State of the Art in Visualizing Multivariate Networks

C. Nobre¹ , M. Meyer¹ , M. Streit² , and A. Lex¹ 

¹University of Utah, Utah, USA

²Johannes Kepler University Linz, Austria

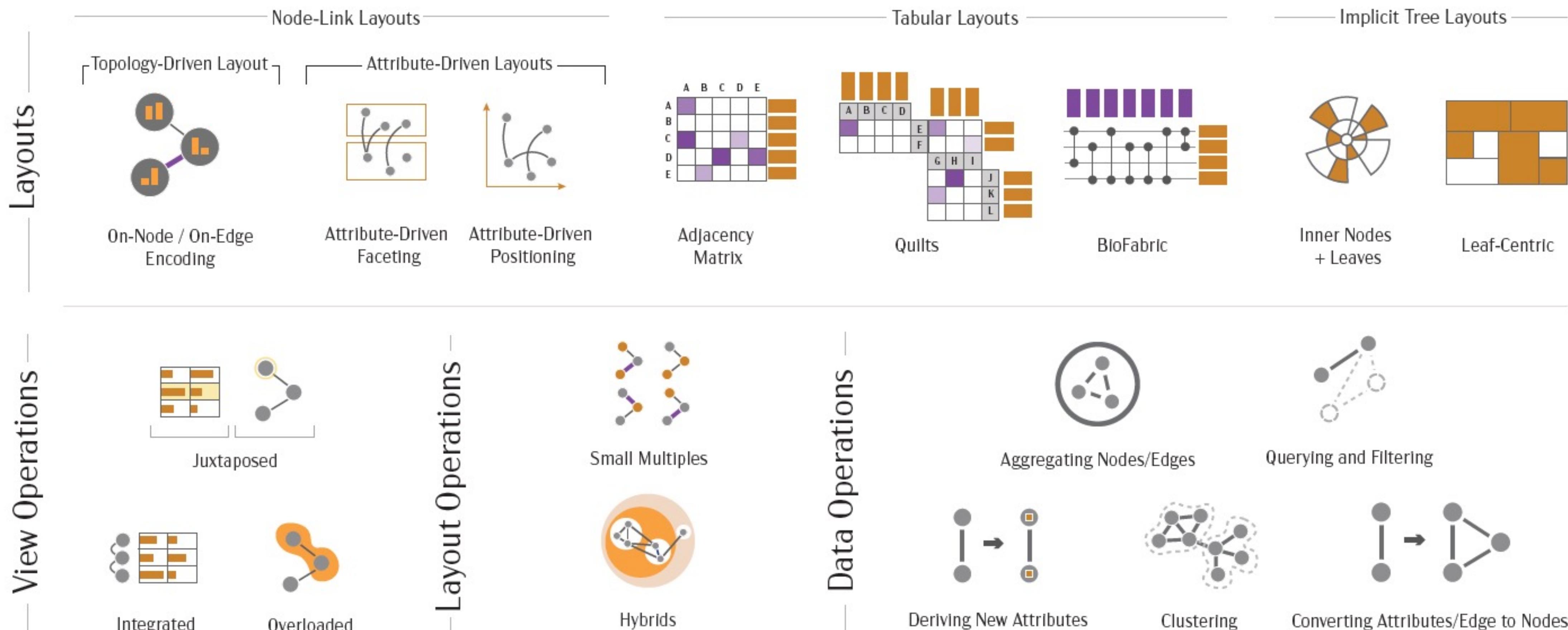
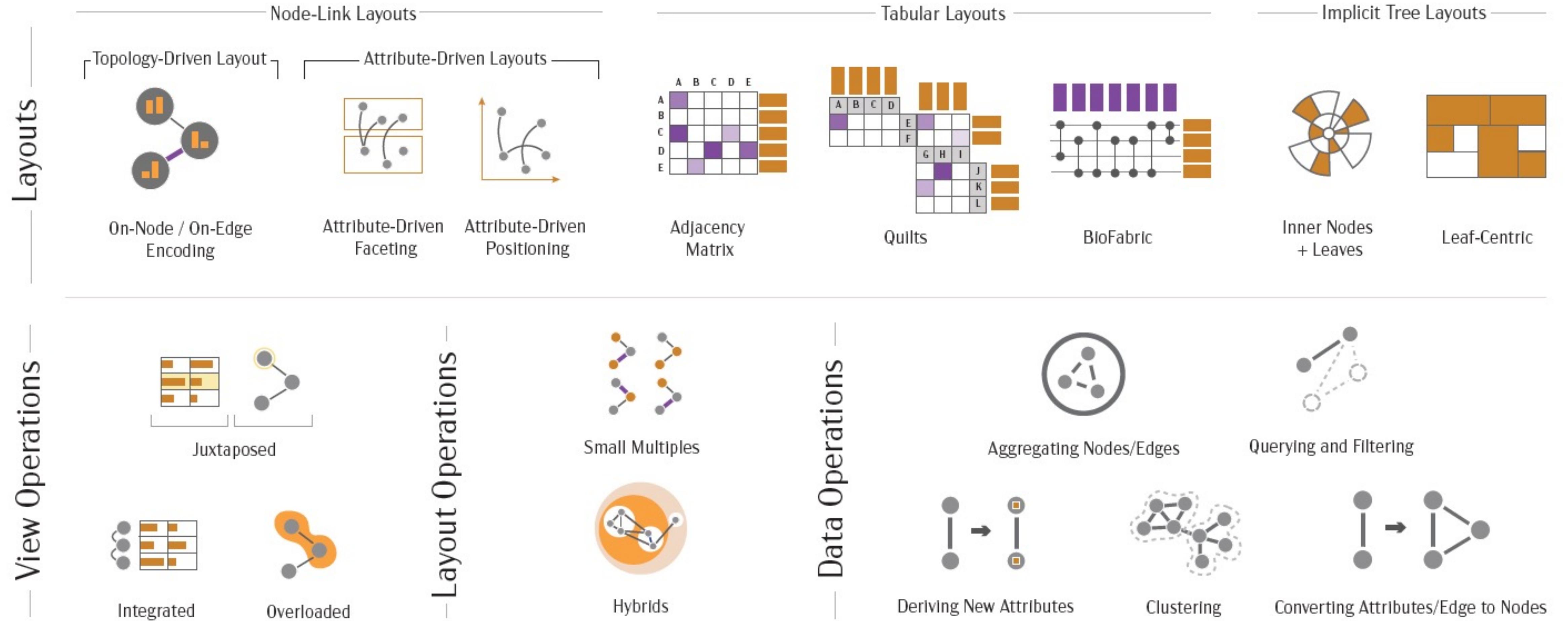


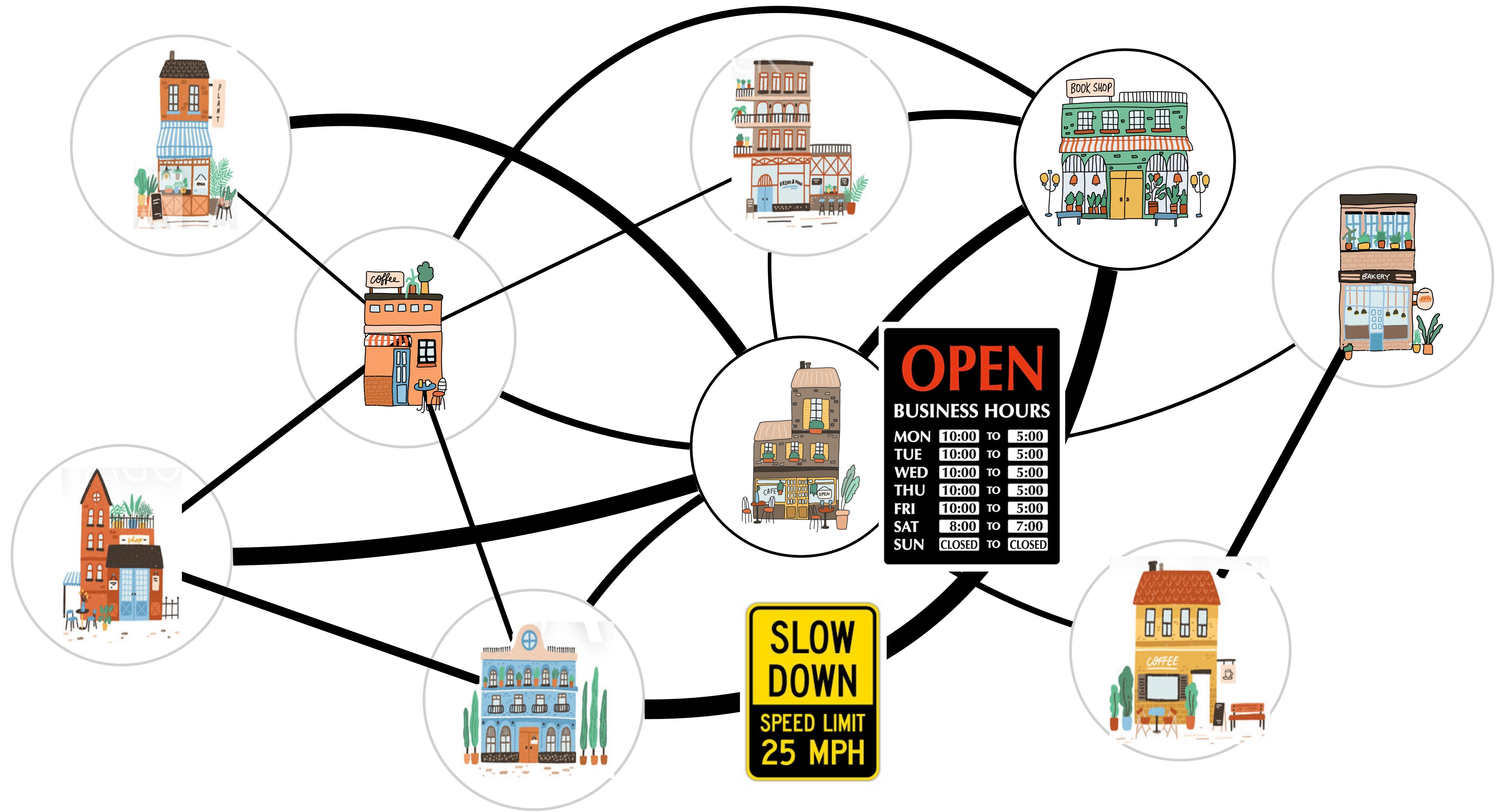
Figure 1: A taxonomy of network visualization techniques, organized into four main categories: Layouts, View Operations, Layout Operations, and Data Operations.



MVNV Tasks

How is an MVN task different than a regular graph task?

MVN Tasks rely on both the topology of the network and the **attributes** of the nodes and edges



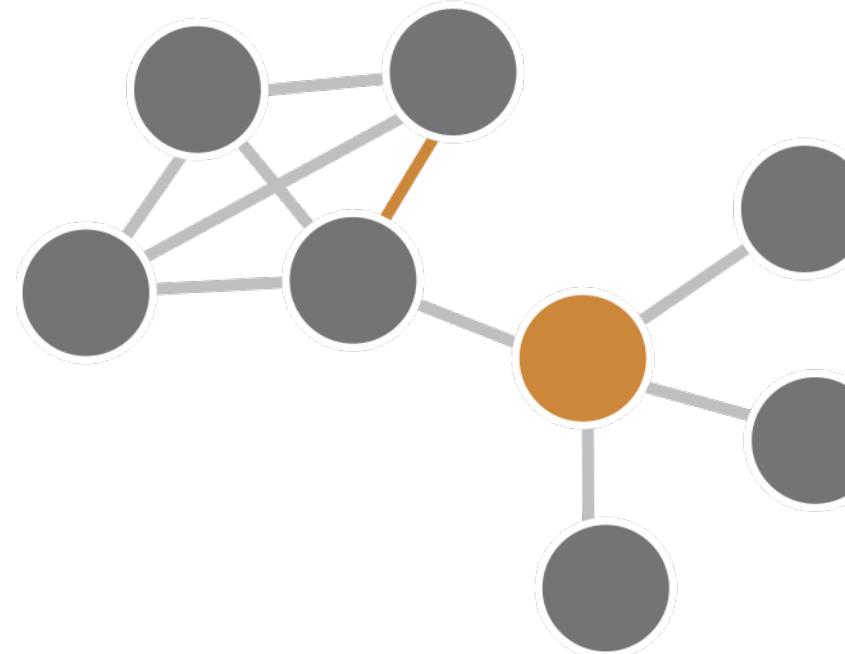
What is an efficient way I can complete all my errands?

-
- ▶ What is the **fastest route** to get all my errands done?

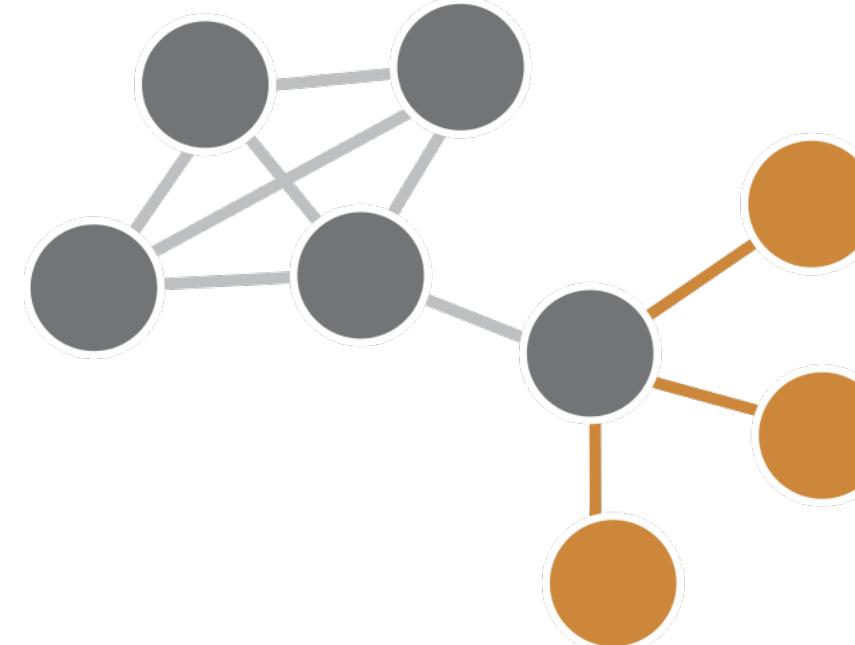
Tasks that rely on the **topology** of the network
and the **attributes** of the nodes and edges

MVN tasks are applied to topological structures

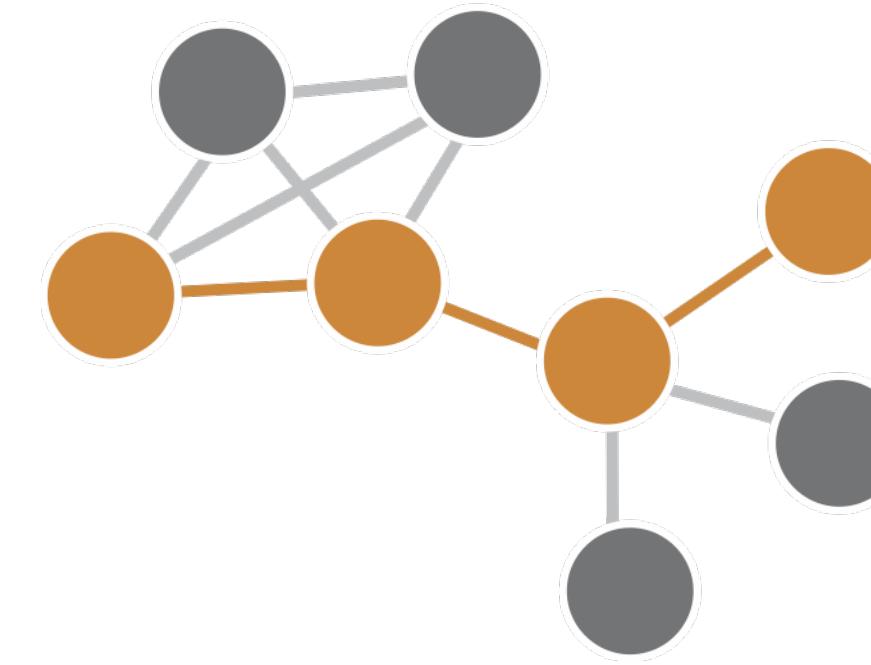
Single Node/Edge



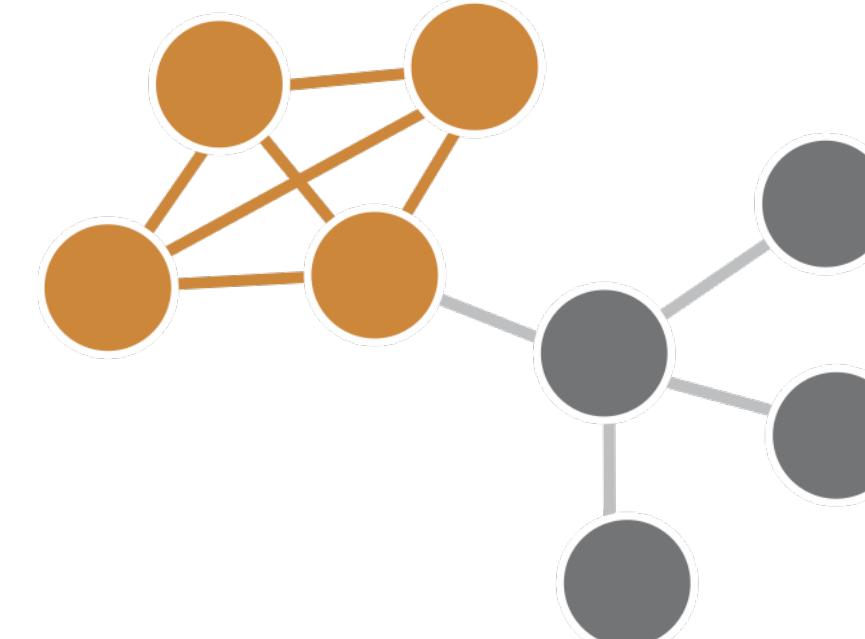
Node Neighbors



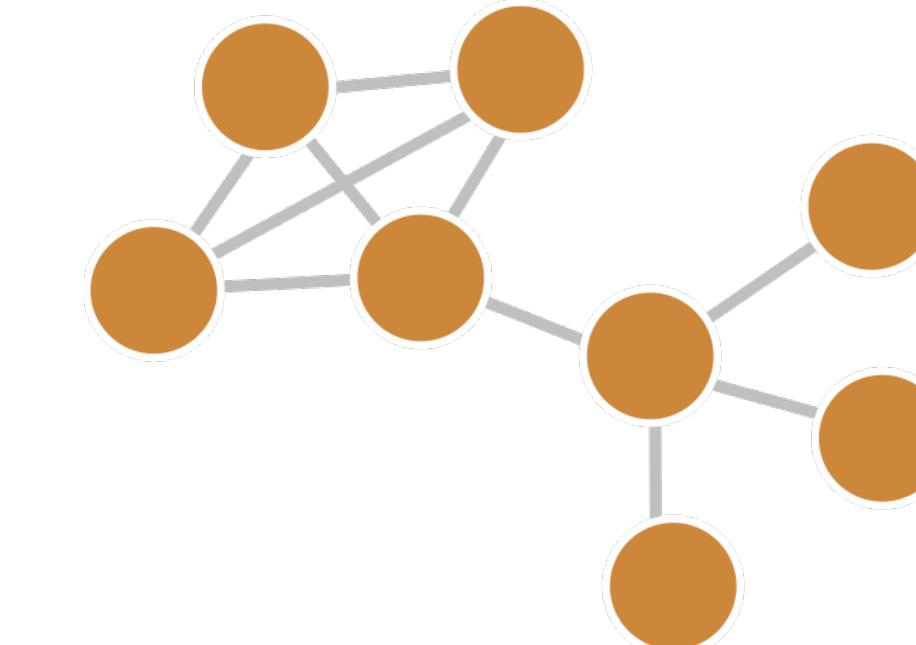
Path



Cluster

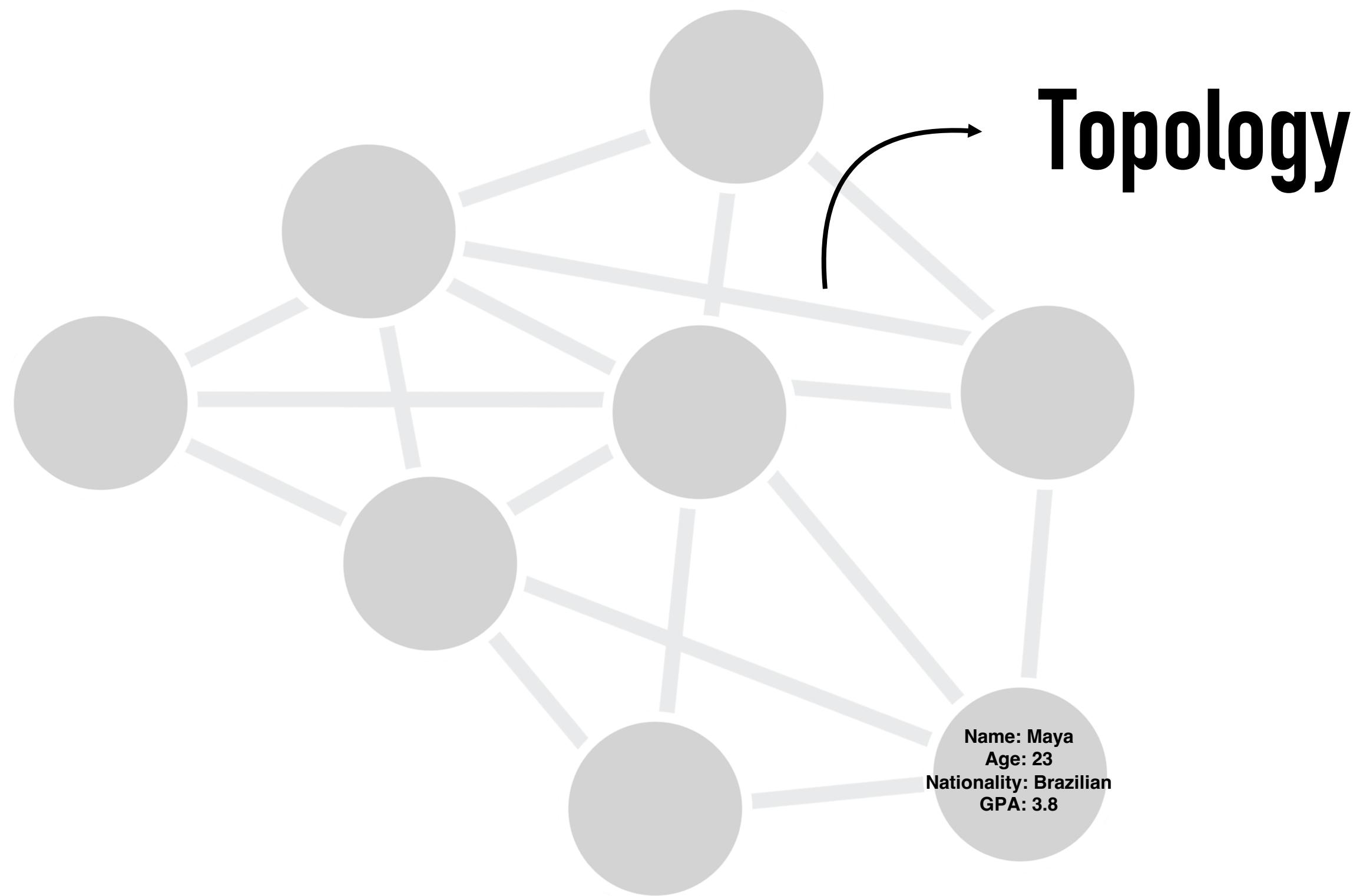


Network/Subnetwork

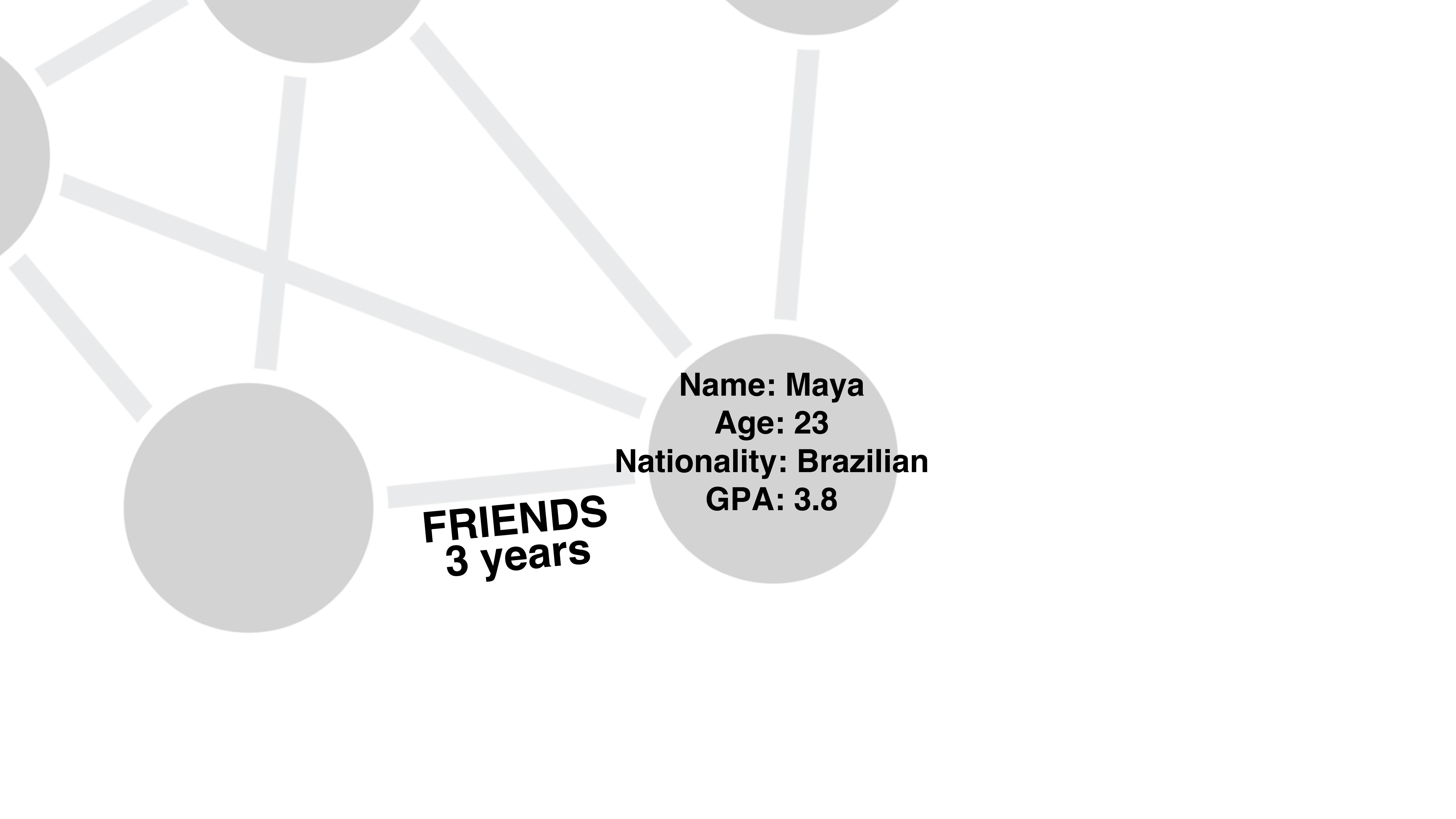


Network and Attribute Characteristics





Topology



Name: Maya

Age: 23

Nationality: Brazilian

GPA: 3.8

**FRIENDS
3 years**

**FRIENDS
3 years**

Name: Maya
Age: 23
Nationality: Brazilian
GPA: 3.8
Degree: 4

ity

Person

Name: Pedro

Age: 25

Nationality: Brazilian

Brazilians

GPA: 3.3

DEGREE: 3

Person

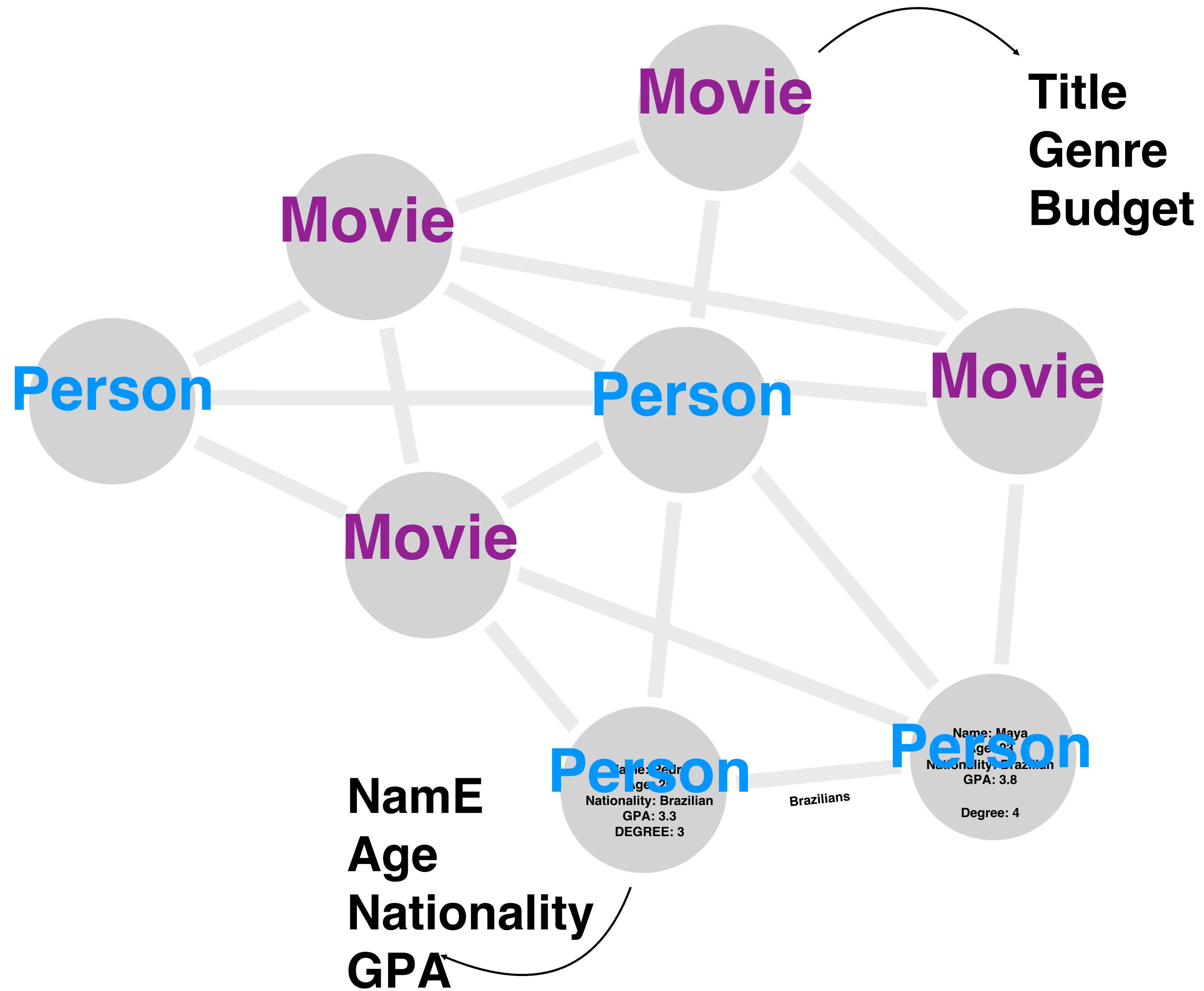
Name: Maya

Age: 23

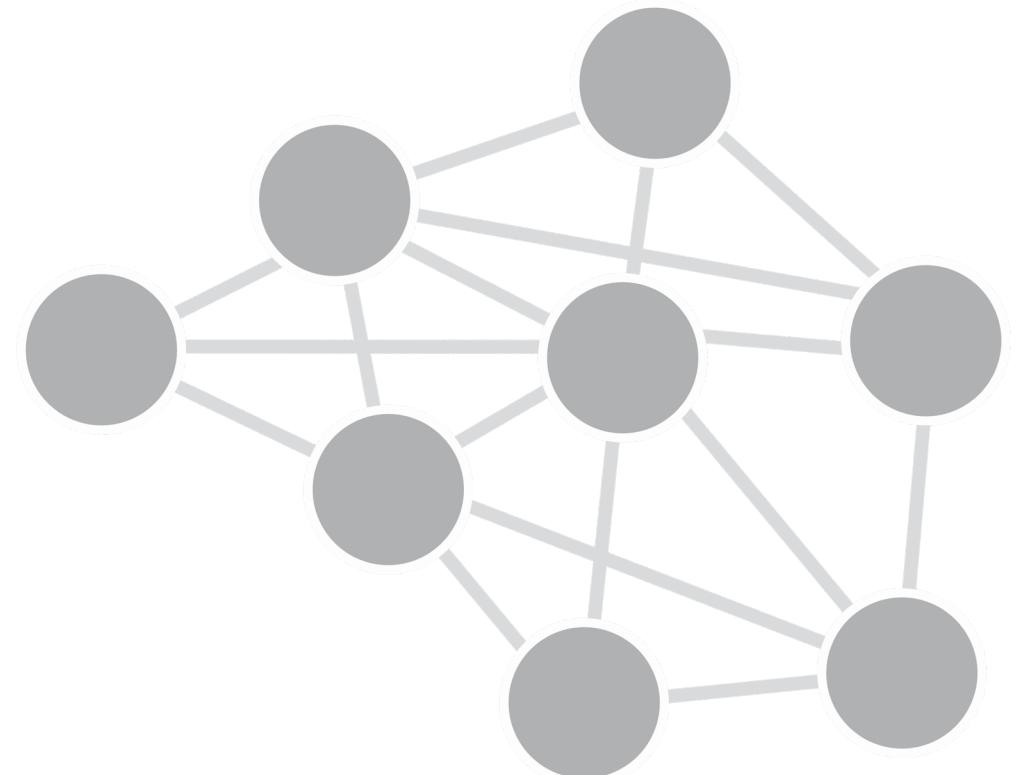
Nationality: Brazilian

GPA: 3.8

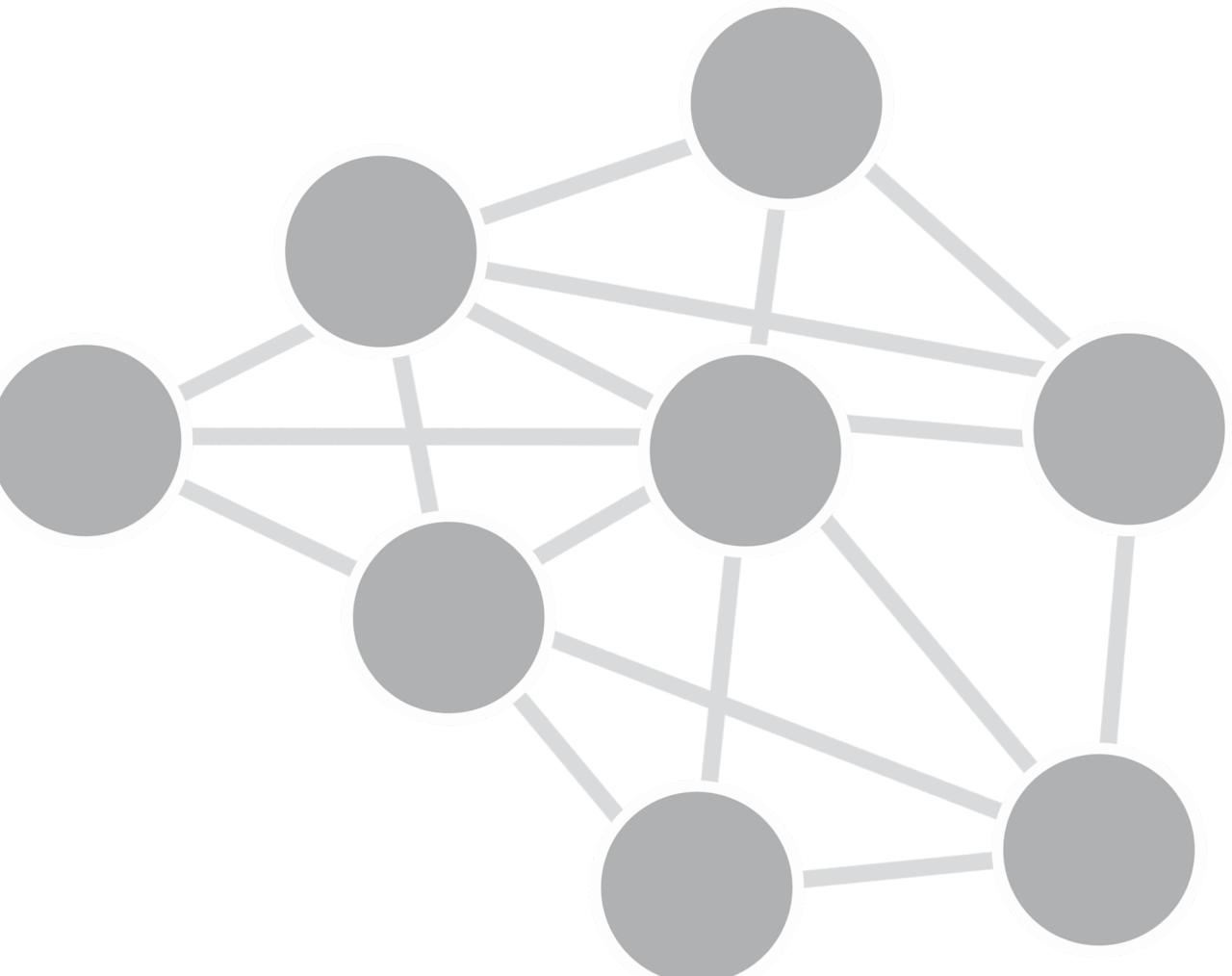
Degree: 4



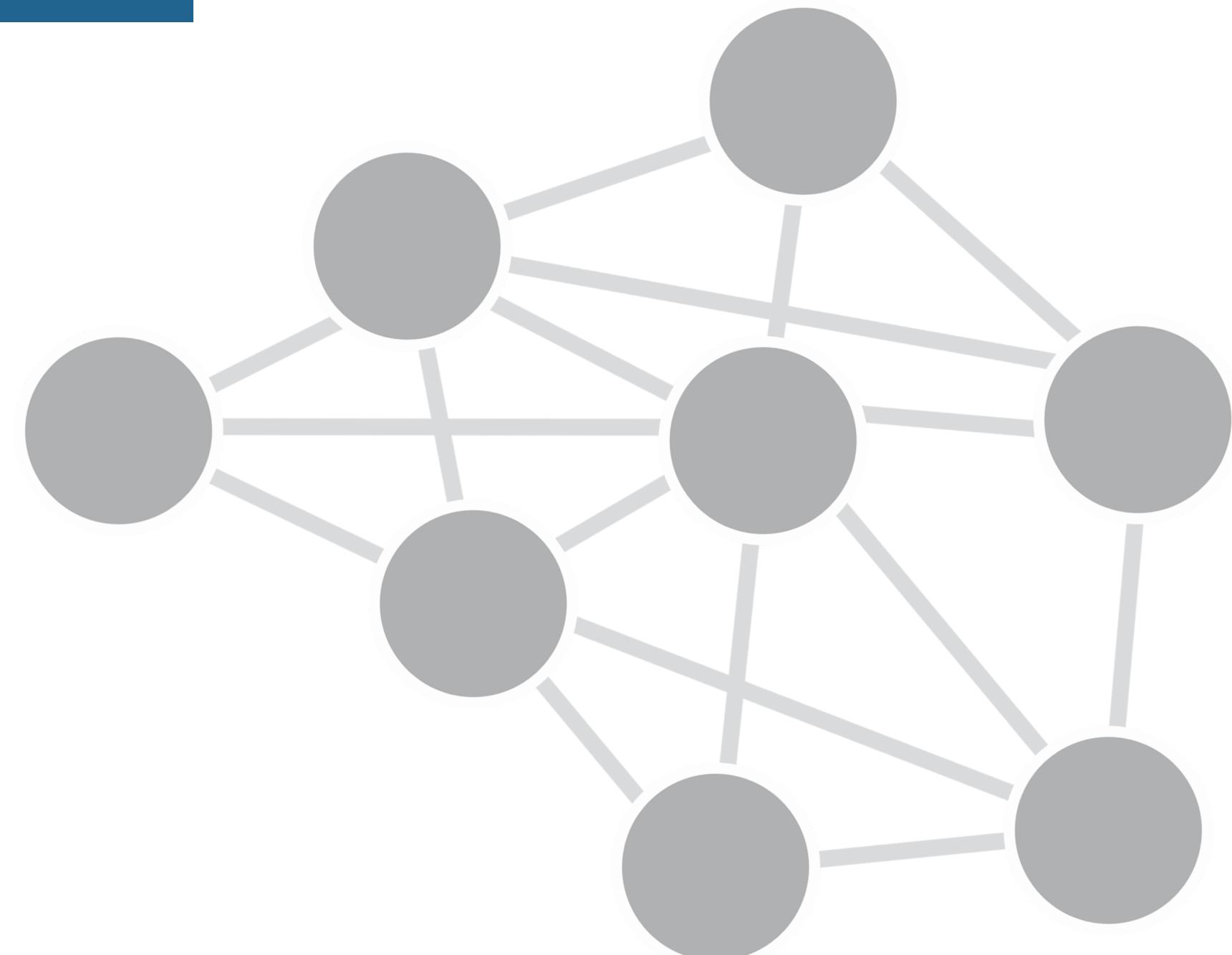
Network Size



Small
 <100

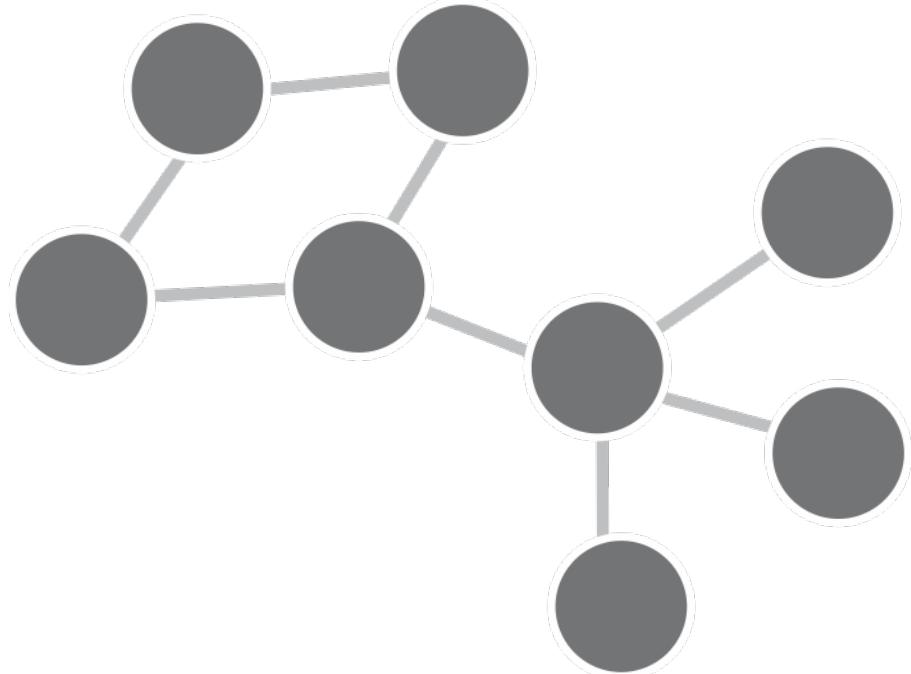


Medium
100-1000

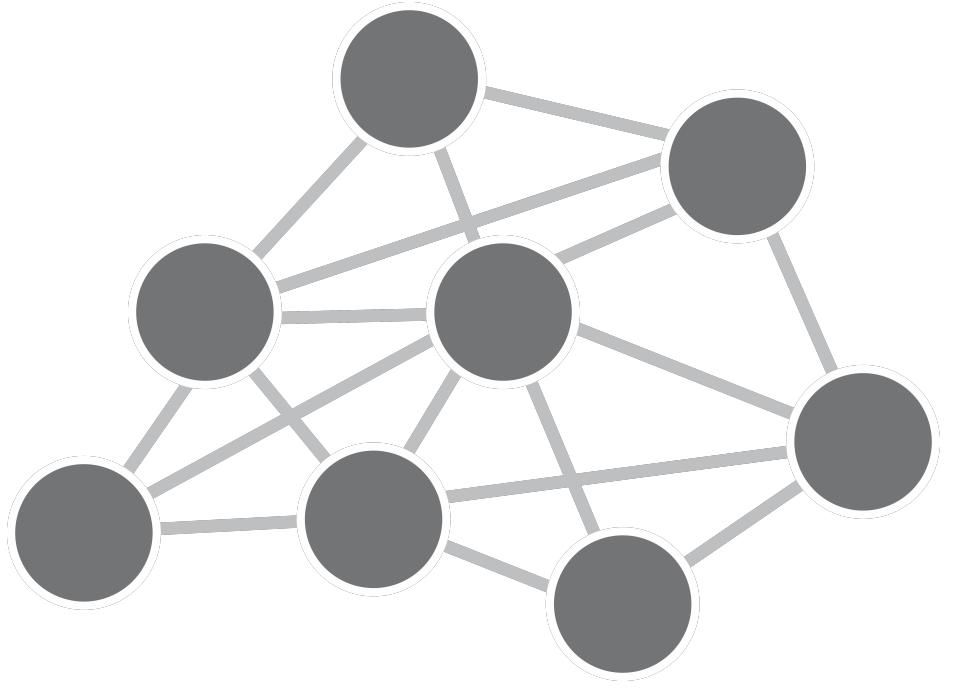


Large
 >1000

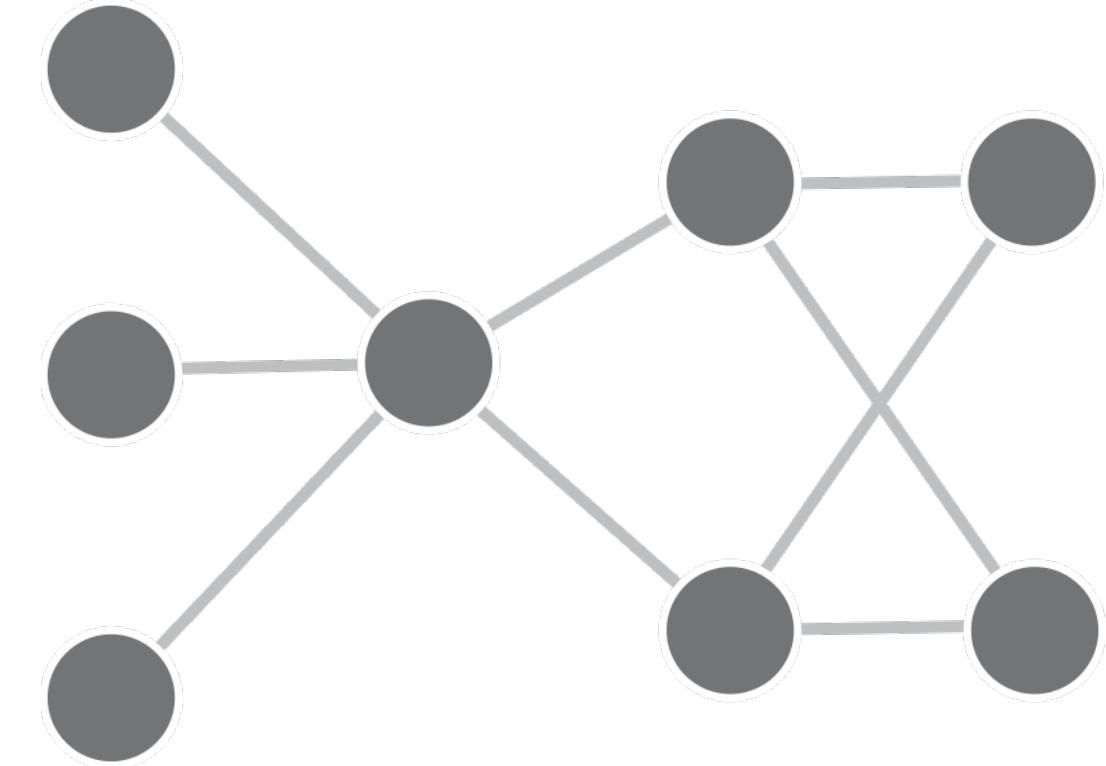
Network Types



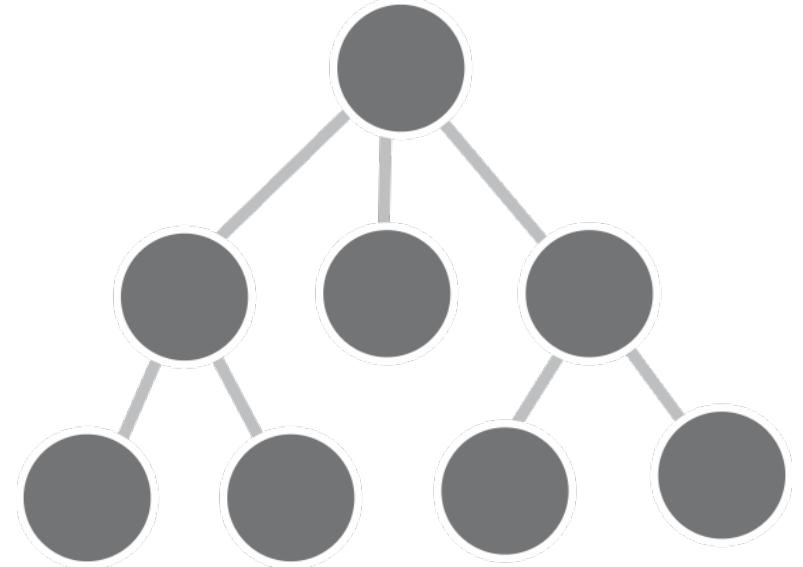
Sparse



Dense



Layered



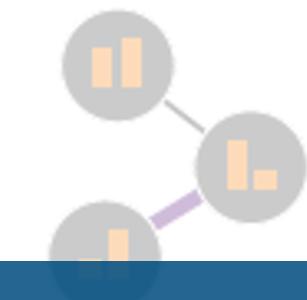
Trees

Taxonomy of Layouts and Operations

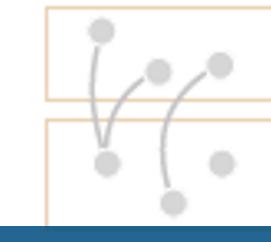
Layouts

Node-Link Layouts

Topology-Driven Layout



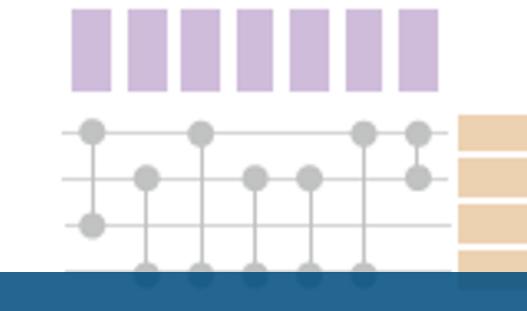
Attribute-Driven Layouts



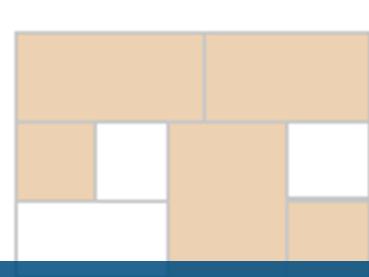
Tabular Layouts

A	B	C	D	E
A				
B				
C				
D				
E				

A	B	C	D
E			
F			
G			
H			
I			
J			
K			
L			



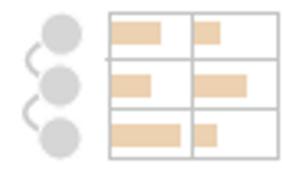
Implicit Tree Layouts



View Operations



Juxtaposed



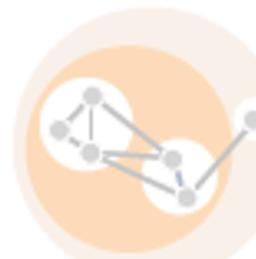
Integrated



Overloaded

Layout Operations

Small Multiples



Hybrids

Data Operations

Aggregating Nodes/Edges

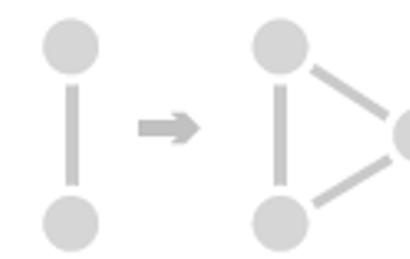


Deriving New Attributes

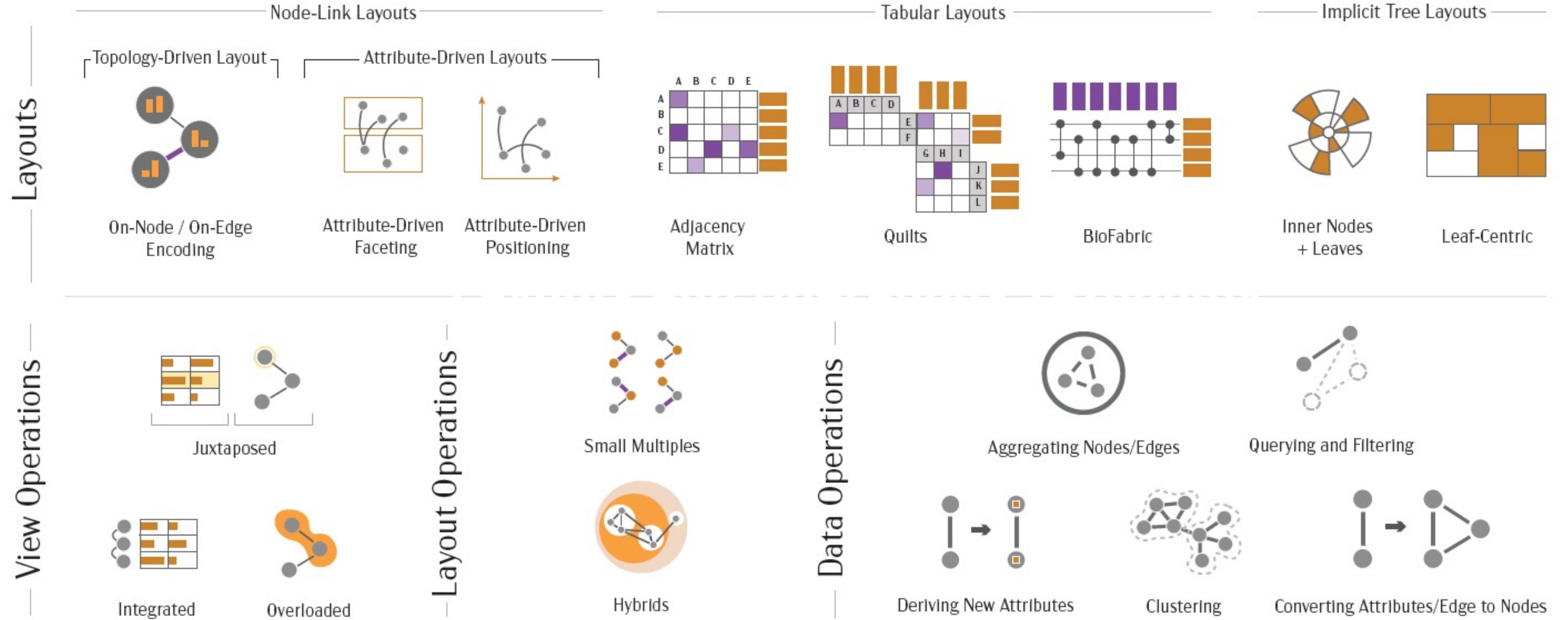


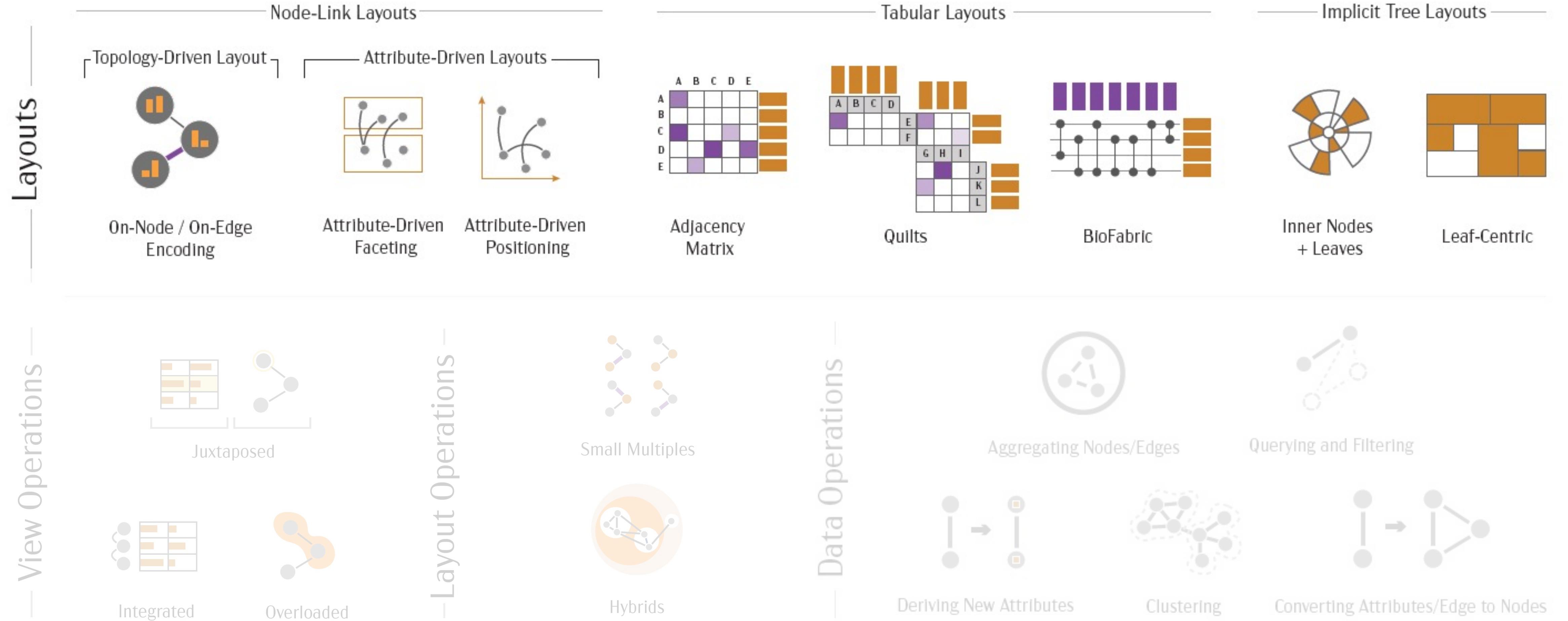
Clustering

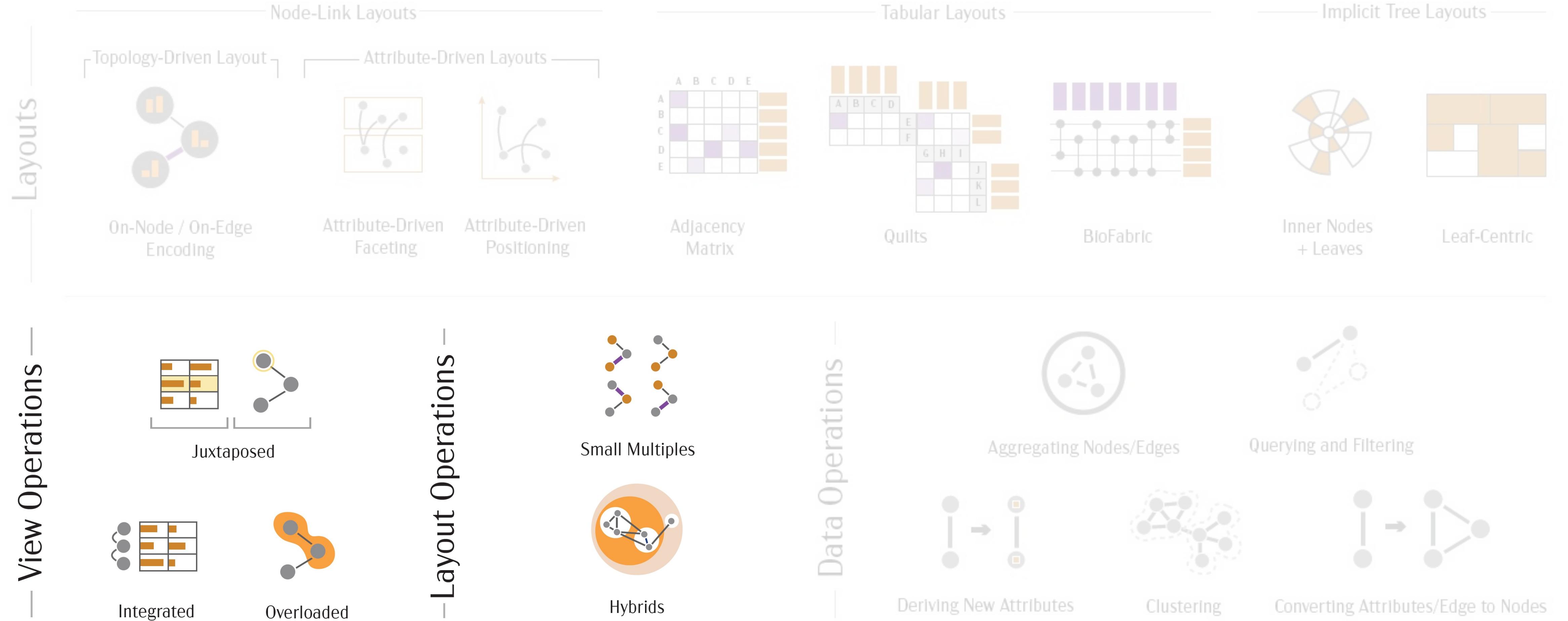
Querying and Filtering



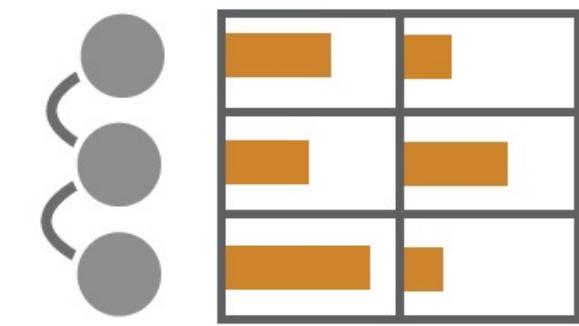
Converting Attributes/Edge to Nodes



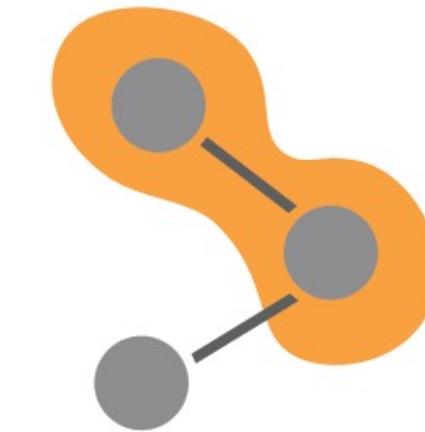




— View Operations —

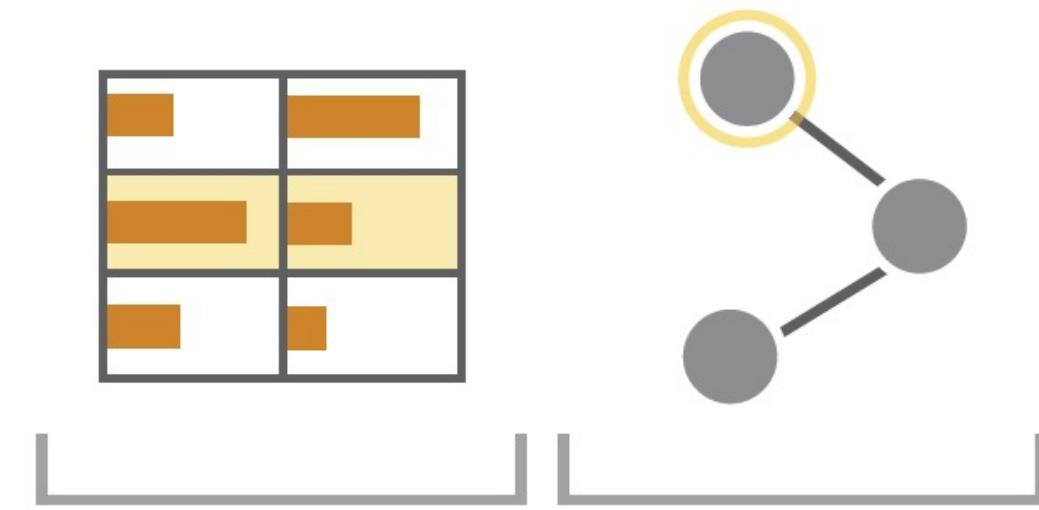


Integrated

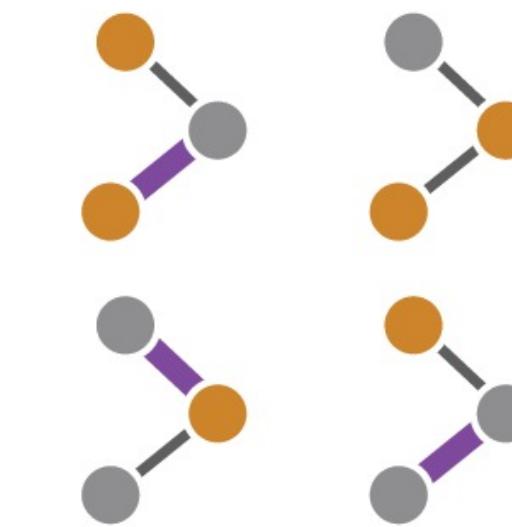


Overloaded

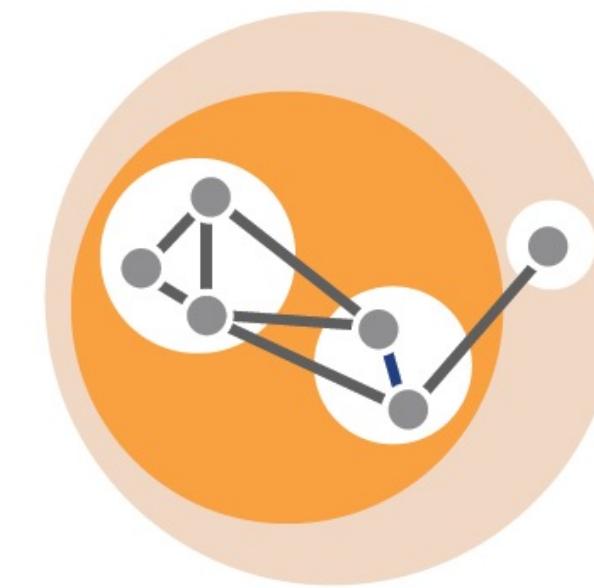
Operations



Juxtaposed



Small Multiples

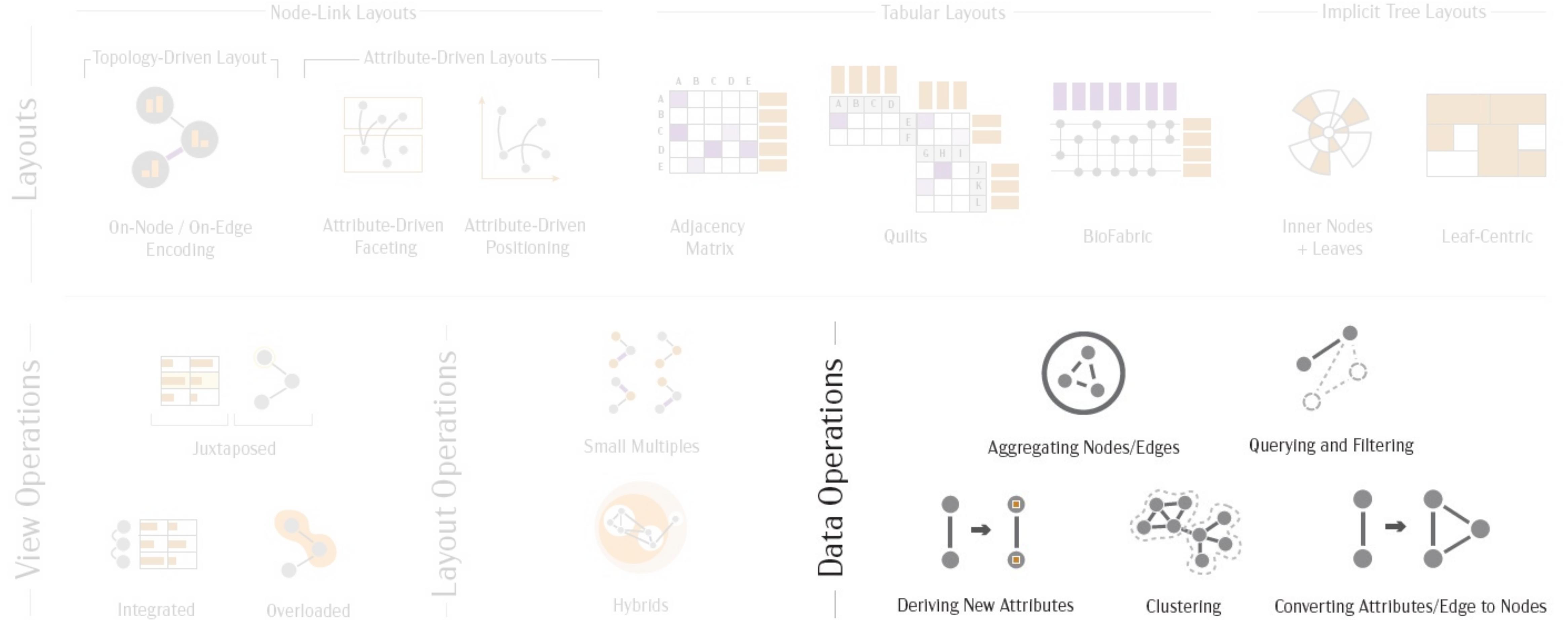


Hybrids

**Separate views for
Topology and Attributes**

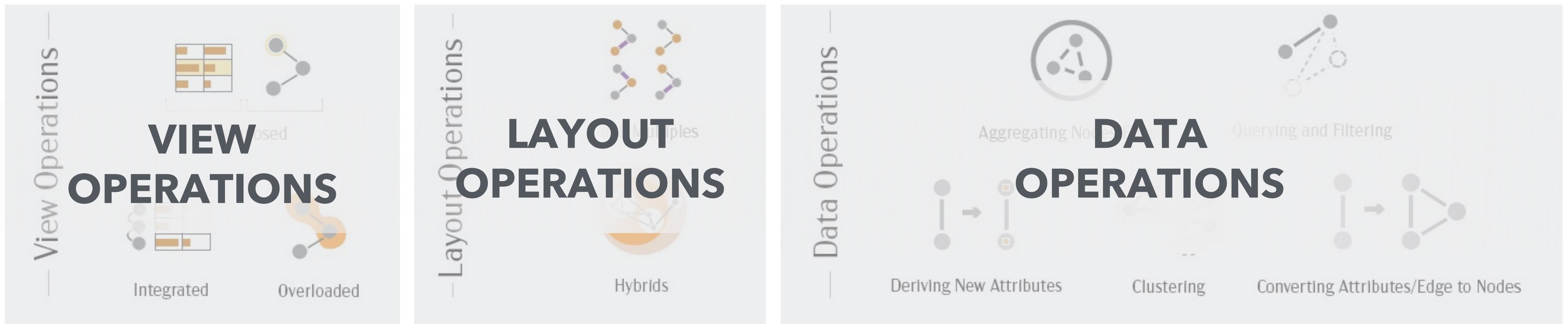
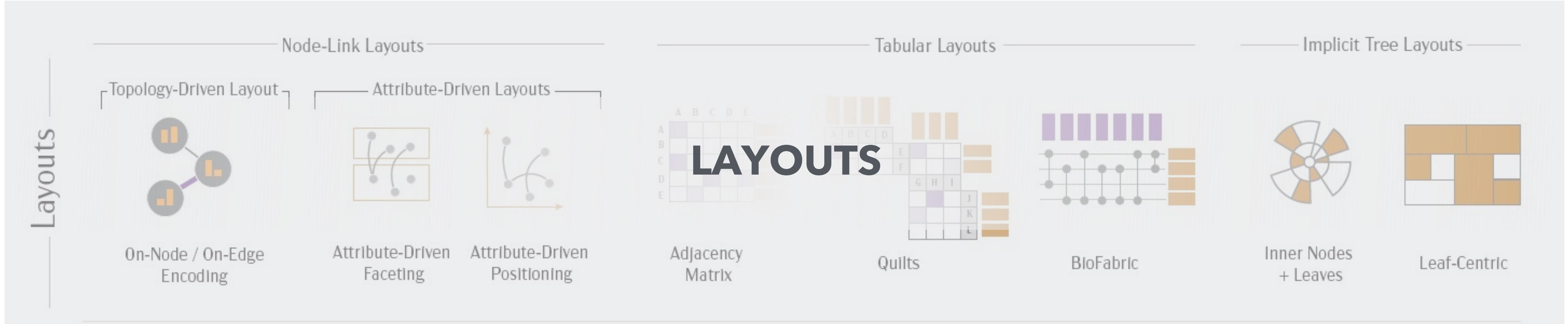
— Layout Operations —

**Multiple layouts for
Topology or Attributes**

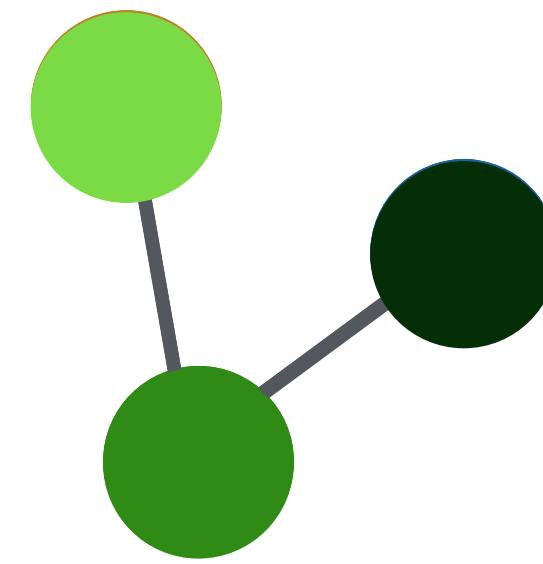
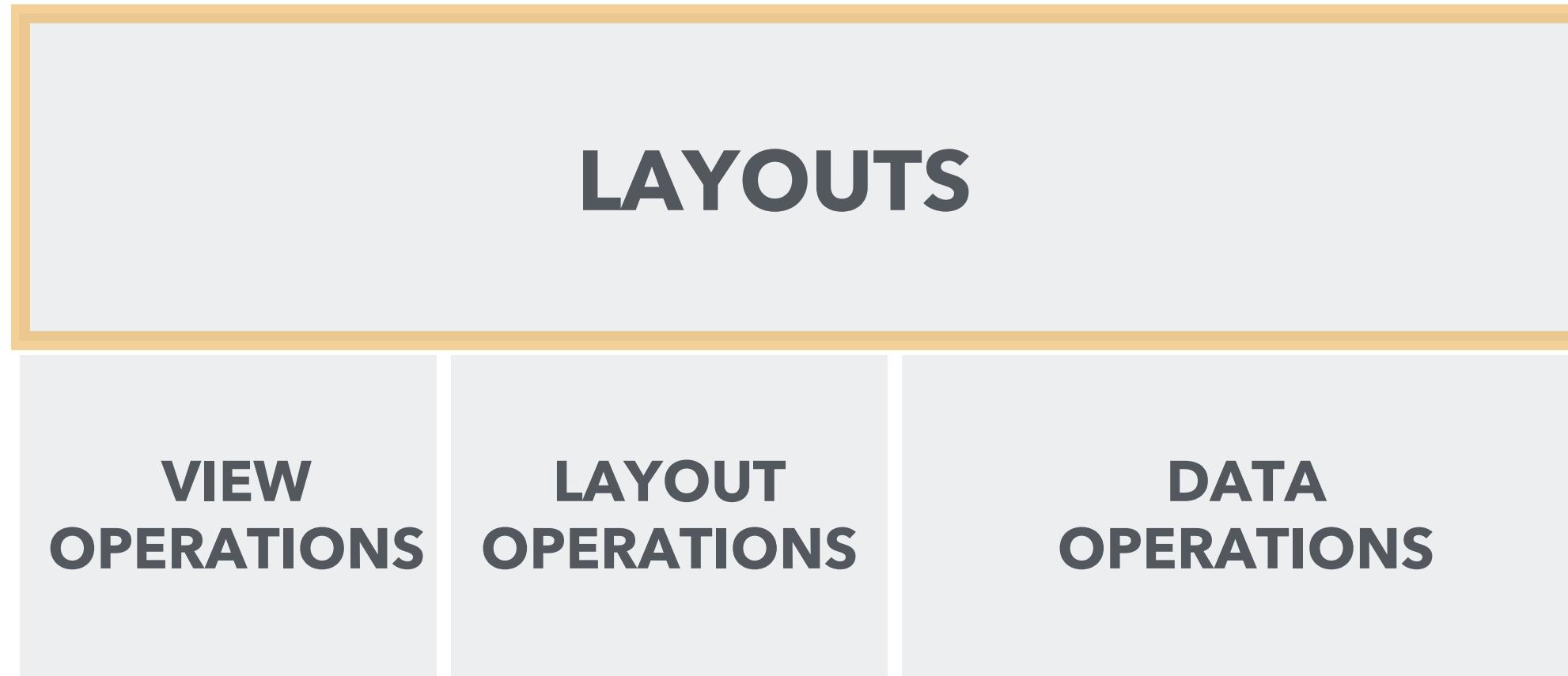


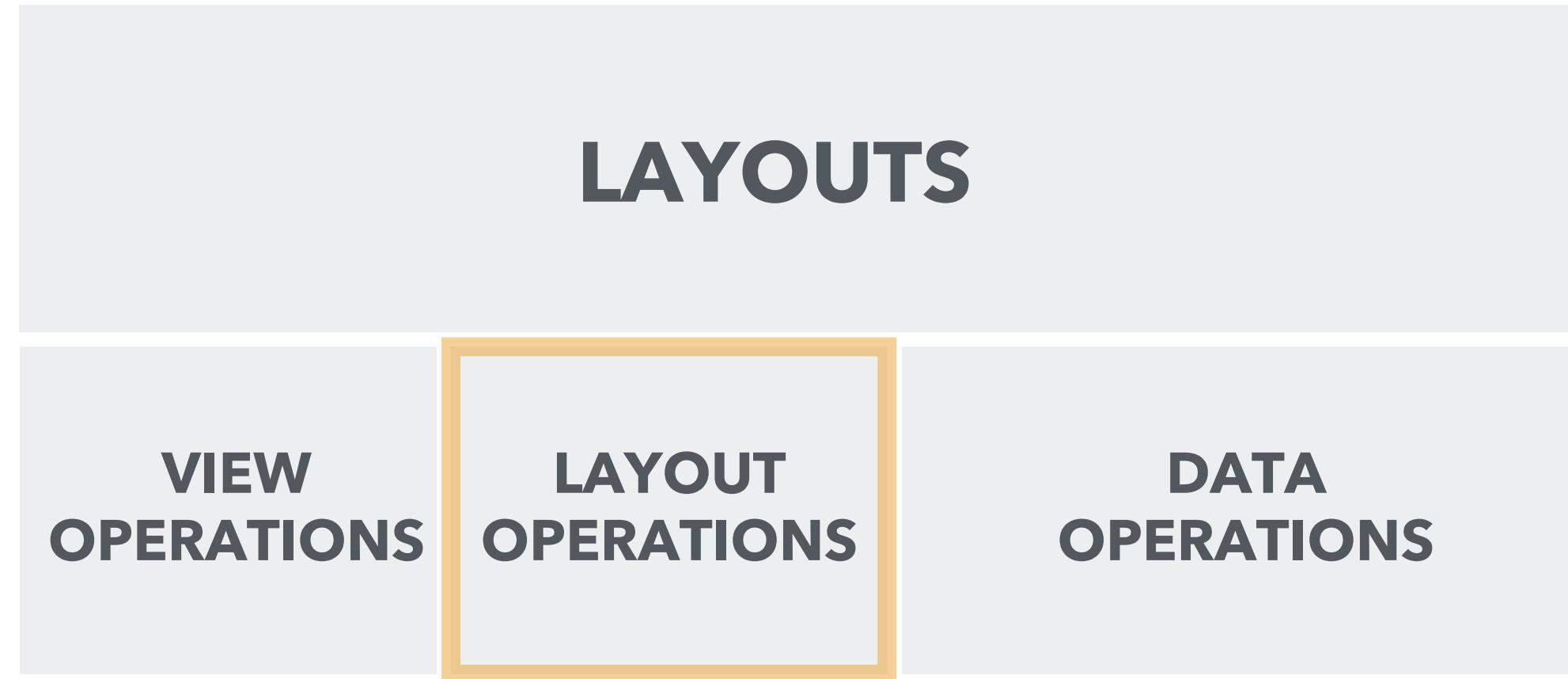
	Size	Type	Node Attributes	Edge Attributes	Topolog. Structures			
	Small <100 nodes)	Complex (sparse)	Few (<5)	Few (<3)	Single node/edge			
	Medium (<1,000)	Complex (dense)	Several (≥ 5)	Several (≥ 3)	Neighbors			
	Large (>1,000 nodes)	Layered/K-Partite Trees	Homog. (1 type) Hetero. (>1 type)	Homog. (1 type) Hetero. (>1 type)	Paths			
					Clusters			
					Entire/sub network			
Node-Link Layouts	On-node/edge encoding		3 2 1	3 1 3 3	2 1 3 2	2 1 3 1	3 3 2 2 2	0
	Attr.-driven faceting		3 1 1	3 1 3 1	3 1 3 3	2 1 2 1	3 2 1 1 1	1
	Attr.-driven positioning		3 1 1	3 1 1 1	3 1 3 1	2 1 2 1	3 2 1 1 2	2
Tabular Layouts	Adjacency matrix		3 1 1	2 3 2 1	2 3 3 2	3 2 3 2	3 3 1 3 2	3
	Quilts		3 1 1	3 1 3 3	3 3 3 3	3 3 3 2	3 3 2 2 2	3
	BioFabric		3 1 1	3 3 2 1	3 3 3 3	3 3 3 3	3 1 1 1 2	3
Implicit	Inner nodes & leaves		3 2 1	0 0 0 3	3 1 3 1	0 0 0 0	3 3 3 0 3	0
	Leaves		3 2 2	0 0 0 3	3 1 3 1	0 0 0 0	3 2 1 0 3	1
View Operations	Juxtaposed		3 2 1	3 1 3 3	3 3 3 3	3 3 3 3	2 1 1 2 2	2
	Integrated		3 2 1	3 1 3 3	3 3 3 3	2 2 3 3	3 3 3 1 2	2
	Overloaded		3 2 1	3 1 3 3	3 1 3 1	1 1 1 1	3 3 2 3 2	3

Does *not* support
Supports poorly
Supports
Optimized for

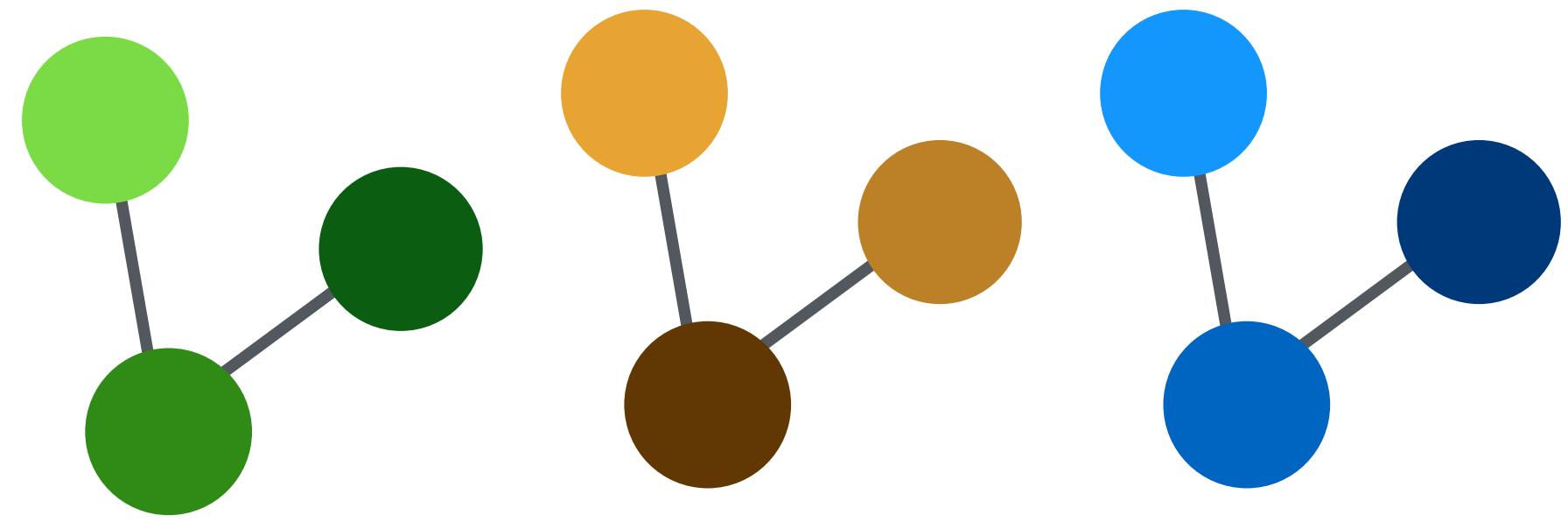


Node-Link Diagram with on-node encoding

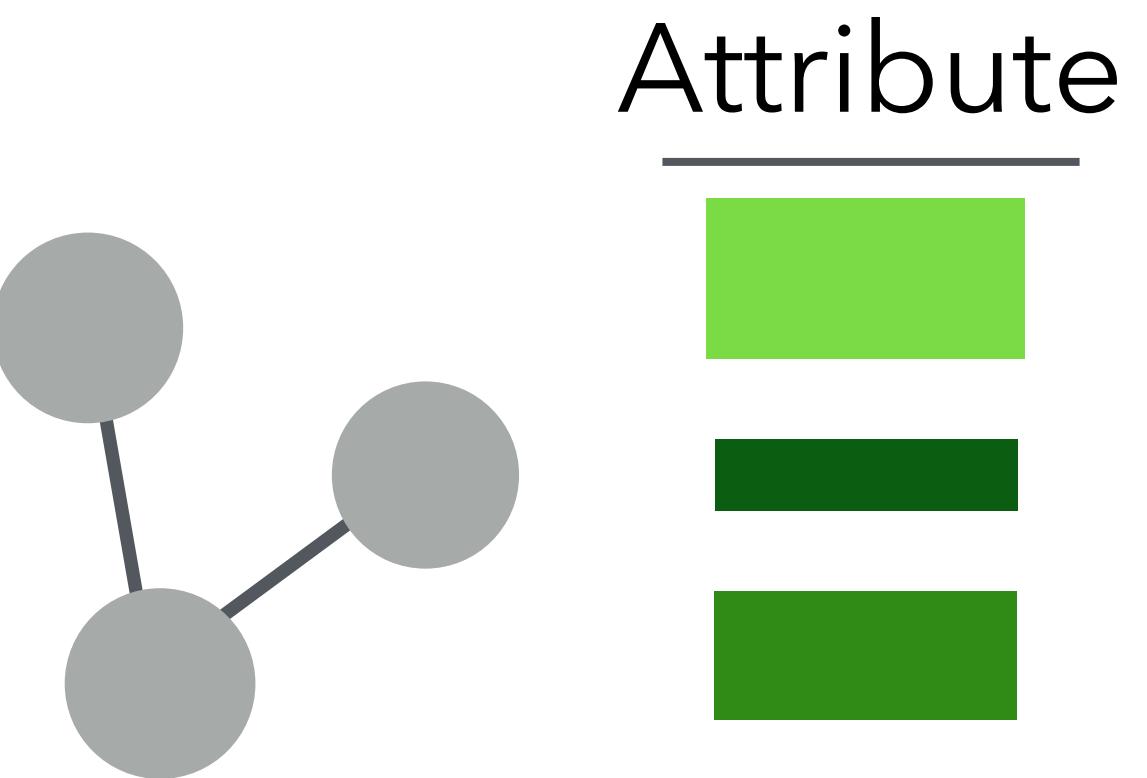
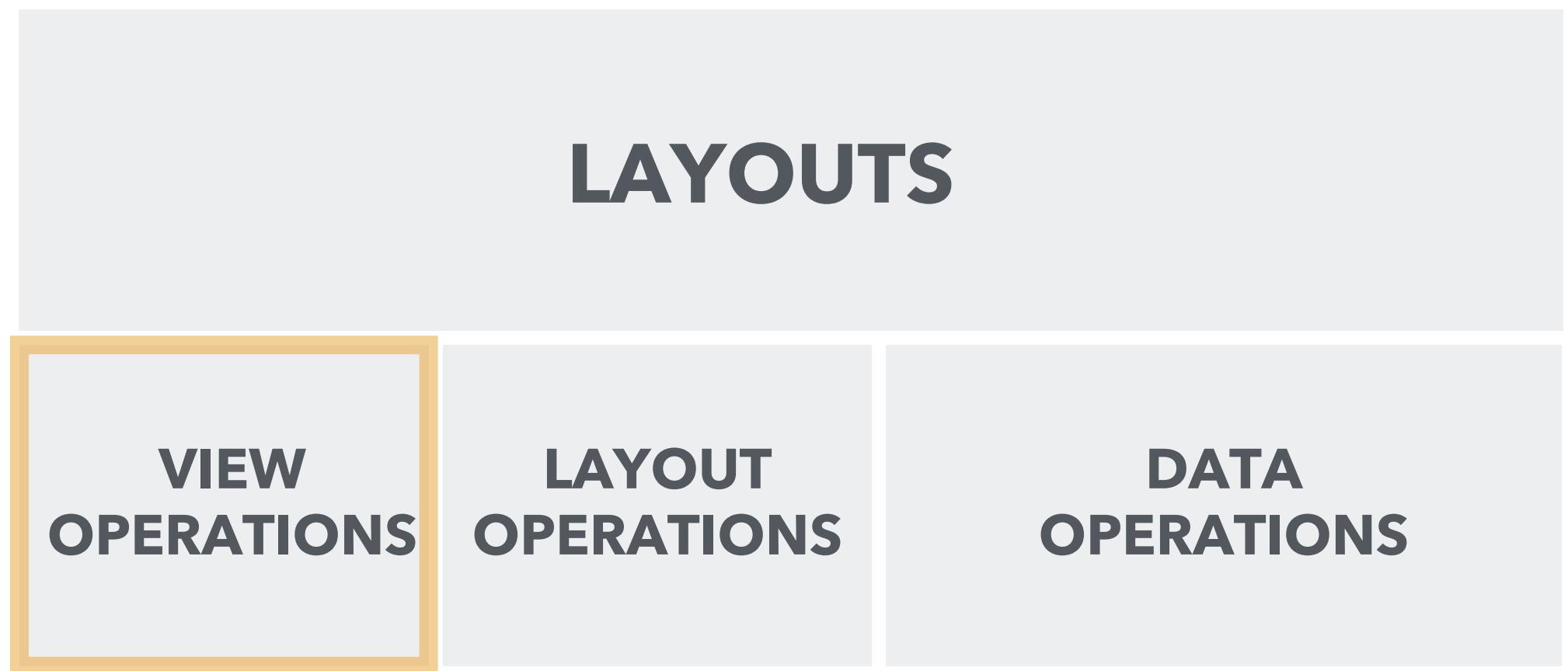


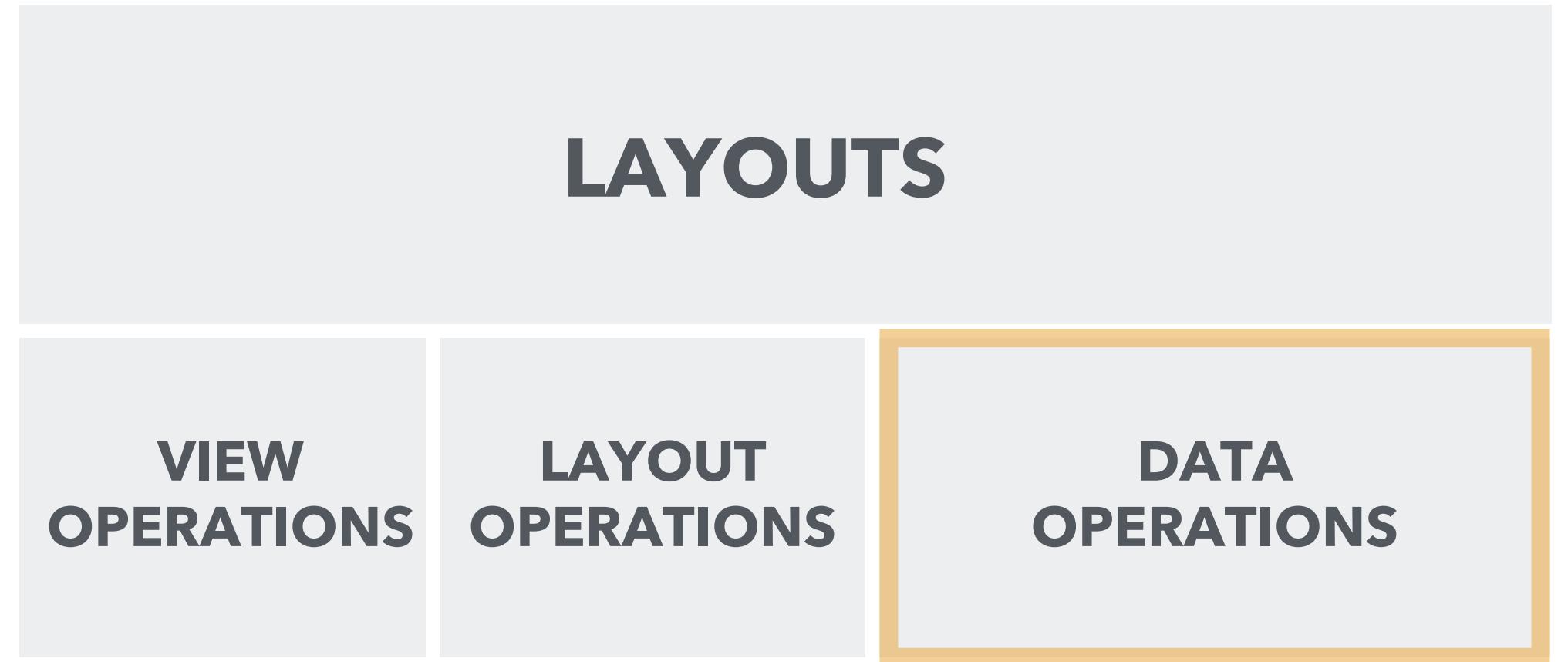


Small Multiples



Juxtaposed Views

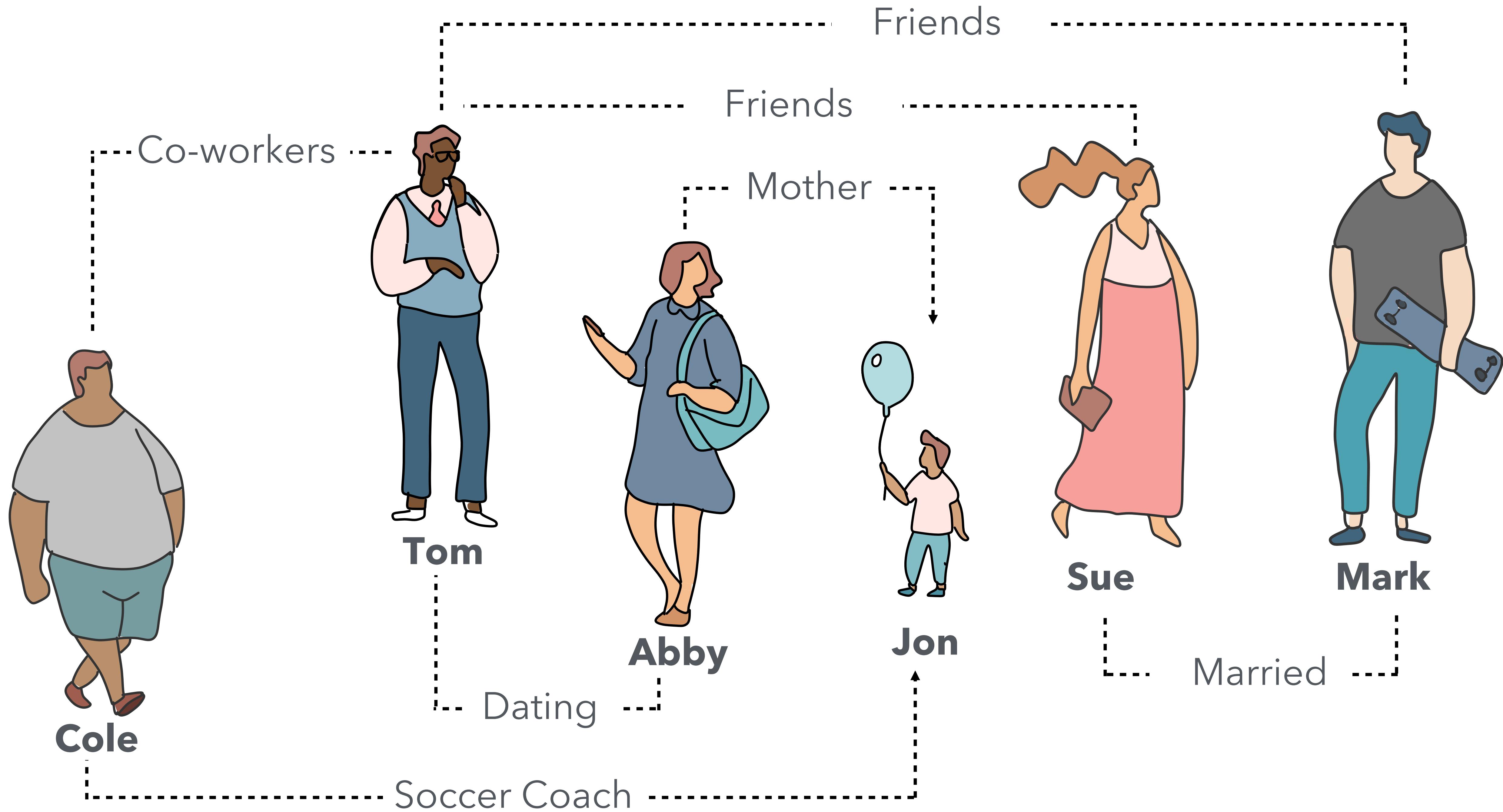


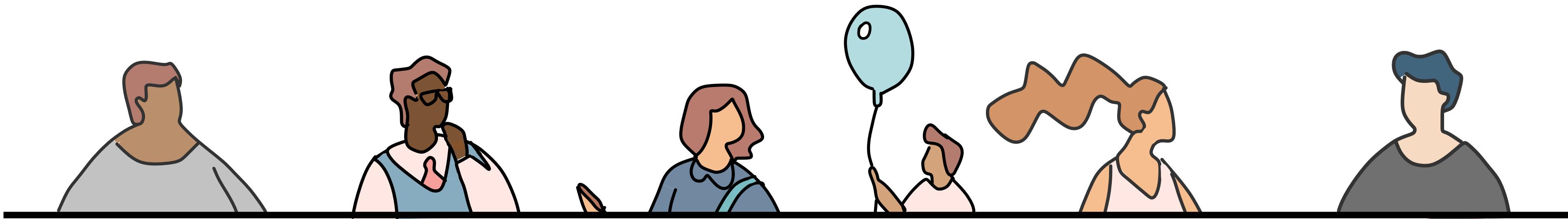


Filter Data

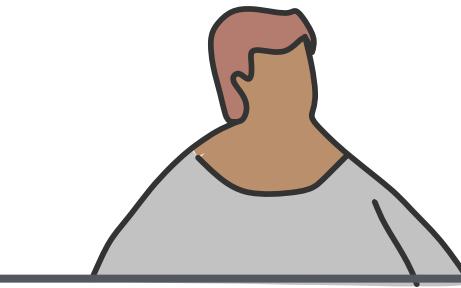
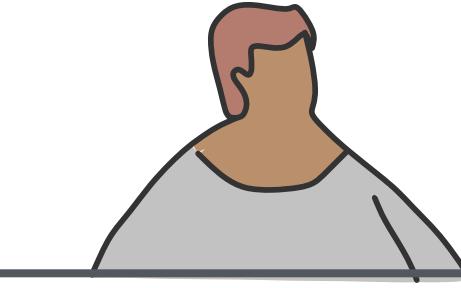
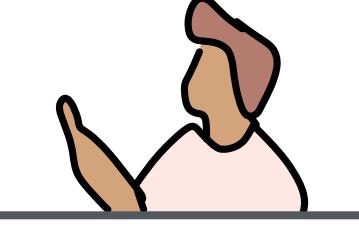
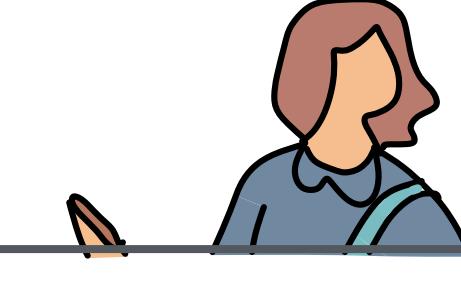
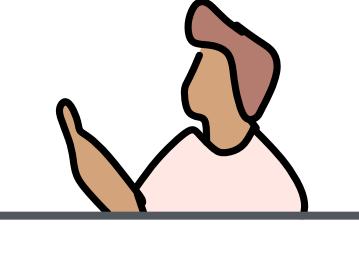
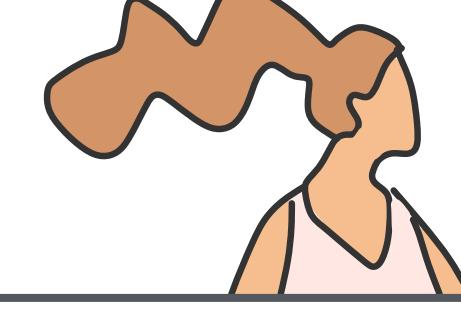
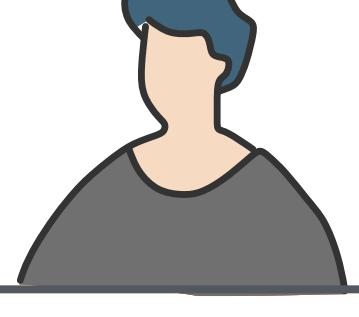
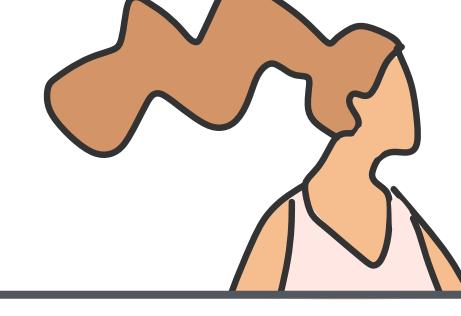
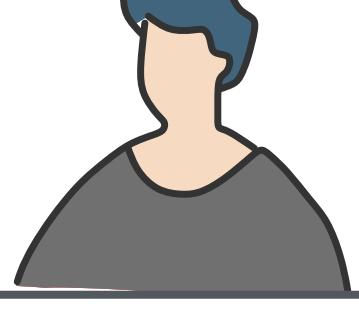
Attribute



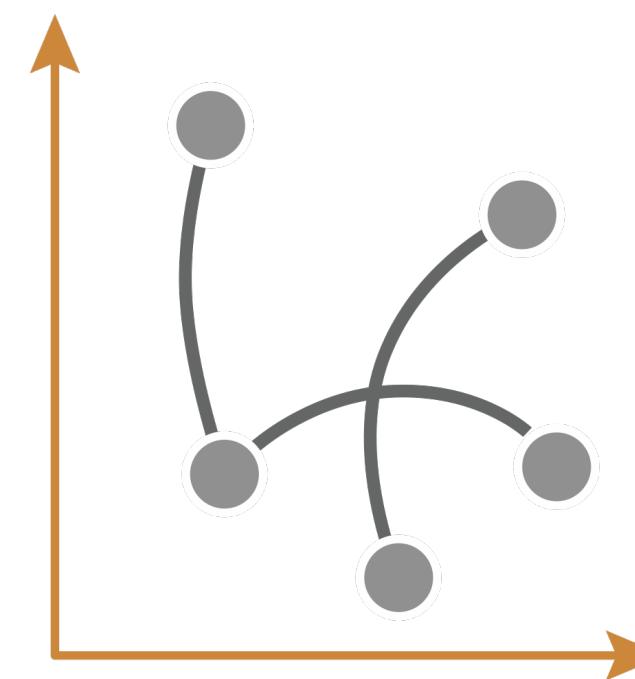
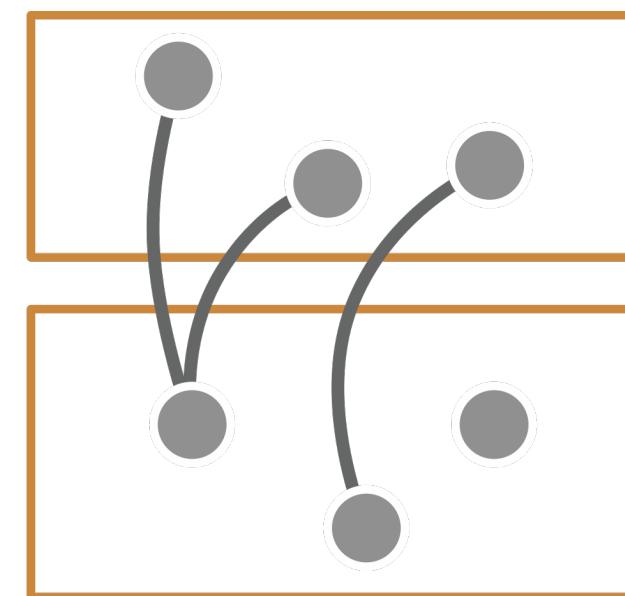
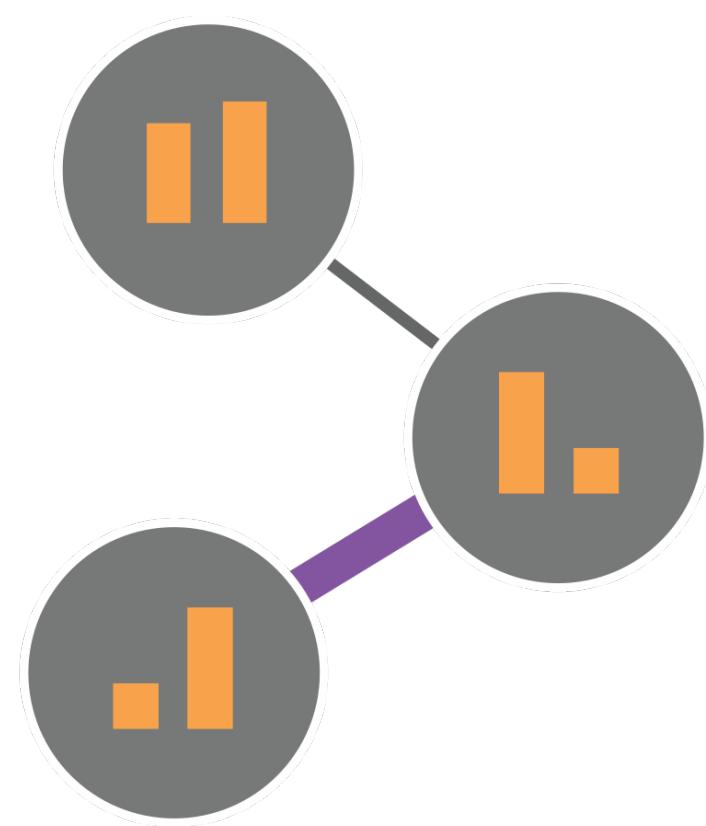




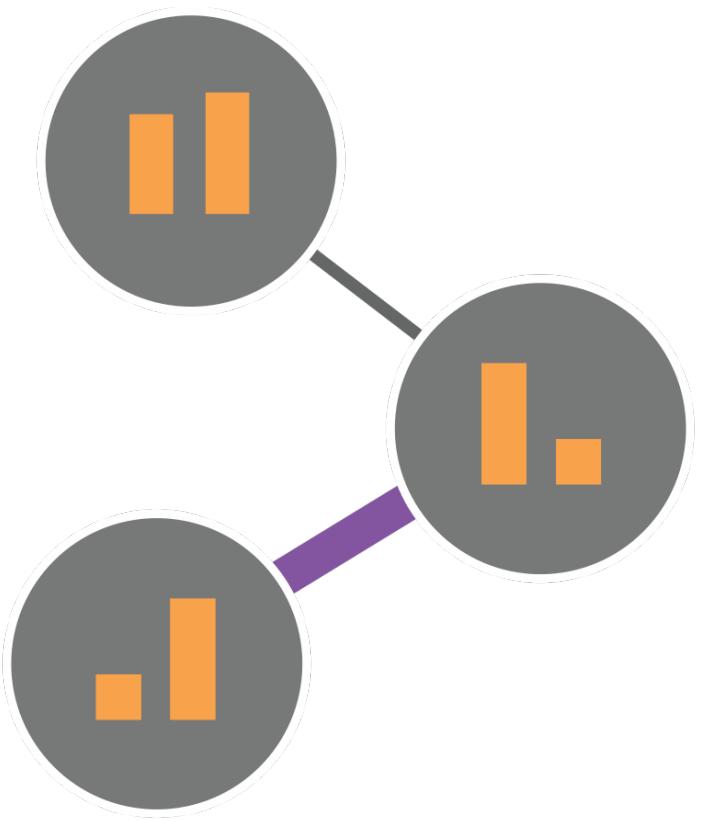
Name	Cole	Tom	Abby	Jon	Sue	Mark
Beverage	Port	Beer	Port	Coke	Coke	Beer
Day 1	1	0	4	3	3	5
Day 2	0	2	5	3	5	5
Day 3	4	1	2	2	4	3

Source	Target	Type	Duration
		Co-workers	3 years
		Soccer Coach	2 years
		Dating	1 year
		Mother / Son	7 years
		Friends	12 years
		Friends	3 years
		Married	6 years

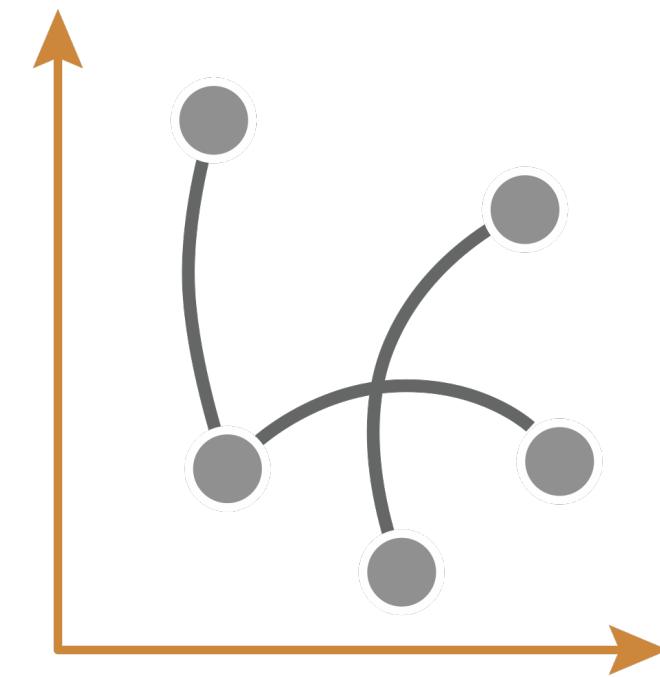
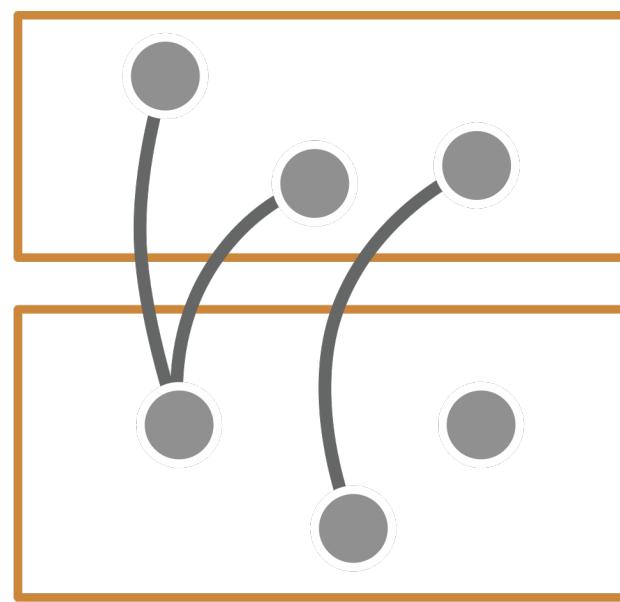
Node-Link Layouts



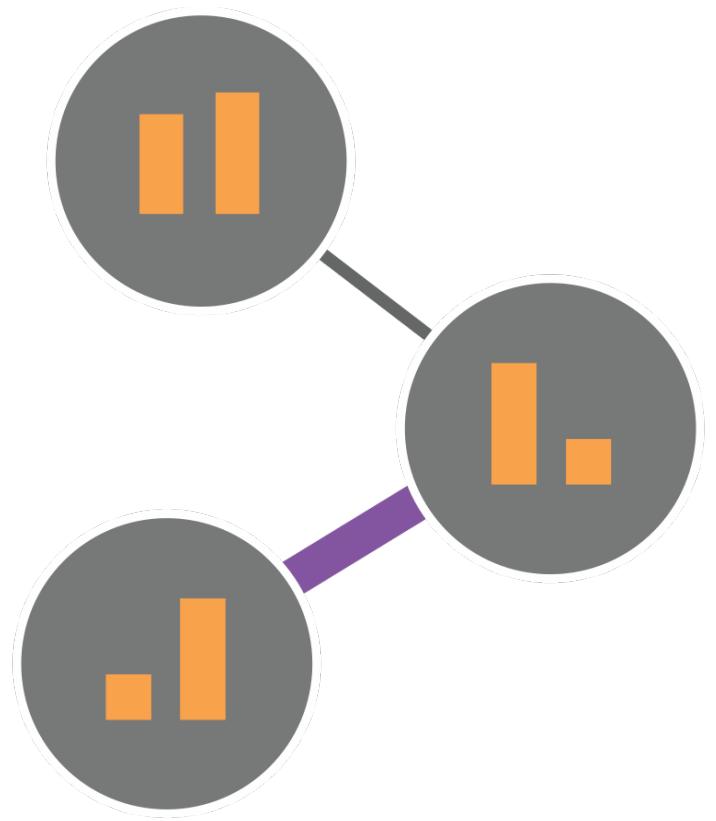
Topology Driven Layout



Attribute Driven Layouts

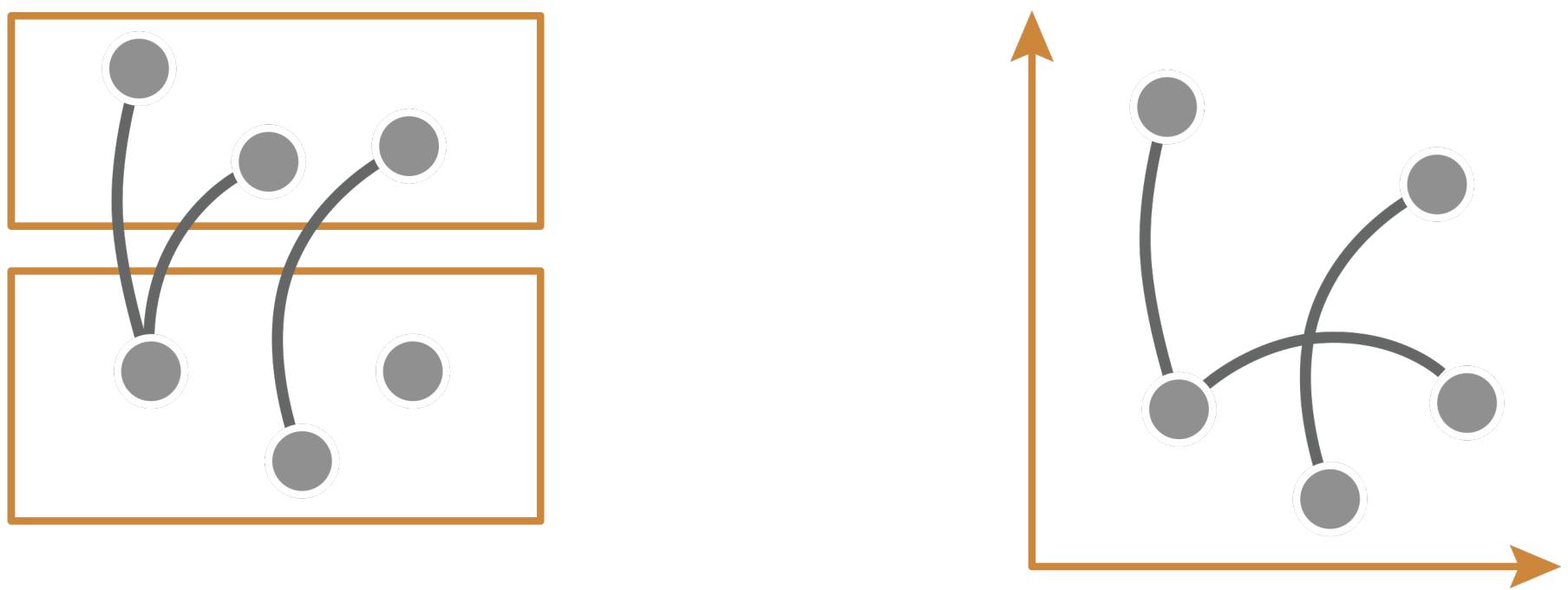


Topology Driven Layout

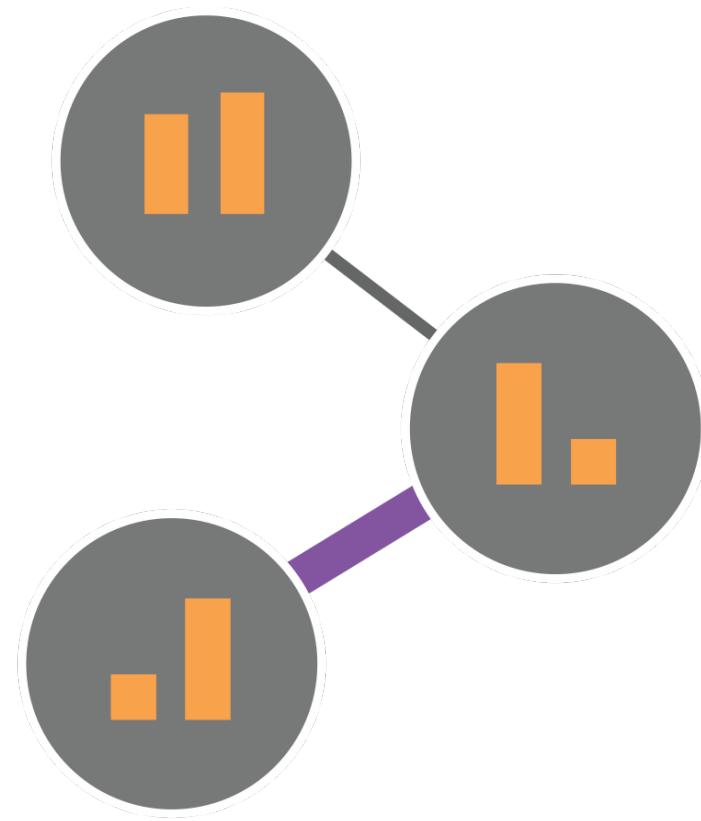


On-Node / On-Edge
Encoding

Attribute Driven Layouts

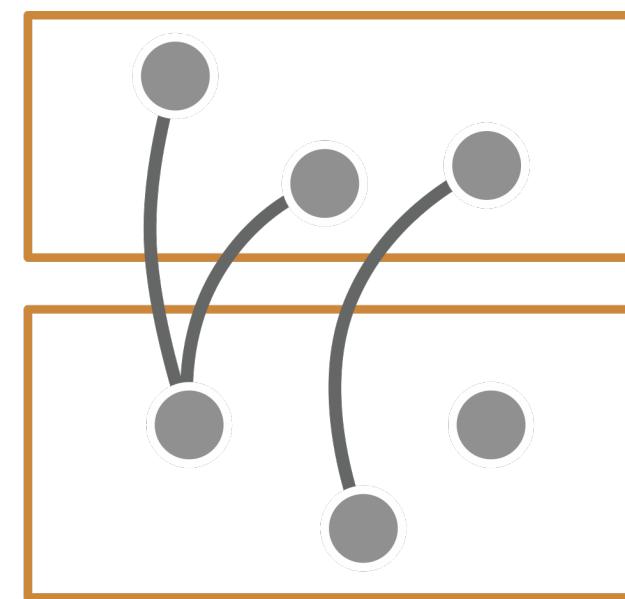


Topology Driven Layout

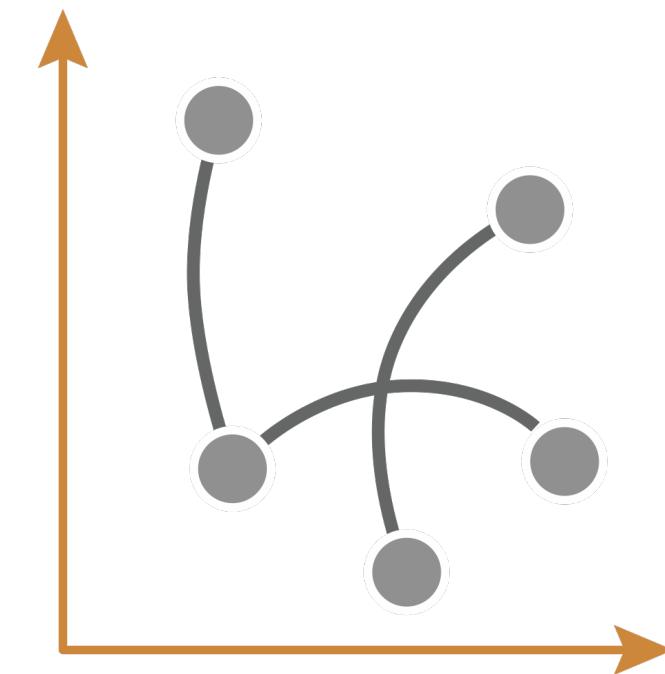


On-Node / On-Edge
Encoding

Attribute Driven Layouts

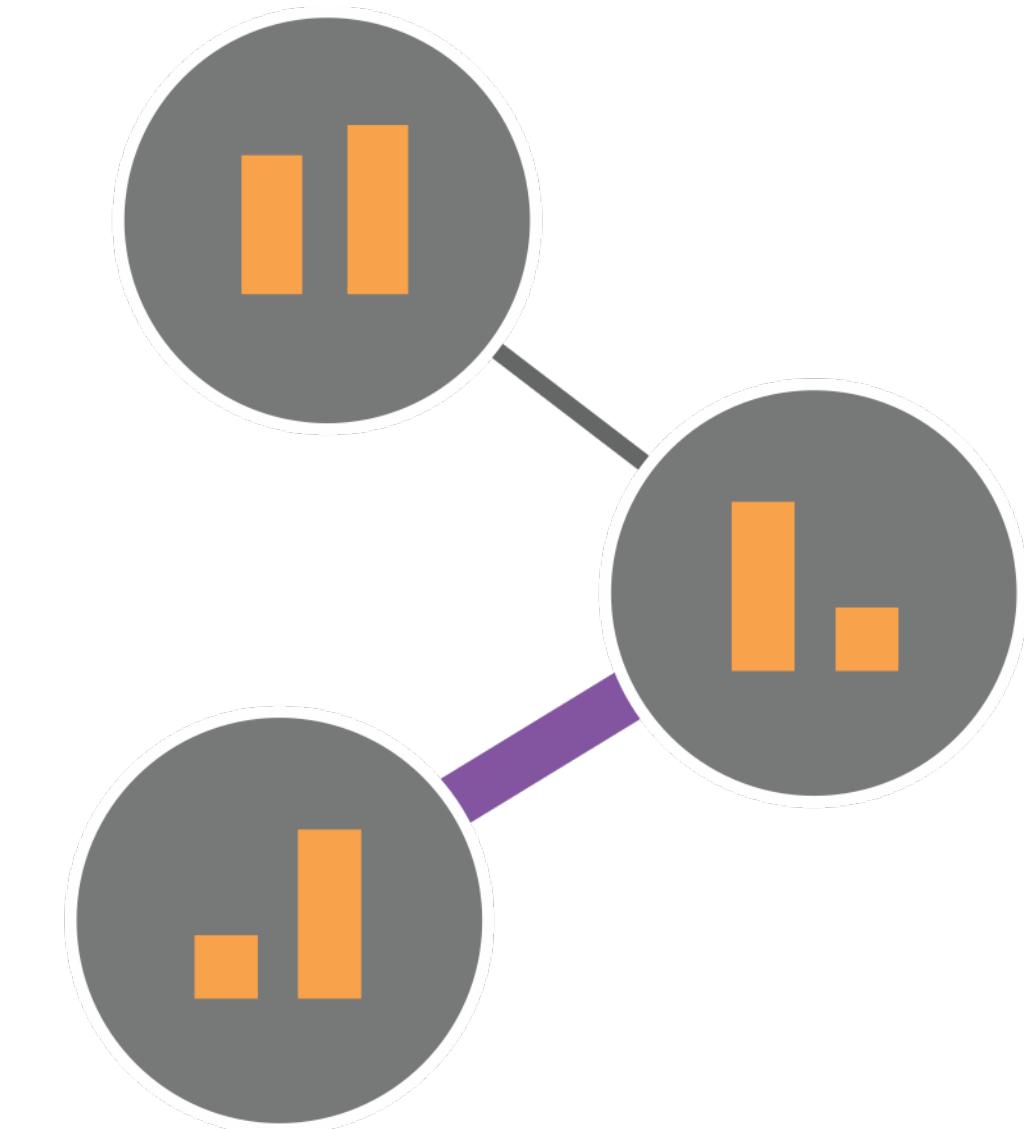


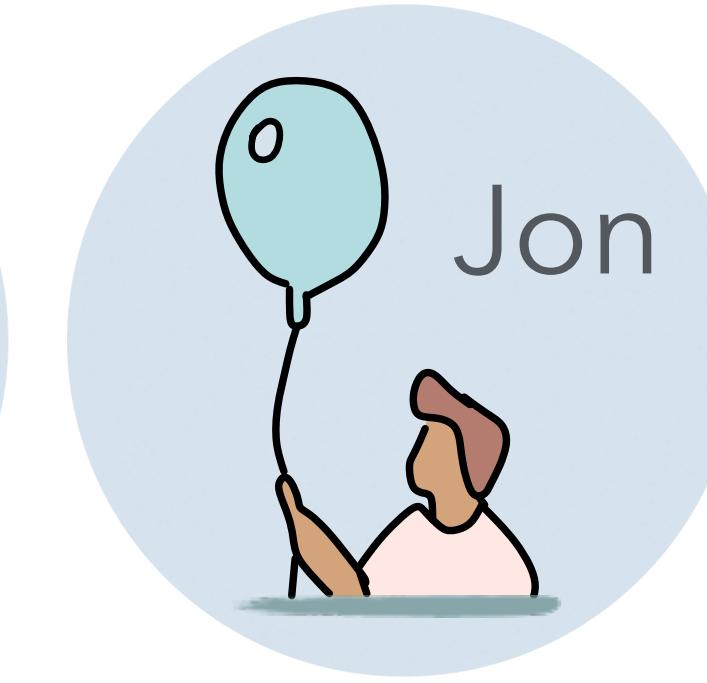
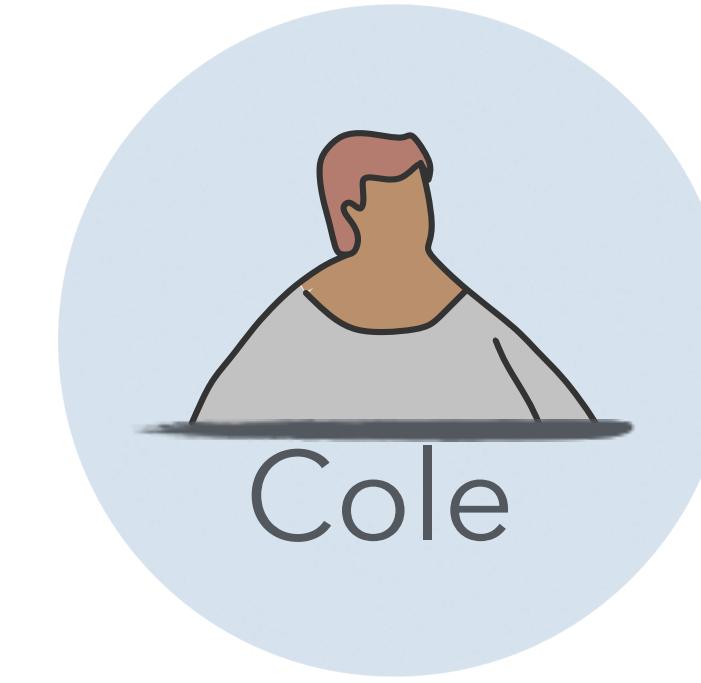
Attribute-Driven
Faceting

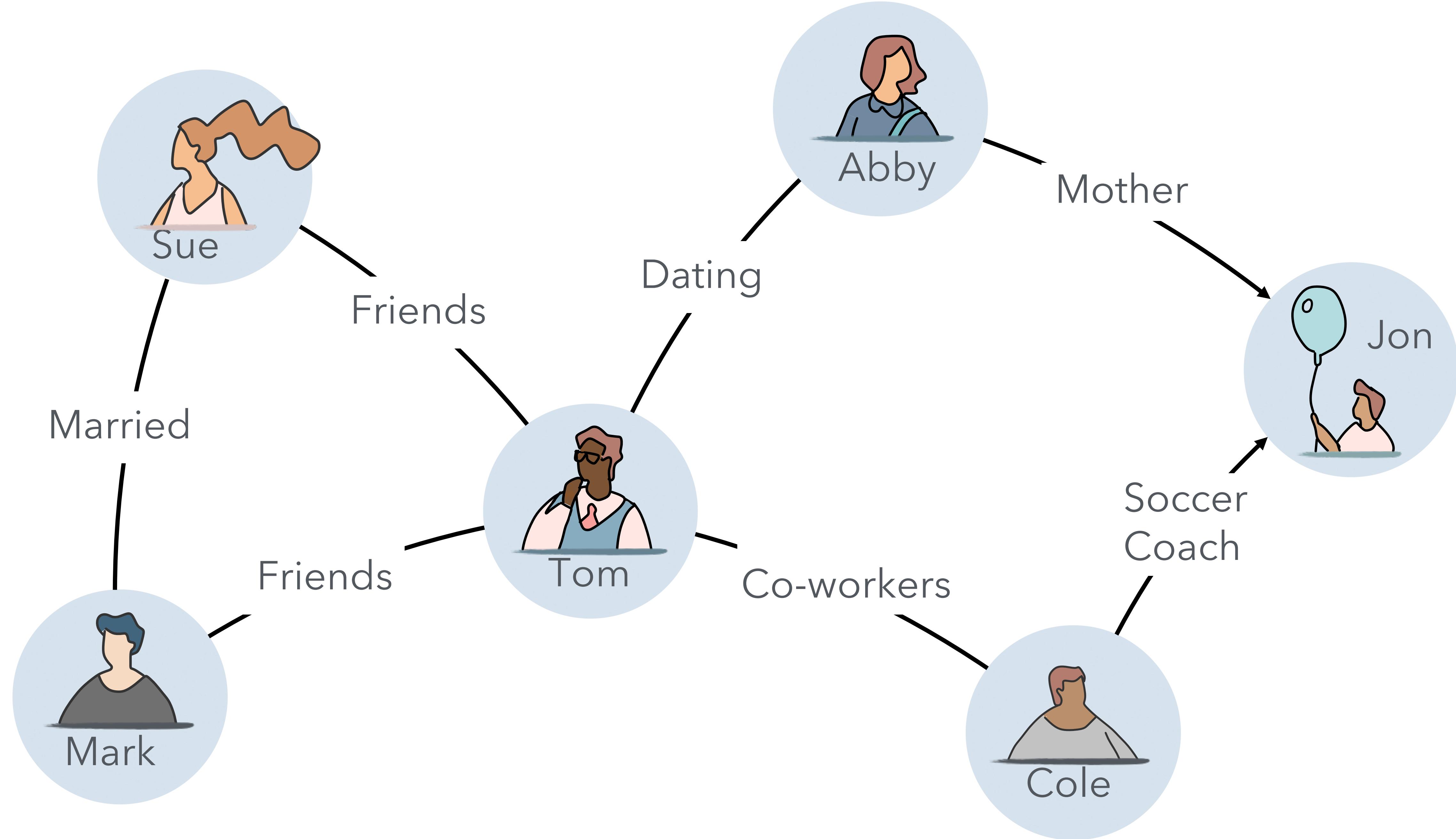


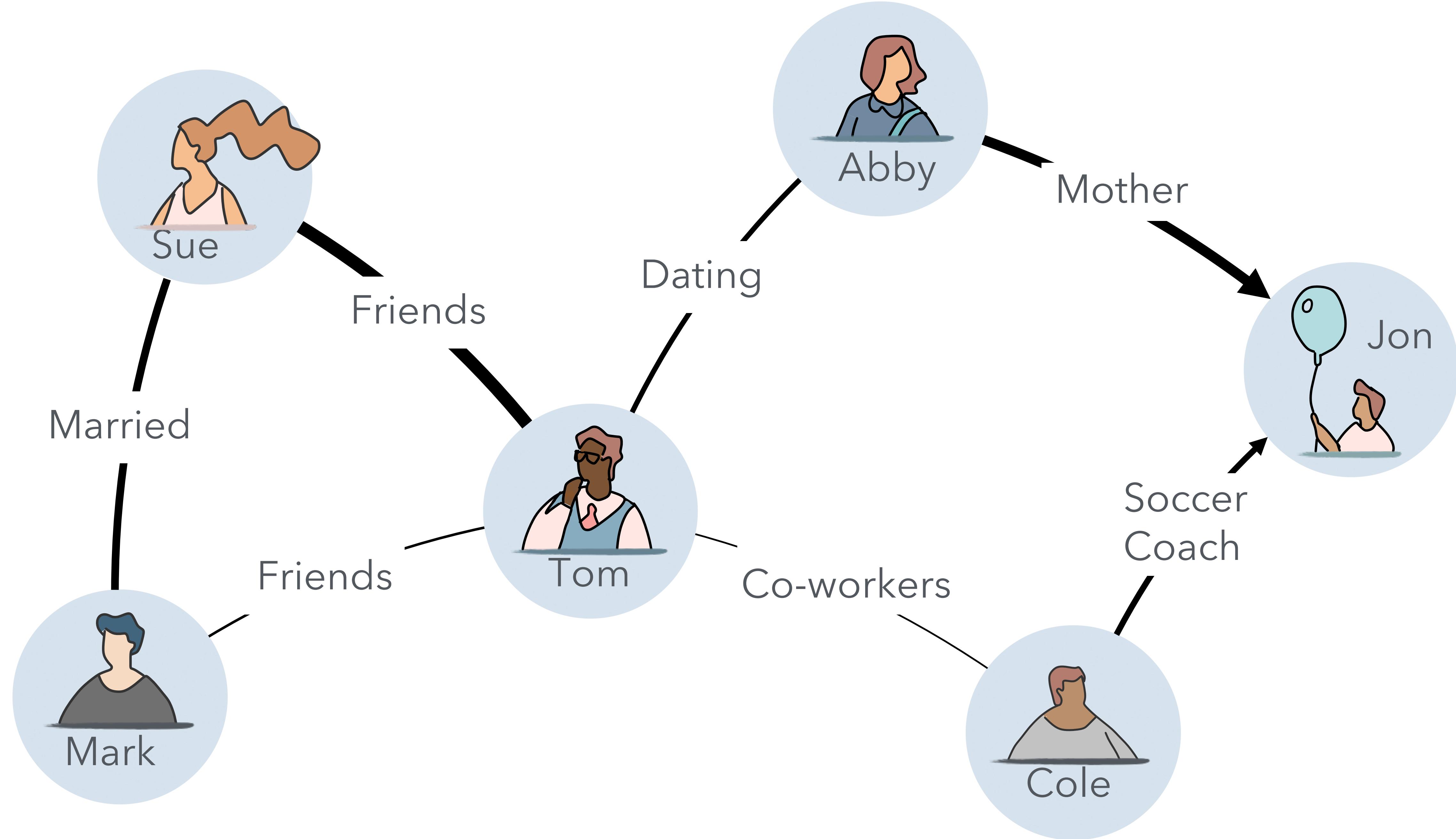
Attribute-Driven
Positioning

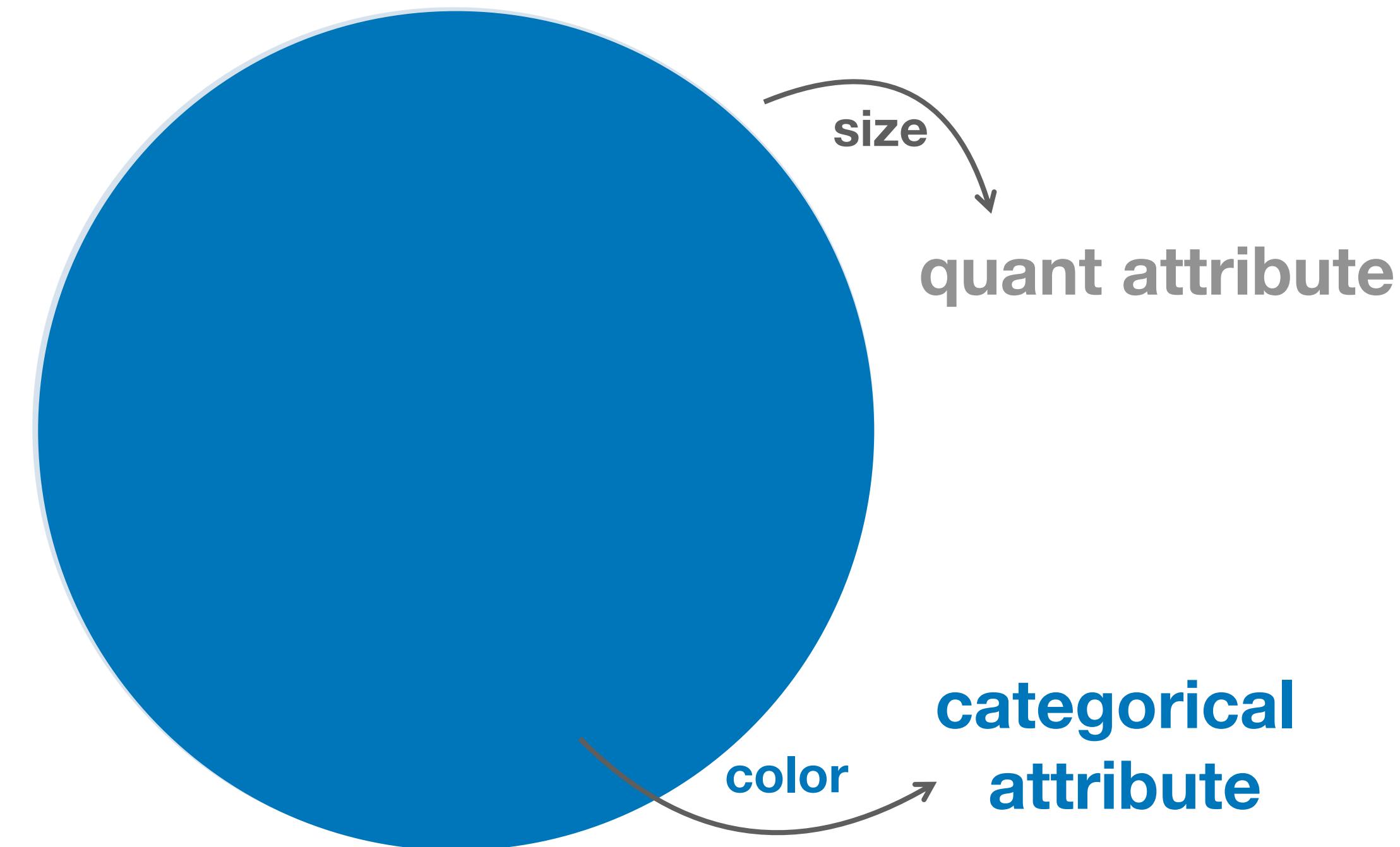
On-Node / On-Edge Encoding

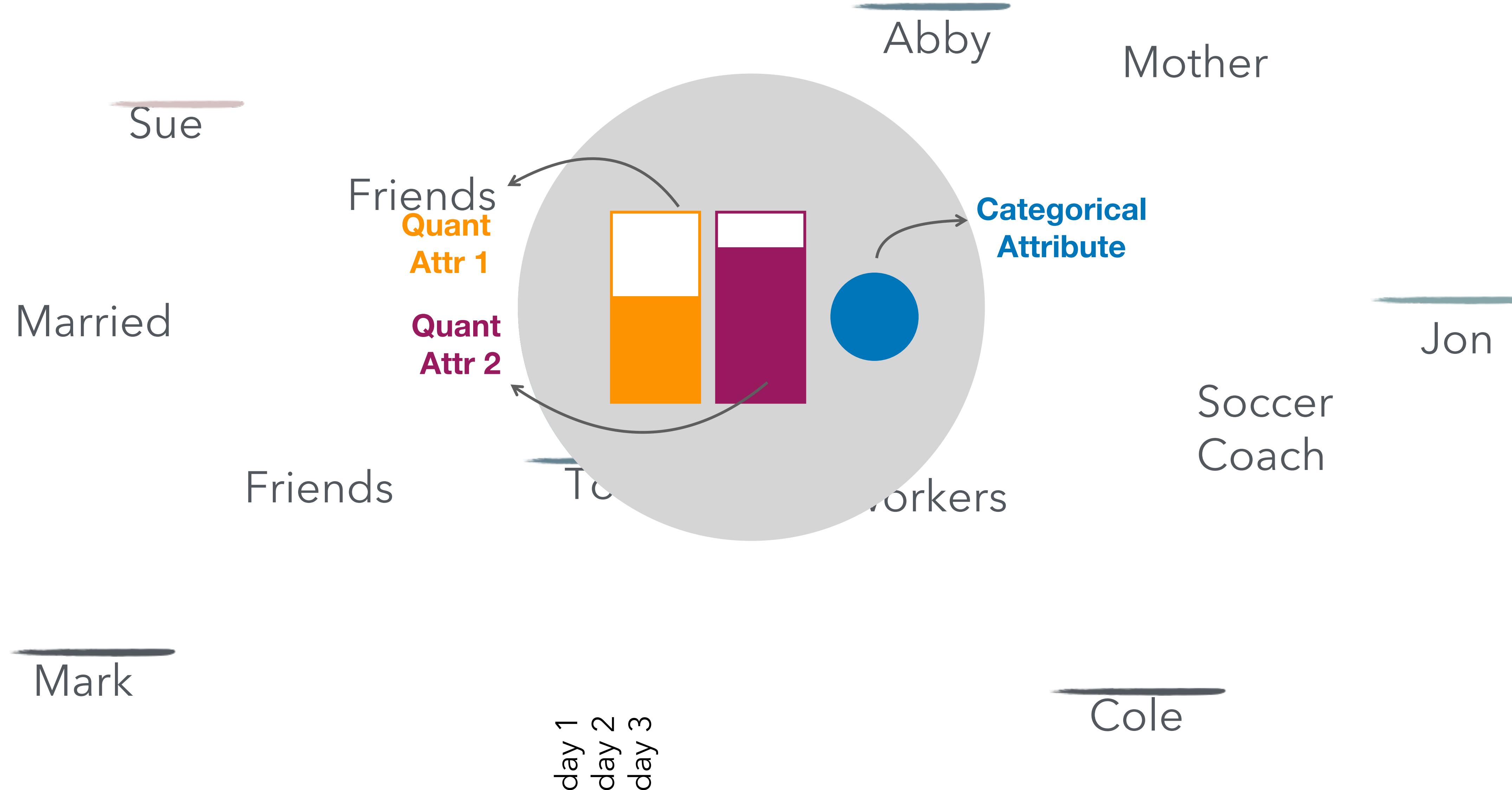


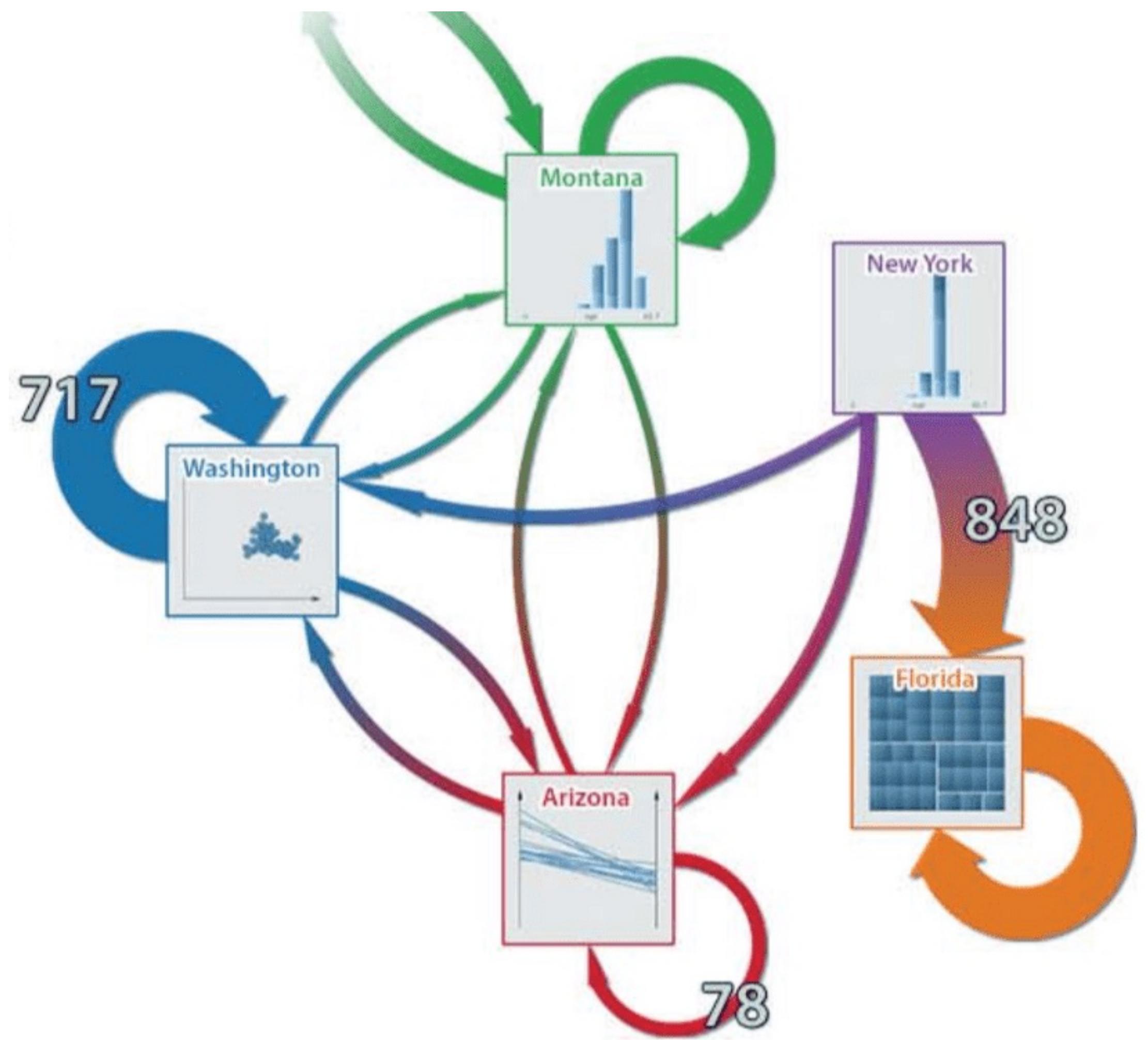




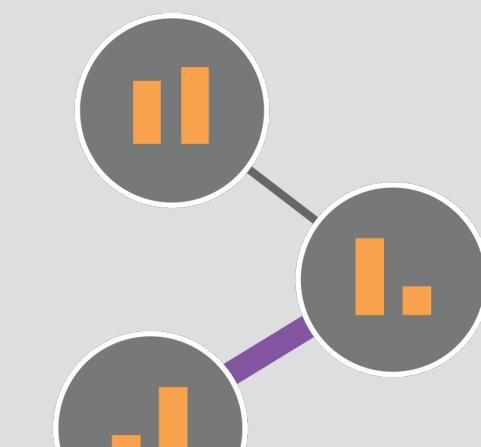




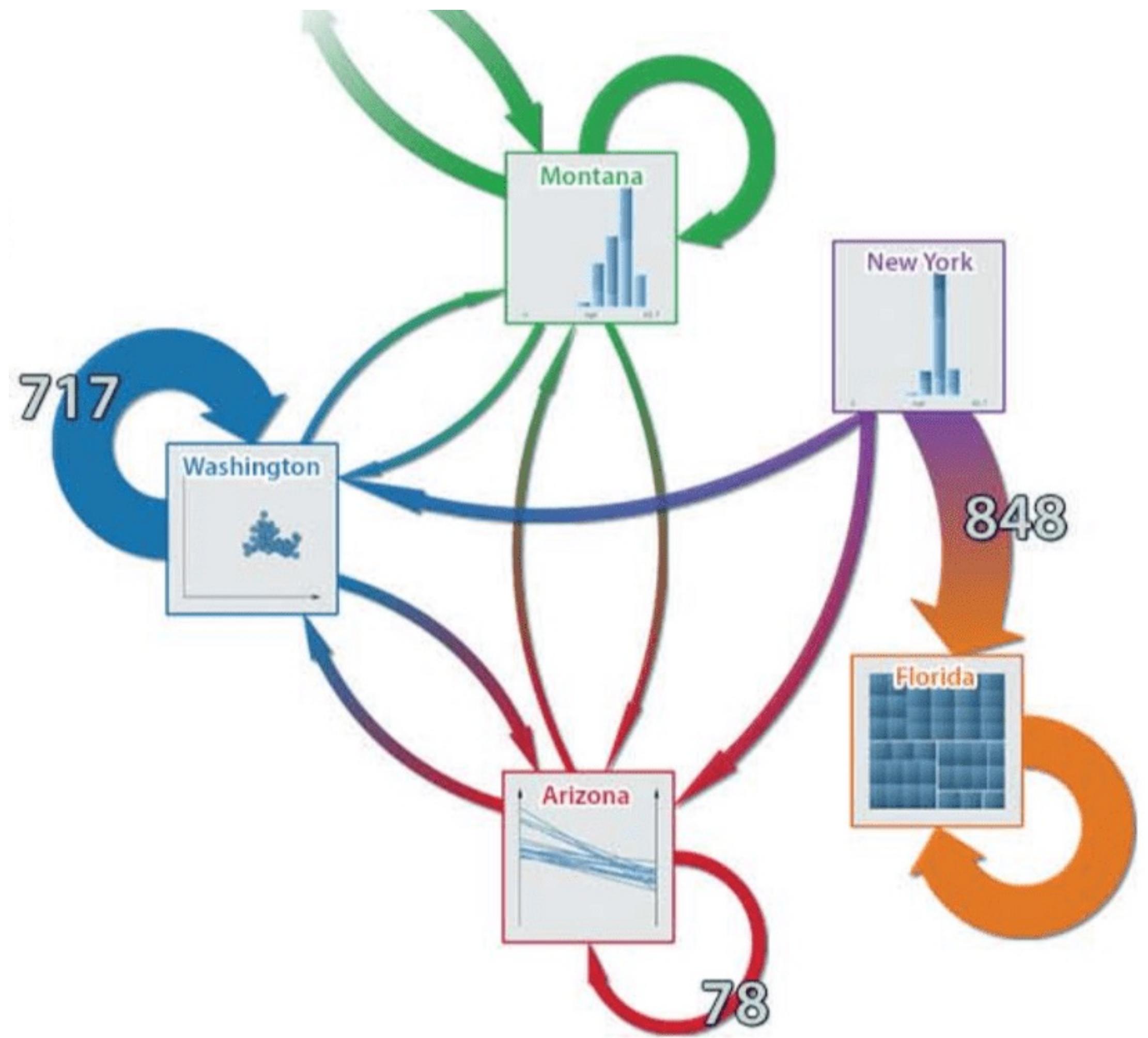




Elzen and Wijk, 2014



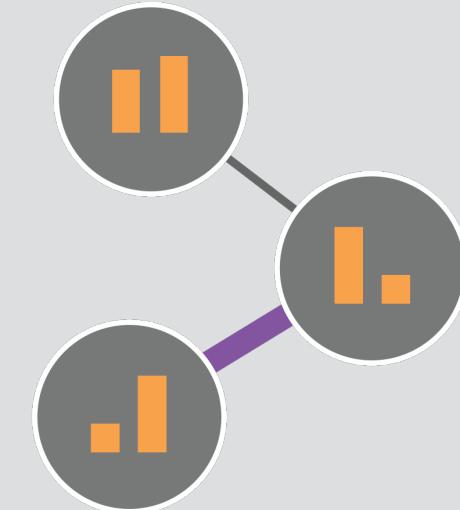
On-Node / On-Edge
Encoding



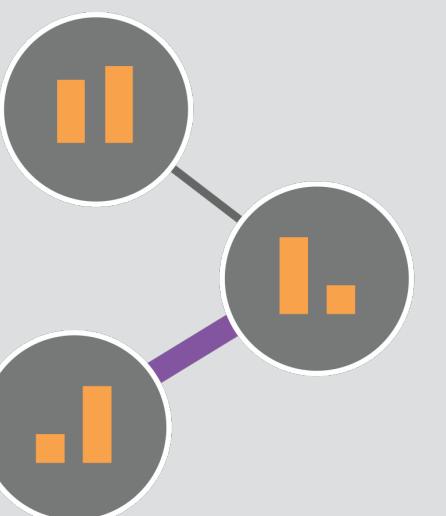
Elzen and Wijk, 2014



Aggregating Nodes/Edges



On-Node / On-Edge
Encoding



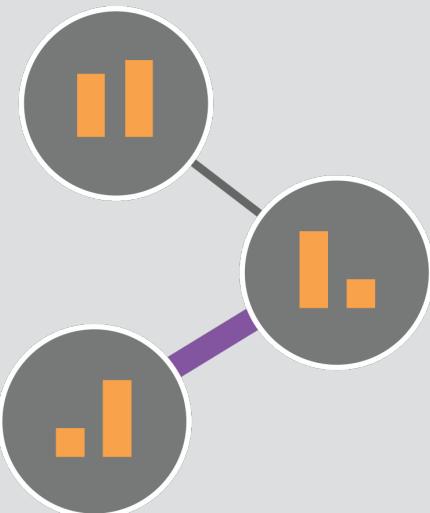
On-Node / On-Edge
Encoding



Is easily understood by most users
Works well for all types of networks



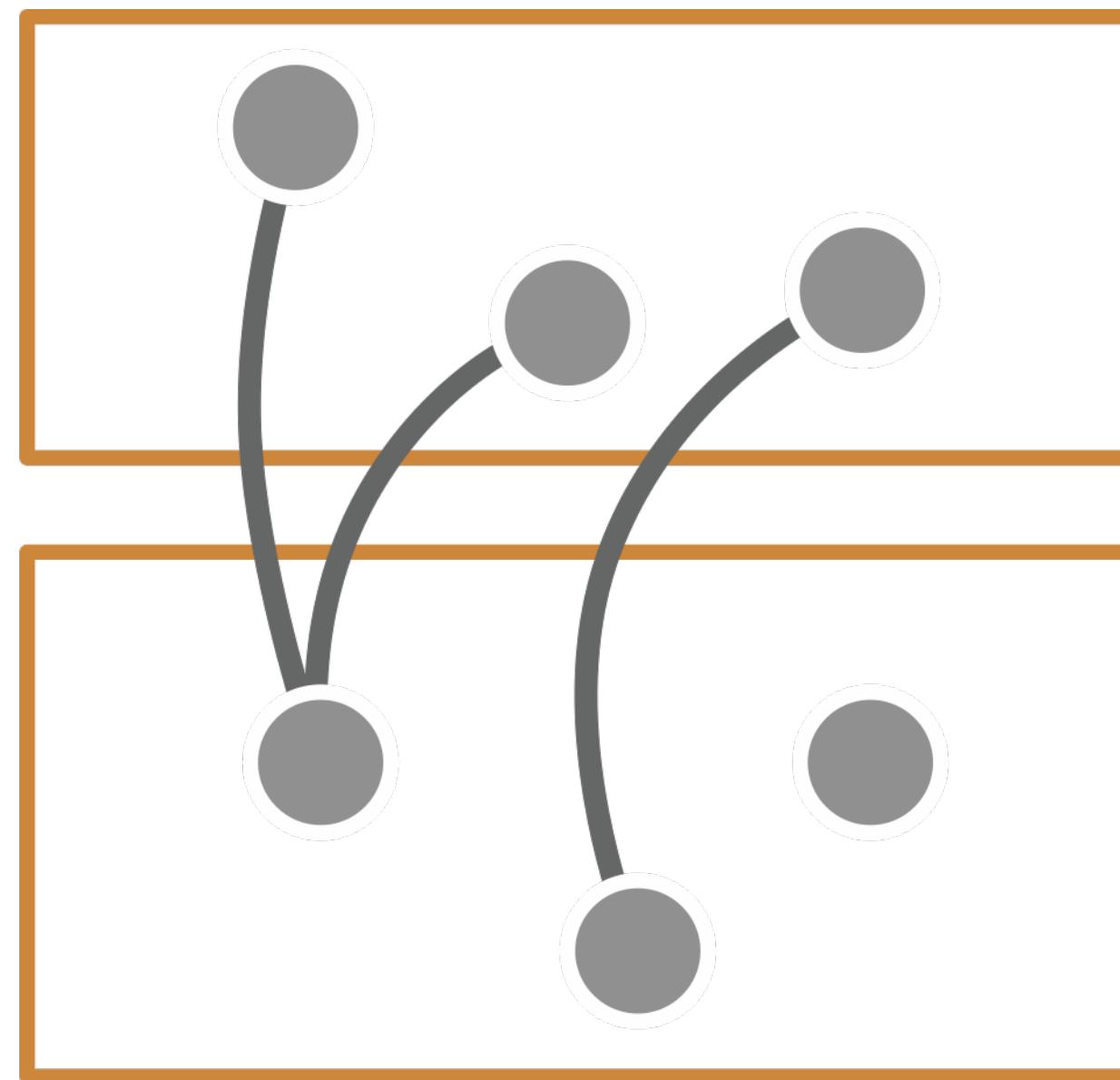
Scalability.
Node size leaves little space to encode attributes.



On-Node / On-Edge
Encoding

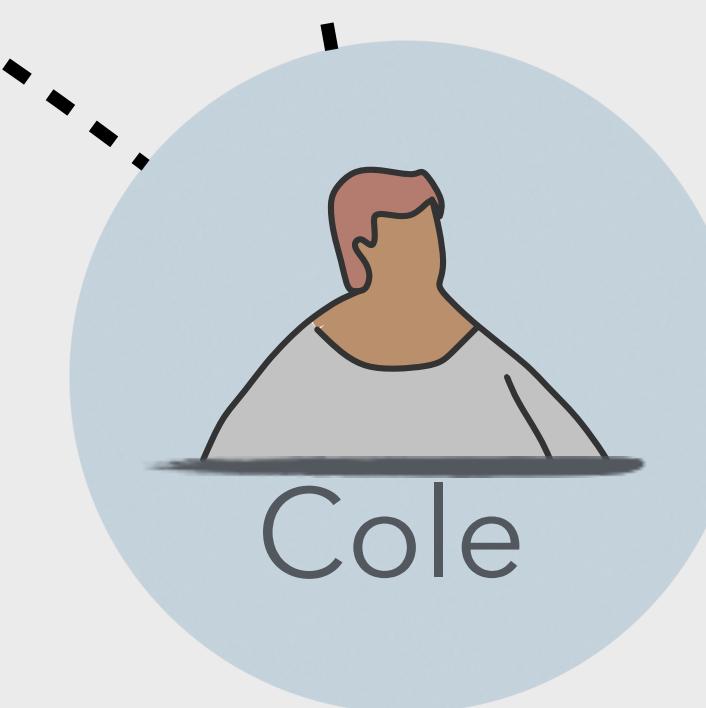
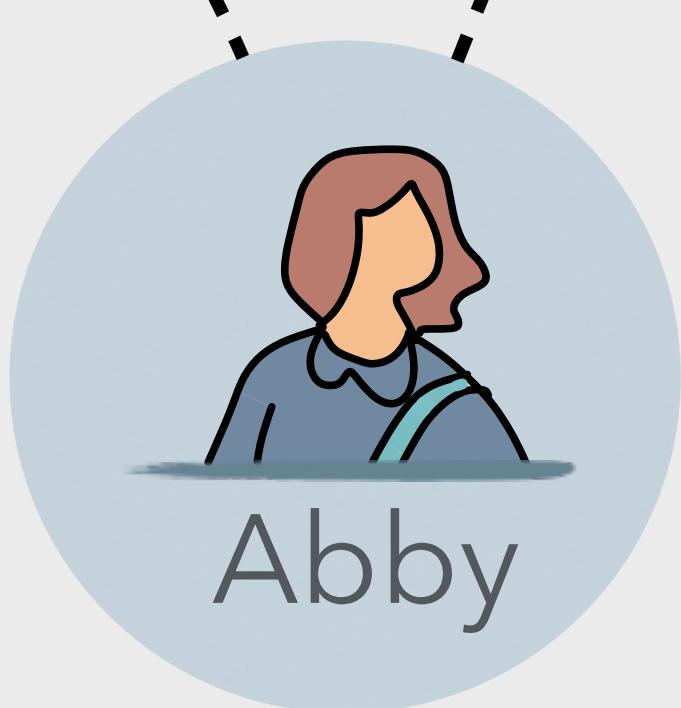
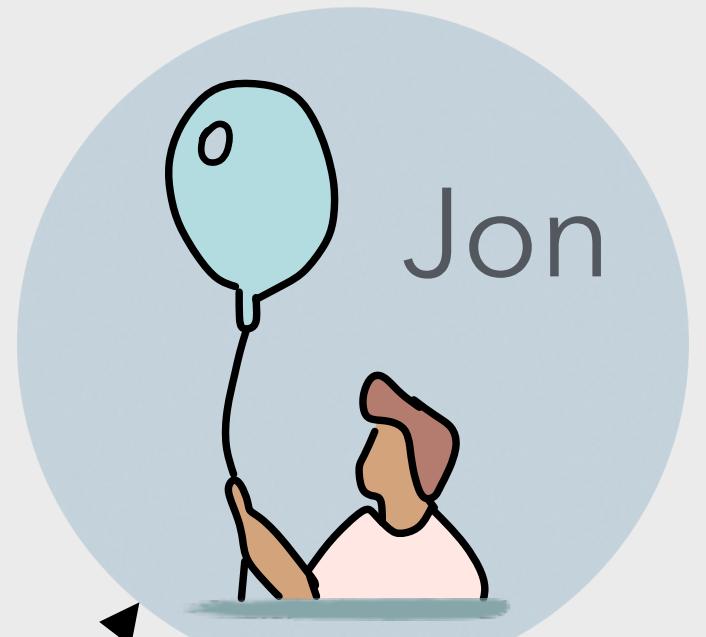
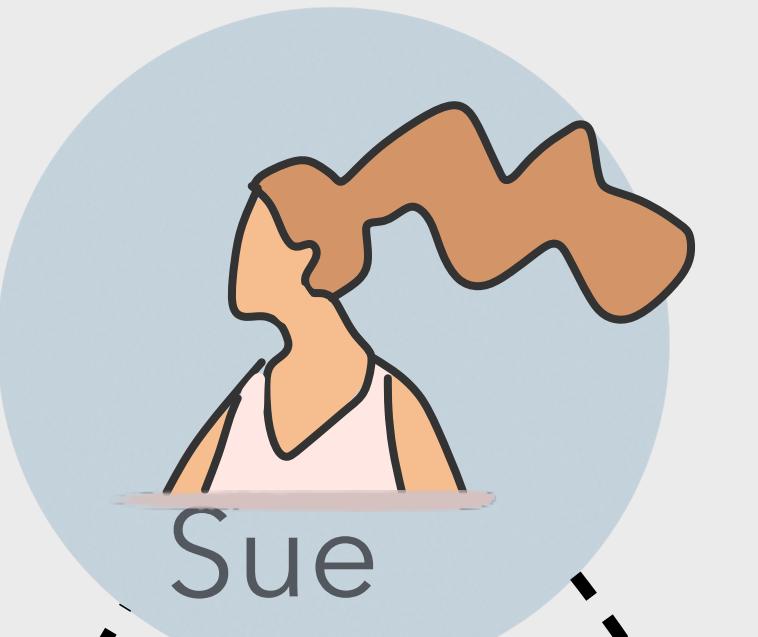
Recommended for small networks when only a few (usually under five) attributes on the nodes are shown, or in combination with a zooming/filtering strategy

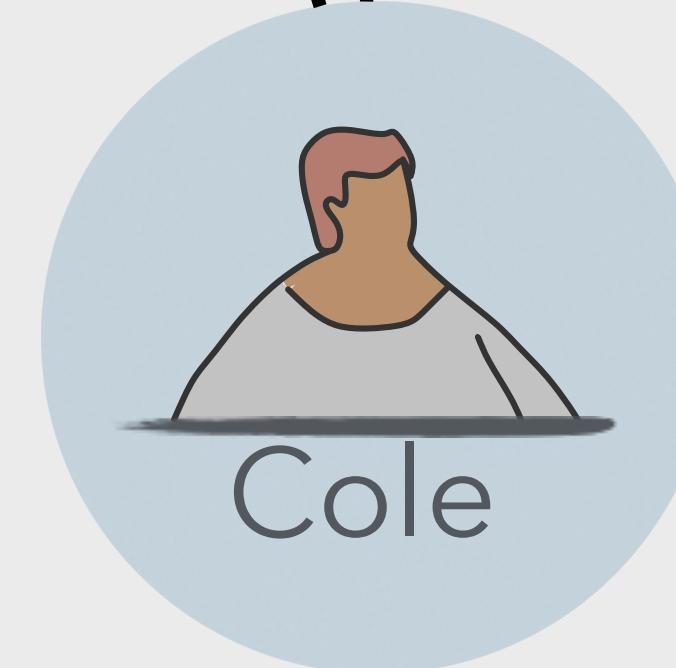
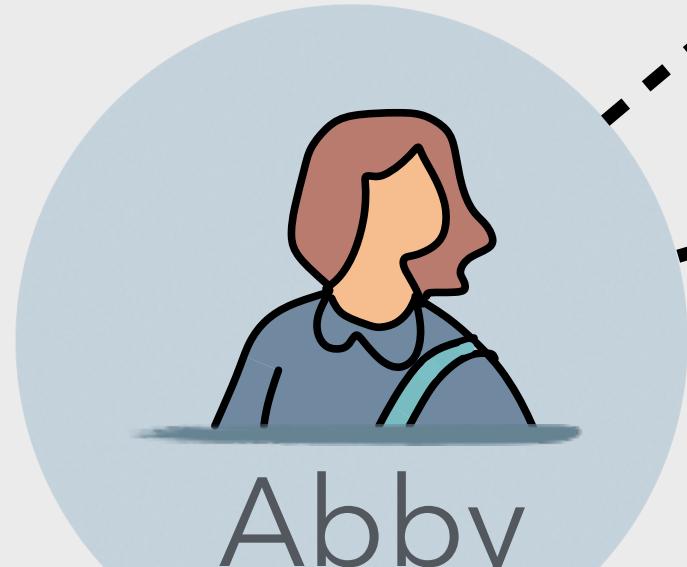
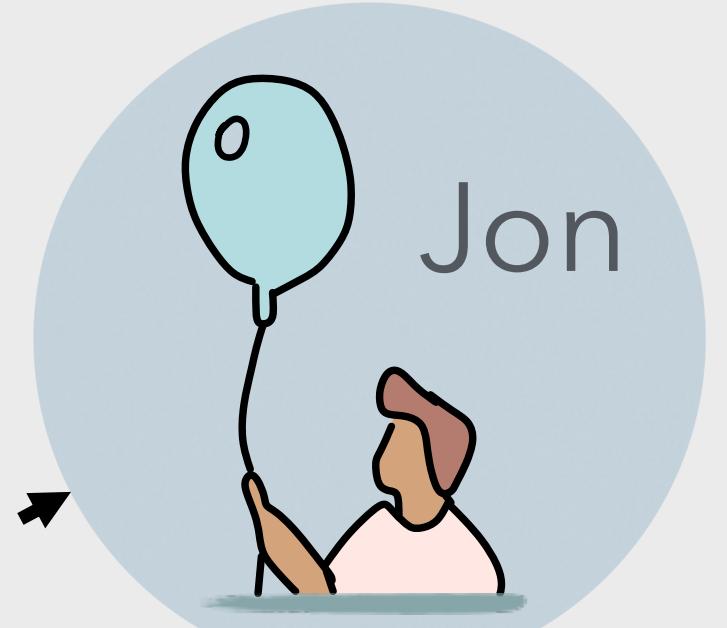
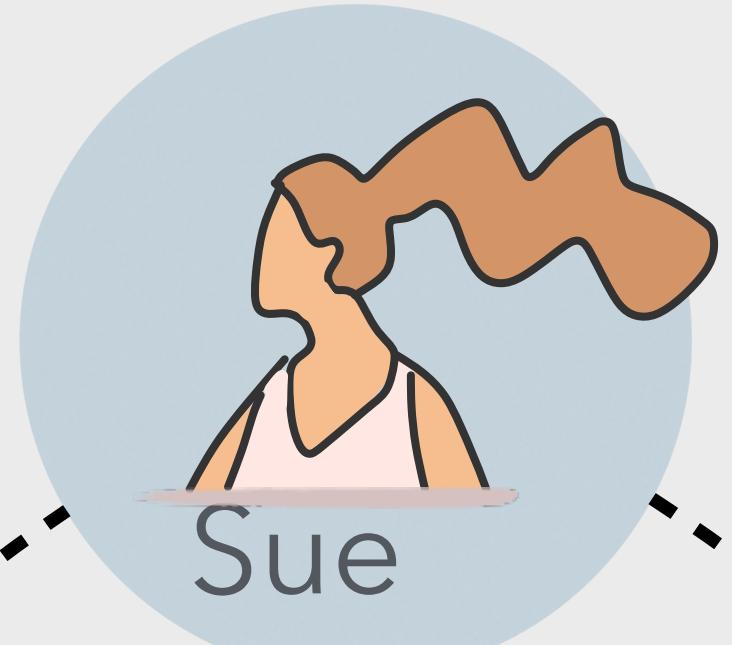
Attribute-Driven Faceting



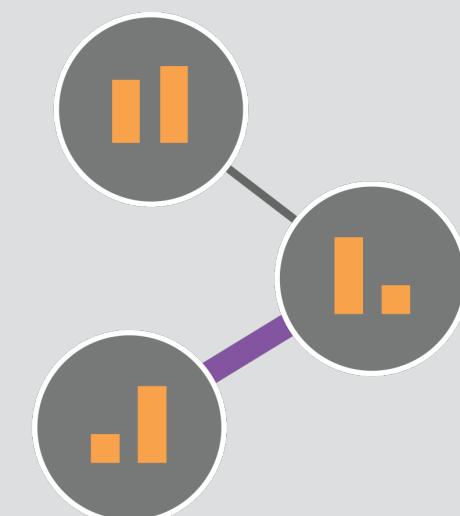
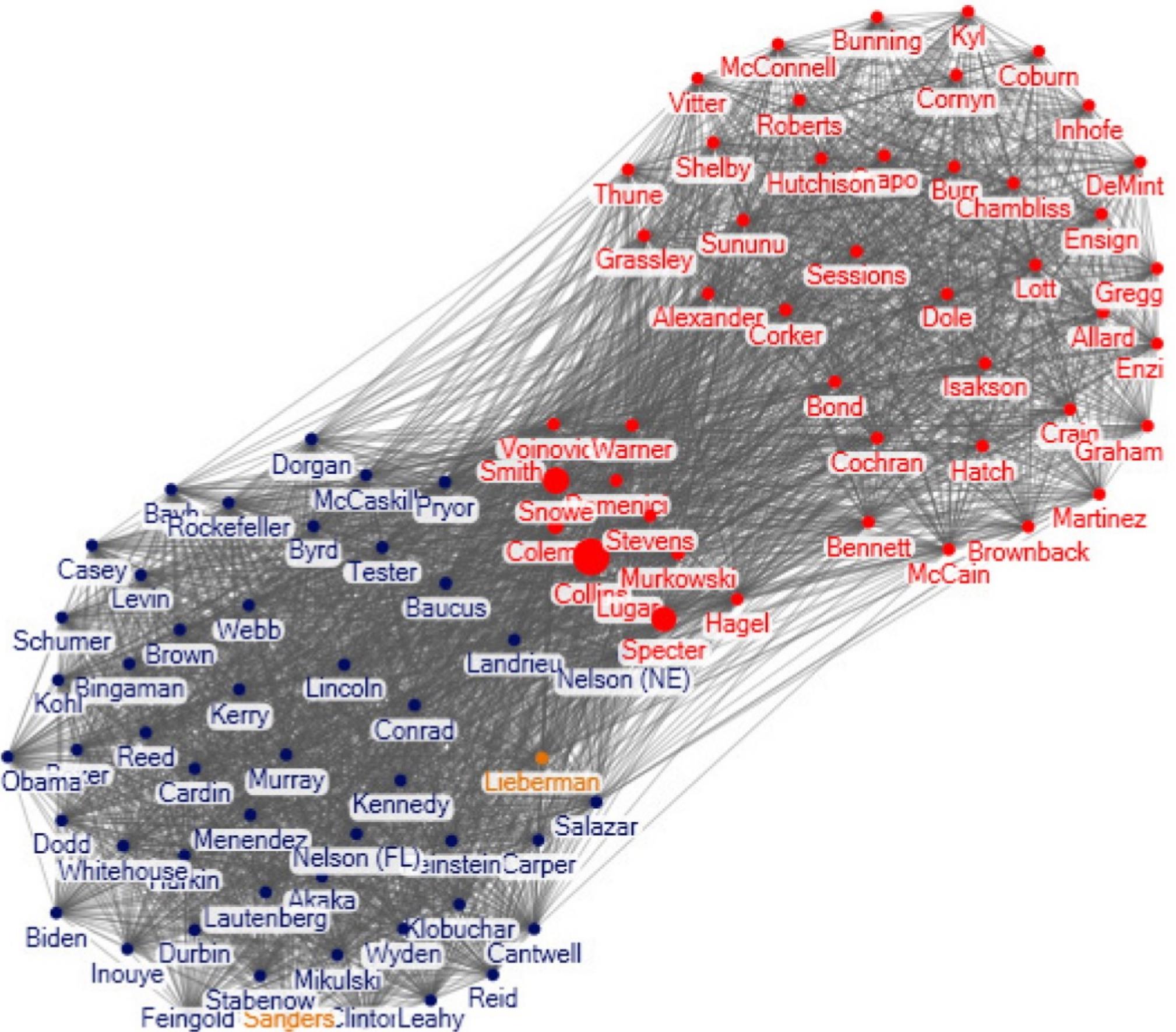






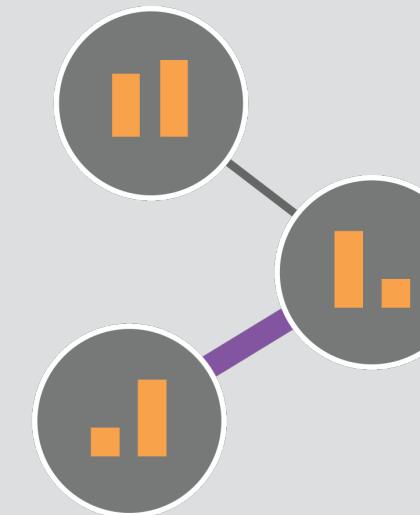
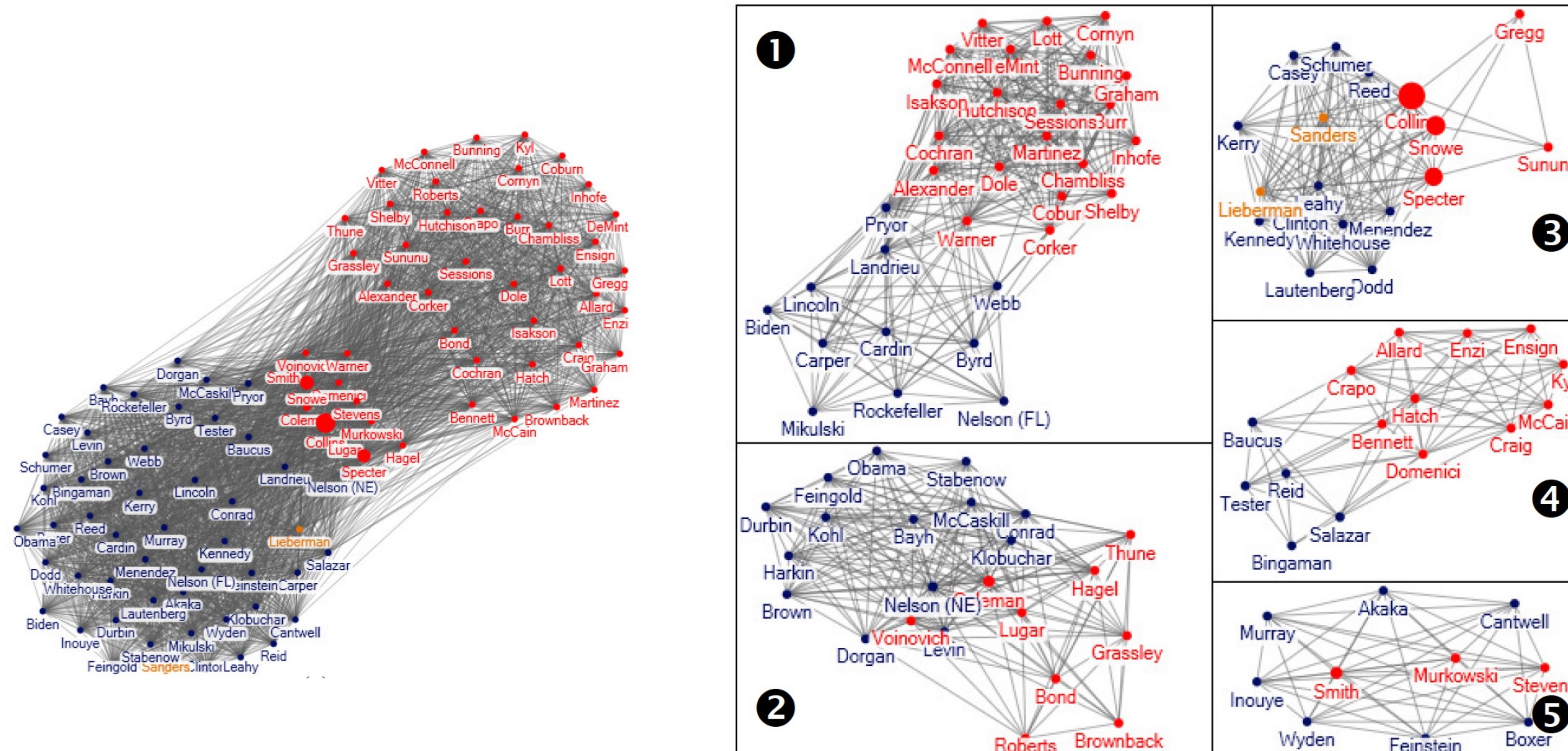


Group-in-a-box Rodrigues et al. 2011

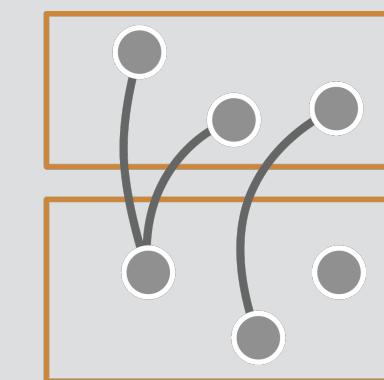


On-Node / On-Edge Encoding

Group-in-a-box Rodrigues et al. 2011

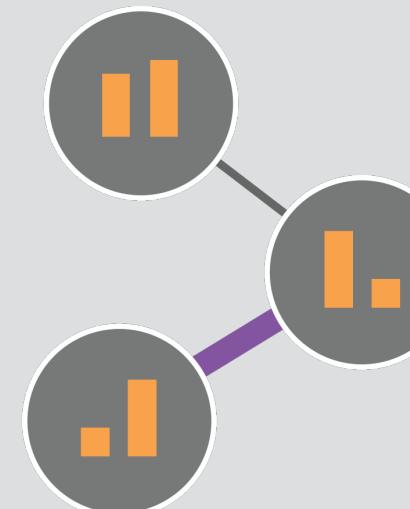
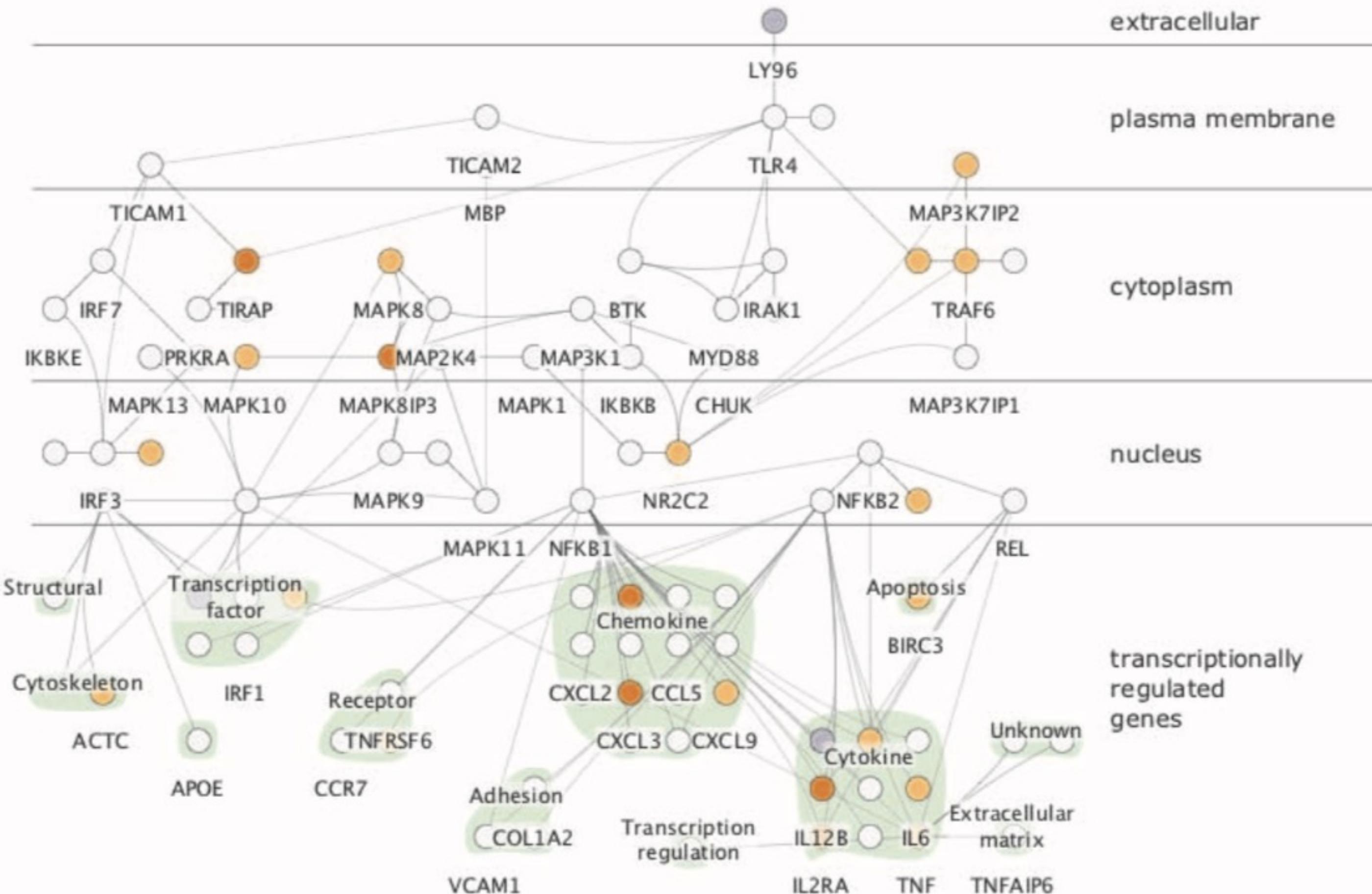


On-Node / On-Edge
Encoding

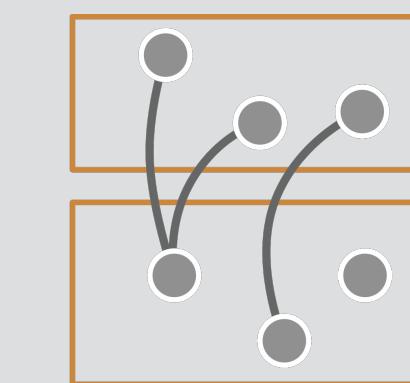


Attribute-Driven
Faceting

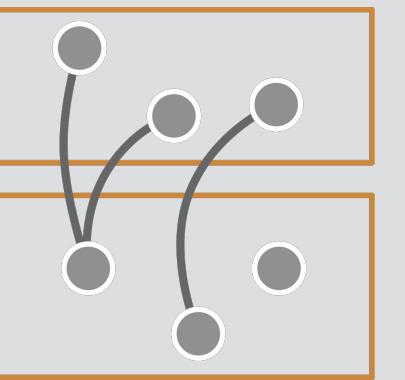
Cerebral Barsky et al. 2008



On-Node / On-Edge
Encoding

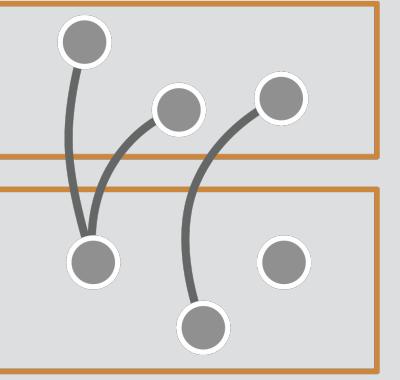


Attribute-Driven
Faceting



Attribute-Driven
Faceting





Attribute-Driven
Faceting

Well suited for networks with different node types or with an important categorical or set-like attribute.

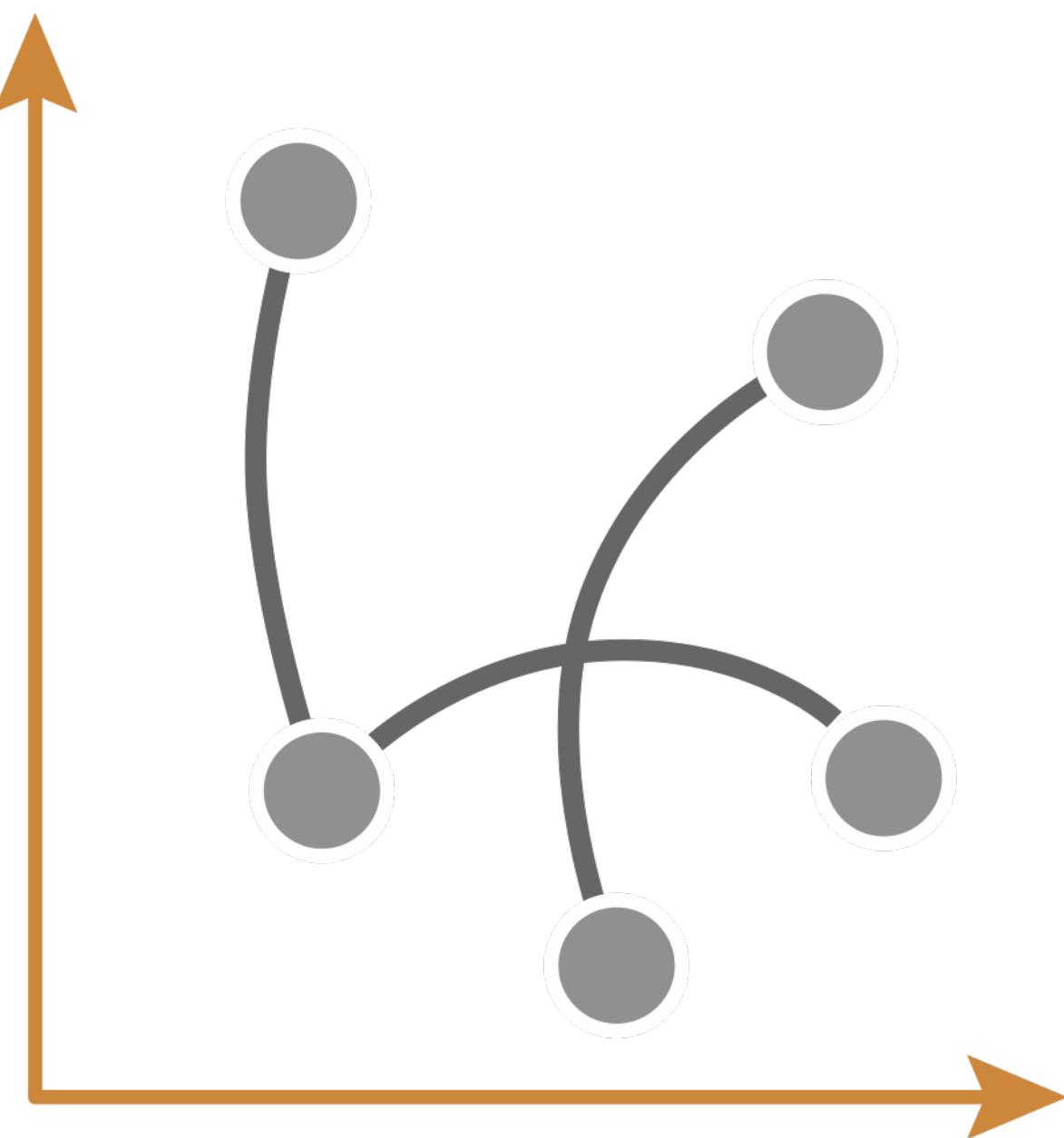


Less scalable with respect to the number of nodes and network density than node-link layouts.

Neighborhoods, paths, and clusters are not easily visible if they span different facets.

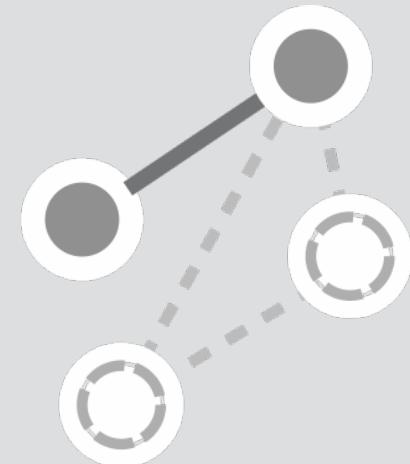
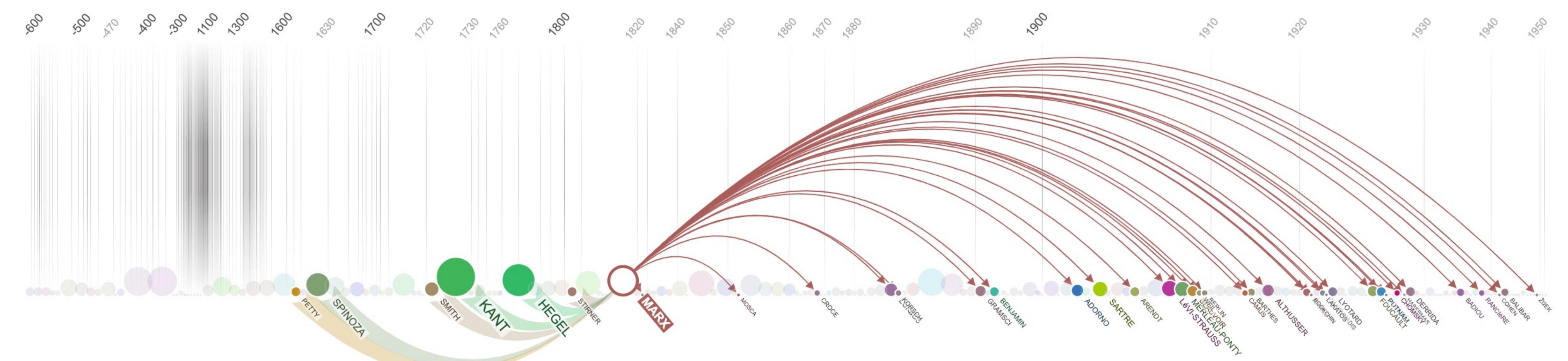
Recommended for networks where nodes can be separated into groups easily and where these groups are central to the analysis

Attribute-Driven Positioning

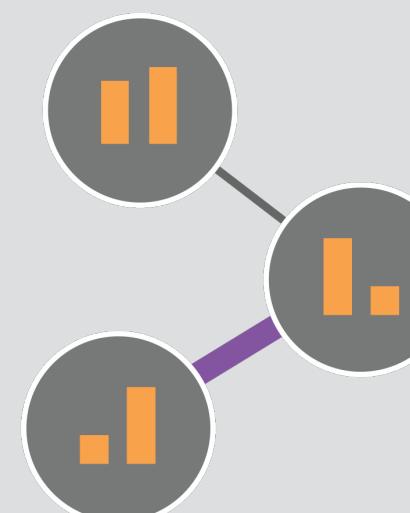




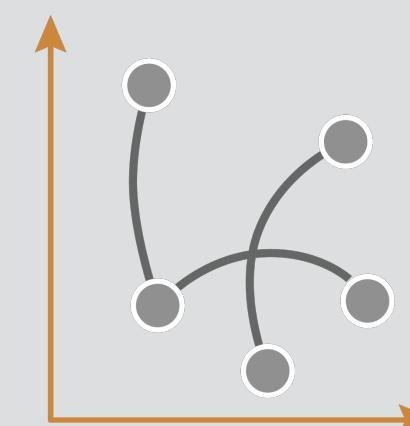
Edge Map Dork et al. 2011



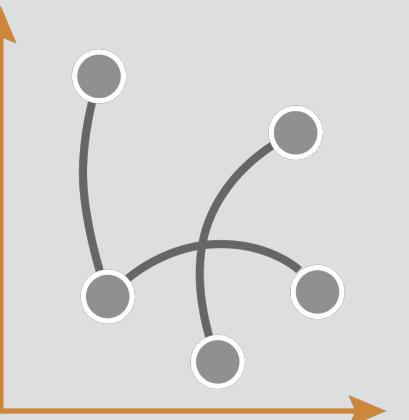
Querying and Filtering



On-Node / On-Edge
Encoding



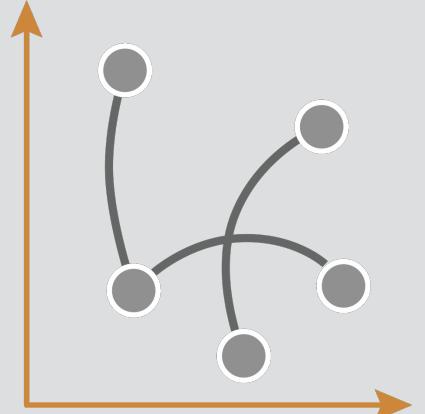
Attribute-Driven
Positioning



Attribute-Driven
Positioning



Well suited for quantitative attributes



Attribute-Driven
Positioning



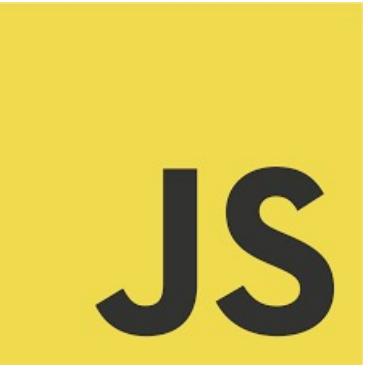
Does not lend itself well to visualizing
the topology of the network.

Recommended for smaller, sparse networks where relationships between node attributes are paramount to the analysis task, and topological features only provide context

Tools and Applications

For graphic designer and developers

developer



⚡ Observable Teams Demo ... Fork Sign in

Welcome. This is live code! Click the left margin to view or edit.

D3 · Nov 15, 2017
Bring your data to life.
By Mike Bostock

Listed in d3-drag, d3-force, and Visualization 178 forks

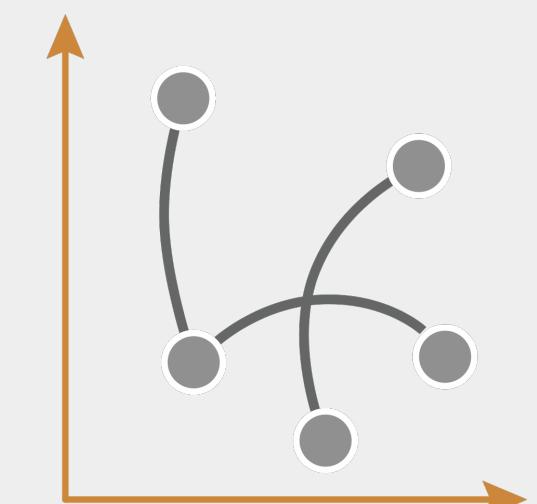
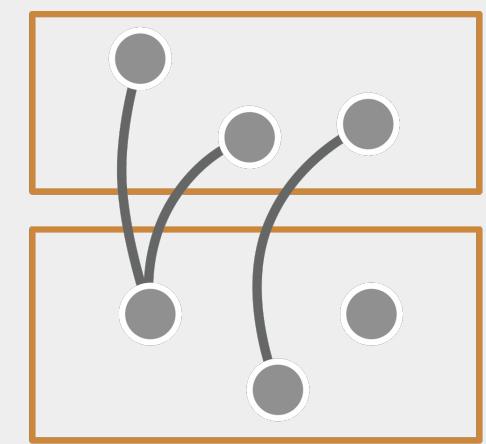
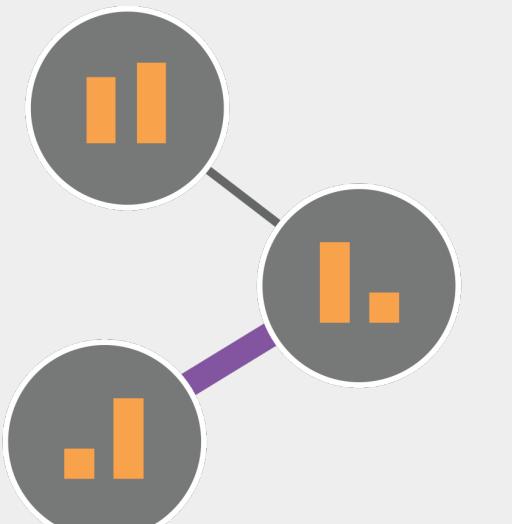
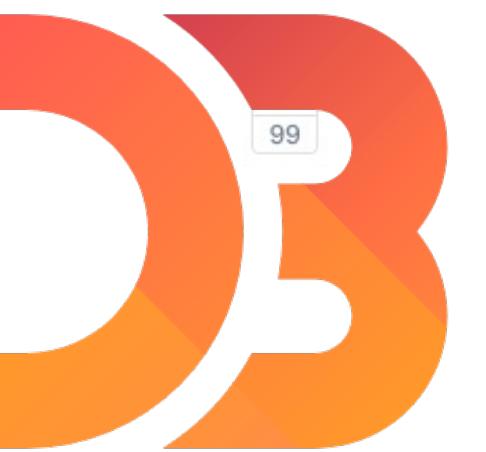
Force-Directed Graph

This network of character co-occurrence in *Les Misérables* is positioned by simulated forces using d3-force. See also a [disconnected graph](#), and compare to [WebCoLa](#).

```
chart = {
  const links = data.links.map(d => Object.create(d));
  const nodes = data.nodes.map(d => Object.create(d));

  const simulation = d3.forceSimulation(nodes)
    .force("link", d3.forceLink(links).id(d => d.id))
    .force("charge", d3.forceManyBody())
    .force("center", d3.forceCenter(width / 2, height / 2));

  const svg = d3.create("svg")
```



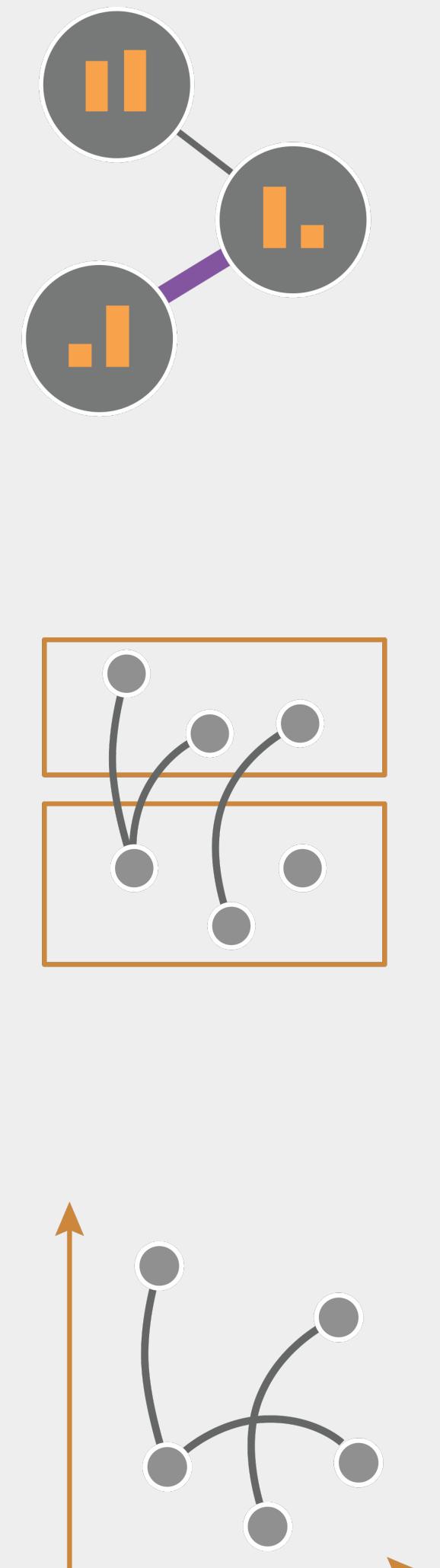
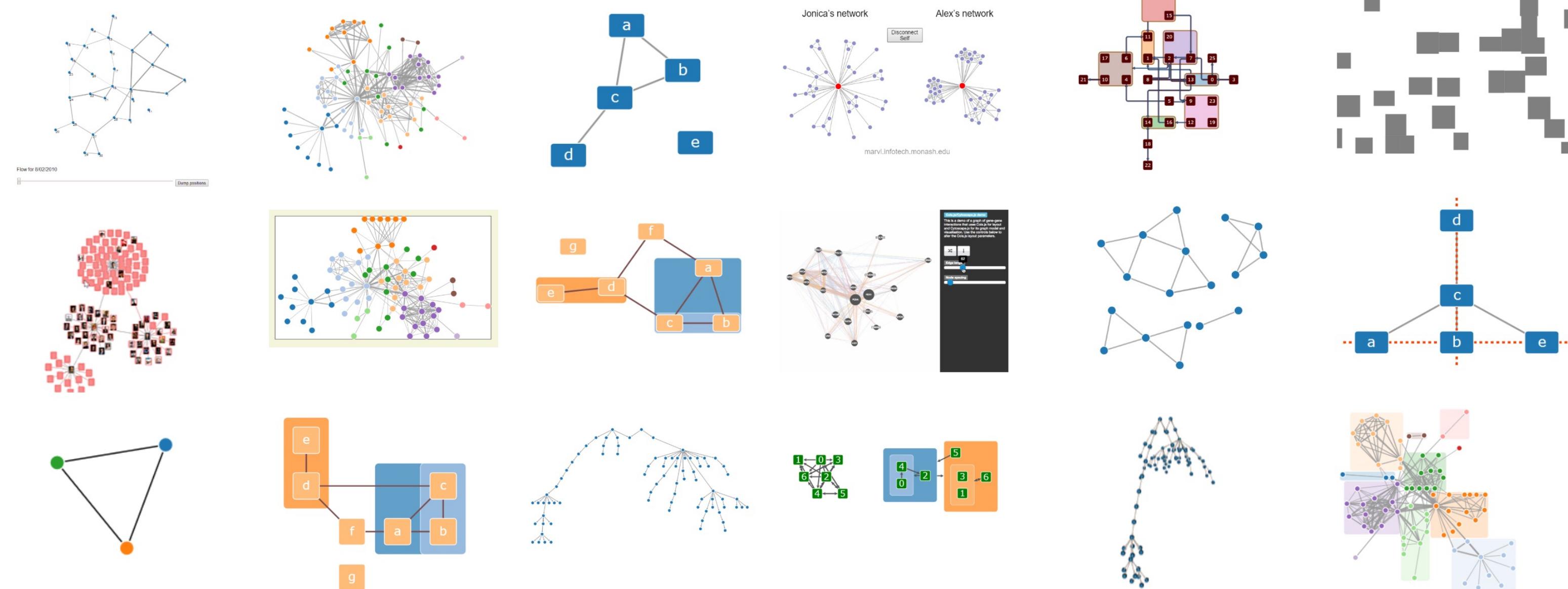
developer



[Overview](#) [Wiki](#) [API](#) [Source](#)

cola.js

Constraint-Based Layout in the Browser



Cola.js (A.K.A. "WebCoLa") is an open-source JavaScript library for arranging your HTML5 documents and diagrams using constraint-based optimization techniques.

developer



GGRAPH 1.0.2.9999 Reference Getting Started ▾ Articles ▾ News ▾

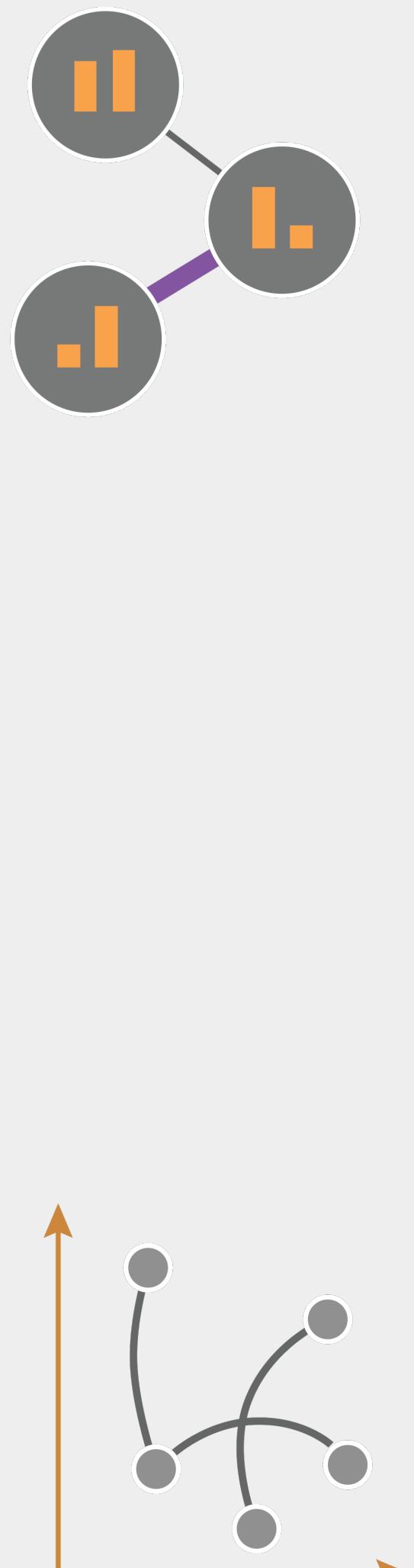
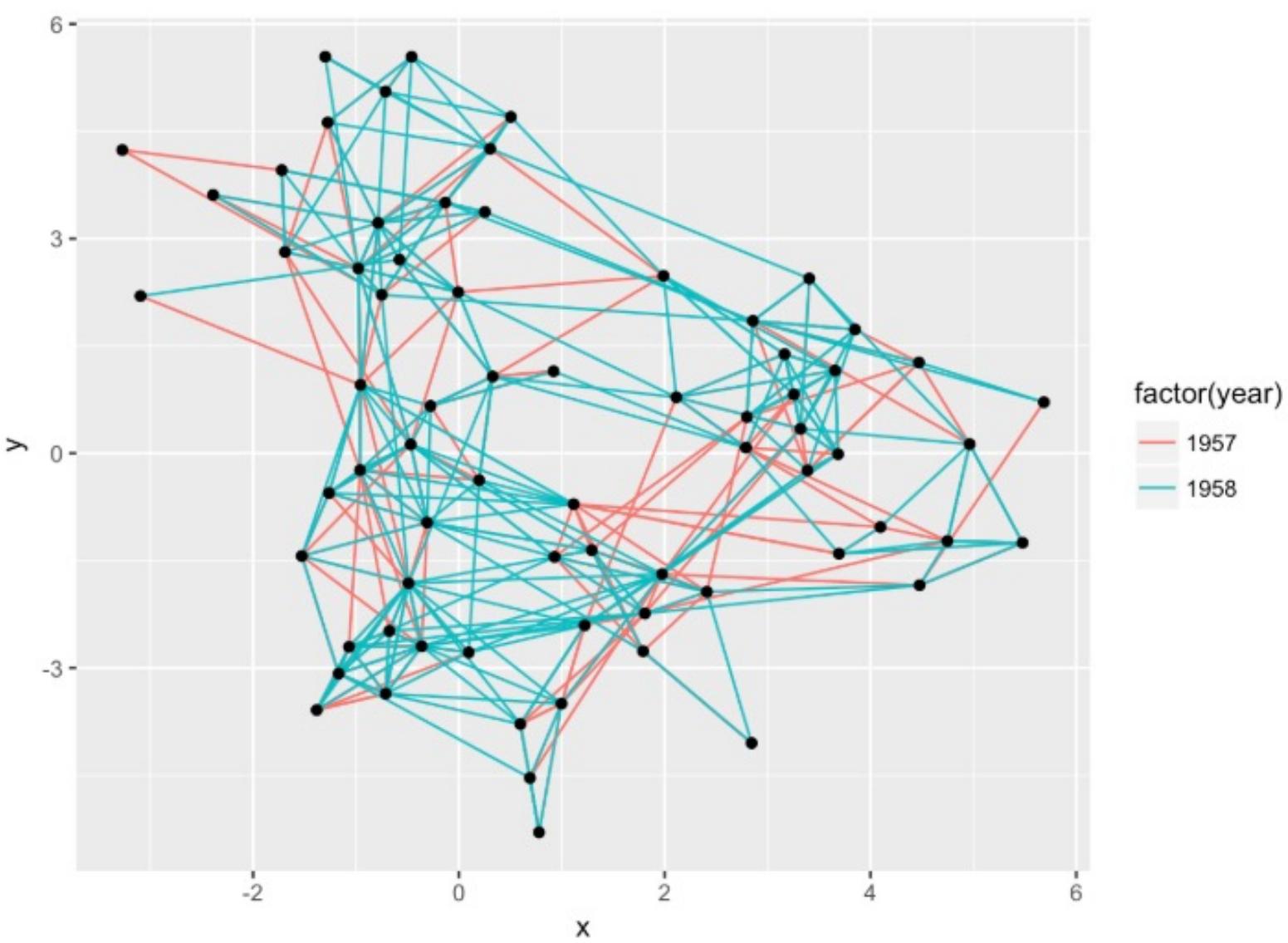
ggraph

/dʒi:dʒɪ'ra:f/ (or g-giraffe)



A grammar of graphics for relational data

ggraph is an extension of `ggplot2` aimed at supporting relational data structures such as networks, graphs, and trees. While it builds upon the foundation of `ggplot2` and its API it comes with its own self-contained set of geoms, facets, etc., as well as adding the concept of *layouts* to the grammar.



developer



plotly | Graphing Libraries

DEMO DASH

Help Open Source Graphing Libraries Python Scientific Network Graphs

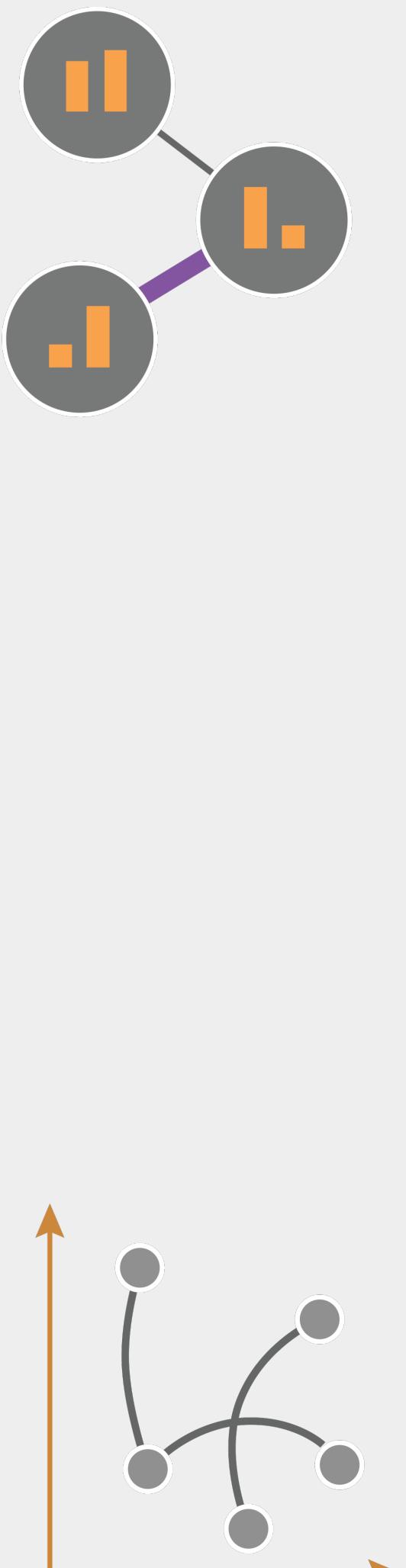
Edit this page on GitHub

Create Network Graph

```
fig = go.Figure(data=[edge_trace, node_trace],
                 layout=go.Layout(
                     title='<br>Network graph made with Python',
                     titlefont_size=16,
                     showlegend=False,
                     hovermode='closest',
                     margin=dict(b=20,l=5,r=5,t=40),
                     annotations=[ dict(
                         text="Python code: <a href='https://plot.ly/ipython-notebooks/network-graphs/'> https://plot.ly/ipython-notebooks/network-graphs/</a>",
                         showarrow=False,
                         xref="paper", yref="paper",
                         x=0.005, y=-0.002 ) ],
                     xaxis=dict(showgrid=False, zeroline=False, showticklabels=False),
                     yaxis=dict(showgrid=False, zeroline=False, showticklabels=False)
                 )
                 fig.show()
```

Network graph made with Python

Python code: <https://plot.ly/ipython-notebooks/network-graphs/>



developer



NetworkX

Stable (notes)

2.3 – April 2019

[download](#) | [doc](#) | [pdf](#)

Latest (notes)

2.4 development

[github](#) | [doc](#) | [pdf](#)

Archive

Contact

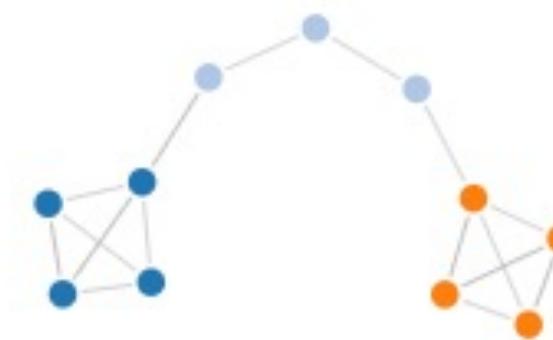
[Mailing list](#)

[Issue tracker](#)



Software for complex networks

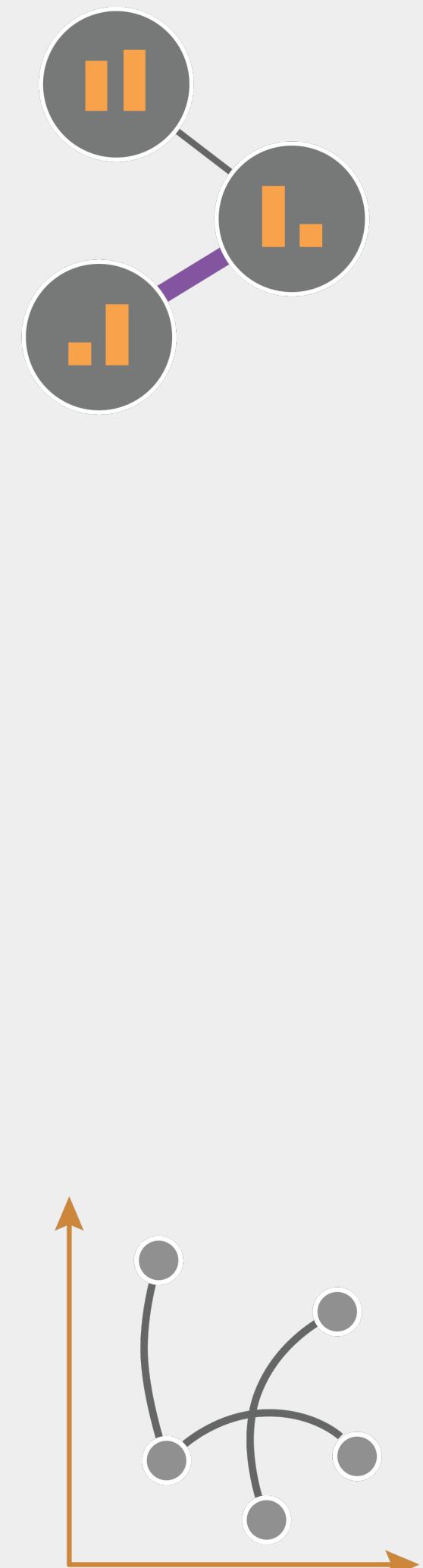
NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

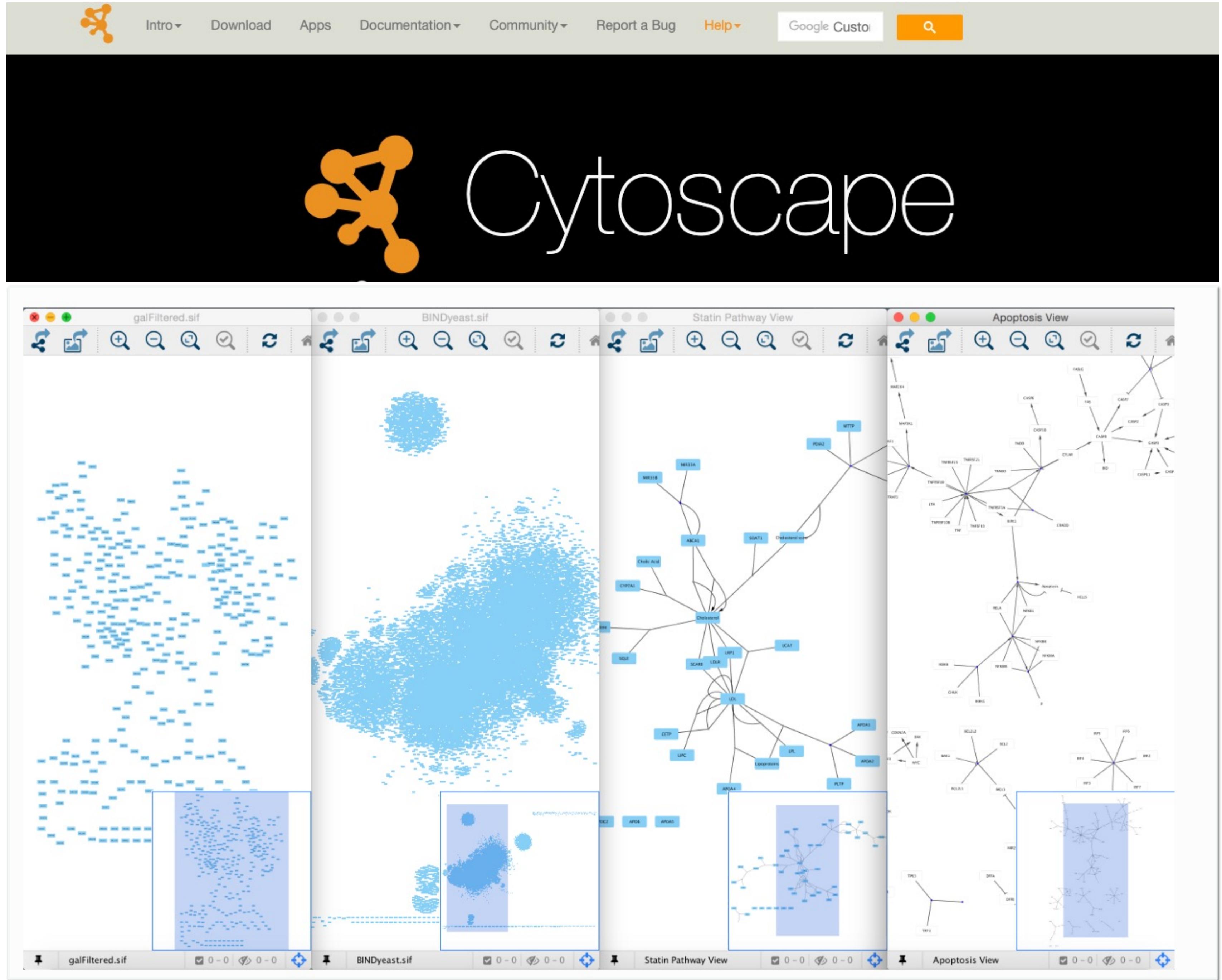


Features

- Data structures for graphs, digraphs, and multigraphs
- Many standard graph algorithms
- Network structure and analysis measures
- Generators for classic graphs, random graphs, and synthetic networks
- Nodes can be "anything" (e.g., text, images, XML records)
- Edges can hold arbitrary data (e.g., weights, time-series)
- Open source [3-clause BSD license](#)
- Well tested with over 90% code coverage
- Additional benefits from Python include fast prototyping, easy to teach, and multi-platform

©2014-2019, NetworkX developers. | Powered by [Sphinx 2.0.1](#) & [Alabaster 0.7.12](#)







[Download](#) [Blog](#) [Wiki](#) [Forum](#) [Support](#) [Bug tracker](#)

[Home](#) [Features](#) [Learn](#) [Develop](#) [Plugins](#) [Services](#) [Consortium](#)

graphic designer

The Open Graph Viz Platform

Gephi is the leading visualization and exploration software for all kinds of graphs and networks. Gephi is open-source and free.

Runs on Windows, Mac OS X and Linux.

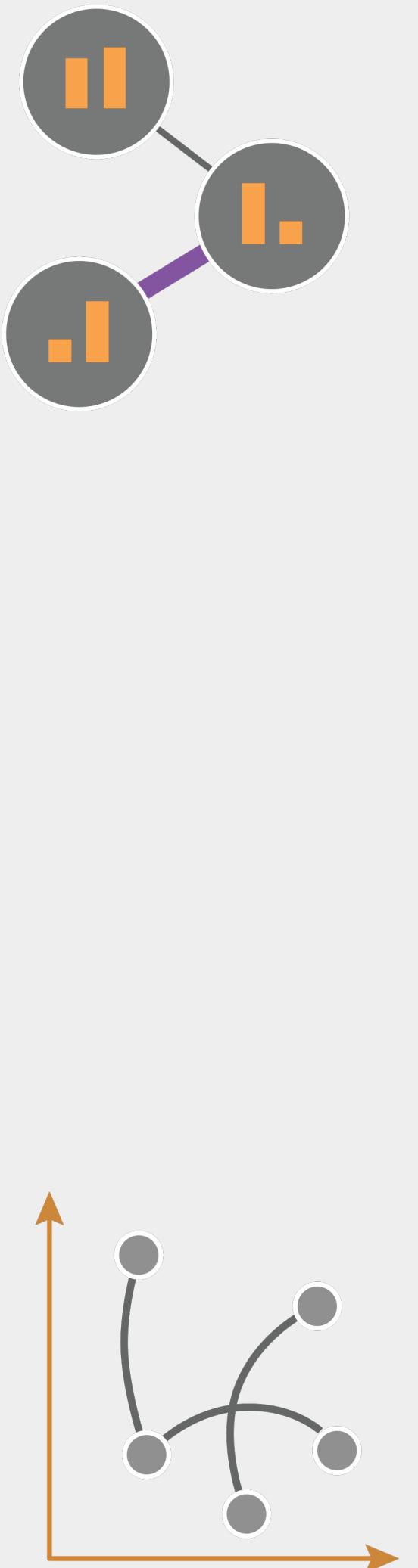
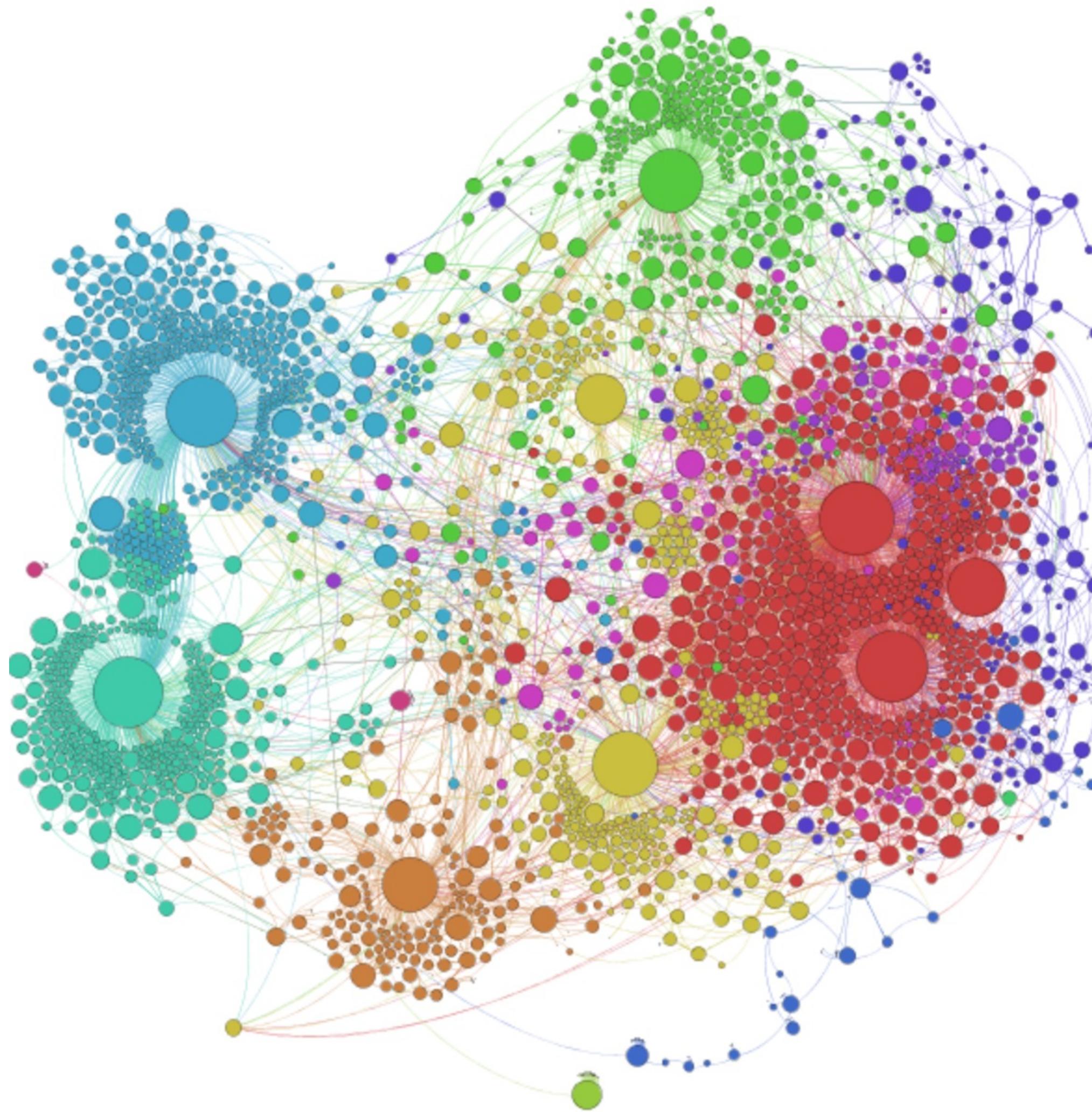
[Learn More on Gephi Platform »](#)

Download FREE
Gephi 0.9.2

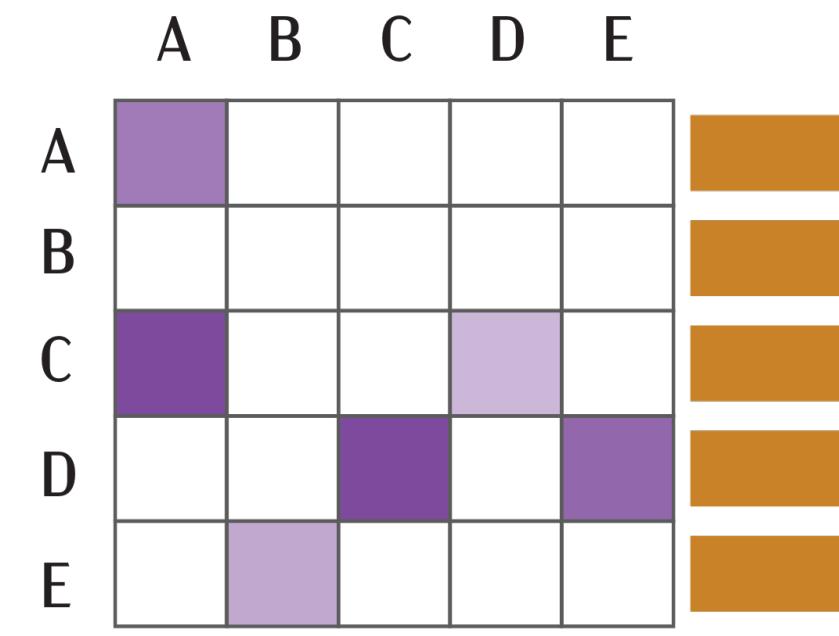
[Release Notes](#) | [System Requirements](#)

► [Features](#)
► [Quick start](#)

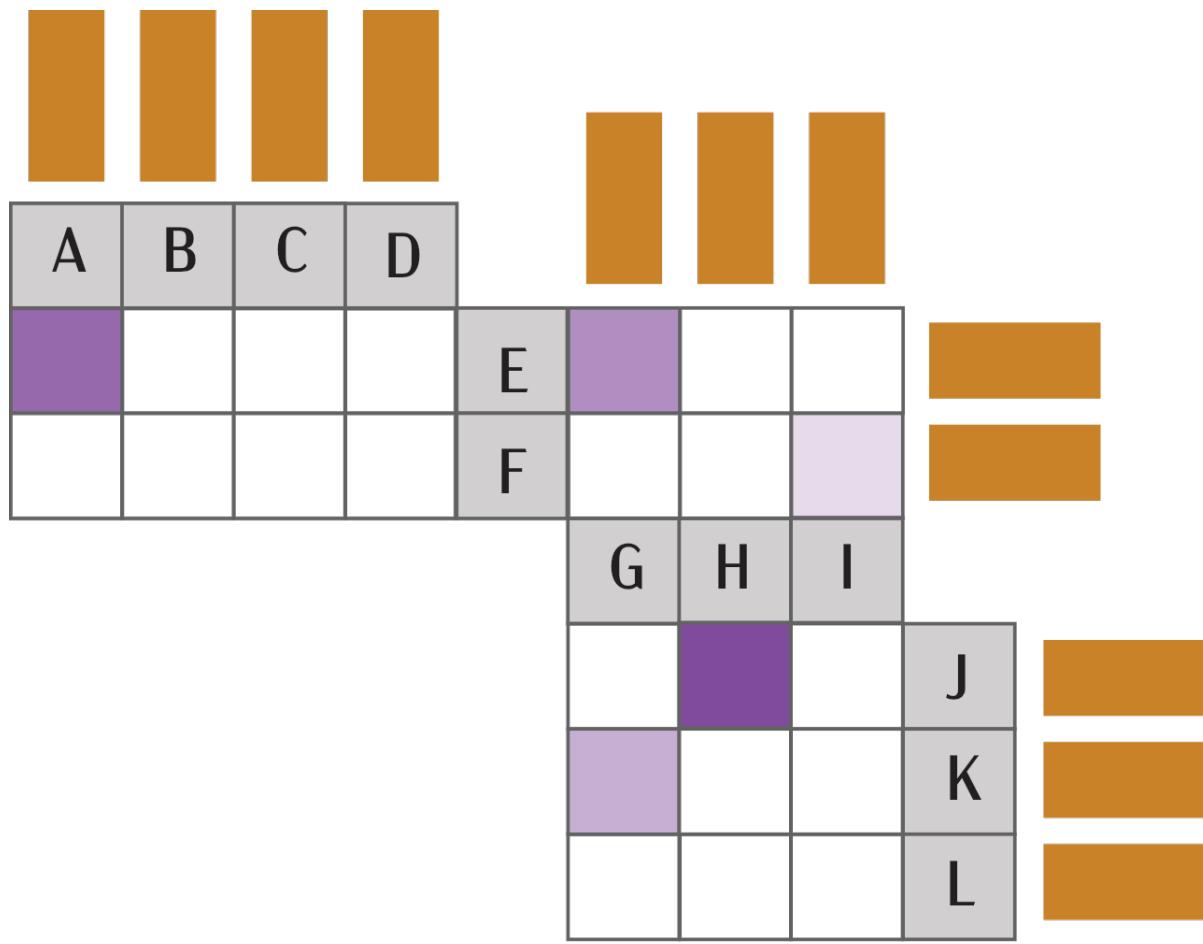
► [Screenshots](#)
► [Videos](#)



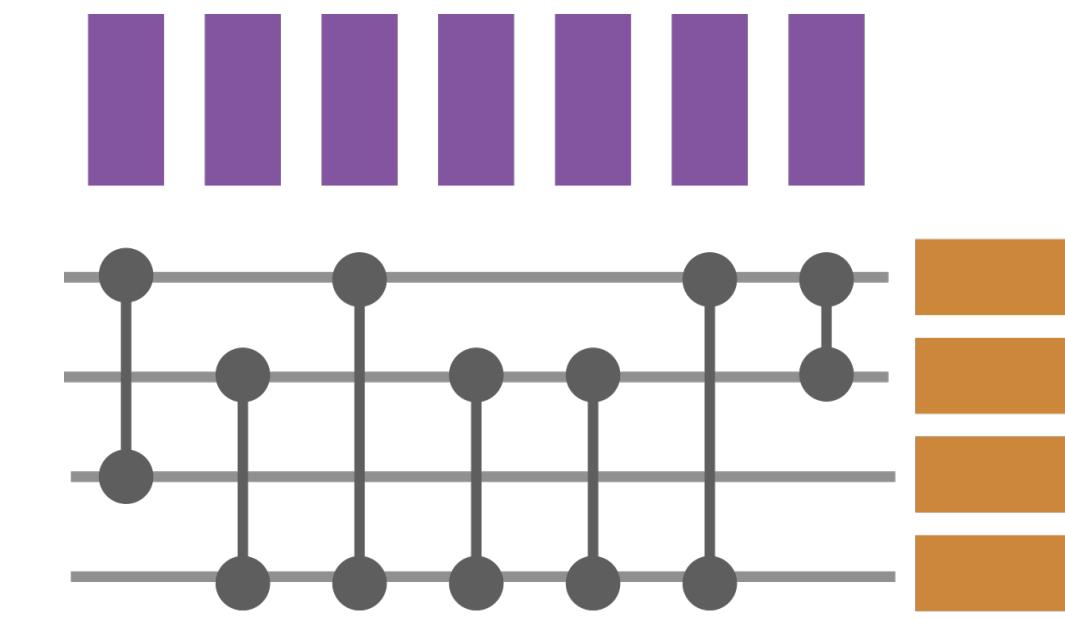
Tabular Layouts



Adjacency
Matrix



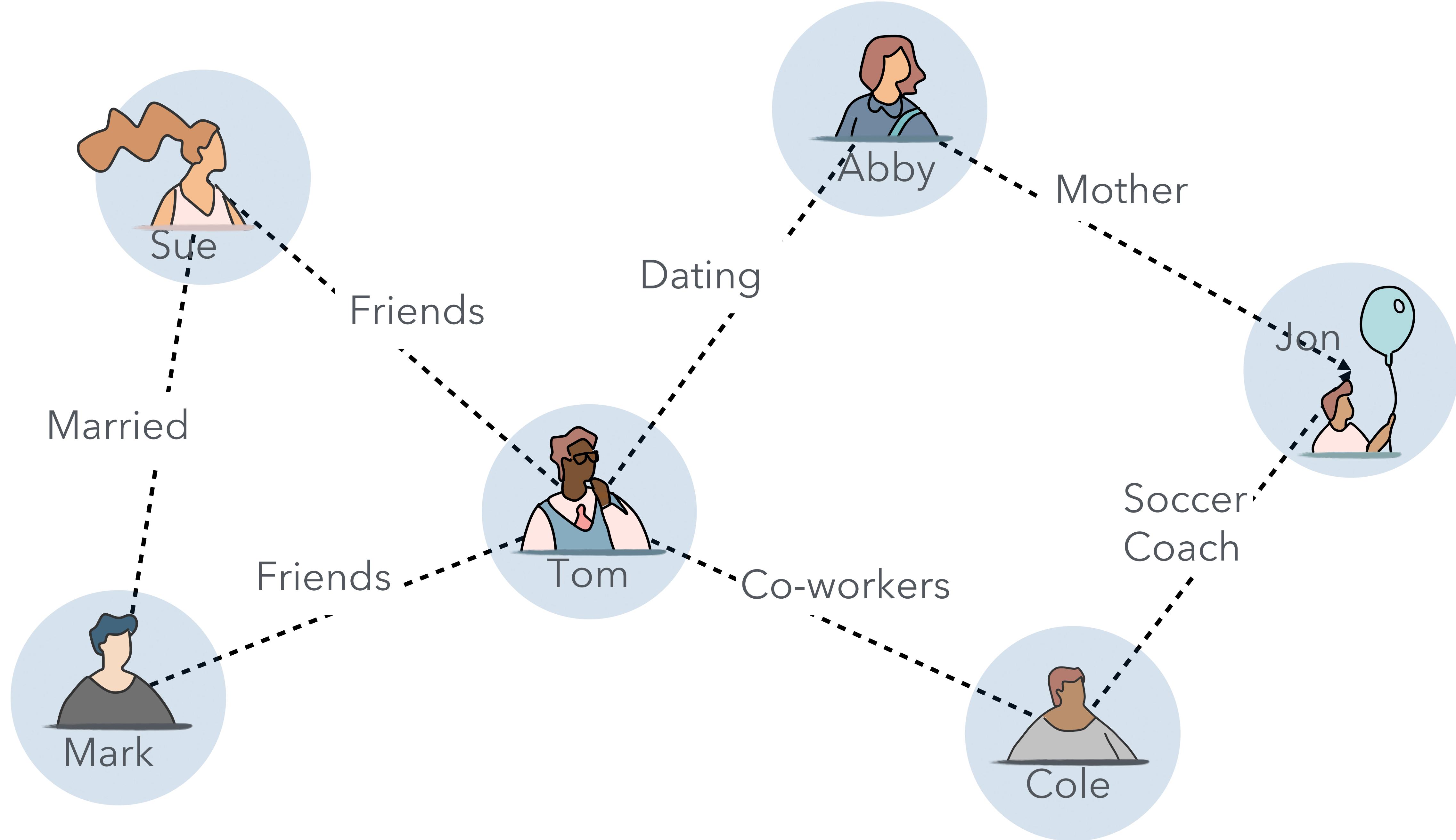
Quilts

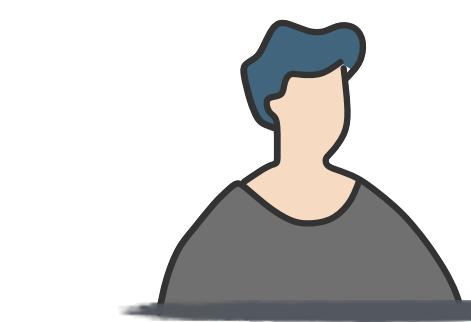
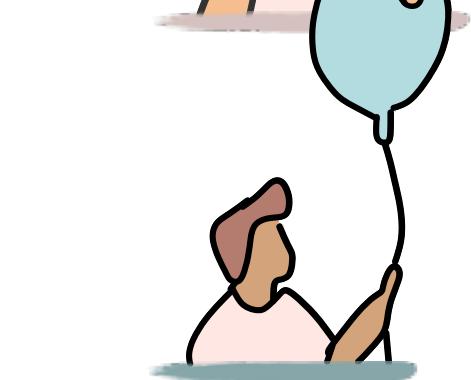


BioFabric

Adjacency Matrix

	A	B	C	D	E	
A	■					■
B						■
C	■			■		■
D			■		■	■
E		■				■



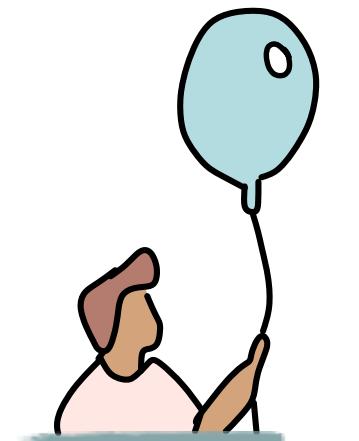
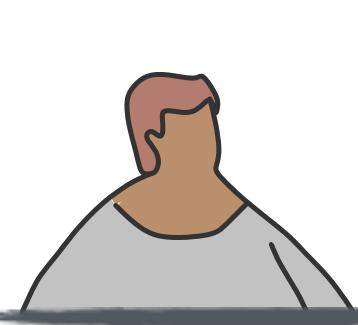
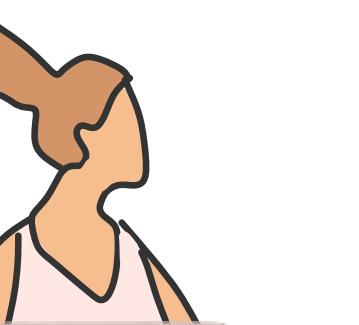


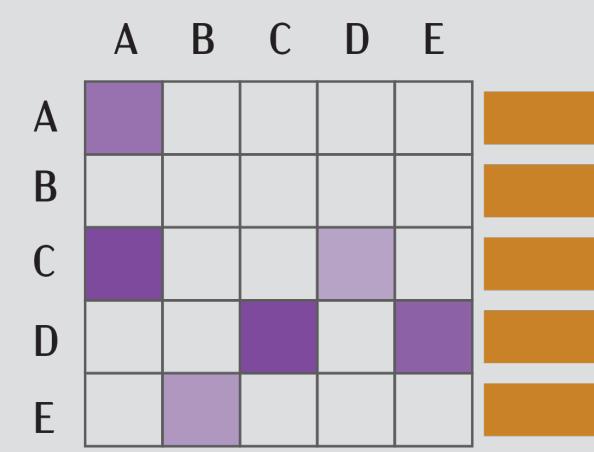
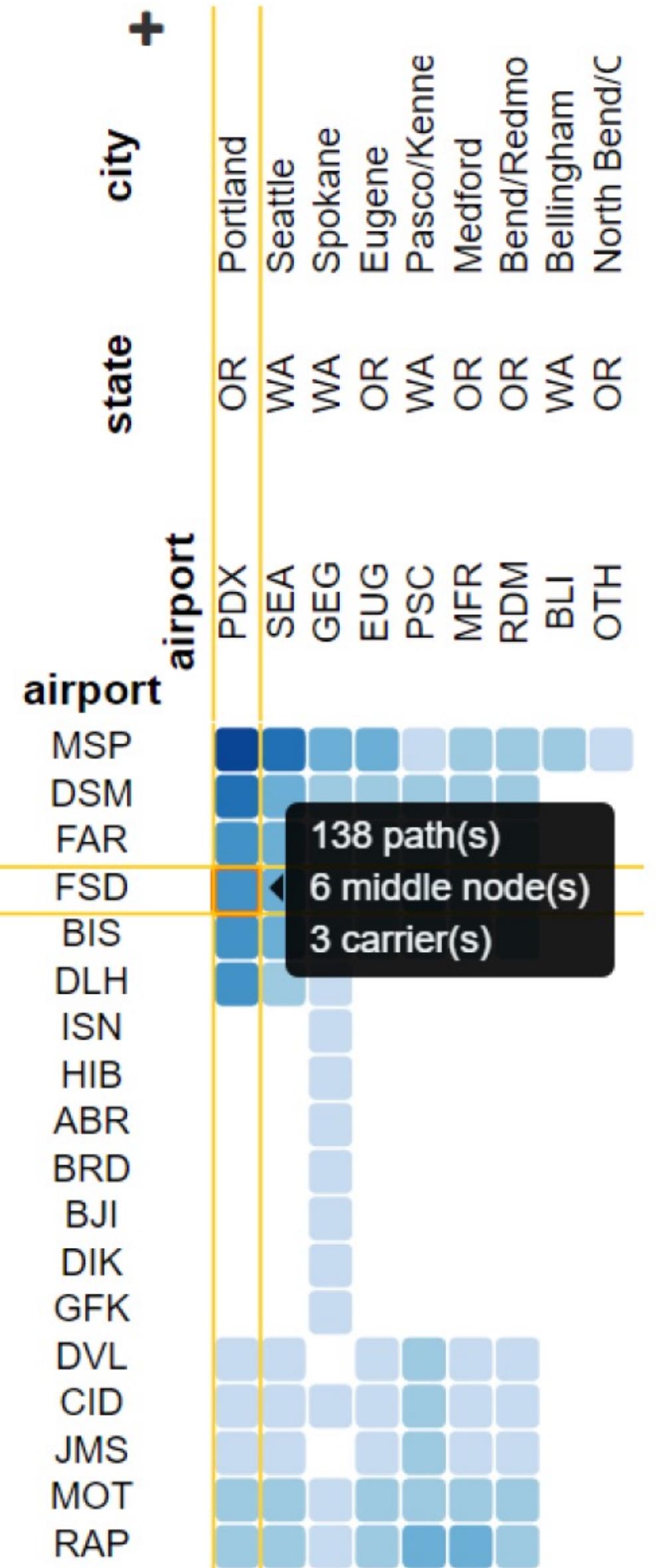
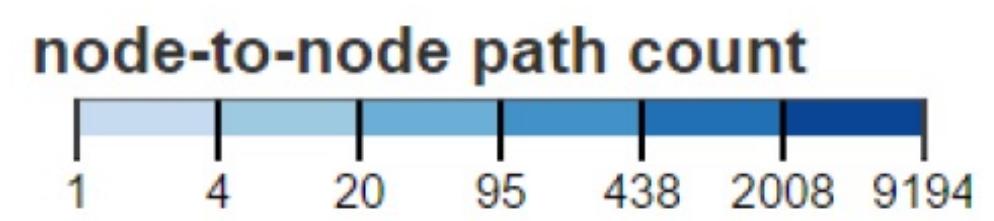
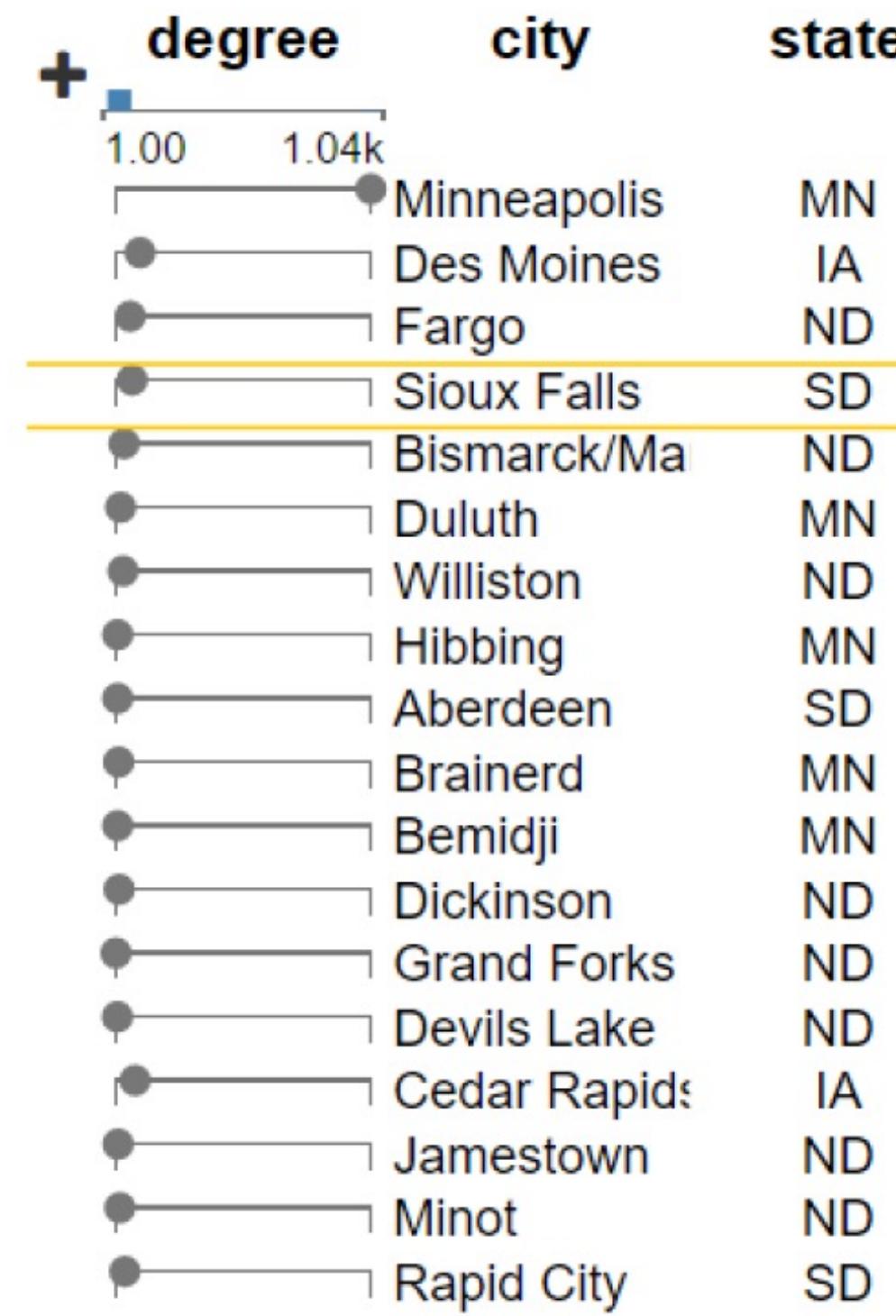
			Mother	Dating		
				Friends	Married	
	Dating	Friends			Friends	Co-Worker
		Married		Friends		
			Coach	Co-Worker		

			Mother	Dating		
				Friends	Married	
	Dating	Friends			Friends	Co-Worker
		Married		Friends		
			Coach	Co-Worker		

The diagram features a 5x7 grid of icons and labels. The icons are stylized human figures in various colors (blue, orange, pink, grey) with different hairstyles and accessories like a balloon. The labels represent social roles: 'Mother' (top row), 'Dating' (top row), 'Friends' (second row), 'Married' (second row), 'Co-Worker' (third row), 'Friends' (fourth row), 'Friends' (fourth row), 'Coach' (bottom row), and 'Co-Worker' (bottom row). The grid is bounded by dashed lines.

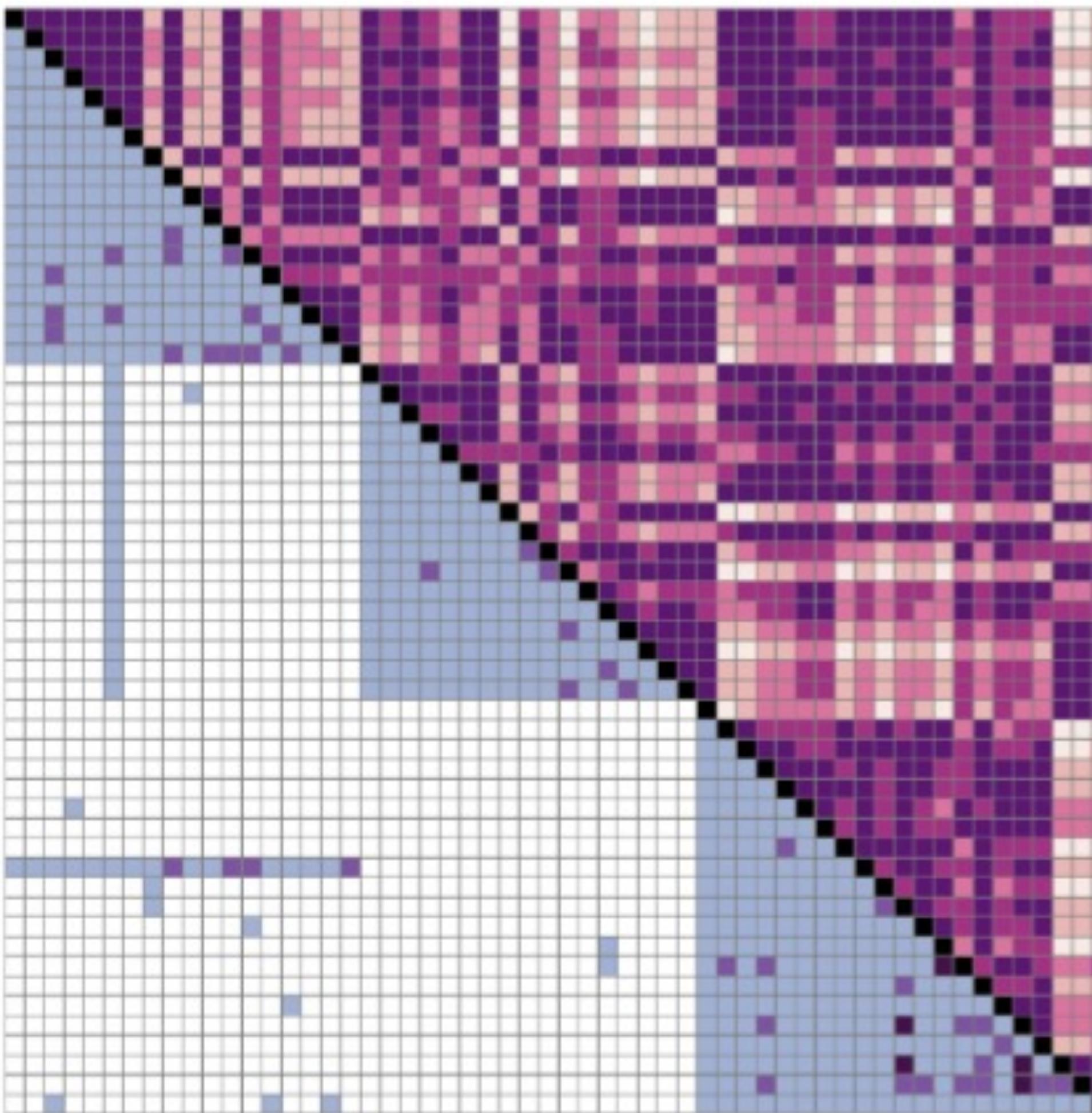
Name	Beverage	Day 1
Abby	Port	1
Sue	Coke	0
Jon	Coke	4
Tom	Beer	5
Mark	Beer	2
Cole	Port	3

						Name	Beverage	Day 1
		Co-Worker	Friends	Dating	Friends	Tom	Beer	5
						Jon	Coke	4
	Co-Worker	Coach				Cole	Port	3
	Friends				Married	Mark	Beer	2
	Dating	Mother				Abby	Port	1
	Friends		Married			Sue	Coke	0

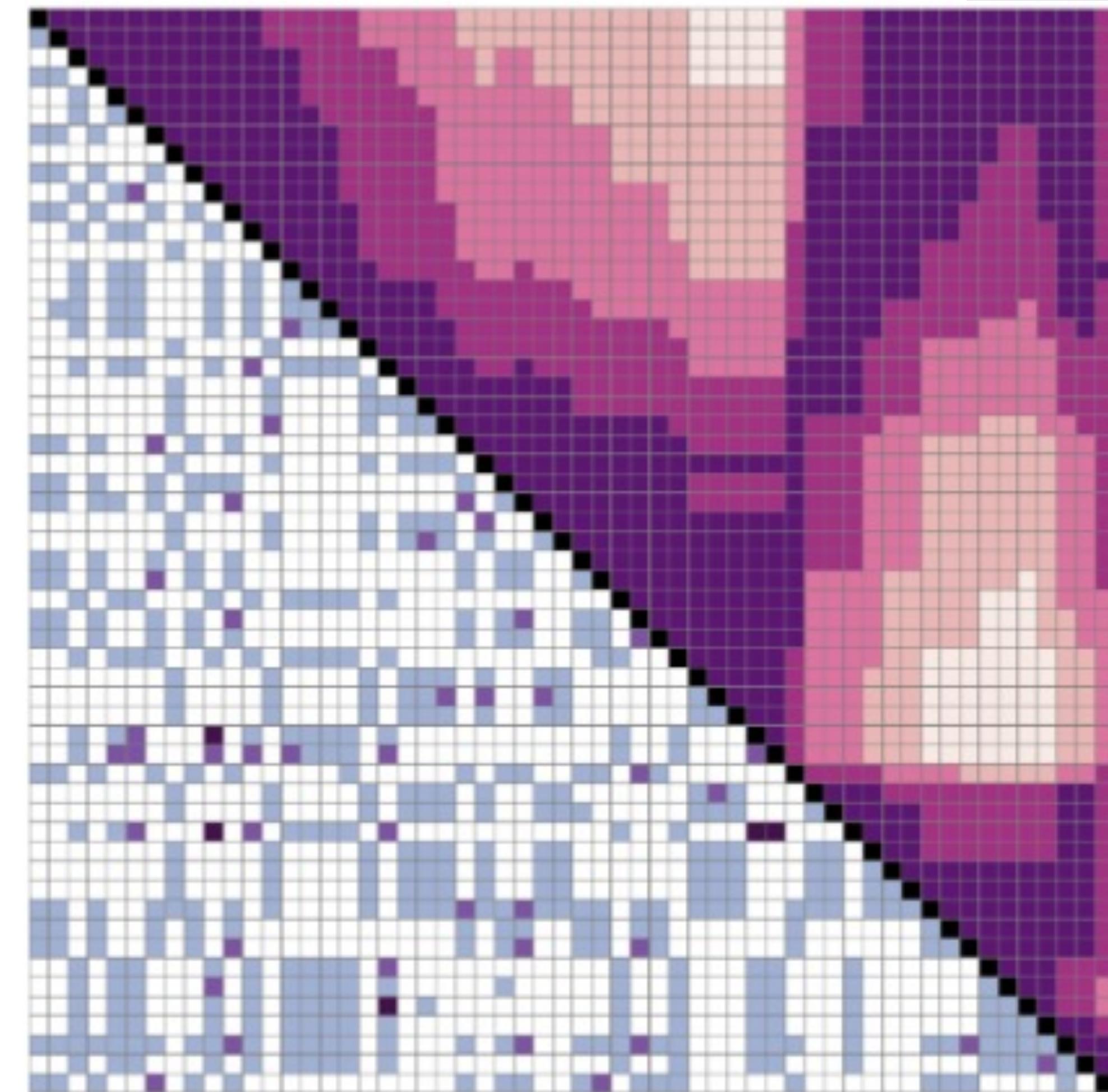


Adjacency
Matrix

Kerzner et al, 2017



(a) Sorted by structure.

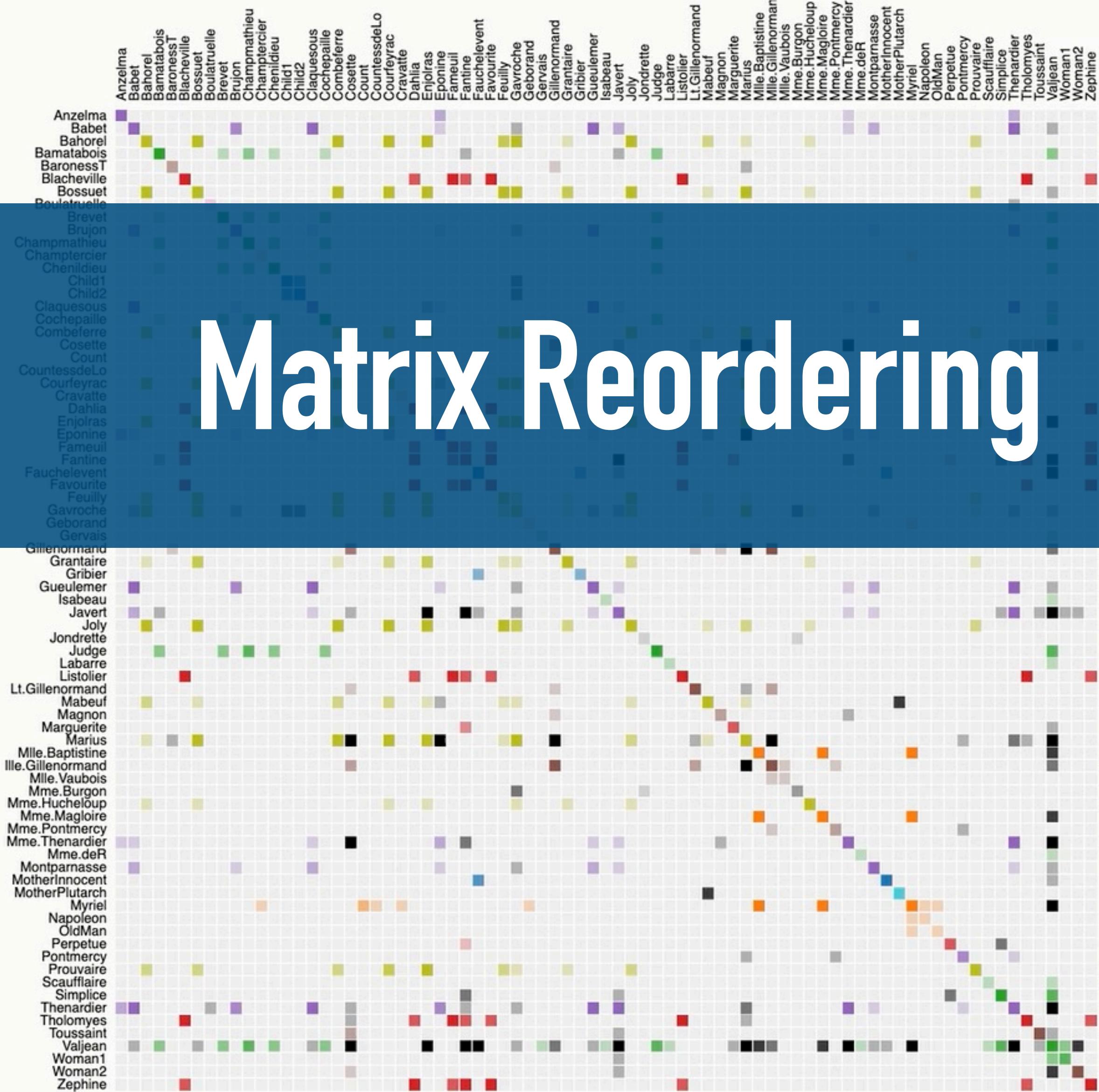


(b) Sorted by attribute similarity.

	A	B	C	D	E	
A	■					■
B		■				■
C	■		■			■
D				■	■	■
E		■				■

Adjacency
Matrix

Les Misérables Co-occurrence



This matrix diagram visualizes
character co-occurrences in
Victor Hugo's *Les Misérables*.

Each colored cell represents
two characters that appeared in
the same chapter; darker cells
indicate characters that co-
occurred more frequently.

Use the drop-down menu to
reorder the matrix and explore
the data.

Built with d3.js.

Matrix Reordering

Home

Jean-Daniel Fekete edited this page on Apr 23, 2015 · 2 revisions

Reorder.js is a library to reorder tables and graph/networks.

Resources

- [Introduction](#)
- [API Reference](#)

Browser / Platform Support

Reorder.js is mainly developed on Chrome and [Node.js](#). Use `npm install reorder.js` to install, and `require("reorder")` to load.

Installing

Download the latest version here:

- <https://github.com/jdfekete/reorder.js/releases>

Reorder.js

+ Add a custom footer

▼ Pages 12

Find a Page...

[Home](#)

[API Reference](#)

[Conversion](#)

[Core](#)

[Gallery](#)

[Graph](#)

[Introduction](#)

[LinearAlgebra](#)

[Matrix](#)

[Measure](#)

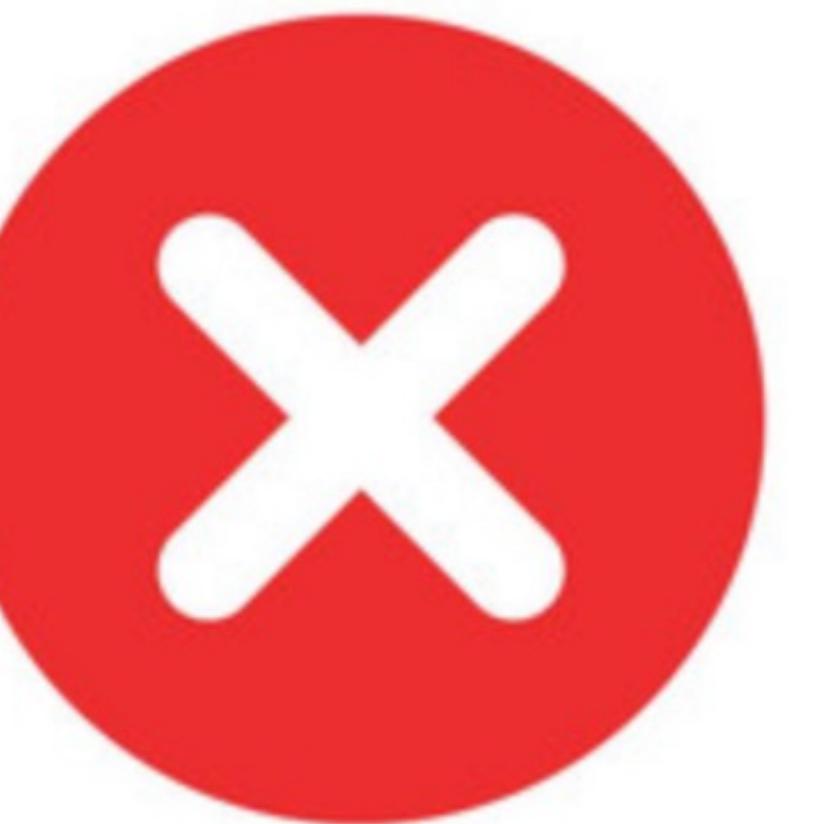
[Permutation](#)

[Reordering](#)

+ Add a custom sidebar

	A	B	C	D	E	
A	■					■
B						■
C	■			■	■	■
D			■	■	■	■
E		■				■

Adjacency
Matrix



	A	B	C	D	E
A	■				■
B					■
C	■			■	■
D			■	■	■
E		■			■

Adjacency
Matrix

Ideal for dense and completely connected networks

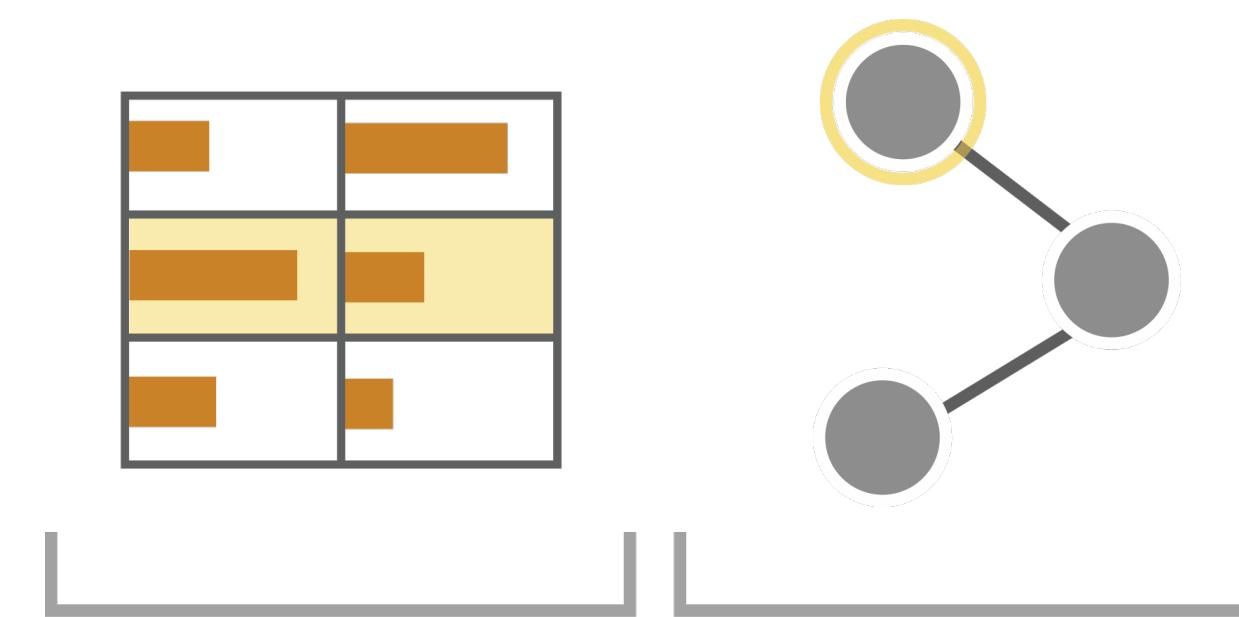


Requires quadratic space with respect to the number of nodes.

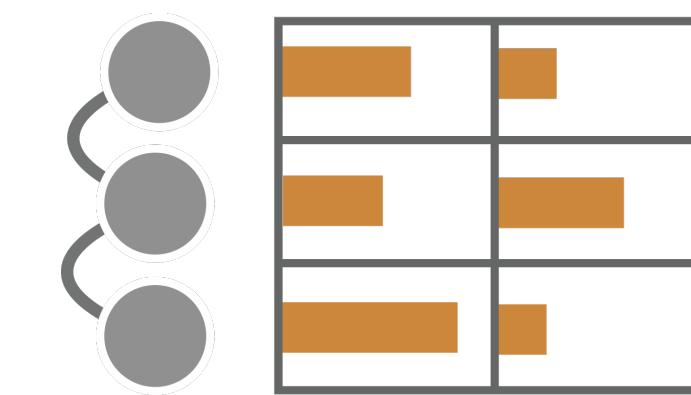
Complexity of choosing the right reordering algorithm

Recommended for smaller, complex and dense networks with rich node and/or edge attributes, for all tasks except for those involving paths

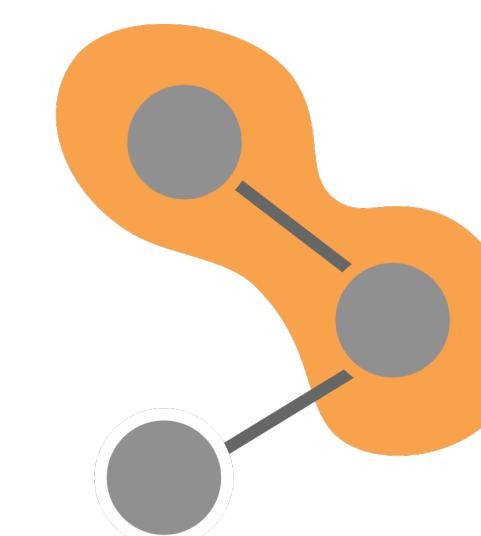
View Operations



Juxtaposed

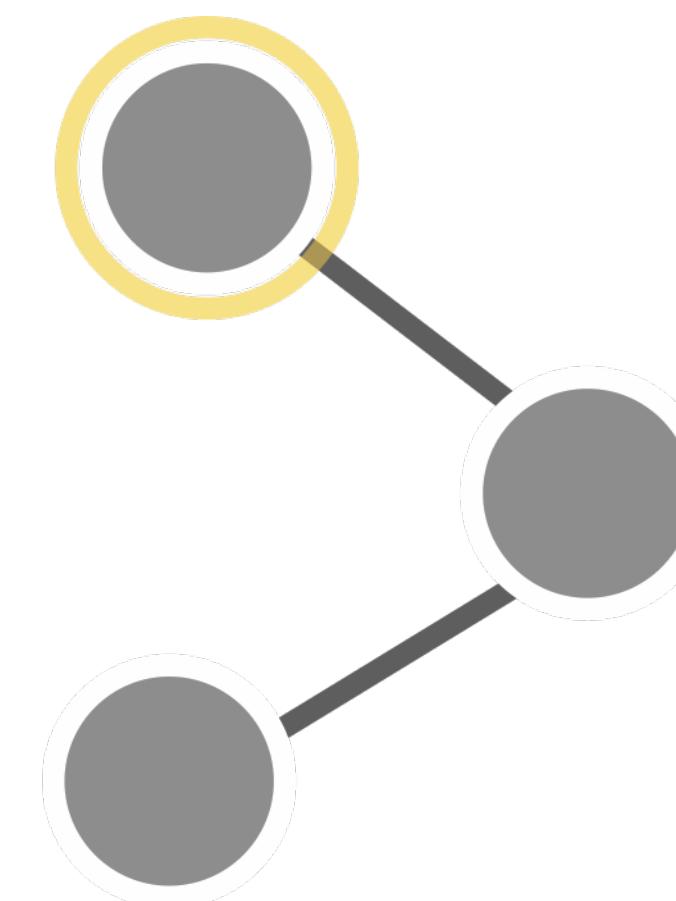
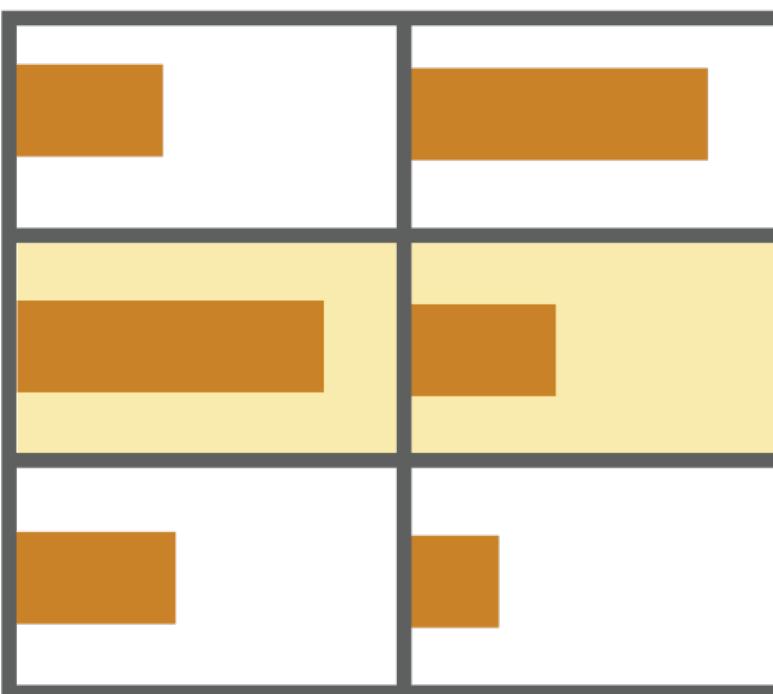


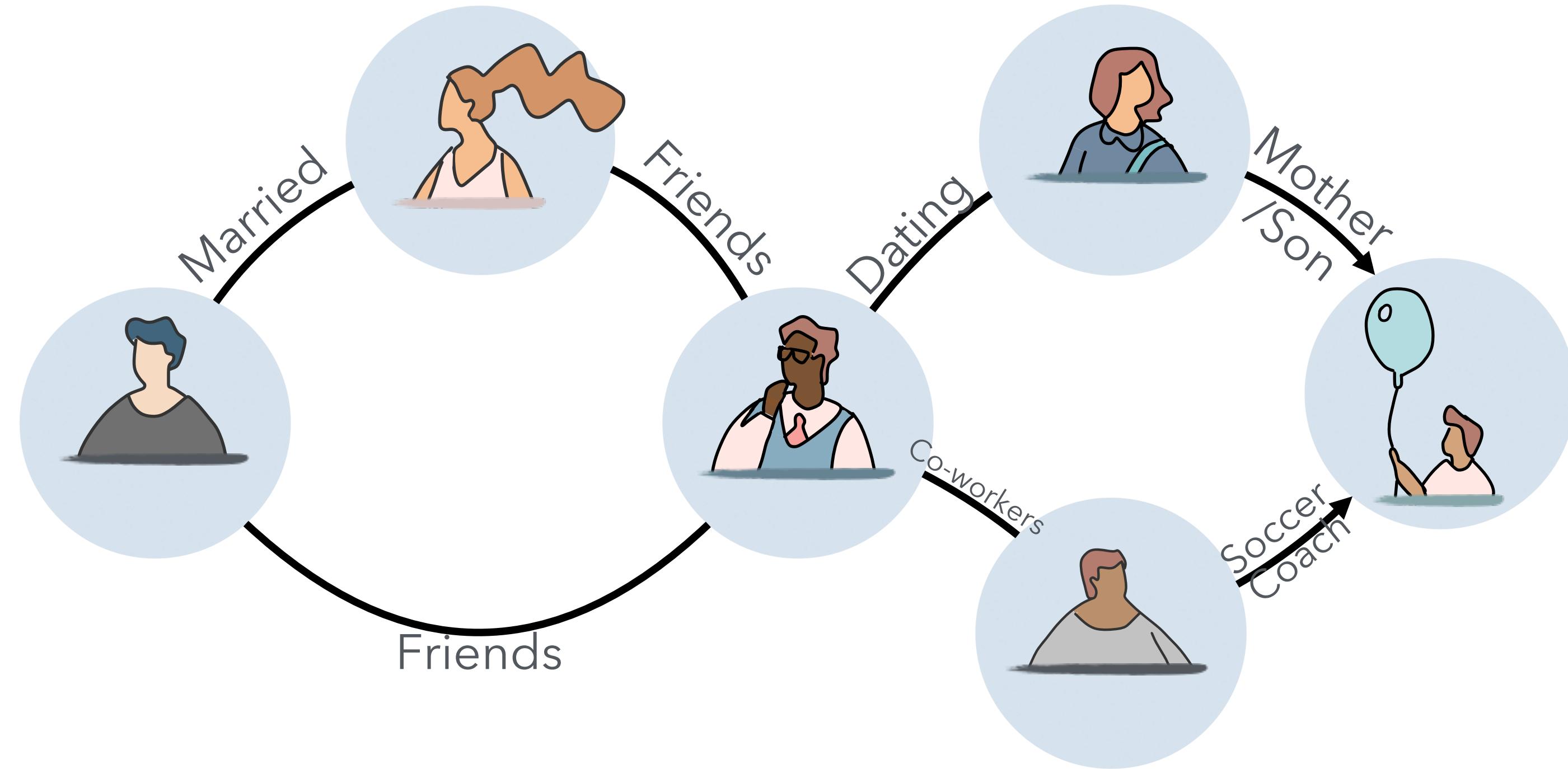
Integrated

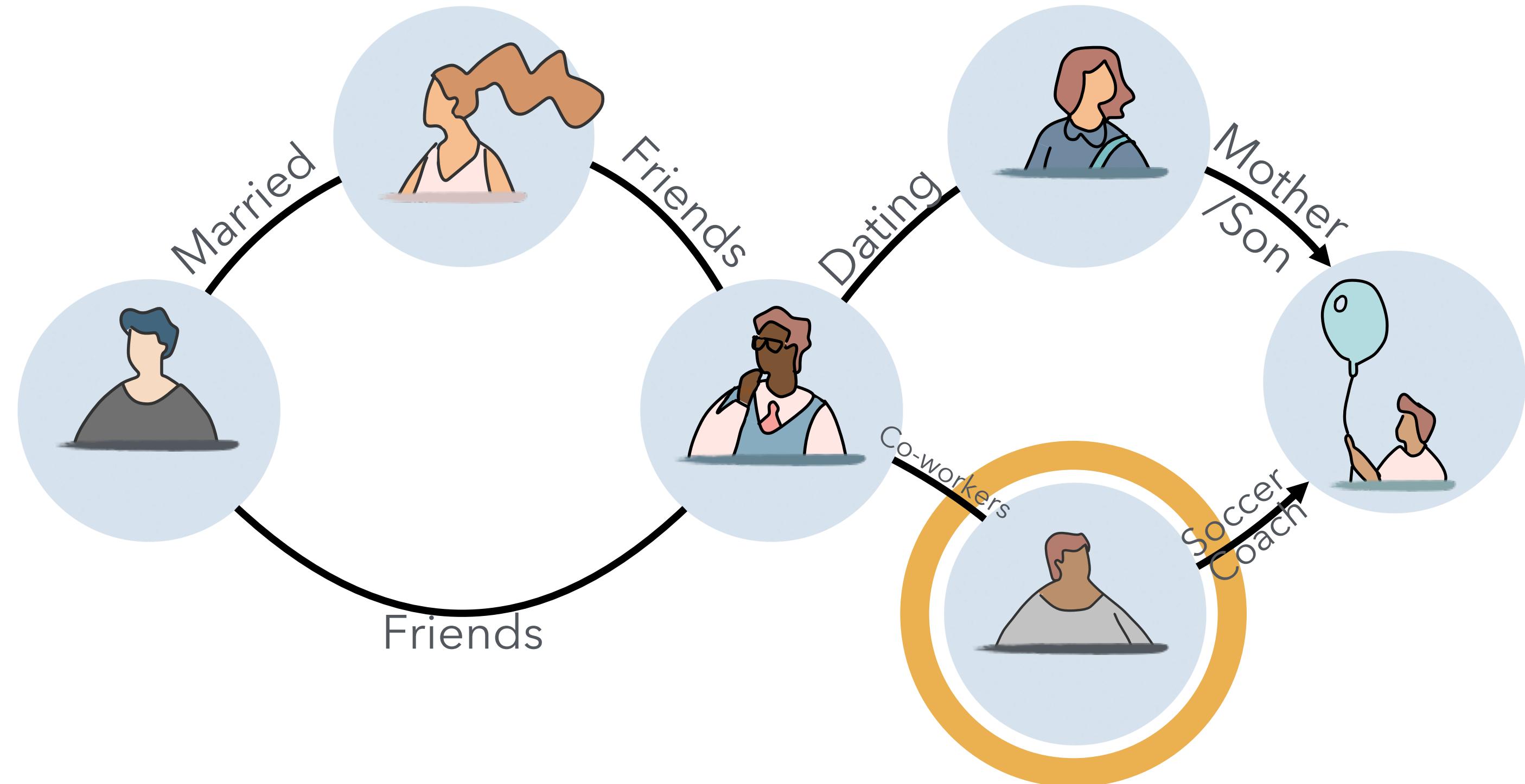


Overloaded

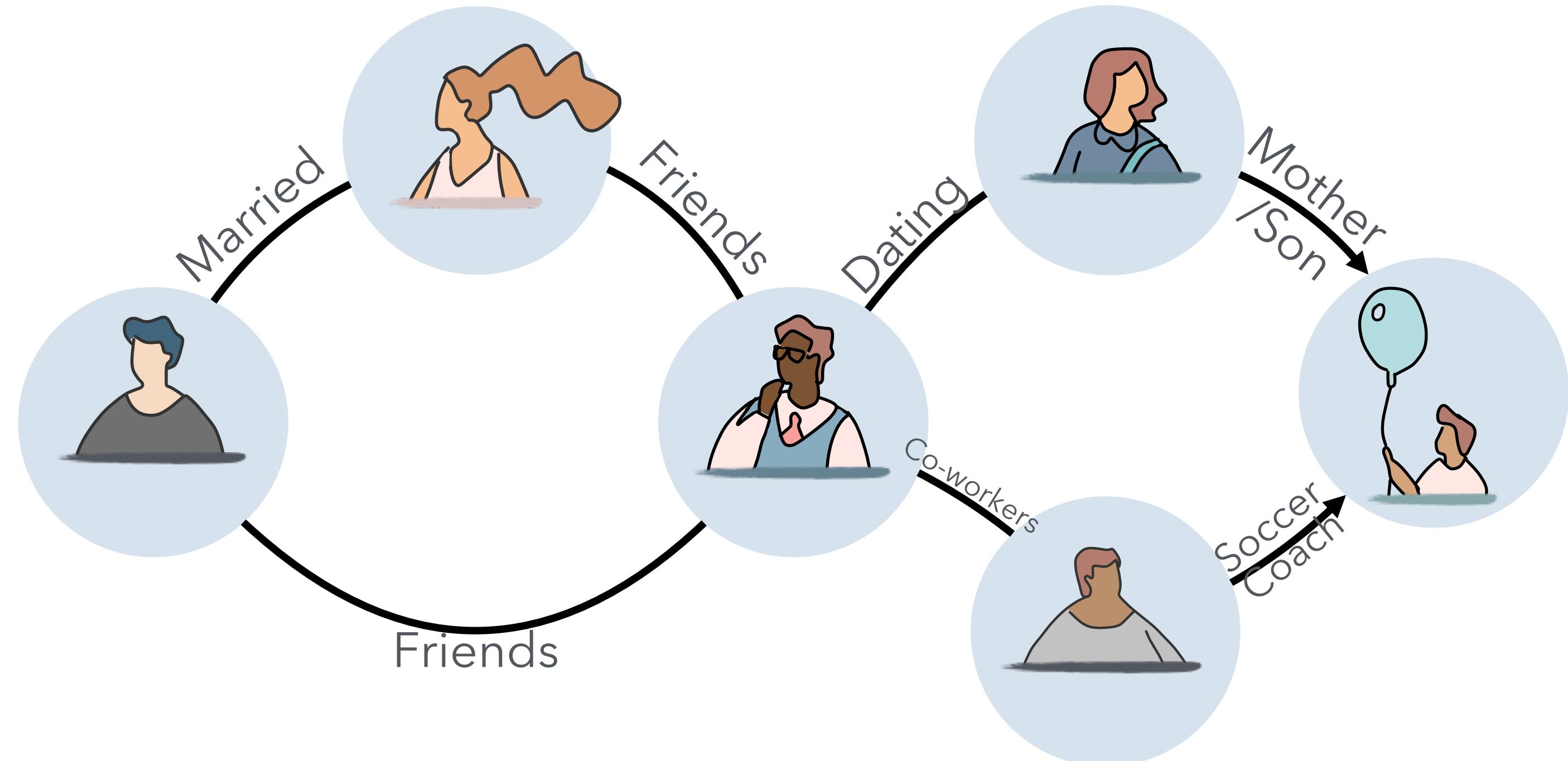
Juxtaposed





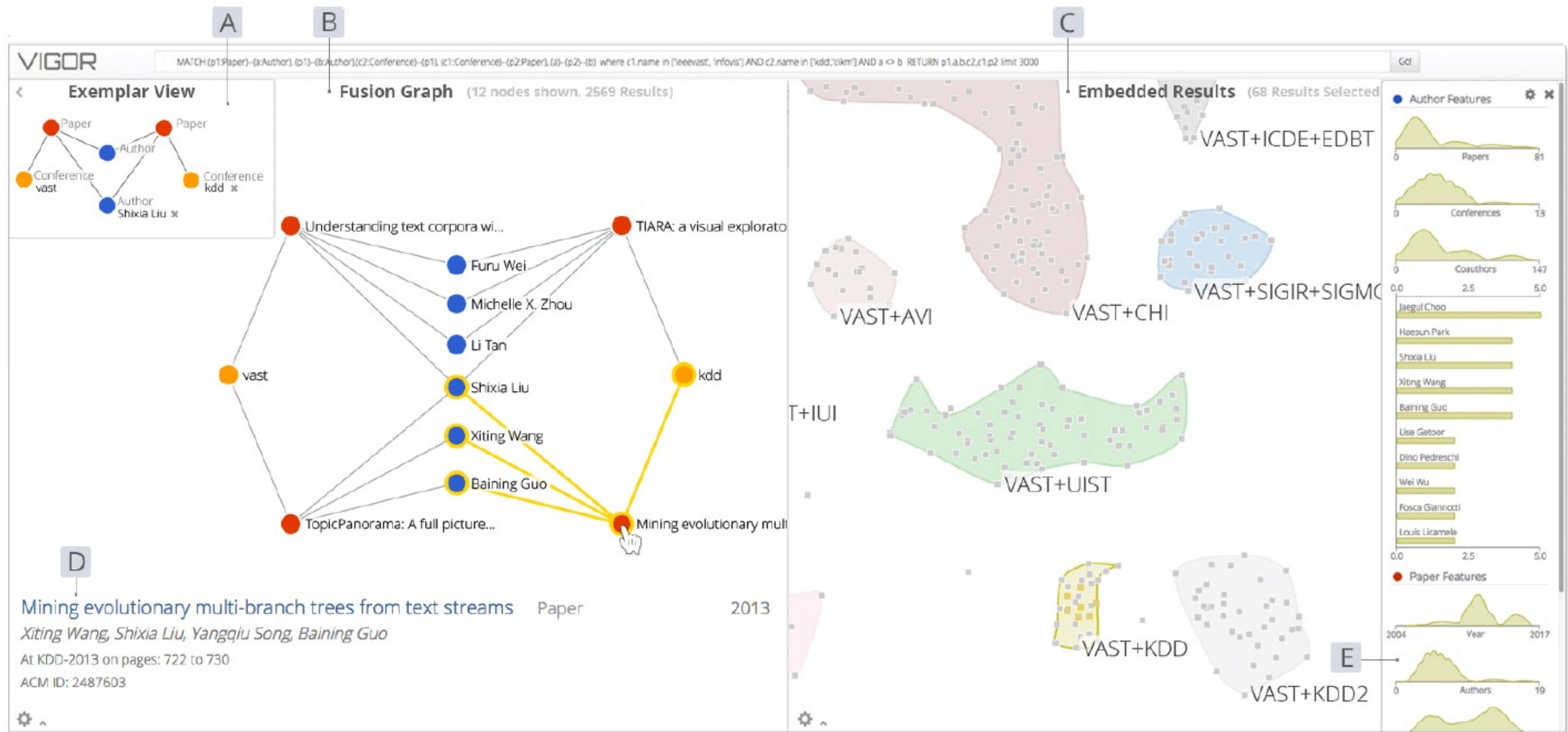


Name	Beverage	Day 1
Mark	Beer	1
Sue	Coke	0
Cole	Port	4
Jon	Coke	5
Tom	Beer	2
Abby	Port	3

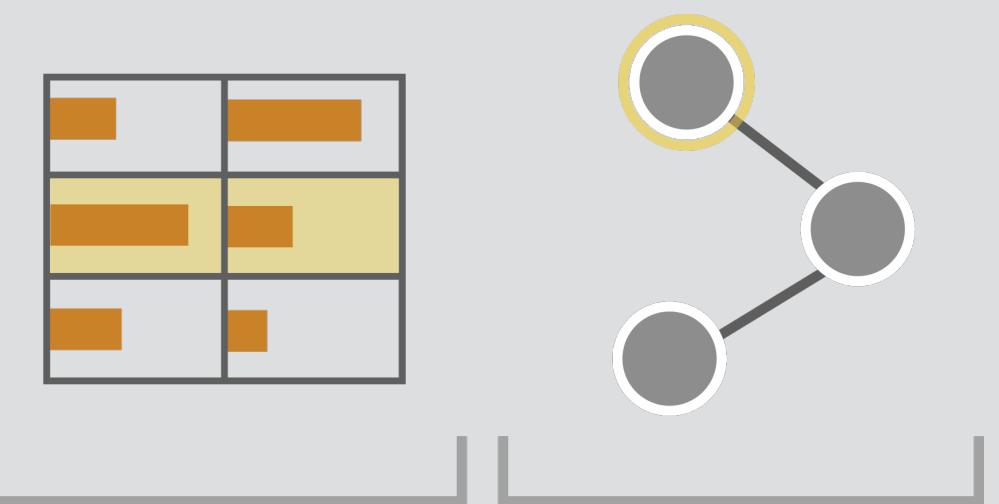


Name	Beverage	Day 1
Mark	Beer	1
Sue	Coke	0
Cole	Port	4
Jon	Coke	5
Tom	Beer	2
Abby	Port	3

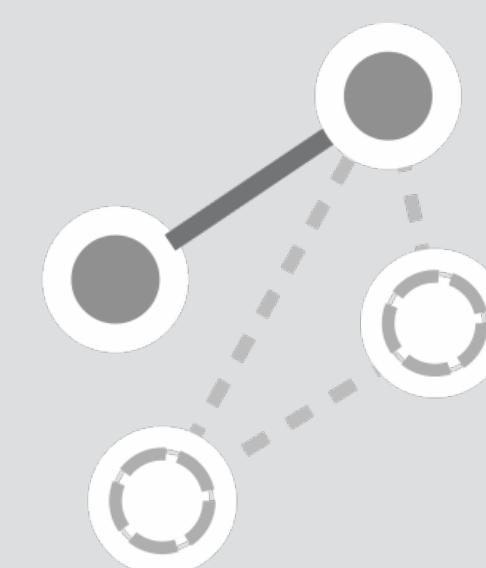
Relationship	Years
Dating	4
Mother / Son	12
Co-workers	3
Soccer Coach	2
Friends	8
Friends	3
Married	4



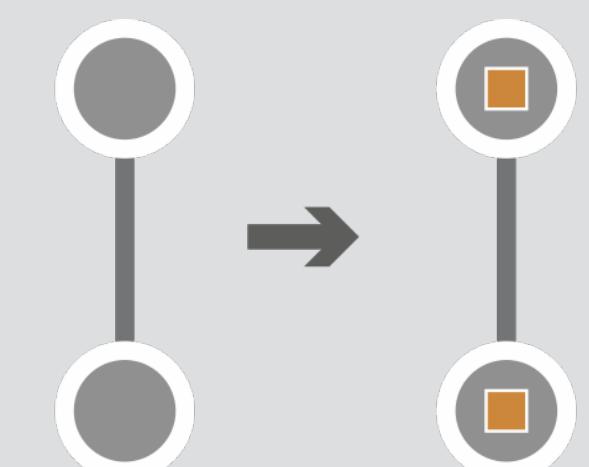
VIGOR Pienta et al. 2018



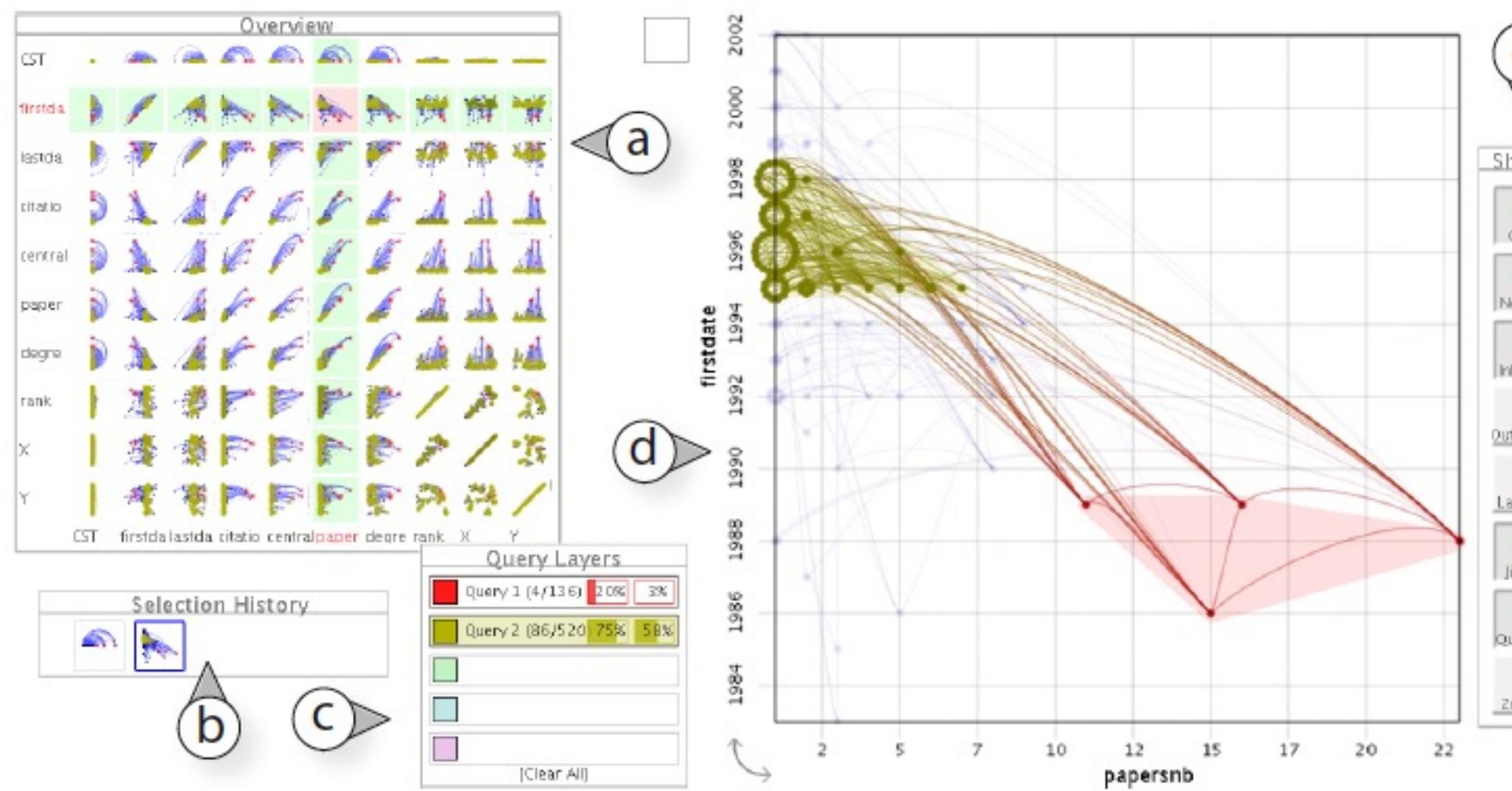
Juxtaposed



Querying and Filtering



Deriving New Attributes



e

Show

- Grid
- Nodes
- InLinks
- OutLinks
- Labels
- Jitter
- Queries
- Zoom

Details

id	label	first...	papersub	rank								
104	ACMID	status	centrality	clustrank	degree	firstd	fullname	id	label	first...	papersub	rank
105	P169127	0	4	8	1992	Elud Rivlin	n1965	Rivlin	1992	1	79	
106	P75893	0	5	4	1992	Daniel C. Robbins	n1870	Robbins	1998	1	92	
107	P95916	P95917	15811...	180	32	1989	George C. Robertson	n2012	Robertson	1999	11	117
108	P75487	P73472	0	4	2	1997	Edward L. Robertson	n1961	Robertson	1997	1	31
109	P73472	0	2	2	1996	E. L. Robertson	n1954	Robertson	1996	1	32	
110	PL19895	0	7	8	1996	Anne Rose	n1234	Rose	1996	1	70	
111	P270271	P270271	759.5	33	18	1990	Steven F. Roth	n1423	Roth	1999	8	25
112	P573425	P270271	1058.5	17	22	1995	S. F. Roth	n1844	Roth	1997	4	24
113	P299898	P573522	0	1	6	1995	William Ruh	n1499	Ruh	1995	1	62
114	P59113	P573031	0	5	6	1993	Daniel M. Russell	n1871	Russell	1993	1	111
115	P50762	0	0	4	2002	Varan Saini	n1726	Saini	2002	1	50	
116	P220113	0	2	6	1996	Patricia Schank	n1292	Schank	1996	1	110	
117	P571188	P573188	0	0	4	1999	Jeffrey Senn	n1814	Senn	1999	1	1
118	P541243	P573188	0	7	14	1996	J. A. Senn	n1575	Senn	1996	1	10
119	P28882	P28399	3391	178	48	1988	Ben Shneiderman	n1471	Shneiderman	2002	25	115
120	P76836	0	5	10	1995	Elizabeth Shopp	n1970	Shopp	1996	2	105	
121	P201702	0	2	14	1978	Myron Spaulding	n1250	Spaulding	1998	1	137	
122	PL49483	0	1	2	1992	Joseph L. Steffen	n1067	Steffen	1992	1	57	
123	PL91151	0	5	6	1995	Mark J. Steffin	n1997	Stefin	1995	1	112	
124	PL35514	0	2	8	1995	Mark J. Steffin	n1997	Stefin	1995	1	112	

f

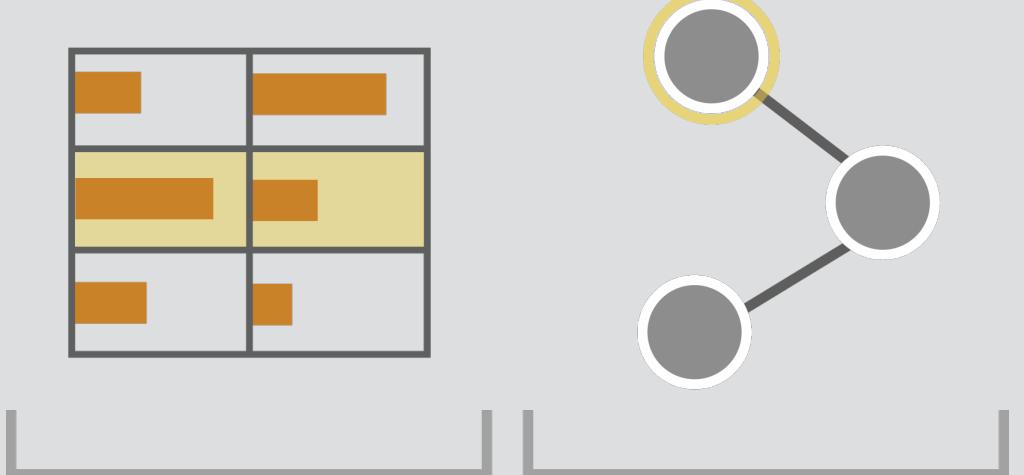
Edge Details

id	sourcevertices	targetvertices	text
694	Maximer	Robertson	1 som205326
695	Robertson	Maximer	1 som205326
696	Maximer	Card	1 som205326
697	Card	Maximer	1 som205326
698	Card	Mackinlay	1 som205326
699	Mackinlay	Maximer	1 som205326
700	Hearst	Haworsen	1 som205326
701	Haworsen	Hearst	1 som205326
702	Hearst	Rao	1 som205326
703	Rao	Hearst	1 som205326
704	Hearst	Roberson	1 som205326
705	Roberson	Hearst	1 som205326
706	Hearst	Card	1 som205326
707	Card	Hearst	1 som205326
708	Hearst	Mackinlay	1 som205326
709	Mackinlay	Hearst	1 som205326
710	Haworsen	Rao	1 som205326

g

Filter Text:

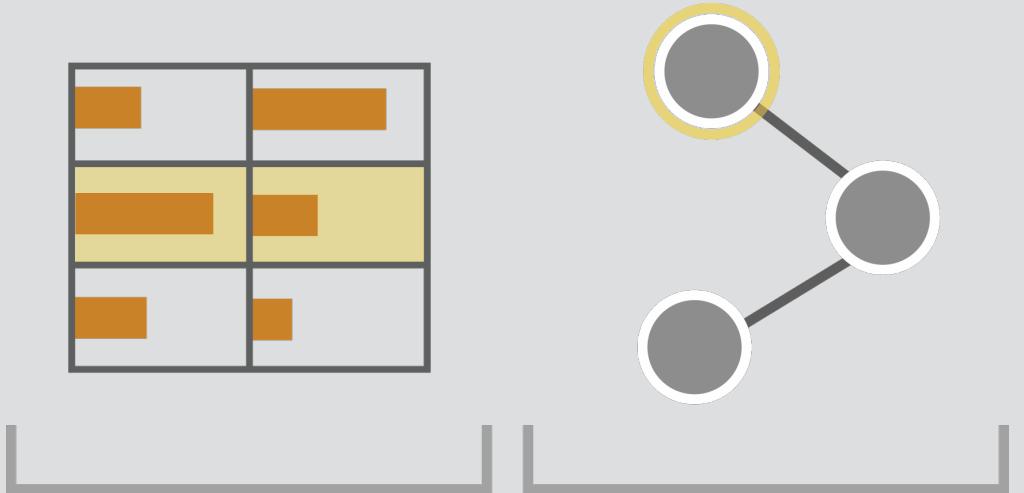
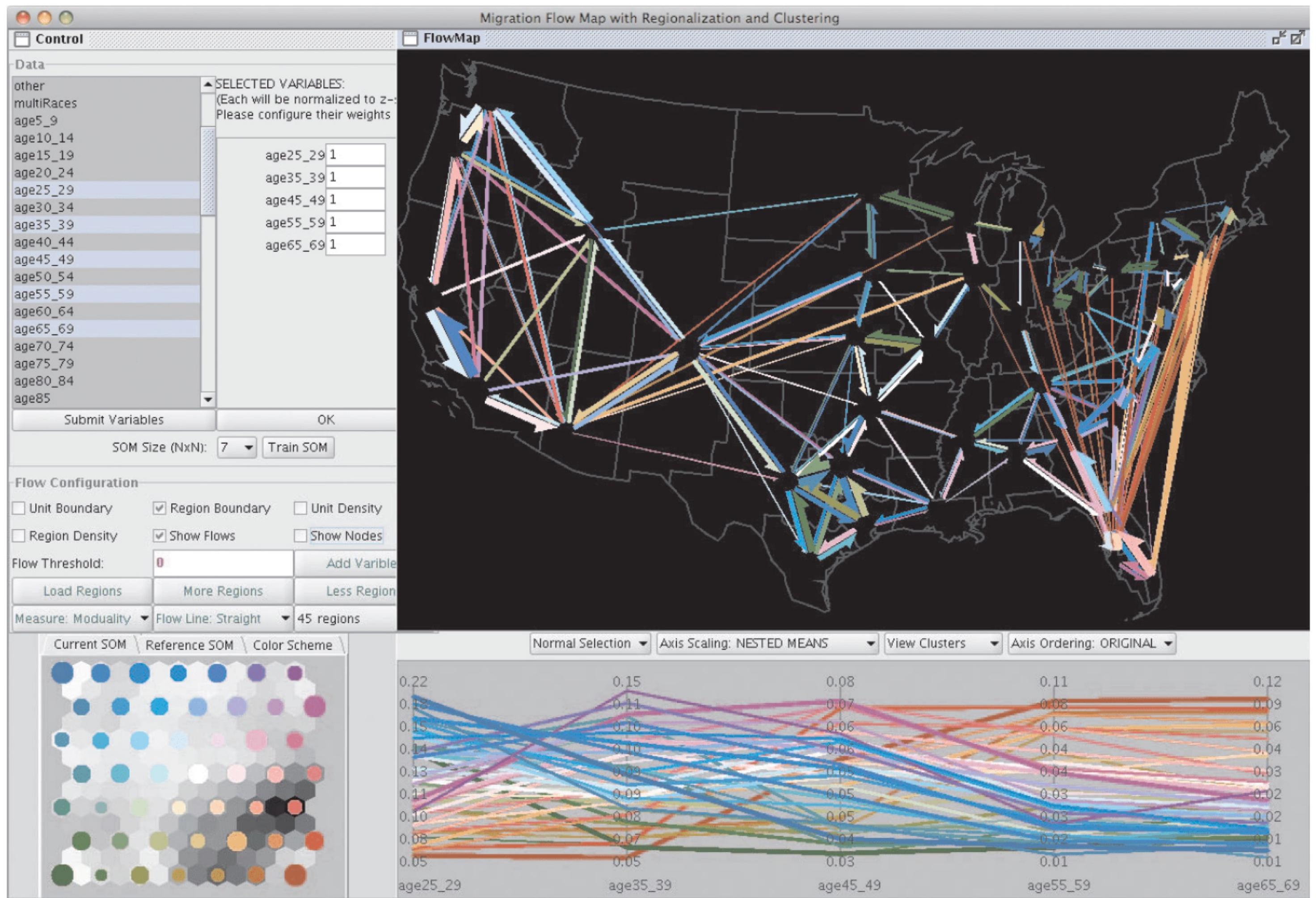
The figure shows two detailed views of network data. View (e) shows a grid of nodes with various properties like firstdate, papersub, and rank. View (f) shows a list of edges with source and target vertices and associated text. View (g) shows a zoomed-in view of a specific edge between 'Hearst' and 'Rao'.



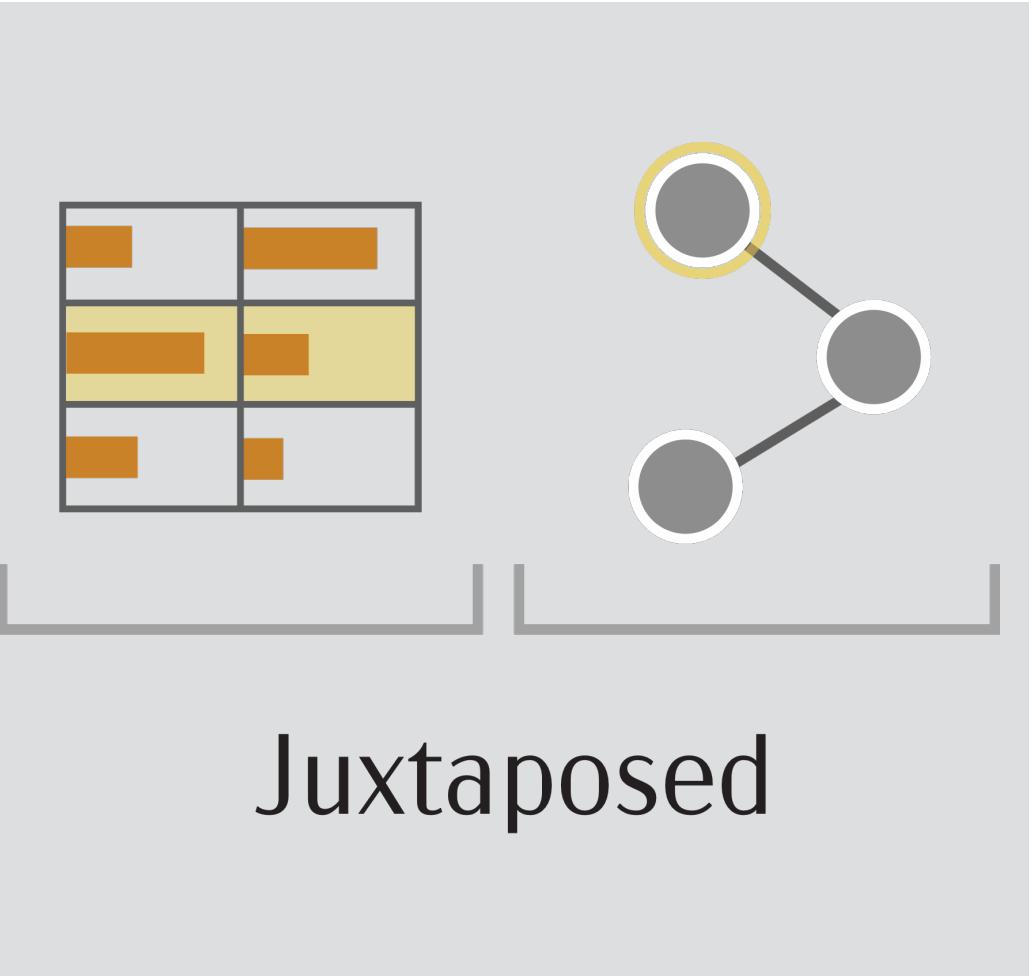
Graph Dice Bezerianos et al. 2010

Juxtaposed

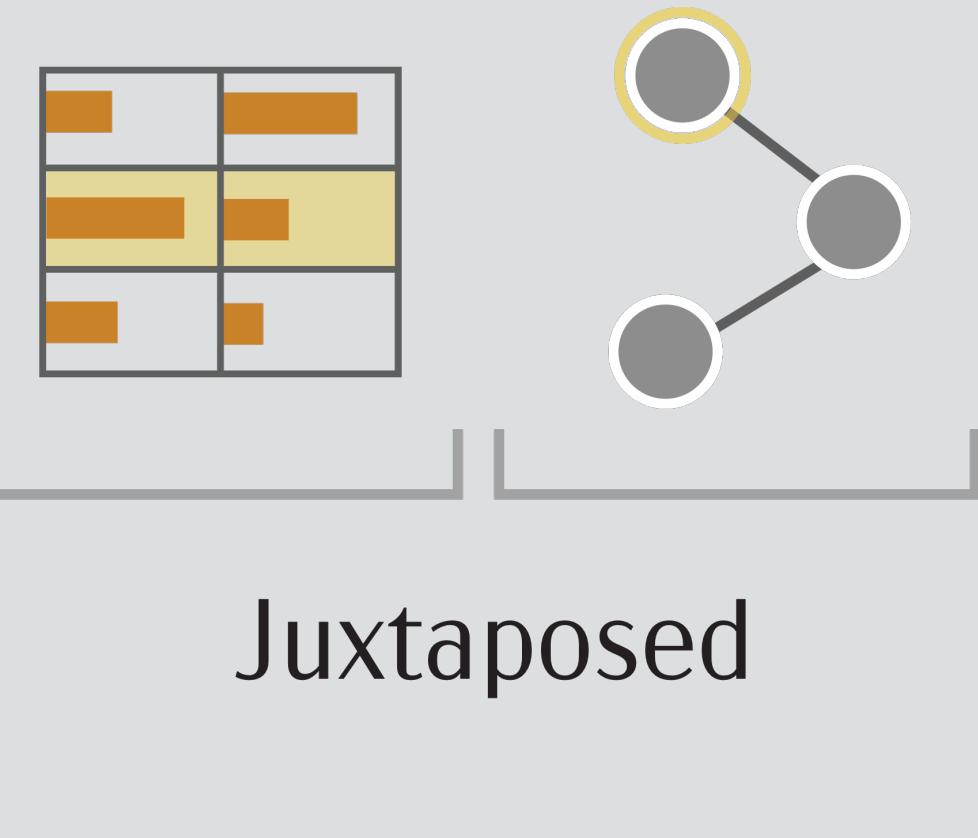
Guo, 2009



Juxtaposed



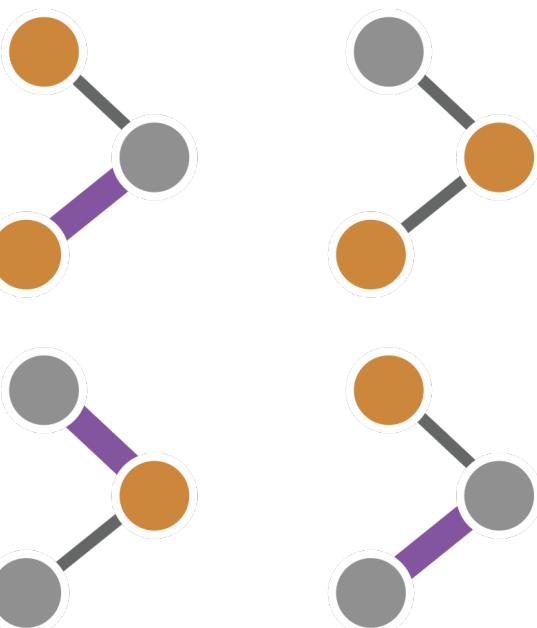
Independent views can optimize for topology and attribute independently.



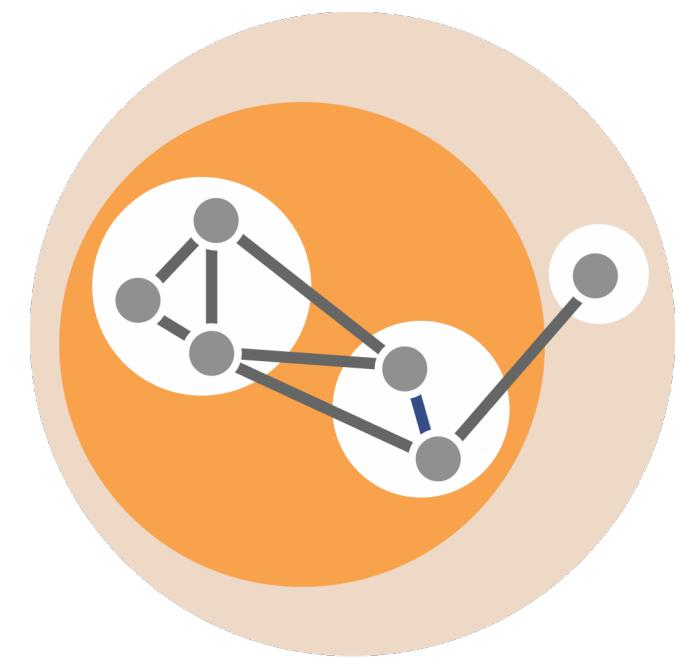
Not great for tasks on topological structures beyond a single node or edge.

Recommended for large networks and/or very large numbers or heterogeneous types of node and link attributes

Layout Operations

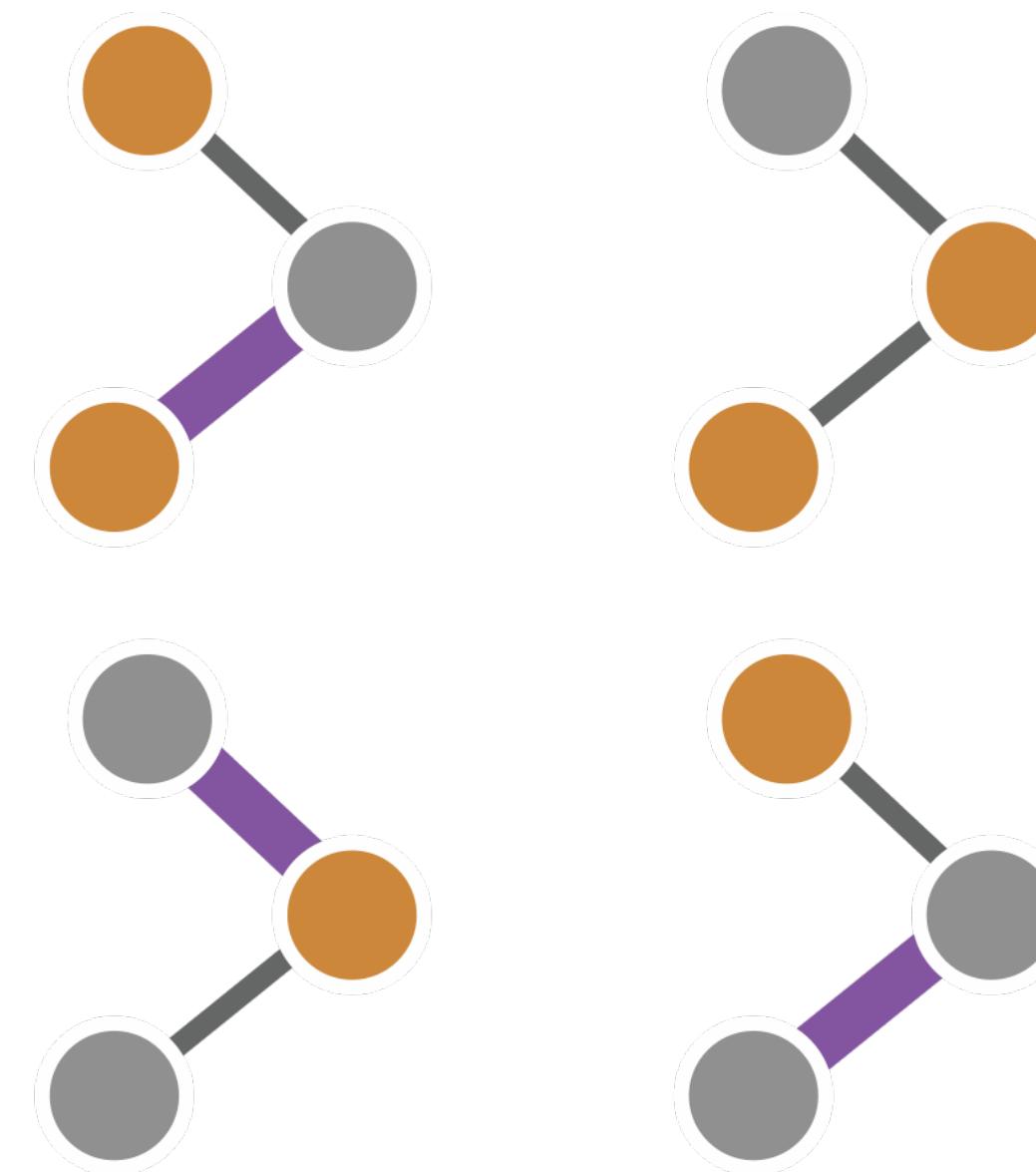


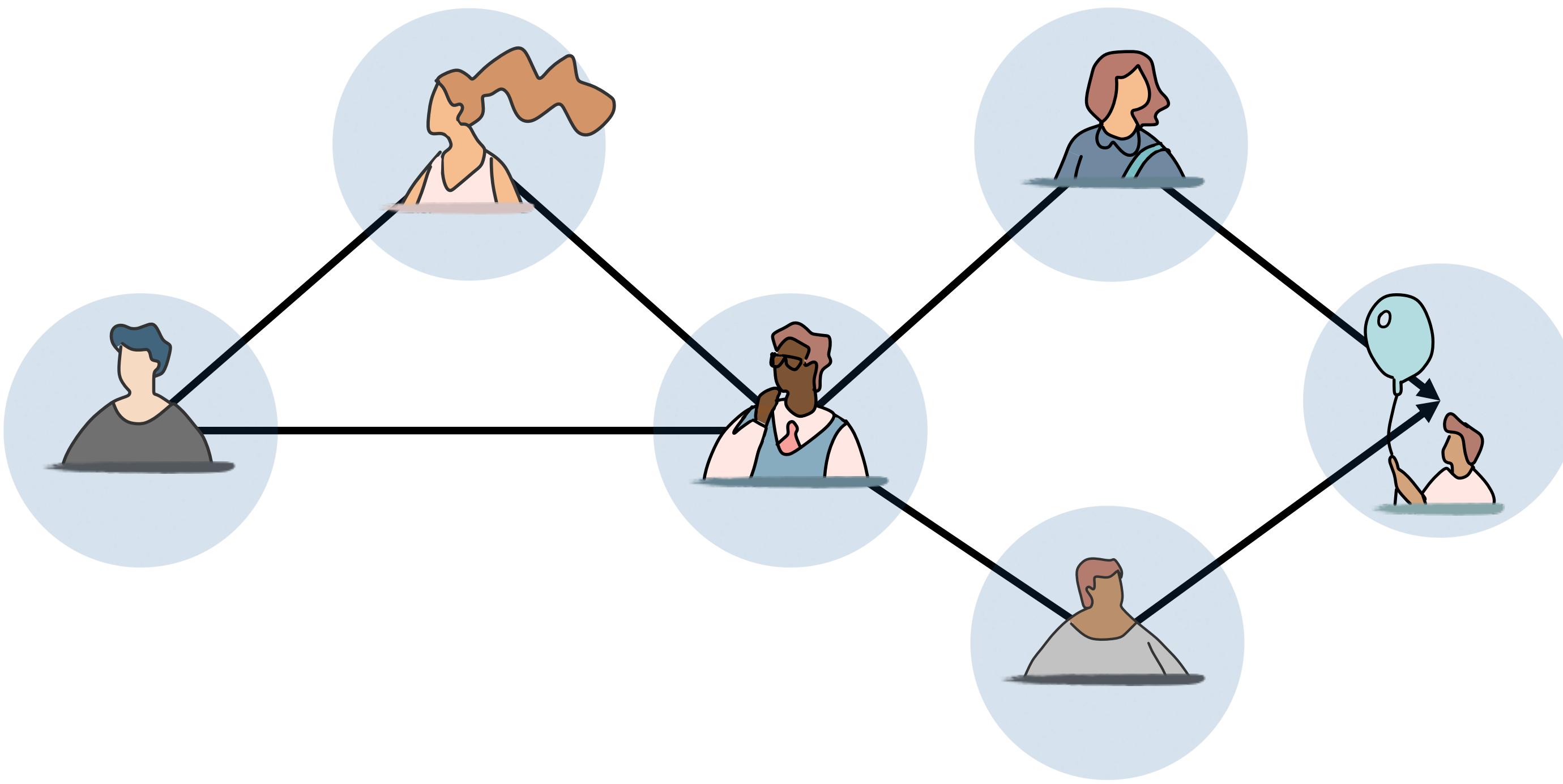
Small Multiples



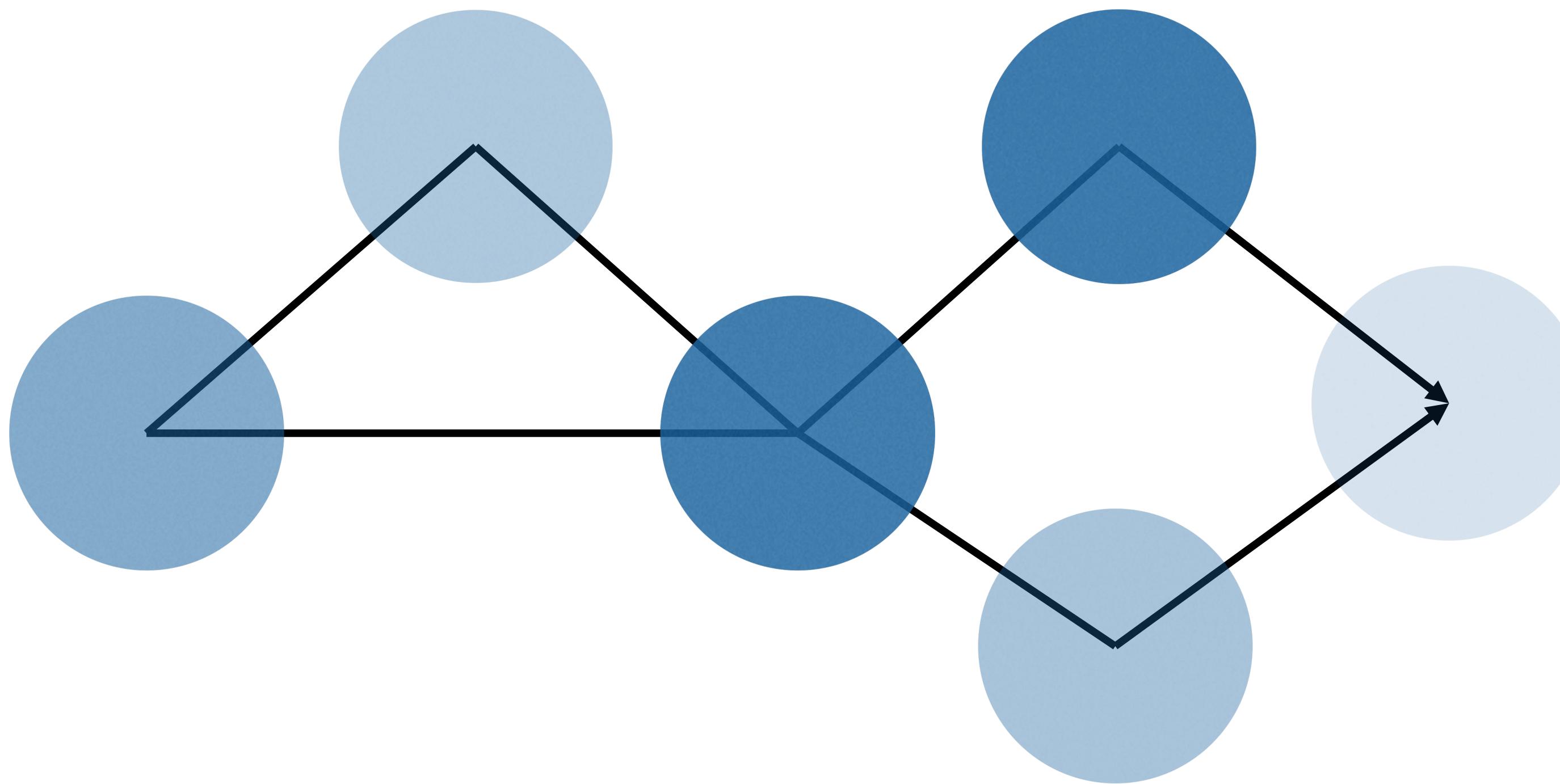
Hybrids

Small Multiples

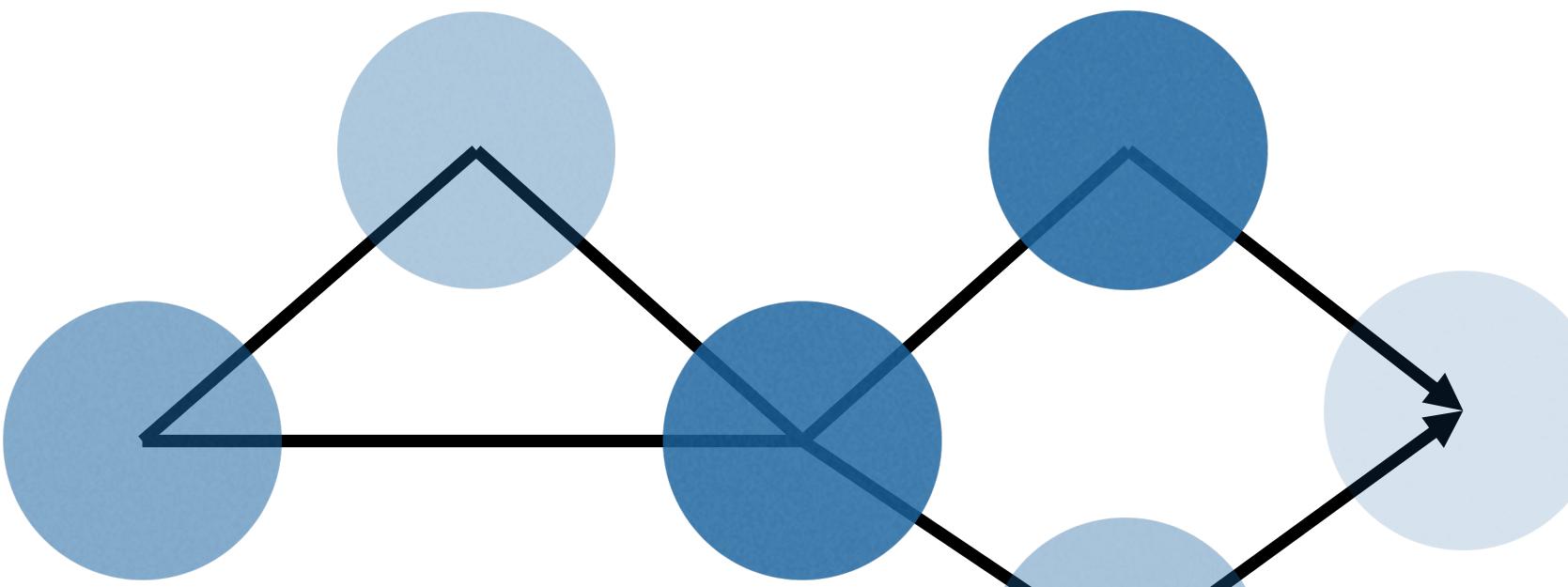




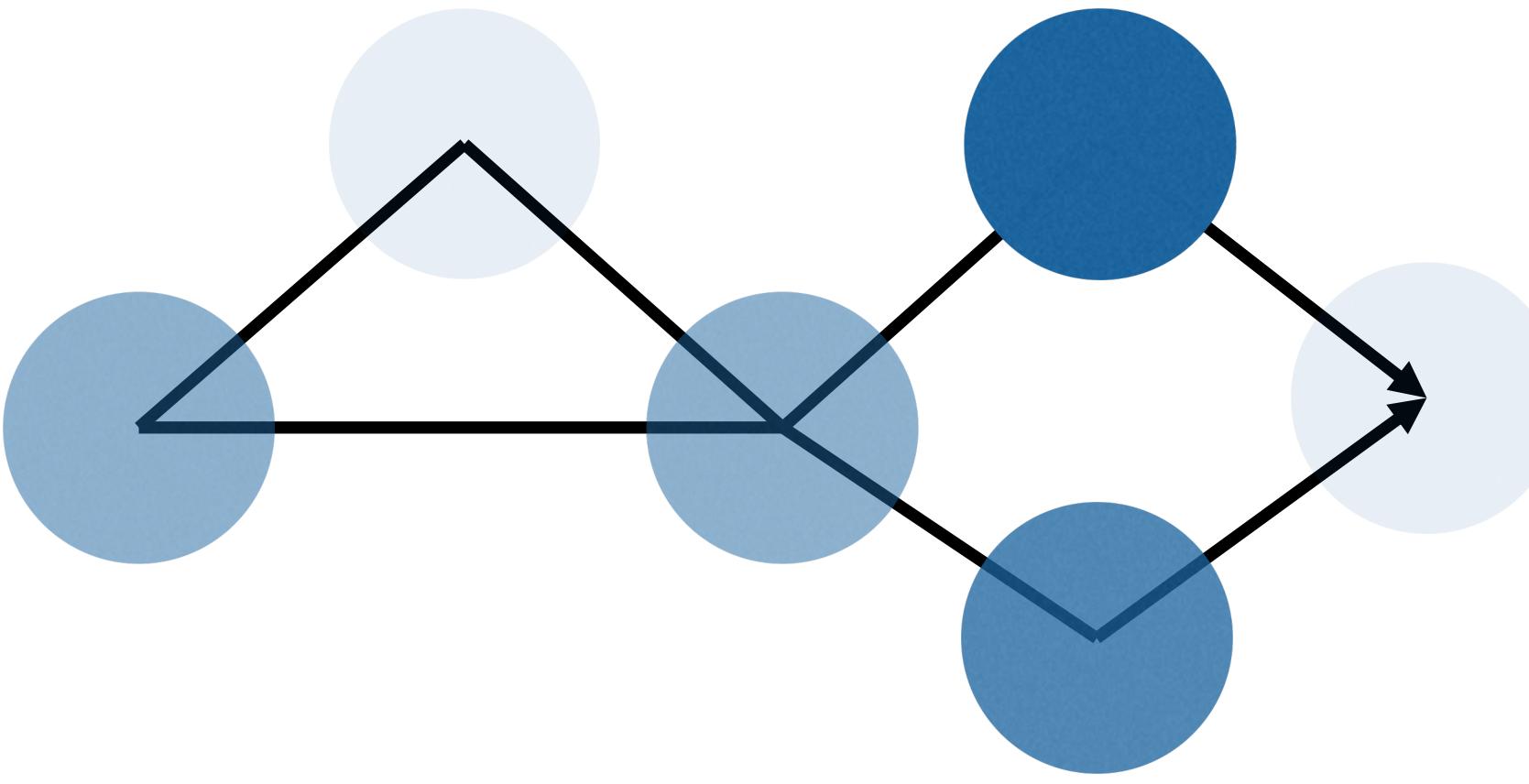
Day 1



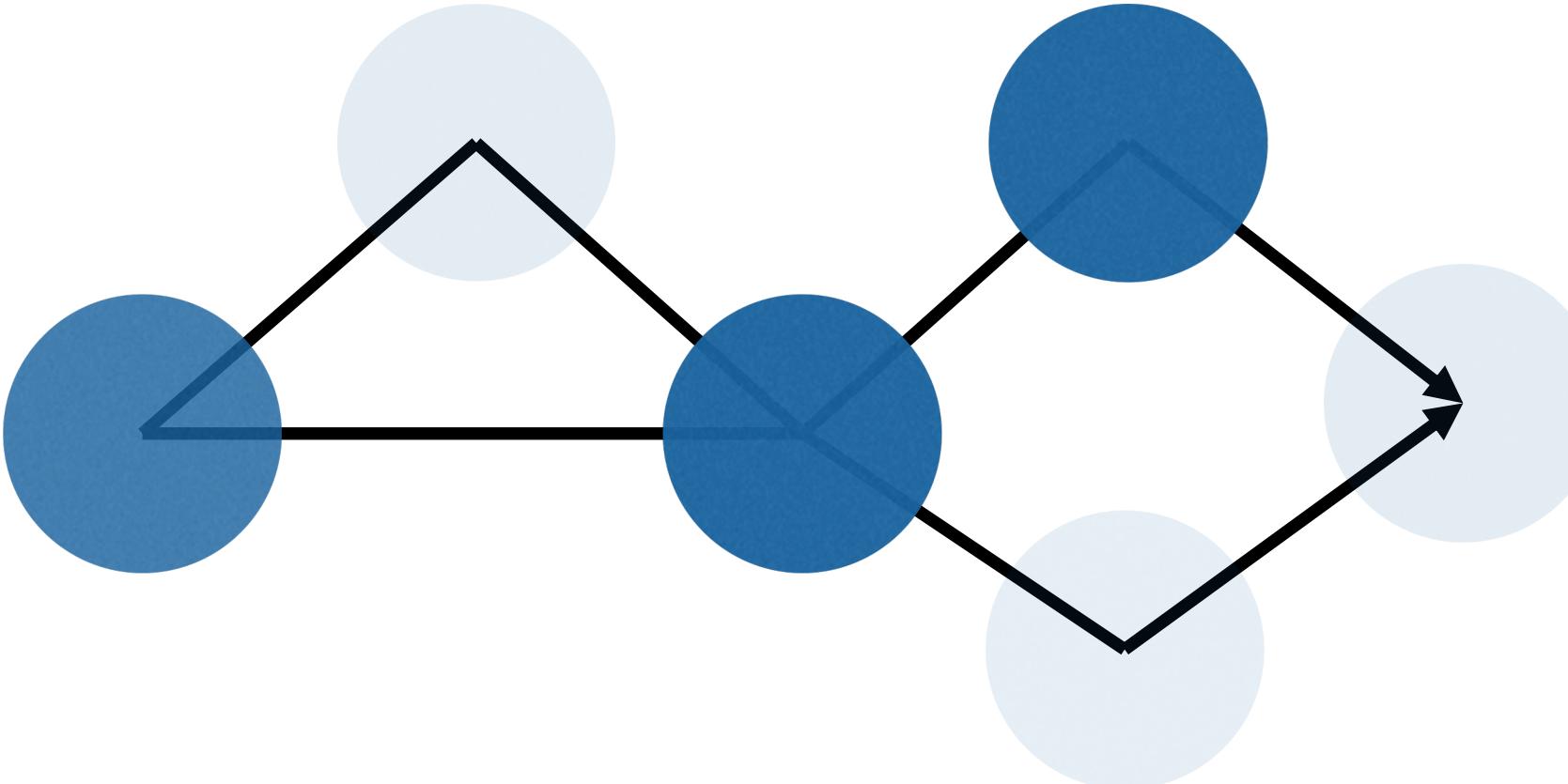
Day 1

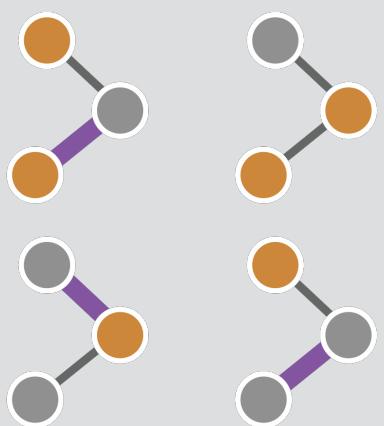
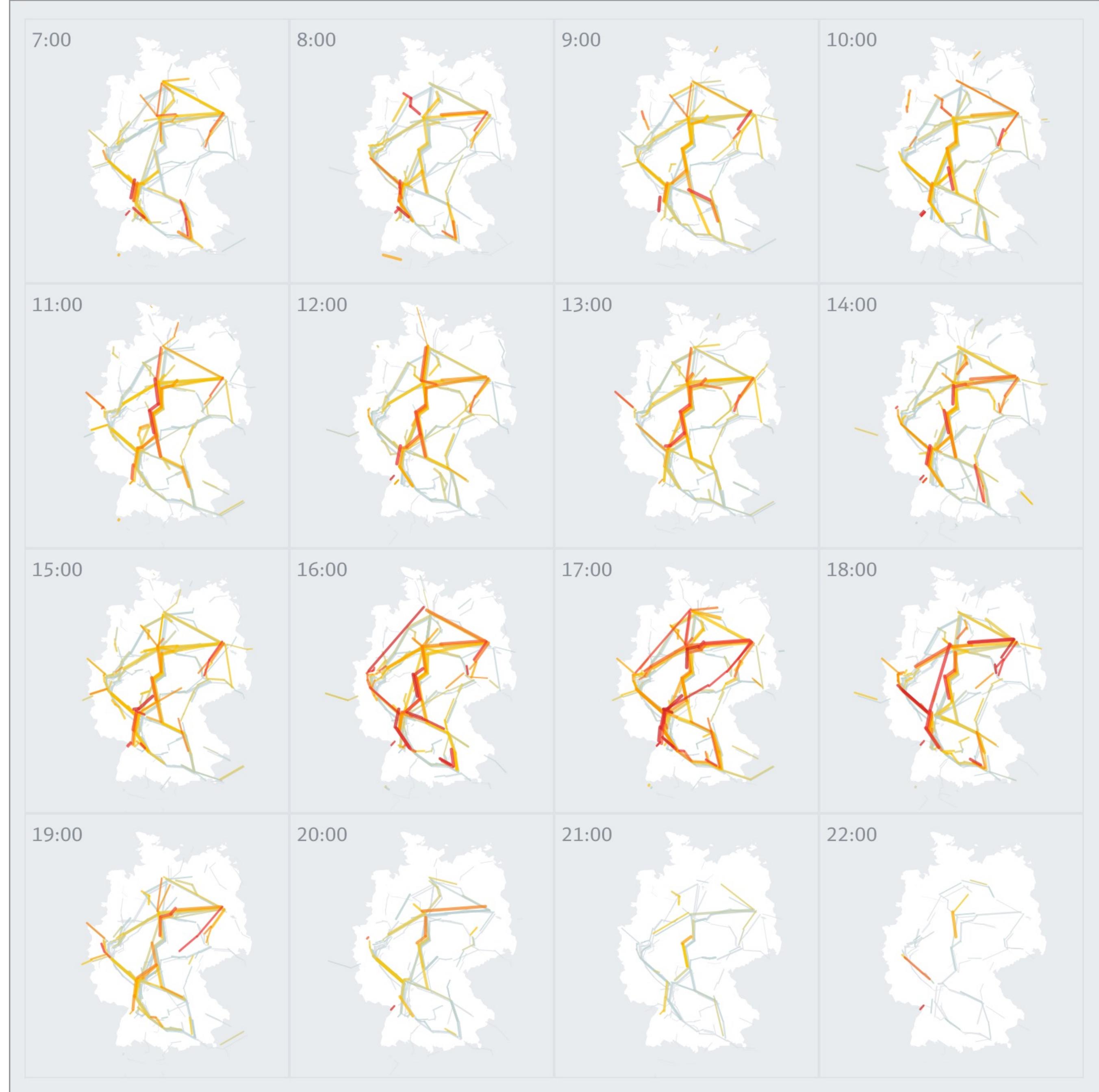


Day 2

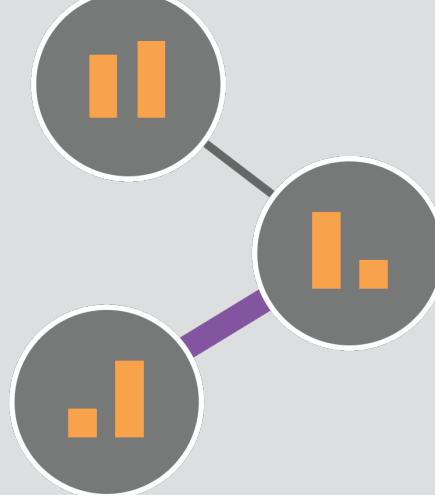


Day 3

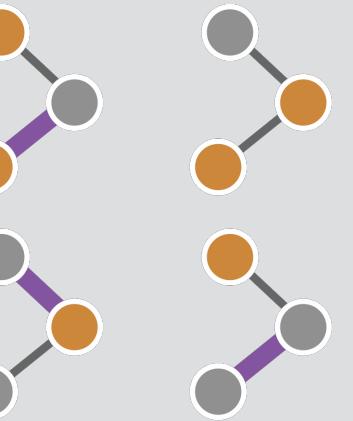
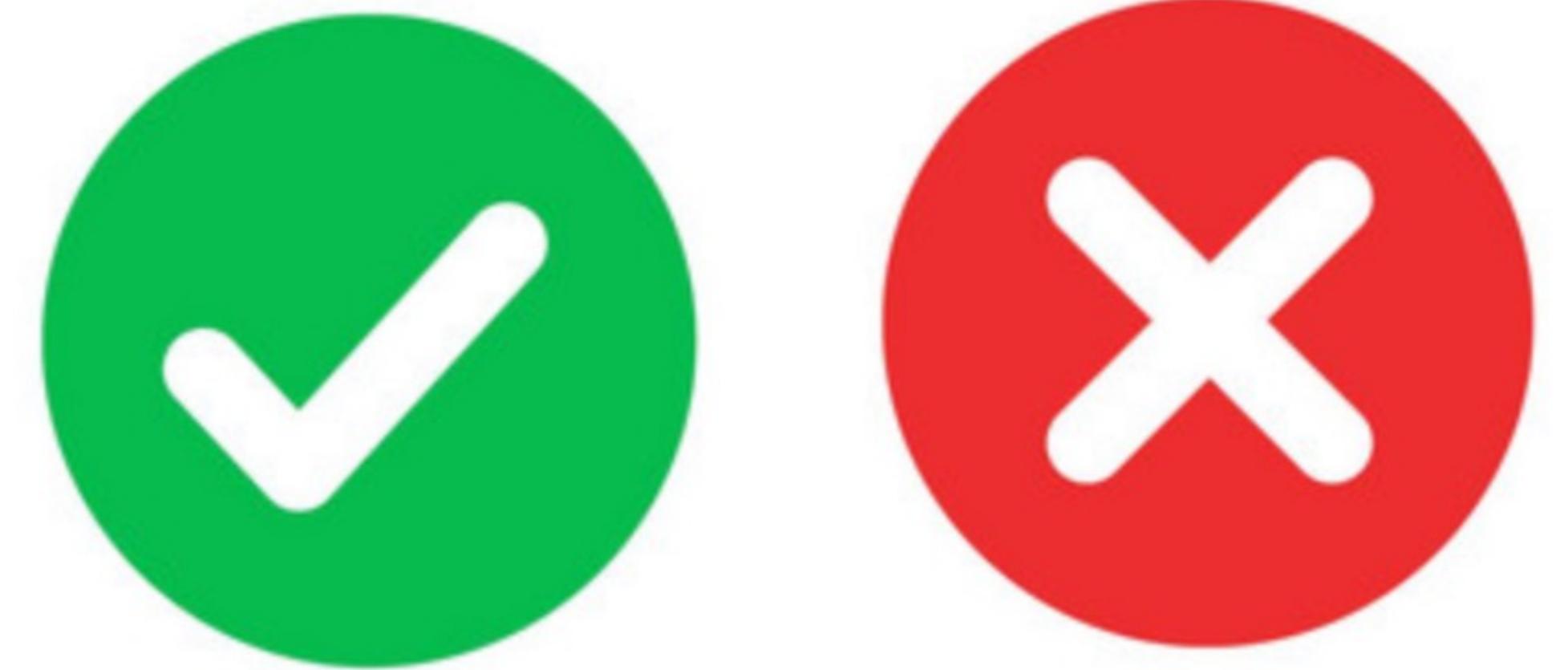




Small Multiples

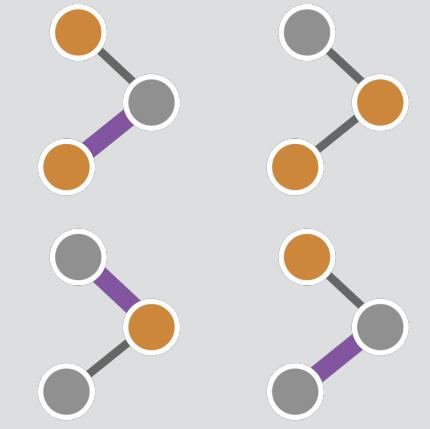


On-Node / On-Edge
Encoding



Small Multiples

Common layout facilitates attribute comparisons in specific topological features



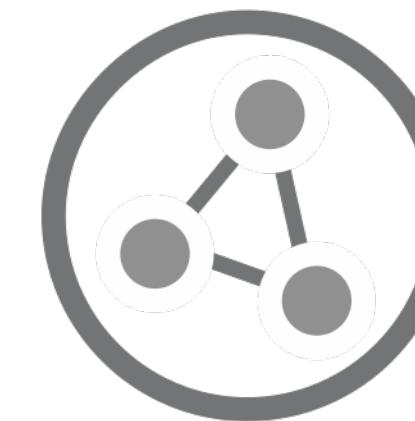
Small Multiples



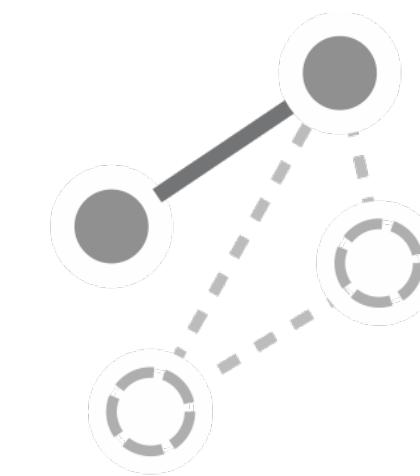
Not ideal for large networks, or tasks on clusters

Recommended for small networks where the tasks are focused on attribute comparison

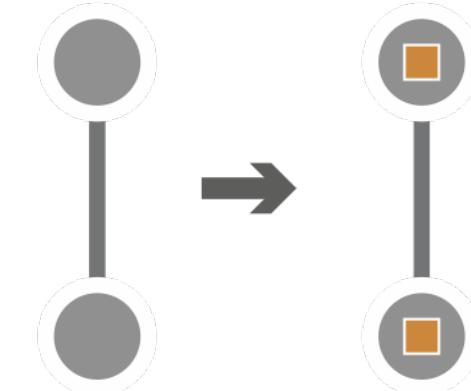
Data Operations



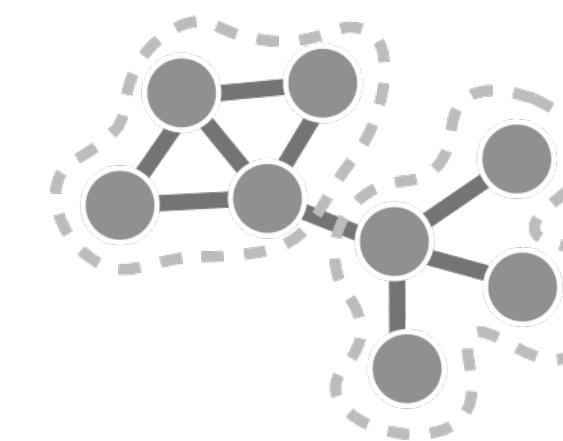
Aggregating Nodes/Edges



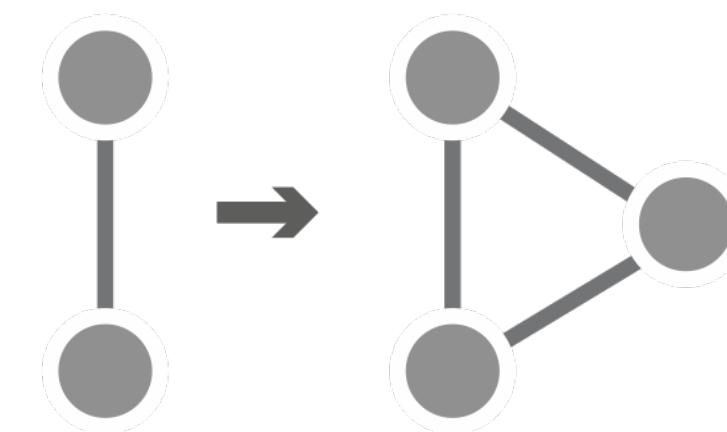
Querying and Filtering



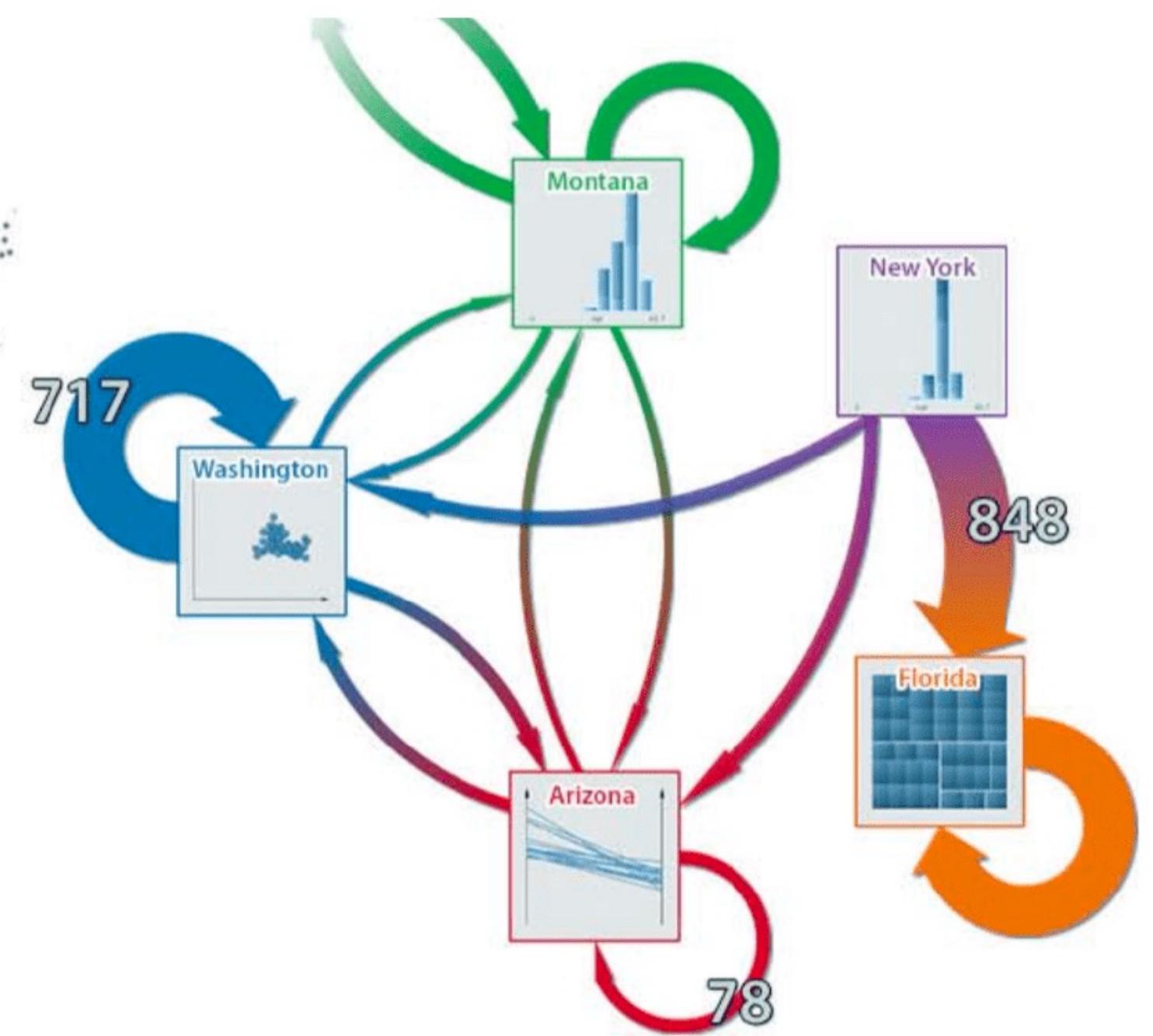
Deriving New Attributes



Clustering



Converting Attributes/Edge to Nodes



Aggregating Nodes/Edges

Elzen and Wijk, 2014

Multivariate Network Visualization Techniques

A companion website for the STAR Report on Multivariate Network Visualization Techniques.

HOME

TECHNIQUES

WIZARD

About

This is a companion website for a review article on multivariate network visualization techniques.

Multivariate networks are networks where both the structure of the network and the attributes of the nodes and edges matter. It turns out, these are very common. Every person in a social network, for example, has both, relationships and lots of other characteristics, such as their age, the school they went to, or the city they live in. Multivariate network visualization techniques are designed to be able to show both, these attributes and the structure. Using these visualization techniques, we can analyze, for example, if a network of friends predominantly went to the same high school.

The visualization research community has developed many techniques to visualize these kinds of networks, and our review article – and this website – are designed to help you sort through these options.

Browse through the techniques illustrated below, or use our wizard to find the right multivariate network visualization technique for your datasets and tasks!

[Get in touch](#) if you have questions or comments.

Use the Wizard

Technique recommendations to fit your needs!

Navigate to the [wizard tab](#) and select your specific network characteristics, such as the size of the network and its type, and what tasks are relevant for your analysis and receive technique recommendations that are best suited to your selection.

Read the Review Article

[The State of the Art in Visualizing Multivariate Networks](#)

Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex

To appear in Computer Graphics Forum (EuroVis 2019)

vdl.sci.utah.edu/mvnv/