

Web Science:

Visualizing Social Networks

(Part 1 - Graph Drawing and Layouts)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)



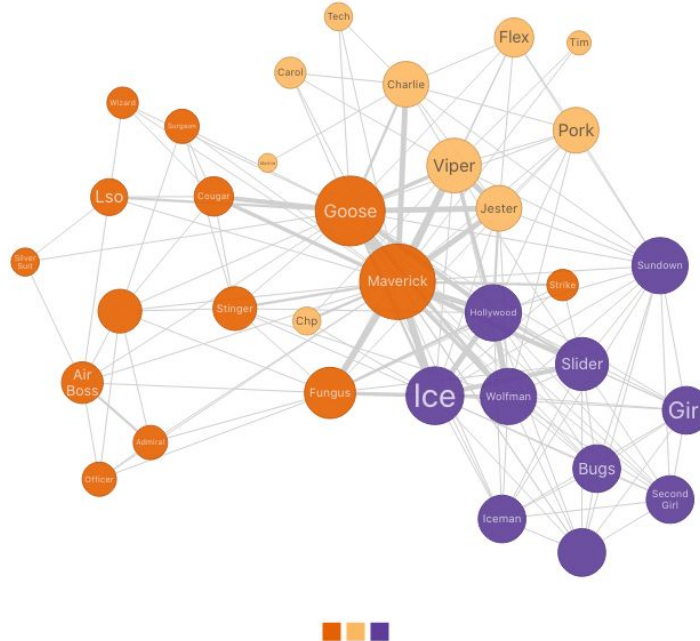
'Big Data' doesn't just mean increasing the font size

[Tall Infographics](#) (xkcd comic)

Top Gun

1986

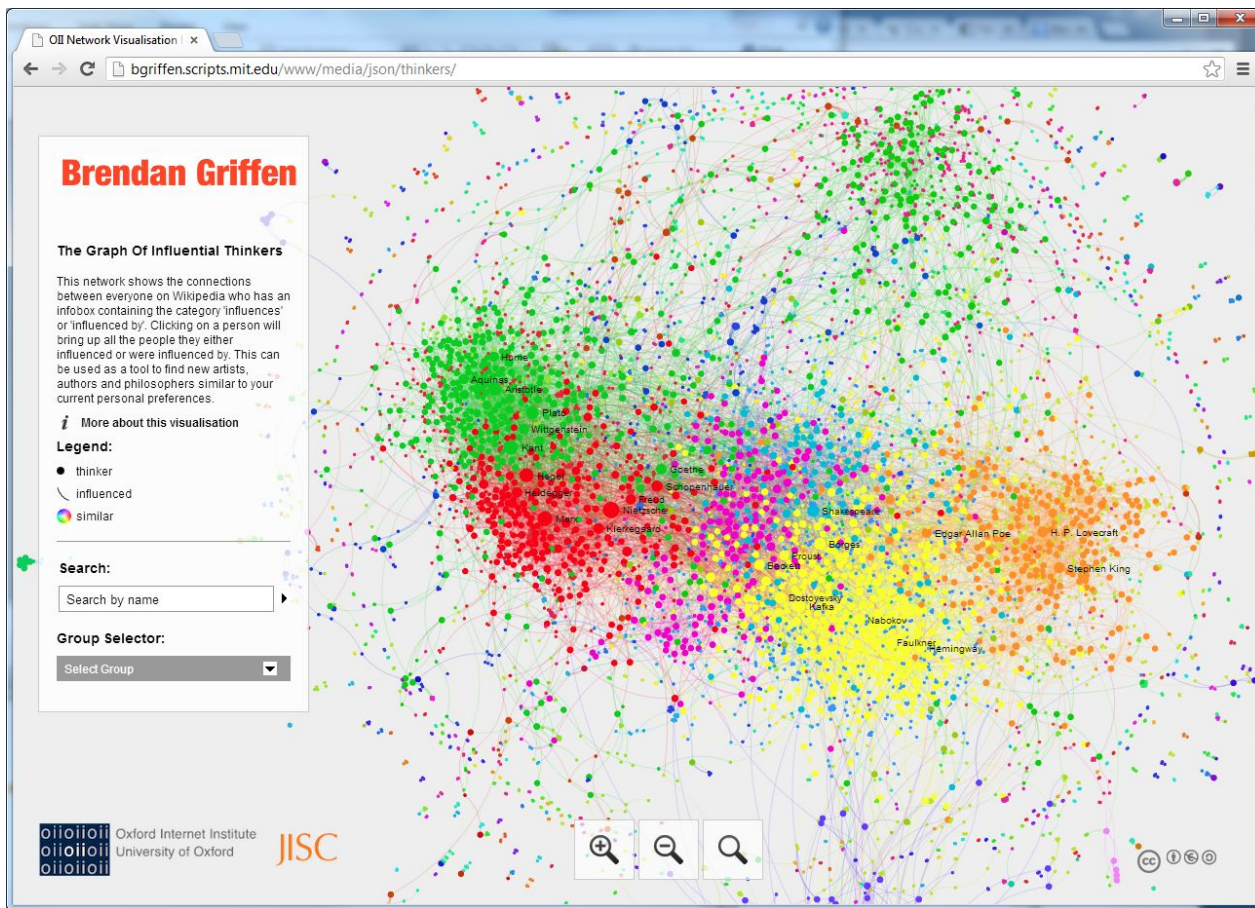
Network Characters Compare Resources Metrics



Maverick

Name	Maverick
Degree	31
Betweenness Centrality	180
Closeness Centrality	1.061
Eigenvector Centrality	0
Triangles	132
Eccentricity	2
Group	2
Pagerank	0.14

[Moviegalaxies](#) (Top Gun)



Alma mater University of Poitiers
University of Franeker
Leiden University

Era 17th-century philosophy

Region Western Philosophy

School Rationalism
Founder of Cartesianism

Main interests Metaphysics, epistemology, mathematics, physics, cosmology

Notable ideas *Cogito ergo sum*, method of doubt, method of normals, Cartesian coordinate system, Cartesian dualism, foundationalism, ontological argument for the existence of God, *mathesis universalis*, folium of Descartes, dream argument, evil demon, conservation of

Influences [hide]
Plato, Aristotle, Archimedes, Alhazen, Al-Ghazali,^[3] Averroes, Avicenna, Anselm, Augustine, Stoics, Aquinas, Ockham, Suarez, Mersenne, Sextus Empiricus, Montaigne, Golius, Beeckman, Duns Scotus^[4]

Influenced [hide]
Virtually all subsequent Western philosophy, especially Spinoza, Leibniz, John Locke, Nicolas Malebranche, Jacques-Bénigne Bossuet^[5] Blaise Pascal, Isaac Newton, Immanuel Kant, Johann Gottlieb Fichte, Edmund Husserl, Noam Chomsky, Slavoj Žižek, David Chalmers

Signature

René Descartes

Graphs of Wikipedia: Influential Thinkers

René Descartes (Wikipedia)

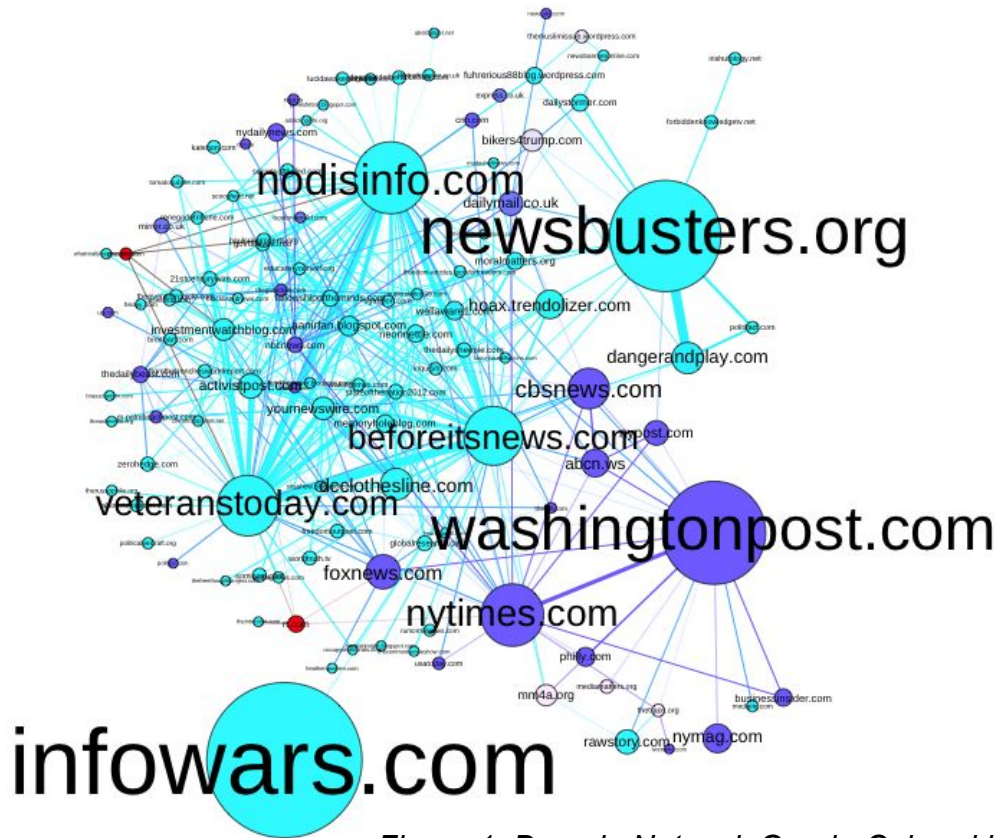


Figure 1. Domain Network Graph, Colored by Media Type

Purple = mainstream media; Aqua = alternative media; Red = government controlled media

Tweets about shooting events that also contained terms such as “false flag”, “hoax”, and “crisis actor”

Node - domain

Edge - if the same Twitter account posted tweets citing the two domains (in separate tweets)

Creating a Visualization

1. Obtain the **data**
2. Convert data into **format** appropriate for importing
3. Import into visualization **software**
4. Choose a **layout**
5. Tinker, tinker, tinker
6. Wow your friends and family

Graph Data Formats

	Edge List/Matrix	Structure	XML	Edge Weight	Attributes	Visualization	Attribute Default Value	Hierarchical Graphs	Dynamics
CSV									
DL Ucinet									
DOT Graphviz									
GDF									
GEXF									
GML									
GraphML									
NET Pajek									
TLP Tulip									
VNA Netdraw									
Spreadsheet*									

[Supported Graph Formats](https://gephi.org) (gephi.org)

GEXF

GEXF (Graph Exchange XML Format) is a language for describing complex networks structures, their associated data and dynamics. Started in 2007 at Gephi.

This is a minimal file for a static graph containing 2 nodes and 1 edge between them:

<http://gexf.net/data/hello-world.gexf>

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.2draft" version="1.2">
  <meta lastmodifieddate="2009-03-20">
    <creator>Gexf.net</creator>
    <description>A hello world! file</description>
  </meta>
  <graph mode="static" defaultedgetype="directed">
    <nodes>
      <node id="0" label="Hello" />
      <node id="1" label="Word" />
    </nodes>
    <edges>
      <edge id="0" source="0" target="1" />
    </edges>
  </graph>
</gexf>
```


GML

GML (Graph Modeling Language) is a text file format supporting network data with a very easy syntax.

It is used by Graphlet, Pajek, yEd, LEDA and NetworkX.

[GML Format](#)

```
graph
[
  node
  [
    id A
    label "Node A"
  ]
  node
  [
    id B
    label "Node B"
  ]
  node
  [
    id C
    label "Node C"
  ]
  edge
  [
    source B
    target A
    label "Edge B to A"
  ]
  edge
  [
    source C
    target A
    label "Edge C to A"
  ]
]
```

GraphML

The GraphML file format uses .graphml extension and is XML structured. It supports attributes for nodes and edges, hierarchical graphs and benefits from a flexible architecture. This format is supported by NodeXL, Sonivis, GUESS and NetworkX.

Basic Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
<graph id="G" edgedefault="undirected">
<node id="n0"/>
<node id="n1"/>
<edge id="e1" source="n0" target="n1"/>
</graph>
</graphml>
```

GraphViz DOT

DOT is the text file format of the suite GraphViz. It has a human-readable syntax that describes network data, including subgraphs and elements appearances (i.e. color, width, label). NetworkX, Tulip or ZGRViewer can import DOT files as well.

```
digraph sample2 {  
  A -> B [ label = "Edge A to B" ];  
  B -> C [ label = "Edge B to C" ];  
  A [label="Node A"];  
}
```

JSON

JSON is the preferred file format for JavaScript-based tools, such as d3.js.

The Python library NetworkX can write out JSON.

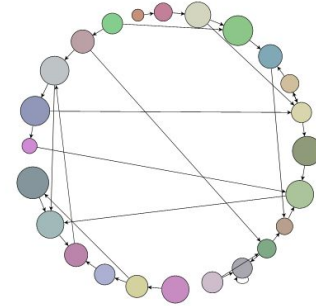
["Data preparation with R for networks in d3.js"](#) has R code to convert from adjacency matrix, edge list, and edge and node lists to JSON

[JSON — NetworkX 2.5 documentation](#)

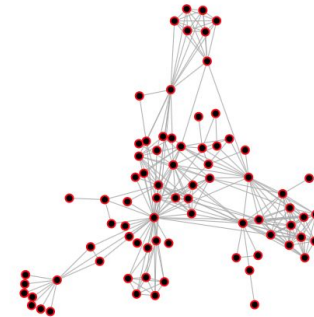
```
{ nodes: [  
  { name: "Adam" },  
  { name: "Bob" },  
  { name: "Carrie" },  
  { name: "Donovan" },  
  { name: "Edward" },  
  { name: "Felicity" },  
  { name: "George" },  
  { name: "Hannah" },  
  { name: "Iris" },  
  { name: "Jerry" }  
],  
links: [  
  { source: 0, target: 1 },  
  { source: 0, target: 2 },  
  { source: 0, target: 3 },  
  { source: 0, target: 4 },  
  { source: 1, target: 5 },  
  { source: 2, target: 5 },  
  { source: 2, target: 5 },  
  { source: 3, target: 4 },  
  { source: 5, target: 8 },  
  { source: 5, target: 9 },  
  { source: 6, target: 7 },  
  { source: 7, target: 8 },  
  { source: 8, target: 9 }  
]}
```

Graph Layouts

- Circular - nodes are placed on a circle at even spacings
- Force-directed - uses a physics simulator to emulate gravity and force which can be applied to elements
- Many others...



Img source: <https://www.nwoods.com/products/goxam/layout.html>



Img source: <https://bl.ocks.org/steveharoz/8c3e2524079a8c440df60c1ab72b5d03>

Force-Directed Layout

- Assigns forces on the nodes and edges
- Spring-like forces *attract* connected nodes
- *Repulsion* forces (like electrically charged particles) separate pairs of nodes
- In equilibrium
 - edges tend to have uniform length (because of the spring forces)
 - nodes that are not connected by an edge tend to be drawn further apart (because of the electrical repulsion)
- Gravity can be added to pull nodes toward a certain location (to prevent disconnected nodes from flying off)

Web Science:

Visualizing Social Networks

(Part 2 - Graph Creation Software)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Graph Creation Software

- Many good commercial products available
- Here we only cover free, open-source software
- By no means exhaustive

Stand-Alone Software

- [Gephi](#)
 - Interactive visualization and exploration platform
- [Graphviz](#)
 - Many libraries from other programming languages use it
- [NodeXL](#)
 - Microsoft Excel template

Gephi, an open source gra x
gephi.org
Download Blog Store Wiki Forum Support Bugtracker
Home Features Learn Develop Plugins Services Consortium

Gephi

makes graphs handy

The Open Graph Viz Platform

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

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

[Learn More on Gephi Platform >](#)

Download FREE
Gephi 0.8.2-beta

Release Notes | System Requirements

Features
Quick start

Screenshots
Videos

Gephi 0.8.1-beta has been released! Discover a new Timeline, dynamic ranking and weighted community detection. [Learn More >>](#)

APPLICATIONS

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

“Like Photoshop™ for graphs.”
— the Community

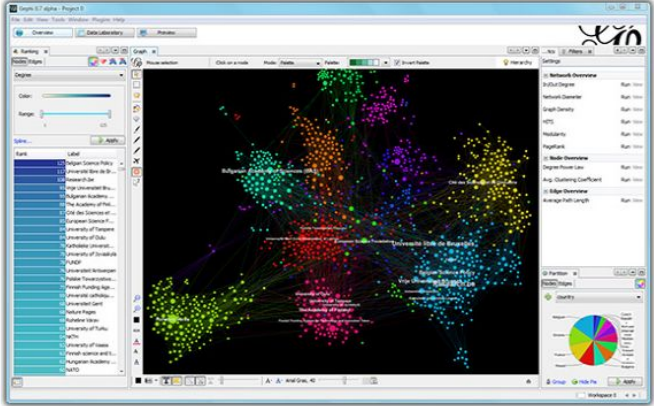
LATEST NEWS

- Rebuilding Gephi's core for the 0.9 versi..
March 5, 2013
- A month of Gephi Marketplace
February 20, 2013
- rgexf: An R library to work with GEXF graph fil..
February 12, 2013
- Graph visualization meet-up in Paris

PAPERS


Gephi: An Open Source Software for Exploring and Manipulating Networks.
Matthieu Latapy, Jérôme Lemaître, Vincent Loebe, 2012

Visualization and Exploration of Large Graphs.
Matthieu Latapy, Jérôme Lemaître, Vincent Loebe, 2012



Gephi - The Open Graph Viz Platform

Graphviz | Graphviz - Grap
www.graphviz.org



Graphviz - Graph Visualization Software

Drawing graphs since 1988

[Forums](#) | [Wiki](#) | [Contact Us](#)

- Home
- About
- Download
- News
- Gallery
- Documentation
- Theory
- Bug and Issue Tracking
- Mailing List
- License
- Resources
- Credits
- Forums
- FAQ
- Wiki


User login

Username: *

Password: *

- [Create new account](#)
- [Request new password](#)

Graphviz



Welcome to Graphviz

Available translations: Belorussian, Romanian, Russian, Russian (more natural?) Serbo-Croatian

What is Graphviz?

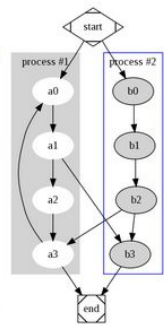
Graphviz is open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. It has important applications in networking, bioinformatics, software engineering, database and web design, machine learning, and in visual interfaces for other technical domains.

Features

The Graphviz layout programs take descriptions of graphs in a simple text language, and make diagrams in useful formats, such as images and SVG for web pages, PDF or Postscript for inclusion in other documents; or display in an interactive graph browser. (Graphviz also supports GXL, an XML dialect.) Graphviz has many useful features for concrete diagrams, such as options for colors, fonts, tabular node layouts, line styles, hyperlinks, rolland custom shapes.

Roadmap

dot "hierarchical" or layered



Active forum topics

- [Graphviz on Windows 7 / 64 bits - doesn't work](#)
- [Left justify multiline node text?](#)
- [Coordinates in a graph](#)
- [Align edge labels to fit its path in SVG output](#)
- [Workaround for splines=ortho crash?](#)

[more](#)

New forum topics

- [Graphviz on Windows 7 / 64 bits - doesn't work](#)
- [Coordinates in a graph](#)
- [Workaround for splines=ortho crash?](#)
- [Left justify multiline node text?](#)
- [Unable to install binray graphviz-2.28.0-1.el6.x86_64.rpm](#)

[more](#)

Graphviz - Graph Visualization Software

NodeXL: Network Overview

nodexl.codeplex.com

CodePlex Project Hosting for Open Source Software


Register Sign In Search all projects

NODEXL Network Graphs
The Social Media Research Foundation


NodeXL: Network Overview, Discovery and Exploration for Excel

HOME SOURCE CODE DOWNLOADS DOCUMENTATION DISCUSSIONS ISSUE TRACKER PEOPLE LICENSE

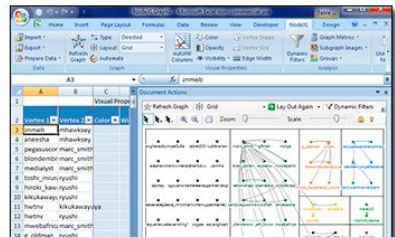
Page Info Change History (all pages) ★ Follow (246) Subscribe






 **socialmedia**
RESEARCH FOUNDATION

OPEN TOOLS, OPEN DATA, OPEN SCHOLARSHIP FOR SOCIAL MEDIA

find us @  Office Marketplace

NodeXL is a free, open-source template for Microsoft® Excel® 2007 and 2010 that makes it easy to explore network graphs. With NodeXL, you can enter a network edge list in a worksheet, click a button and see your graph, all in the familiar environment of the Excel window.



Search Wiki & Documentation

download

CURRENT	NodeXL Excel Template, version 1.0.1.229
DATE	Sat Nov 24, 2012 at 2:00 AM
STATUS	Beta
DOWNLOADS	18,660
RATING	★★★★★ 1 rating

RECENT REVIEWS

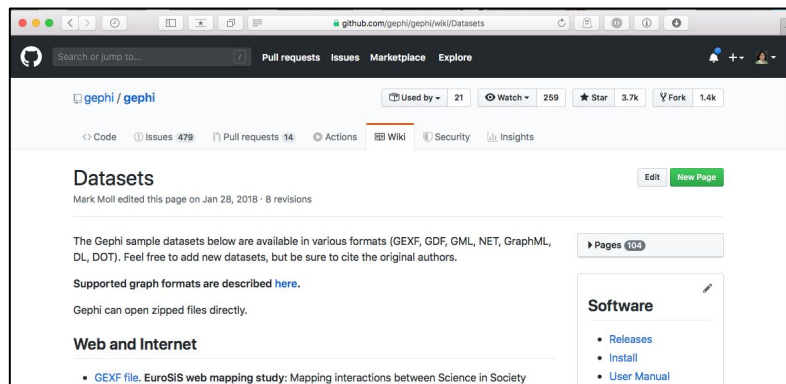
★★★★★ Congratulations for an amazing project! I've tested several visualization tools before and I'm quite glad with NodeXL which has gre... (more)

[View all reviews](#)

NodeXL

Gephi Example

Download data file from [Gephi Datasets](#)



- **GML file. Zachary's karate club:** social network of friendships between 34 members of a karate club at a US university in the 1970s. W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977).



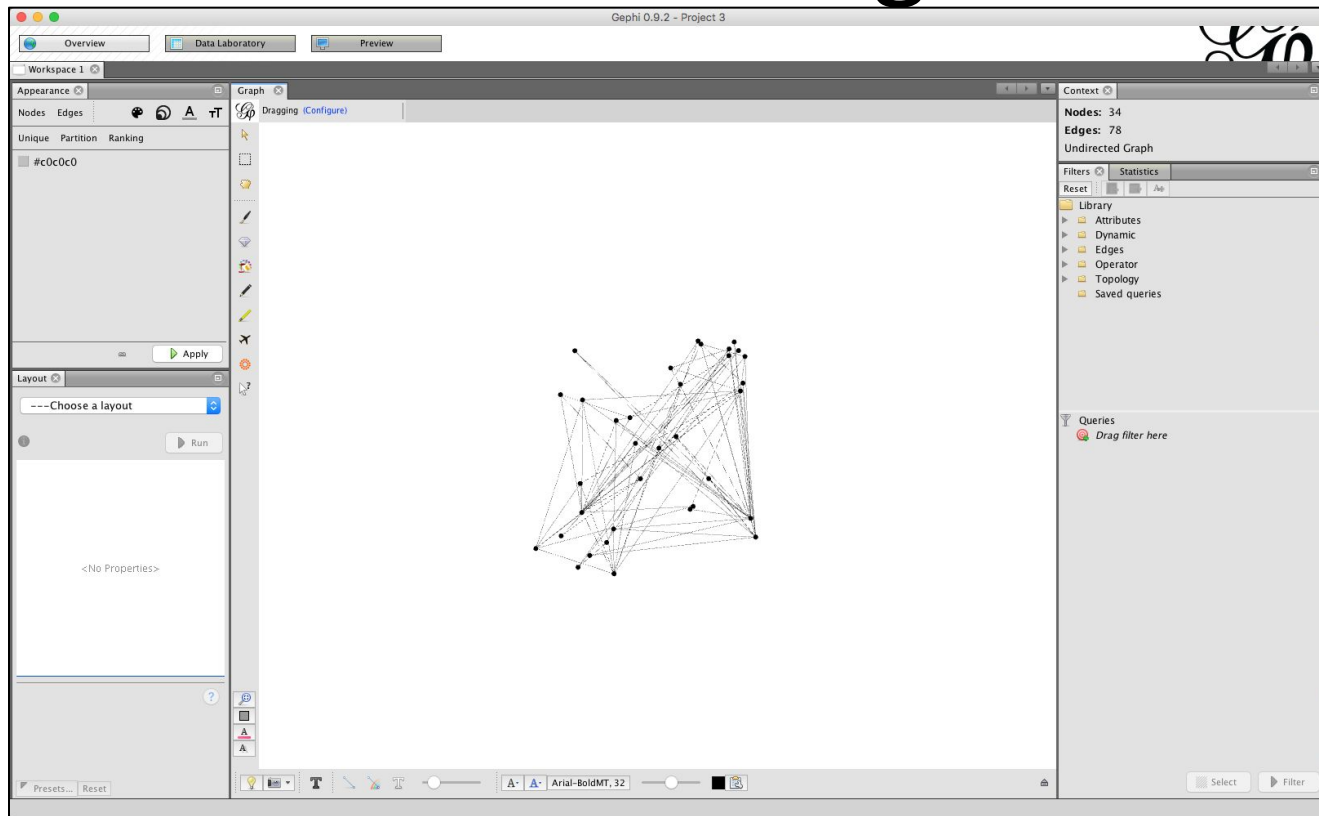
- **GEXF file. HyperText 2009 dynamic contact network:** Contact network during the HyperText 2009 conference. Source: Sociopatterns.org.
- **GEXF file. CLASS OF 1880/81:** friendship network of a German boys' school class from 1880/1881. It's based on the probably first ever primarily collected social network dataset, assembled by the primary school teacher Johannes Deltsch. The data was reanalyzed and compiled for the article: Heidler, R., Gamper, M., Herz, A., Eßer, F. (2014): Relationship patterns in the 19th century: The friendship network in a German boys' school class from 1880 to 1881 revisited. Social Networks 13: 1--13.
- **GML file. Zachary's karate club:** social network of friendships between 34 members of a karate club at a US university in the 1970s. W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977).
- **GML file. Coauthorships in network science:** coauthorship network of scientists working on network theory and experiment, as compiled by M. Newman in May 2006. A figure depicting the largest component of this network can be found here. M. E. J. Newman, Phys. Rev. E 74, 036104 (2006).

- Import Dynamic Data
- Scripting Plugin

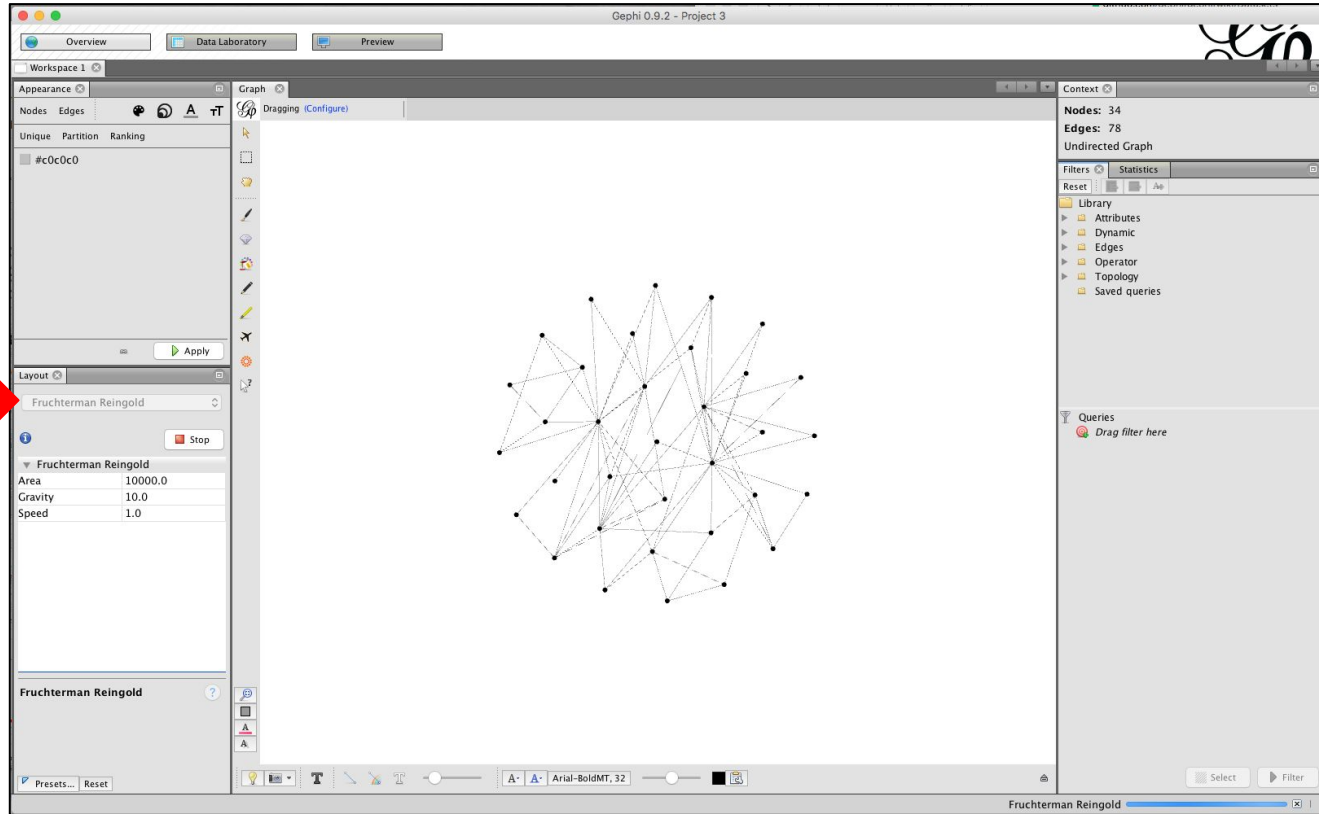
Community

- Manifesto
- Roadmap
- Idea List
- GSoC
- Code-in
- Student Program
- Connectors

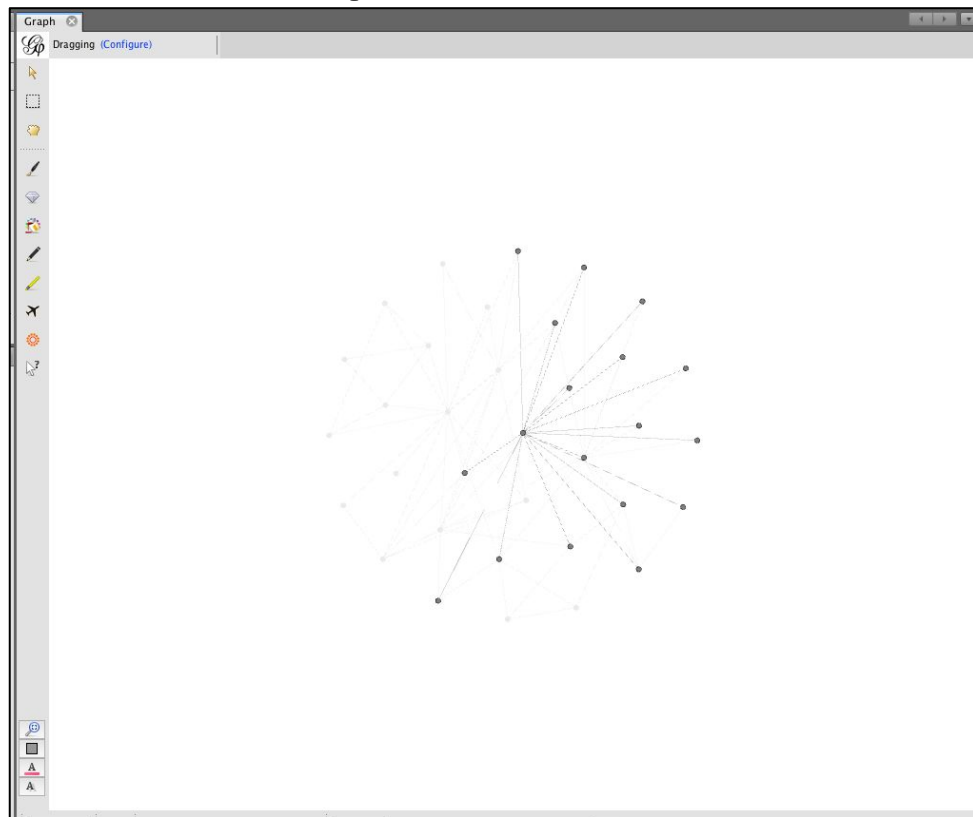
Load karate.gml



Set the layout



Hover to explore connections



Explore the data - nodes

The screenshot shows the Gephi 0.9.2 interface with the 'Data Laboratory' tab selected. The 'Nodes' tab is also selected within the Data Laboratory. A table of nodes is displayed with columns for 'Id', 'Label', and 'Interval'. The 'Id' column contains numbers from 1 to 34, and the 'Label' column contains corresponding labels. The 'Interval' column is empty. The bottom toolbar contains various data manipulation tools.

Id	Label	Interval
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	
11	11	
12	12	
13	13	
14	14	
15	15	
16	16	
17	17	
18	18	
19	19	
20	20	
21	21	
22	22	
23	23	
24	24	
25	25	
26	26	
27	27	
28	28	
29	29	
30	30	
31	31	
32	32	
33	33	
34	34	

Bottom toolbar tools: Add column, Merge columns, Delete column, Clear column, Copy data to other column, Fill column with a value, Duplicate column, Create a boolean column from regex match, Create column with list of regex matching groups, Negate boolean values, Convert column to dynamic.

Explore the data - edges

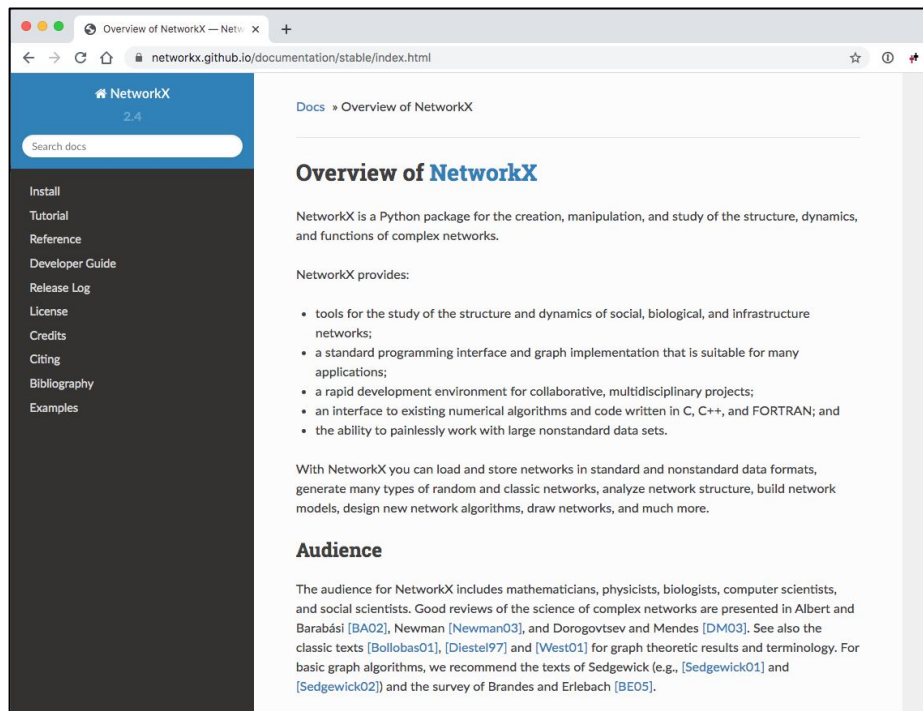
The screenshot shows the Gephi 0.9.2 interface with the 'Data Laboratory' tab selected. The 'Edges' table is active, displaying a list of edges with columns for Source, Target, Type, Id, Label, Interval, and Weight. The 'Edges' tab is circled in red. The 'Configuration' tab is also circled in red. The table contains 29 rows of data, all with a weight of 1.0. The bottom toolbar includes buttons for 'Add column', 'Merge columns', 'Delete column', 'Clear column', 'Copy data to other column', 'Fill column with a value', 'Duplicate column', 'Create a boolean column from regex match', 'Create column with list of regex matching groups', 'Negate boolean values', and 'Convert column to dynamic'.

Source	Target	Type	Id	Label	Interval	Weight
1	1	Undirected	156			1.0
3	1	Undirected	157			1.0
3	2	Undirected	158			1.0
4	1	Undirected	159			1.0
4	2	Undirected	160			1.0
4	3	Undirected	161			1.0
5	1	Undirected	162			1.0
6	1	Undirected	163			1.0
7	1	Undirected	164			1.0
7	5	Undirected	165			1.0
7	6	Undirected	166			1.0
8	1	Undirected	167			1.0
8	2	Undirected	168			1.0
8	3	Undirected	169			1.0
8	4	Undirected	170			1.0
9	1	Undirected	171			1.0
9	3	Undirected	172			1.0
10	3	Undirected	173			1.0
11	1	Undirected	174			1.0
11	5	Undirected	175			1.0
11	6	Undirected	176			1.0
12	1	Undirected	177			1.0
13	1	Undirected	178			1.0
13	4	Undirected	179			1.0
14	1	Undirected	180			1.0
14	2	Undirected	181			1.0
14	3	Undirected	182			1.0
14	4	Undirected	183			1.0
17	6	Undirected	184			1.0
17	7	Undirected	185			1.0
18	1	Undirected	186			1.0
18	2	Undirected	187			1.0
20	1	Undirected	188			1.0
20	2	Undirected	189			1.0
22	1	Undirected	190			1.0
22	2	Undirected	191			1.0
26	24	Undirected	192			1.0
26	25	Undirected	193			1.0
28	3	Undirected	194			1.0
28	24	Undirected	195			1.0
28	25	Undirected	196			1.0
29	3	Undirected	197			1.0

Python Libraries

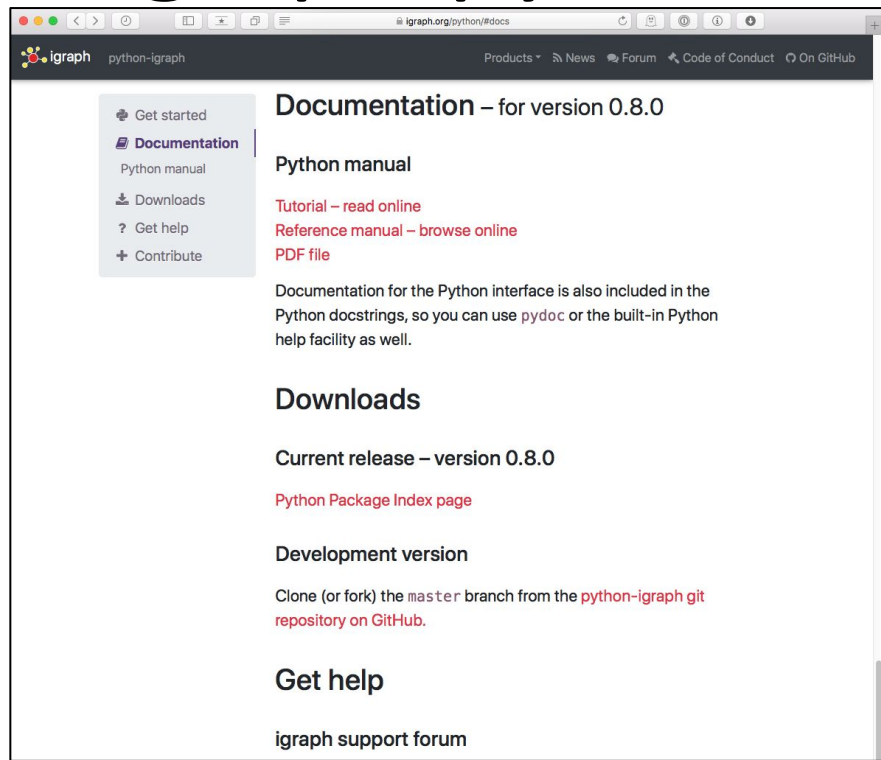
- [NetworkX](#) – creating and manipulating graphs
 - uses Matplotlib or Graphviz for displaying graphs
- [igraph](#) – creating, manipulating, and displaying graphs

NetworkX



Tutorials, Documentation at [Software for Complex Networks — NetworkX 2.5 documentation](https://networkx.github.io/documentation/stable/index.html)
Additional examples at [NETWORK CHART](https://networkx.github.io/documentation/stable/index.html)

igraph-python

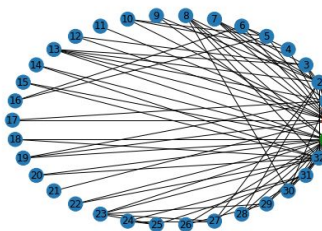


Tutorials, Documentation at [python-igraph](https://python-igraph.org)

NetworkX Examples in Google Colab

```
node_color = ["#f78b4"] * 34  
node_color[0] = "red"  
node_color[33] = "green"  
node_color  
nx.draw_circular(G, with_labels=True, node_color=node_color)  
plt.show()
```

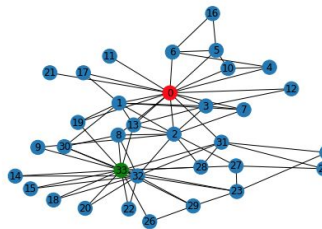
↳



Draw with force-directed layout

```
[41] nx.draw_kamada_kawai(G, with_labels=True, node_color=node_color)  
plt.show()
```

↳



JavaScript Graphing Libraries

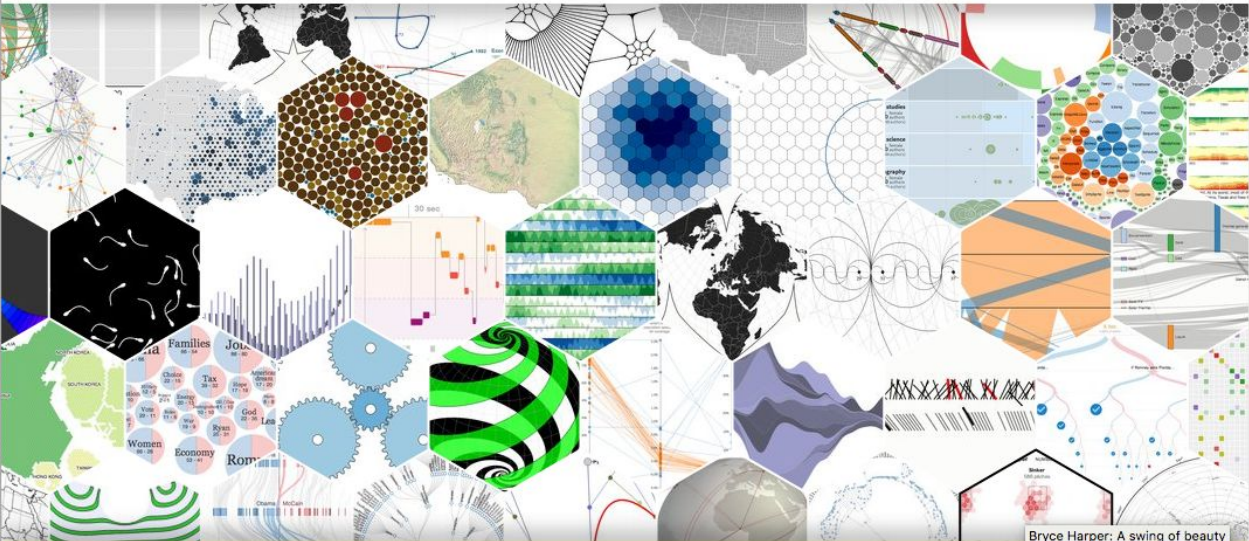
- [D3.js](#)
 - library for all types of visualization, also has higher-level APIs (Vega, Vega-Lite)
- [Arbor.js](#)
 - library for graph visualization using jQuery, uses HTML canvas, so it won't work in older browsers.
- [Cytoscape.js](#)
 - library for graph theory analysis and visualization.
- [Sigma.js](#)
 - lightweight library for graph visualization, also uses HTML canvas

D3.js - Data-Driven Documents

Overview Examples Documentation API Source

Data-Driven Documents

Fork me on GitHub

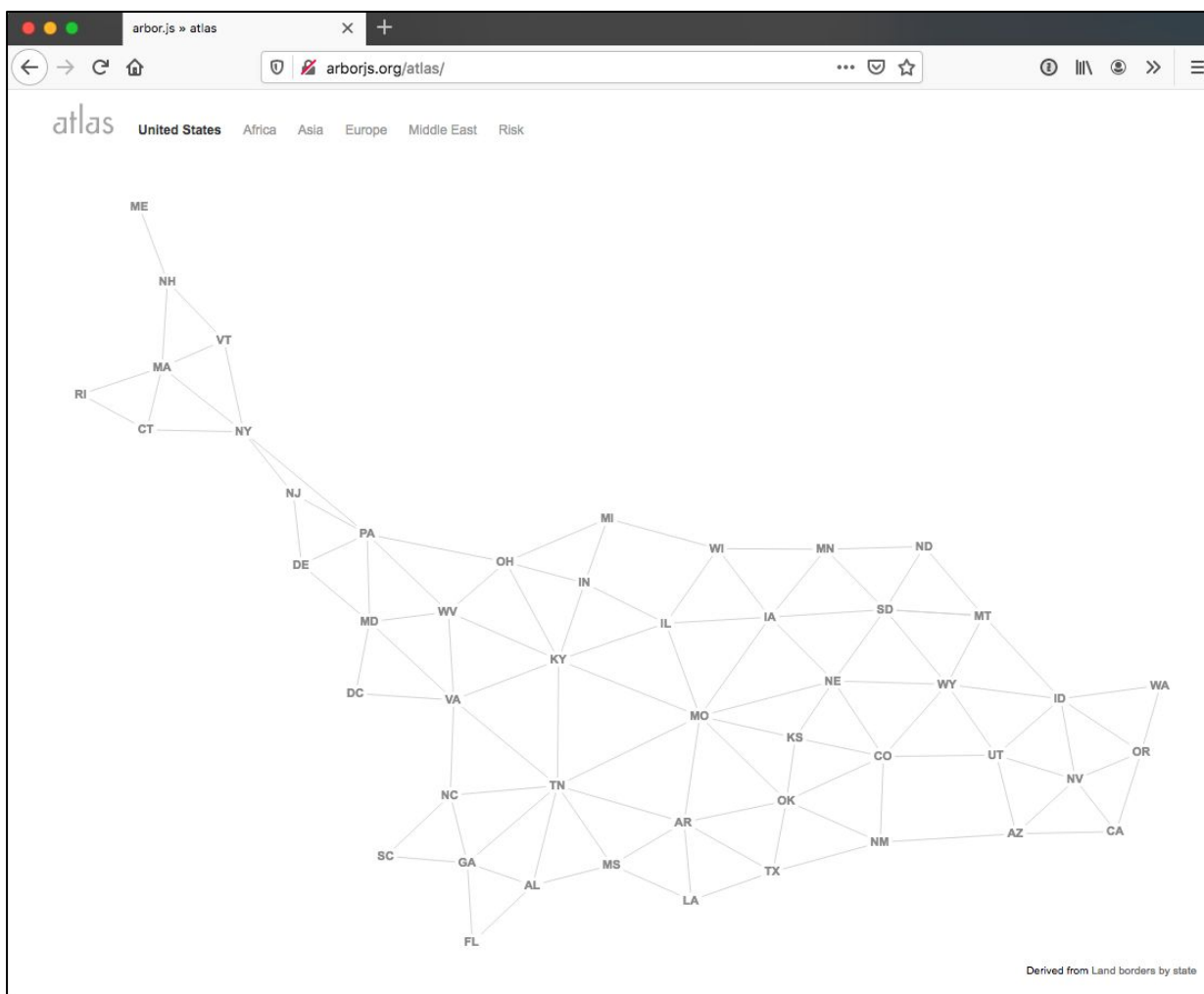


Bryce Harper: A swing of beauty

Like visualization and creative coding? Try interactive JavaScript notebooks in **Observable**!

www.washingtonpost.com/wp-srv/special/sports/bryce-harper-swing-of-beauty/...nts based on data. **D3** helps you bring data to See more examples.

[D3.js - Data-Driven Documents](https://d3js.org)



[arbor.js](https://arborjs.org)

Cytoscape.js

Find sections

Demos

Introduction

- Factsheet
- About
- Packages
- Releases
- Citation
- Funding

Notation

- Graph model
- Architecture & API
- Functions
- Object ownership
- Gestures
- Position
- Elements JSON
- Compound nodes

Getting started

- Including Cytoscape.js
- Initialisation
- Specifying basic options
- Next steps

Core

- Initialisation
- Graph manipulation
 - `cy.add()`
 - `cy.remove()`
 - `cy.collection()`
 - `cy.getElementById()`

Cytoscape.js

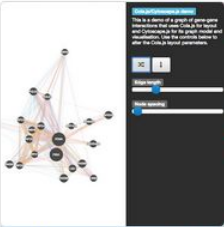
Graph theory (network) library for visualisation and analysis

Repo [GitHub](#) Updates [Twitter](#) News and tutorials [Blog](#) Questions [StackOverflow](#) Ask a question [StackOverflow](#)


License [MIT](#) DOI [10.5281/zenodo.3658315](#) Cite [Oxford Bioinformatics Article](#) npm [v3.14.0](#) Download [v3.14.0](#)

Extensions [52](#) npm installs [98k/month](#) master branch [passing](#) unstable branch [passing](#) Greenkeeper [enabled](#)

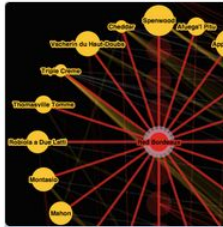
Demos



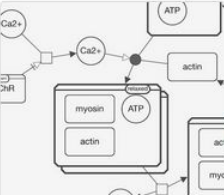
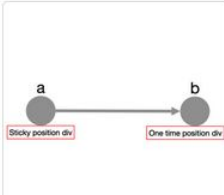
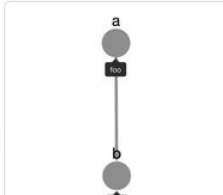
Cola.js gene-gene graph



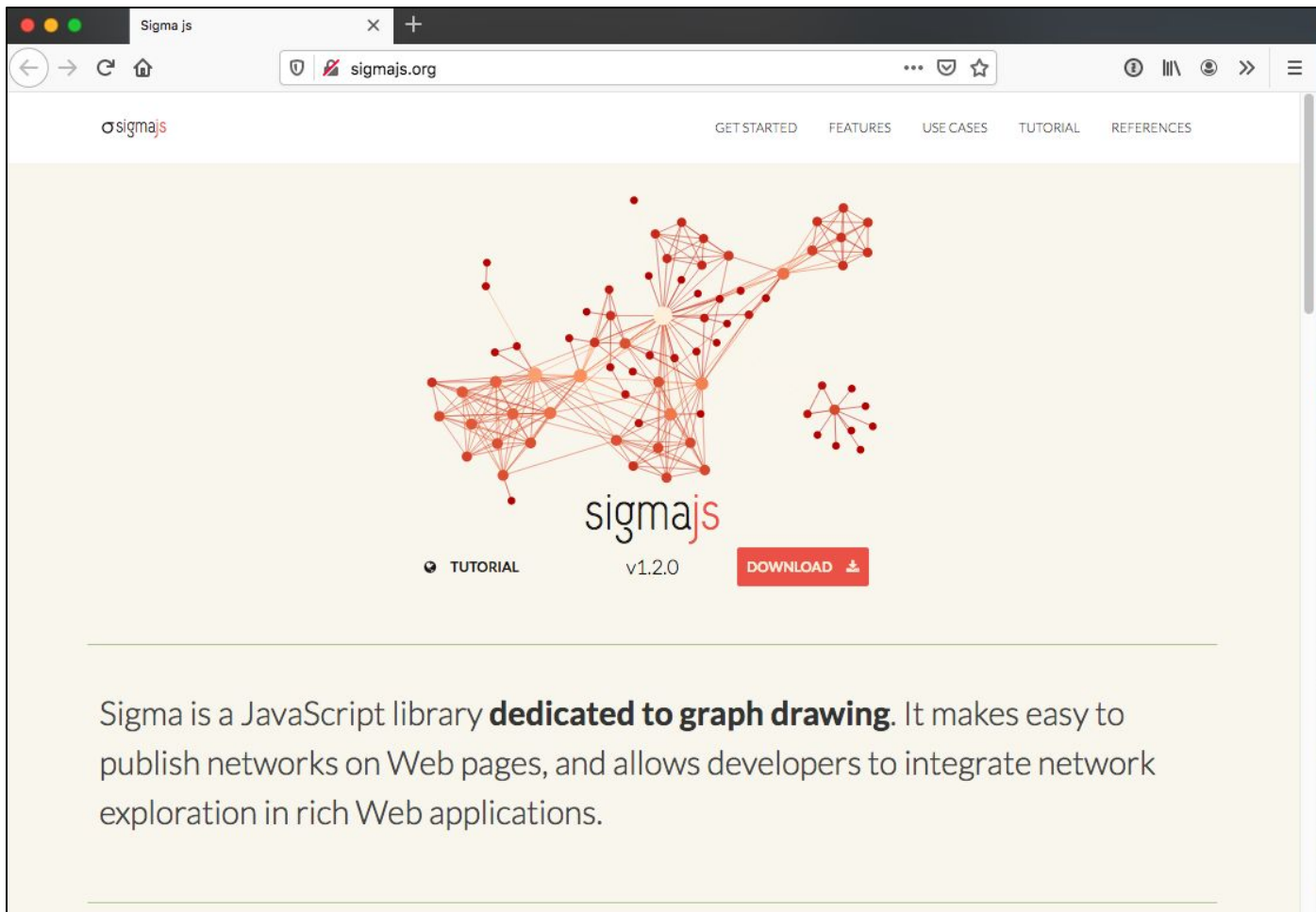
Tokyo railways



Wine & cheese

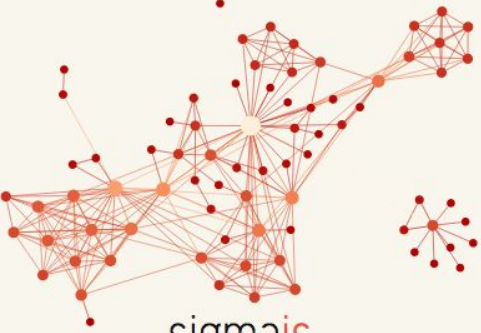




[Cytoscape.js](https://js.cytoscape.org)

A screenshot of a web browser displaying the Sigma.js website. The browser's address bar shows 'sigmajs.org'. The website has a navigation bar with links: 'GET STARTED', 'FEATURES', 'USE CASES', 'TUTORIAL', and 'REFERENCES'. The main content area features a large, complex network graph with red nodes and orange edges. Below the graph, the text 'sigmajs' is displayed in a large, stylized font. Underneath this, there is a 'TUTORIAL' link, the version number 'v1.2.0', and a red 'DOWNLOAD' button with a download icon. A horizontal line separates this section from the text below. The text describes Sigma as a JavaScript library dedicated to graph drawing, highlighting its ease of use for publishing networks and integrating network exploration into web applications.

Sigma.js

GET STARTED FEATURES USE CASES TUTORIAL REFERENCES

sigmajs

[TUTORIAL](#) v1.2.0 [DOWNLOAD](#)

Sigma is a JavaScript library **dedicated to graph drawing**. It makes easy to publish networks on Web pages, and allows developers to integrate network exploration in rich Web applications.

[Sigma.js](https://sigmajs.org)

Web Science:

Visualizing Social Networks

(Part 3 - D3 Node-Link Walkthrough)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Force-Directed Layouts in D3

- Centering - attracts every node to a specific position
`.force("center", d3.forceCenter())`
- Collision - consider nodes as circles with radius and try to avoid overlapping
`.force('collision', d3.forceCollide())`
- Links - pushes linked nodes together, according to a link distance
`.force("link", d3.forceLink())`
- Many-Body - apply general attraction (if positive) or repulsion (if negative) between nodes
`.force("charge", d3.forceManyBody())`
- Positioning - push each node towards a desired position
`.force("x", d3.forceX())`
`.force("y", d3.forceY())`

D3 Node-Link References

- [Intro to Forced Layouts](#)
- [Interactive & Dynamic Force-Directed Graphs with D3](#),
 - *uses d3.v4 so some calls may be different than d3.v5*

```
var dataset = {
  nodes: [
    { name: "Adam" },
    { name: "Bob" },
    { name: "Carrie" },
    { name: "Donovan" },
    { name: "Edward" },
    { name: "Felicity" },
    { name: "George" },
    { name: "Hannah" },
    { name: "Iris" },
    { name: "Jerry" }
  ],
  edges: [
    { source: 0, target: 1 },
    { source: 0, target: 2 },
    { source: 0, target: 3 },
    { source: 0, target: 4 },
    { source: 1, target: 5 },
    { source: 2, target: 5 },
    { source: 2, target: 5 },
    { source: 3, target: 4 },
    { source: 5, target: 8 },
    { source: 5, target: 9 },
    { source: 6, target: 7 },
    { source: 7, target: 8 },
    { source: 8, target: 9 }
  ]
};
```

Scott Murray's [Interactive Data Visualization for the Web](#), 2nd Ed., Ch. 13 Layouts

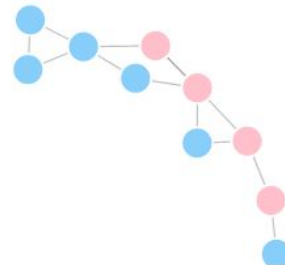
Force-Directed Layout Example - CS 432/532

Based on *Interactive Data Visualization for the Web, Ch 13*, Mike Bostock's *force-directed layout example* and *disjoint force-directed layout example*

Small dataset of friendships from *Interactive Data Visualization for the Web, Ch 13*

- `connected_data` - original dataset with gender labels added to the nodes
- `disconnected_data` - edge between Iris and Hannah removed to create disconnected graph

```
connected_data = ▶ Object {nodes: Array(10), links: Array(13)}
```



```
connected_chart = {
```

```
  const simulation = d3.forceSimulation(connected_data.nodes)
    .force("link", d3.forceLink(connected_data.links))
```


Further Reading on D3

- [D3 Intro](#)
- Mike Bostock's D3 examples
 - [at bl.ocks.org](#)
 - [at observablehq.com](#)
 - [d3 gallery](#)

Objectives

- List five network data formats.
- Describe a circular graph layout.
- Describe a force-directed graph layout, including the effect of the forces on the edges and nodes.
- Use a Python library to generate a node-link diagram of Zachary's Karate Club.