Web Science: Visualizing Social Networks

(Part 1 - Graph Drawing and Layouts)

CS 432/532

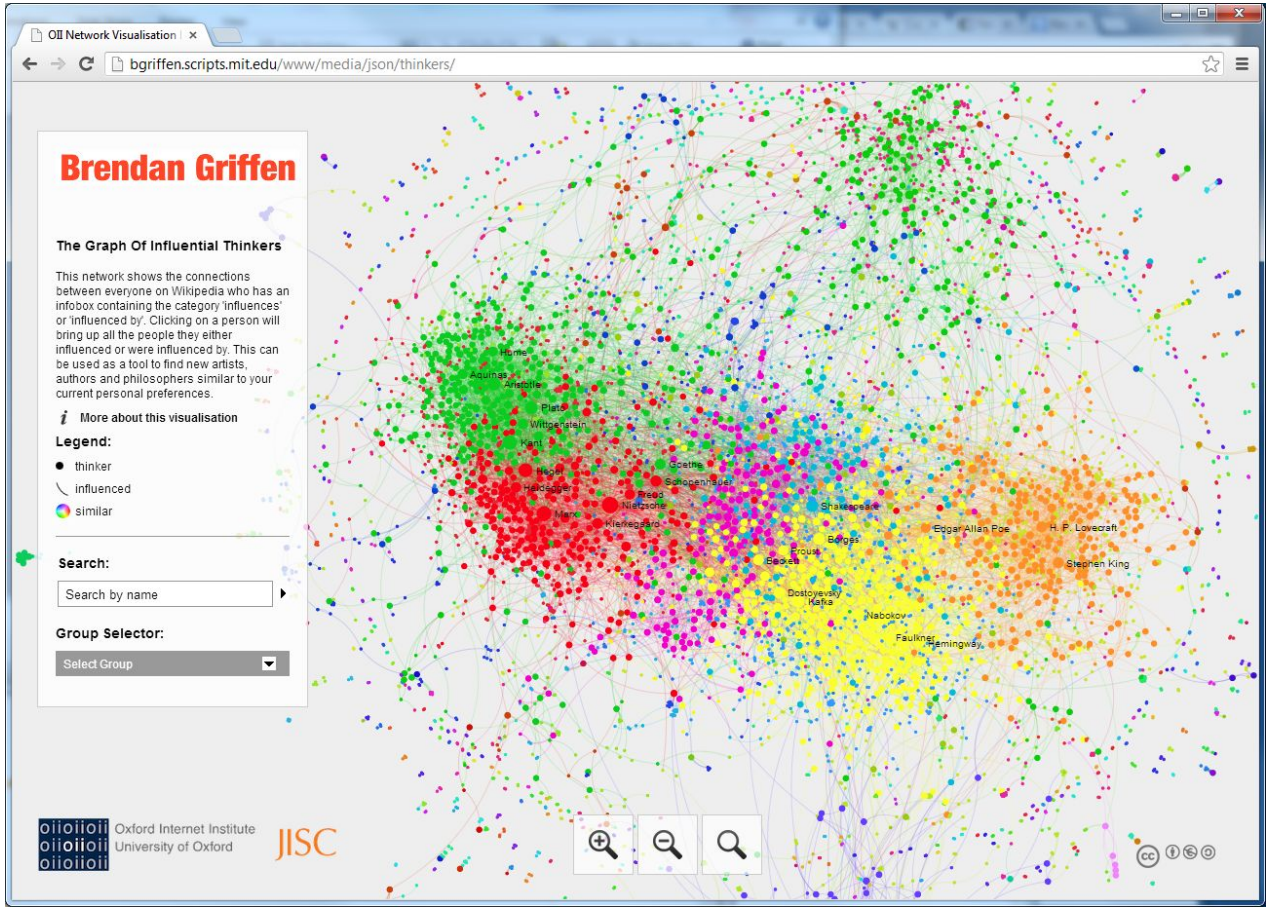
Old Dominion University

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

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

[Tall Infographics](#) (xkcd comic)

[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 Zizek, David Chalmers

Signature

[Graphs of Wikipedia: Influential Thinkers]

[René Descartes] (Wikipedia)

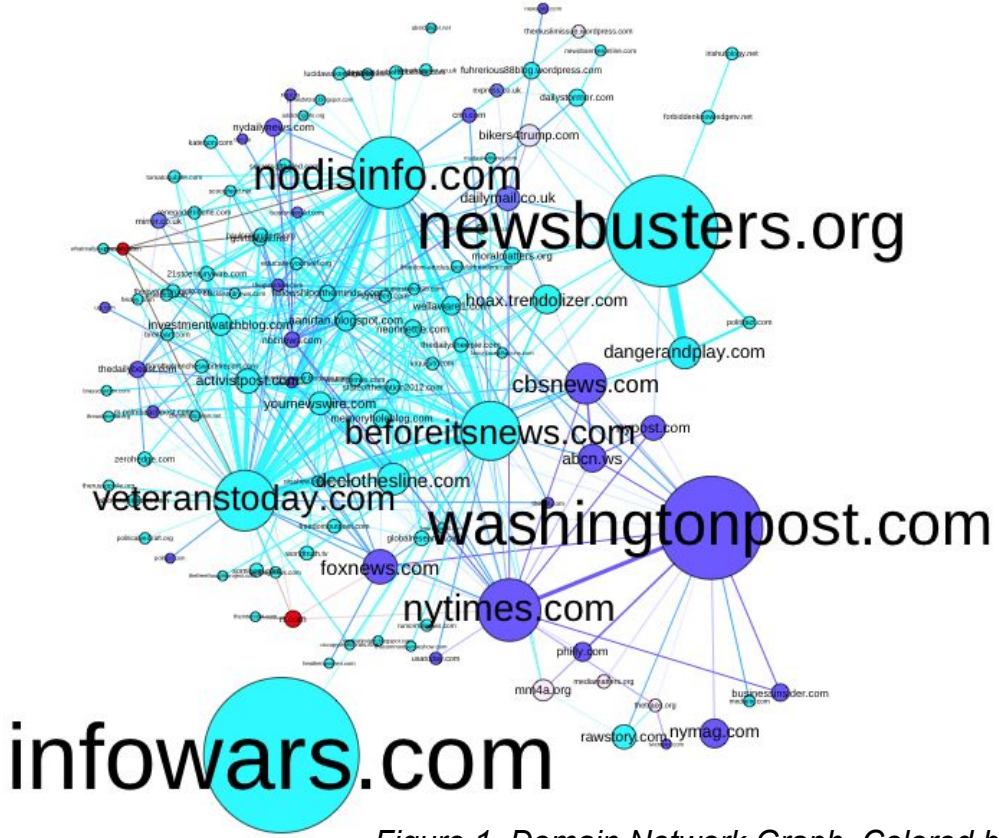


Figure 1. Domain Network Graph, Colored by Media Type
Purple = mainstream media; Aqua = alternative media; Red = government controlled media

Kate Starbird, [“Information Wars: A Window into the Alternative Media Ecosystem”]

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

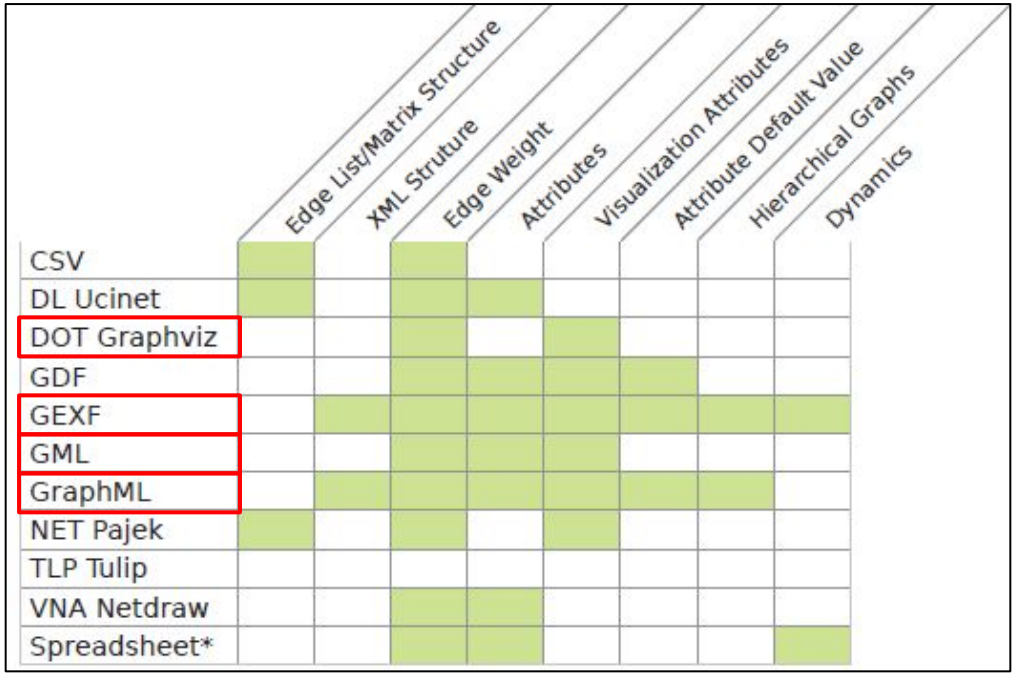
Node - domain

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 Structure	Edge Weight	Attributes	Visualization Attributes	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](#) (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" />
```

GEXF File Format

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.

```
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"
  ]
]
```

[GML Format]

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>
```

GraphML Format

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"];
}
```

GraphViz DOT Format

JSON

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

The Python library NetworkX can write out JSON.

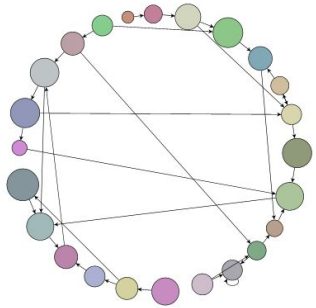
"Data preparation with R for networks in d3.js" has R code to convert from adjacency matrix, edge list, and edge and node lists to JSON

JSON — NetworkX 2.5 documentation

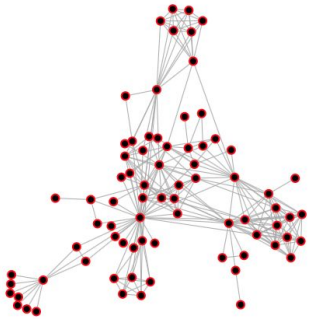
```
{ nodes: [
        { name: "Adam" },
        { name: "Bob" },
        { name: "Carrie" },
        { name: "Donovan" },
        { name: "Edward" },
        { name: "Felicity" },
        { name: "George" },
        { name: "Hannah" },
        { name: "Iris" },
        { name: "Jerry" }
    ],
    links: [
        { 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)

Force-directed graph drawing (Wikipedia)

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

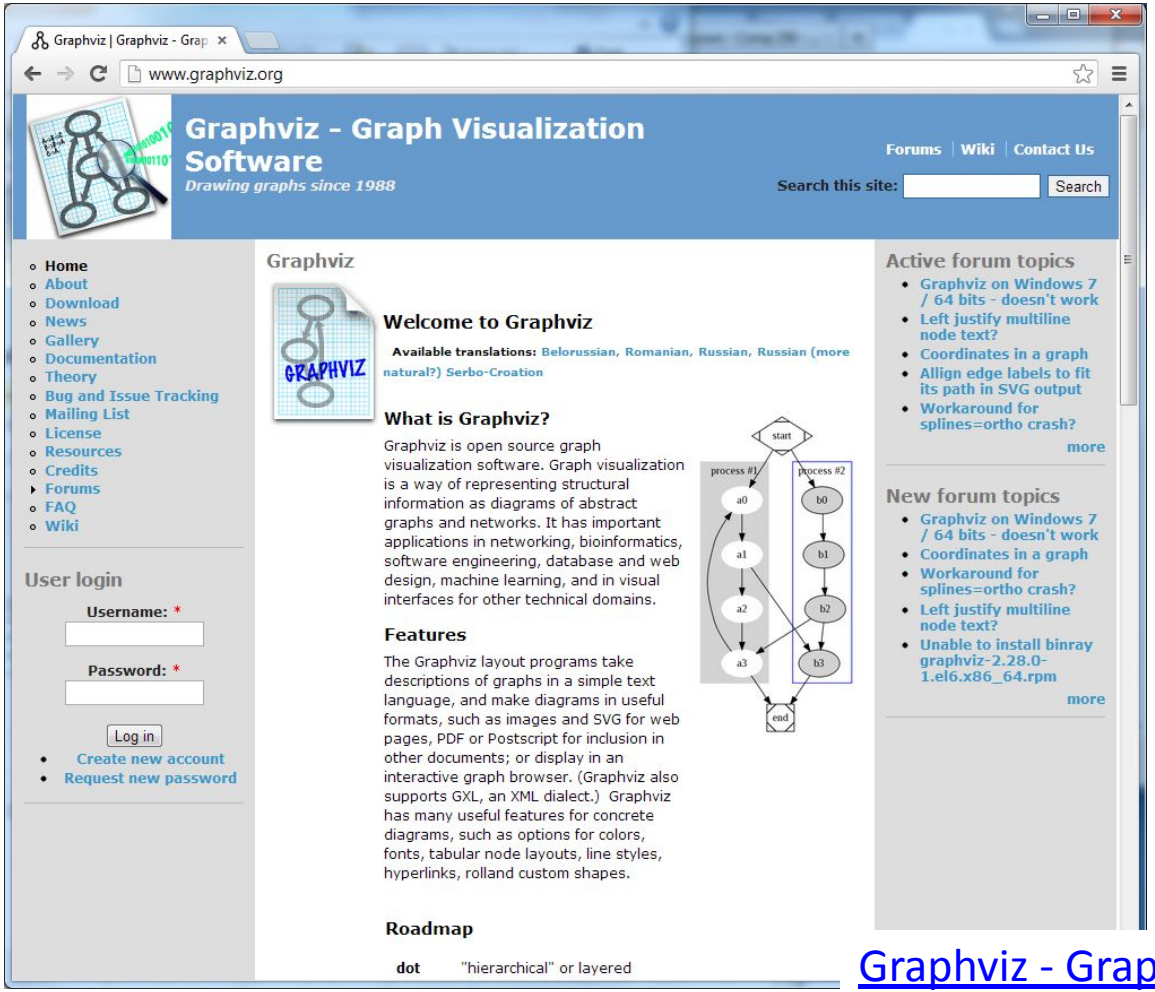
Graph Creation Software

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

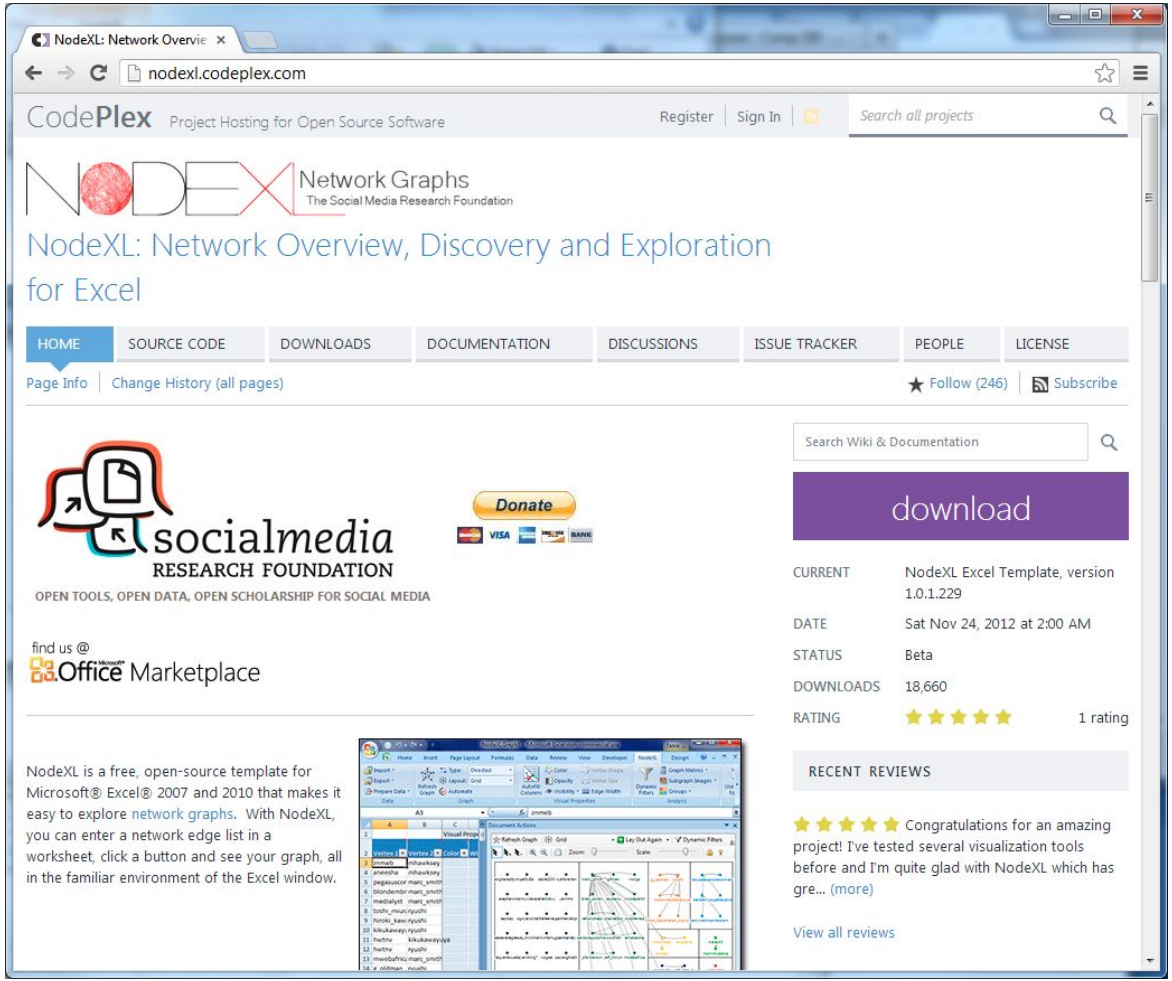
Stand-Alone Software

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

[Gephi - The Open Graph Viz Platform]



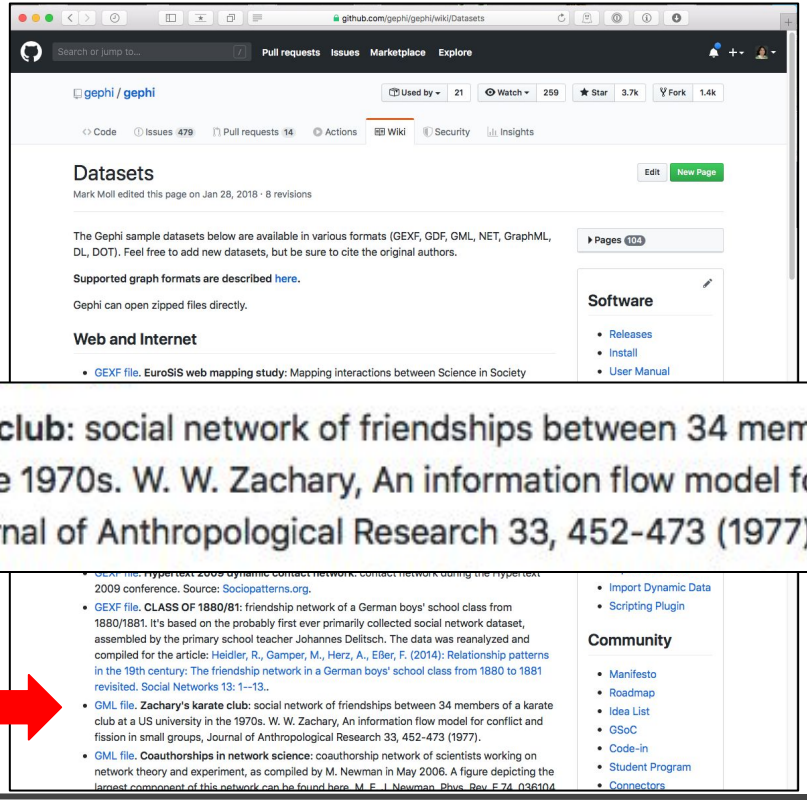
[Graphviz - Graph Visualization Software]


NodeXL: Network Overvie
nodexl.codeplex.com
CodePlex Project Hosting for Open Source Software
Register
Sign In
Search all projects
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
Search Wiki & Documentation
socialmedia RESEARCH FOUNDATION
OPEN TOOLS, OPEN DATA, OPEN SCHOLARSHIP FOR SOCIAL MEDIA
Donate
find us @ Office Marketplace
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 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.

[NodeXL]

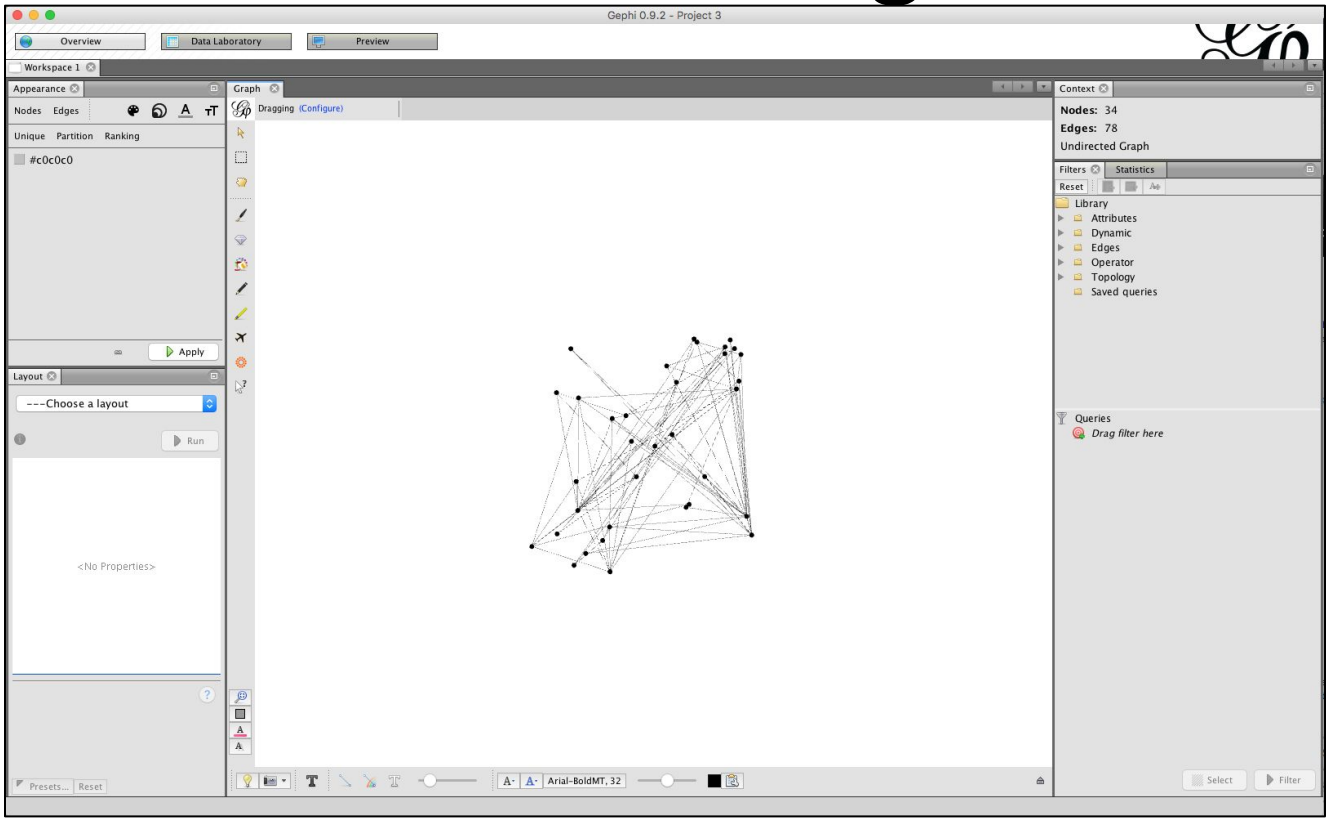
Gephi Example

Download data file from [Gephi Datasets]

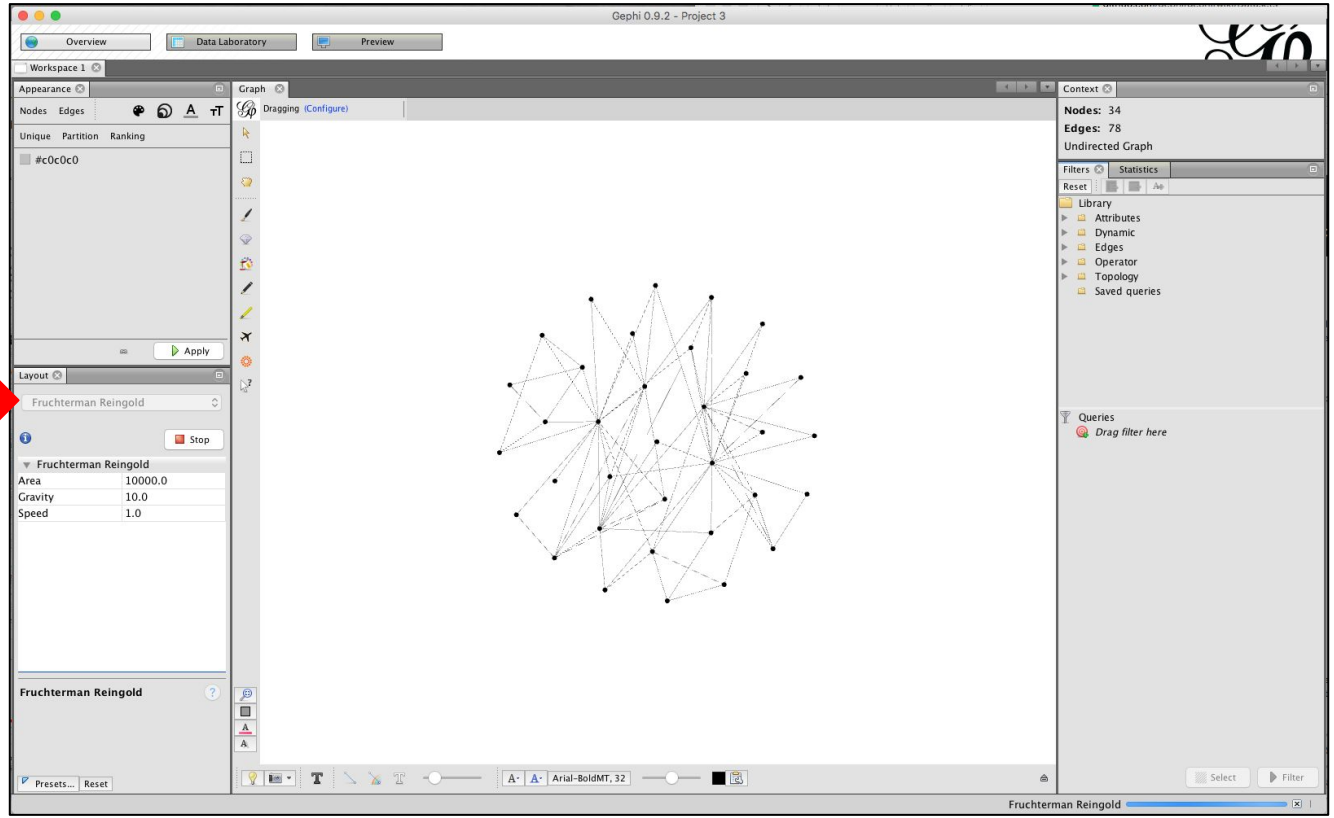


- GML file. **Zachary's karate club**: social network of friendships between 34 members of a karate club at a US university in the 1970s. W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977).

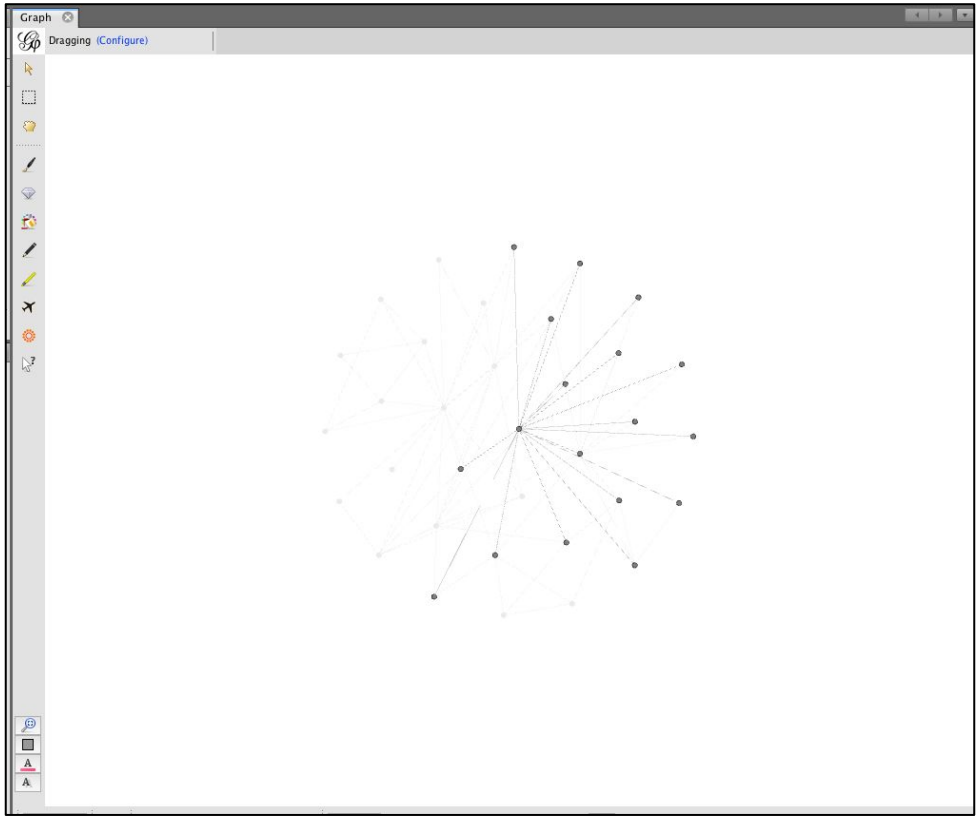
Load karate.gml



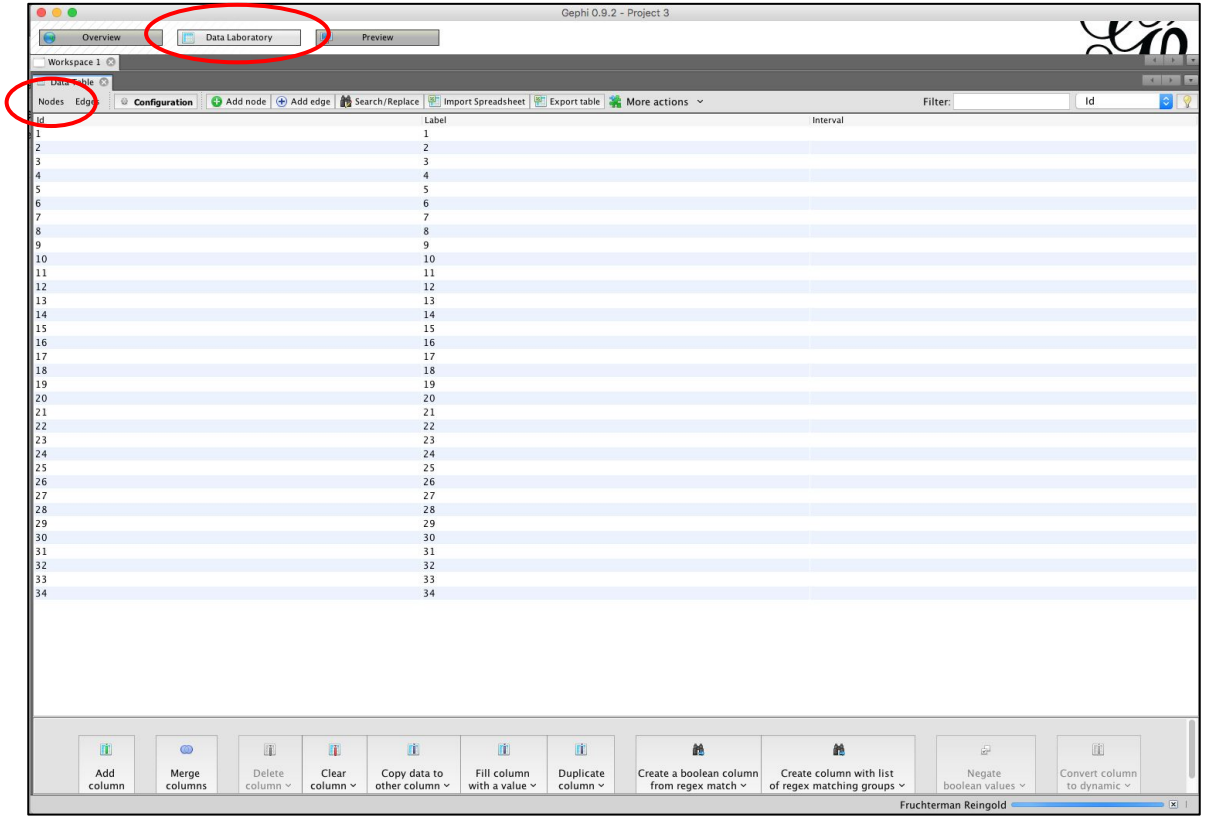
Set the layout



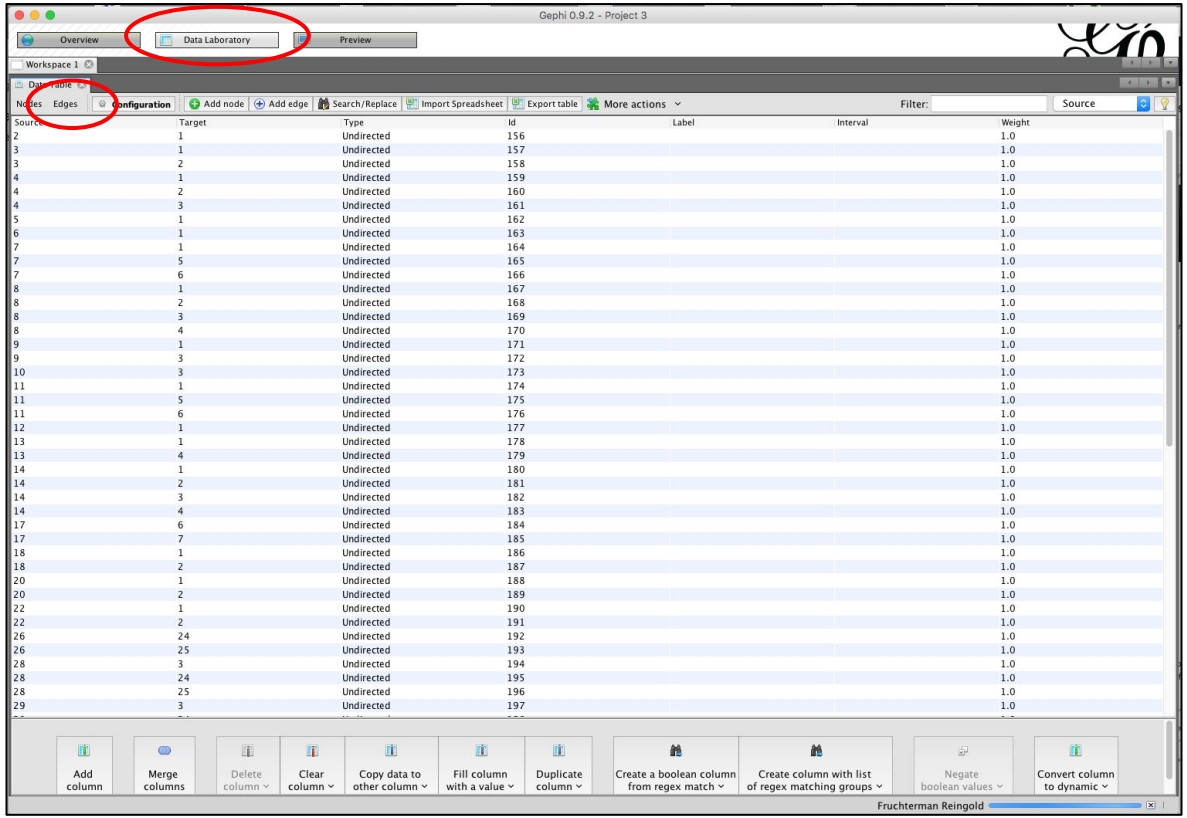
Hover to explore connections



Explore the data - nodes



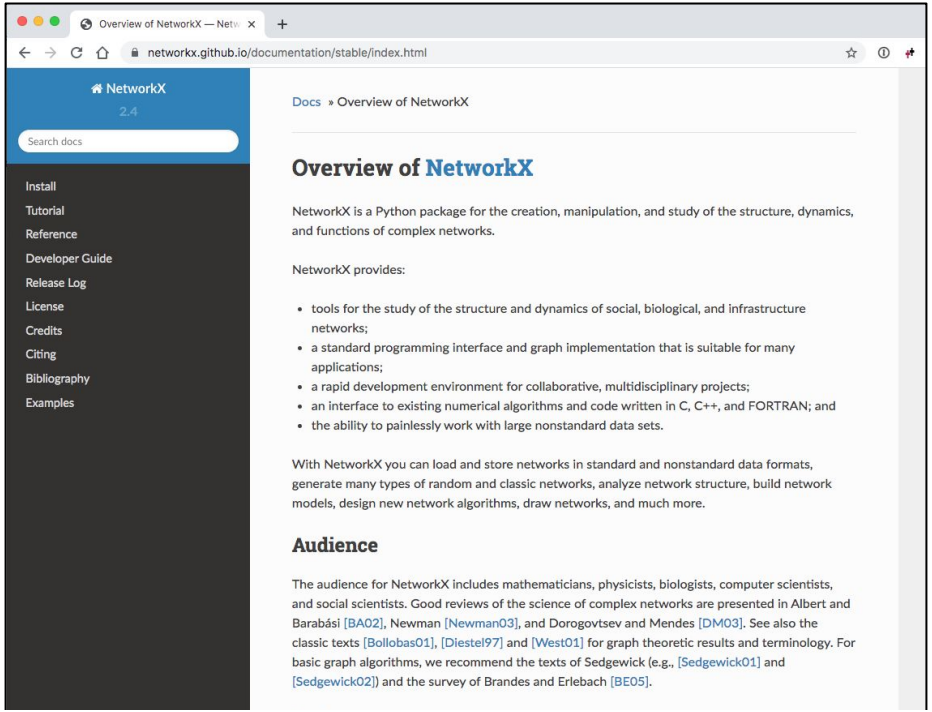
Explore the data - edges



Python Libraries

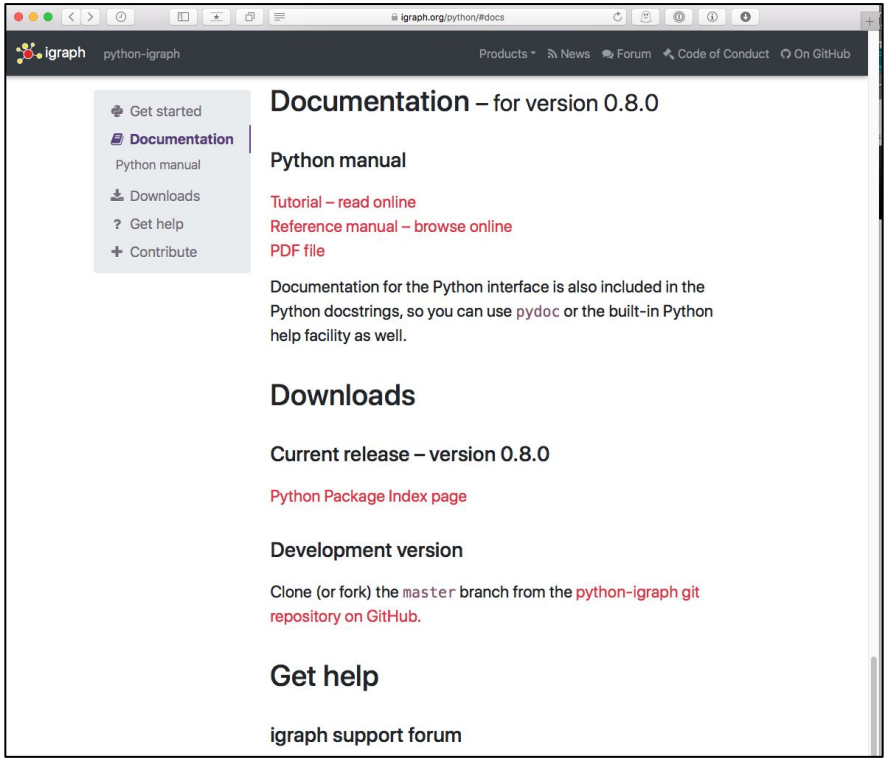
- NetworkX – creating and manipulating graphs
  - uses Matplotlib or Graphviz for displaying graphs
- igraph – creating, manipulating, and displaying graphs

NetworkX



Tutorials, Documentation at [Software for Complex Networks — NetworkX 2.5 documentation]
Additional examples at [NETWORK CHART]

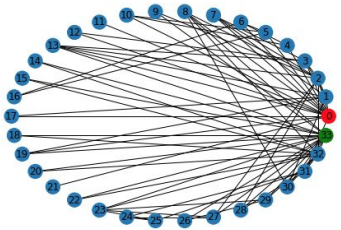
igraph-python



Tutorials, Documentation at [python-igraph]

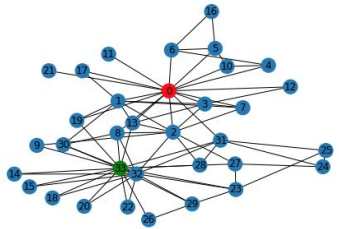
NetworkX Examples in Google Colab

```
node_color = ["#1f78b4"] * 34
node_color[0] = "red"
node_color[33] = "green"
node_color
nx.draw_circular(G, with_labels=True, node_color=node_color)
plt.show()
```



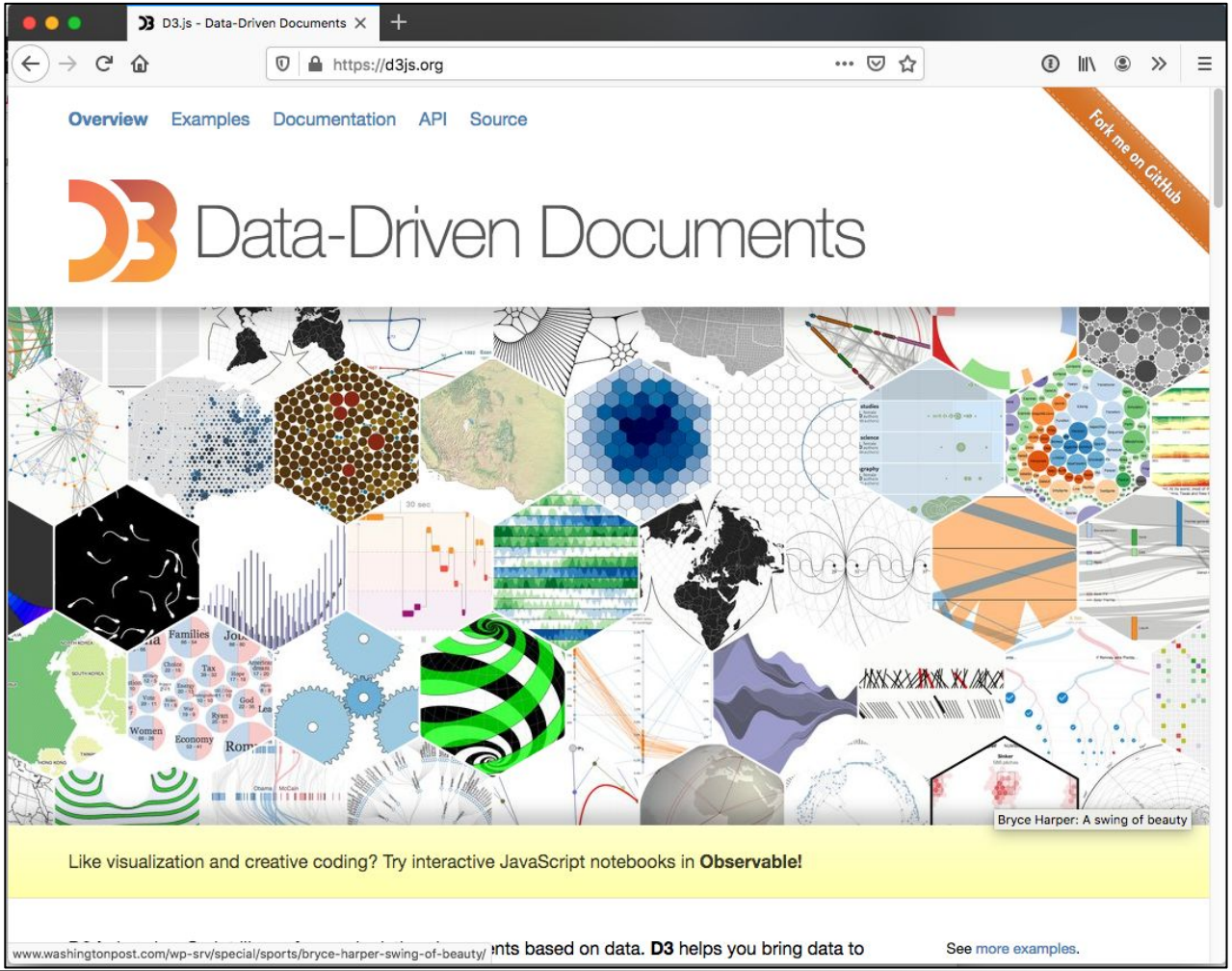
Draw with force-directed layout

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

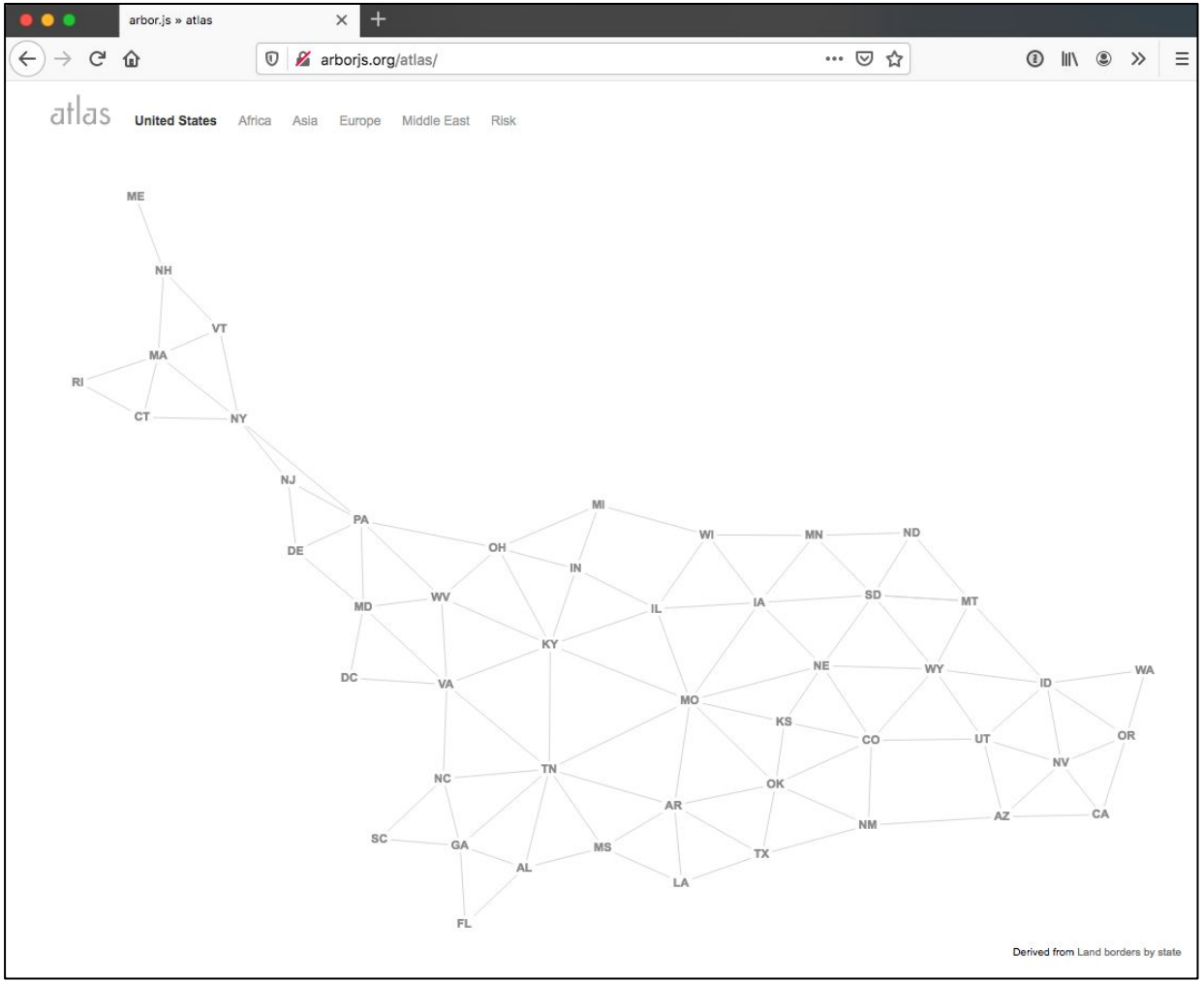


JavaScript Graphing Libraries

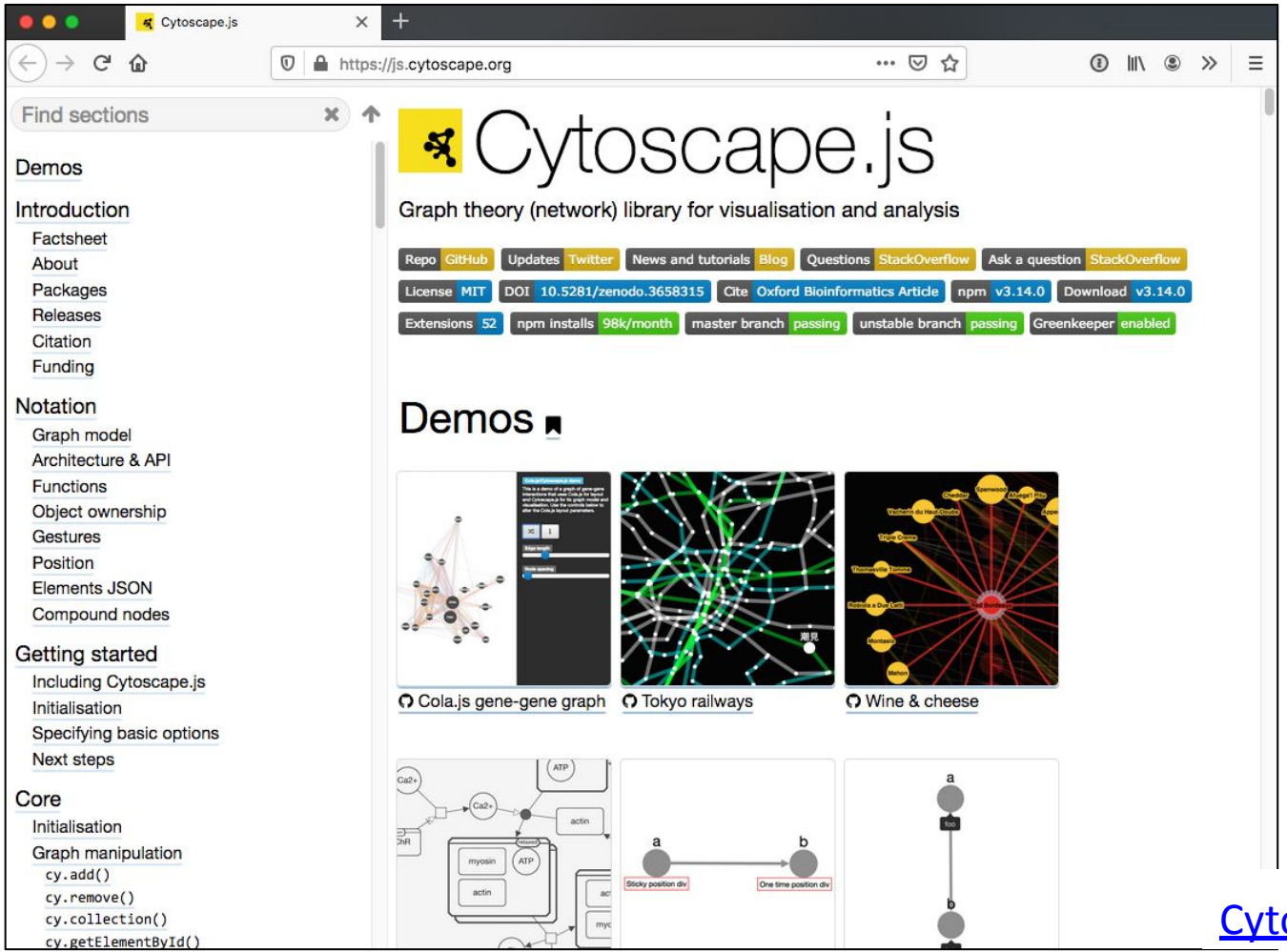
- D3.js
  - library for all types of visualization, also has higher-level APIs (Vega, Vega-Lite)
- Arbor.js
  - library for graph visualization using jQuery, uses HTML canvas, so it won't work in older browsers.
- Cytoscape.js
  - library for graph theory analysis and visualization.
- Sigma.js
  - lightweight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



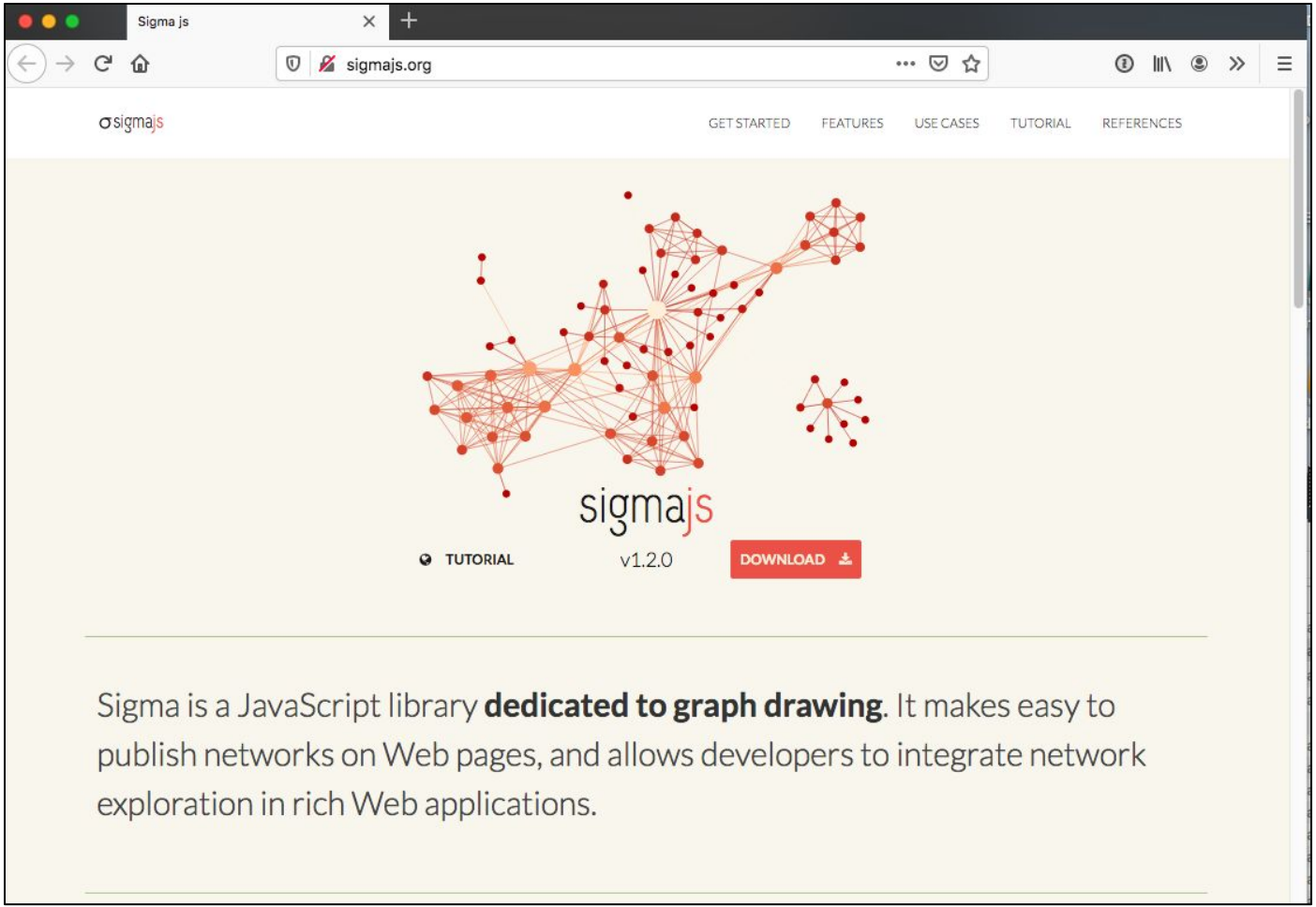
[D3.js - Data-Driven Documents]


arbor.js » atlas
arborjs.org/atlas/
atlas
United States
Africa
Asia
Europe
Middle East
Risk
ME
NH
VT
MA
RI
CT
NY
NJ
PA
DE
MD
WV
OH
MI
IN
KY
WI
IL
MN
IA
ND
SD
MT
NE
WY
ID
WA
DC
VA
MO
KS
CO
UT
OR
NV
CA
NC
TN
OK
AR
NM
AZ
SC
GA
AL
MS
LA
TX
FL
Derived from Land borders by state

[arbor.js]



[Cytoscape.js]


Sigma js
sigmajs.org
sigmajs
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]

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

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())
```

Network Graph | the D3 Graph Gallery
d3/d3-force: Force-directed graph layout.

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

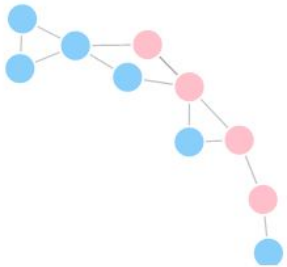
Force-Directed Layout Example - CS 432/532

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.