Frontiers of Network Science
Fall 2018

Class 8: Introduction to Gephi
Tools for network analysis and visualization

Boleslaw Szymanski

Main Topics

- Overview of tools for network analysis and visualization
- Installing and using Gephi
- Gephi hands-on labs

Tools for network analysis and visualization

- Computing model and interface
  - Desktop GUI applications
  - API/code libraries, Web services
  - Web GUI front-ends (cloud, distributed, HPC)
- Extensibility model
  - Only by the original developers
  - By other users/developers (add-ins, modules, additional packages, etc.)
- Source availability model
  - Open-source
  - Closed-source
- Business model
  - Free of charge
  - Commercial

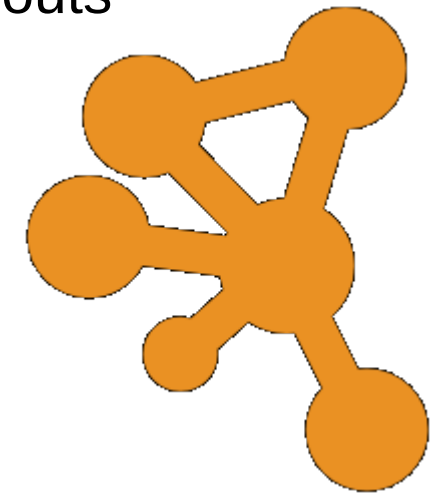
CyberInfrastructure for NETwork science

- Accessed via a Web-based portal (http://cinet.vbi.vt.edu/granite/granite.html)
- Supported by grants, no charge for end users
- Aims to provide researchers, analysts, and educators interested in Network Science with an easy-to-use cyber-environment that is accessible from their desktop and integrates into their daily work
- Users can contribute new networks, data, algorithms, hardware, and research results
- Primarily for research, teaching, and collaboration
- No programming experience is required

Network Dynamics & Simulation Science Laboratory

Network Data Integration, Analysis, and Visualization

- A standalone GUI application (http://www.cytoscape.org/)
- A platform for visualizing complex networks and integrating these with any type of attribute data
- Originally developed for biological research
- Includes features for data integration, analysis, and visualization
- A variety of layout algorithms, including cyclic, tree, force-directed, edge-weight, and yFiles Organic layouts
- Implemented in Java
- Runs on any Java-supported platform
- Modular architecture extensible through plugins (called Apps)
- Open-source and free of charge



The Open Graph Viz Platform

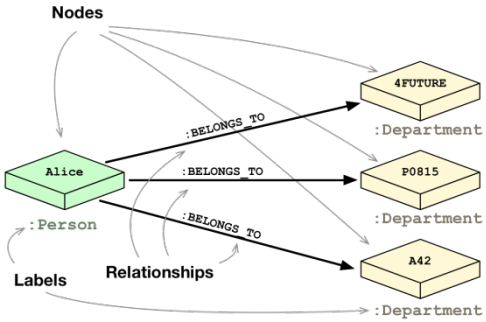
- A standalone GUI application
- An interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs
- Static and dynamic networks
- Clustering and hierarchical graphs, community detection
- Visualization layouts supported: ForceAtlas, Yifan's Hu Multilevel
- Modular architecture customizable with plugins
- Runs on Windows, Linux and Mac OS X
- Implemented in Java. Graph size <1M nodes & edges
- Open-source and free of charge

Gephi
makes graphs handy

Graph Visualization Software

- A graph description language (called DOT) and a set of tools that can generate and/or process DOT files
- Can be used as a standalone tool or as a library
- Only graph drawing
- A wide range of layouts:
  - Hierarchical or layered drawings
  - Spring model layouts
  - Multiscale layout for large graphs
  - Radial layouts
  - Circular layouts
- Implemented in C
- Runs on Linux, Windows and Mac OS X
- Extensible through a scripting API
- Open-source and free of charge

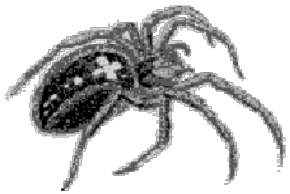
Neo4j Graph Database



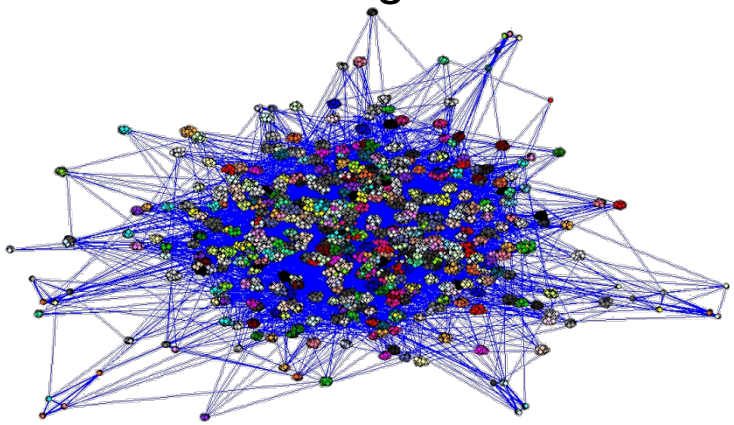
- First class support for nodes, relationships, and properties
- Efficient management of *semi-structured* and *network-oriented* data
- *Embedded* persistence engine – implemented as a small, light-weight, and non-intrusive Java library
- *Robust* – full support for distributed ACID transactions, configurable isolation levels, and transaction recovery
- *Highly scalable* – can handle large networks of data (no limits on the number of nodes, relationships, and properties that can be stored and indexed)
- *High-performance* – index-free adjacency, cost-based query optimizer, parallel indexes capability, binary protocol
- Open source, two editions – Community (GPL v3) and Enterprise (for commercial deployments with enterprise-grade availability, management, and scale-up and scale-out capabilities)

https://neo4j.com/developer/graph-db-vs-rdbms/

Pajek and Pajek-XXL



- A standalone GUI application
- Several partitioning and community detection algorithms
- Network generator (random, Bernoulli/Poisson, scale free, small world, etc.)
- Support for ordinary (directed, undirected, mixed) as well as multi-relational networks, bipartite, and temporal networks
- Capable of analyzing and visualizing large networks with thousands or even millions of nodes and edges (up to ten billion edges with Pajek64-3XL)
- Macro capability enables recording and playback of a sequence of primitive commands
- Implemented in Delphi (Pascal). Only Windows OS are supported (32 and 64 bit). Mac and Linux through wine.
- Freely available for noncommercial use

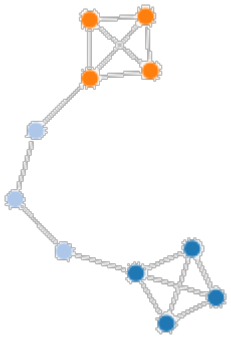


NetworkX – Software for Complex Networks

- Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks (https://networkx.github.io/)
- Supports standard and nonstandard data formats
- Standard graph algorithms
- Network analysis measures
- Visualization (with additional packages or tools, like Matplotlib, Graphviz, or NetworkxD3)
- Network generators
- Free software (open source BSD License)

```
>>> import networkx as nx
>>> G=nx.Graph()
>>> G.add_edges_from([(1,2),(1,3)])
>>> G.add_node(4)
>>> sorted(nx.degree(G).values())
[0, 1, 1, 2]

>>> G=nx.karate_club_graph()
>>> sum(nx.triangles(G).values())
135
```



http://snap.stanford.edu/

Stanford Network Analysis Platform (SNAP)

- A general purpose network analysis and graph mining library (http://snap.stanford.edu/index.html)
- Written in C++ but Python interface is also available
- Scales to massive networks with hundreds of millions of nodes, and billions of edges
- Efficiently manipulates large graphs, calculates structural properties, generates regular and random graphs, and supports attributes on nodes and edges
- Also available through the NodeXL which is a graphical front-end that integrates network analysis into Microsoft Office and Excel

http://snap.stanford.edu/

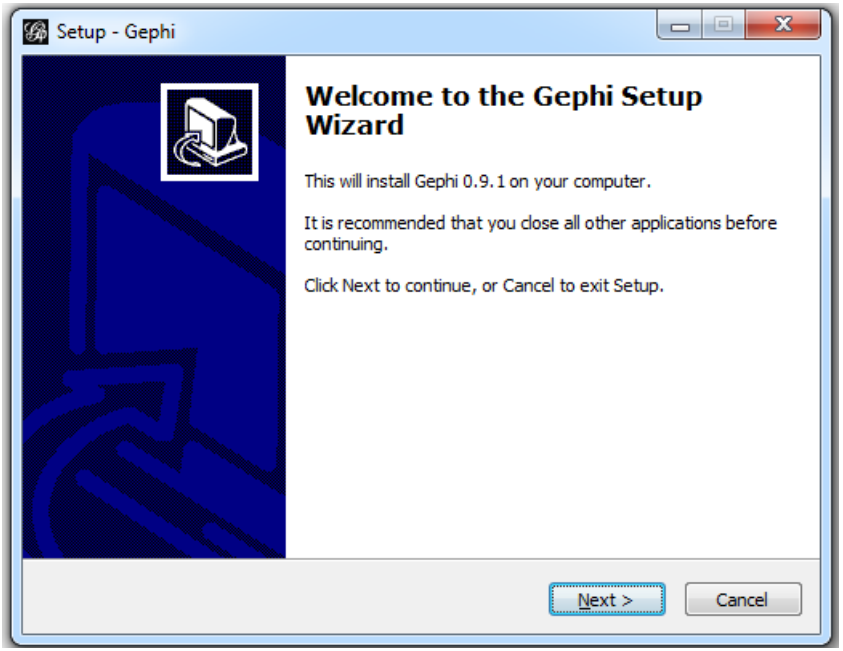
Main Topics

- Overview of tools for network analysis and visualization
- Installing and using Gephi
- Gephi hands-on labs

Installing Gephi

- Current stable version 0.9.2 but you might want to use an older version for compatibility with certain plugins
- Windows, Mac OS X, and Linux are supported
- Requires Java JRE 7 or 8
- Memory requirements:

Network size (nodes + edges)	~Memory suggested
~1000	128 MB
~10,000	512 MB
~100,000	2 GB
~1 M	>8 GB



- By default, Gephi is configured to start with 512 Mb of memory allocated to JVM. This might not be enough for larger graphs. To allocate more memory, increase the value of the **–Xmx** option (e.g., set **-Xmx1024m**) in *gephi.conf* configuration file. Then close and reopen Gephi for the new options to take effect.

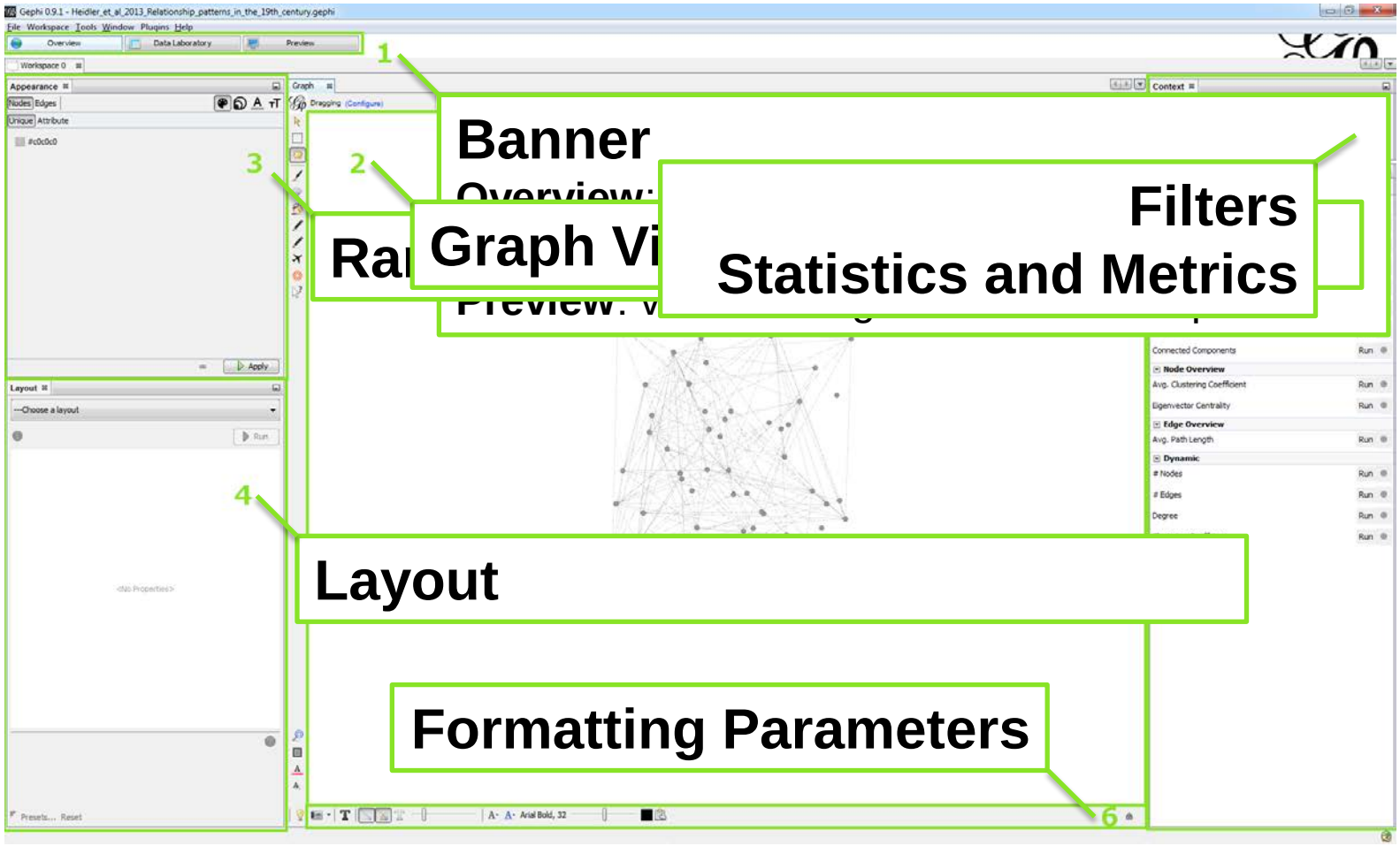
Sample Network Analysis in Gephi

Sample Gephi datasets are available at:

https://github.com/gephi/gephi/wiki/Datasets

Search for Class 1880-1881 and import it into Gephi

Gephi Visual Framework

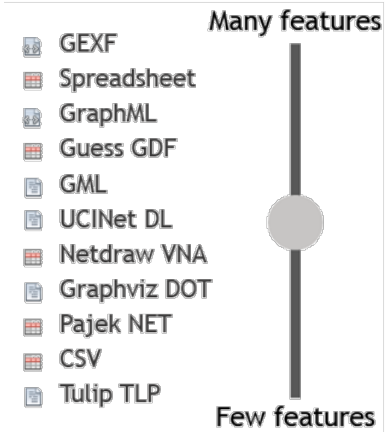


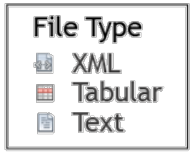
CLASS OF 1880/81: The dataset contains the friendship network of a German boys' school class from 1880/1881. It's based on the probably first ever primarily collected social network data set, assembled by the primary school teacher Johannes Delitsch. The data was reanalyzed and compiled for the article: 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. Available at: https://www.researchgate.net/profile/Richard_Heidler/publication/259515382_Relationship_patterns_in_the_19th_century_The_friendship_network_in_a_German_boys'_school_class_from_1880_to_1881_revisited/links/56a27e2208aeef24c585e7af.pdf

https://github.com/gephi/gephi/wiki/Datasets

Supported Graph Formats

- **GEXF**
- GDF
- GML
- GraphML
- Pajek NET
- GraphViz DOT
- CSV
- Edge List
- UCINET DL
- Tulip TLP
- Netdraw VNA
- TGF
- Spreadsheet
- Compressed ZIP





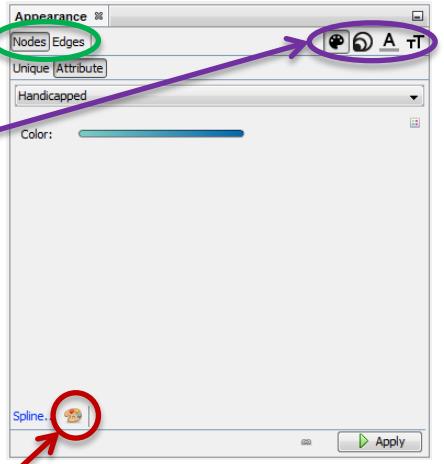
	Edge List/Matrix Structure	XML Struture	Edge Weight	Attributes	Visualization Attributes	Attribute Default Value	Hierarchical Graphs	Dynamics
CSV	■		■					
DL Ucinet	■		■	■				
DOT Graphviz			■		■			
GDF			■	■	■	■		
GEXF		■	■	■	■	■	■	■
GML			■	■	■			
GraphML		■	■	■	■	■	■	
NET Pajek	■		■		■			
TLP Tulip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

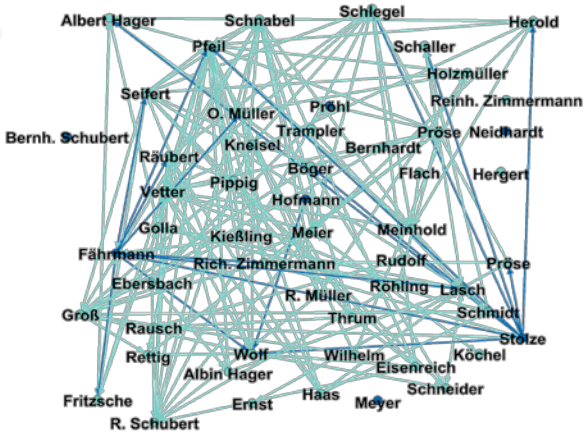
*** Spreadsheet:** Node tables and edge tables can be loaded in the Data Laboratory only.

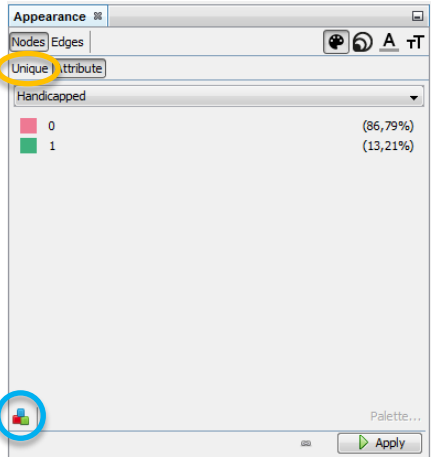
https://gephi.org/users/supported-graph-formats/

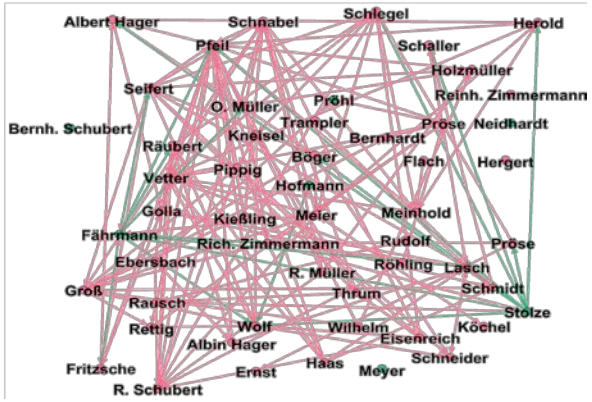
Changing the Size and Color of Nodes and Edges

- Ranking – set numerical attributes to distribute node and edge colors, sizes, label colors, and label sizes
- Partition – set node and edge grouping on attribute or computed metric. A color is assigned to each group.
- Use the Unique button to color / resize all nodes and edges uniformly (same color / size)
- Use Switch to partition and Switch to ranking to toggle between the ranking mode and the partitioning mode



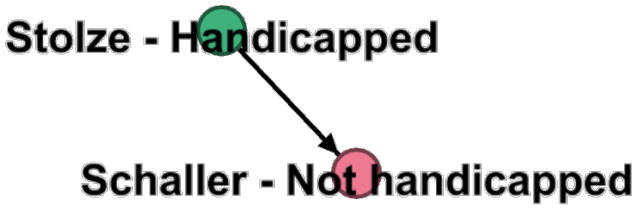




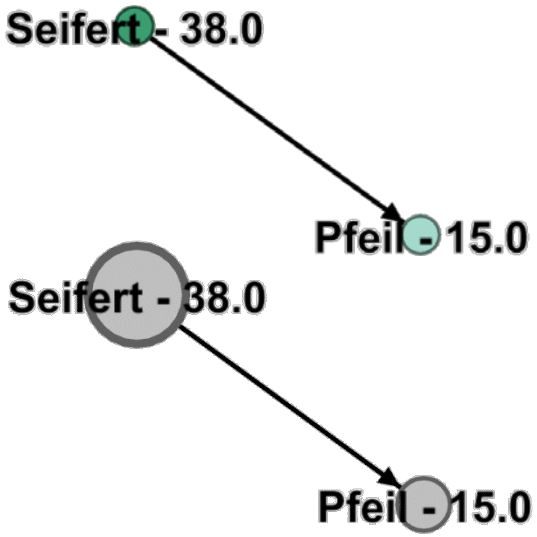


Partitioning or Ranking?

- Partitioning works for attributes that classify nodes **in categories**: “male or female”, “East, West, South or North”, “country of residence”, etc. Each category is represented by a different color.

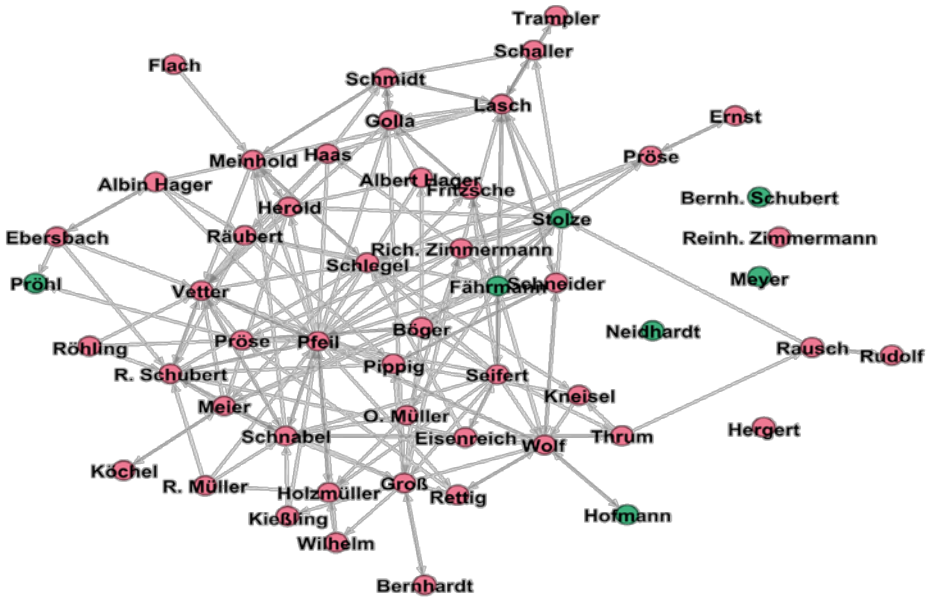


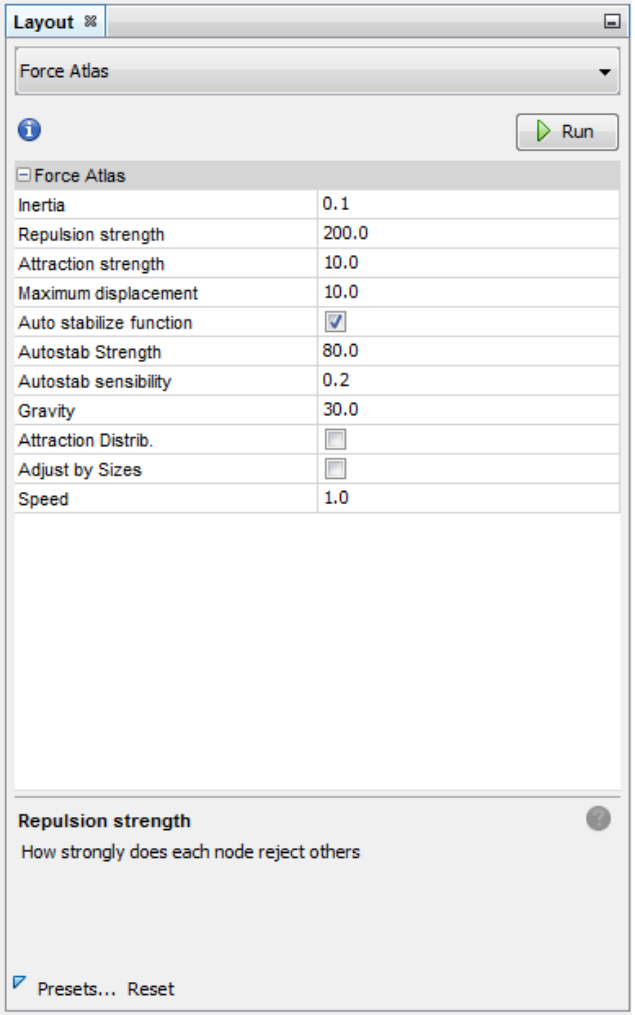
- Ranking works for attributes that are **gradual, not categorical** (age, height, grade, i.e., any numerical attribute). The graduation is represented visually either by bigger sizes, or by changes in shades of color (from light to dark, or from light colors to warm colors).



Automatically Repositioning the Nodes

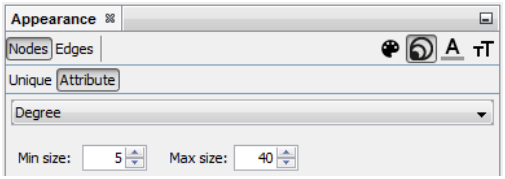
- Select an algorithm and set parameters
- Can also save parameters into presets
- Force Atlas layout tries to ensure as few edges cross as possible which usually works well for social networks
- Additional layouts are available as plugins

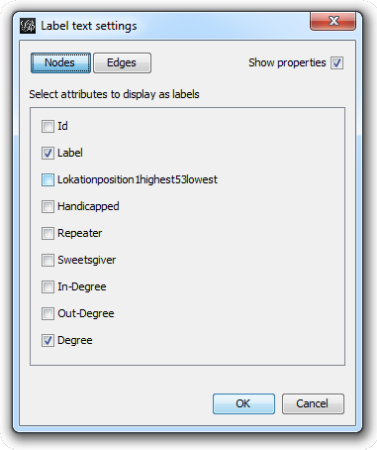


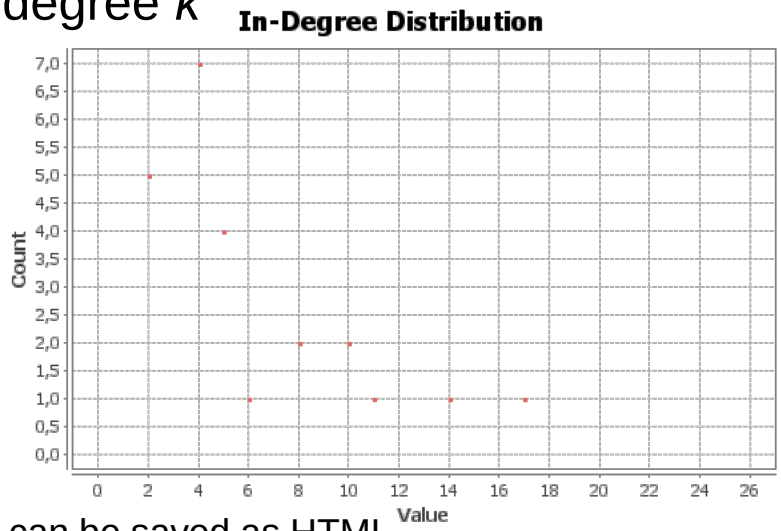


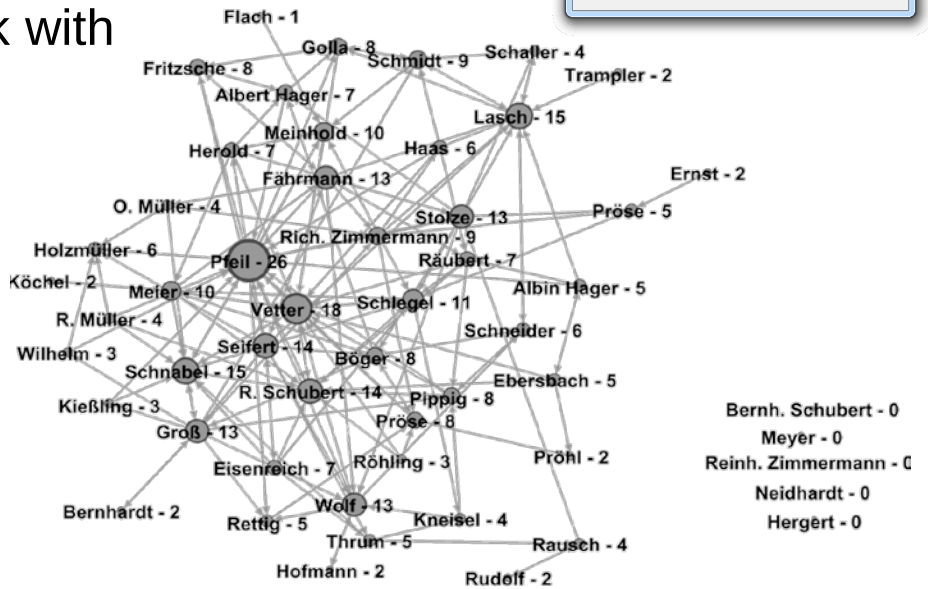
Computing Graph Metrics I

- Degree – the number of edges incident to a node (in-degree and out-degree for directed graphs). Nodes which have lots of edges (high degree) are called *hubs*.
- Degree distribution $P(k)$ of a network is the fraction of nodes in the network with degree k





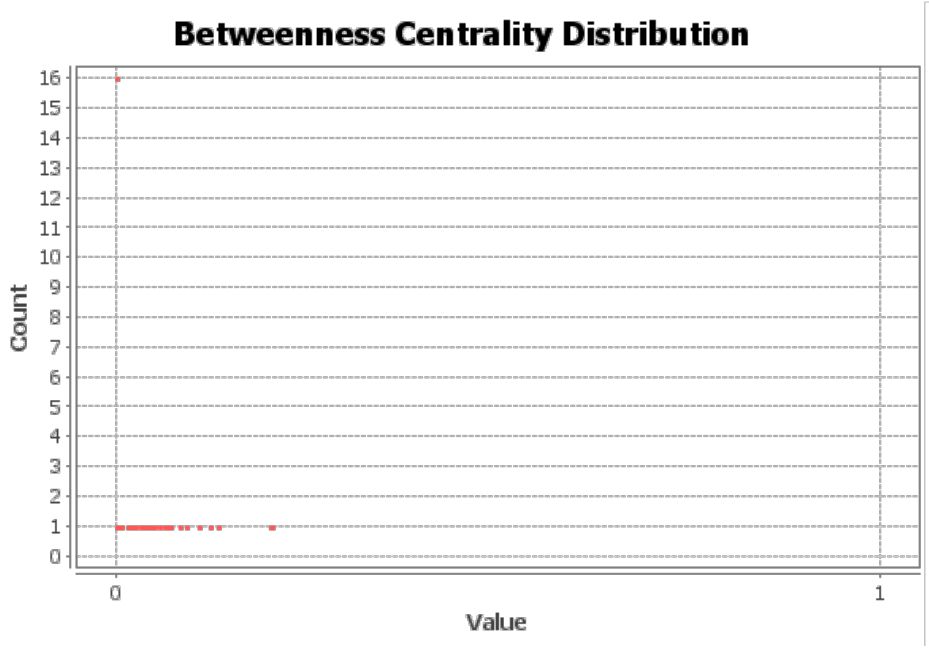


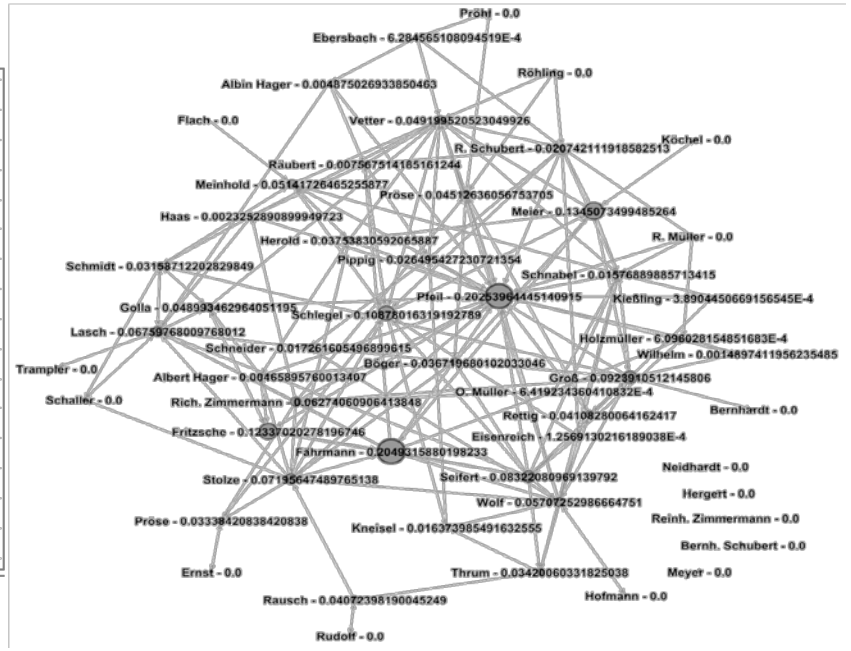


Reports can be saved as HTML

Computing Graph Metrics II

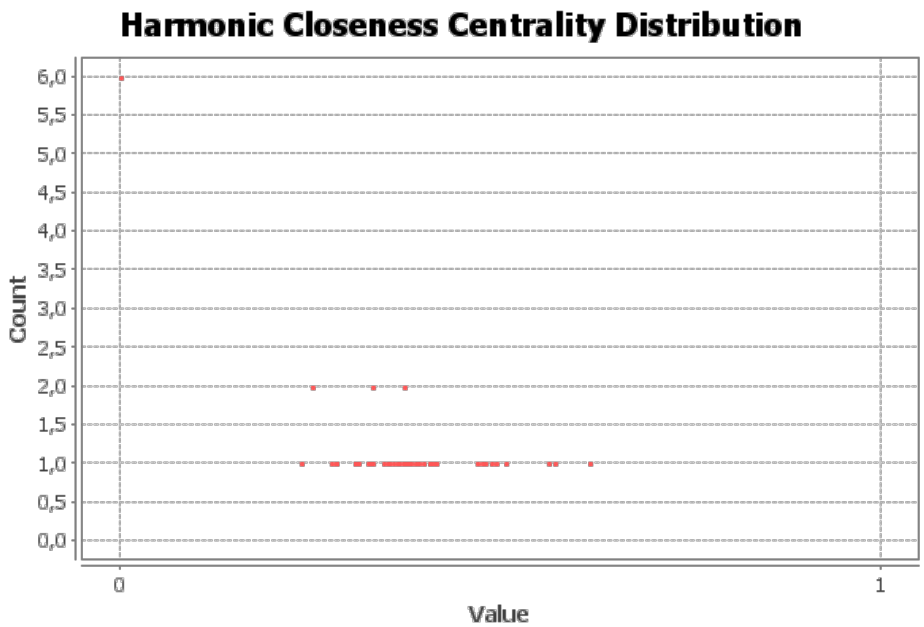
- Betweenness centrality – the number of shortest paths from all nodes to all others that pass through that node; measures the frequency of occurrence of a node on the shortest paths between network nodes

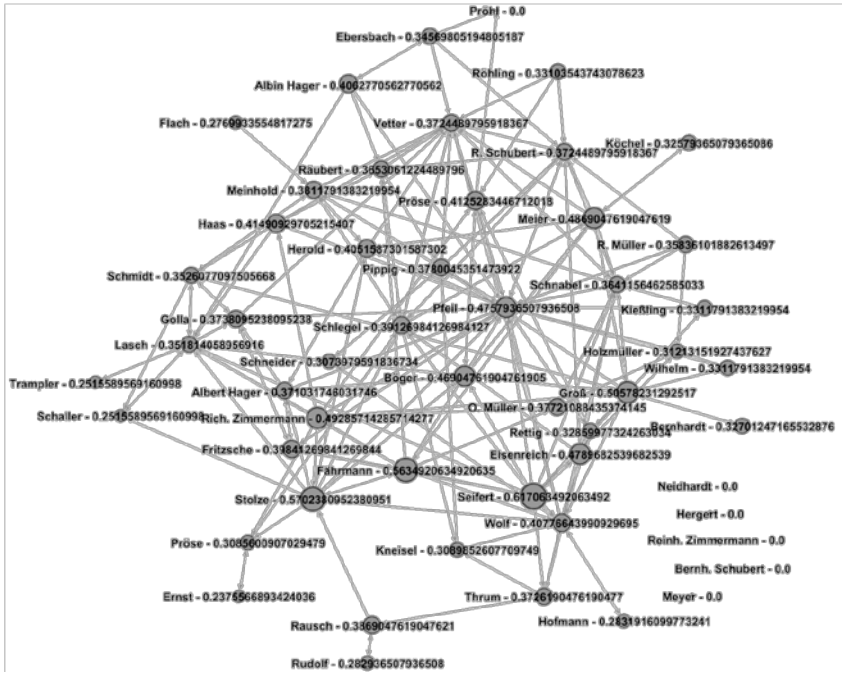




Computing Graph Metrics III

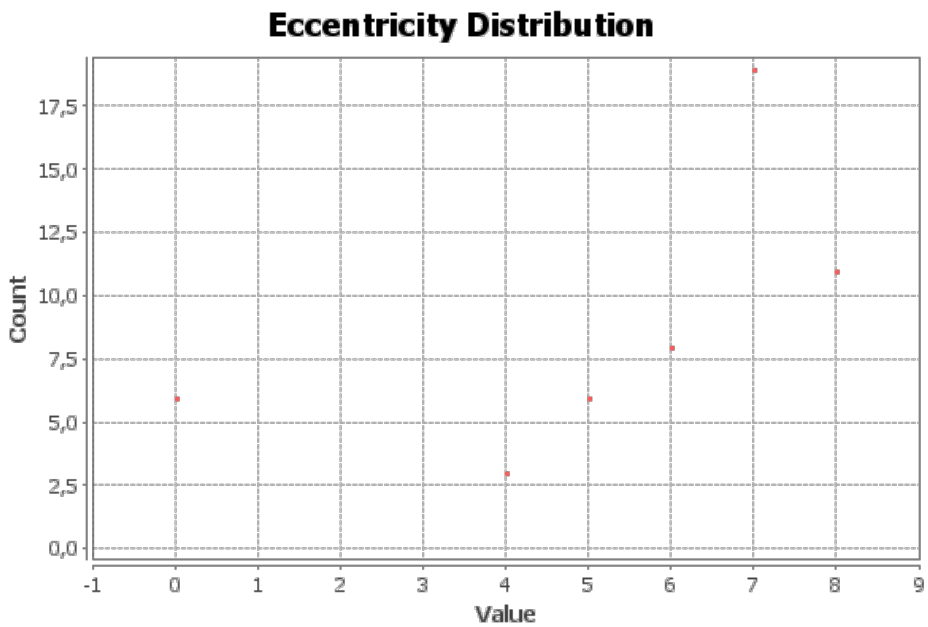
- Closeness centrality – a reciprocal of the sum of the shortest paths from a node to all others; measures the average distance between a node and all other nodes (the more central a node is the lower its total distance from all other nodes)

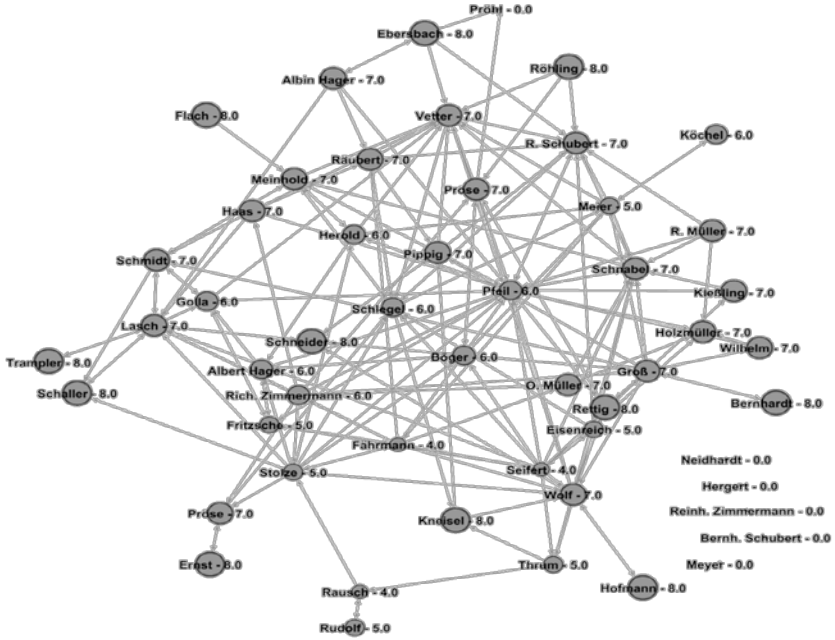




Computing Graph Metrics IV

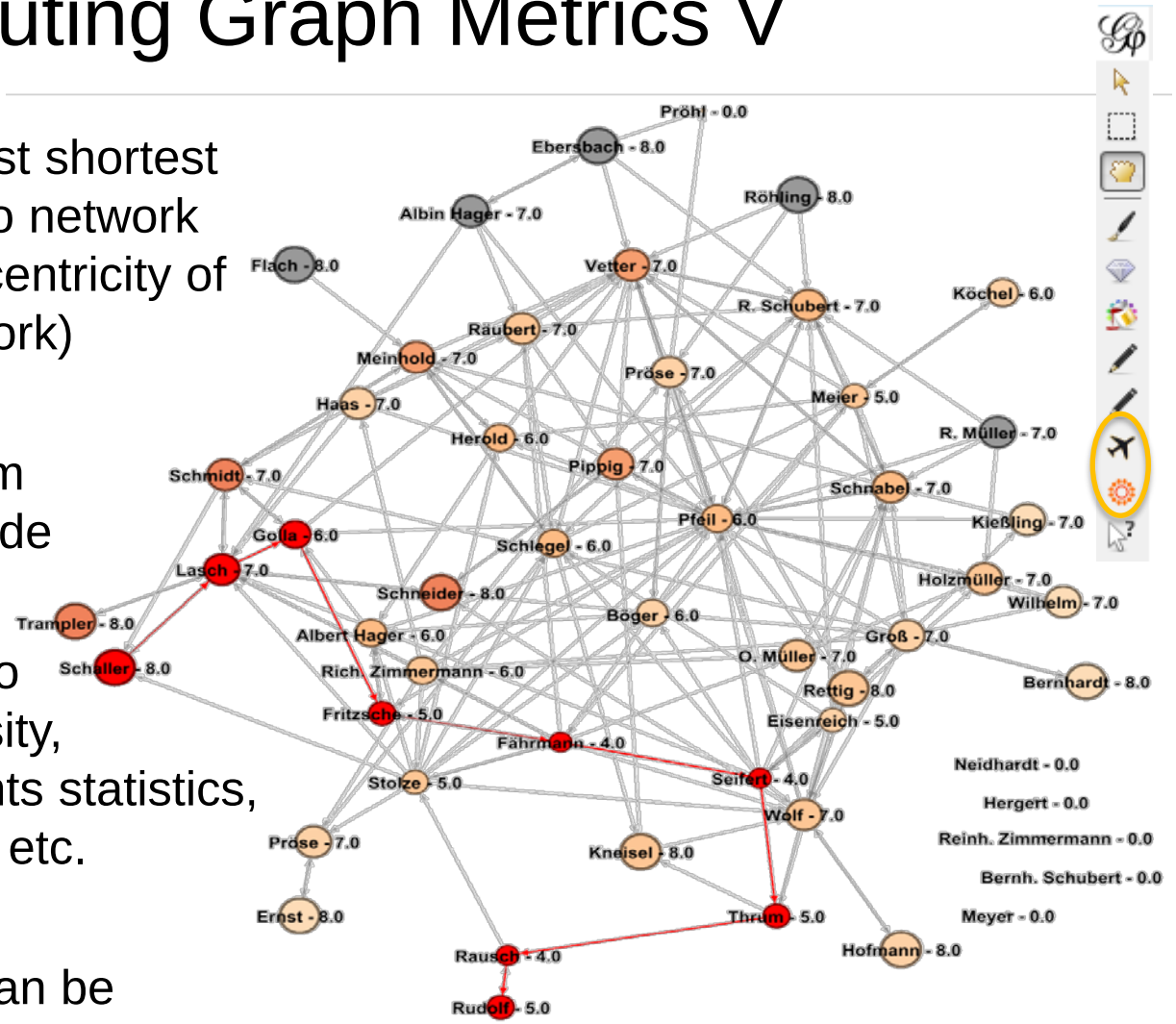
- Eccentricity – the greatest geodesic distance between a node and any other node in the network; measures how far a node is from the node most distant from it in the network





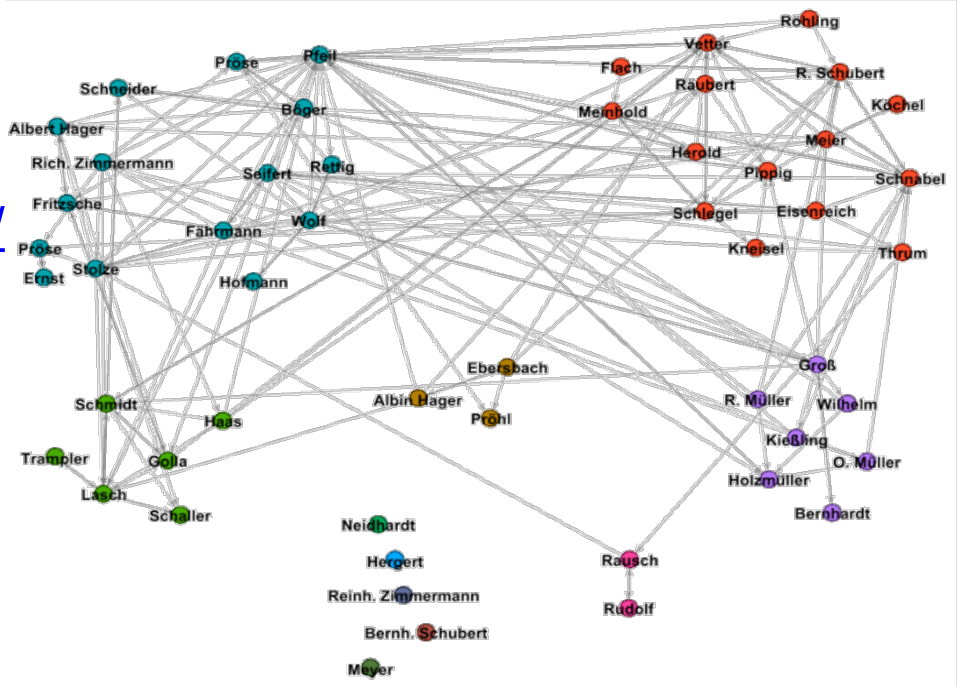
Computing Graph Metrics V

- Diameter – the longest shortest path between any two network nodes (maximum eccentricity of any node in the network)
- Radius – the minimum eccentricity of any node
- Other metrics are also available: graph density, connected components statistics, clustering coefficient, etc.
- Additional statistics can be loaded as plugins



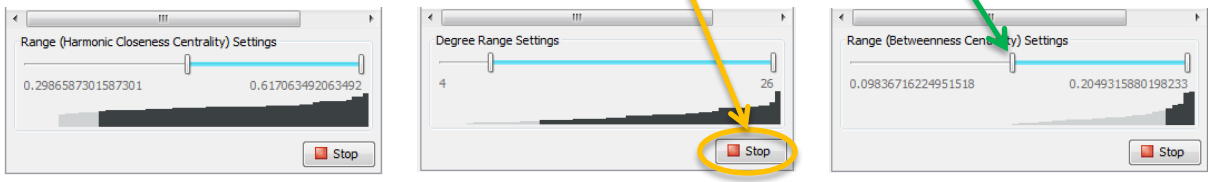
Network Clustering

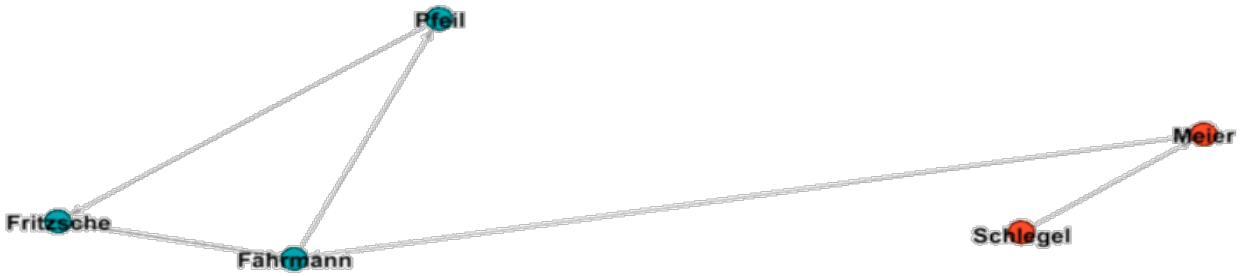
- Community – a group of nodes in a network that are more densely connected internally than with the rest of the network
- Networks may or may not display community structure (e.g., random graphs and Barabási–Albert model do not display community structure)
- Gephi implements the Louvain modularity optimization algorithm
- Additional community detection methods are available as plugins at https://marketplace.gephi.org/plugin_categories/plugin-clusters/
  - Label Propagation Clustering
  - Girvan Newman Clustering
  - Markov Cluster Algorithm

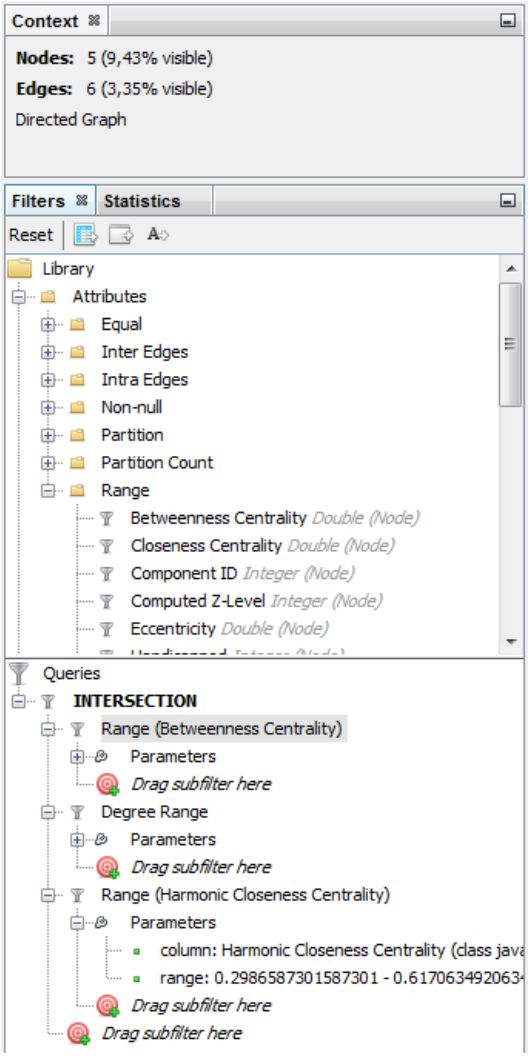


Filtering Nodes and Edges I

- A library of filters is available
- Filters can be combined to form a complex query
- Filter parameters can be set visually
- Filtering can be turned on and off
- Nodes can be filtered by attribute value or based on node statistics:

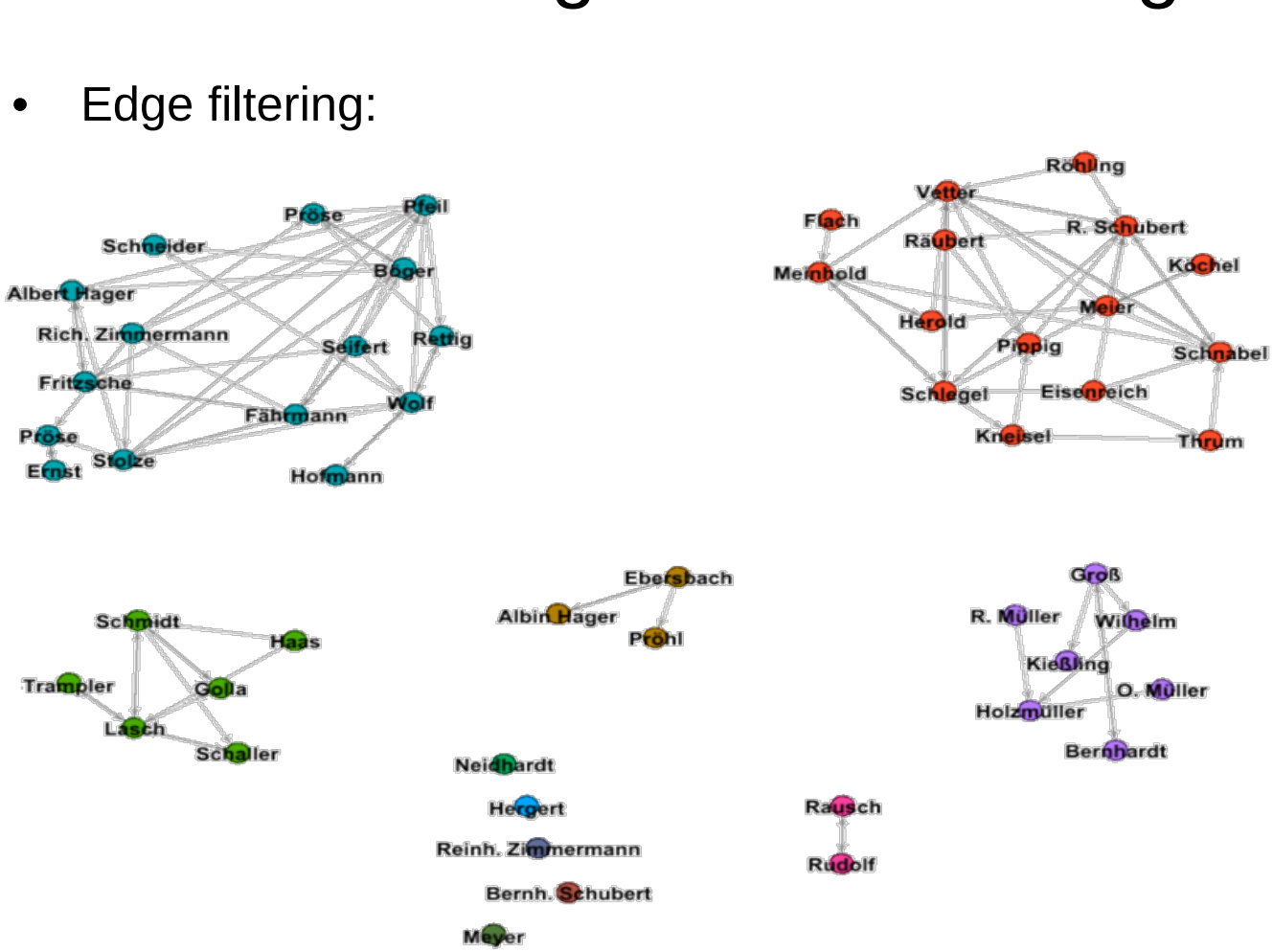


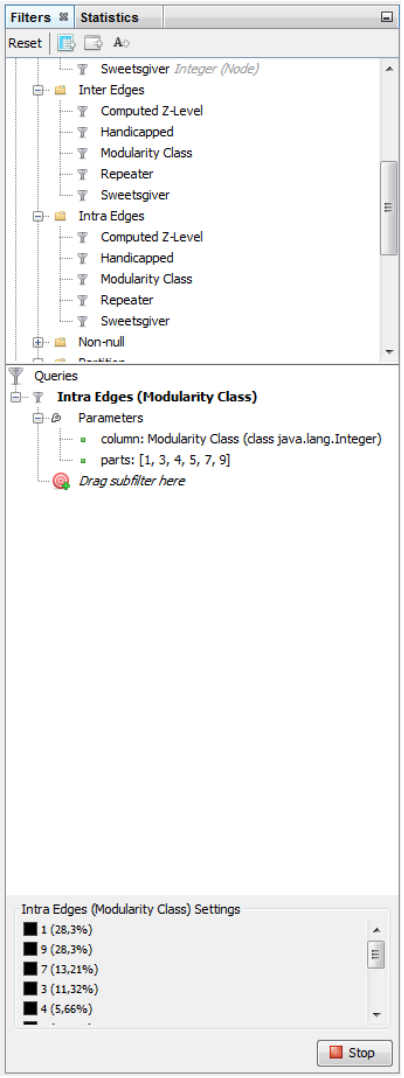




Filtering Nodes and Edges II

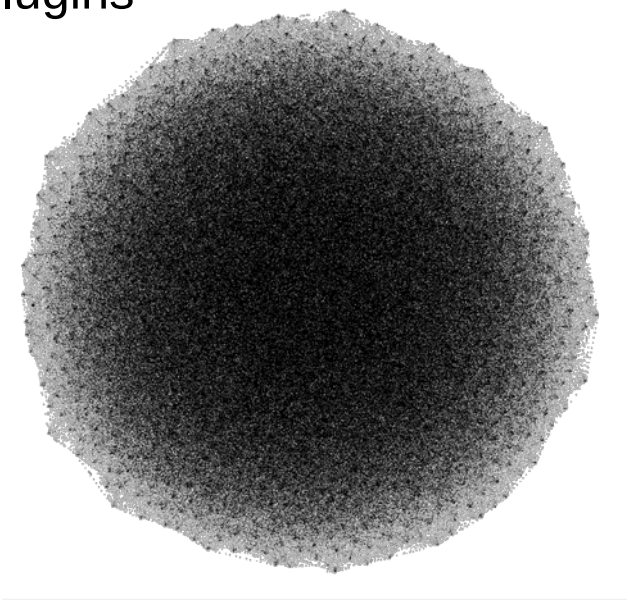
- Edge filtering:

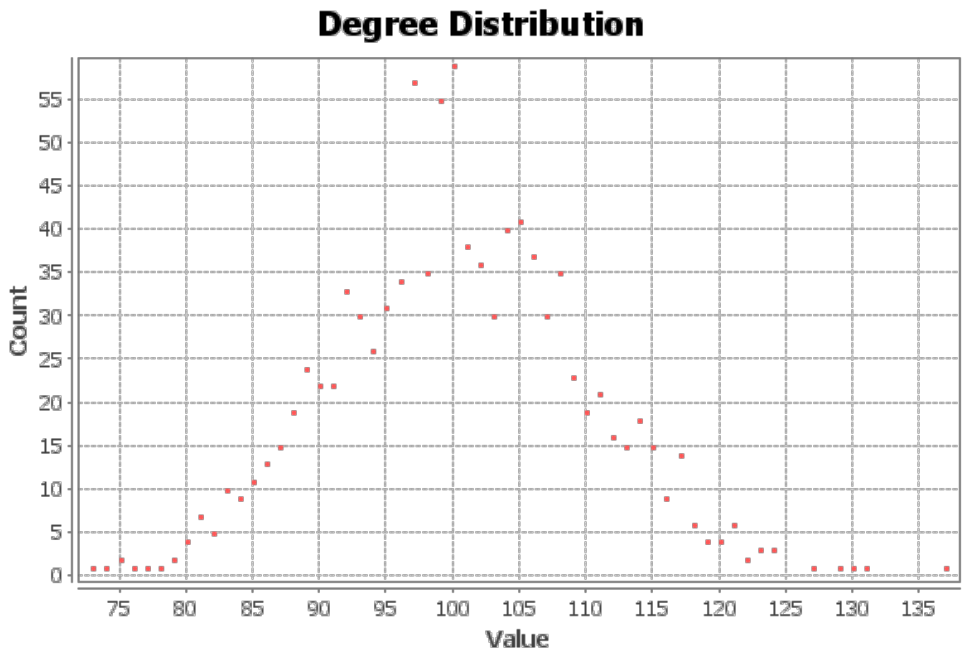




Generators

- Random graphs
- Dynamic graphs
- Multi-graphs
- Additional generators (e.g., Barabasi-Albert Model) are available as plugins

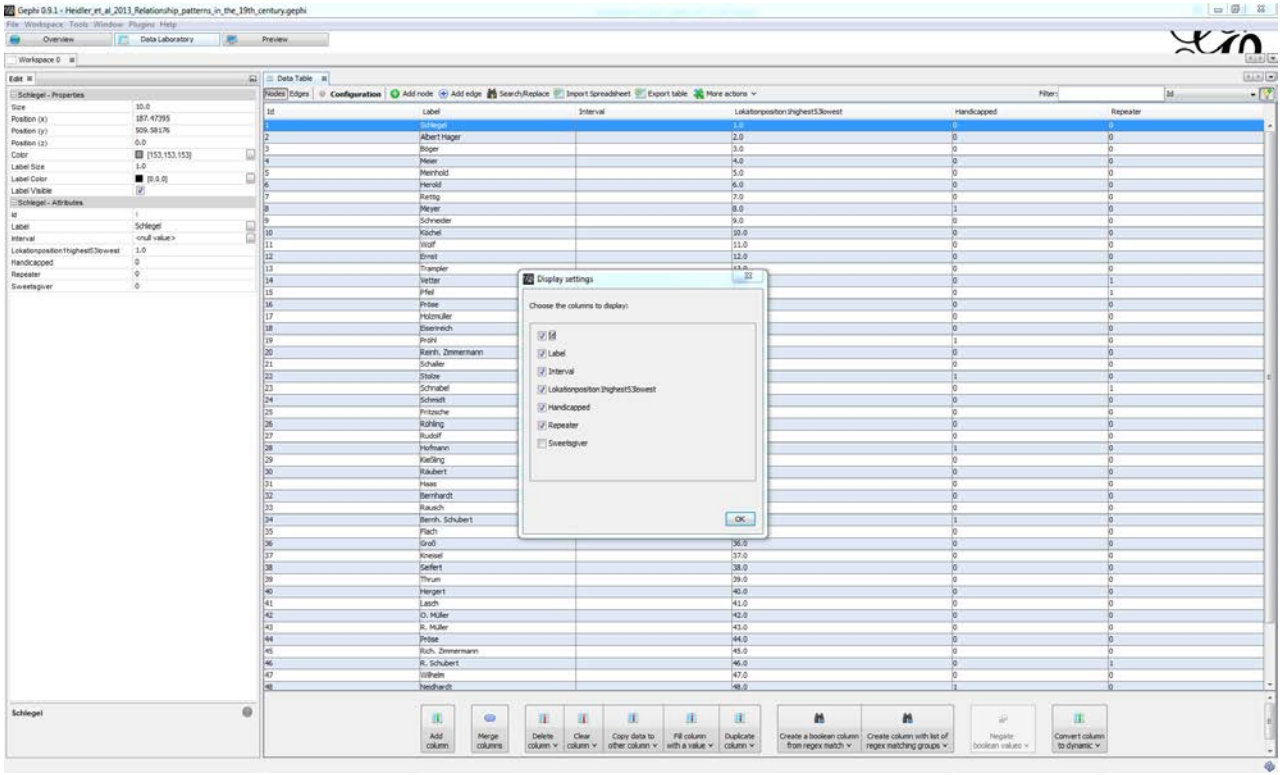




Random graph, 1000 nodes, wiring probability p = 0.1

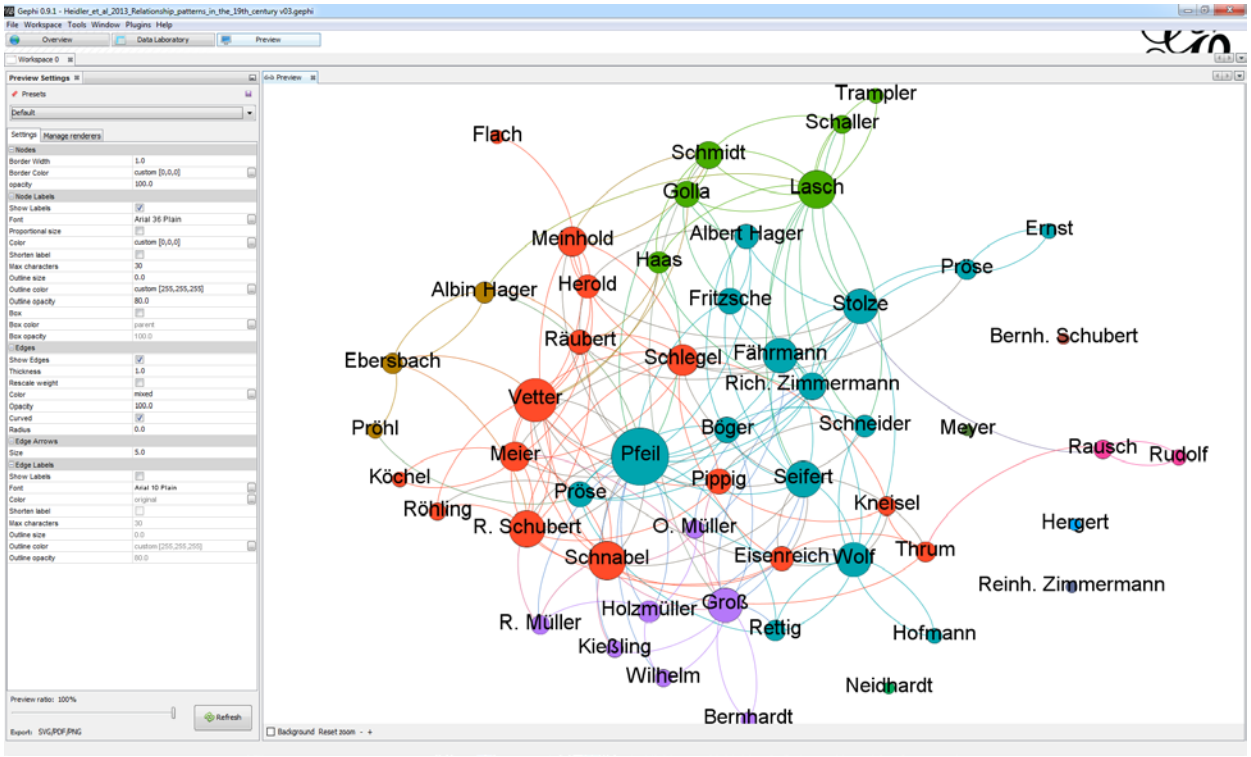
Node and Edge Attribute Data in Tabular Form

- Attributes can be edited
- Nodes and edges can be created or deleted
- Columns can be added, deleted, merged, copied, etc.
- Import from spreadsheet is supported
- Dynamic regexp filtering



Rendering a Graph for Export

- Define the appearance of the graph before exporting for publication
- Parameters can be saved in presets and reused later



Exporting Data from Gephi

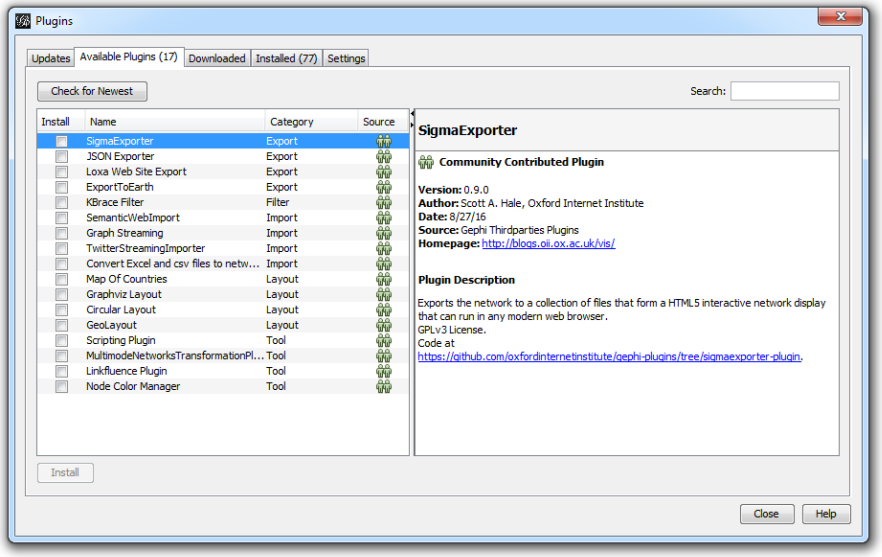
Gephi generates file results in:

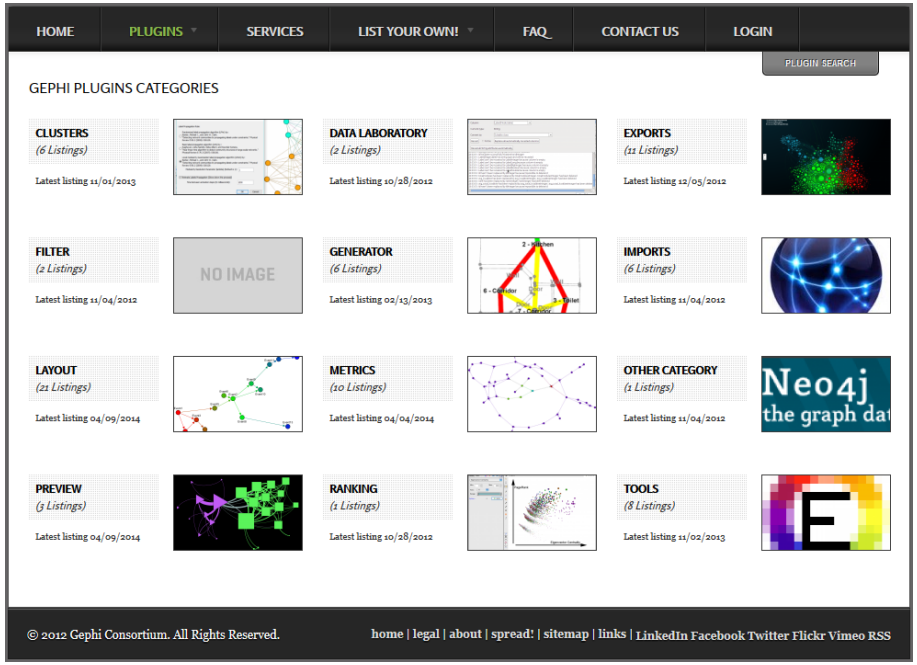
- PDF
- SVG
- PNG
- CSV
- GDF
- GEXF
- GML
- GraphML
- UCINET DL
- Pajek NET
- Netdraw VNA
- Spreadsheet


```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.3" version="1.3" xmlns:viz="http://www.gexf.net/1.3/viz" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.gexf.net/1.3
http://www.gexf.net/1.3/gexf.xsd">
  <meta lastmodifieddate="2016-09-09">
    <creator>Gephi 0.9</creator>
    <description></description>
  </meta>
  <graph defaultedgetype="directed" idtype="integer" mode="static">
    <attributes class="node" mode="static">
    <nodes>
      <node id="1" label="Schlegel">
        <attvalues>
          <attvalue for="lokationposition1highest53lowest" value="1.0"></attvalue>
          <attvalue for="handicapped" value="0"></attvalue>
          <attvalue for="repeater" value="0"></attvalue>
          <attvalue for="sweetsgiver" value="0"></attvalue>
          <attvalue for="eccentricity" value="6.0"></attvalue>
          <attvalue for="closnesscentrality" value="0.33070866141732286"></attvalue>
          <attvalue for="harmonicclosnesscentrality" value="0.3912698412698412 7"></attvalue>
          <attvalue for="betweenesscentrality" value="0.10878016319192789"></attvalue>
          <attvalue for="componentnumber" value="0"></attvalue>
          <attvalue for="strongcompnum" value="1"></attvalue>
          <attvalue for="indegree" value="8"></attvalue>
          <attvalue for="outdegree" value="3"></attvalue>
          <attvalue for="degree" value="11"></attvalue>
          <attvalue for="clustering" value="0.12222222238779068"></attvalue>
        </attvalues>
        <viz:size value="10.0"></viz:size>
        <viz:position x="-26.989138" y="94.667274"></viz:position>
        <viz:color r="237" g="248" b="251"></viz:color>
      </node>
      <node id="2" label="Albert Hager">
      <node id="3" label="Böger">
      <node id="4" label="Meier">
```

Plugins

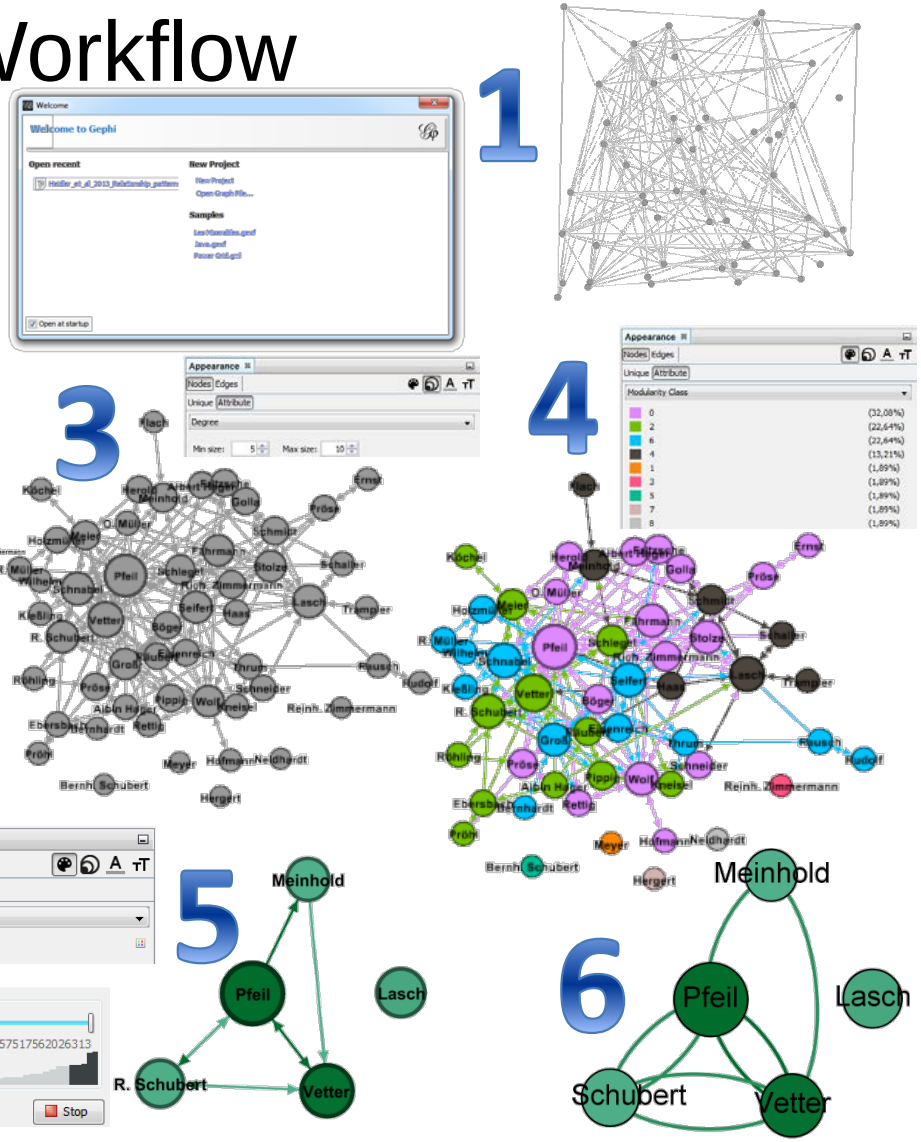
- A third-party code which adds features to Gephi
- Can be installed automatically or manually





Gephi Workflow

1. Import the network
2. Spatialize the network
3. Visualize attributes of the network
4. Compute new attributes with Gephi and visualize them
5. Filter out nodes / edges to highlight important findings about the network
6. Export the results

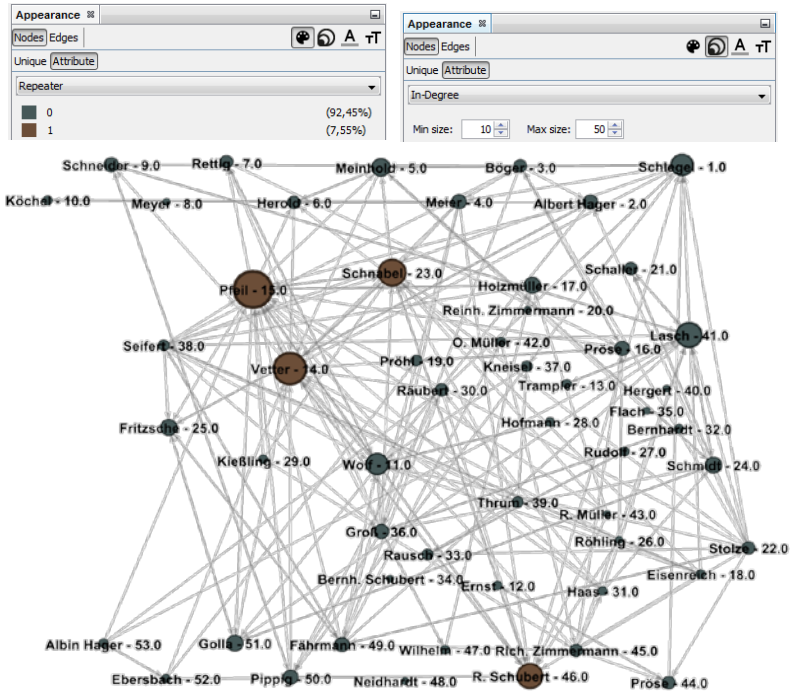


Analyzing a Social Network Using Gephi

- German boys' school class from 1880/1881 network

Nodes	Edges	Directed	Weighted	Diameter	Density	Average degree	Average path length
53	179	Y	N	8	0.065	6.755	3.38

- Outcome
  - friendship ties tend to be reciprocated
  - friendships form along the hierarchy of school performance which is embodied in a school ranking system
  - 4 repeaters (so-called street heroes which were strong, lively, and energetic “especially outside of the classroom”) and a “sweets giver” (a guy who gives sweets to other pupils), have a higher popularity in the class
  - pupils with a physical, psychological or social disability are less popular



ARTICLES AND BOOKS

- Bastian, Mathieu, Sebastien Heymann, and Mathieu Jacomy. "Gephi: an open source software for exploring and manipulating networks." *ICWSM* 8 (2009): 361-362.
- Cherven, Ken. *Network graph analysis and visualization with Gephi*. Packt Publishing Ltd, 2013.
- Cherven, Ken. *Mastering Gephi network visualization*. Packt Publishing Ltd, 2015.
- Khokhar, Devangana. *Gephi Cookbook*. Packt Publishing Ltd, 2015.

- Gephi additional tutorial slides are available at: https://www.slideshare.net/GraceBenefield/basics-gephi-tutorial
- More publications are listed at: https://gephi.org/users/publications/
- Sample Gephi datasets are available at: https://github.com/gephi/gephi/wiki/Datasets

Main Topics

- Overview of tools for network analysis and visualization
- Installing and using Gephi
- Gephi hands-on labs

Overview

Exercise 1

- Learn how to use Gephi interface
- Import a network file
- Compute simple network measures

Exercise 2

- Analyze a larger set of networks with Gephi
- Use the output of Gephi to compute additional network measures and study correlations between graph parameters

Exercise 3

- Use Gephi to visualize networks
- Explore different layouts and visualization parameters

http://pmtips.net/Blog/handson-project-manager

In this exercise

- Review network statistics available in Gephi
- Practice setting up network analysis and using different measures
- Compute three measures for each of the two networks:
  - Dolphins Social Network in New Zealand:
  - http://www-ersonal.umich.edu/~mejn/netdata/dolphins.zip
  - Erdős Collaboration Network:
  - http://vlado.fmf.uni-lj.si/pub/networks/data/
- Fill in the following table:

Network	# of nodes	# of edges	Density	# of triangles	Diameter
Dolphins	62	159			
Erdős	6,927	11,850			

Exercise review

- What graph formats does Gephi support?
- Does Gephi always correctly infer input file parameters?
- What network analysis measures does Gephi offer?
- Analysis results for the networks:

Network	# of nodes	# of edges	Density	# of triangles	Diameter
Dolphins	62	159			
Erdős	6,927	11,850			

In this exercise

- Compute three measures for each of the five networks. Fill in the following table:

Network	# of nodes	# of edges	Average node degree (<k>)	# of triangles (Δ)	Diameter (D)
Autonomous systems - Oregon-1-010407[1]	10,729	21,999	$< k >_1 =$	$\Delta_1 =$	$D_1 =$
Erdős Collaboration Network	6,927	11,850	$< k >_2 =$	$\Delta_2 =$	$D_2 =$
Autonomous systems - Oregon-1-010331[1]	10,670	22,002	$< k >_3 =$	$\Delta_3 =$	$D_3 =$
Autonomous systems - Oregon-2-010331[2]	10,900	31,180	$< k >_4 =$	$\Delta_4 =$	$D_4 =$
Enron Giant Component[3]	33,696	180,811	$< k >_5 =$	$\Delta_5 =$	$D_5 =$

[1] https://snap.stanford.edu/data/oregon1.html
[2] https://snap.stanford.edu/data/oregon2.html
[3] https://snap.stanford.edu/data/email-Enron.html

- Determine whether certain pairs of graph measures are correlated using **Pearson Correlation Coefficient** (PCC) as the measure of correlation. Draw scatter plots.

Pearson Correlation Coefficient

Suppose we are given a data sample consisting of $n \geq 1$ pairs of numbers $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. Let $\bar{x}$ and $\bar{y}$ denote respectively the mean values of the sets $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$; that is $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$, and $\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$.

The **Pearson Correlation Coefficient** (PCC) r for the sample is given by

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left[\sum_{i=1}^{n}(x_i - \bar{x})^2\right]}\sqrt{\left[\sum_{i=1}^{n}(y_i - \bar{y})^2\right]}}$$

where positive square roots are used for both terms in the denominator. The PCC value r defined above satisfies the condition $-1 \leq r \leq 1$. The value $r = 1$ indicates that a linear equation describes the relationship between the two sets X and Y . Similarly, $r = -1$ indicates a linear relationship between the two sets, with Y values decreasing as the X values increase. The value $r = 0$ indicates that X and Y are not correlated.

Follow these steps

- Import each network into a Gephi workspace. Run the appropriate statistics. If necessary, use filtering to select the required subgraph.
- For each of the five networks, find the **average node degree <k>**, the **number of triangles** Δ, and the **diameter D**. Since each of the five networks is connected, all five diameter values should be finite.
- Compute the PCC value r_1 for the sample $\{(< k >_1, \Delta_1), (< k >_2, \Delta_2), \ldots, (< k >_5, \Delta_5)\}$ using a tool of your choice (a calculator, MATLAB, an Excel spreadsheet, by writing a simple program, etc.).
- Compute the PCC value r_2 for the sample $\{(< k >, D_1), (< k >_2, D_2), \ldots, (< k >_5, D_5)\}$ using a tool of your choice.
- Prepare two **scatter plots**, one showing the pairs and the other $\{(< k >_1, \Delta_1), (< k >_2, \Delta_2), \ldots, (< k >_5, \Delta_5)\}$ showing the pairs. In each case, please show the Δ values along the *X* axis and the other value along the *Y* axis.

Exercise review I

- Did you have to modify edge list files to make sure they load correctly into Gephi? If so, what had to be done?
- Why were you asked to use the giant component to compute statistics for some networks?
- Which metric was the slowest to compute? Why? What algorithm was used to compute it and what is its asymptotic running time?
- Is there a correlation between the network measures you computed? If so, what kind of correlation it is and why? What does it tell you about the networks?

Exercise review II

- Analysis results for five networks:

Network	# of nodes	# of edges	Average node degree (<k>)	# of triangles (Δ)	Diameter (D)
Autonomous systems - Oregon-1-010407	10,729	21,999	$< k >_1 =$	$\Delta_1 =$	$D_1 =$
Erdős Collaboration Network	6,927	11,850	$< k >_2 =$	$\Delta_2 =$	$D_2 =$
Autonomous systems - Oregon-1-010331	10,670	22,002	$< k >_3 =$	$\Delta_3 =$	$D_3 =$
Autonomous systems - Oregon-2-010331	10,900	31,180	$< k >_4 =$	$\Delta_4 =$	$D_4 =$
Enron Giant Component	33,696	180,811	$< k >_5 =$	$\Delta_5 =$	$D_5 =$

In this exercise

- Review layout algorithms and visualization parameters available in Gephi
- Create publication quality vector visualizations for the following networks:

Network	# of nodes	# of edges
Zachary Karate Club[1]	34	78
American College Football[2]	115	613
Amazon product co-purchasing[3]	262,111	1,234,877

[1] http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm#zachary
[2] http://www-personal.umich.edu/~mejn/netdata/
[3] http://snap.stanford.edu/data/amazon0302.html

Exercise review

- What layout algorithms does Gephi offer?
- What are some possible ways of reducing clutter when visualizing large networks?