

Name Shakti Rajput
Reg.no 15BCE1066

Data Visualization(CSE3020)
Teacher: Pattabiraman V

Gephi

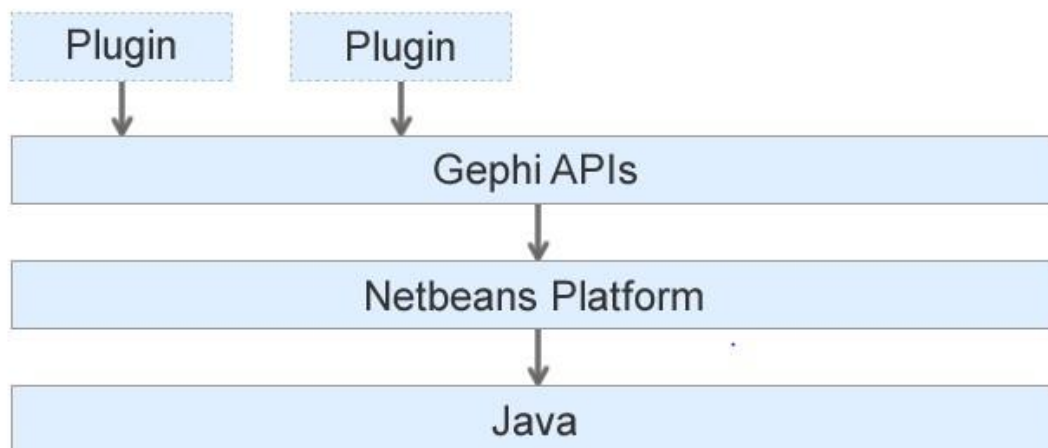
Probably the most popular network visualization package out there. Gephi doesn't require any programming knowledge. It's strength is that it is able to produce very high quality visualizations. It can also handle relatively large graphs - the actual size will depend on your infrastructure (particularly RAM) but you should be able to go up to 100,000 nodes without a problem. It does have the ability to calculate a few of the more common metrics such as degree, centrality, etc. but it's a stronger tool for visualization than analysis. It helps data analysts to intuitively reveal patterns and trends, highlight outliers and tells stories with their data. It uses a 3D render engine to display large graphs in real-time and to speed up the exploration. Gephi combines built-in functionalities and flexible architecture to:

- explore
- analyse
- spatialize
- filter
- cluster
- manipulate
- export all types of networks.

Gephi is based on a visualize-and-manipulate paradigm which allow any user to discover networks and data properties. Moreover, it is designed to follow the chain of a case study, from data file to nice printable maps.

Gephi Architecture

Gephi software architecture is modular and therefore each feature is split into modules. Modules depend on each other, like Java packages. Plugins developers simply create new modules that contains their code, add dependencies to Gephi modules, and distribute their plugins by creating an NBM package. See Gephi APIs documentation to learn more about the API plugins that can be used by modules



The Gephi goes beyond the simple fact that a client can pull data from a master: in fact, clients can interact with the master pushing data to it, in a REST architecture. The same data format used by the master to send graph events to the clients is used by clients to interact with the master.

The visualization package architecture is a compromise between flexibility and performances. In 3D engine design it is quite impossible to have both in the same time. Hence Gephi has flexibility where it doesn't harm efficiency.

Gephi Methods

Main interface for accessing the graph structure and develop algorithms.

A graph object belongs to a graph model, which really contains the data. Graph objects are therefore only accessors, with all convenient methods to read and modify the structure. Hence, **multiple** Graph objects can exist.

Graph Locking Graph structure possess a locking mechanism that avoids multiple threads to modify the structure concurrently. However, several readers are allowed.

Gephi APPLICATIONS AND ADVANTAGES

- **Exploratory Data Analysis:** intuition-oriented analysis by networks manipulations in real time.
- **Link Analysis:** revealing the underlying structures of associations between objects.
- **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.
- **Poster creation:** scientific work promotion with hi-quality printable maps.

Gephi Features

Real-time visualization

Profit from the fastest graph visualization engine to speed-up understanding and pattern discovery in large graphs. Powered by its ad-hoc OpenGL engine, Gephi is pushing the envelope on how interactive and efficient network exploration can be.

- Networks up to 100,000 nodes and 1,000,000 edges
- Iterate through visualization using dynamic filtering
- Rich tools for meaningful graph manipulation

Layout

Layout algorithms give the shape to the graph. Gephi provides state-of-the-art algorithms layout algorithms, both for efficiency and quality. The Layout palette allows user to change layout settings while running, and therefore dramatically increase user feedback and experience.

- Force-based algorithms
- Optimize for graph readability

Metrics

The statistics and metrics framework offer the most common metrics for social network analysis (SNA) and scale-free networks.

- Betweenness Centrality, Closeness, Diameter, Clustering Coefficient, PageRank
- Community detection (Modularity)
- Random generators
- Shortest path

Networks over time

Gephi is at the forefront of innovation with dynamic graph analysis. Users can visualize how a network evolves over time by manipulating the embedded timeline.

- Import temporal graph with the GEXF file format
- Run metrics over time (clustering coefficient)
- Graph streaming ready

Create cartography

Use ranking or partition data to make meaningful the network representation. Customize colors, size or labels to bring sense to the network representation. The vectorial preview module lets you put the final touch and care about aesthetics before exporting in SVG or PDF.

- Customizable PDF, SVG and PNG export
- Save presets

Dynamic filtering

Filter the network to select nodes and/or edges based on the network structure or data. Use interactive user interface to filter the network in real-time.

- Create complex filter query without scripting
- Build new networks from the filtering result
- Save your favorite queries

Data Table and Edition

Gephi has its own Data Laboratory with an Excel-like interface to manipulate data columns, search and transform the data.

- Powerful Search/Replace
- Manipulate columns
- Batch-edit, custom column merge and more

Input/Output

Gephi can read the majority of graph file formats but also supports CSV and relational databases import.

- Spreadsheet import wizard
- Database import
- Save/Load project files

Extensible

The built-in Plugins Center automatically gets the list of plugins available from the Gephi Plugin portal and takes care of all software updates. There are dozens of community-built plugins that extends Gephi's functionalities.

Gephi Layouts

ForceAtlas

ForceAtlas layout Home-brew layout of Gephi, it is made to spatialize SmallWorld / Scale-free networks. It is focused on quality (meaning "being useful to explore real data") to allow a rigorous interpretation of the graph (e.g. in SNA) with the fewest biases possible, and a good readability even if it is slow.

Fruchterman-Reingold

Fruchterman-Reingold layout It simulates the graph as a system of mass particles. The nodes are the mass particles and the edges are springs between the particles. The algorithms try to minimize the energy of this physical system. It has become a standard but remains very slow.

Yifan Hu Multilevel

Yifan Hu Multilevel layout It is a very fast algorithm with a good quality on large graphs. It combines a force-directed model with a graph coarsening technique (multilevel algorithm) to reduce the complexity. The repulsive forces on one node from a cluster of distant nodes are approximated by a Barnes-Hut calculation, which treats them as one super-node. It stops automatically.

OpenOrd

OpenOrd layout It expects undirected weighted graphs and aims to better distinguish clusters. It can be run in parallel to speed up computing, and stops automatically. The algorithm is originally based on Fruchterman-Reingold and works with a fixed number of iterations controlled via a simulated annealing type schedule (liquid, expansion, cool-down, crunch, and simmer). Long edges are cut to allow clusters to separate.

ForceAtlas 2

ForceAtlas 2 layout Improved version of the Force Atlas to handle large networks while keeping a very good quality. Nodes repulsion is approximated with a Barnes-Hut calculation, which therefore reduces the algorithm complexity. Replace the "attraction" and "repulsion" forces by a "scaling" parameter.

Circular layout

Circular layout It draws nodes in a circle ordered by ID, a metric (degree, betweenness centrality...) or by an attribute. Use it to show a distribution of nodes with their links.

Radial Axis Layout

Radial Axis Layout It is provided with the Circular Layout plugin. It groups nodes and draws the groups in axes (or spars) radiating outwards from a central circle. Groups are generated using a metric (degree, betweenness centrality...) or an attribute. Use it to study homophily by showing distributions of nodes inside groups with their links.

Geographic map

Geographic map with GeoLayout The GeoLayout uses latitude/longitude coordinates to set nodes position on the network. Several projections are available, including Mercator which is used by Google Maps and other online services. The two node attribute columns for coordinates should be in numeric format.

Conclusion

- Gephi has a very easy interface to use and get to the tools you need quickly. Without much training or learning, it's pretty simple to figure out.
- The data import process is very easy in CSV format and the software produces a graph automatically once the correct data is loaded and mapped together (edges and nodes).
- The visualization is very easy to edit, drag around, and customize. There is flexibility to change the size and color of nodes and edges to represent various characteristics of the graph.
- In Gephi, there's really not a great export feature for the map you've created. Screenshots can be taken, but you can't currently export to an image or HTML document.

- All the interactivity is lost if not using the Gephi file itself. A screenshot does not have nearly the impact as if you could move the edges and nodes around like you can in the software itself.
- All the text label size is edited by one master control. It would be nice if this would size proportionately with the size of the nodes or if this could be a manual selection at times. With all the edges flowing around, sometimes it's hard to read the text if it's too small or too large.

Dataset Taken from this website

The screenshot shows the Stanford Network Analysis Platform (SNAP) website. The main heading is "Social circles: Facebook". Below it, there is a "Dataset information" section explaining that the dataset consists of 'circles' (or 'friends lists') from Facebook, collected from survey participants using a Facebook app. It mentions that the data is anonymized and includes node features (profiles), circles, and ego networks. A "Dataset statistics" table is also present, showing metrics like Nodes (4039), Edges (88234), and various clustering coefficients. The website is by Jure Leskovec and includes a sidebar with navigation links like "People", "Papers", "Projects", and "Citing SNAP".

File	Description
facebook.tar.gz	Facebook data (10 networks, anonymized)
facebook_combined.txt.gz	Edges from all egonets combined
readme-Ego.txt	Description of files

Download These files

File	Description
facebook.tar.gz	Facebook data (10 networks, anonymized)
facebook_combined.txt.gz	Edges from all egonets combined
readme-Ego.txt	Description of files

Data download is in form of text format

```
C:\Users\Shakti\Desktop\New folder (3)\facebook_combined.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
5hFeb.java Lab_5hFeb.java mappersplit.java new_1.java partition.java split.java facebook_combined.txt
1 0 1
2 0 2
3 0 3
4 0 4
5 0 5
6 0 6
7 0 7
8 0 8
9 0 9
10 0 10
11 0 11
12 0 12
13 0 13
14 0 14
15 0 15
16 0 16
17 0 17
18 0 18
19 0 19
20 0 20
21 0 21
22 0 22
23 0 23
24 0 24
25 0 25
26 0 26
27 0 27
28 0 28
29 0 29
30 0 30
31 0 31
32 0 32
33 0 33
34 0 34
35 0 35
36 0 36
37 0 37
38 0 38
Normal text file length: 854362 lines: 88235 Ln: 1 Col: 1 Sel: 0 | 0 UNIX UTF-8 INS
Type here to search 23:35 20-02-2018
```

Used this Website to convert in to GML format which is used by Gephi

advanCSE 'Byte'ing the bits :-)

Graph Clustering Visualization - .txt to .gml converter - LINUX

Here, I present a fundamental implementation of converting files used for visualization of graphs, from the .txt format to the .gml format.

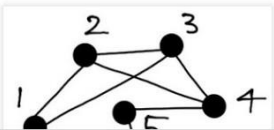
The following program follows these assumptions -

1. Graph is Undirected
2. Only attributes in .txt file are the edge connections
3. Nodes are present as numbers, and not as labels

The tar ball for Linux can be downloaded as [TXT2GMLv1.0](#).

I present a basic example, for illustration of why this conversion is important.

Lets take a simple graph -



Digital Assignment....pdf facebook_combined.g...

After Converting into Gml Format


```
C:\Users\Shakti\Desktop\facebook_combined.gml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
5hFeb.java Lab_5hFeb.java mappersplit.java new_1.java partition.java split.java facebook_combined.txt facebook_combined.gml
1 Creator "Karan Baja" - Source file: facebook_combined.txt
2 graph
3 [
4   node
5   [
6     id 1
7   ]
8   node
9   [
10    id 2
11  ]
12  node
13  [
14    id 3
15  ]
16  node
17  [
18    id 4
19  ]
20  node
21  [
22    id 5
23  ]
24  node
25  [
26    id 6
27  ]
28  node
29  [
30    id 7
31  ]
32  node
33  [
34    id 8
35  ]
36  node
37  [
38    id 9
```

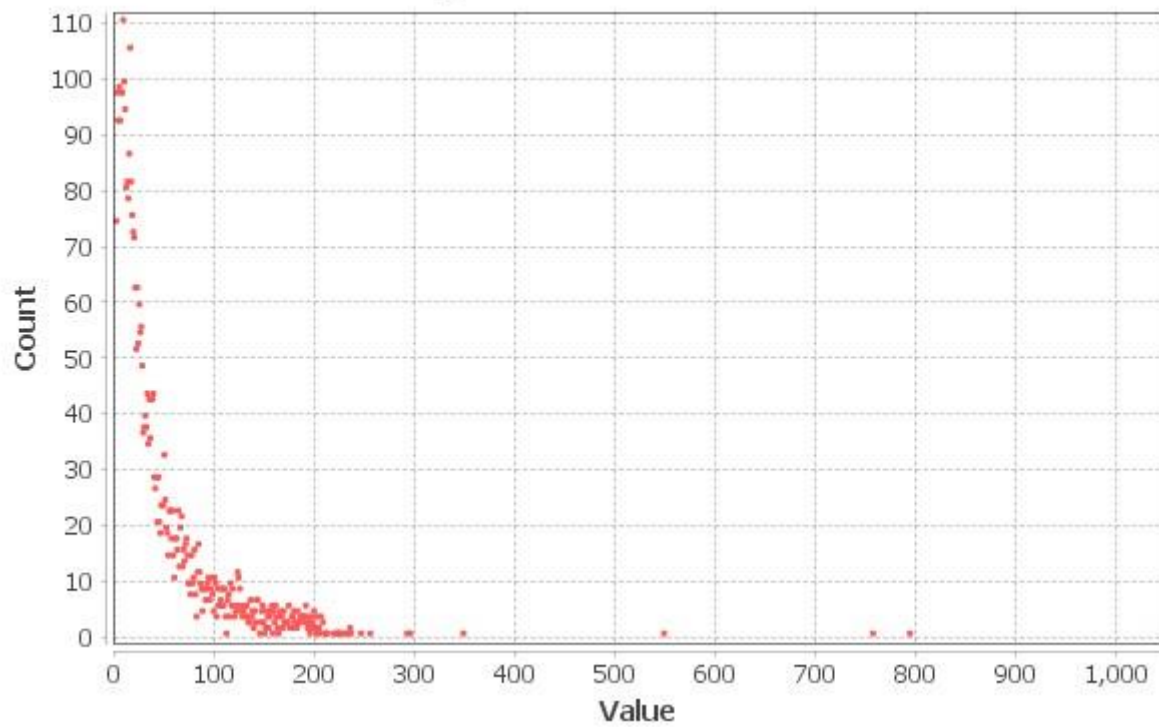
Visualization Result

Degree Report

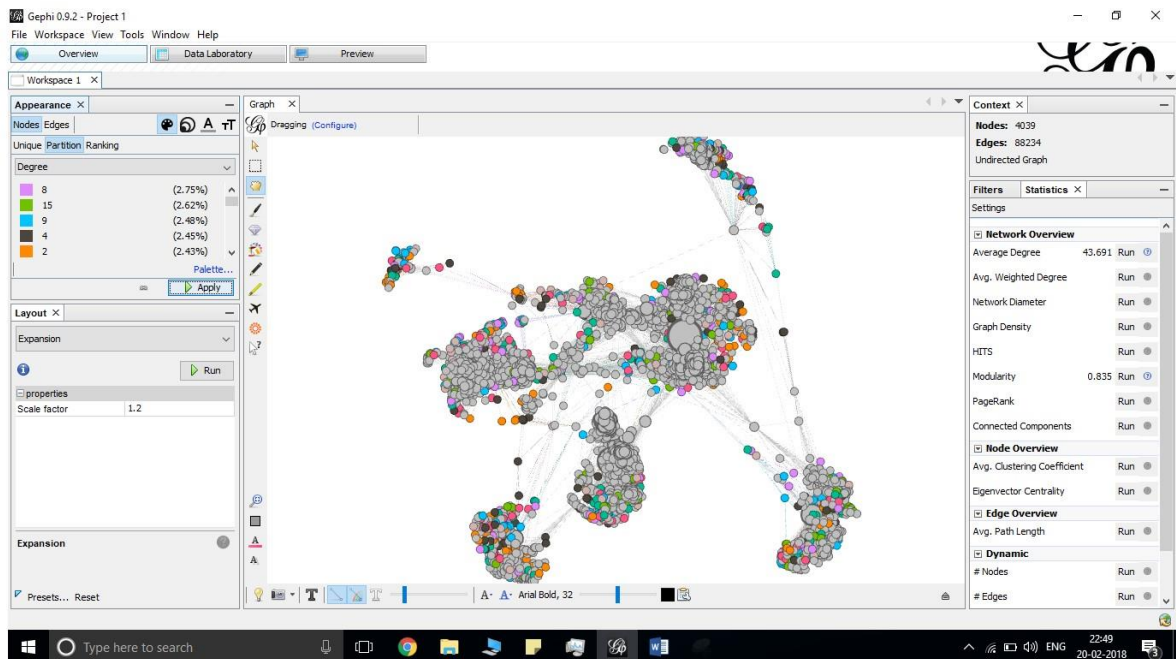
Results:

Average Degree: 43.691

Degree Distribution

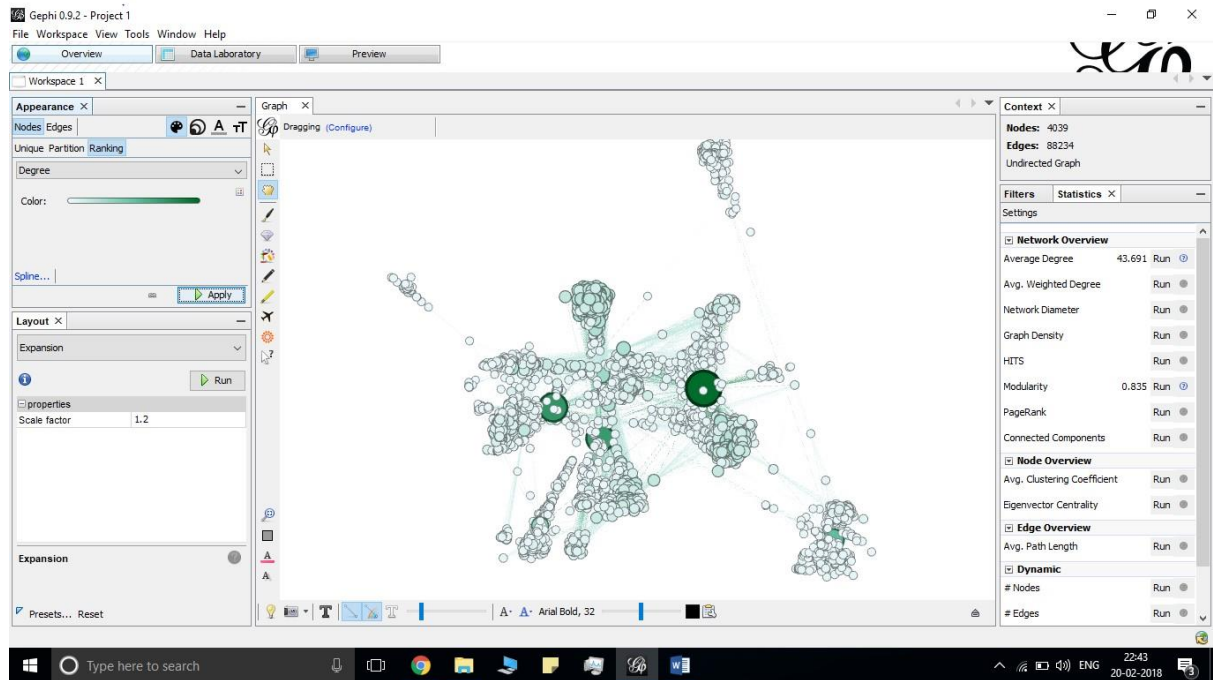


Various Degree Representation Layout(Yifanhu)



Ranking on Degree

Layout(FORCEATLUS)



Modularity Report

Parameters:

Randomize: On

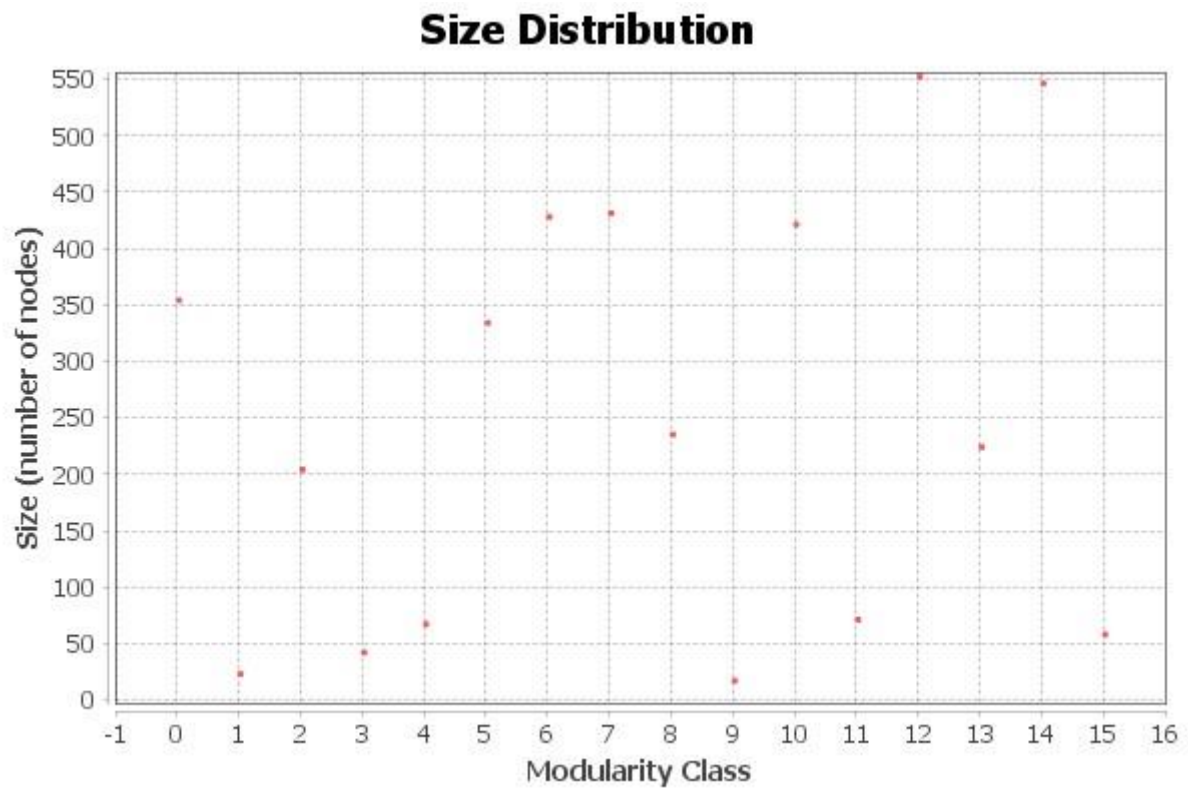
Use edge weights: On Resolution:
1.0

Results:

Modularity: 0.835

Modularity with resolution: 0.835

Number of Communities: 16



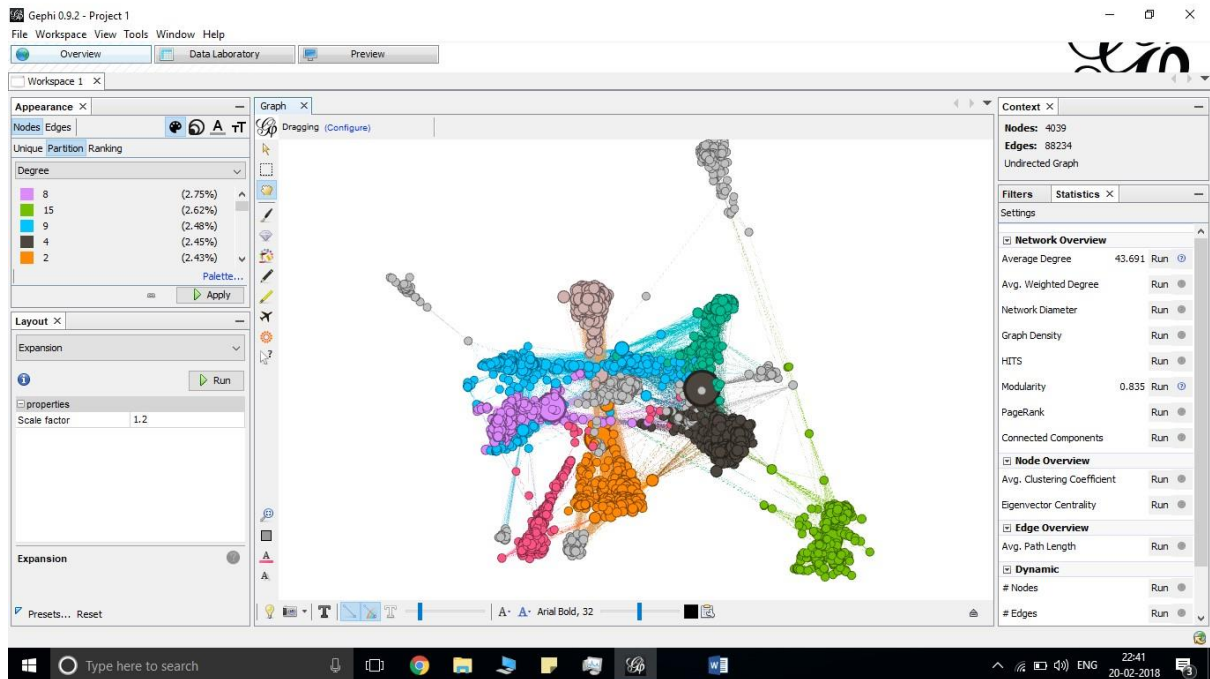
Algorithm:

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, *Fast unfolding of communities in large networks*, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

Resolution:

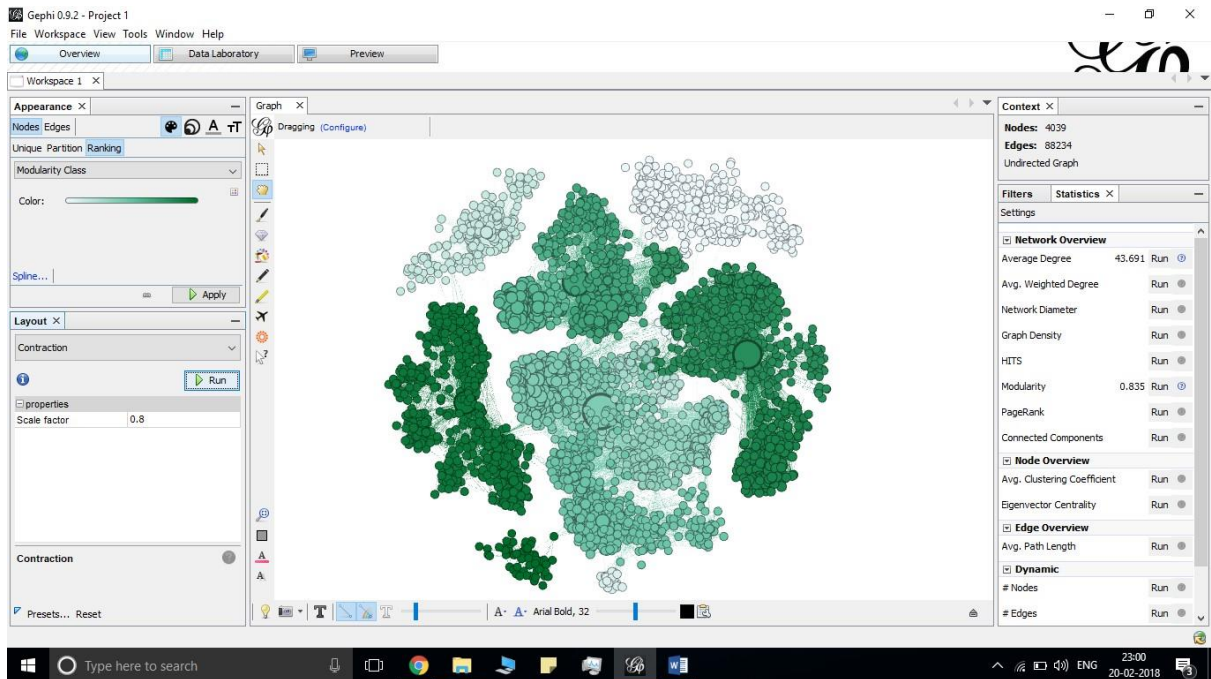
R. Lambiotte, J.-C. Delvenne, M. Barahona *Laplacian Dynamics and Multiscale Modular Structure in Networks* 2009

Modularity Representation of various Nodes



Ranking on Modularity

Layout(OpenOrd)



PageRank Report

Parameters:

Epsilon = 0.001

Probability = 0.85

Results:

PageRank Distribution

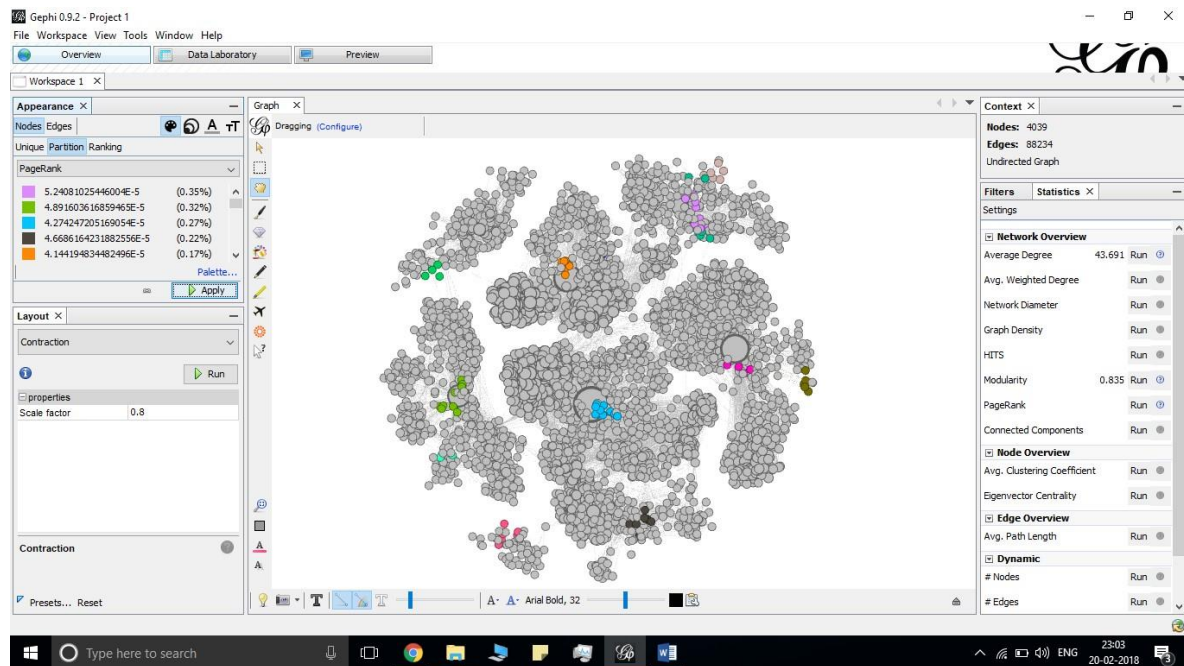


Algorithm:

Sergey Brin, Lawrence Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, in Proceedings of the seventh International Conference on the World Wide Web (WWW1998):107-117

Page Rank Nodes Visualization

Layout(Radial Axis Layout)



Graph Density Report

Parameters:

Network Interpretation: undirected

Results:

Density: 0.011

Eigenvector Centrality Report

Parameters:

Network Interpretation: undirected

Number of iterations: 100

Sum change: 0.4279073669804361

Results:

A dot plot showing the distribution of scores. The x-axis is labeled 'Score' and ranges from 0 to 1. The y-axis is labeled 'Count' and ranges from 0 to 14. Red dots represent the frequency of each score. Most scores are clustered near 0, with a count of 14 for a score of 0. A single dot is at a score of 1 with a count of 1.

Score	Count
0.00	14
0.01	13
0.02	11
0.03	9
0.04	7
0.05	6
0.06	5
0.07	4
0.08	3
0.09	2
0.10	1
0.11	1
0.12	1
0.13	1
0.14	1
0.15	1
0.16	1
0.17	1
0.18	1
0.19	1
0.20	1
0.21	1
0.22	1
0.23	1
0.24	1
0.25	1
0.26	1
0.27	1
0.28	1
0.29	1
0.30	1
0.31	1
0.32	1
0.33	1
0.34	1
0.35	1
0.36	1
0.37	1
0.38	1
0.39	1
0.40	1
0.41	1
0.42	1
0.43	1
0.44	1
0.45	1
0.46	1
0.47	1
0.48	1
0.49	1
0.50	1
0.51	1
0.52	1
0.53	1
0.54	1
0.55	1
0.56	1
0.57	1
0.58	1
0.59	1
0.60	1
0.61	1
0.62	1
0.63	1
0.64	1
0.65	1
0.66	1
0.67	1
0.68	1
0.69	1
0.70	1
0.71	1
0.72	1
0.73	1
0.74	1
0.75	1
0.76	1
0.77	1
0.78	1
0.79	1
0.80	1
0.81	1
0.82	1
0.83	1
0.84	1
0.85	1
0.86	1
0.87	1
0.88	1
0.89	1
0.90	1
0.91	1
0.92	1
0.93	1
0.94	1
0.95	1
0.96	1
0.97	1
0.98	1
0.99	1
1.00	1

Layout(FORCEATLUS2)

