Network Science

Hands on: Gephi

Albert-László Barabási

with

Emma K. Towlson, Sebastian Ruf,
Michael Danziger, and Louis Shekhtman

www.BarabasiLab.com

Step 0: Installation and data

Hopefully you already downloaded and successfully installed Gephi... if not please navigate to https://gephi.org/users/download/ and do so NOW.

You have been emailed 2 datasets:

- Les Mis characters
- Network Science co-authorship

...please also download these now

Network visualisation tools

• Gephi is one of several pieces of software out there that we use to visualise networks. There is no such thing as “THE Tool” to use. We chose it to focus on in this class as it:

(a) is free and open source

(b) is capable of performing rudimentary network analyses at the click of a button

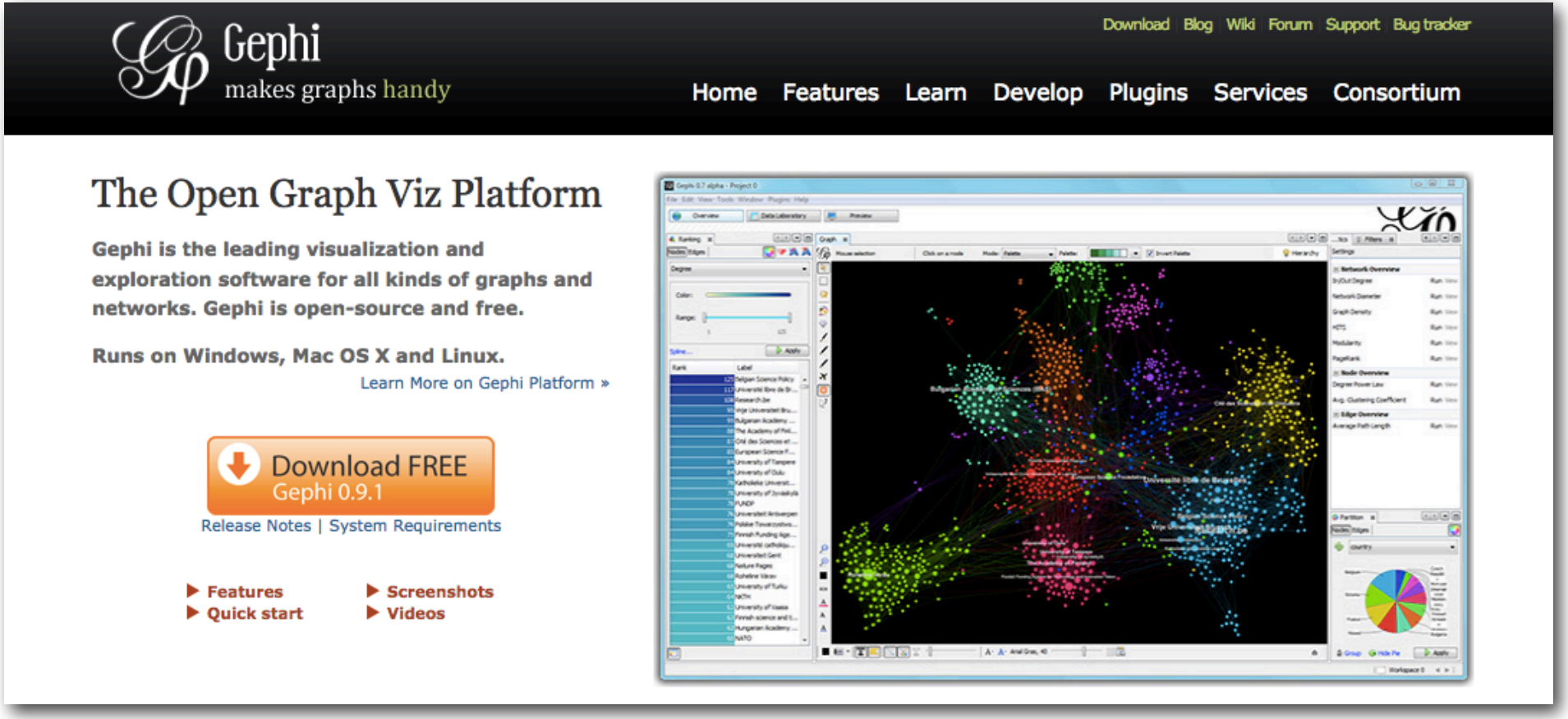
• The choice of software, programming language, library, module etc etc, depends on the application, the problem, the field, your personal preference...

• ...As such, see these slides as guidelines. If you are already familiar with another tool, you are free to use it.

• Consider (lecturer’s personal preferences...) researching Cytoscape, d3, NetworkX etc for your project / future endeavours... the principles are the same, the implementation is slightly different.

Gephi

The following tutorial is inspired by Roberta Sinatra's inspiration from https://gephi.github.io



Victor Hugo: Les Misérables

Network visualisation: telling a story or reading a story?

What does the file look like? -> .csv

Nodes:

Attributes

node_list.csv | edge_list.csv

ID	label	maincharacter
0	"Myriel"	0
1	"Napoleon"	0
2	"MlleBaptistine"	0
3	"MmeMagloire"	0
4	"CountessDeLo"	0
5	"Geborand"	0
6	"Champtercier"	0
7	"Cravatte"	0
8	"Count"	0
9	"OldMan"	0
10	"Labarre"	0
11	"Valjean"	1
12	"Marguerite"	0
13	"MmeDeR"	0
14	"Isabeau"	0
15	"Gervais"	0
16	"Tholomyes"	0
17	"Listolier"	0
18	"Fameuil"	0
19	"Blacheville"	0
20	"Favourite"	0
21	"Dahlia"	0
22	"Zephine"	0
23	"Fantine"	1
24	"MmeThenardier"	0
25	"Thenardier"	1
26	"Cosette"	1
27	"Javert"	1
28	"Fauchelevent"	0
29	"Bamatabois"	0
30	"Perpetue"	0
31	"Simplice"	0
32	"Scaufflaire"	0
33	"Woman1"	0
34	"Judge"	0

What does the file look like? -> .csv

Edges:

node_list.csv | edge_list.csv

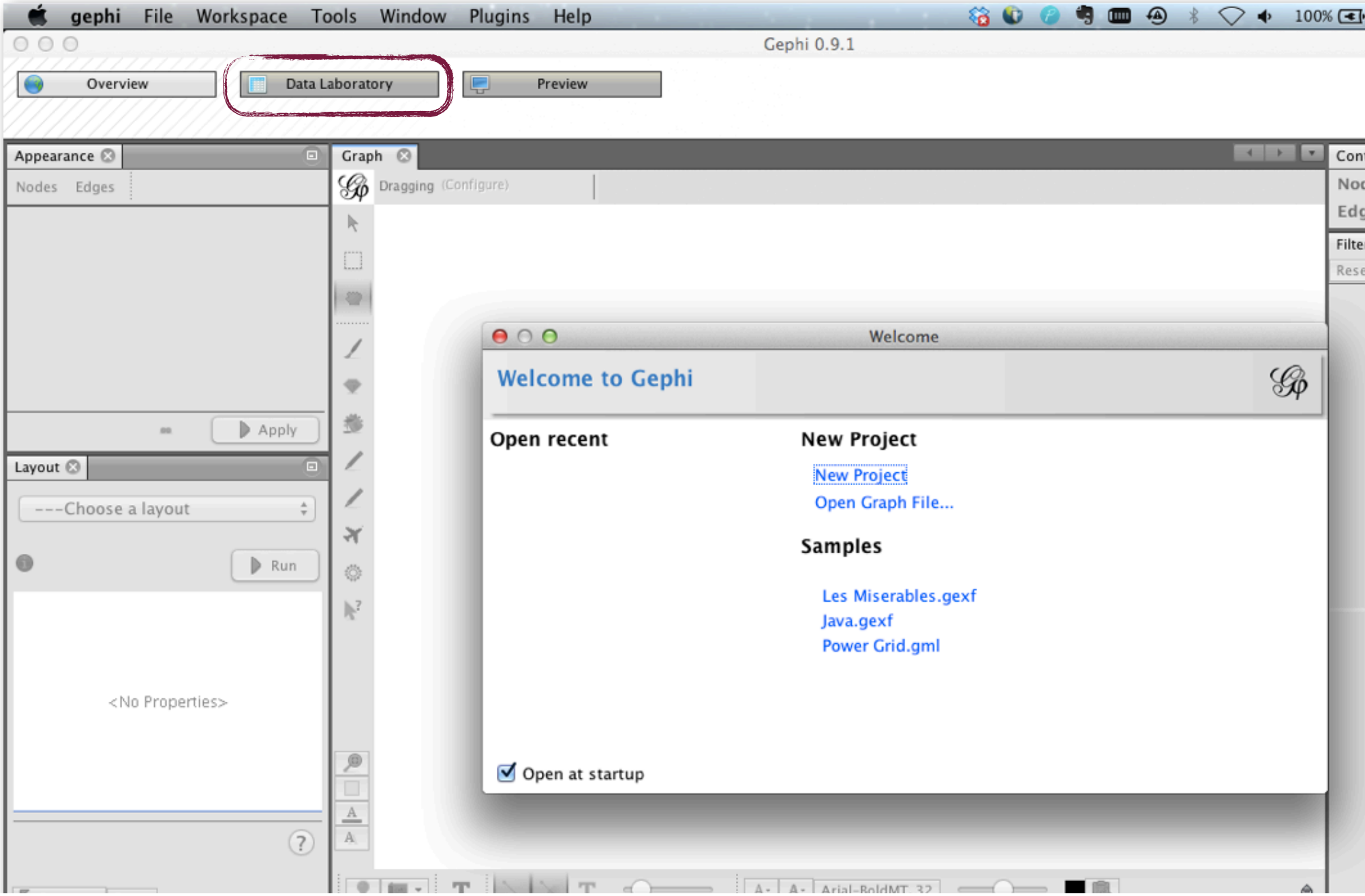
Source	Target	Weight
1	0	1
2	0	8
3	0	10
3	2	6
4	0	1
5	0	1
6	0	1
7	0	1
8	0	2
9	0	1
11	10	1
11	3	3
11	2	3
11	0	5
12	11	1
13	11	1
14	11	1
15	11	1
17	16	4
18	16	4
18	17	4
19	16	4
19	17	4
19	18	4
20	16	3
20	17	3
20	18	3
20	19	4
21	16	3
21	17	3
21	18	3
21	19	3
21	20	5
22	16	3
22	17	3

What does the file look like? -> .gml

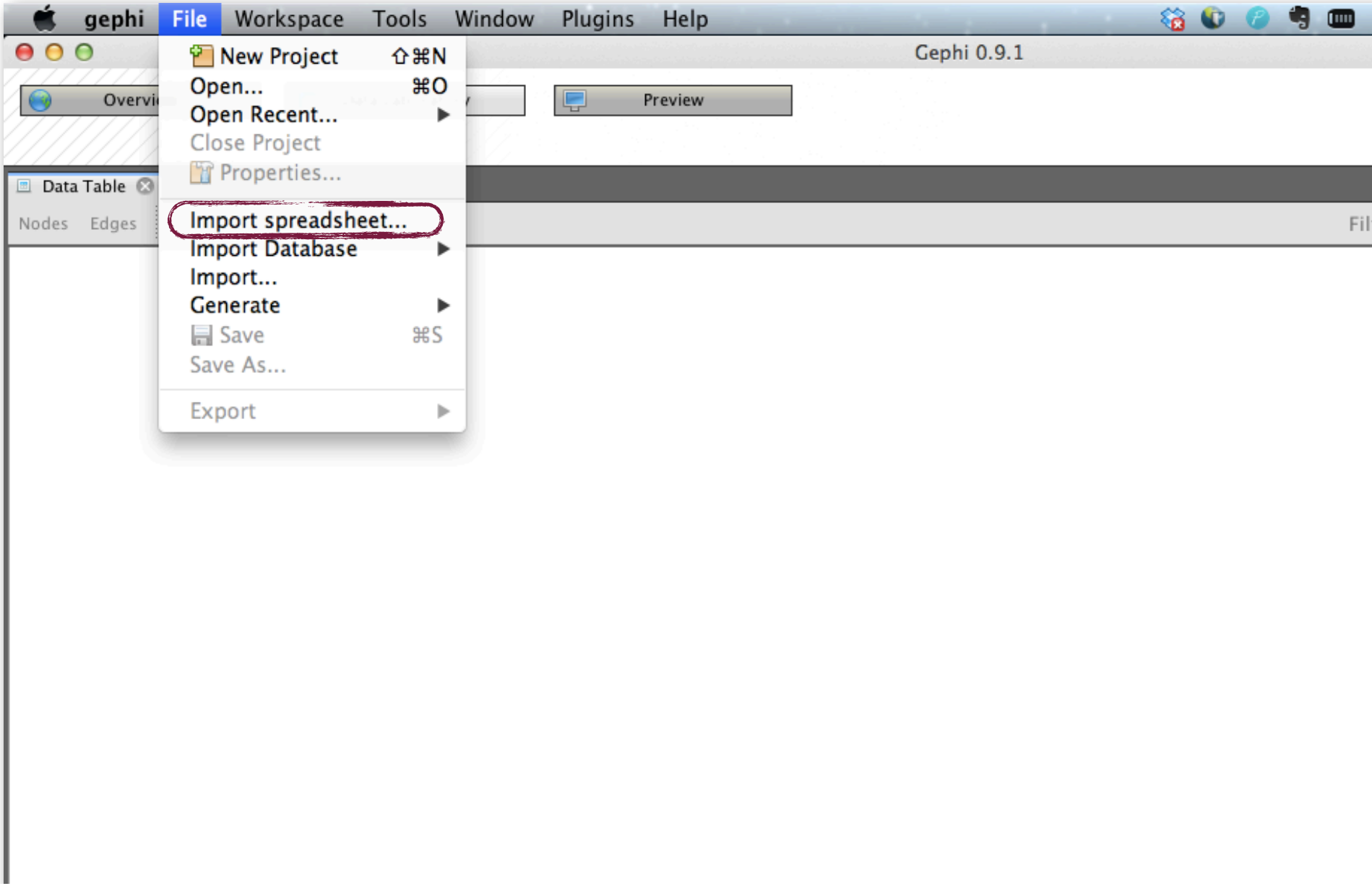
lesmiserables.gml

```
Creator "Mark Newman on Fri Jul 21 12:44:53 2006"
graph
[
  node
  [
    id 0
    label "Myriel"
    maincharacter 0
  ]
  node
  [
    id 1
    label "Napoleon"
    maincharacter 0
  ]
  node
  [
    id 2
    label "MlleBaptistine"
    maincharacter 0
  ]
  node
  [
    id 3
    label "MmeMagloire"
    maincharacter 0
  ]
  node
  [
    id 4
    label "CountessDeLo"
    maincharacter 0
  ]
  node
  [
    id 5
```

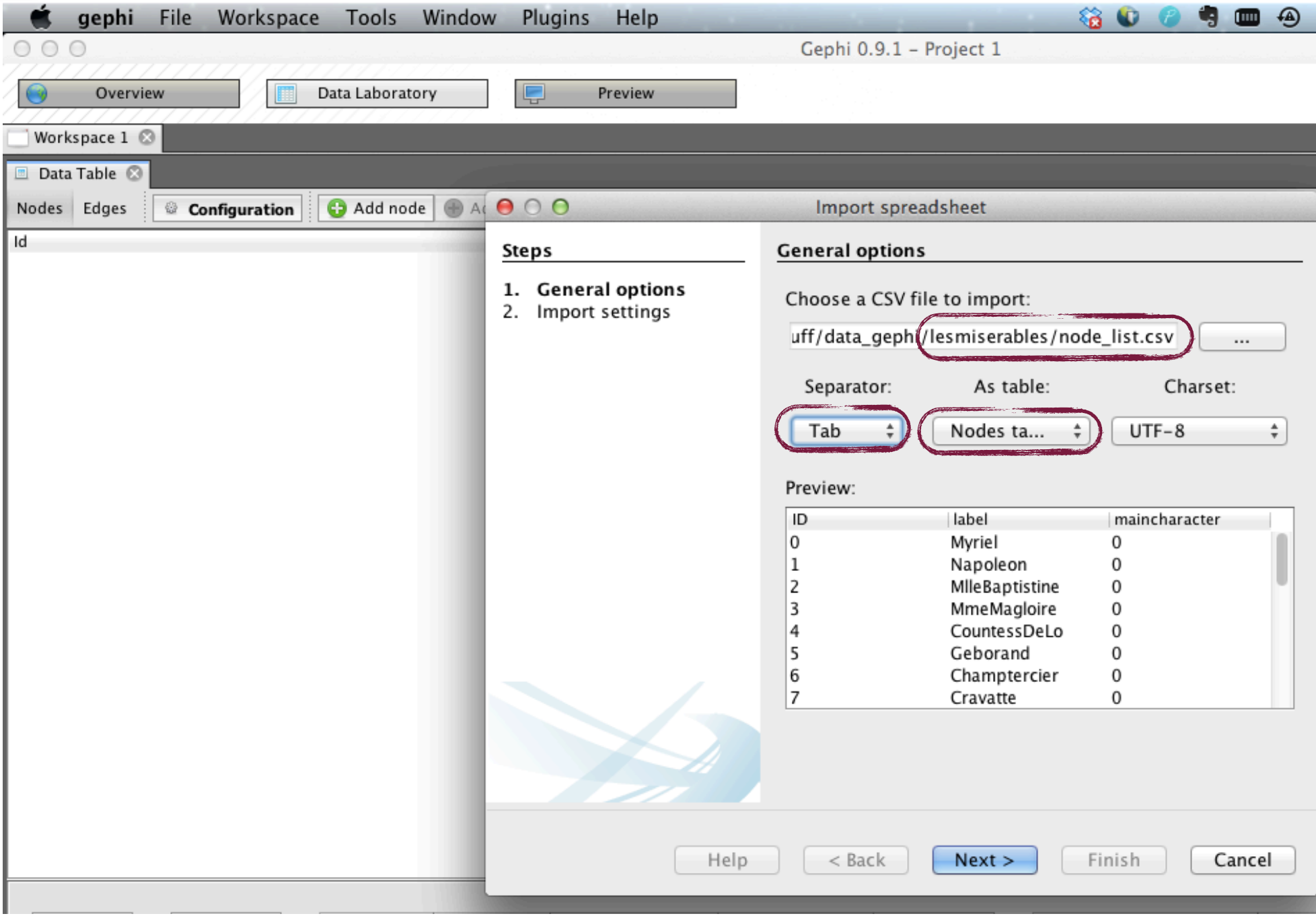
Let's open Gephi now



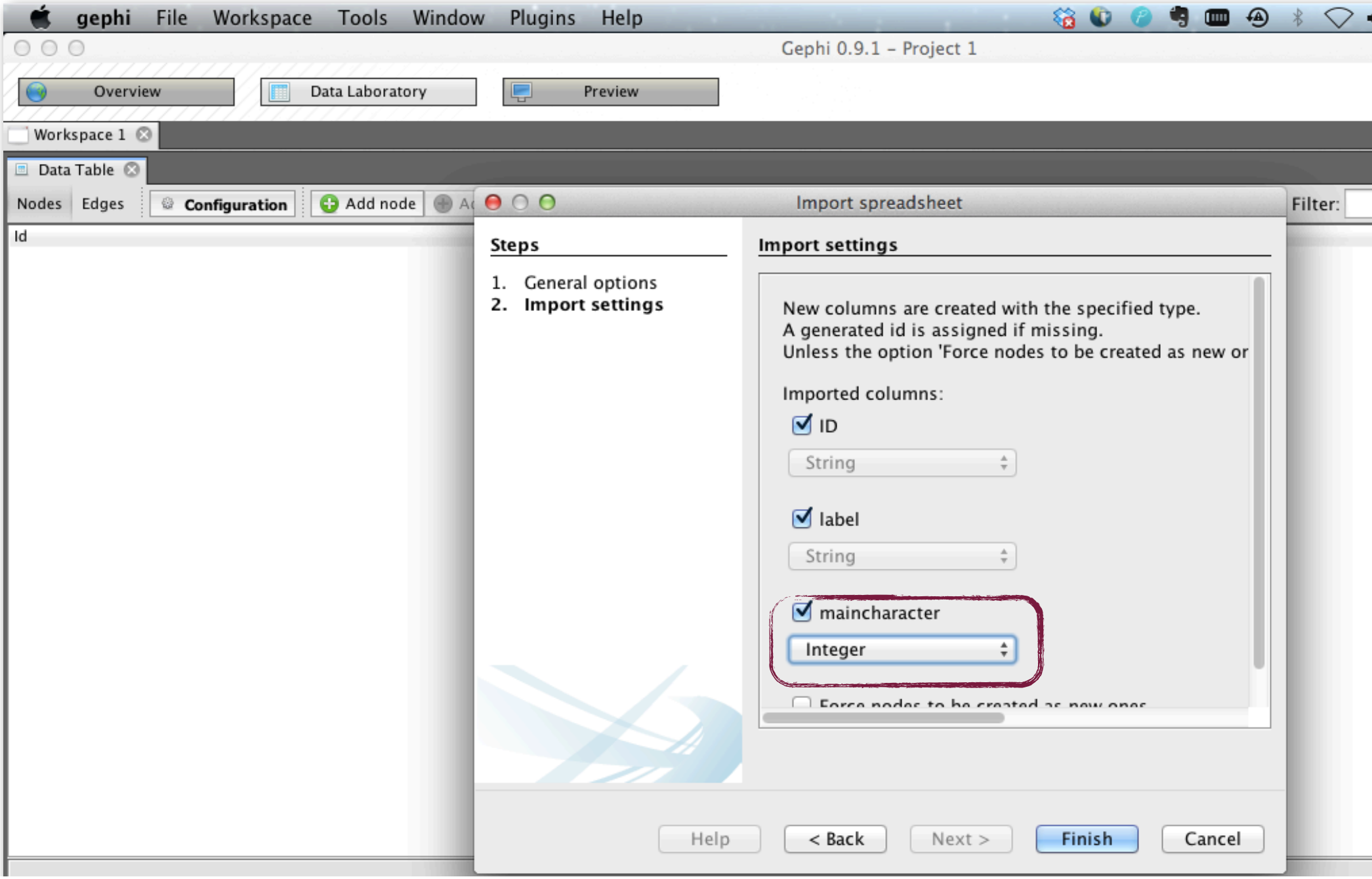
Importing the file -> .csv



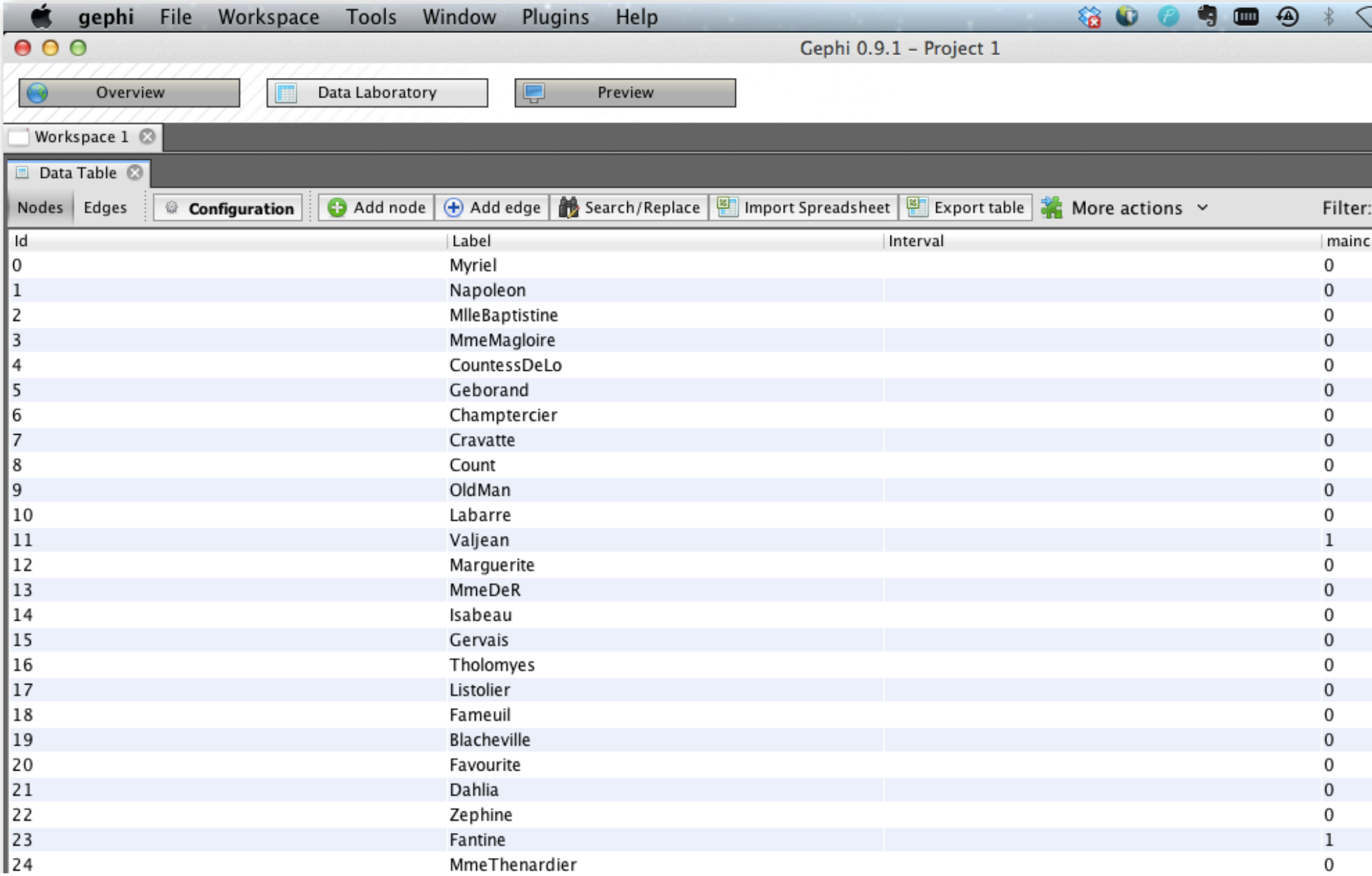
Importing the file -> .csv



Importing the file -> .csv

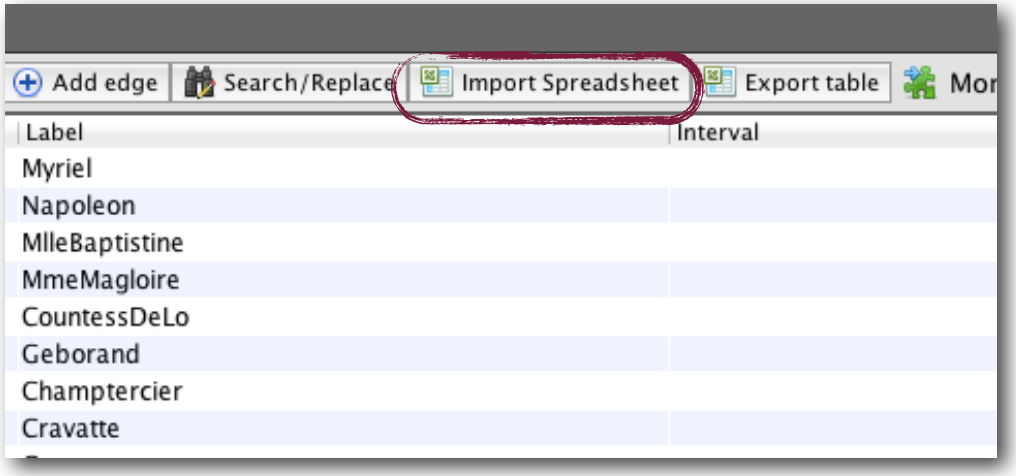


Importing the file -> .csv

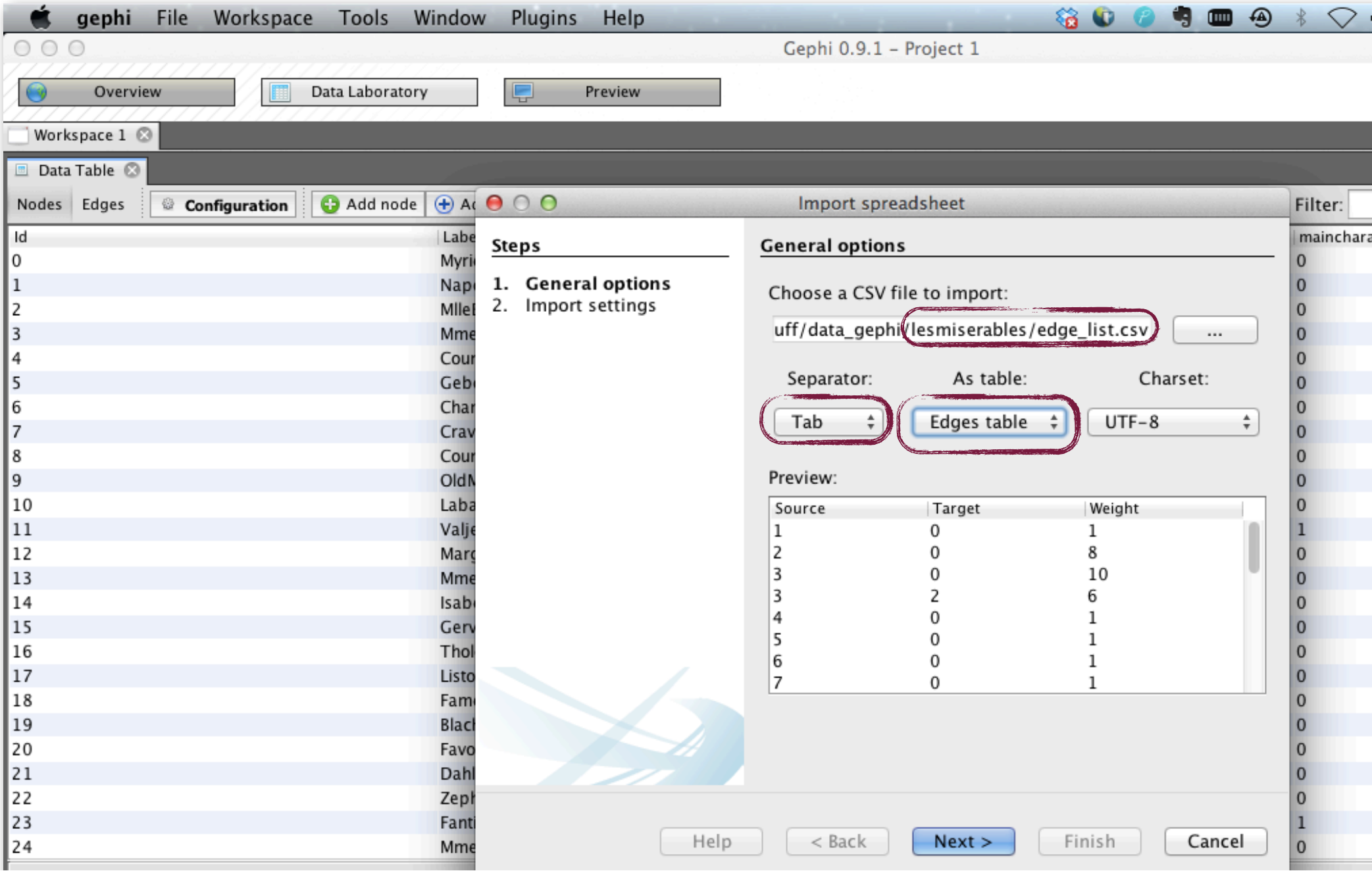


Id	Label	Interval	mainc
0	Myriel		0
1	Napoleon		0
2	MlleBaptistine		0
3	MmeMagloire		0
4	CountessDeLo		0
5	Geborand		0
6	Champtercier		0
7	Cravatte		0
8	Count		0
9	OldMan		0
10	Labarre		0
11	Valjean		1
12	Marguerite		0
13	MmeDeR		0
14	Isabeau		0
15	Gervais		0
16	Tholomyes		0
17	Listolier		0
18	Fameuil		0
19	Blacheville		0
20	Favourite		0
21	Dahlia		0
22	Zephine		0
23	Fantine		1
24	MmeThenardier		0

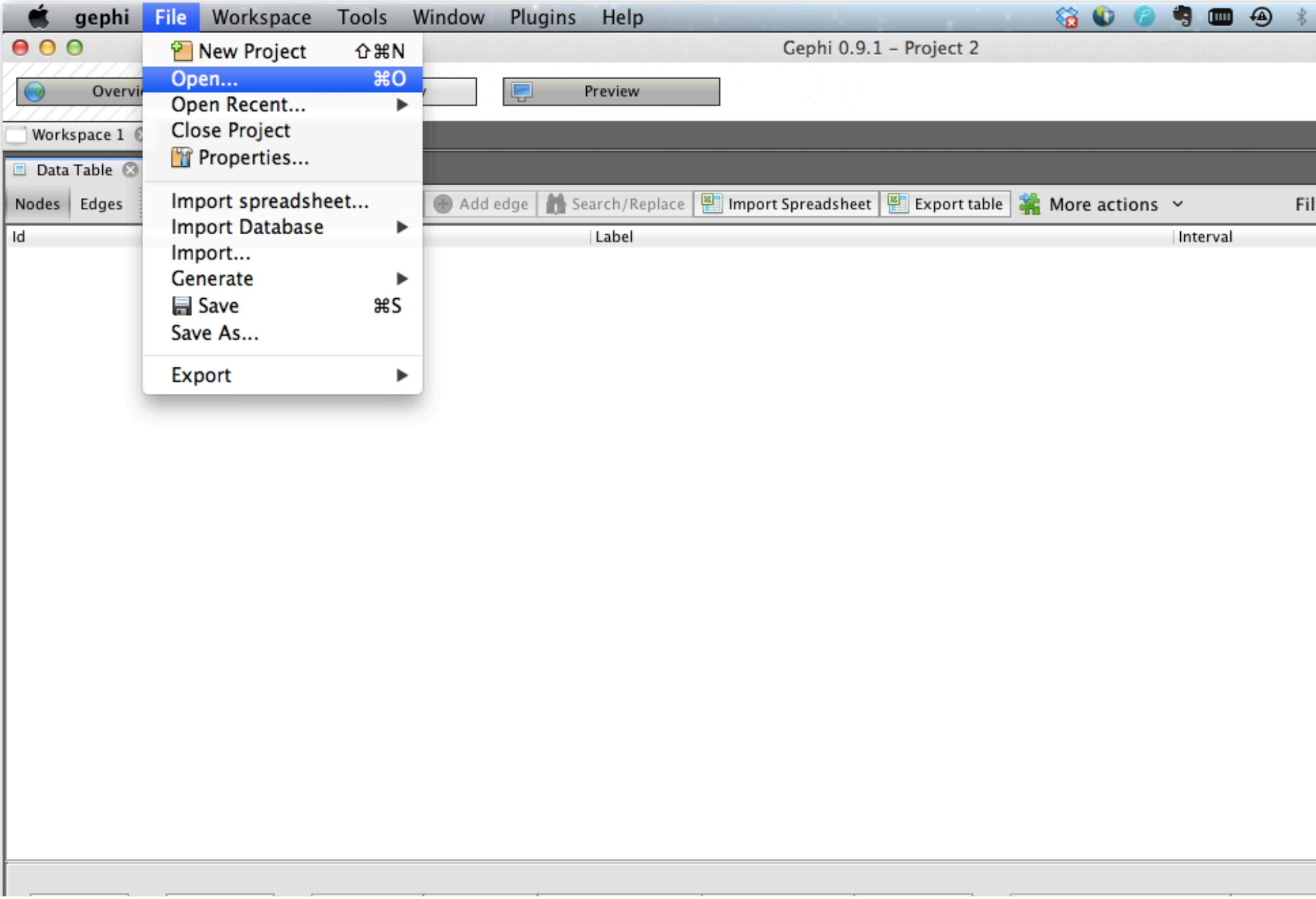
Importing the file -> .csv



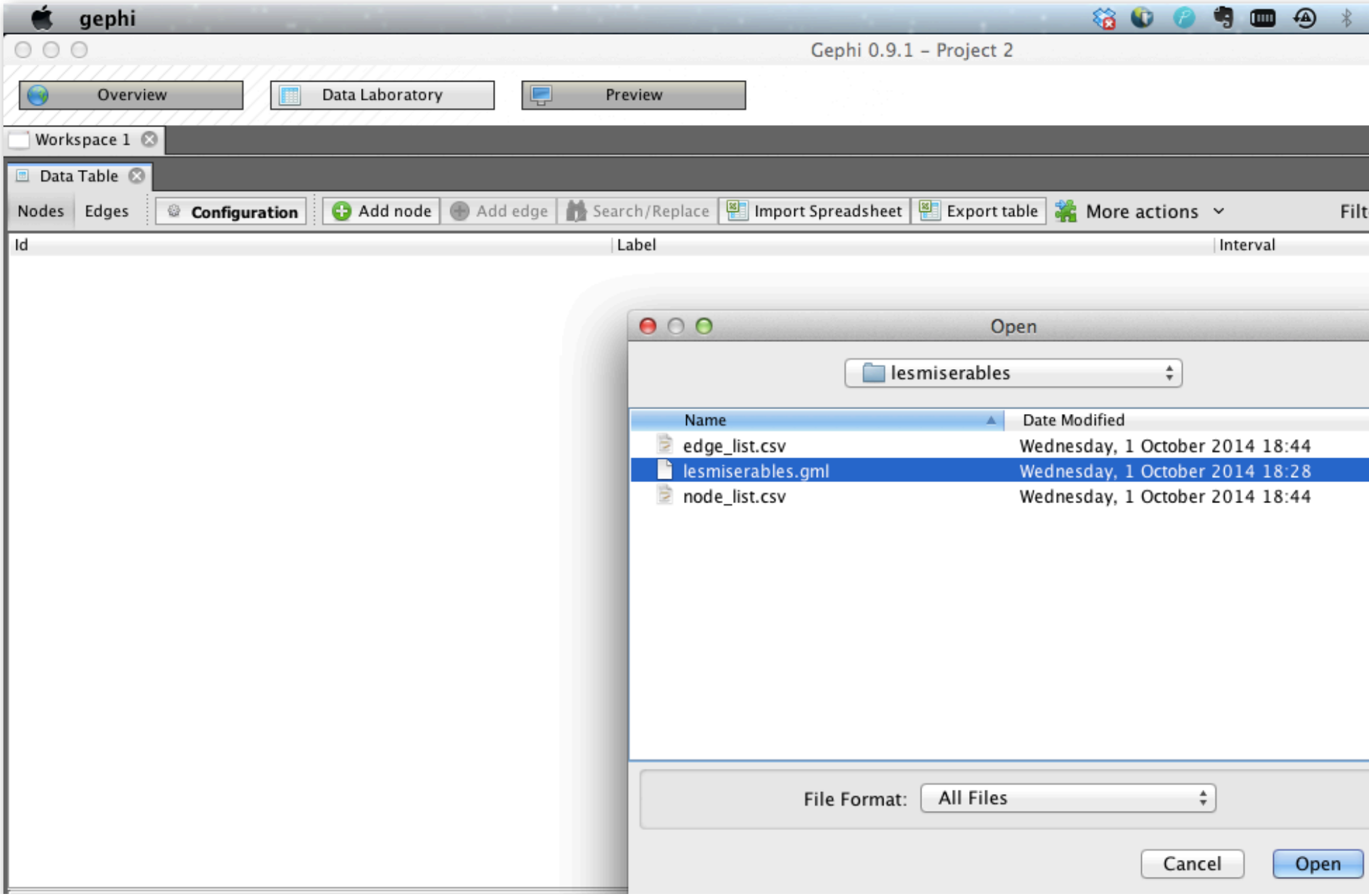
Importing the file -> .csv



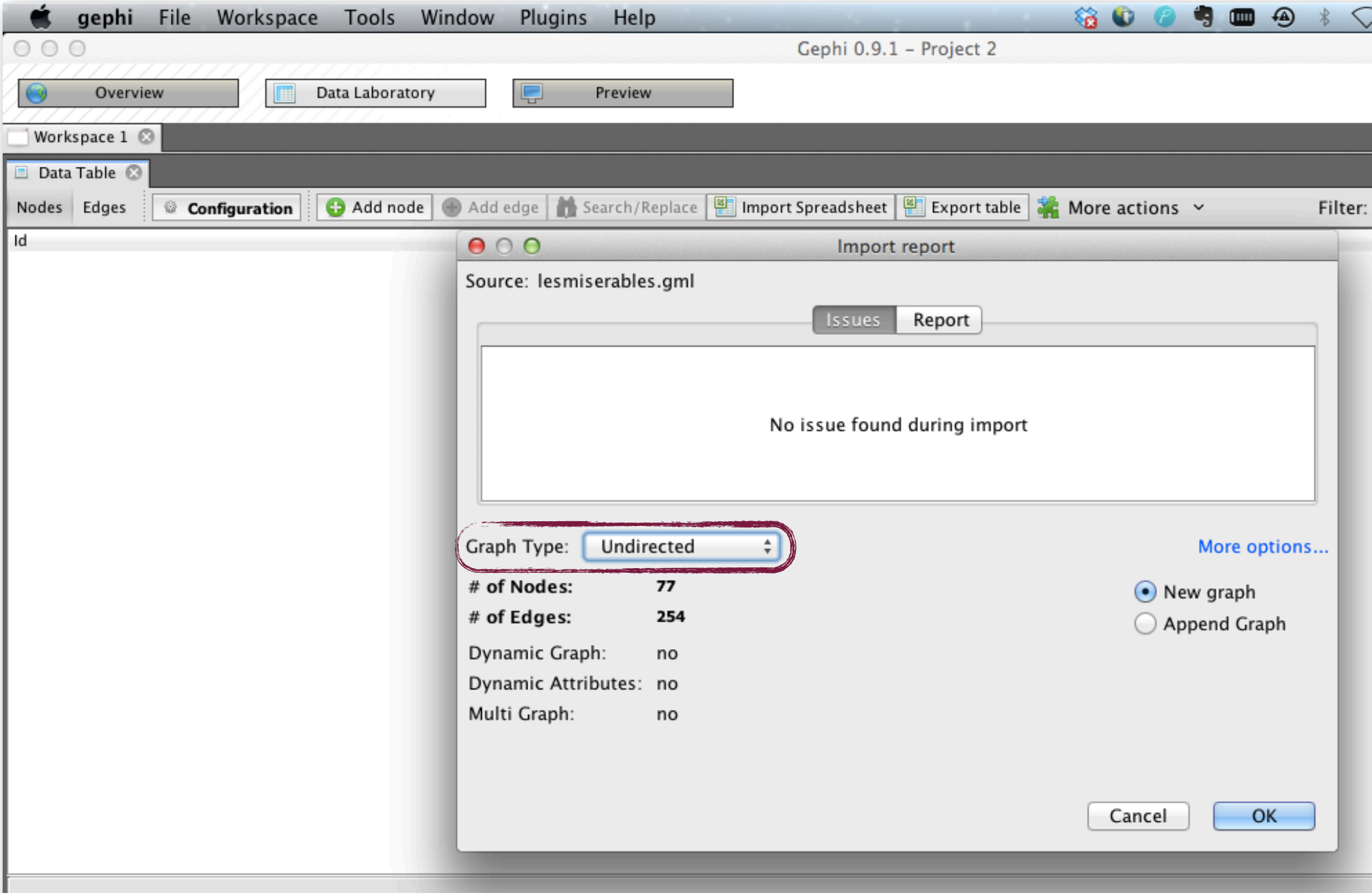
Importing the file -> other formats



Importing the file -> other formats

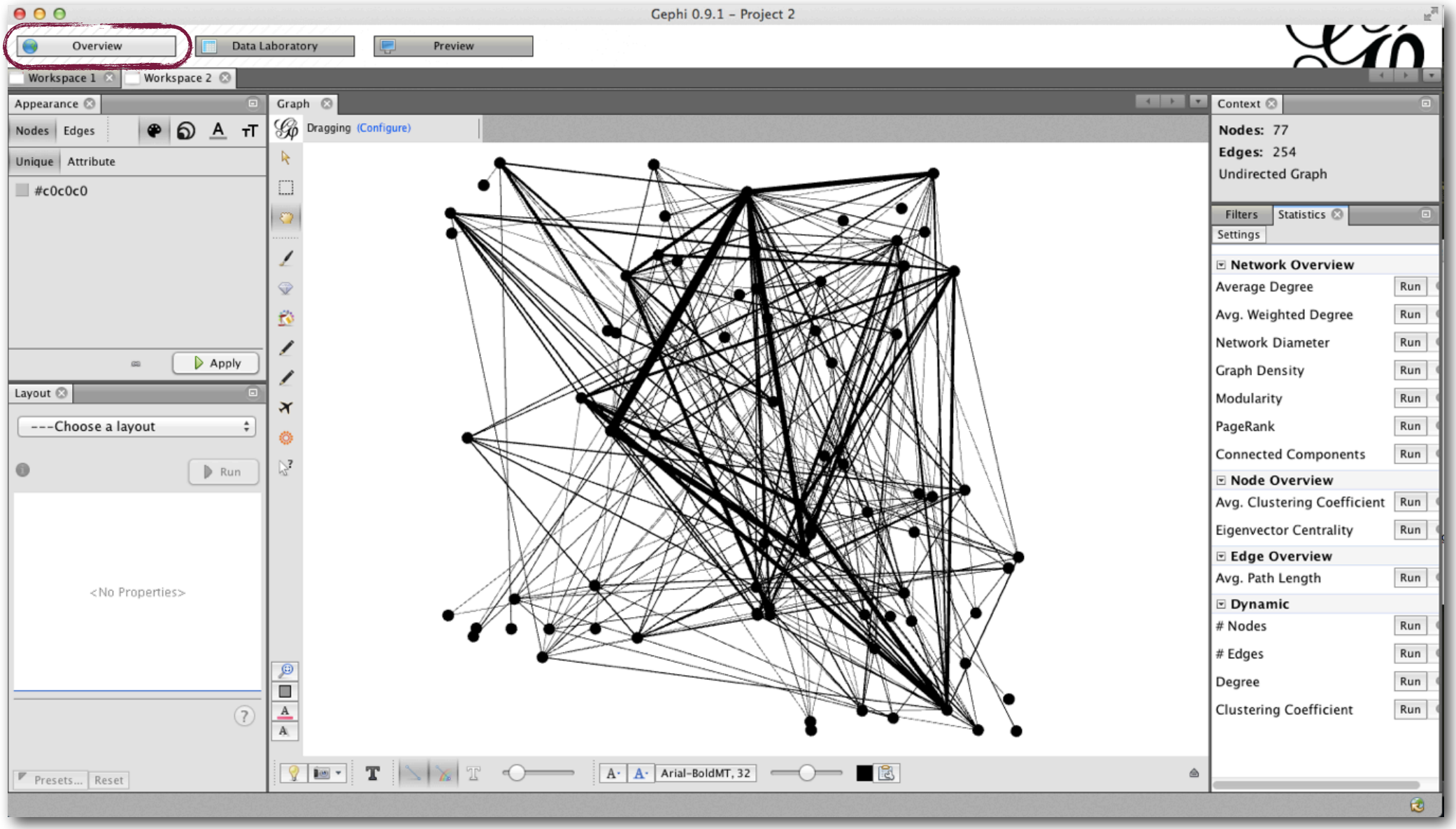


Importing the file -> other formats



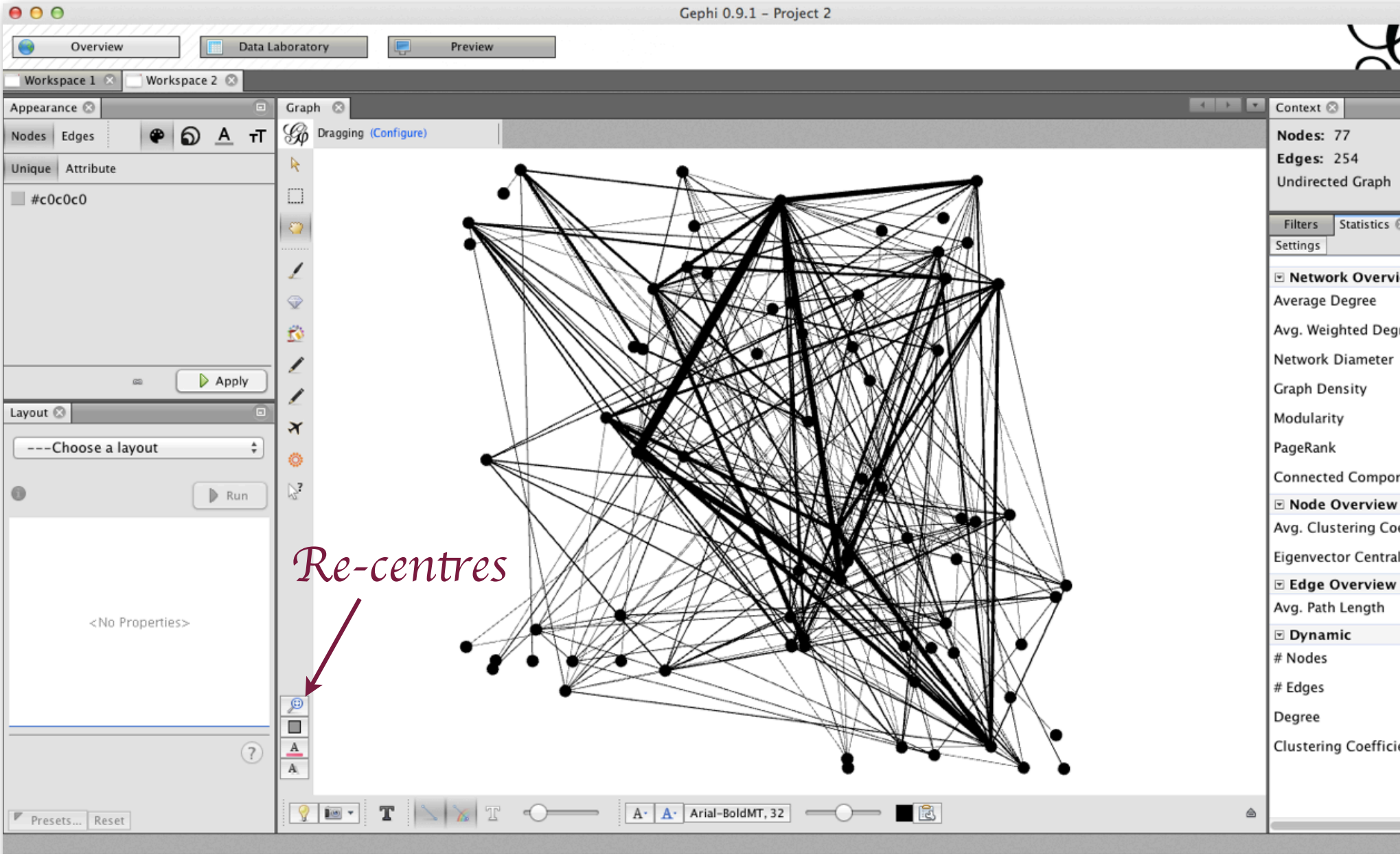
A Misérables visualisation

Click the “Overview” button - opens with a random layout

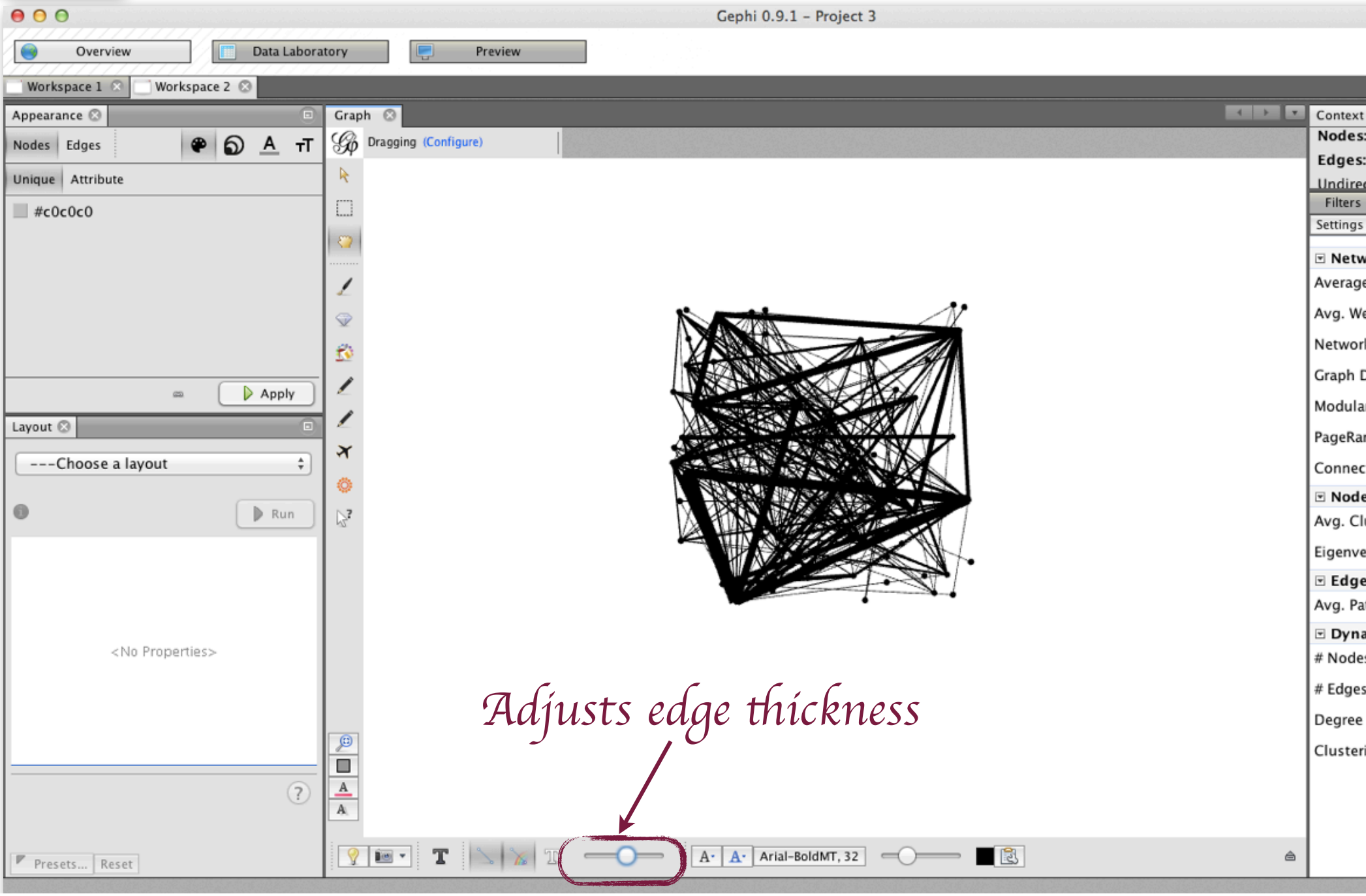


Pan/zoom with your mouse

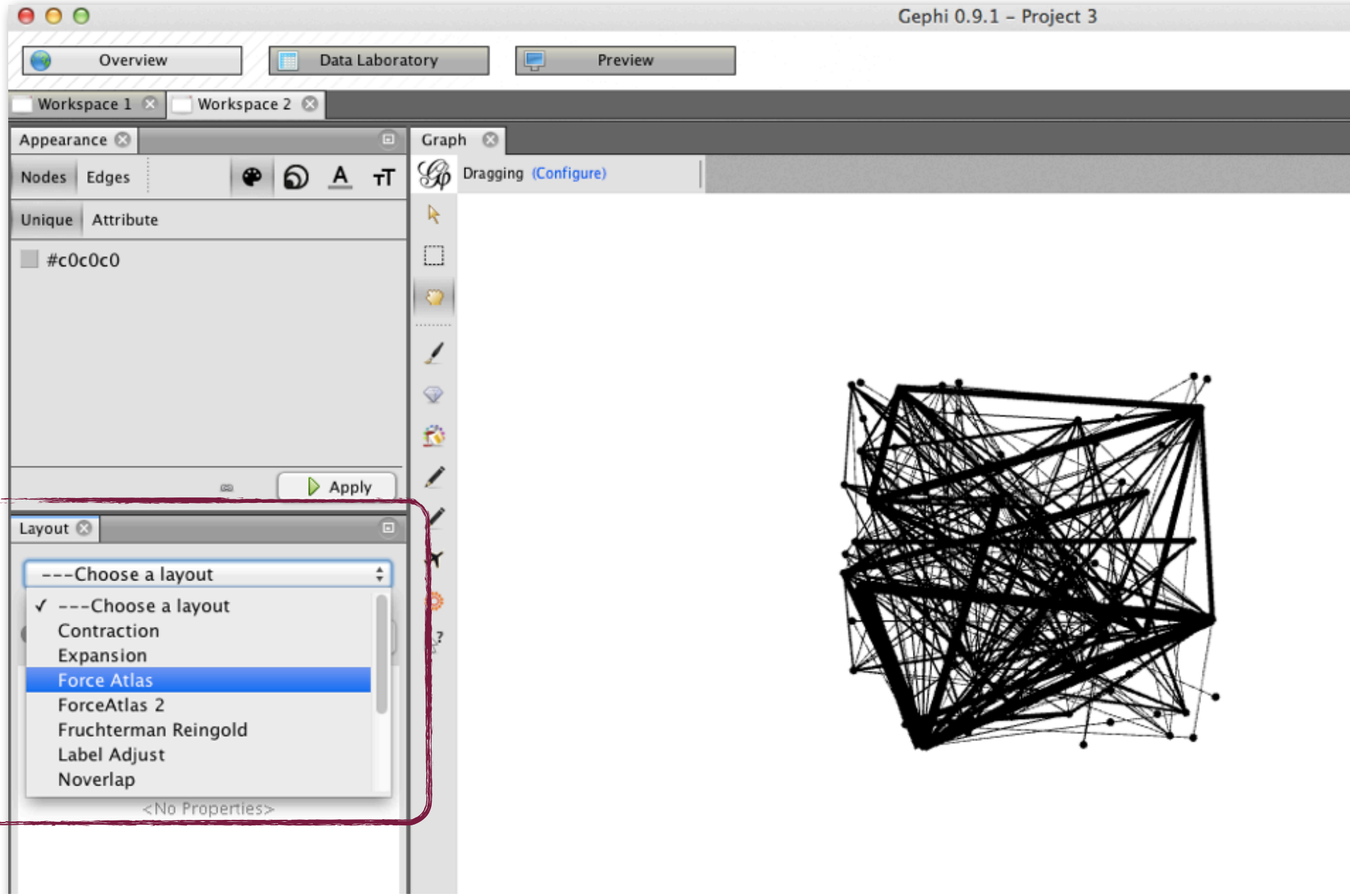
...and try dragging some nodes around.



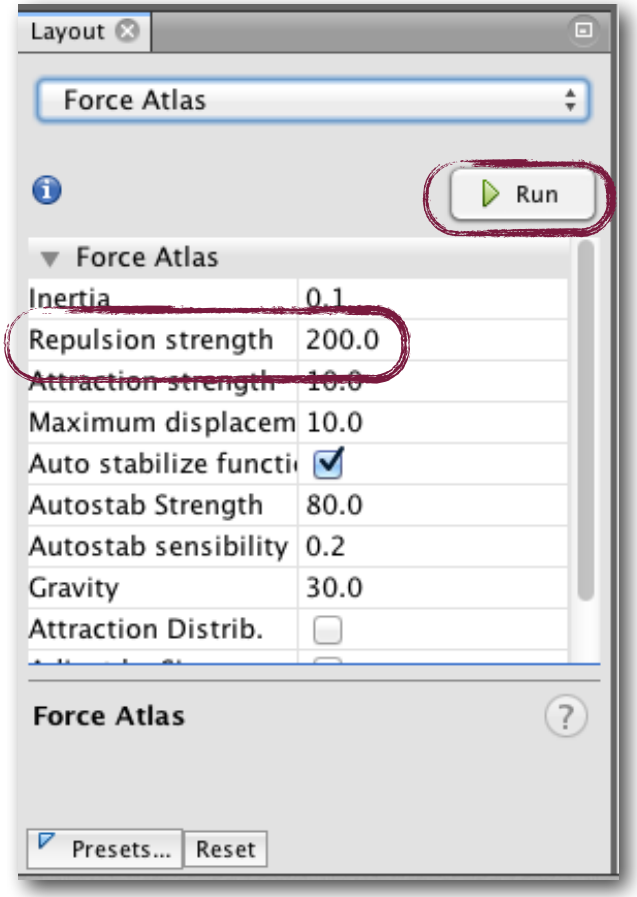
Edge thickness



Force-directed layouts

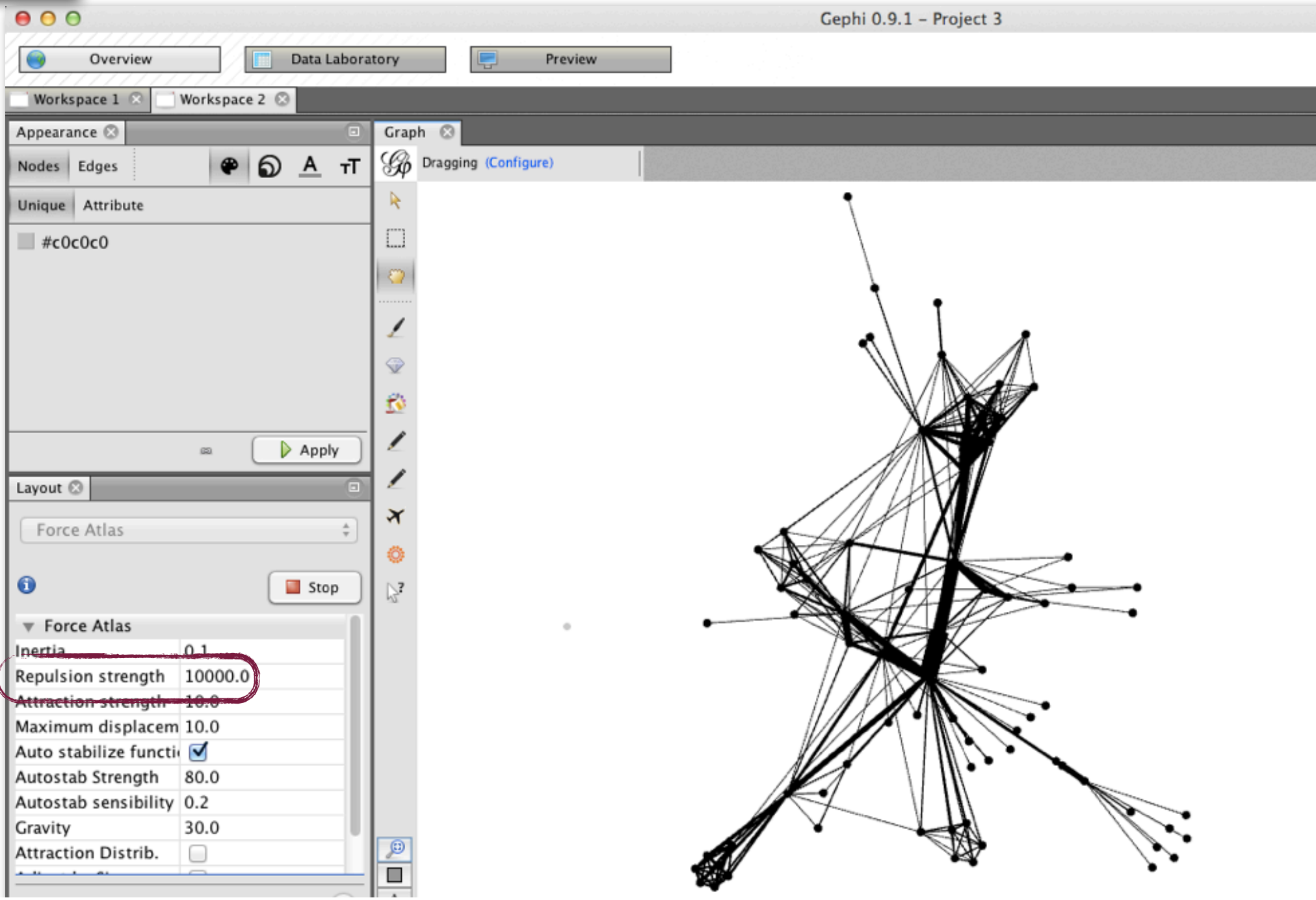


Fine tuning attraction/repulsion



You will need to adjust the parameters...

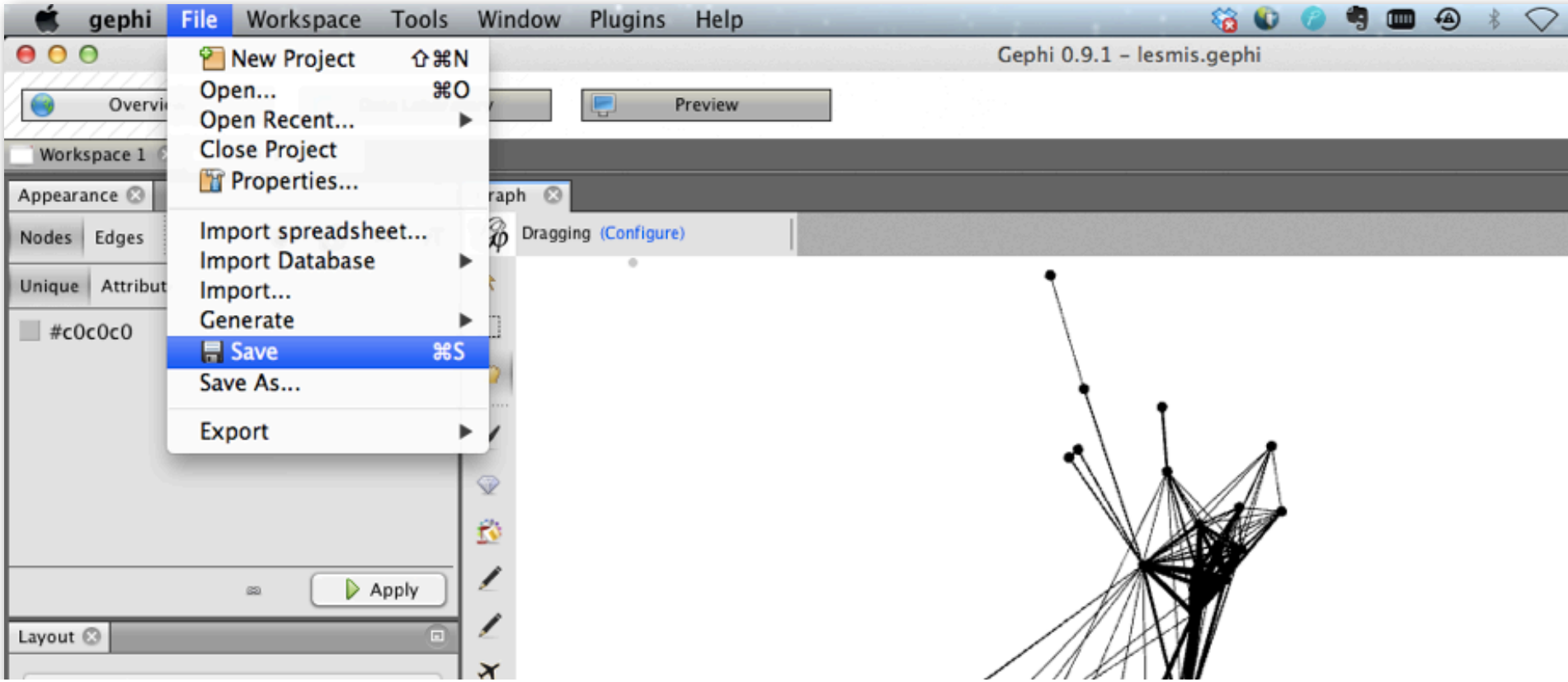
Fine tuning attraction/repulsion



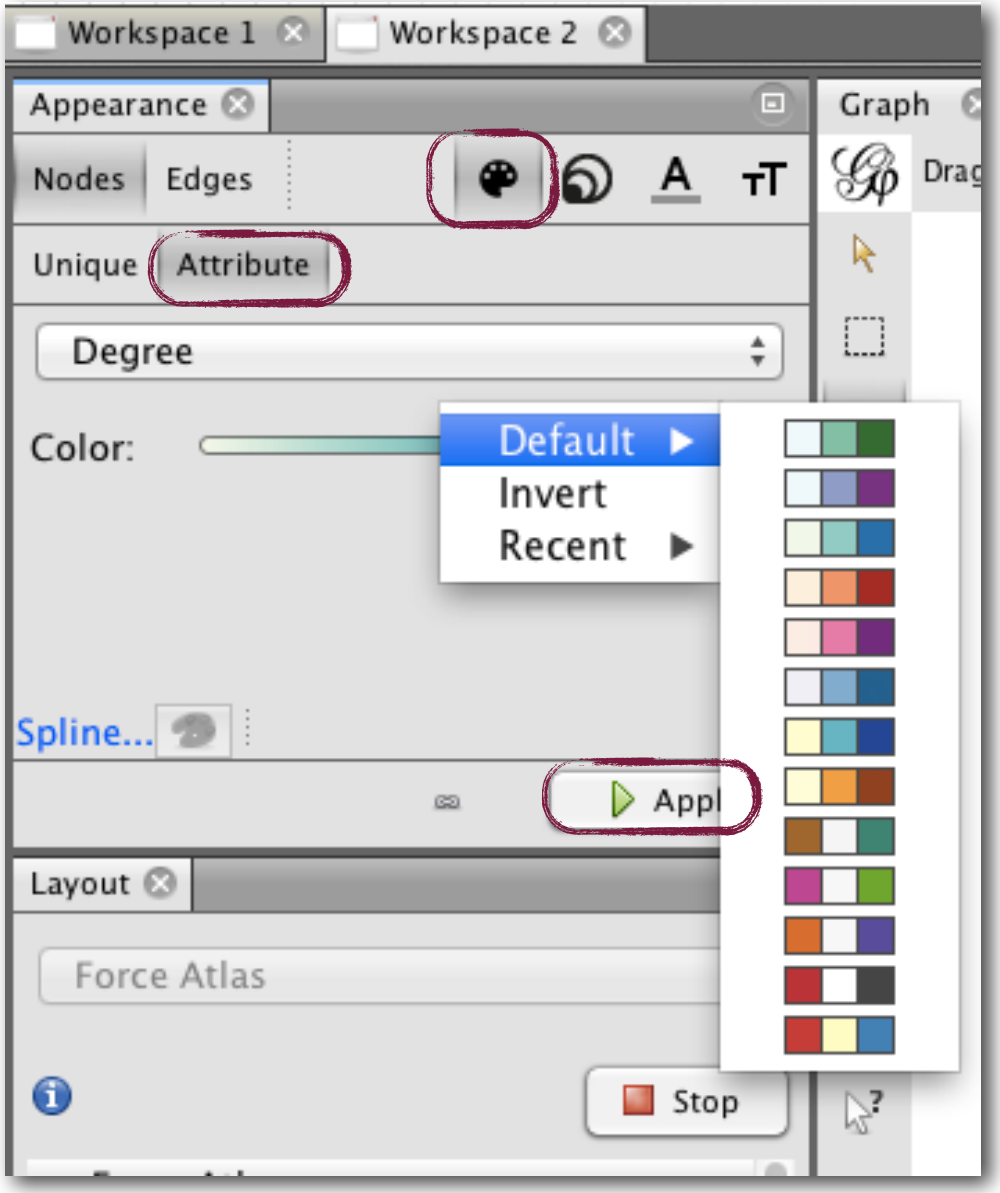
Sidenote: saving

You may have noticed by now that Gephi has one particularly annoying feature... A distinct lack of an undo button.

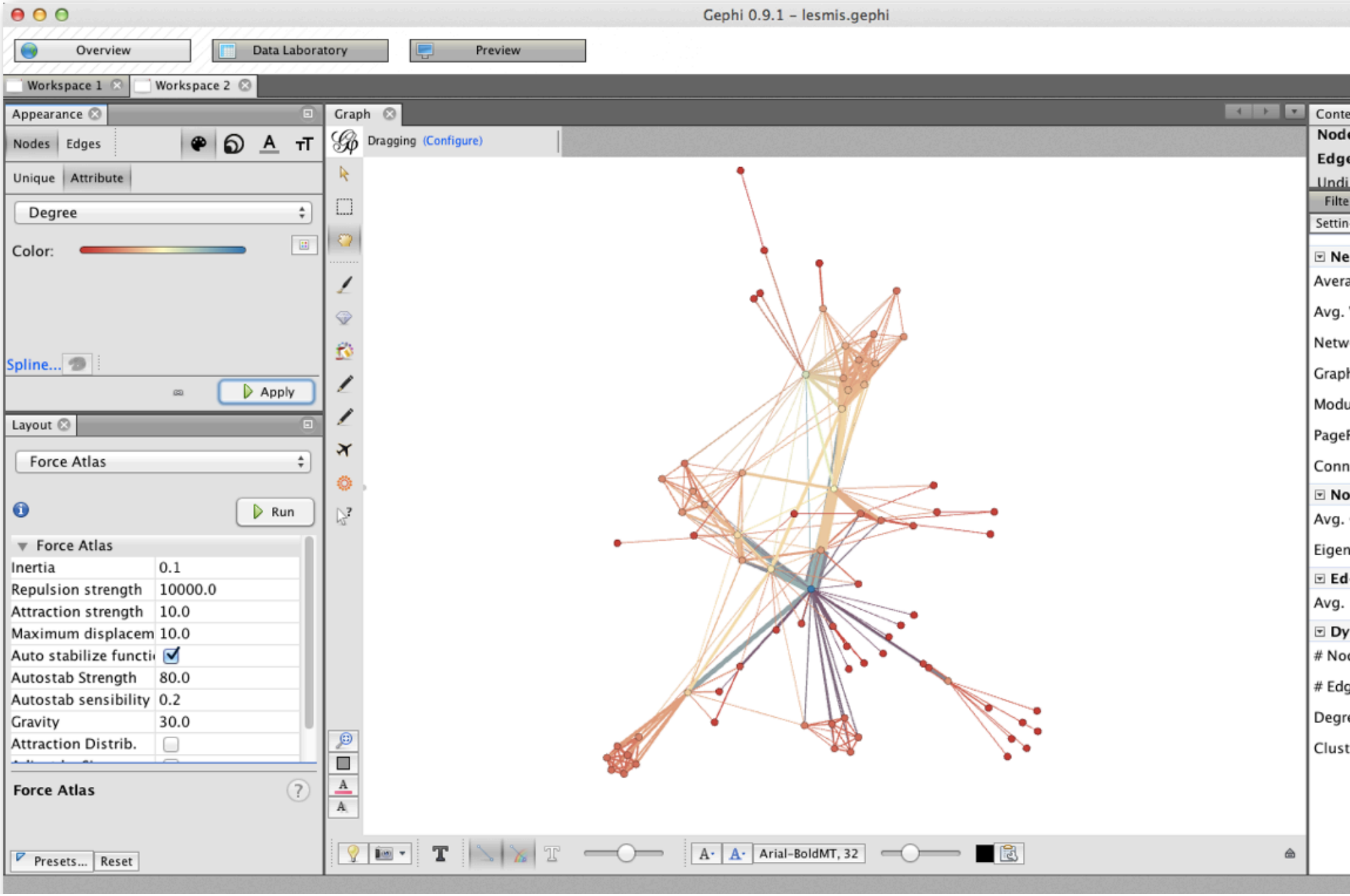
Save your work in progress regularly. Saves as .gephi so you can pick up where you left off.



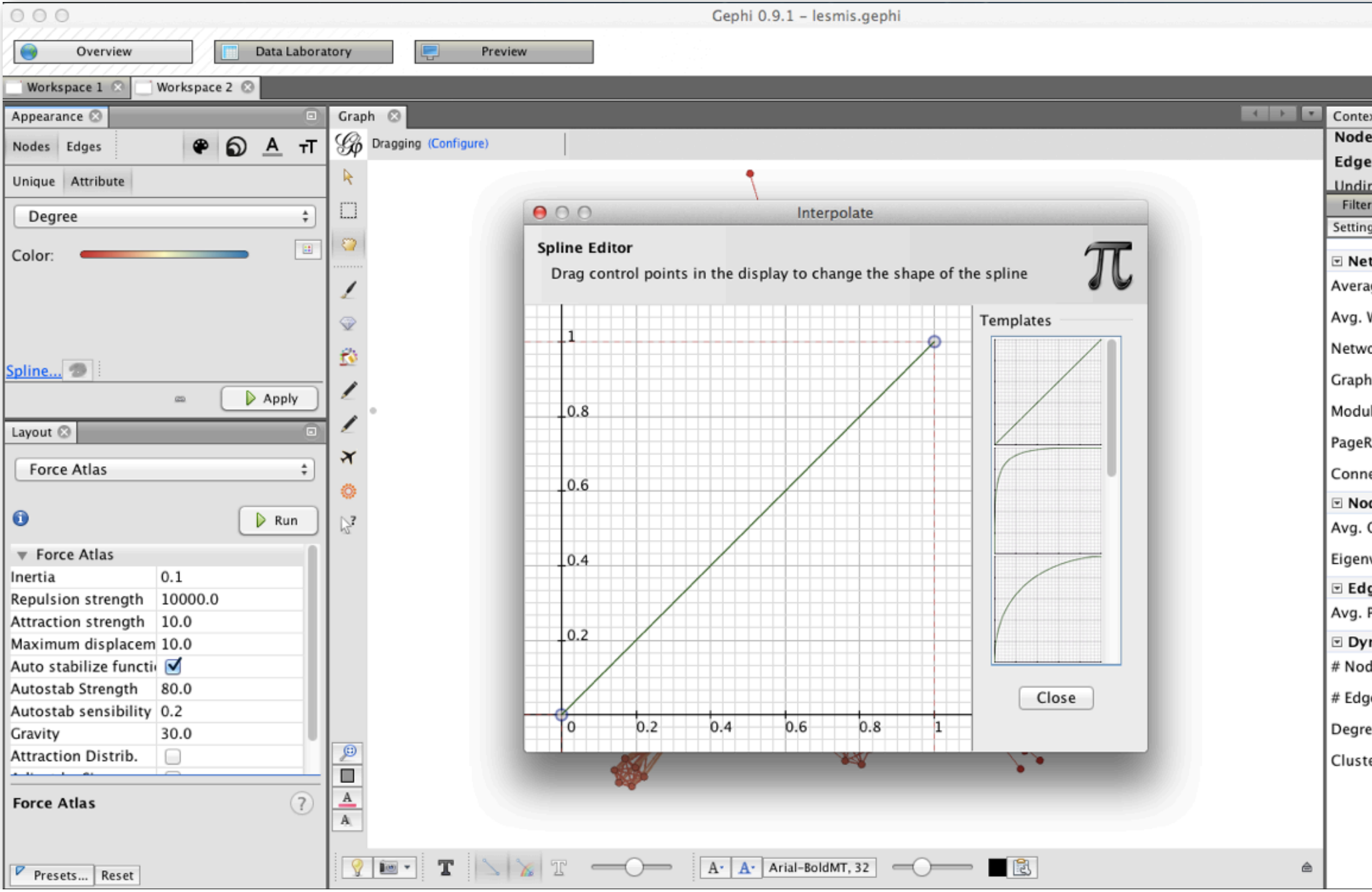
Ranking nodes: colour



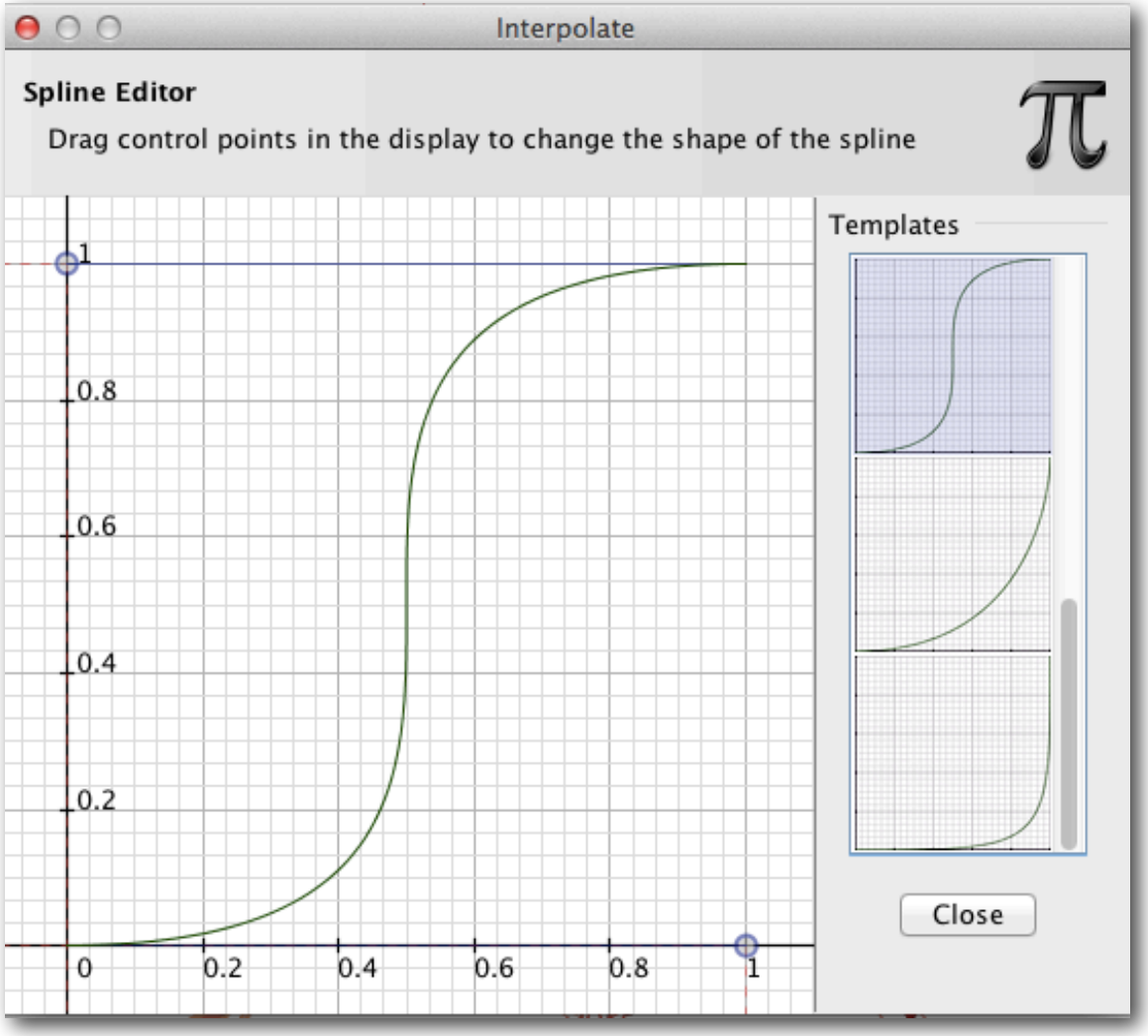
Ranking nodes: colour



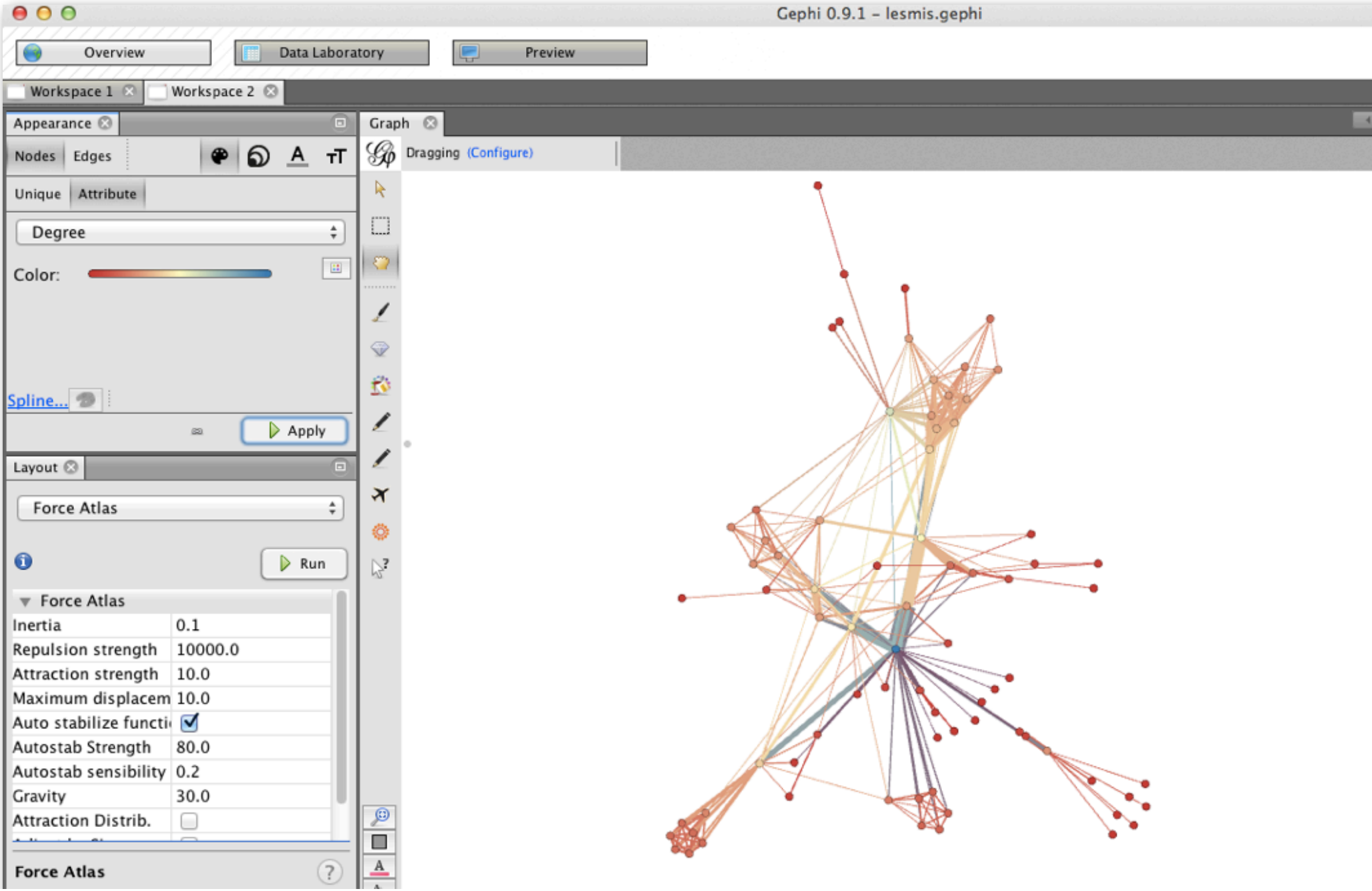
Ranking nodes: colour



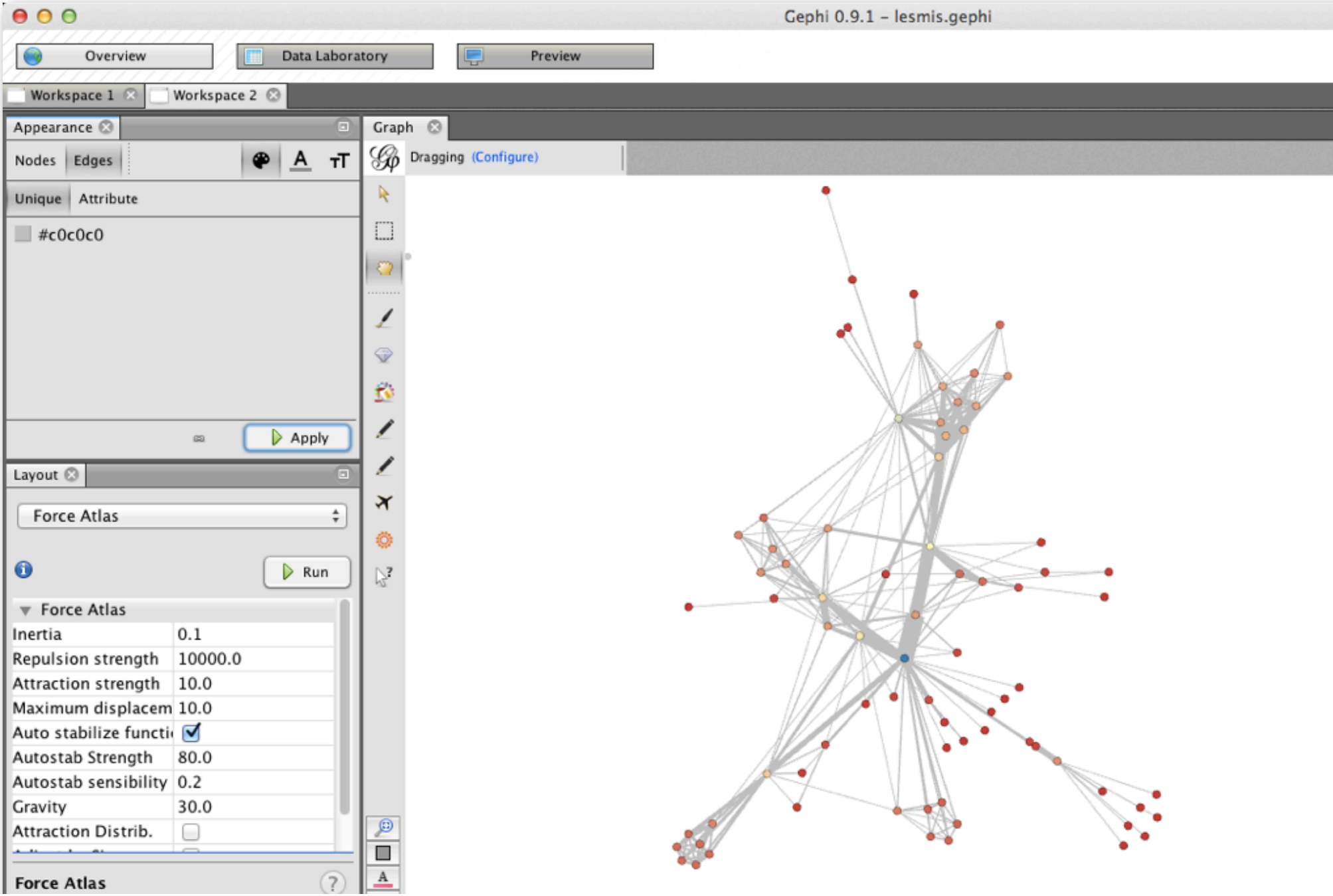
Ranking nodes: colour



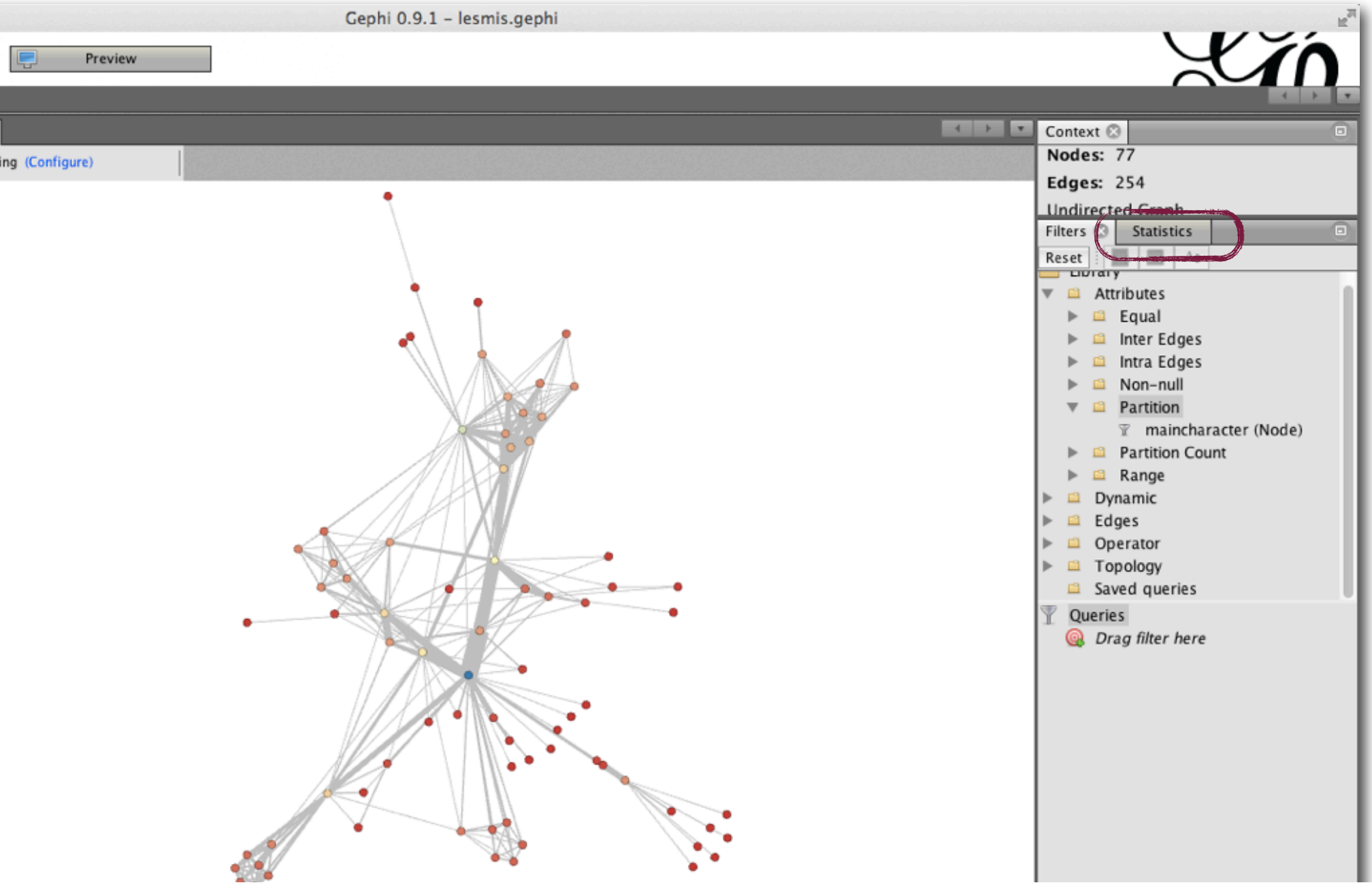
Ranking nodes: colour



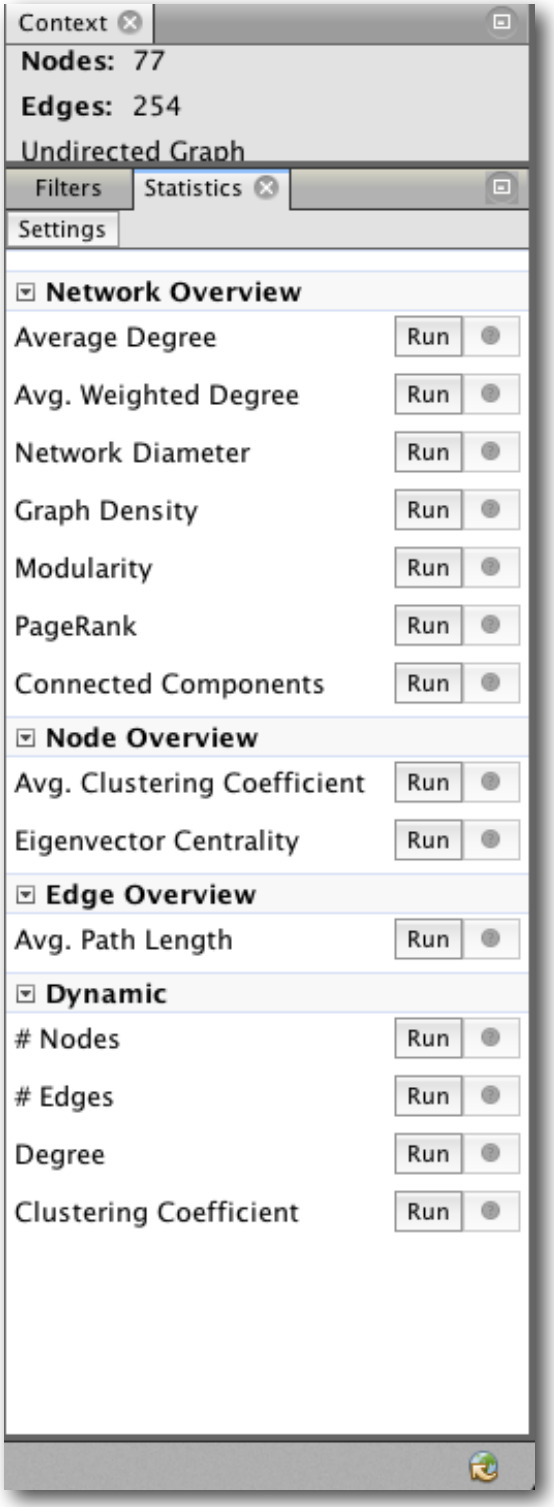
Ranking nodes: colour



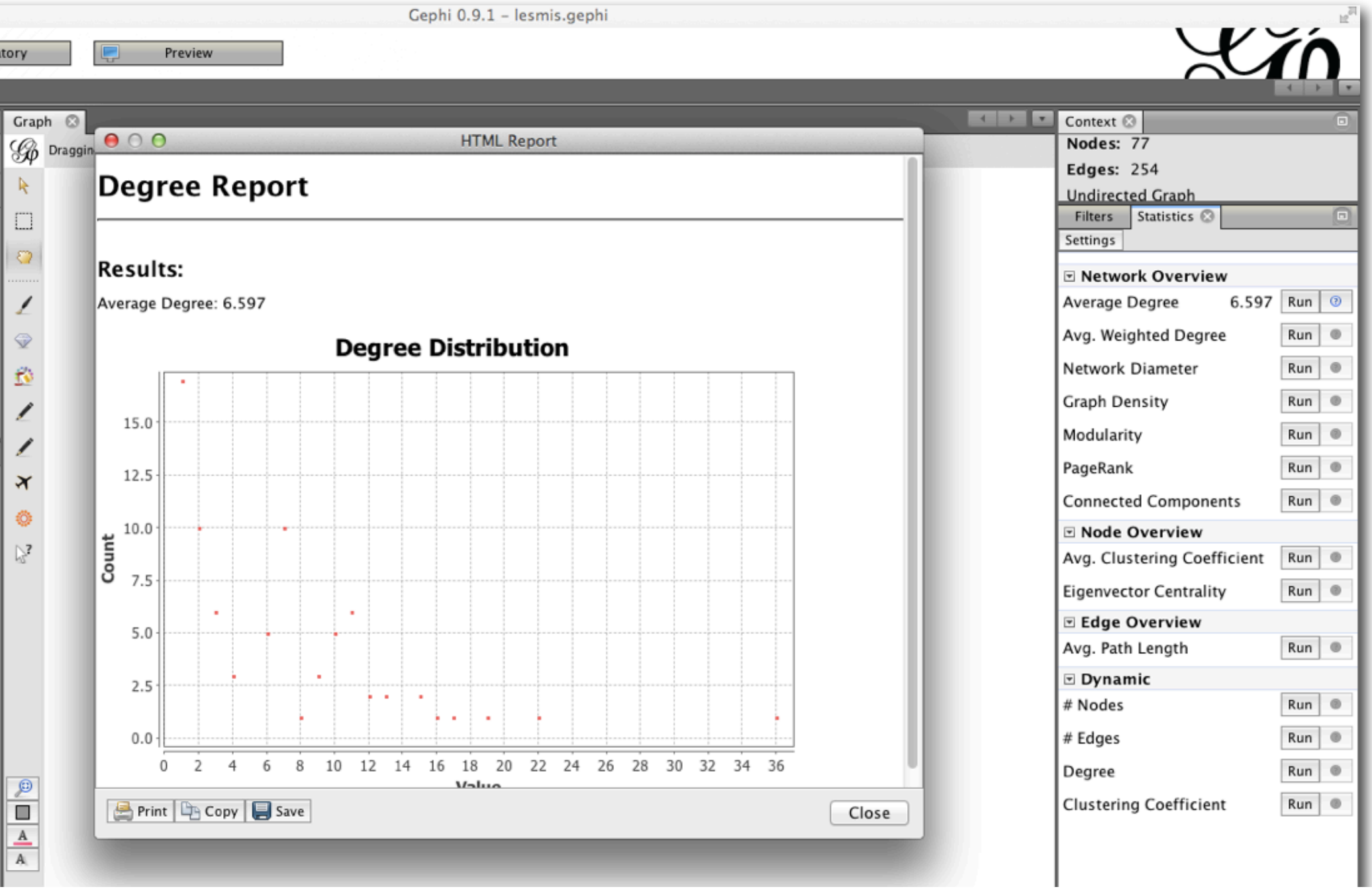
Metrics


Gephi 0.9.1 - lesmis.gephi
Preview
ing (Configure)
Context
Nodes: 77
Edges: 254
Undirected Graph
Filters
Statistics
Reset
Attributes
Equal
Inter Edges
Intra Edges
Non-null
Partition
maincharacter (Node)
Partition Count
Range
Dynamic
Edges
Operator
Topology
Saved queries
Queries
Drag filter here

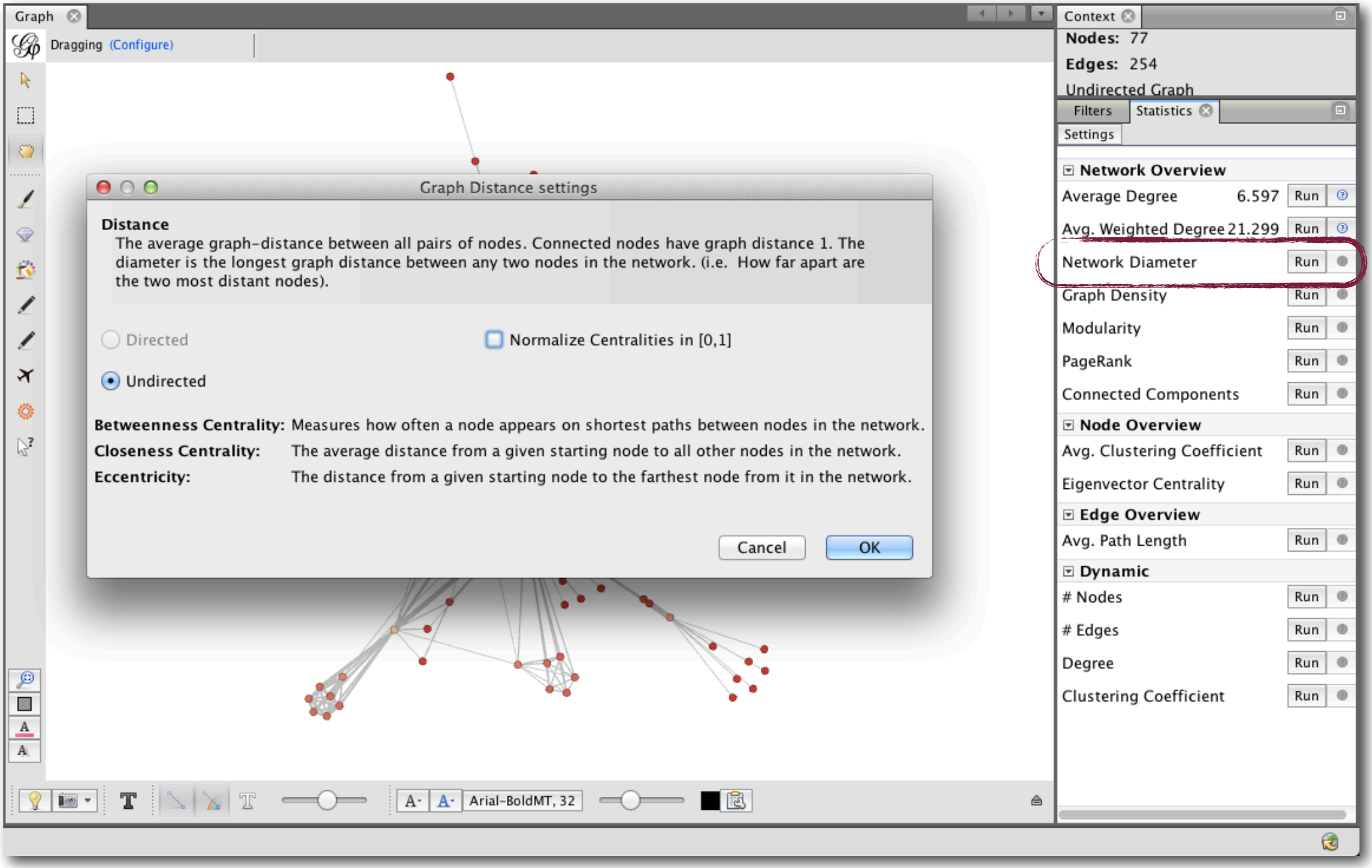
Metrics


Context
Nodes: 77
Edges: 254
Undirected Graph
Filters
Statistics
Settings
Network Overview
Average Degree
Run
Avg. Weighted Degree
Run
Network Diameter
Run
Graph Density
Run
Modularity
Run
PageRank
Run
Connected Components
Run
Node Overview
Avg. Clustering Coefficient
Run
Eigenvector Centrality
Run
Edge Overview
Avg. Path Length
Run
Dynamic
Nodes
Run
Edges
Run
Degree
Run
Clustering Coefficient
Run

Metrics


Gephi 0.9.1 - lesmis.gephi
Preview
Graph
HTML Report
Degree Report
Results:
Average Degree: 6.597
Degree Distribution
Count
Value
Print
Copy
Save
Close
Context
Nodes: 77
Edges: 254
Undirected Graph
Filters
Statistics
Settings
Network Overview
Average Degree 6.597 Run
Avg. Weighted Degree Run
Network Diameter Run
Graph Density Run
Modularity Run
PageRank Run
Connected Components Run
Node Overview
Avg. Clustering Coefficient Run
Eigenvector Centrality Run
Edge Overview
Avg. Path Length Run
Dynamic
Nodes Run
Edges Run
Degree Run
Clustering Coefficient Run

Metrics


Graph
Dragging (Configure)
Graph Distance settings
Distance
The average graph-distance between all pairs of nodes. Connected nodes have graph distance 1. The diameter is the longest graph distance between any two nodes in the network. (i.e. How far apart are the two most distant nodes).
Directed
Normalize Centralities in [0,1]
Undirected
Betweenness Centrality: Measures how often a node appears on shortest paths between nodes in the network.
Closeness Centrality: The average distance from a given starting node to all other nodes in the network.
Eccentricity: The distance from a given starting node to the farthest node from it in the network.
Cancel
OK
Context
Nodes: 77
Edges: 254
Undirected Graph
Filters
Statistics
Settings
Network Overview
Average Degree 6.597 Run
Avg. Weighted Degree 21.299 Run
Network Diameter Run
Graph Density Run
Modularity Run
PageRank Run
Connected Components Run
Node Overview
Avg. Clustering Coefficient Run
Eigenvector Centrality Run
Edge Overview
Avg. Path Length Run
Dynamic
Nodes Run
Edges Run
Degree Run
Clustering Coefficient Run
Arial-BoldMT, 32

Metrics

HTML Report

Graph Distance Report

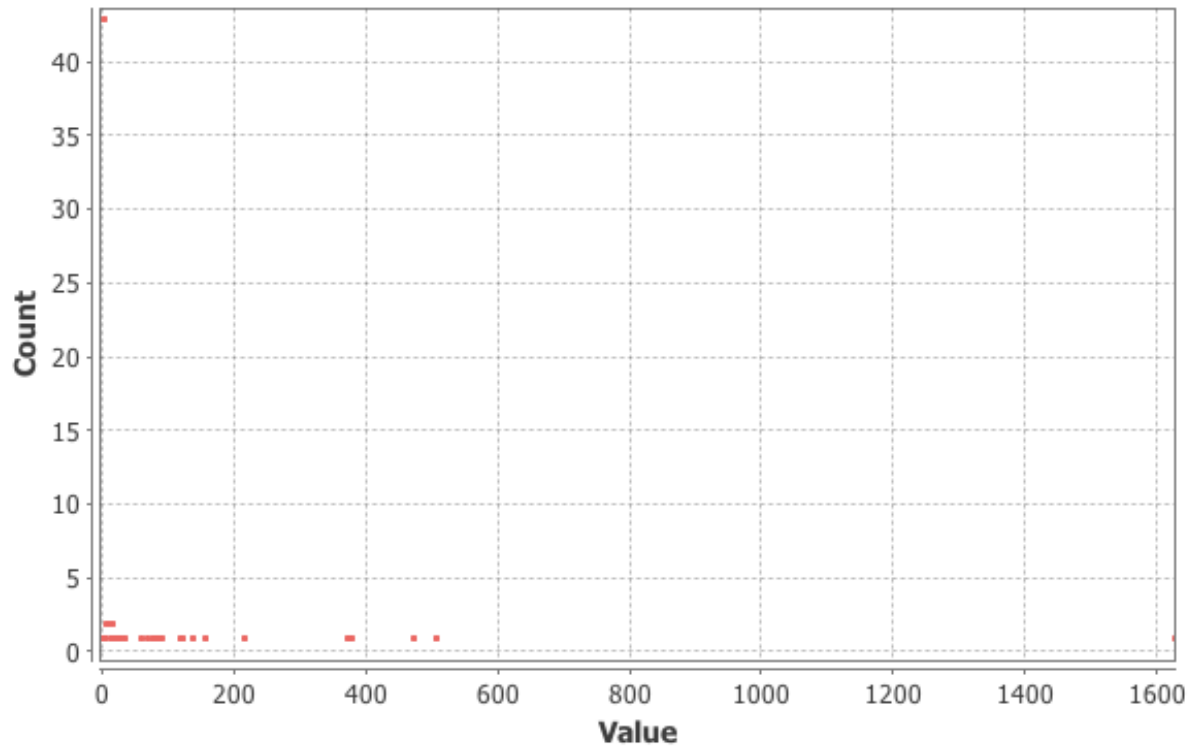
Parameters:

Network Interpretation: undirected

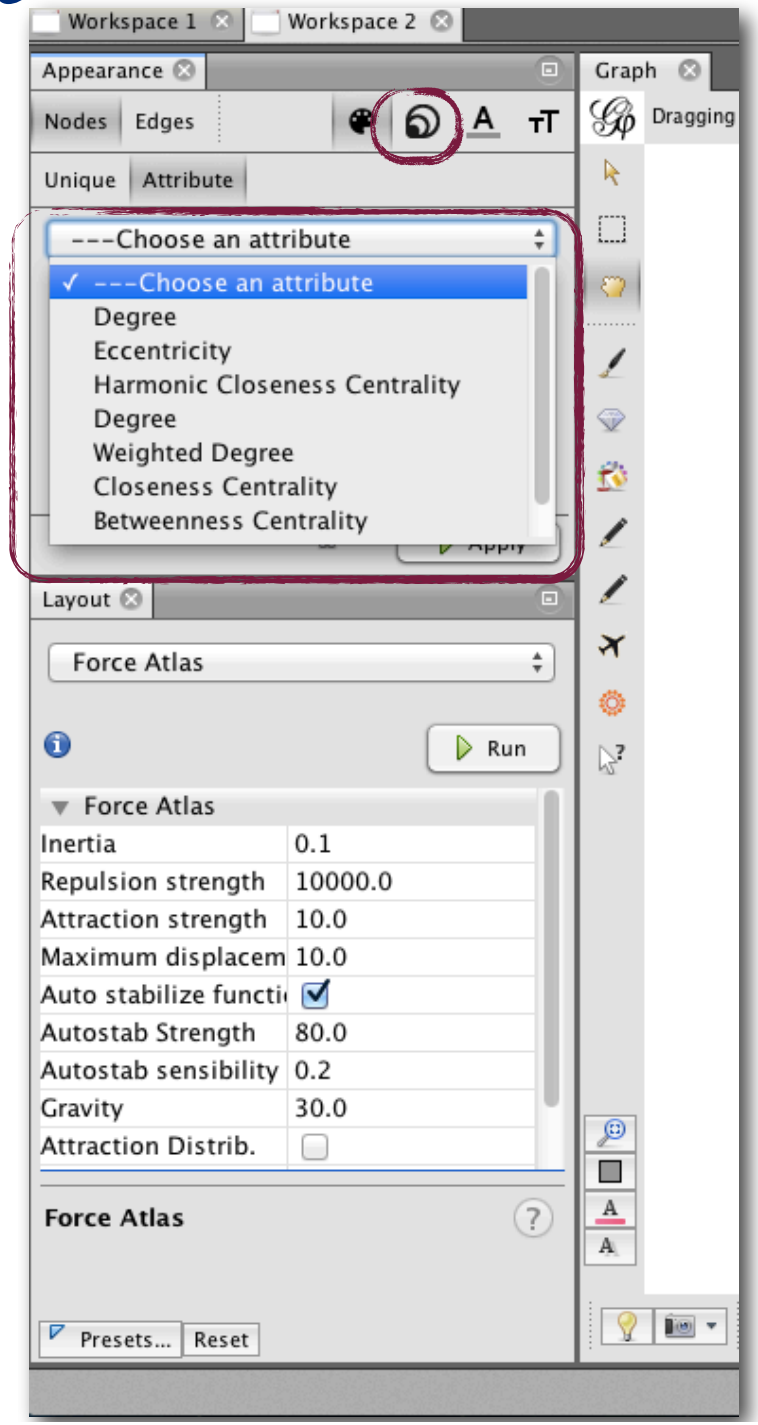
Results:

Diameter: 5
Radius: 3
Average Path length: 2.6411483253588517

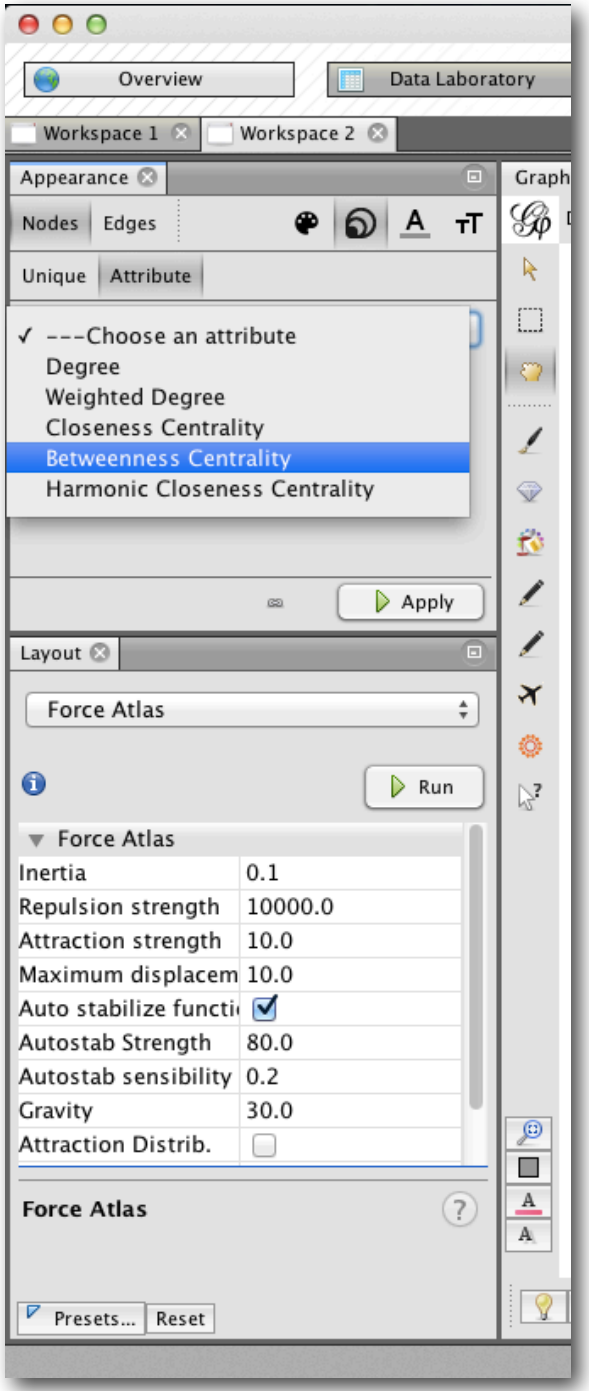
Betweenness Centrality Distribution



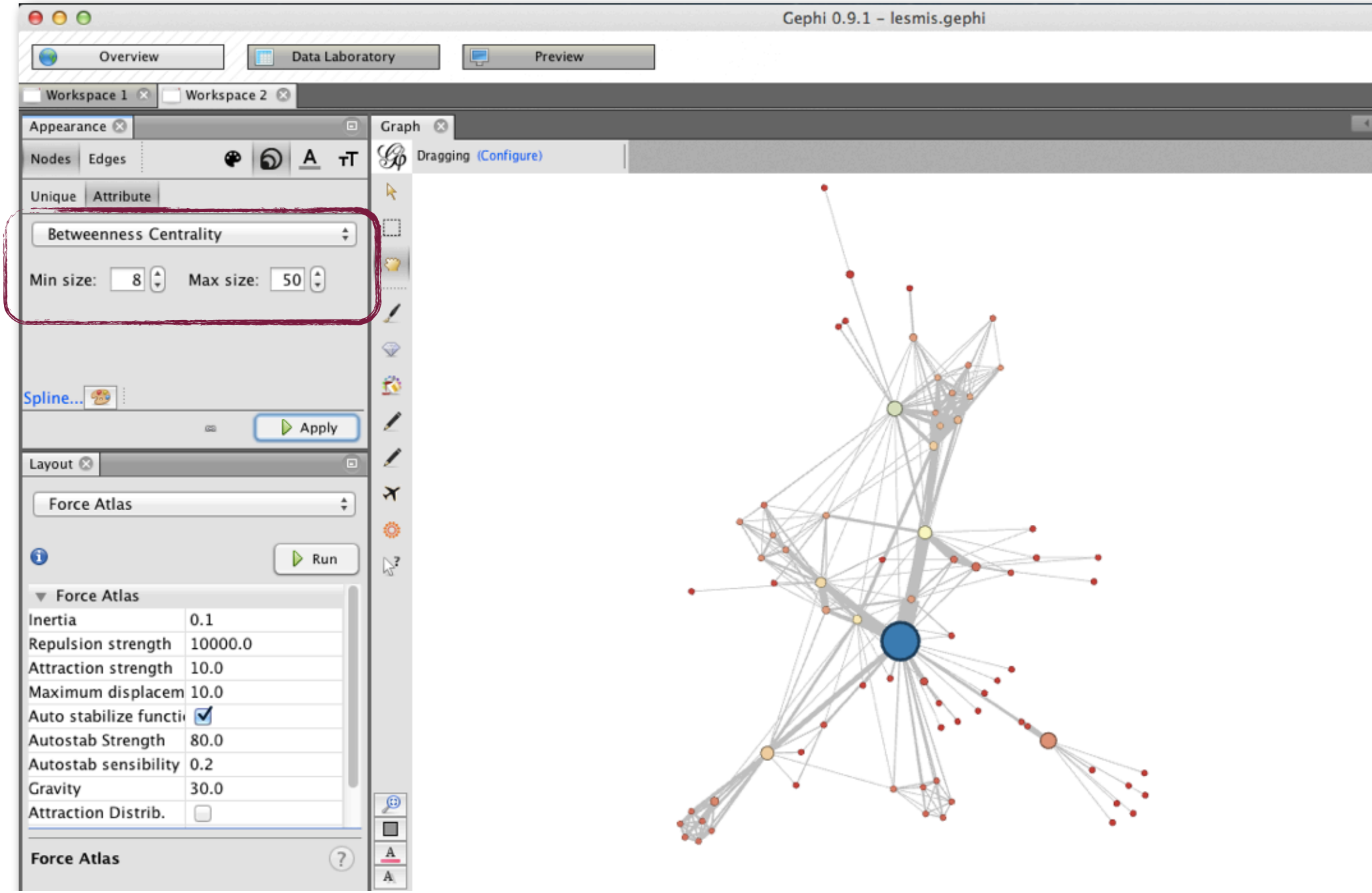
Node ranking: size



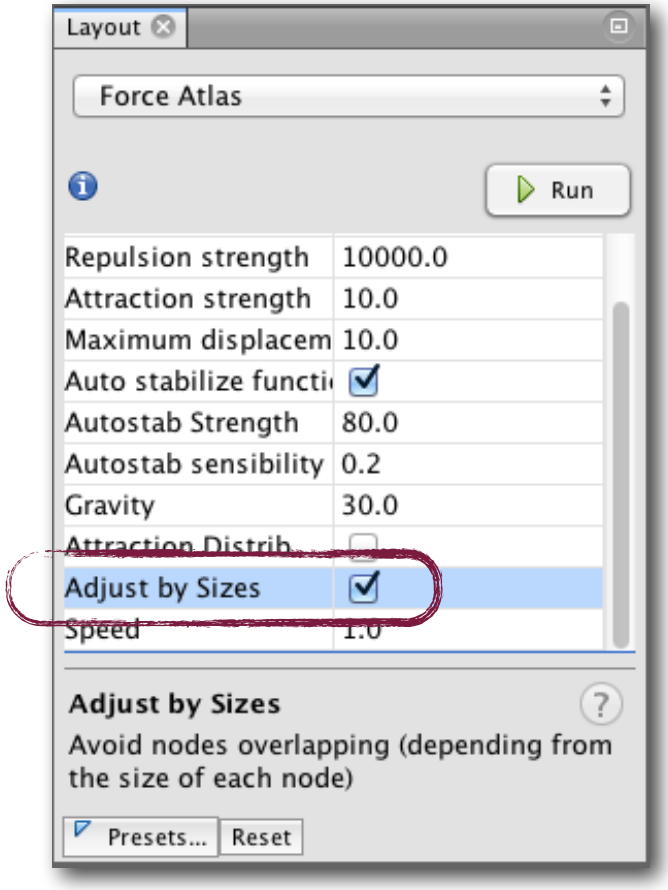
Node ranking: size



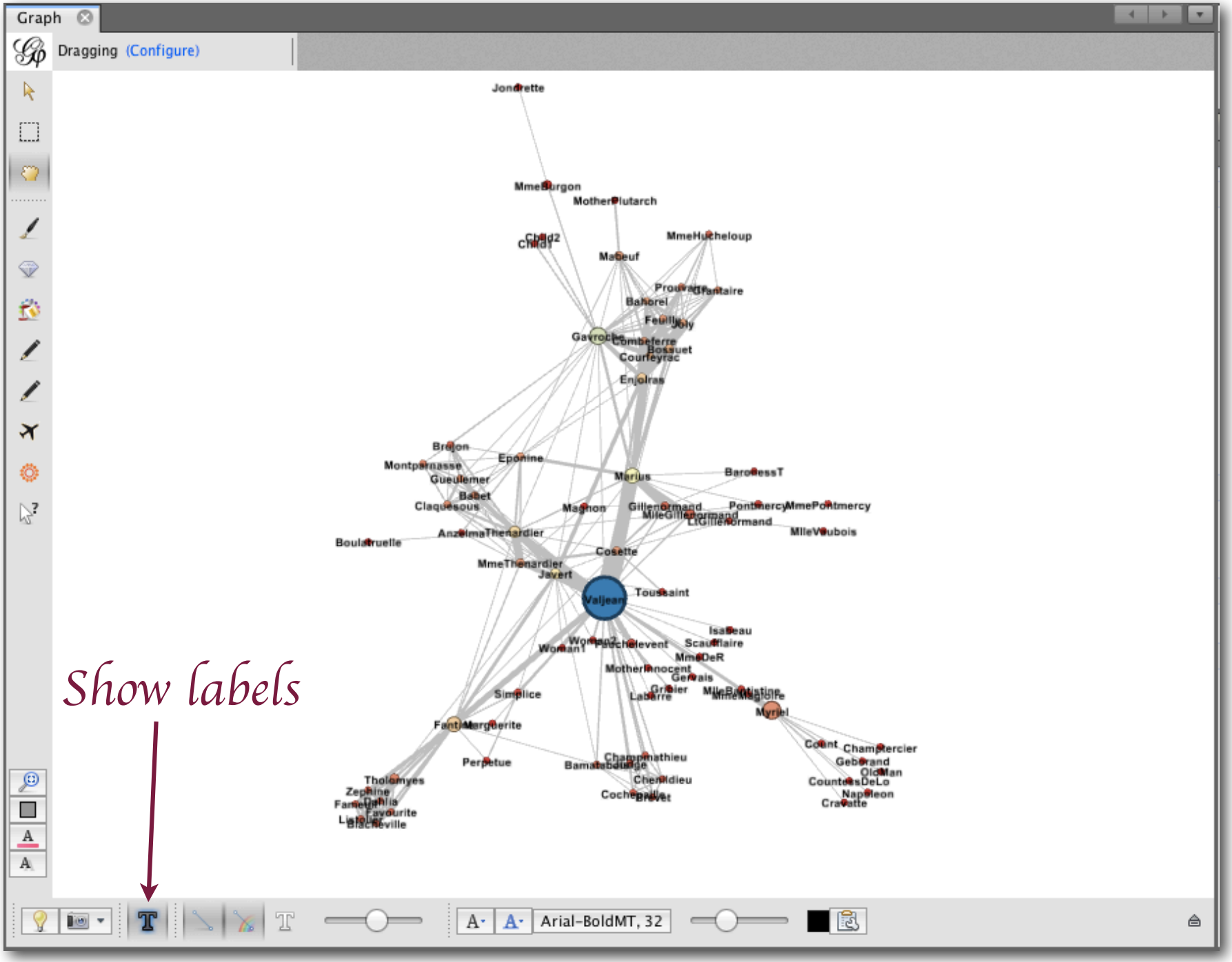
Node ranking: size



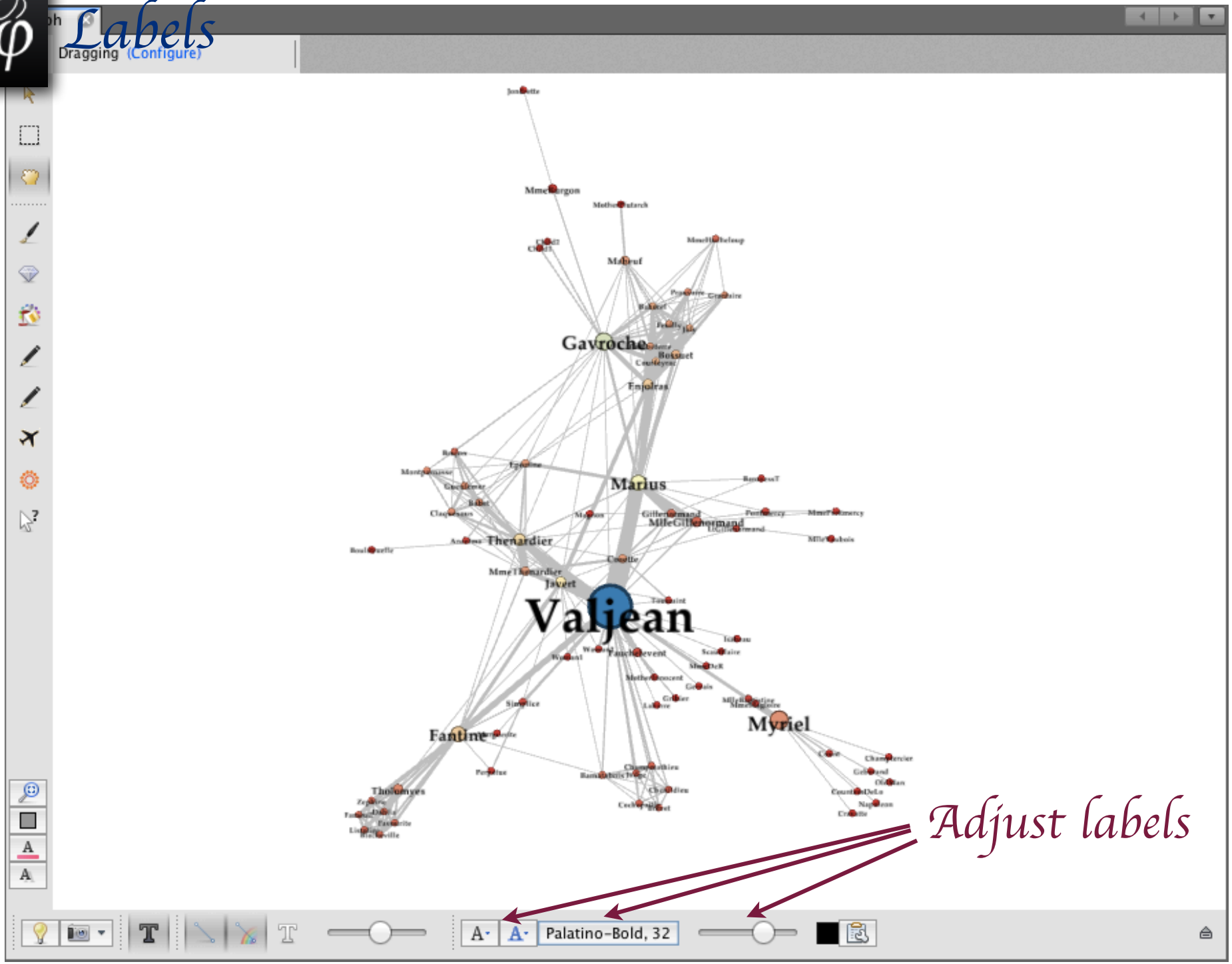
Adjust layout



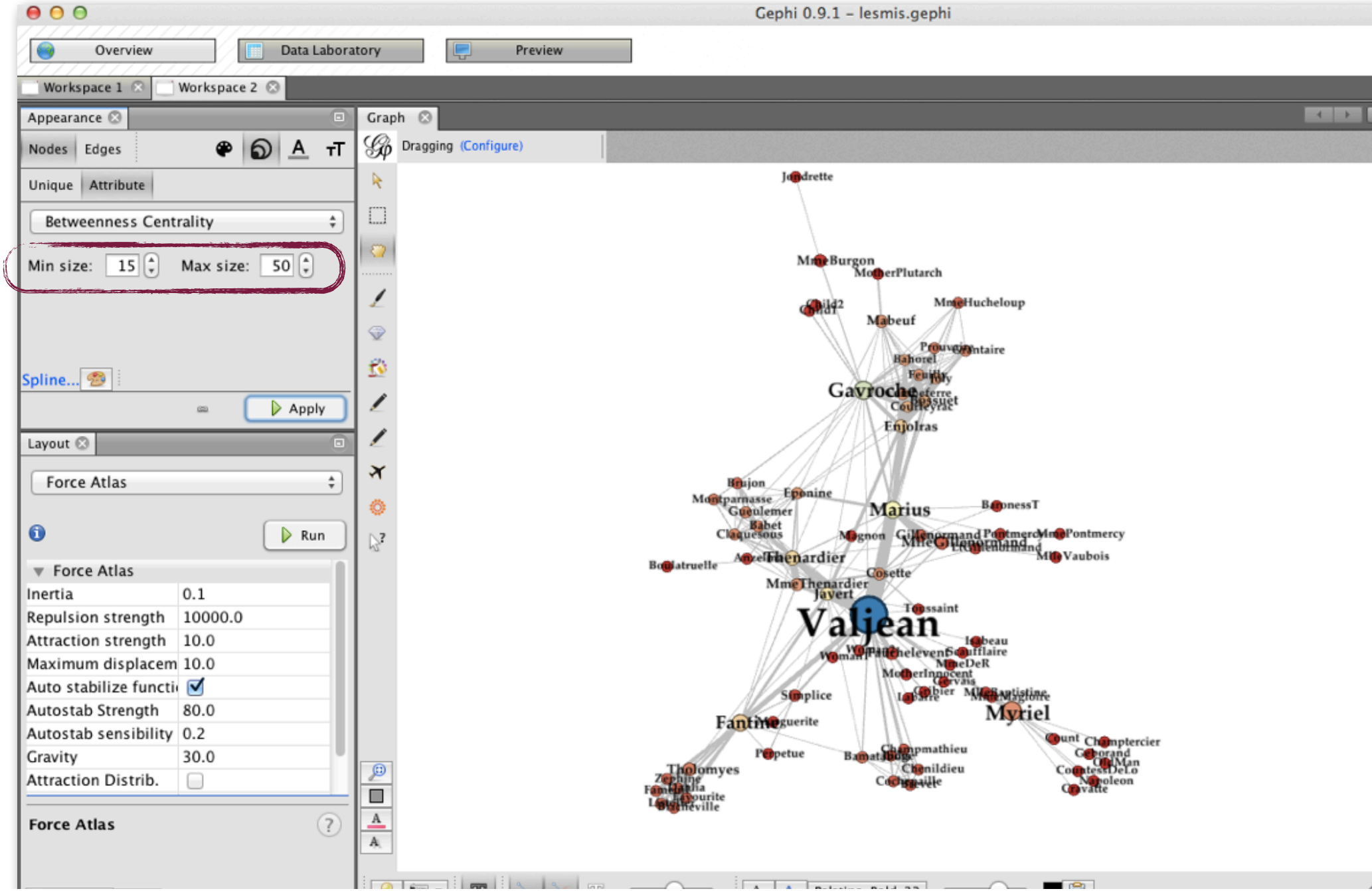
Labels


Graph
Dragging (Configure)
Jondrette
MmeBurgon
MotherPlutarch
MmeHucheloup
Child2
Child1
Mabeuf
Bahorel
Feuilly
Joly
Gavroche
Combeferre
Bossuet
Courfeyrac
Enjolras
Brujon
Montparnasse
Eponine
Gueulemer
Babet
Claquesous
Marius
BaronessT
Magnon
GillenormandMlle
PontmercyMmePontmercy
MlleVaubois
AnzelmaThenardier
Boulatruelle
Cosette
MmeThenardier
Javert
Valjean
Toussaint
Isabeau
Scaufflaire
MmeDeR
MotherInnocent
Gervais
Simplice
Labarre
Myriel
Count
Champtercier
Geborand
OldMan
CountessDeLo
Napoleon
Cravatte
Perpetue
Champmathieu
Cochepaille
Brevet
Tholomyes
Zephine
Dahlia
Favourite
Blacheville
Show labels
Arial-BoldMT, 32

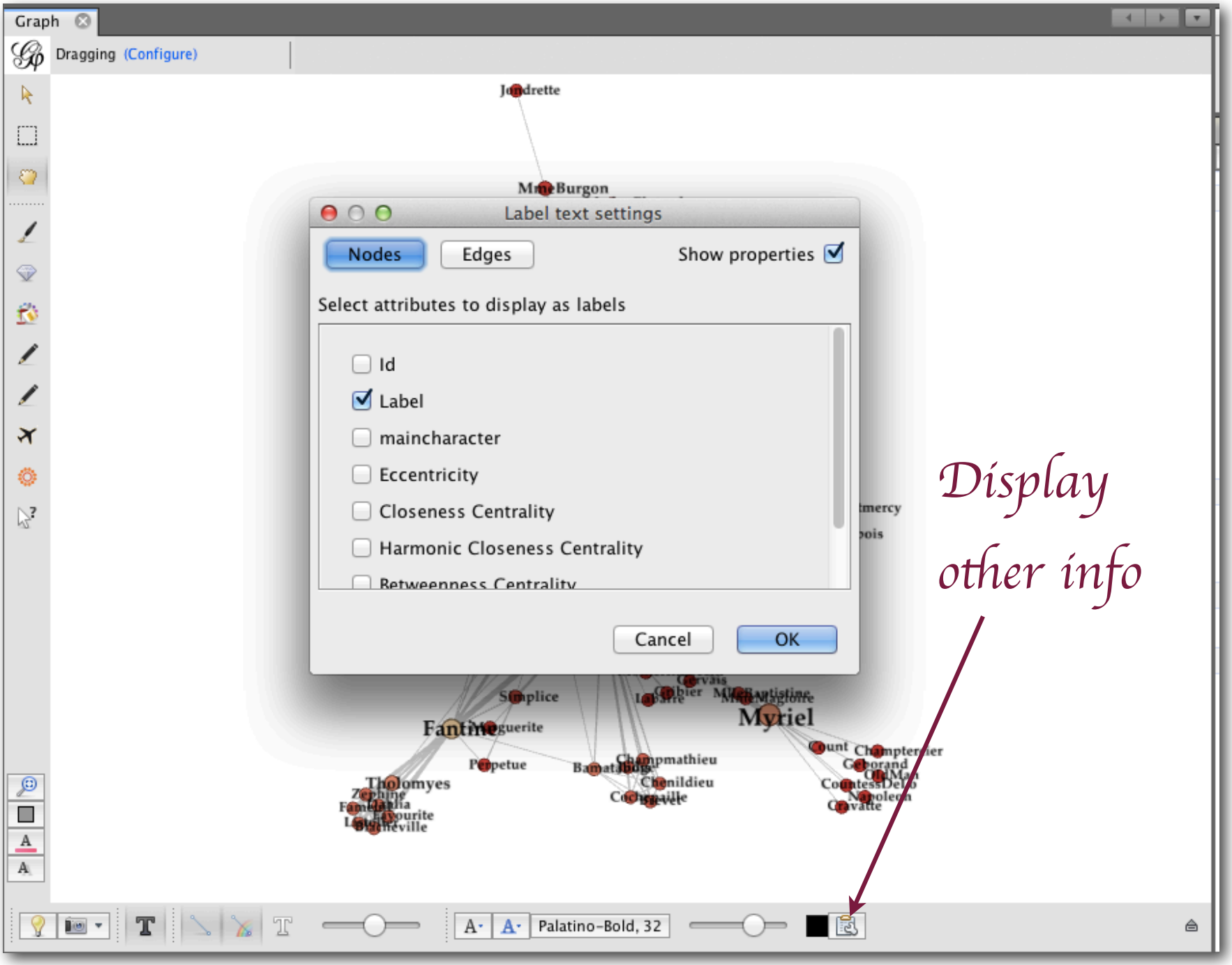
Labels


Dragging (Configure)
Gavroche
Marius
Thenardier
Valjean
Fantine
Myriel
Javert
Palatino-Bold, 32
Adjust labels

Labels

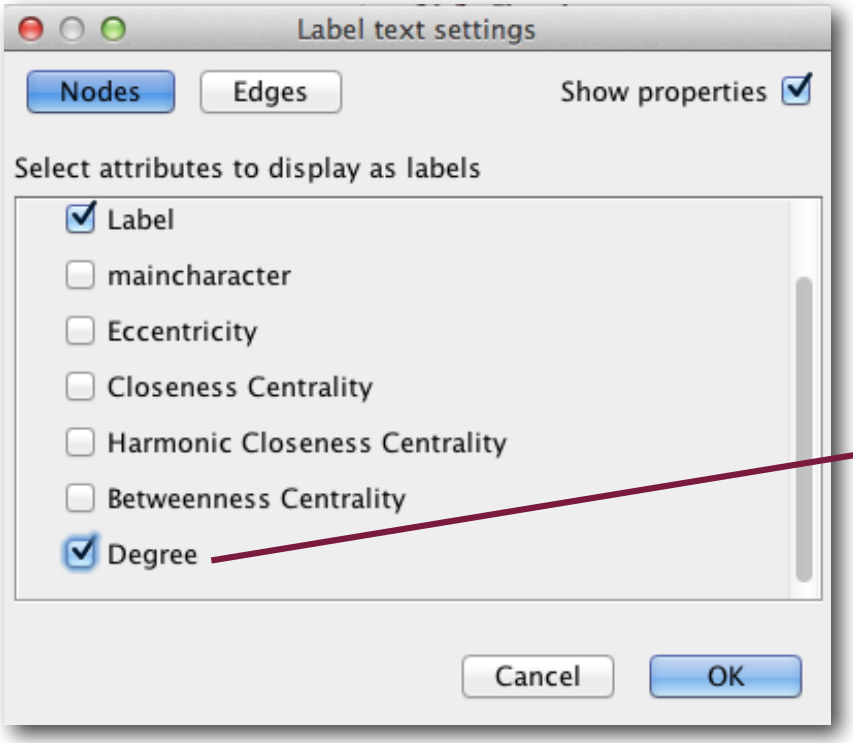

Gephi 0.9.1 - lesmis.gephi
Overview
Data Laboratory
Preview
Workspace 1
Workspace 2
Appearance
Nodes
Edges
Unique
Attribute
Betweenness Centrality
Min size: 15
Max size: 50
Spline...
Apply
Layout
Force Atlas
Run
Force Atlas
Inertia 0.1
Repulsion strength 10000.0
Attraction strength 10.0
Maximum displacem 10.0
Auto stabilize functi
Autostab Strength 80.0
Autostab sensibility 0.2
Gravity 30.0
Attraction Distrib.
Force Atlas
Graph
Dragging (Configure)
Gavroche
Marius
Valjean
Thenardier
Fantine
Myriel
Tholomyes
Enjolras
Mabeuf
Javert
Cosette
Eponine
Babet
Claquesous
Montparnasse
Gueulemer
Brujon
Mme Burgon
MotherPlutarch
Mme Hucheloup
Bahorel
Bossuet
Courfeyrac
Mme Thenardier
Mme Pontmercy
Mlle Vaubois
Baronness T
Toussaint
Simplice
Perpetue
Champmathieu
Chenildieu
Count
Napoleon

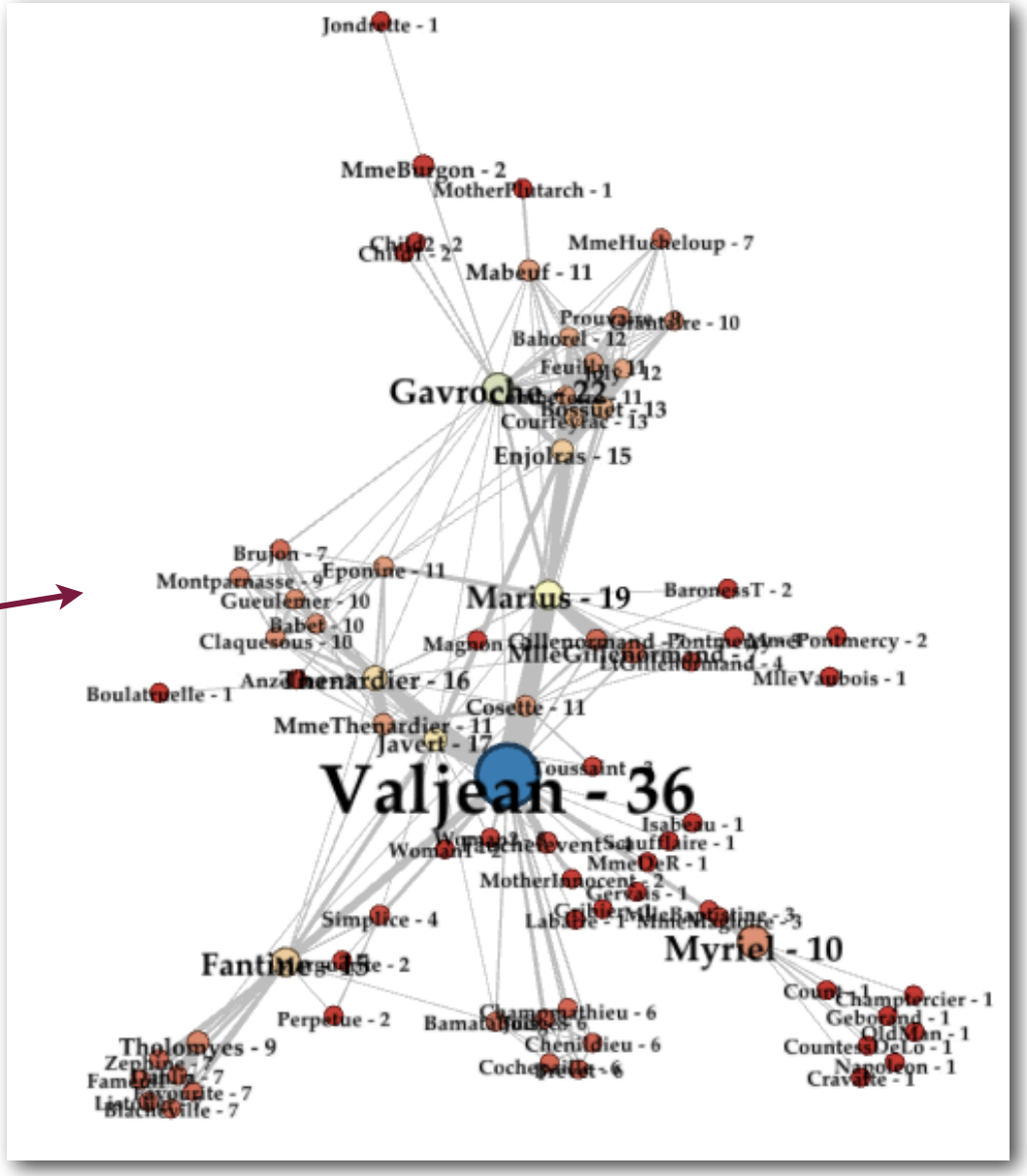
Labels



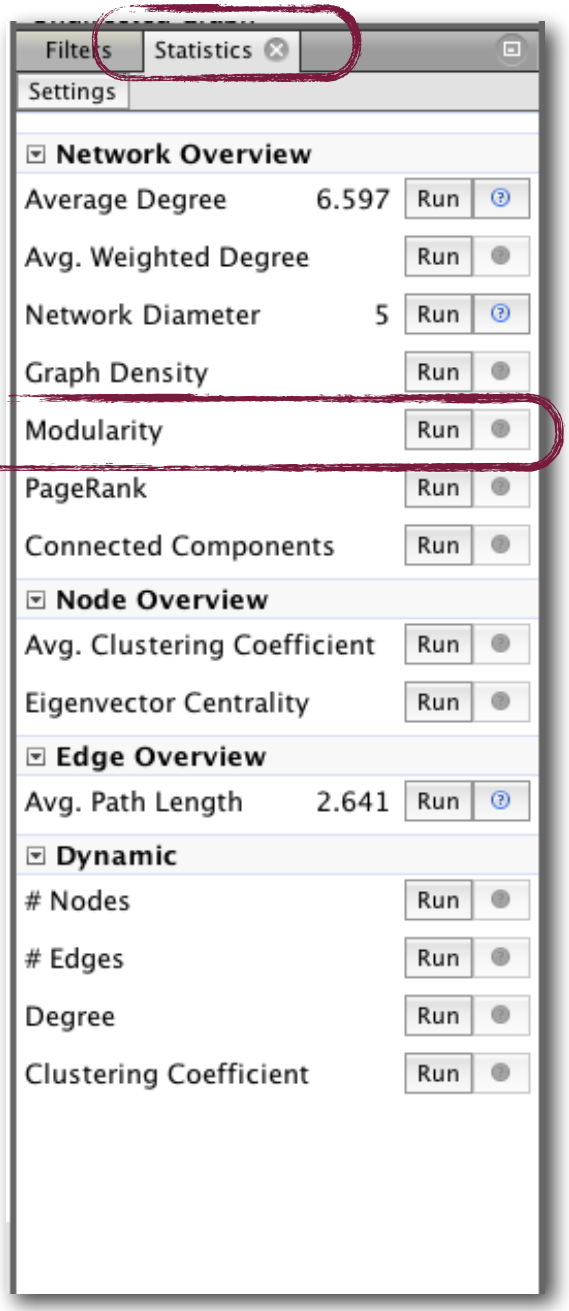
Display other info

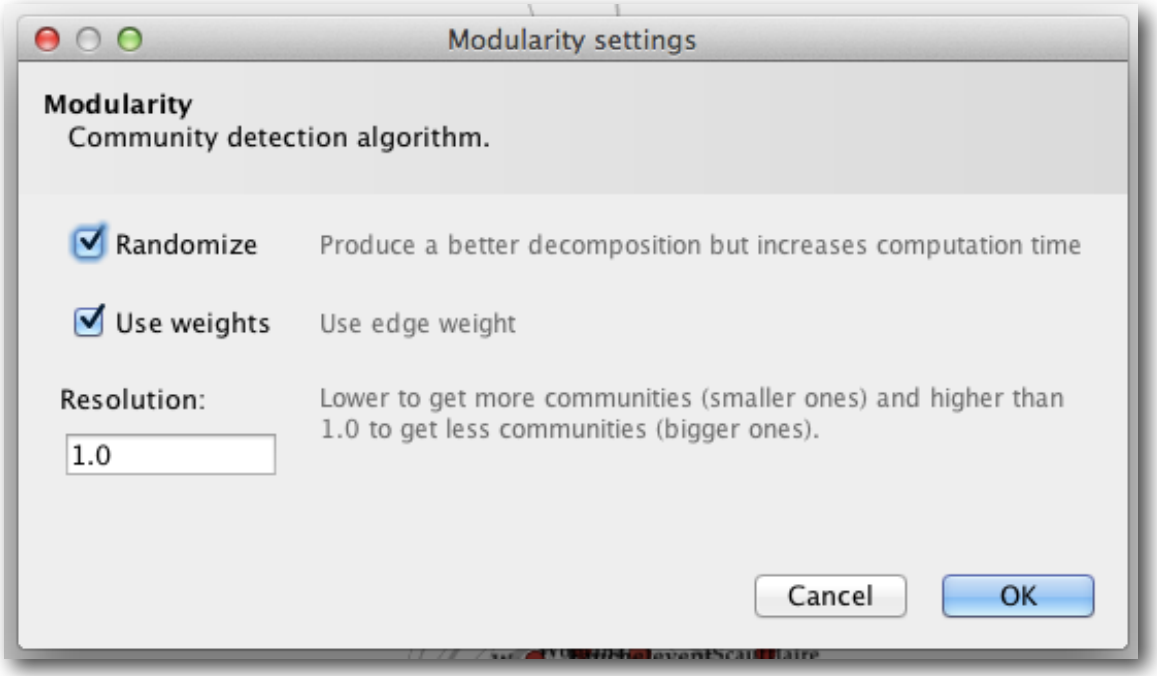
Labels


Label text settings
Nodes
Edges
Show properties
Select attributes to display as labels
Label
maincharacter
Eccentricity
Closeness Centrality
Harmonic Closeness Centrality
Betweenness Centrality
Degree
Cancel
OK


Jondrette - 1
MmeBurgon - 2
MotherPlutarch - 1
MmeHucheloup - 7
Mabeuf - 11
Bahorel - 12
Gavroche - 22
Bossuet - 13
Enjolras - 15
Brujon - 7
Eponine - 11
Montparnasse - 9
Gueulemer - 10
Marius - 19
BaronessT - 2
Babet - 10
Claquesous - 10
Thenardier - 16
Boulatruelle - 1
Cosette - 11
MmeThenardier - 11
Javert - 17
Valjean - 36
Isabeau - 1
Scaufflaire - 1
MotherInnocent - 2
Simplice - 4
Myriel - 10
Perpetue - 2
Tholomyes - 9
Champmathieu - 6
Chenildieu - 6
Champtercier - 1
Geborand - 1
CountessDeLo - 1
Napoleon - 1
Cravatte - 1

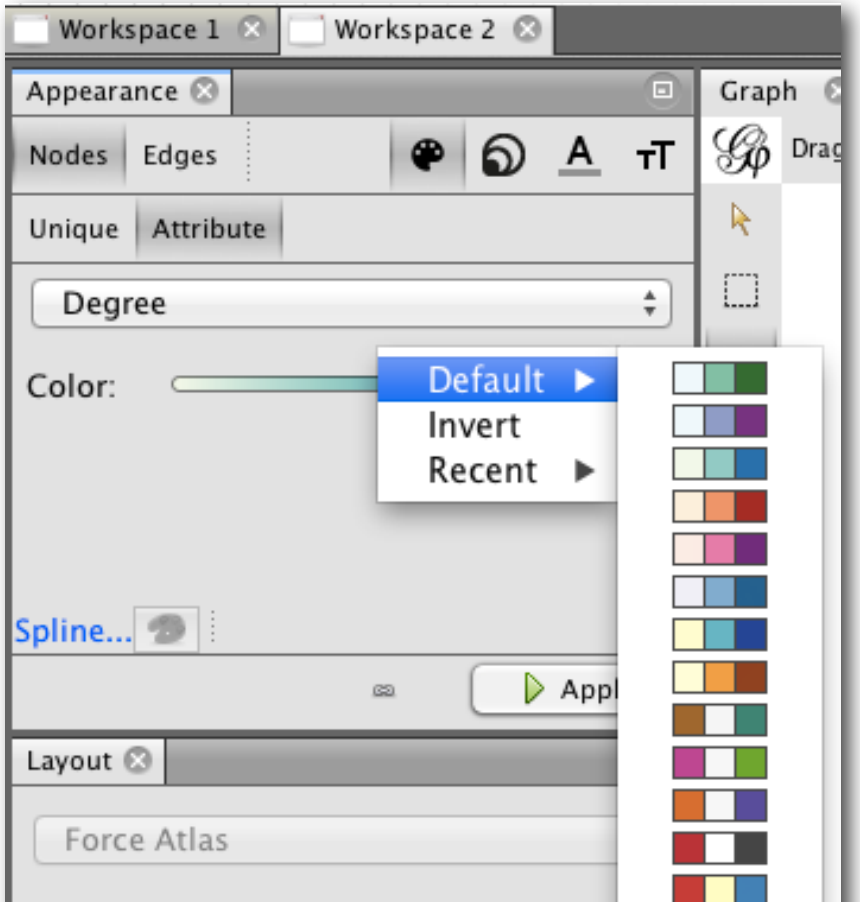
Community detection

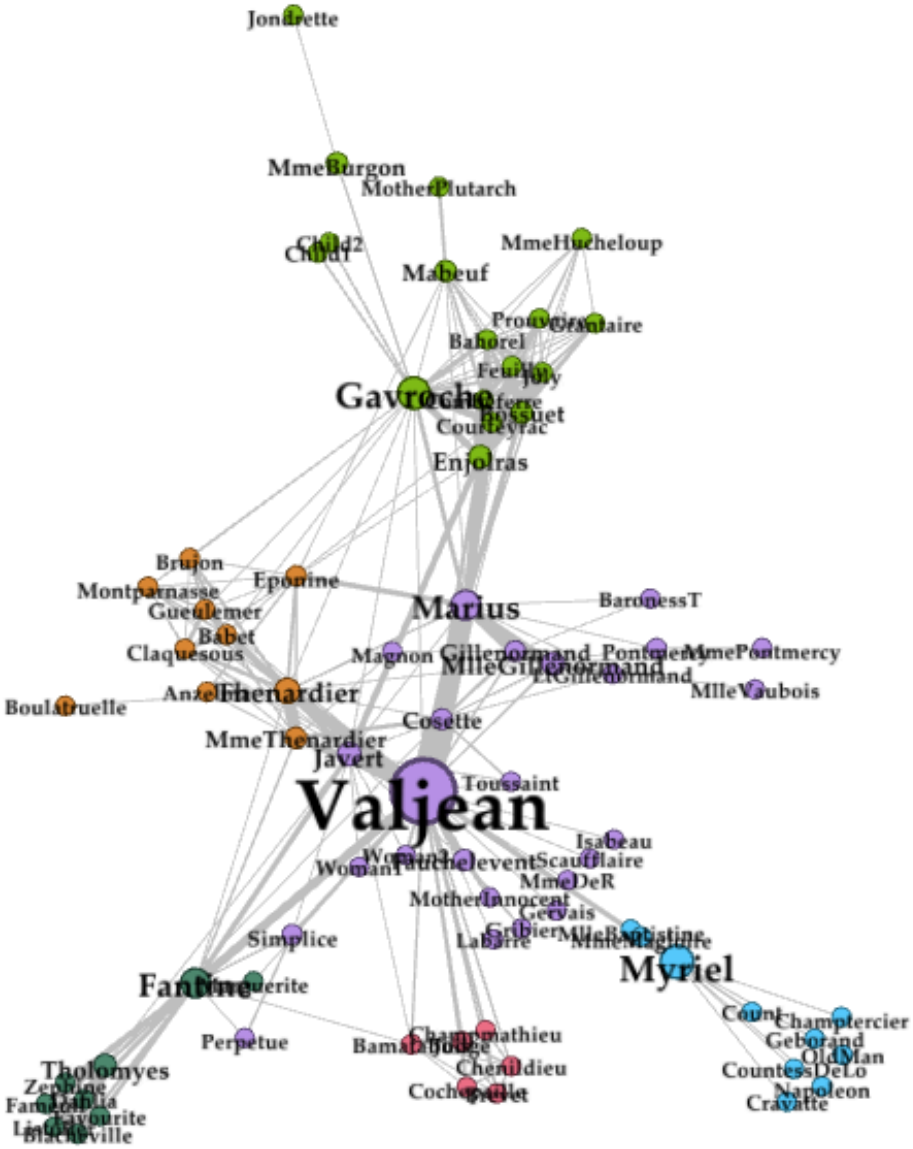




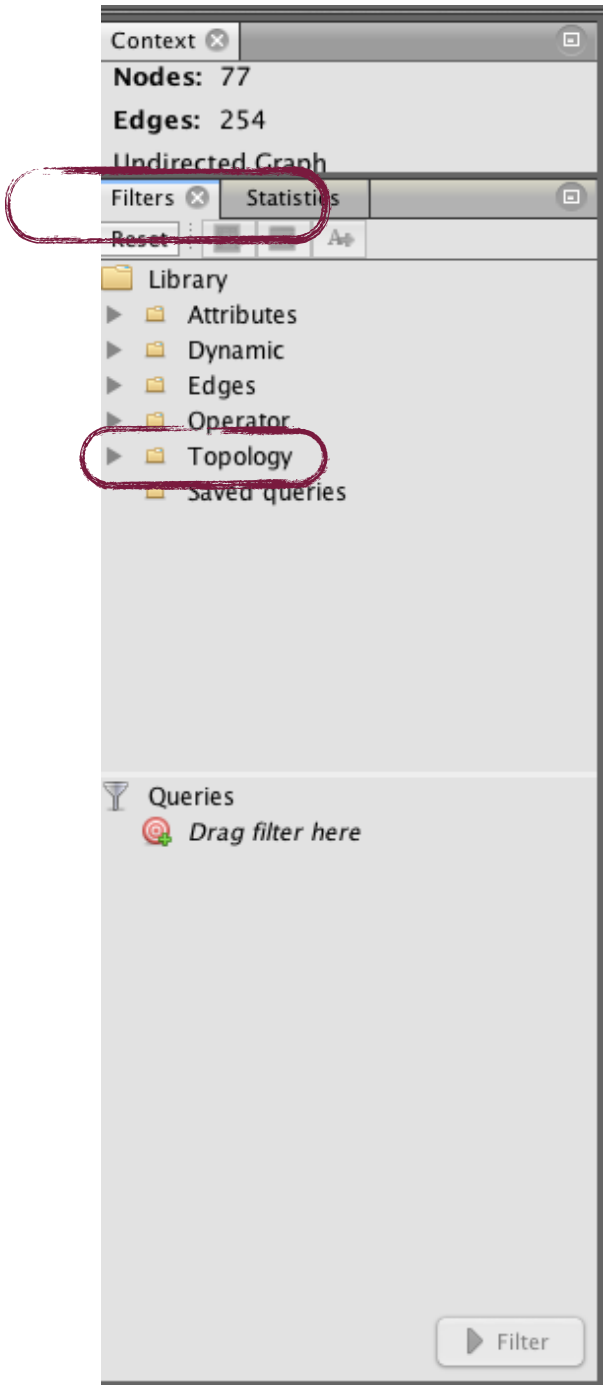
Community detection

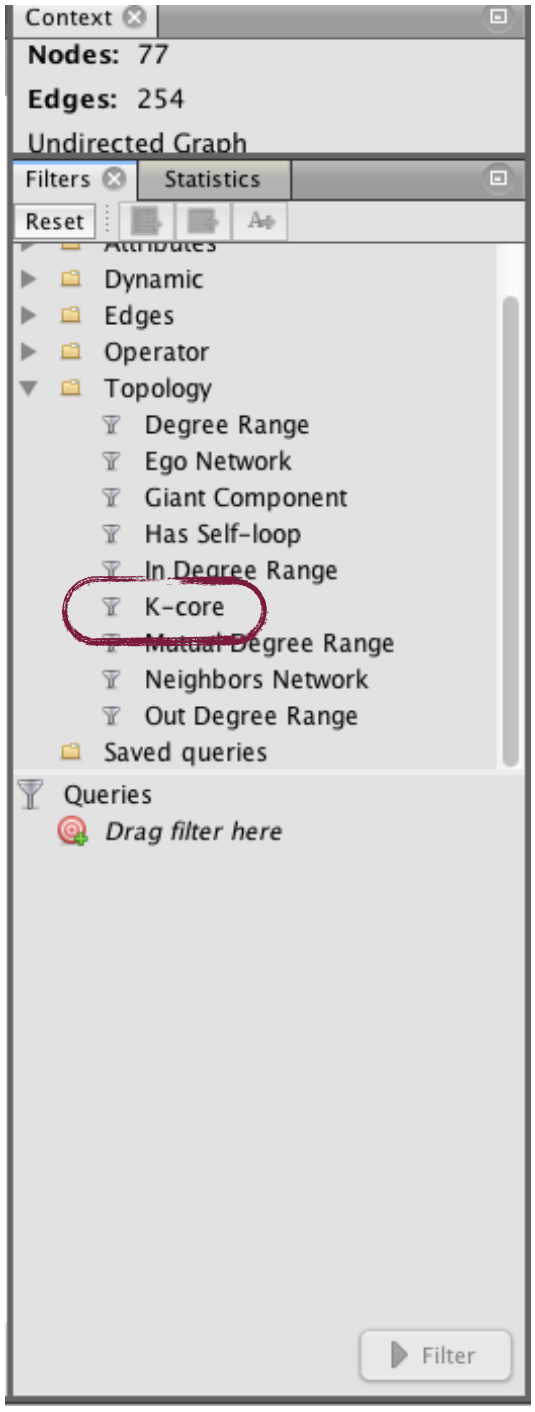
Now you have run the “modularity” function, it appears as an “attribute” under “nodes” in “appearance” -> colour by modularity.



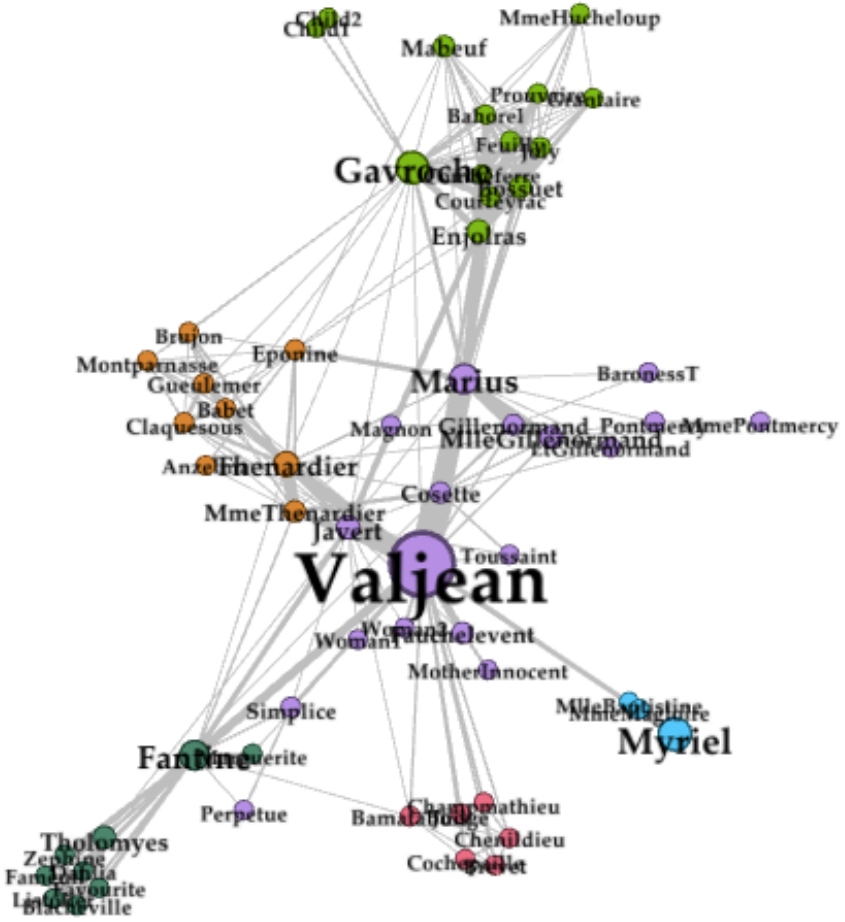


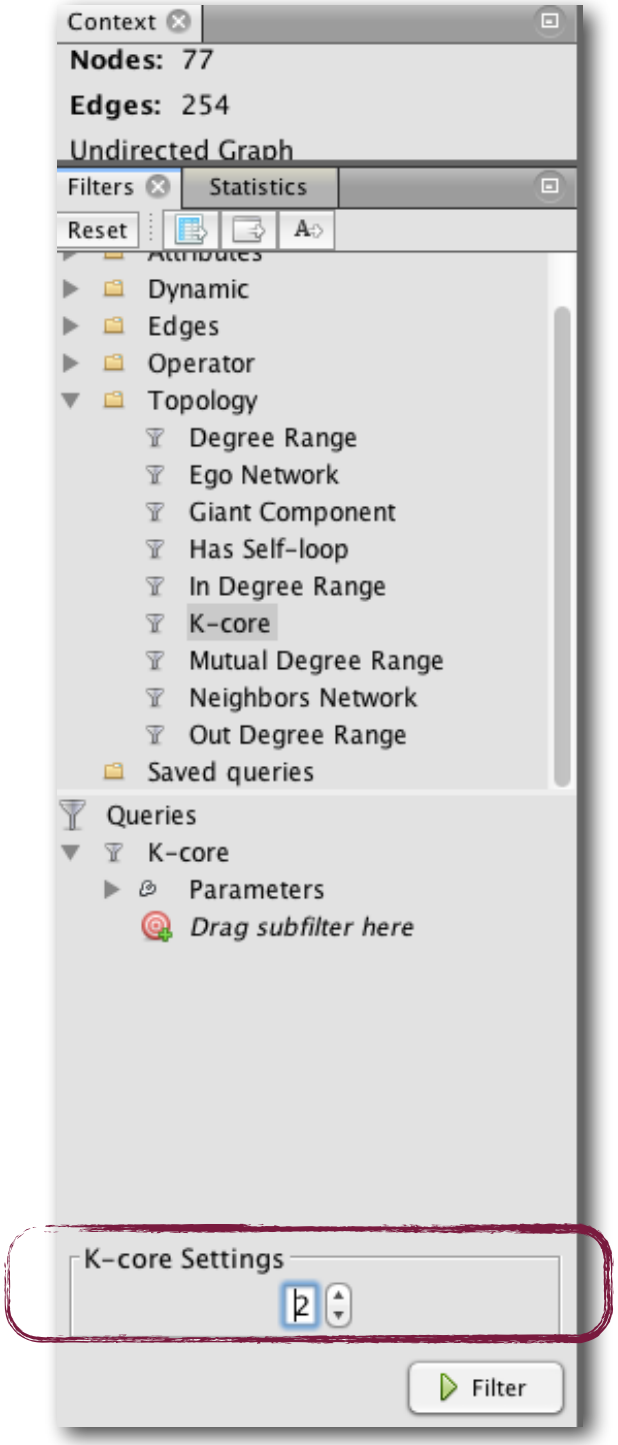
Filters


Context
Nodes: 77
Edges: 254
Undirected Graph
Filters
Statistics
Reset
Library
Attributes
Dynamic
Edges
Operator
Topology
Saved queries
Queries
Drag filter here
Filter

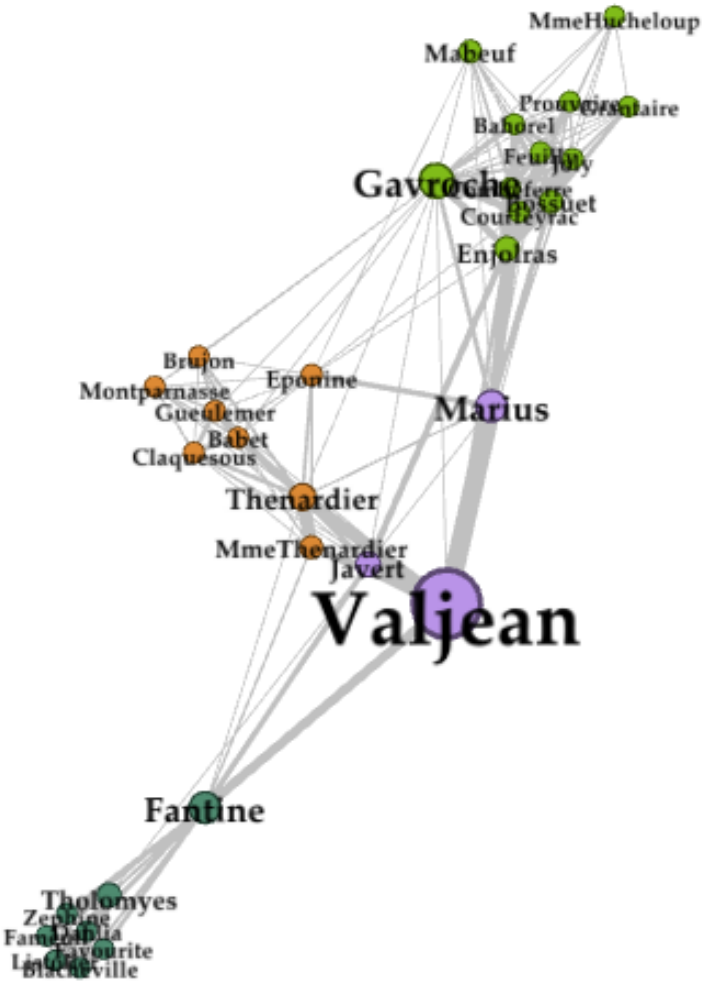

Context
Nodes: 77
Edges: 254
Undirected Graph
Filters
Statistics
Reset
Dynamic
Edges
Operator
Topology
Degree Range
Ego Network
Giant Component
Has Self-loop
In Degree Range
K-core
Neighbors Network
Out Degree Range
Saved queries
Queries
Drag filter here
Filter

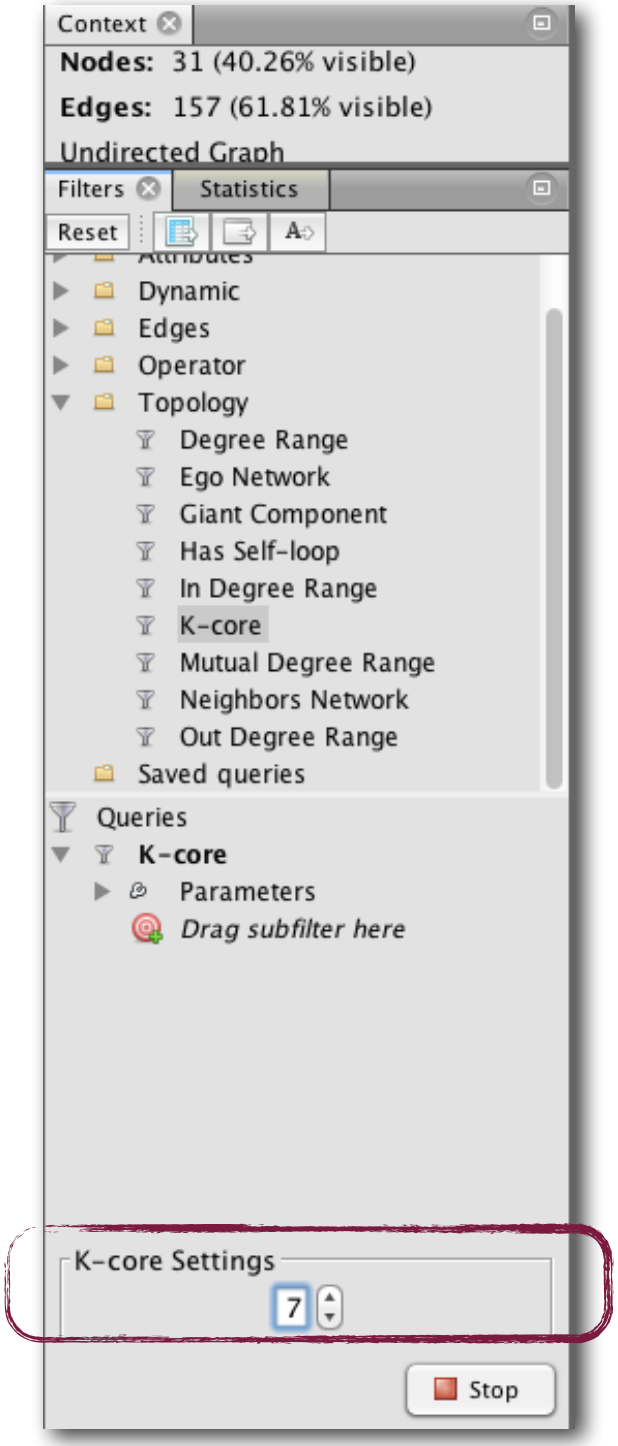
Filters


MmeHucheloup
Mabeuf
Gavroche
Enjolras
Bahorel
Bossuet
Courfeyrac
Brujon
Montparnasse
Gueulemer
Babet
Claquesous
Eponine
Marius
BaronessT
MmePontmercy
Thenardier
Cosette
MmeThenardier
Javert
Valjean
Toussaint
Mother Innocent
Simplice
Fantine
Myriel
Perpetue
Tholomyes

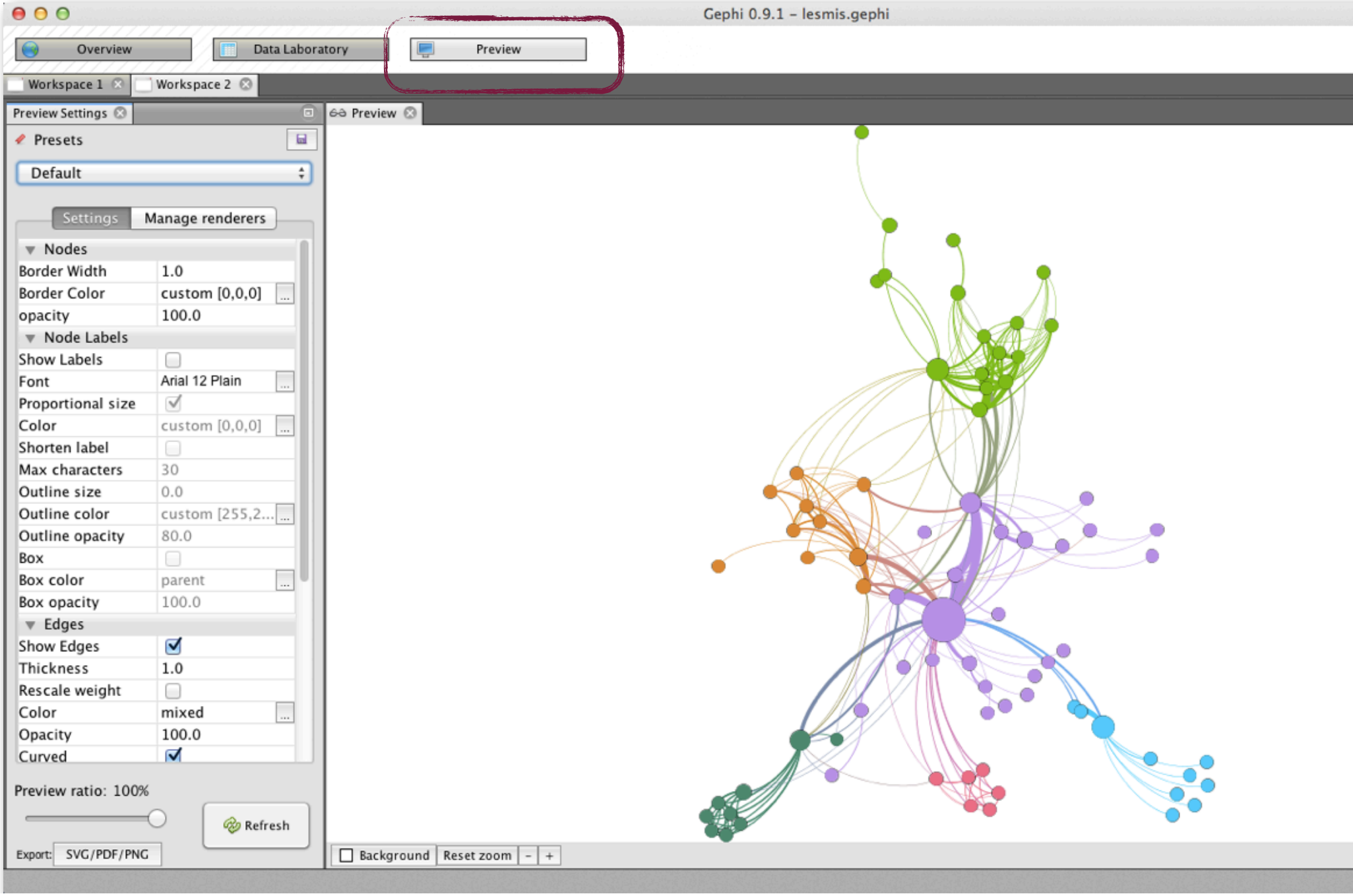

Context
Nodes: 77
Edges: 254
Undirected Graph
Filters
Statistics
Reset
Dynamic
Edges
Operator
Topology
Degree Range
Ego Network
Giant Component
Has Self-loop
In Degree Range
K-core
Mutual Degree Range
Neighbors Network
Out Degree Range
Saved queries
Queries
K-core
Parameters
Drag subfilter here
K-core Settings
2
Filter

Filters

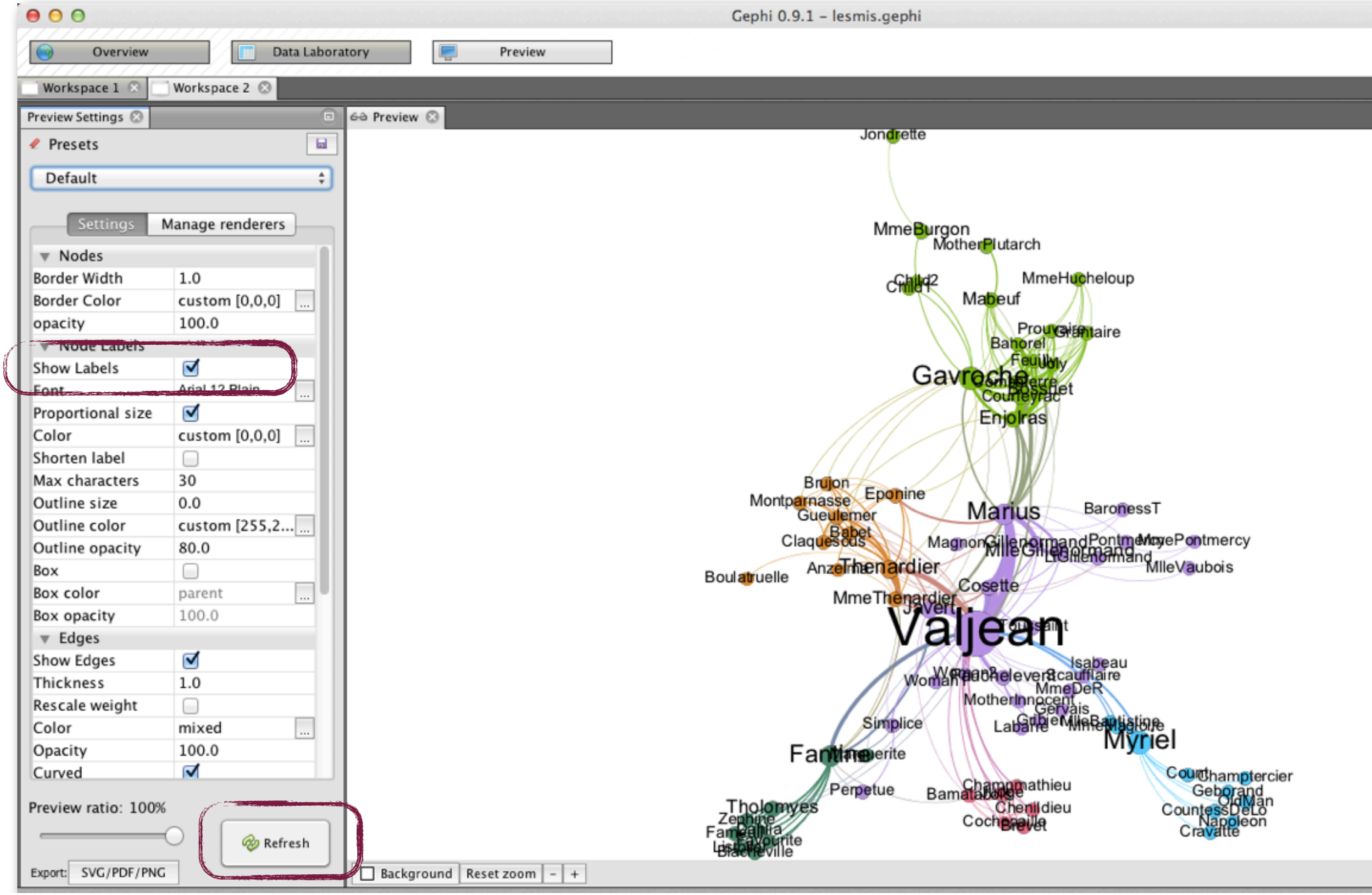

MmeHucheloup
Mabeuf
Bahorel
Gavroche
Bossuet
Courfeyrac
Enjolras
Brujon
Montparnasse
Gueulemer
Babet
Claquesous
Eponine
Marius
Thenardier
MmeThenardier
Javert
Valjean
Fantine
Tholomyes


Context
Nodes: 31 (40.26% visible)
Edges: 157 (61.81% visible)
Undirected Graph
Filters
Statistics
Reset
Dynamic
Edges
Operator
Topology
Degree Range
Ego Network
Giant Component
Has Self-loop
In Degree Range
K-core
Mutual Degree Range
Neighbors Network
Out Degree Range
Saved queries
Queries
K-core
Parameters
Drag subfilter here
K-core Settings
7
Stop

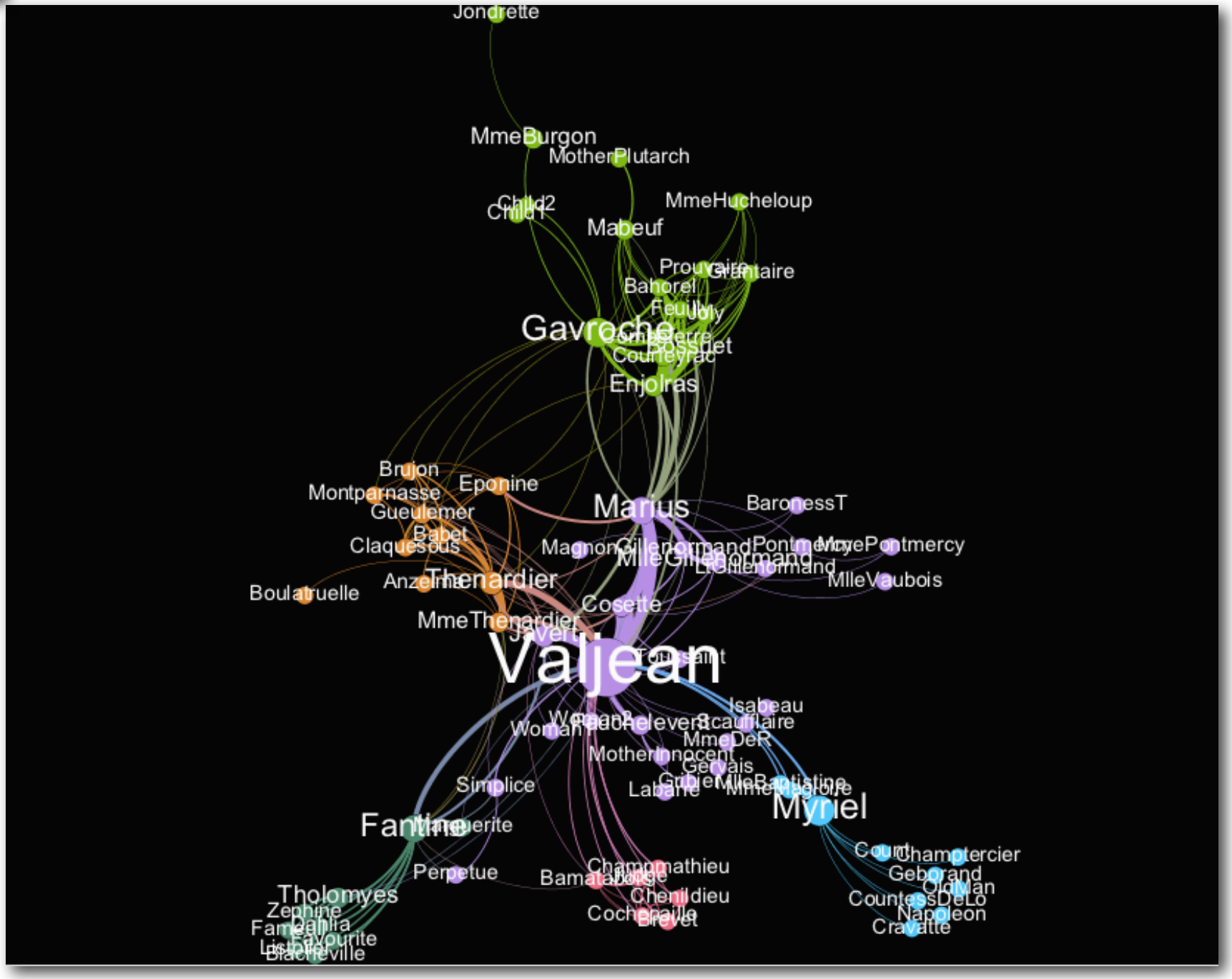
Exporting the image



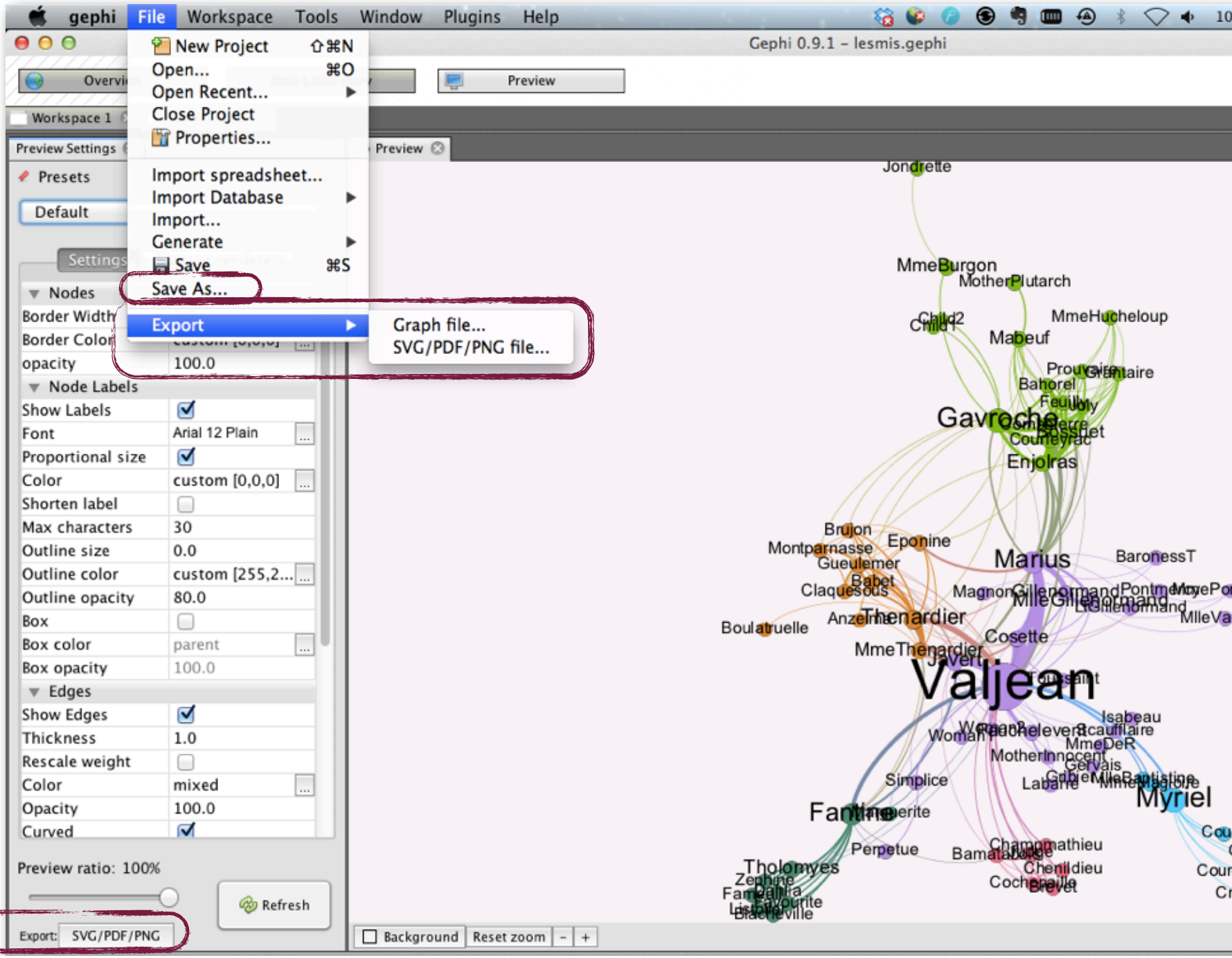
Exporting the image



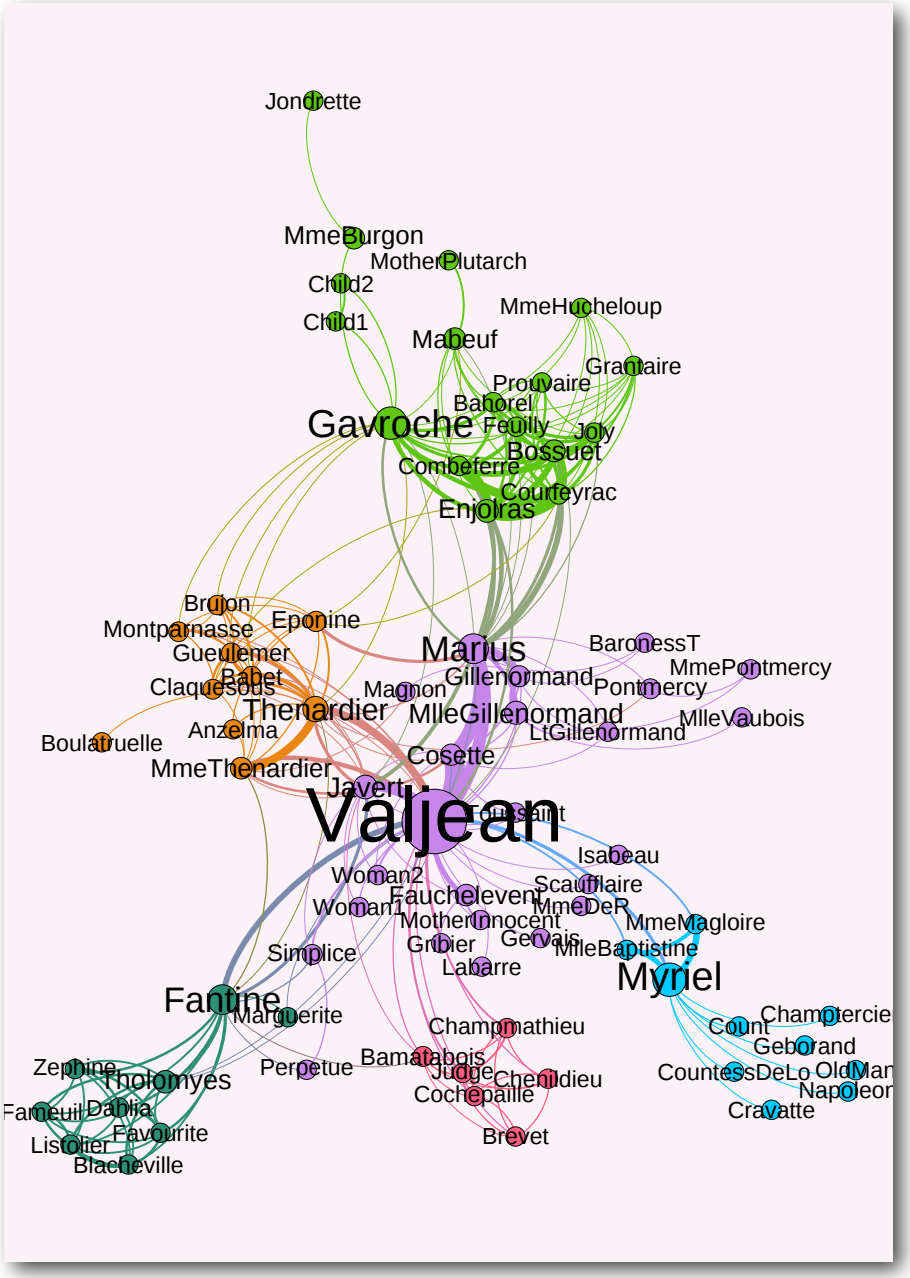
Exporting the image



Exporting the image

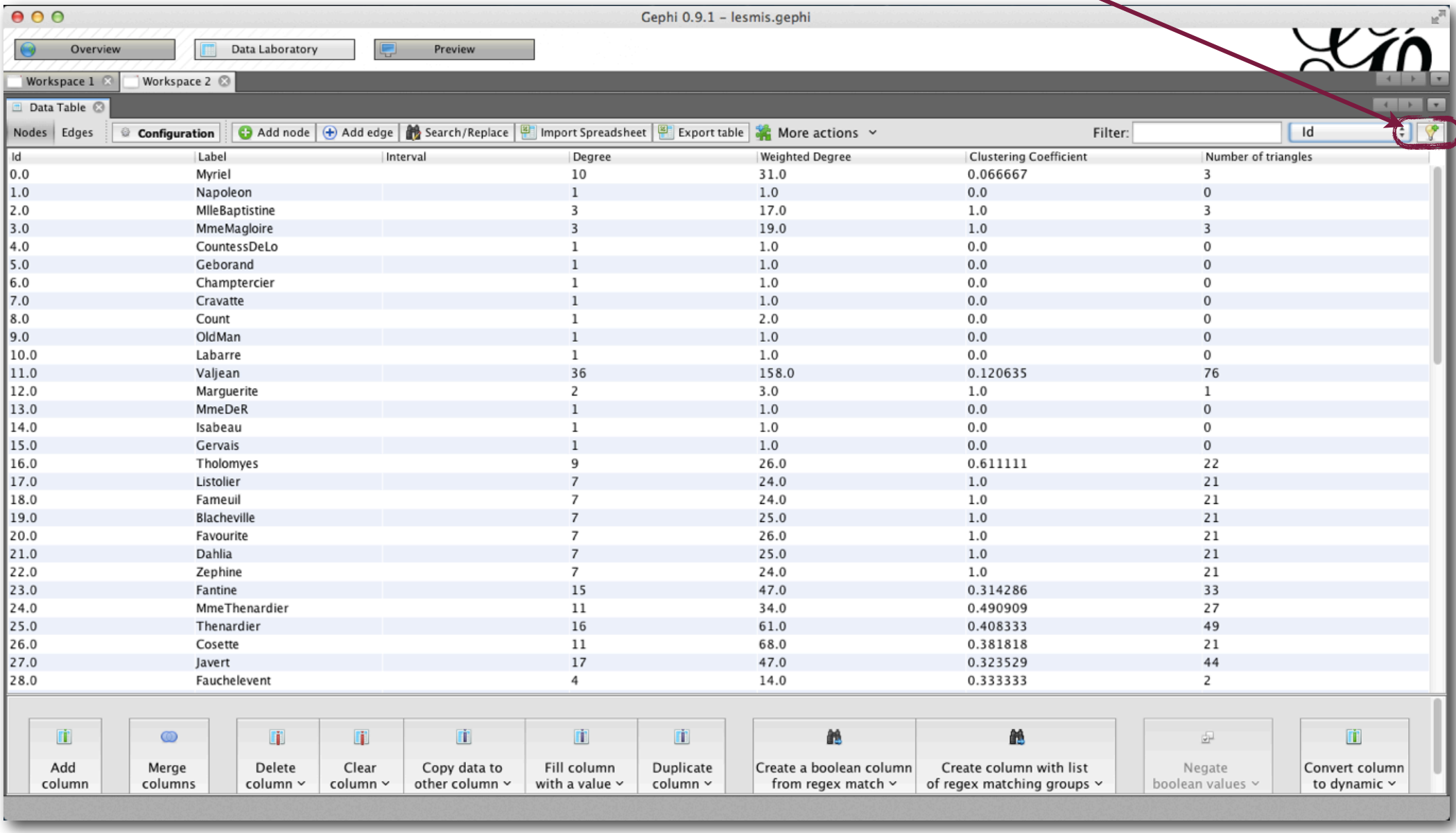


Exporting the image



Exporting statistics

Choose properties to display



Id	Label	Interval	Degree	Weighted Degree	Clustering Coefficient	Number of triangles
0.0	Myriel		10	31.0	0.066667	3
1.0	Napoleon		1	1.0	0.0	0
2.0	MlleBaptistine		3	17.0	1.0	3
3.0	MmeMagloire		3	19.0	1.0	3
4.0	CountessDeLo		1	1.0	0.0	0
5.0	Geborand		1	1.0	0.0	0
6.0	Champtercier		1	1.0	0.0	0
7.0	Cravatte		1	1.0	0.0	0
8.0	Count		1	2.0	0.0	0
9.0	OldMan		1	1.0	0.0	0
10.0	Labarre		1	1.0	0.0	0
11.0	Valjean		36	158.0	0.120635	76
12.0	Marguerite		2	3.0	1.0	1
13.0	MmeDeR		1	1.0	0.0	0
14.0	Isabeau		1	1.0	0.0	0
15.0	Gervais		1	1.0	0.0	0
16.0	Tholomyes		9	26.0	0.611111	22
17.0	Listolier		7	24.0	1.0	21
18.0	Fameuil		7	24.0	1.0	21
19.0	Blacheville		7	25.0	1.0	21
20.0	Favourite		7	26.0	1.0	21
21.0	Dahlia		7	25.0	1.0	21
22.0	Zephine		7	24.0	1.0	21
23.0	Fantine		15	47.0	0.314286	33
24.0	MmeThenardier		11	34.0	0.490909	27
25.0	Thenardier		16	61.0	0.408333	49
26.0	Cosette		11	68.0	0.381818	21
27.0	Javert		17	47.0	0.323529	44
28.0	Fauchelevent		4	14.0	0.333333	2

Exporting statistics

Gephi 0.9.1 – lesmis.gephi

Overview | Data Laboratory | Preview

Workspace 1 | Workspace 2

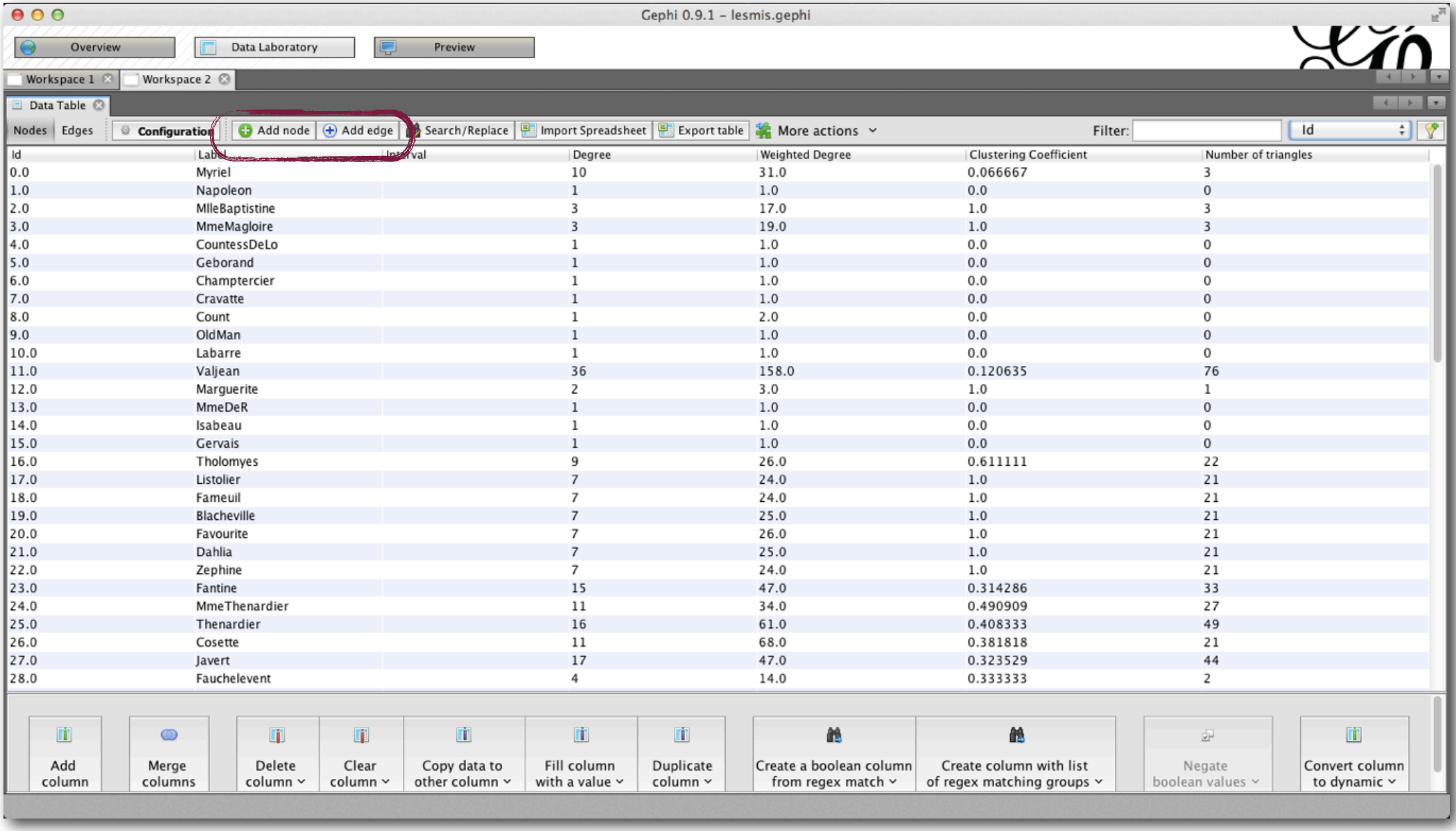
Data Table

Nodes | Edges | Configuration | Add node | Add edge | Search/Replace | Import Spreadsheet | Export table | More actions | Filter: | Id

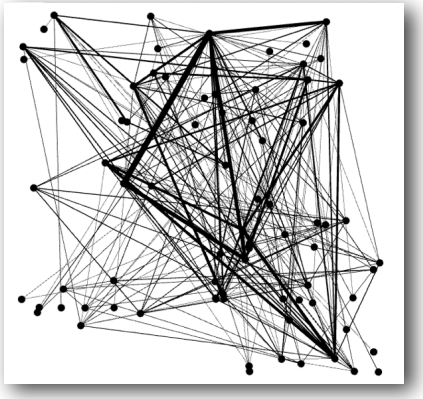
Id	Label	Interval	Degree	Weighted Degree	Clustering Coefficient	Number of triangles
0.0	Myriel		10	31.0	0.066667	3
1.0	Napoleon		1	1.0	0.0	0
2.0	MlleBaptistine		3	17.0	1.0	3
3.0	MmeMagloire		3	19.0	1.0	3
4.0	CountessDeLo		1	1.0	0.0	0
5.0	Geborand		1	1.0	0.0	0
6.0	Champtercier		1	1.0	0.0	0
7.0	Cravatte		1	1.0	0.0	0
8.0	Count		1	2.0	0.0	0
9.0	OldMan		1	1.0	0.0	0
10.0	Labarre		1	1.0	0.0	0
11.0	Valjean		36	158.0	0.120635	76
12.0	Marguerite		2	3.0	1.0	1
13.0	MmeDeR		1	1.0	0.0	0
14.0	Isabeau		1	1.0	0.0	0
15.0	Gervais		1	1.0	0.0	0
16.0	Tholomyes		9	26.0	0.611111	22
17.0	Listolier		7	24.0	1.0	21
18.0	Fameuil		7	24.0	1.0	21
19.0	Blacheville		7	25.0	1.0	21
20.0	Favourite		7	26.0	1.0	21
21.0	Dahlia		7	25.0	1.0	21
22.0	Zephine		7	24.0	1.0	21
23.0	Fantine		15	47.0	0.314286	33
24.0	MmeThenardier		11	34.0	0.490909	27
25.0	Thenardier		16	61.0	0.408333	49
26.0	Cosette		11	68.0	0.381818	21
27.0	Javert		17	47.0	0.323529	44
28.0	Fauchelevent		4	14.0	0.333333	2

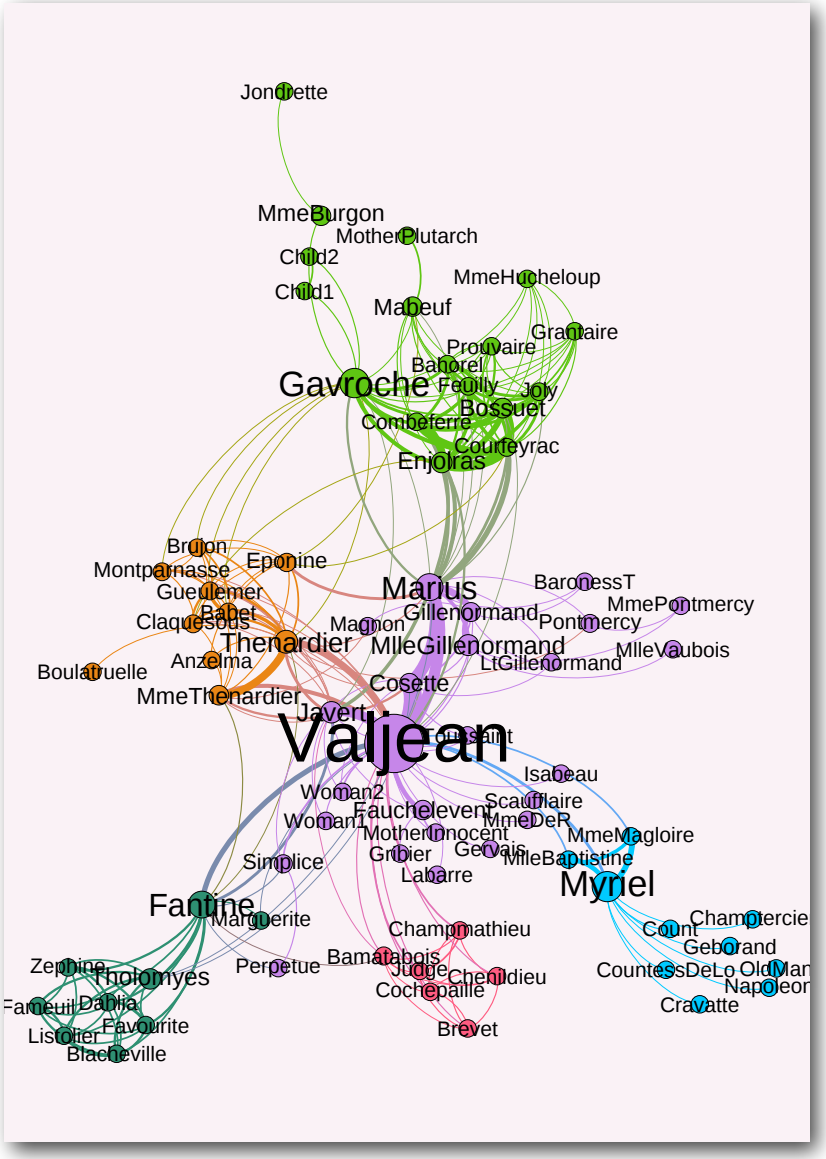
Add column | Merge columns | Delete column | Clear column | Copy data to other column | Fill column with a value | Duplicate column | Create a boolean column from regex match | Create column with list of regex matching groups | Negate boolean values | Convert column to dynamic

Update network



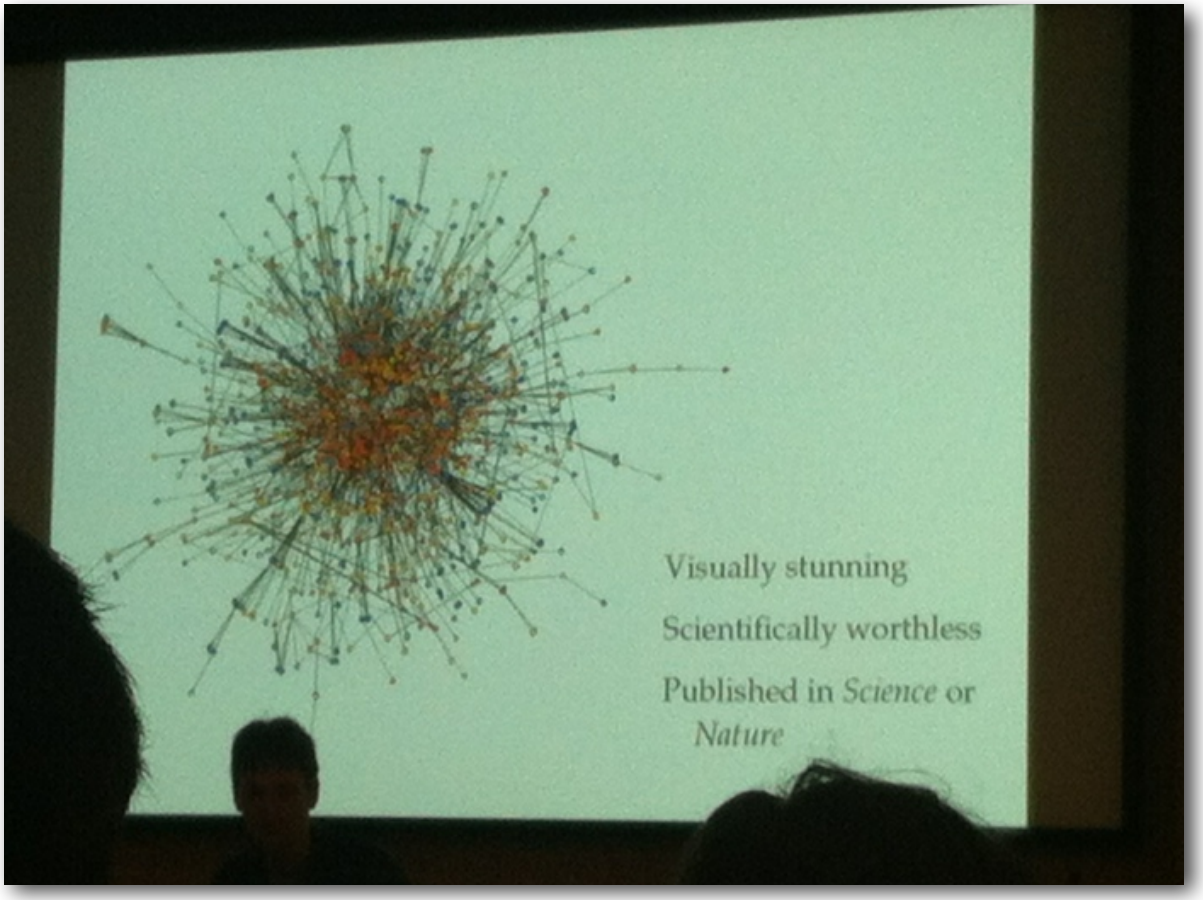
About the network





NOT done at this point! What have we learned about the network from the visualisation exercise? What are we showing?

Hairballs are not useful



Definition: “Ridiculogram”

Visually stunning. Scientifically worthless. Published in Science or Nature.

- Mark Newman 2014

Visualising networks

- Good to have a ‘feeling’ for the network

- It is possible to visualise very large networks, but the data should be processed in a meaningful way first

- If you work with networks, sooner or later you will need to operations that are not already implemented

- Start exploring/creating your own tools! One of your best friends in this process:

Practice, practice, practice!

Now it is your turn!

You have another network - a (rather outdated) network science collaboration network. Visualise and explore this network with Gephi.

Be prepared to comment on and tell us about it. Is it modular? If so, what does these modules correspond to? Are there hubs? If so, how would you interpret them? What do other centrality measures tell you?

Wednesday: NetworkX

NetworkX

NetworkX Home | Documentation | Download | Developer (Github)

High-productivity software for complex networks

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

Documentation
all documentation

Examples
using the library

Reference
all functions and methods

Features

- Python language data structures for graphs, digraphs, and multigraphs.
- Many standard graph algorithms
- Network structure and analysis measures
- Generators for classic graphs, random graphs, and synthetic networks
- Nodes can be "anything" (e.g. text, images, XML records)
- Edges can hold arbitrary data (e.g. weights, time-series)
- Open source BSD license
- Well tested: more than 1800 unit tests, >90% code coverage
- Additional benefits from Python: fast prototyping, easy to teach, multi-platform

Versions

Latest Release

networkx-1.11
30 January 2016
downloads | docs | pdf

Development

2.0dev
github | docs | pdf
build passing
coverage 94%

Contact

Mailing list
Issue tracker

NetworkX Home | Documentation | Download | Developer (Github)