Lab 4-1 Instructions

Big Data Analytics and Social Media

In this lab, we will start practising Network Graph Visualisation with Gephi. Gephi is an open-source software package that enables basic exploration and analysis of network graphs through visualisation. You will use Gephi to visualise the network graphs you created in Lab 3-1 and analyse their **Page Rank, Centrality,** and **Communities**.

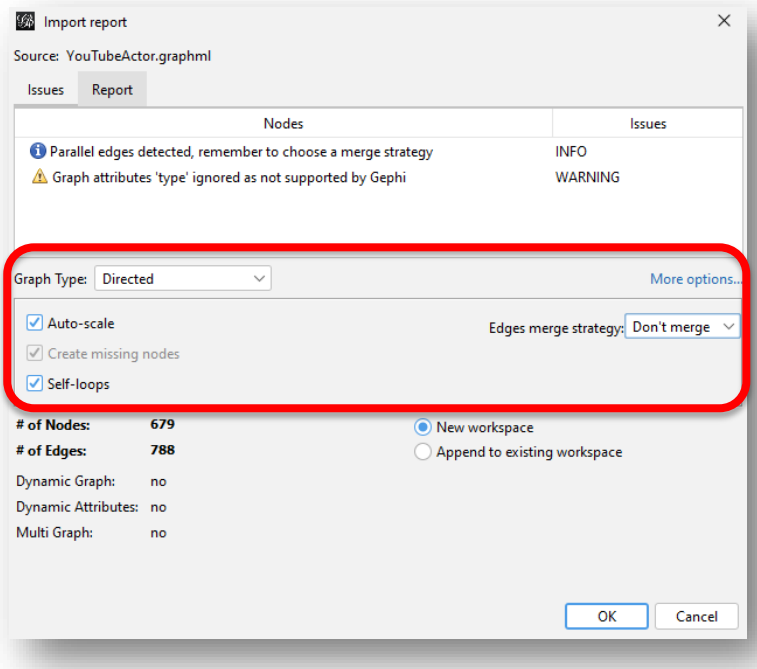
If you haven't used the scripts from Lab 3-1 to create your network graphs, please do so before you continue with this lab.

Contents

Load the Graph

Open **Gephi**.

Click *File* > *Open* and select one of the *GRAPHML* files you created previously. Here, we will use a YouTube Actor network graph. It will bring up the following import dialog:



You can ignore the two issues listed.[1] Make sure to select '**Directed'** for the *Graph Type* if you are loading an **Actor** network and '**Undirected'** if you are loading a **Semantic** network. Choose '**Don't merge**' for the *Edges merge strategy*. (Click on *More options…* if you cannot see the extra options.) Make sure that the three boxes for *Auto-scale, Create missing nodes,* and *Self-loops* are ticked. Then click *OK*.

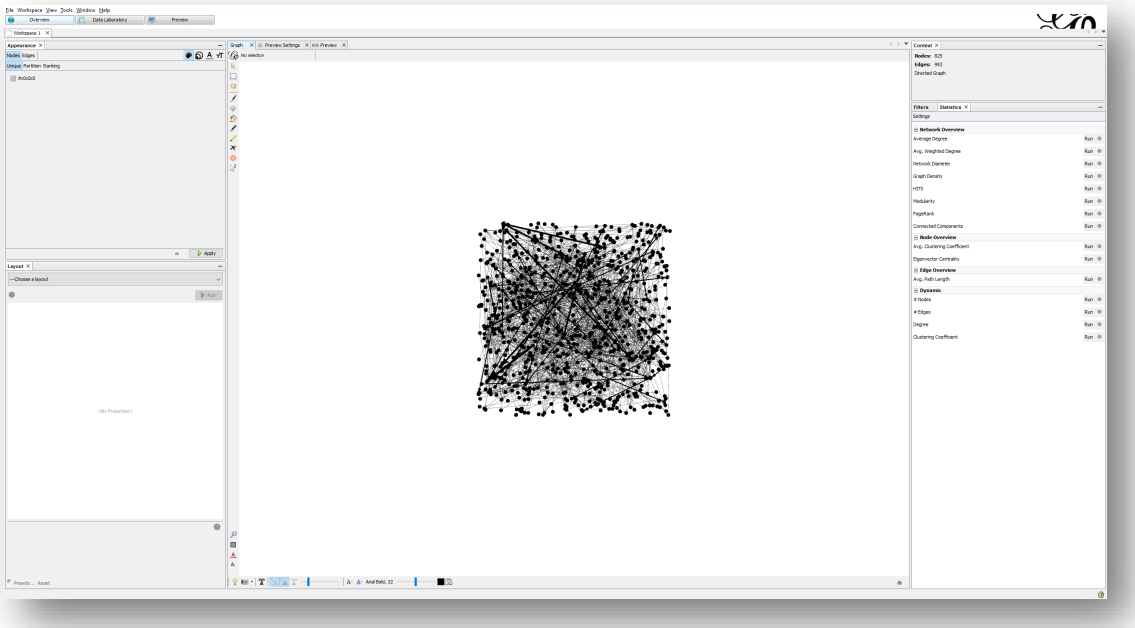
[1] *Parallel edges* means that more than one (directed) edge exists between two nodes. Gephi wants us to address this by choosing one of the possible merge strategies.

The *type* attribute is required by the vosonSML package and specifies the source of the data (i.e., YouTube). This attribute is not used to visualise the graph, so we can safely ignore this warning.

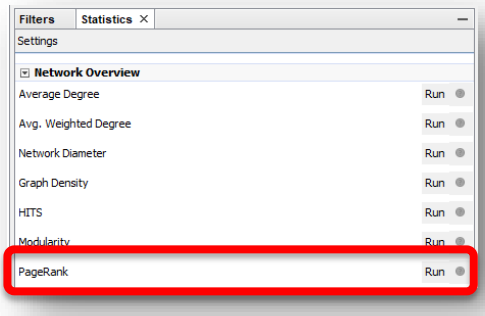
Analyse Page Rank

You should see a graph showing your YouTube actor data as a network of nodes and edges. You can zoom in/out with the scroll wheel on your mouse. You can move the graph by holding the right mouse button and dragging.

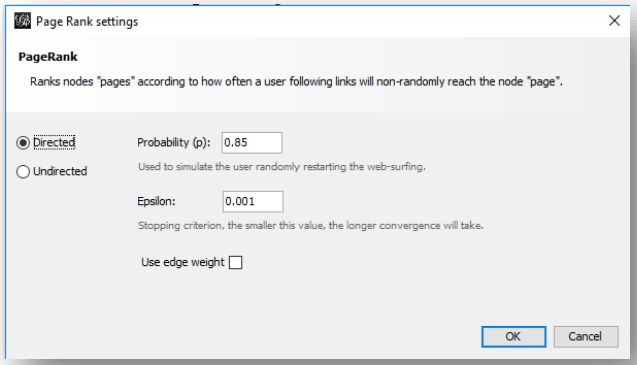
(If you cannot see the graph, click on *Data Laboratory* once and then on *Overview* in the top left of your Gephi window.)



Run the *PageRank* tool on the right panel under *Statistics*.

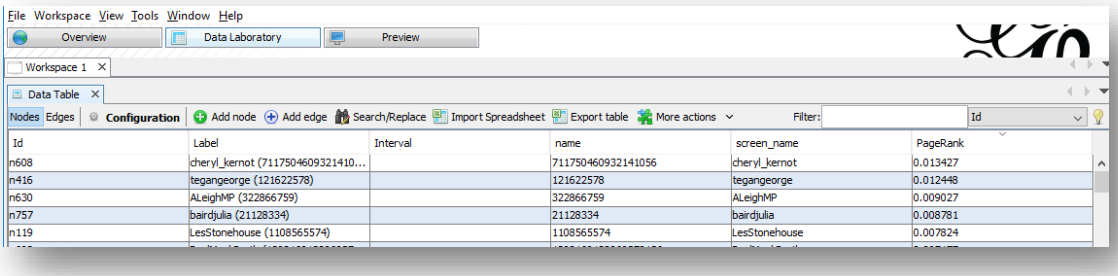


You can leave the default settings for *Probability* and *Epsilon.* Make sure that *Directed* is selected, *Use edge weight* is unticked, and click *OK*.



You can close the dialog box with the PageRank report.

The PageRank algorithm in Gephi runs similarly as in RStudio. In fact, let's compare your results in Gephi to your results in RStudio. Click on *Data Laboratory* at the top left in Gephi. You will find a list of all the nodes in your graph. The right-most column shows you the PageRank score for each node. Click on the header of that column twice, to sort it in descending order.



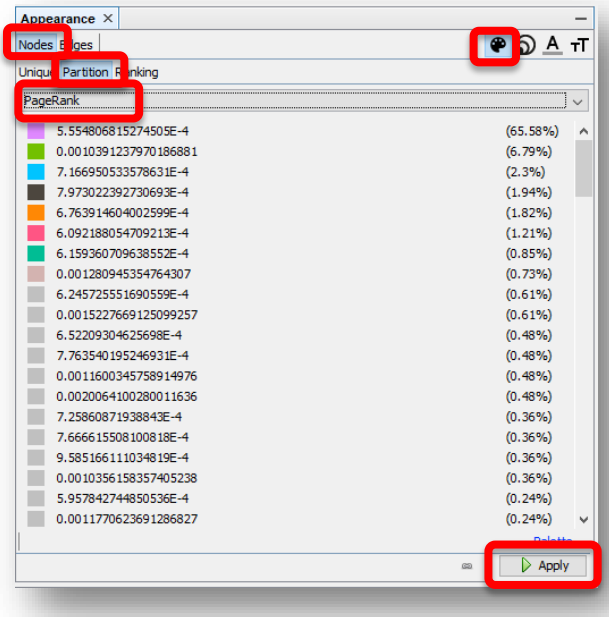
Now go to RStudio and run the PageRank algorithm on your graph. If you forgot how to do that, download the "4-1_Lab_Script.R" from the course site. Make sure to choose the *RDS* file that corresponds to the *GRAPHML* file that you just loaded into Gephi, e.g., in our case here we would replace *XXX* with *YouTubeActor*:

```
library(vosonSML)
library(igraph)

# Load your chosen network graph

network_graph <- readRDS("XXX.rds")

# Run the Page Rank algorithm

rank_yt_actor <- sort(page_rank(network_graph)$vector, decreasing = TRUE)
rank_yt_actor[1:10]
```

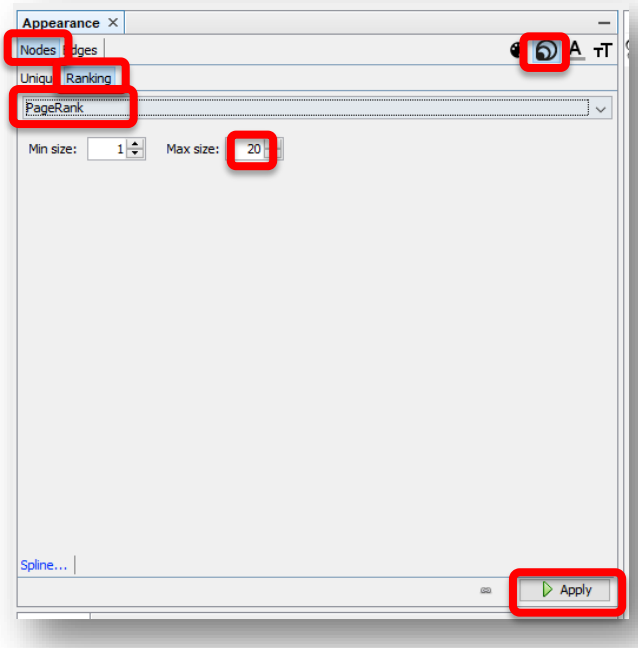
Compare the results in RStudio with the list here in Gephi. You should be able to find the top users somewhere near the top of the list. However, they will not necessarily be directly the same in Gephi, because the PageRank algorithm is a non-deterministic algorithm (which basically means that you get different results every time you run it).

Set the Appearance and Visual Properties

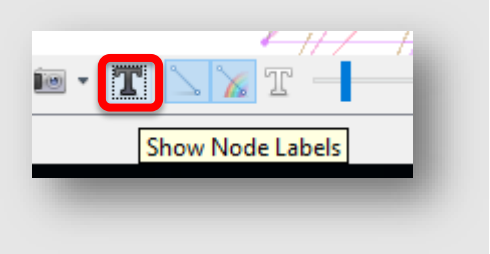
Go back to the *Overview* tab to see your graph. In the *Appearance* panel on the top left, choose *Nodes* → *Partition* and then *PageRank*. Click *Apply*. This will set the colour of nodes based on their PageRank.



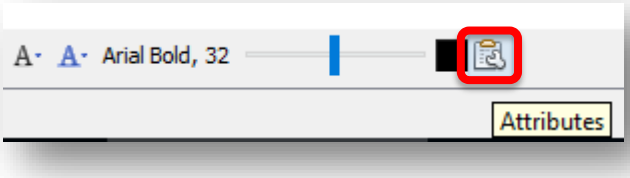
Keep *Nodes* selected and go to *Size* (the multiple circle icon) → *Ranking* and choose *PageRank* and set the max size to 20. Click *Apply* to confirm. This will emphasise the significance of dominant nodes in the visualisation.

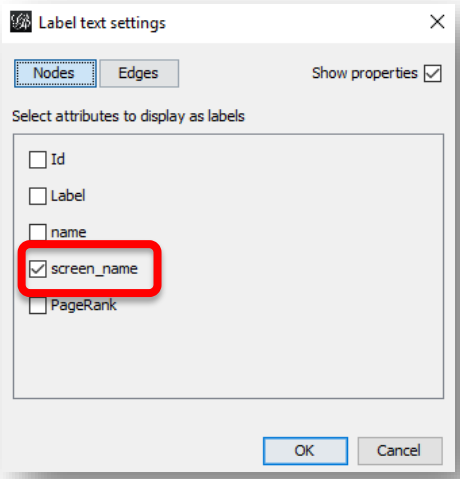


Click on the bold *T* to show node labels.

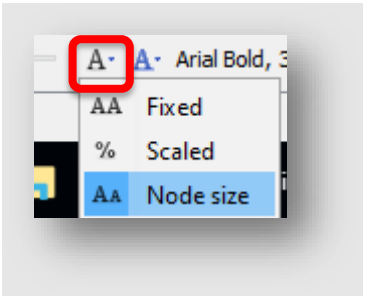


Click on the *Attributes* symbol, then untick the box for *Label* and tick the box for *screen_name*.

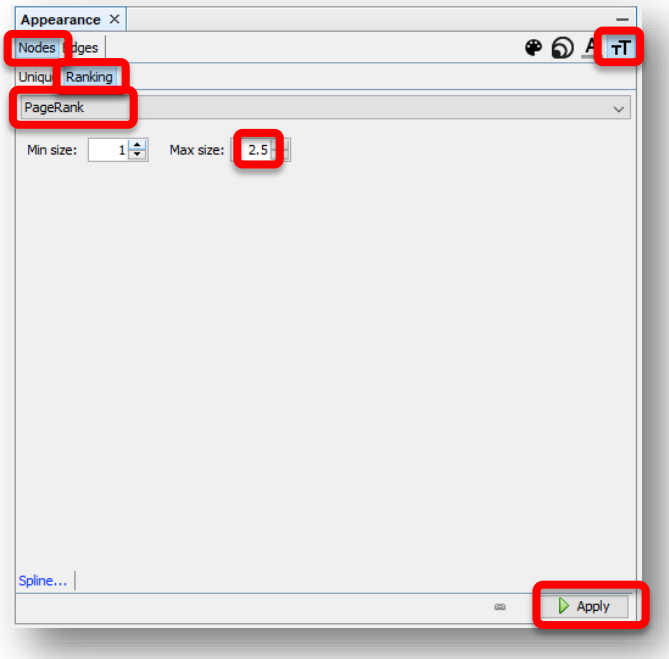




Then click on the black *A* and select *Node size* to set the label size according to the node size.

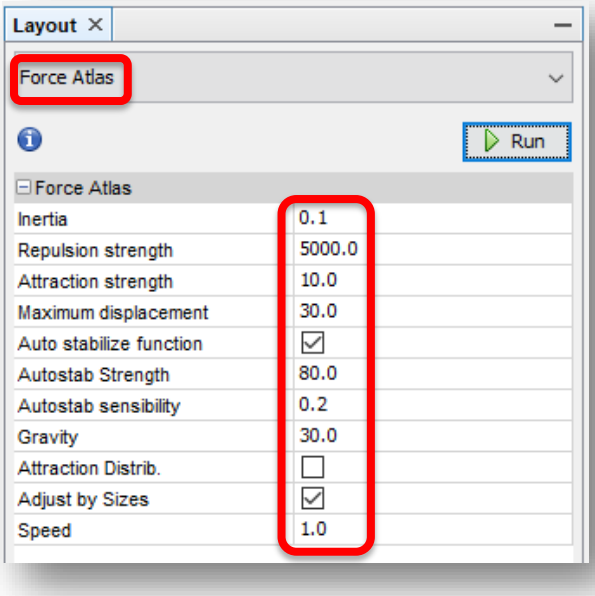


Go back to the *Appearance* panel on the top left, choose *Label Size* (the icon with two Ts) → *Ranking* and choose *PageRank* and set the max size to 2.5. Click *Apply* to confirm. This will emphasise the node labels even further.

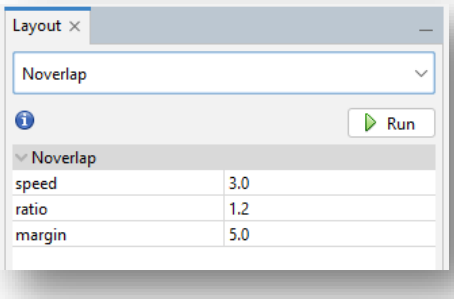


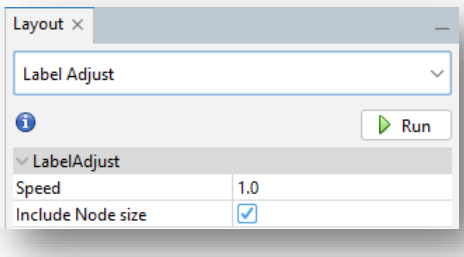
Adjust the Layout

Go to the *Layout* panel and select *ForceAtlas*. Set the tuning to the settings below. These settings will change the layout to better accommodate your network. Click *Run* and wait until the clustering has converged. If you waited a while or some of your nodes are drifting very far away from the centre, you can click on *Stop* to stop the process.



Run the *Noverlap* and *Label Adjust* layout functions to prevent nodes and labels from overlapping, respectively.



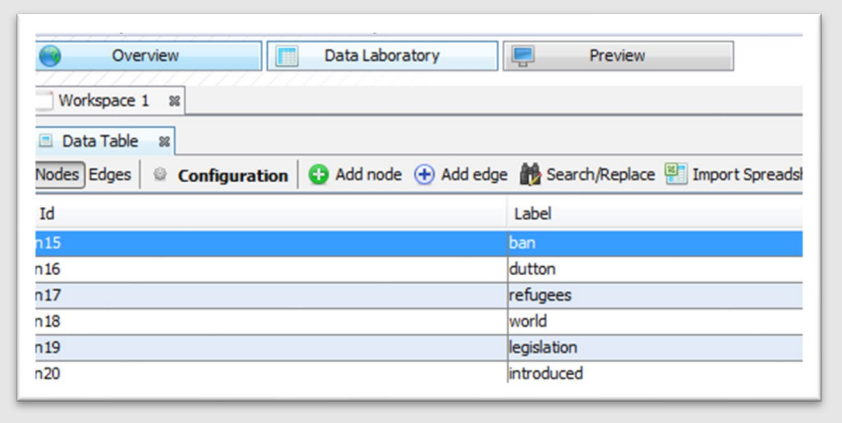


Play around with the settings for Appearance, Visual Properties, and Layout to see what will change.

Apply Filters

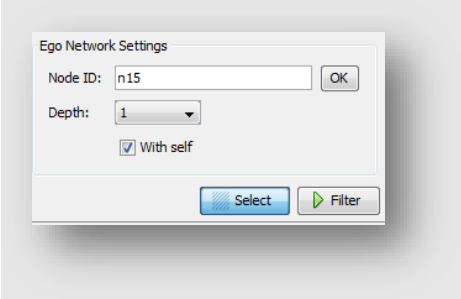
Select a random big node in your graph, right-click, and choose "S*elect in data laboratory*". If the option is greyed-out, click on the *Data Laboratory* tab at the top to refresh the data, then go back to the *Overview* tab again and try again.

Then go to *Data Laboratory* and record or memorise the *Id*. (It will be highlighted in blue.) Then go back to the *Overview* tab.

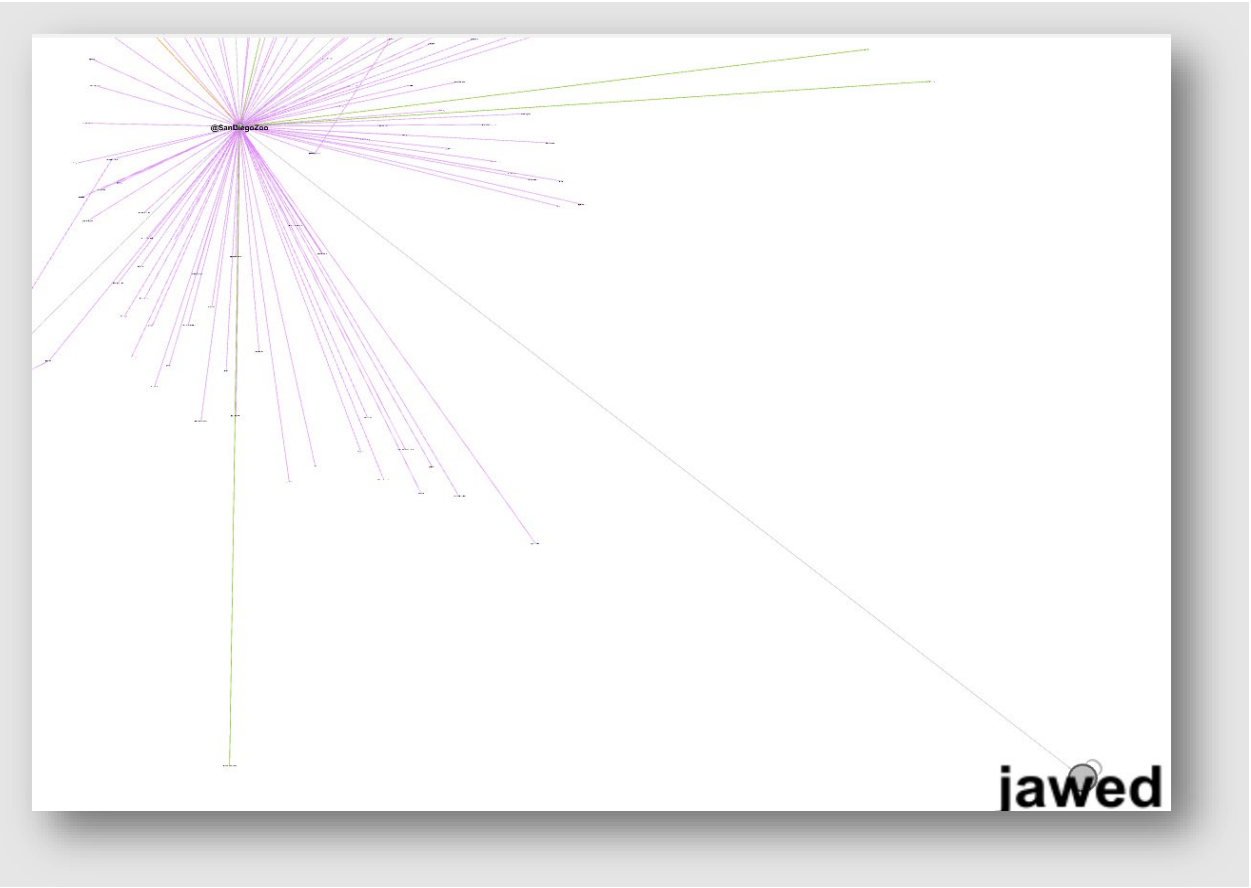


Go to the *Filters* panel on the right side, choose *Topology* and drag *Ego Network* to the *Queries* box below. The Ego Network will show the community for a selected node.

On the *Ego Network Settings* field, you can enter the *Node ID* that you got from the *Data Laboratory.* Click *OK* to confirm.



Run the *Filter* to see its relation to other nodes.



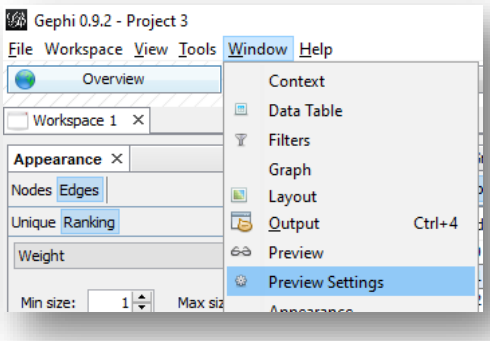
Play around with some of the other filters. You can even combine two or more filters together by using some of the Boolean operators under *Operator.*

When you have selected an interesting sub-graph based on your filters, you can save that sub-graph by going to *File* > *Save As…*

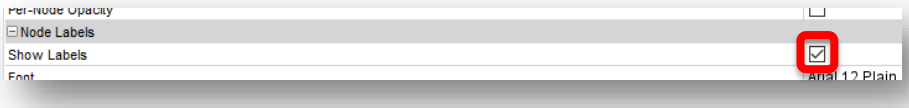
Preview and Export

You can also export any graph from Gephi by going to *File > Export*. However, you should first preview how your graph will be exported.

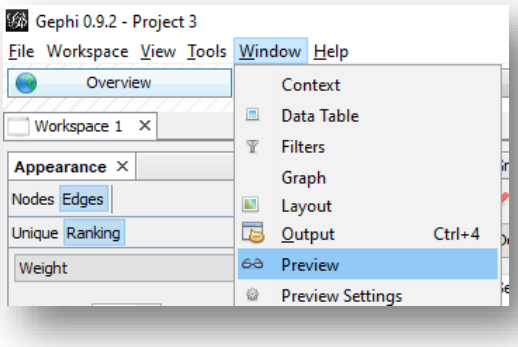
Go to *Window > Preview Settings*.



Tick the box for *Node Labels > Show Labels* and make any other adjustments you like. Then click on *Refresh* on the bottom right.

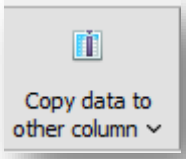


Go to *Window > Preview*.



This will let you preview how your graph will look like as an external image. (If your graph doesn't show up, click on *Reset zoom* at the bottom.)

If for some reason you see the 'ugly' *Labels* for a node instead of the more useful *Screen Name,* you can apply this little workaround: Go to the *Data Laboratory* tab. On the bottom, click on *Copy data to other column*, select *screen_name*, then *Label* in the drop-down list. This will populate the *Label* column with the *screen_name* values.



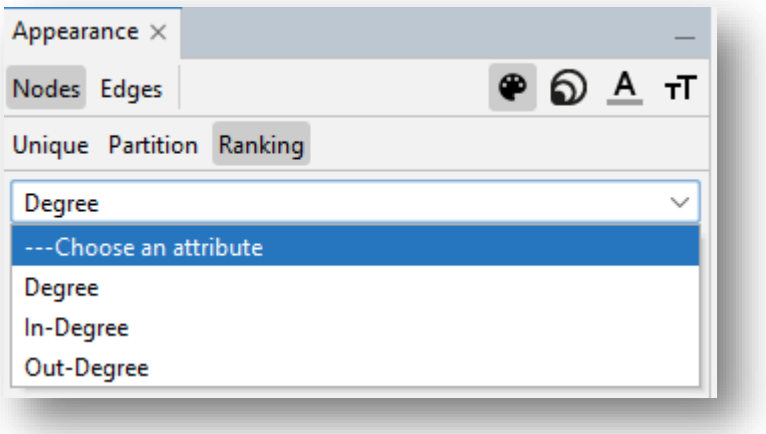
Now you can go back to the Preview and see the result. (Remember to click on *Refresh* and *Reset zoom* in case you cannot see your graph.)

Analyse Centralities

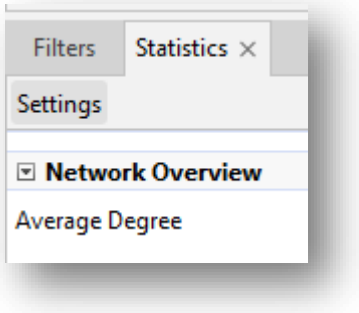
Load the same (or another) GRAPHML file in another workspace window by clicking on *File > Open* in Gephi. Then choose one of the following approaches to calculate & visualise degree, closeness, or betweenness centrality for the graph.

Degree Centrality

To calculate & visualise degree centrality, you can simply follow the steps above. Instead of running *PageRank*, you just choose **Degree**, **In-Degree**, or **Out-Degree** in the *Appearance* panel for node colour, node size, and label size:



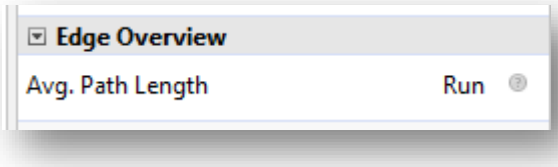
You can also compare the calculation results in Gephi with those in RStudio. Run the *Average Degree* function under the *Statistics* panel.



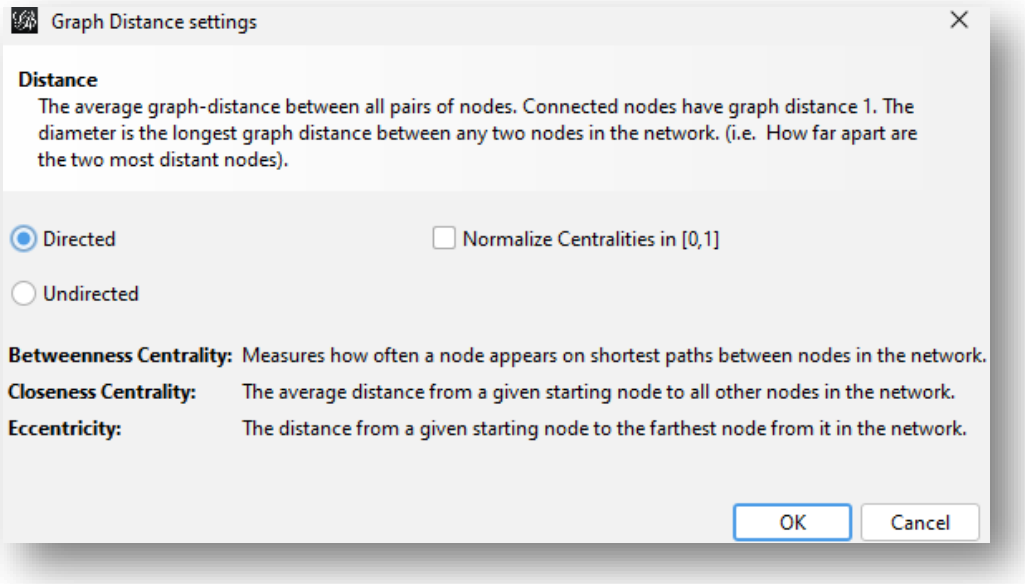
Then go to the *Data Laboratory* tab. You will find three new columns: *In-Degree, Out-Degree,* and *Degree*. (If you can't see them, make sure that you are viewing the data for *Nodes,* not *Edges.*) You can sort each of the columns by clicking on the header. Now run the Lab 3-1 script for your network graph. You should be able to see the same results!

Closeness Centrality

To calculate & visualise closeness centrality, run the *Average Path Length* function in the *Statistics* panel.



Choose '**Directed**' or '**Undirected**' based on the type of graph you loaded (Actor vs Semantic) and leave 'Normalize Centralities in [0,1]' unticked. Then click 'OK'.



Then follow the steps above to tune your visualisation.

If you go to the *Data Laboratory* tab, you will find a new column titled *Closeness*. The scores in that column may not match exactly your scores in RStudio for closeness, due to some difference in the underlying calculation, but generally you should be able to see a similar ordering of nodes.

Betweenness Centrality

Since you already calculated betweenness centrality for the previous step, you can go straight to tuning your visualisation.

If you are curious to see the shortest path(s) between two nodes, you can use the *Shortest Path* tool. Select it and then click on two nodes to see the shortest path(s) – if a path exists between them:

If you go to the *Data Laboratory* tab, you will find a new column titled *Closeness*. The scores in that column may not match exactly your scores in RStudio for closeness, due to some difference in the underlying calculation, but generally you should be able to see a similar ordering of nodes.

Analyse Communities

Load the same (or another) GRAPHML file in another workspace window by clicking on *File > Open* in Gephi. Then choose one of the following approaches to calculate & visualise the communities for your graph.

Louvain Algorithm (Modularity)

To calculate & visualise the communities for your graph according to modularity classes, run the *Modularity* function in the *Statistics* panel. Then follow the steps above to tune your visualisation.

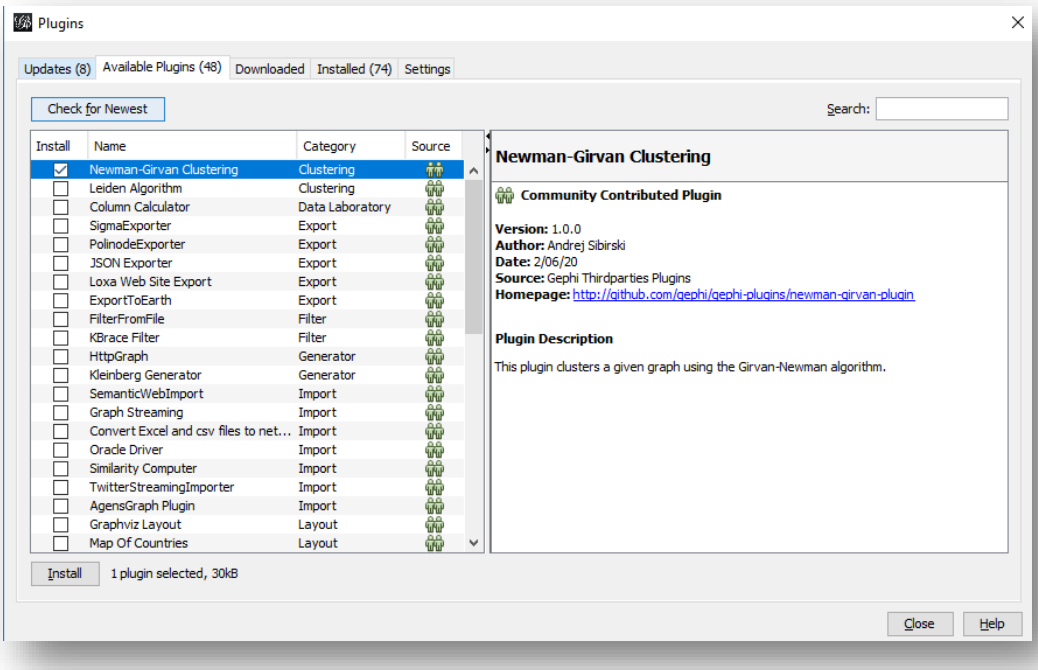
Compare this visualisation with the basic plot you created in RStudio for Lab 3-2. Can you see similar communities in Gephi as in the RStudio plot?

Girvan-Newman Algorithm (Edge Betweenness)

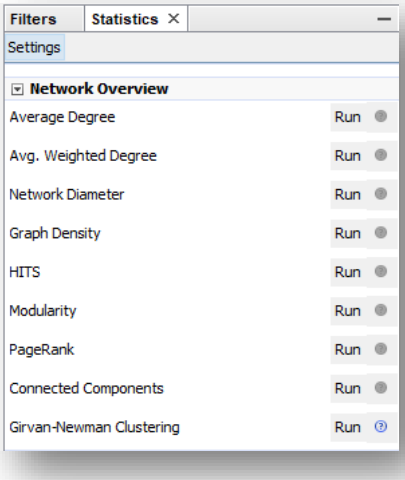
To calculate & visualise the communities for your graph according to edge betweenness, you will need to install the 'Newman-Girvan Clustering' plugin.

Note: In order to install the plugin, you will need to restart Gephi in the process. Thus, if you need to save any of your work, please do so first.

In Gephi, go to 'Tools' > 'Plugins', then select 'Available Plugins' and choose 'Newman-Girvan Clustering'. Click on 'Install'.



After you installed the plugin, you can find it under Statistics.



Run the algorithm with the default settings. It may take a while depending on the size of your graph. Then follow the steps above to tune your visualisation.

Compare this visualisation again with the basic plot you created in RStudio for Lab 3-2. Can you see similar communities in Gephi as in the RStudio plot?