**Network Analysis Tutorial**

This tutorial is intended to provide a brief introduction to how to use perform network analysis with a dataset derived from a historical corpus.

You can jump ahead to other sections here:

[Introduction 1](#_Toc72244417)

[Start a new project and import data into the project. 2](#_Toc72244418)

[Explore the data. 3](#_Toc72244419)

[Change node aesthetics. 5](#_Toc72244420)

[Set node labels. 6](#_Toc72244421)

[Select a network layout. 8](#_Toc72244422)

[Experiment with filters. 8](#_Toc72244423)

[Community detection with the Louvain modularity algorithm. 10](#_Toc72244424)

[Appendix: Svoboda Diaries Project Named Entity Recognition Dataset v0.8 12](#_Toc72244425)

# Introduction

To render the network, we use Gephi, an open source network analysis software. While there are other softwares and methods available to perform this task, such as Cytoscape and Graphvis which are other open source softwares as well as the Networkx library for Python and the igraph library for R, we found that Gephi was best suited for our needs as it provides a good balance between simple usability and powerful algorithmic capabilities.

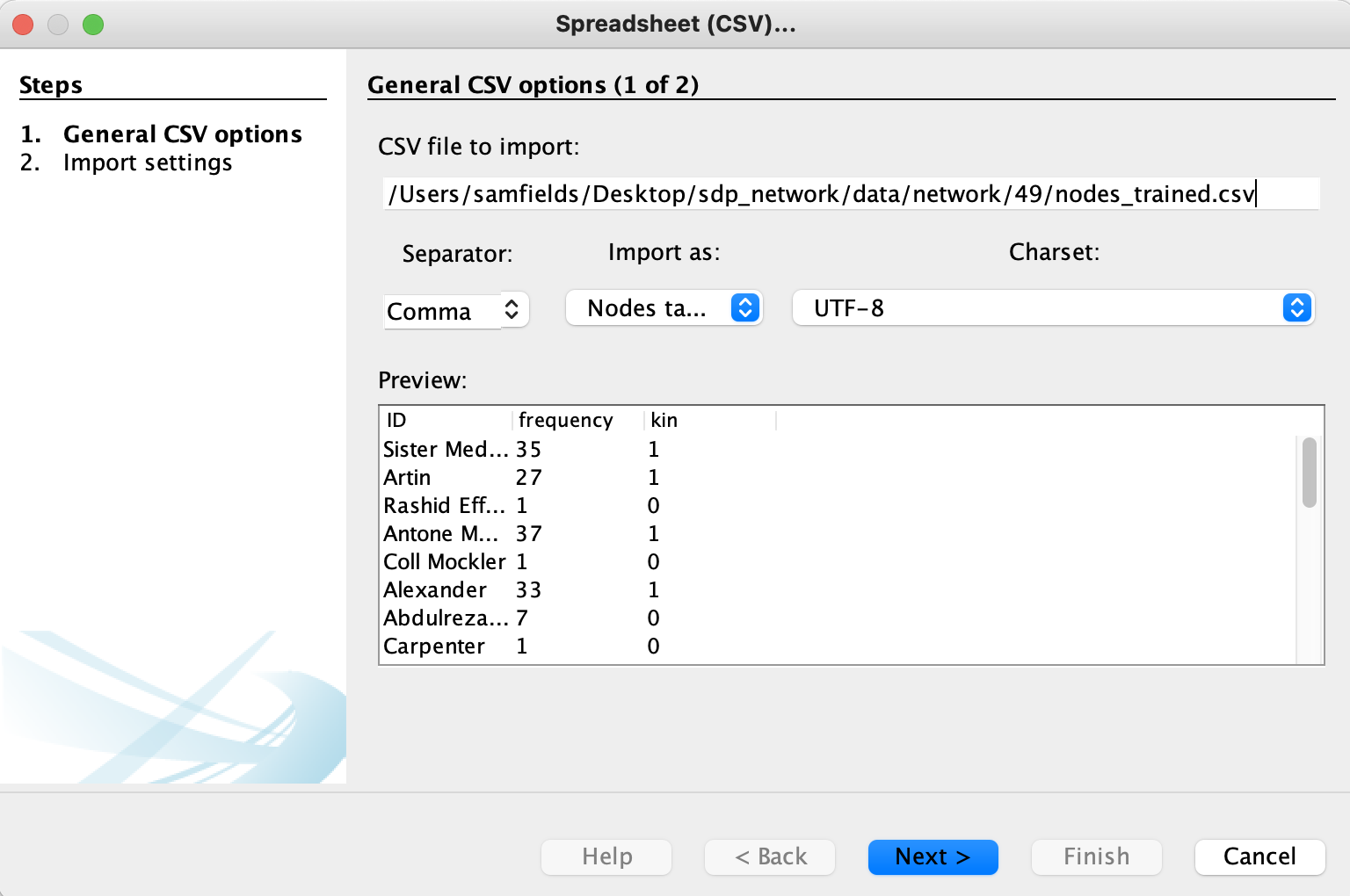
Please download Gephi here: <https://gephi.org/users/download/>.

In this tutorial, we will focus on a dataset constructed from diary 49. To extract the persons mentioned in diary 49, we performed named entity recognition, a natural language processing technique to automatically identify named entities, defined as persons, organizations, place names, and numeric expressions such as dates or amounts from texts.

# Start a new project and import data into the project.

1. Start a new project (File > New Project).
2. Import data.
   1. Click on “Data Laboratory,” the middle bottom at the top of the window
   2. Click on “Import Spreadsheet”.
3. We are going to start out by importing the nodes, or persons mentioned in the diaries.

Select “nodes\_trained.csv”, makes sure the “Separator” is set to “comma,” “Import as…” is set to import as a nodes table, and that “Charset” is set to “UTF-8”

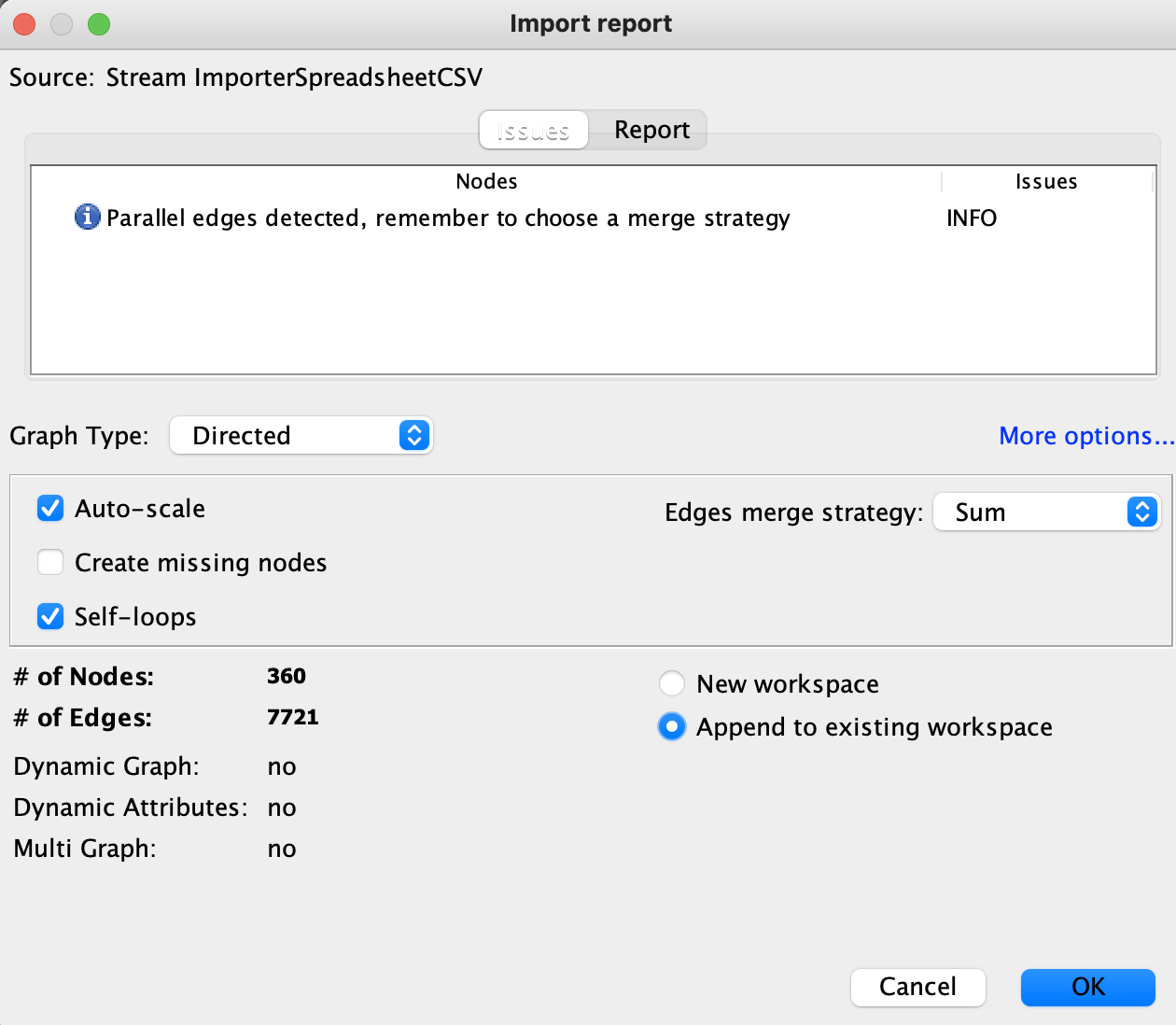


Hit “Next”. The next screen allows you to select the columns to import.

1. Next, we import the edges table to identify the relationship between the nodes (persons).

Select “edges\_trained.csv” and import as an edges table.

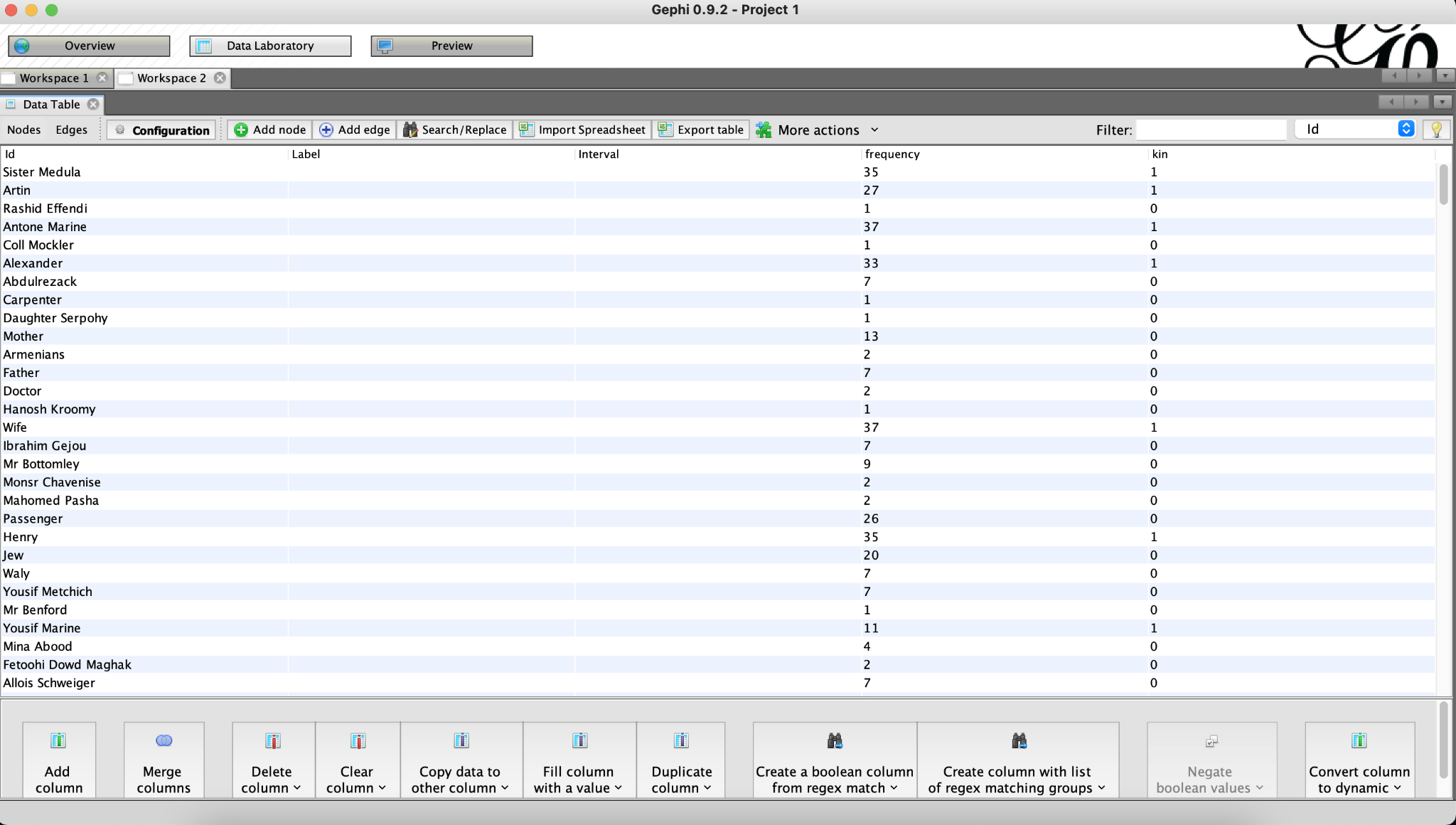
By default, “New workspace” is selected. Click, “Append to existing workspace”. This puts the edges in the same workspace as the nodes that you just added.



# Explore the data.

You just imported two files into Gephi. Let’s explore them. Click on the “Data Laboratory” tab, then “Nodes”.

The first file contains the persons and looks like this:



These are the columns that you should see:

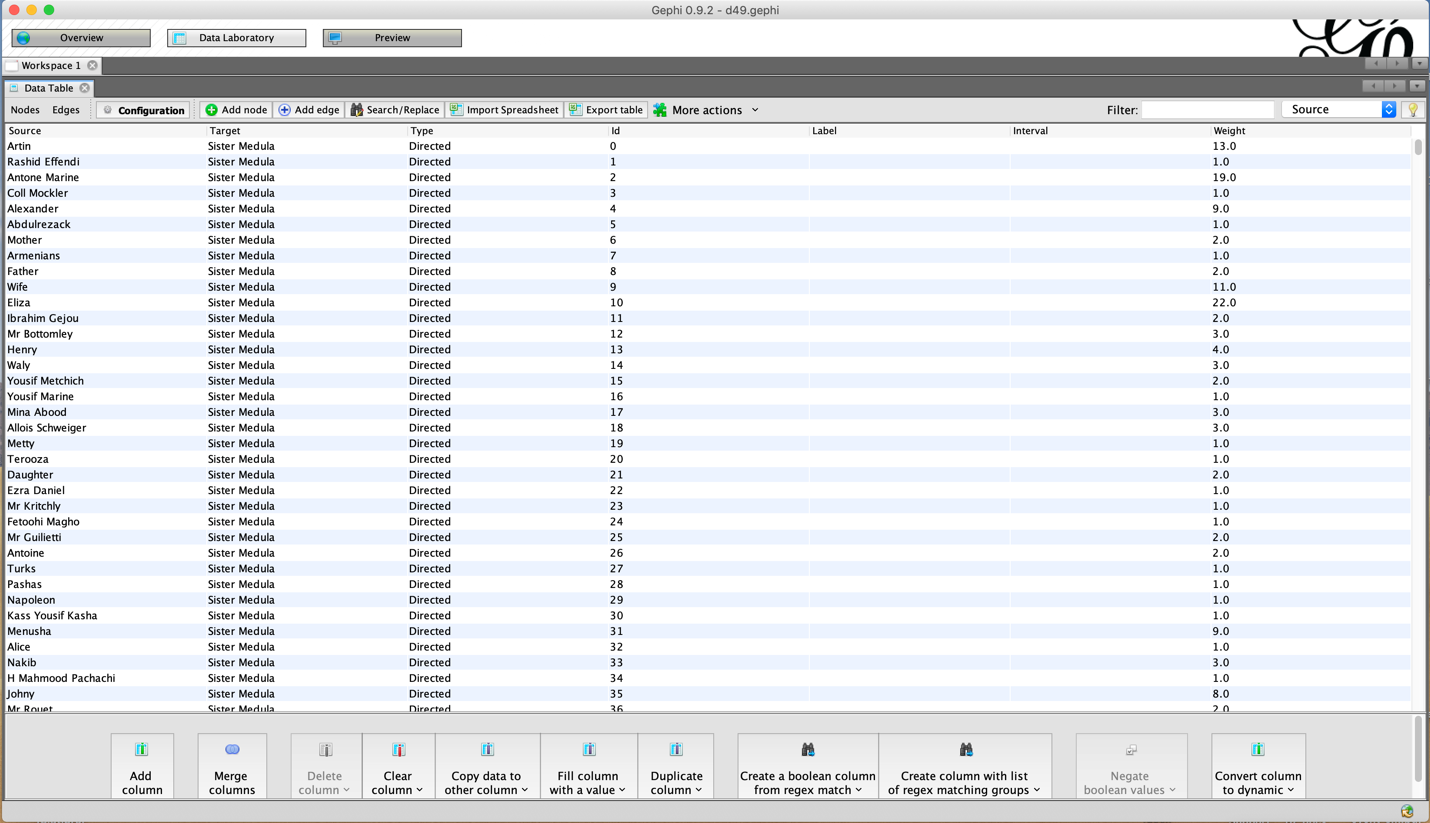
**ID:** Person name.

**Frequency:** Number of times that the person appears in the given diary.

**Kin:** 1 if considered a kin of Joseph Svoboda, 0 otherwise.

Now let’s take a look at the edges file. The edges file depicts connections between two people. In this case, a connection is defined as two persons being mentioned on the same day. Thus, two people being mentioned on the same day served as a proxy for social connection.

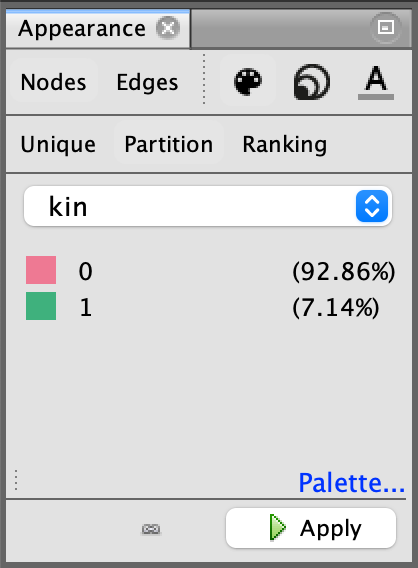
This is what the edges should look like:



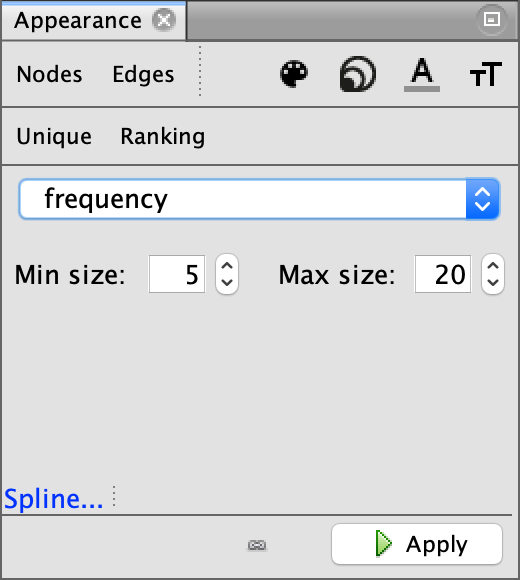
The edge weight is the number of times that two people are mentioned on the same day.

# Change node aesthetics.

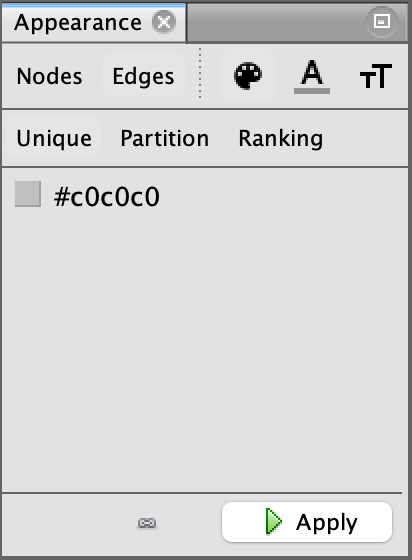
1. Set node color: Under the “Appearance” tab on the top left of the window, first click the paint board icon to the left of the circle. Select “Nodes”, then “Partition”. For attribute, select “kin”. Then select “Apply”.



1. To change the size of the nodes, click the circle icon, then “Ranking,” and select the frequency attribute. Select a Min size of 5 and a Max size of 20.



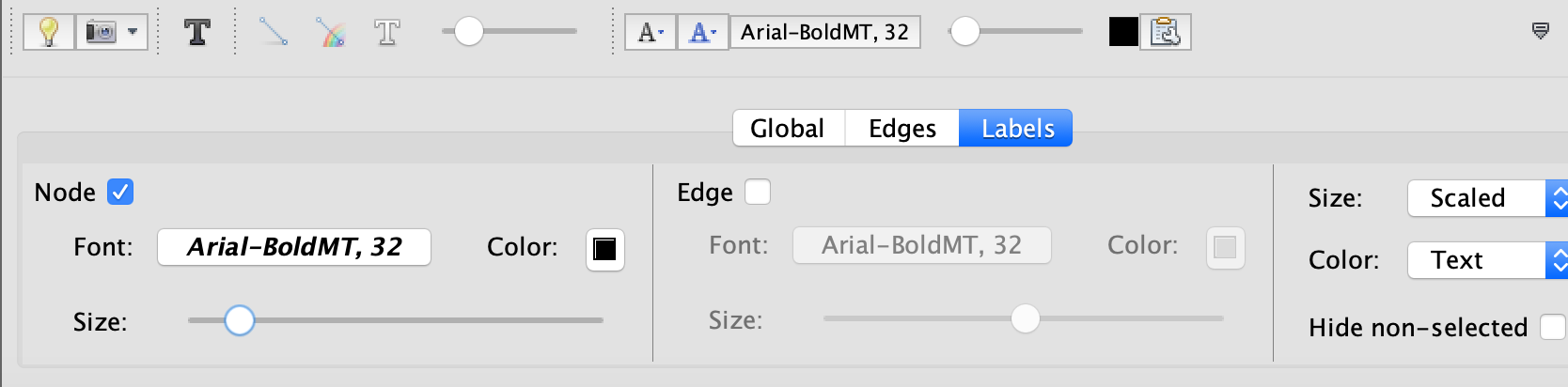
1. To make the edges grey, click “Edges,” then “Unique,” and hit “Apply.”.



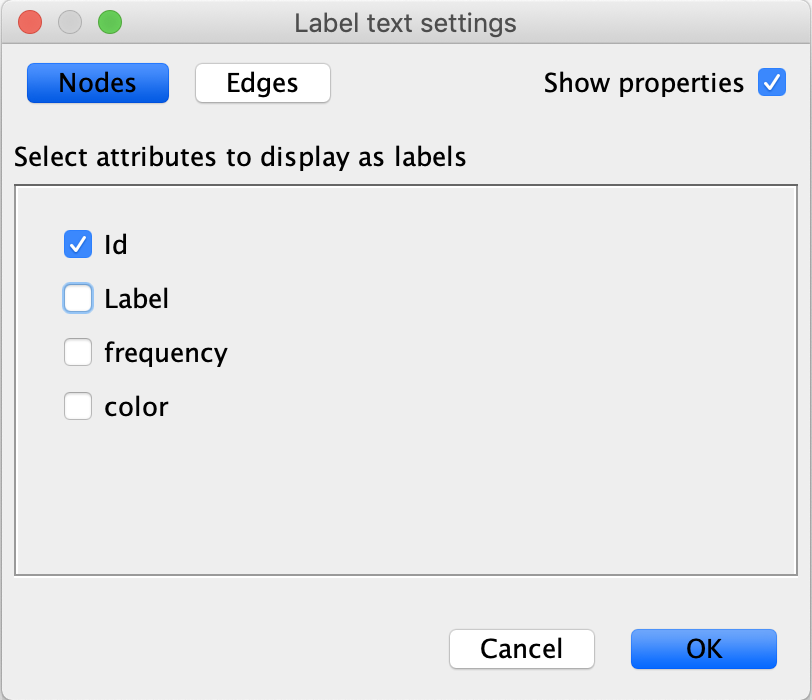
# Set node labels.

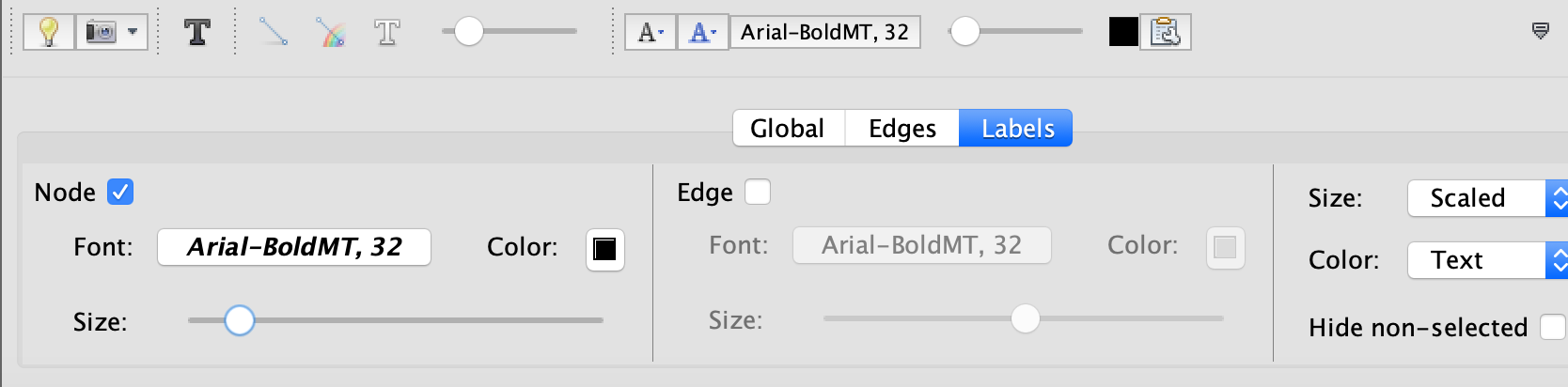
1. Add node labels.

Click on the grey triangle-bar icon in the bottom right hand corner of the application. Then click “Labels”, and “Node”.



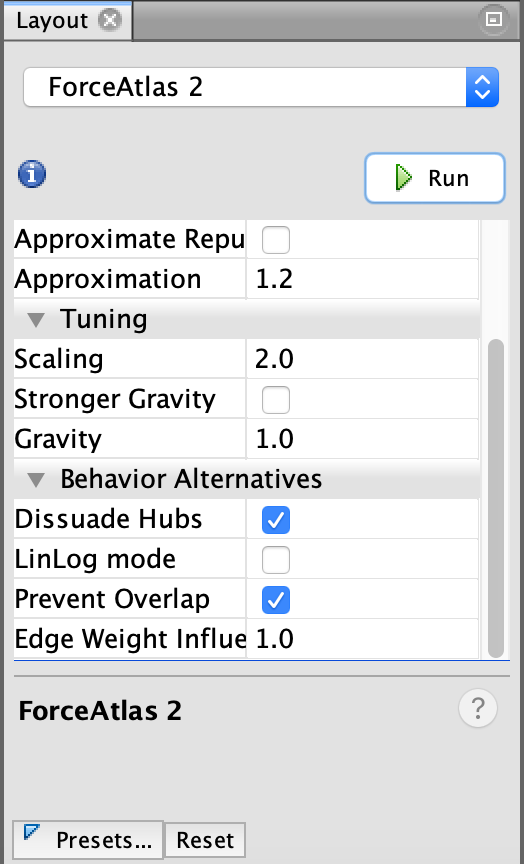
Click on the “Configure” clipboard to set the field to use for the label.



Move the slider to the left so that the labels are not so big.

# Select a network layout.

You want to choose a layout that emphasizes the network topology, or features, that you want to emphasize. For now, we are going to use: “Force Atlas 2”, which is a layout in which nodes that are highly connected to one another tend to appear closer together.1 Click “Dissuade Hubs” and “Prevent Overlap” to prevent overlap between nodes and make your network easier to read.



If you find that your network labels are overlapping and are hard to read, you can then apply the “Label Adjust” layout through a similar process. This will adjust the nodes such that the labels are no longer overlapping.

# Experiment with filters.

There are filters on the right hand side which we can use to filter out nodes by certain attributes.

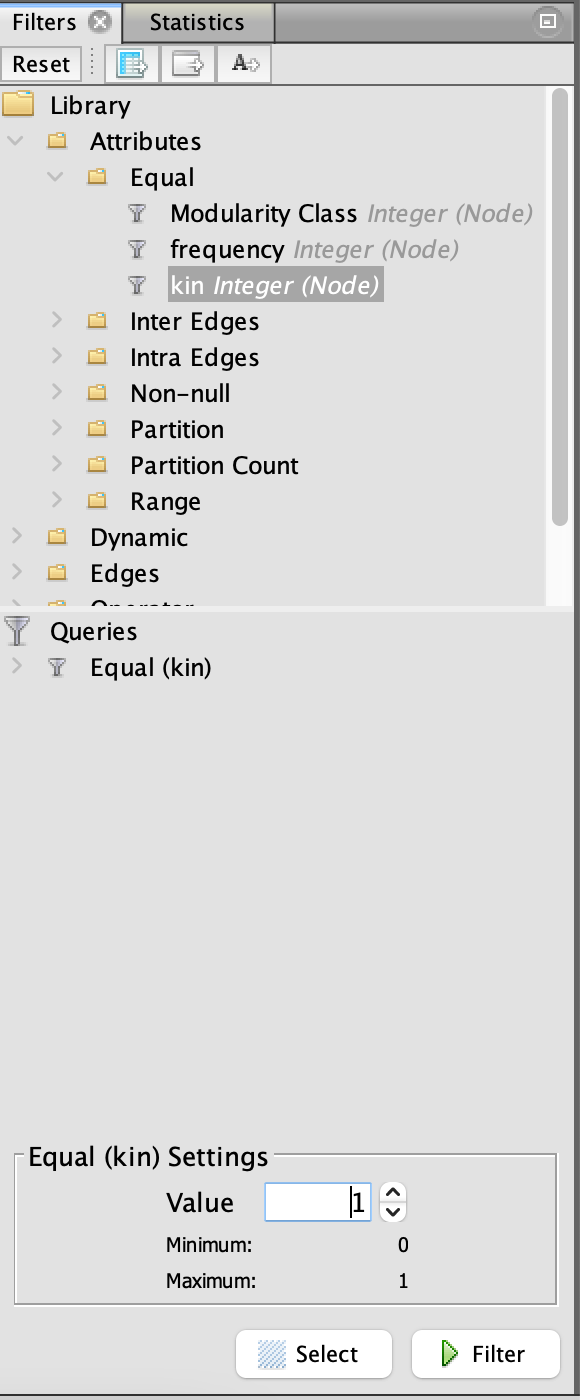
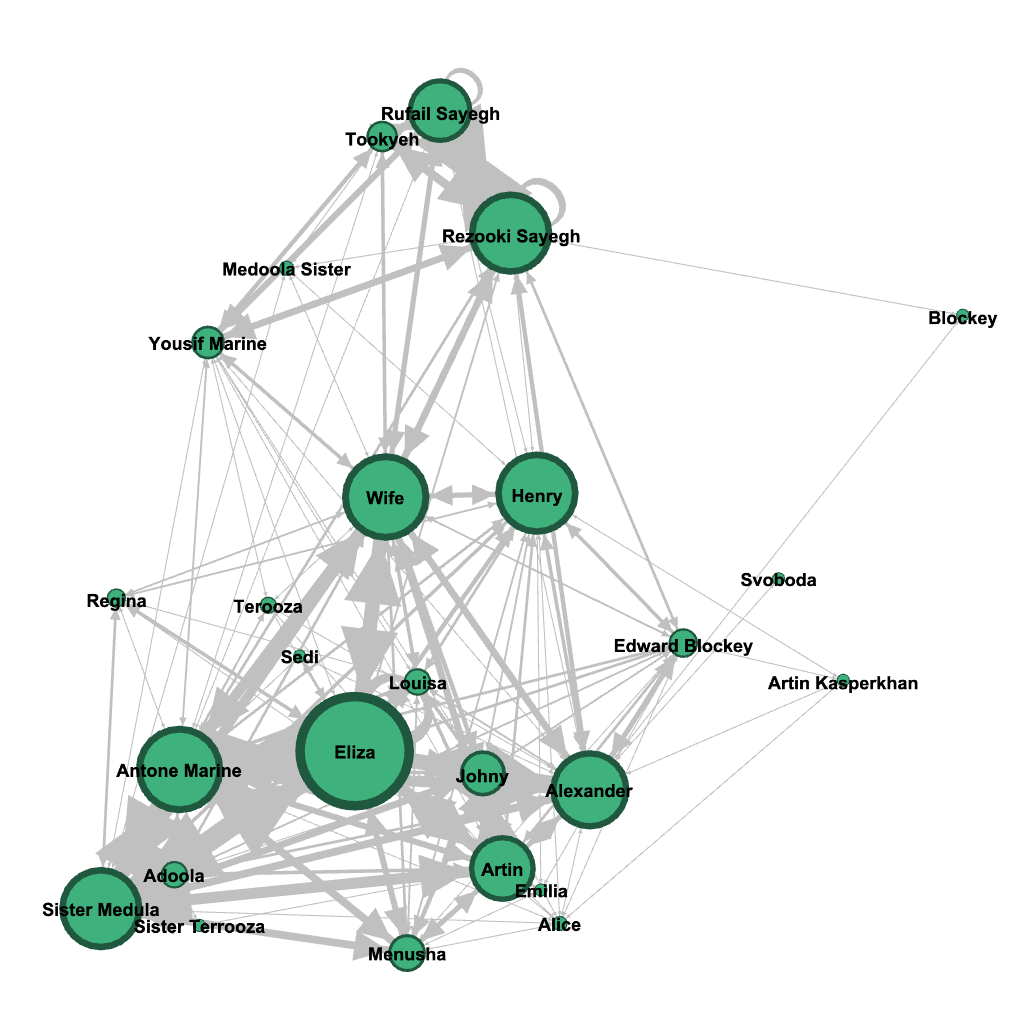
Let’s use the filters to visualize only Joseph’s kin.

Click on the “Filters” tab on the top-right corner next to “Statistics”. Then click into the “Attributes” folder, then “Equal”. Select “color” and drag it to the “Queries” section below. Set “Value” setting on the bottom-right equal to “1” and then hit “Filter”.

Only the green nodes should be depicted, which identifies those that are kin of Joseph Svoboda.

If you find that you are having difficulty seeing the “Filters” tab, as it is hidden behind the white background of the graph, you should make sure to hit the window button for the “Filters” and “Statistics” tab which looks like the icon included below and is located at the top right of the screen.

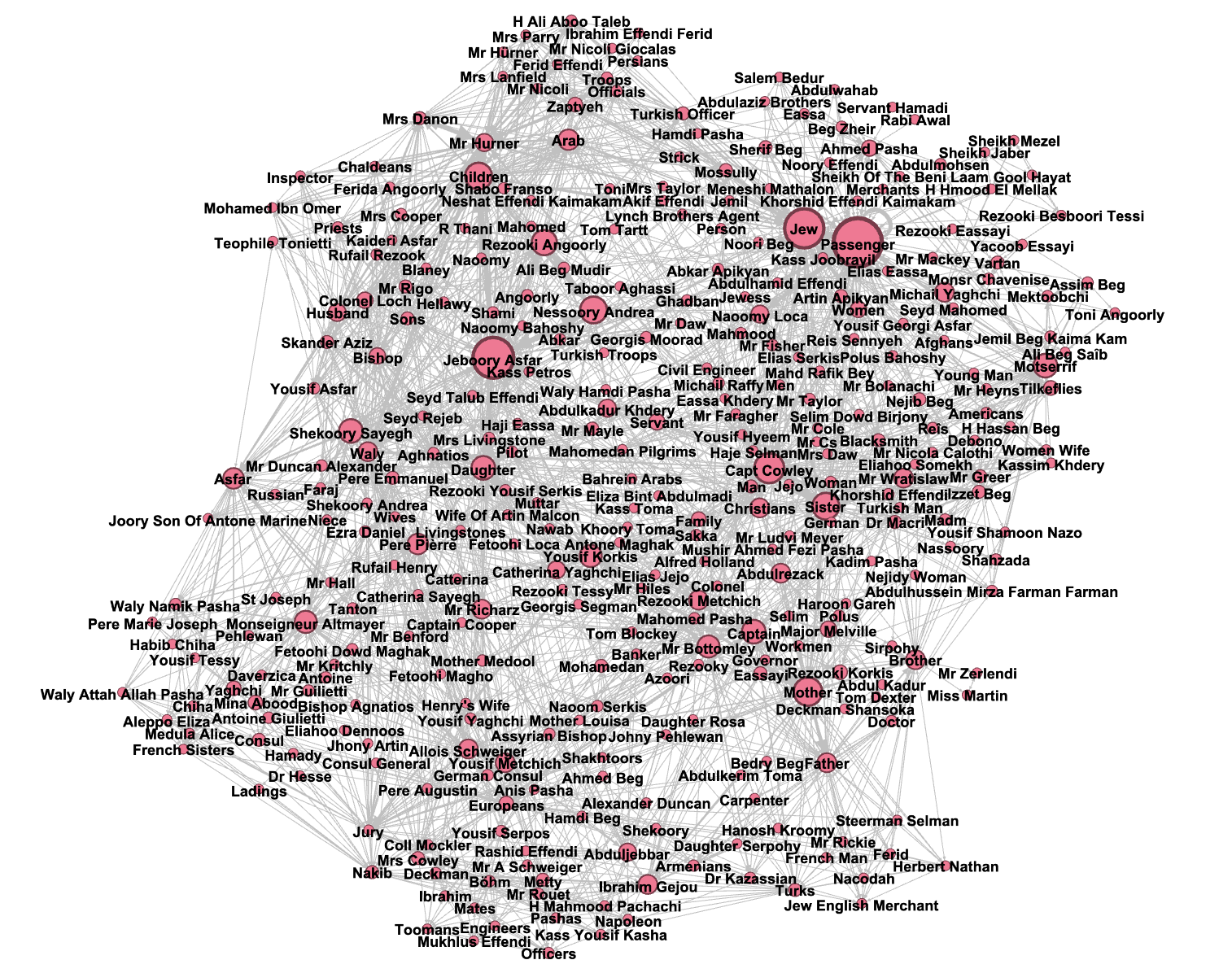




To visualize those who are non-kin:

Set Value equal to “0” and then hit “Filter”.

Only the pink nodes should be depicted, identifying those who are not kin of Joseph.

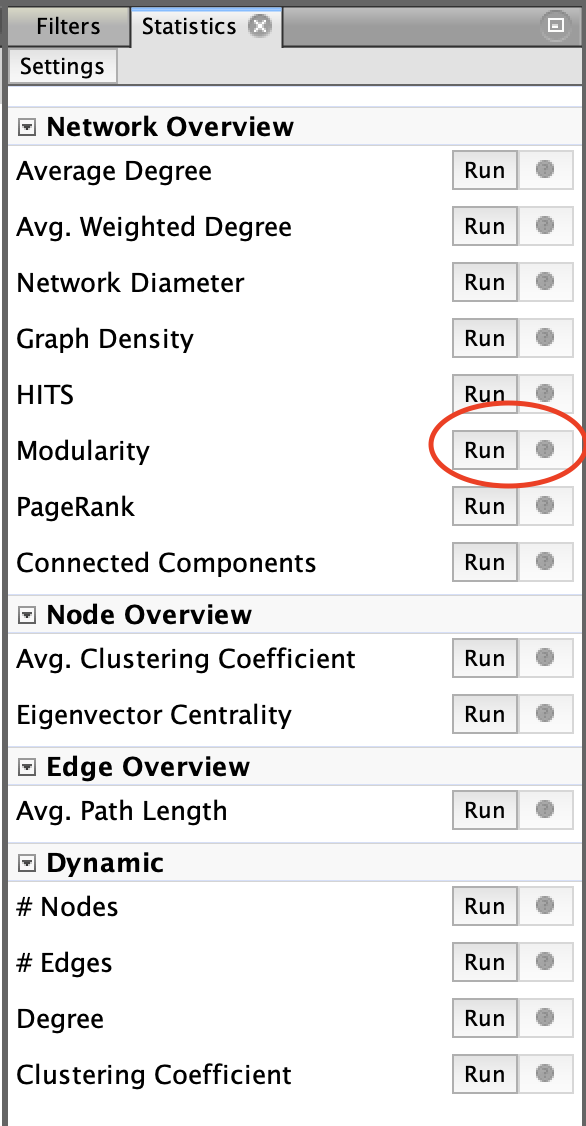
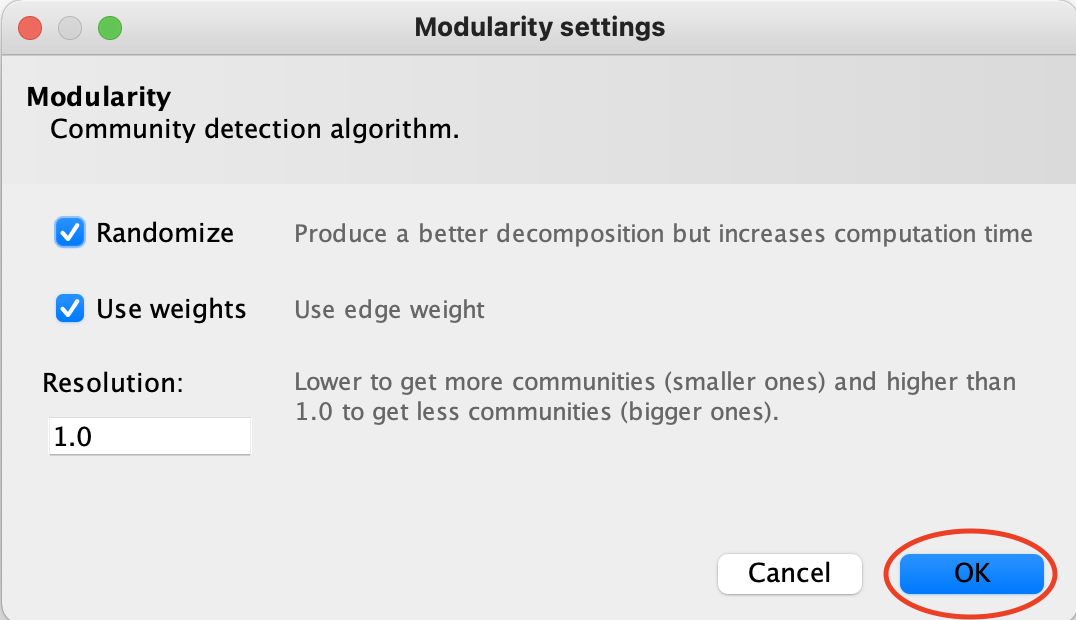


# Community detection with the Louvain modularity algorithm.

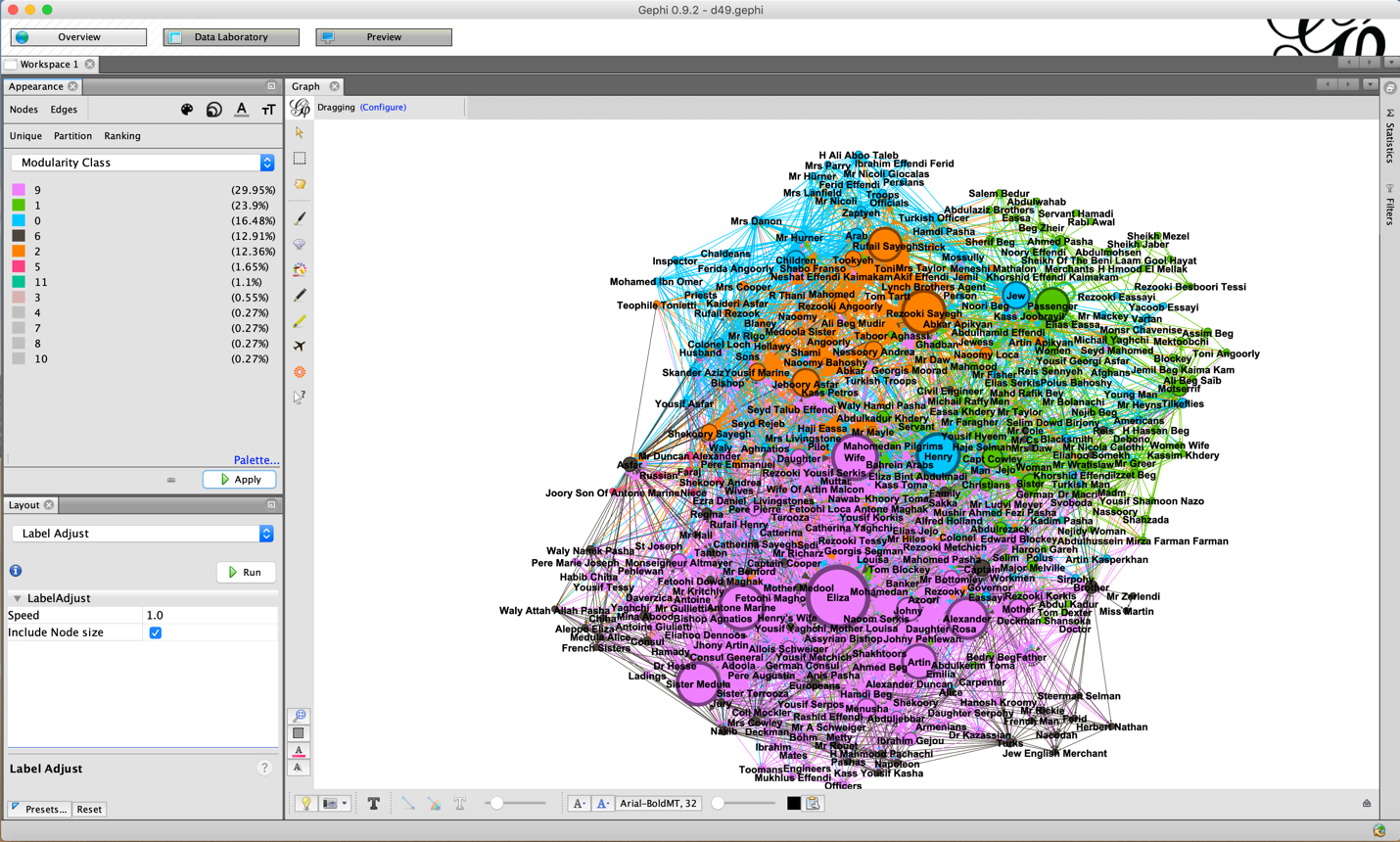
Applying the Louvain modularity algorithm allows for the extraction of community structure in large networks. It uses a hierarchical clustering algorithm to iteratively combine clusters of “close” nodes, which is affected by neighboring connected nodes as well as other aspects of the network to identify these communities. We will demonstrate how we can use this algorithm to visualize the possible community structure present in the daily life of Joseph Svoboda.

You can run the Louvain modularity algorithm on the network by going to the “statistics” tab, which is located next to the “filters” tab, and then clicking the “run” button for “Modularity.”

This will then open a new window providing you the opportunity to control certain parameters of the algorithm. For our task, we will use the default values. So, click “OK.”

Running this algorithm adds a “Modularity Class” attribute to each node in the nodes data table in Gephi. Thus, to visualize each node's assigned modularity class, following a similar process as the kinship visualization, we will color each node by its modularity class. First, click “Nodes” under the appearance tab, and then click the paint palette. Then, click “Partition” and select “Modularity Class” as the feature to partition by. Finally, hit “Apply.”



# Appendix: Svoboda Diaries Project Named Entity Recognition Dataset v0.8

The diaries of Joseph Mathia Svoboda capture 40 years of trade on the Tigris, detailing his day to day journeys as a steamboat purser during the late 19th and 20th centuries, specifically between the cities of Baghdad and Basra. Today, 19 diaries are available in the form of scanned, digital images of the original diary pages, and 29 are available in the form of scanned, digital images of typed diary transcriptions. Currently, diaries 47-49 have been fully digitized and published and are available through the University of Washington.

This dataset was produced by performing named entity recognition, a natural language processing technique, to automatically identify names in the diaries of Joseph Mathia Svoboda. The dataset is comprised of two files:

1) Nodes file containing the following columns:

**ID:** Person name.

**Frequency:** Number of times that the person appears in the given diary.

**Kin:** 1 if considered a kin of Joseph Svoboda, 0 otherwise. In this context, kin are defined as… (*Camille, please fill in*).

Please note the following:

* There are times when names are spelled differently either by Joseph or in the transcriptions, but refer to the same person (e.g., Colonel Lock and Coll. Lock). These have generally been mapped to one name referring to the same person. In other words, in the example here, all names referring to Colonel Lock would appear under “Colonel Lock”.
* When Joseph refers to a person by their first name, these instances have generally been left as is, without resolving them a full name which may also appear in the dataset (e.g., “Ibrahim”, in the case of “Ibrahim Gejou”).

2) Edges file containing the following columns:

**Source:** Person A name.

**Target:** Person B name.

**Weight:** The number of times that person A appears on the same day in the diary as person B.

License: This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-sa/4.0/).