Using filters

Table of Contents

last modified: 2017-02-19

[gephi logo 2010 transparent] | *gephi-logo-2010-transparent.png*

download a network file for practice

download this zip file and unzip it on your computer.

or use this direct link: https://tinyurl.com/gephi-tuto-3

You should find the file `miserables.gexf` in it. Save it in a folder you will remember (or create a folder specially for this small project).

This file contains a network representing "who appears next to whom" in the 19th century novel *Les Misérables* by Victor Hugo[1].

A link between characters A and B means they appeared on the same page or paragraph in the novel.

The file name ends with ".gexf", which just means this is a text file where the network information is stored (name of the characters, their relations, etc.), following some conventions.

open the network in Gephi

- open Gephi. On the Welcome screen that appears, click on `Open Graph File`
- find `miserables.gexf` on your computer and open it

[gephi welcome screen open graph en] | *en/gephi-welcome-screen-open-graph-en.png*

Figure 1. welcome screen

A report window will open, giving you basic info on the network you opened:

[opening file report en] | *en/opening-file-report-en.png*

Figure 2. report window

This tells you that the network comprises 74 characters, connected by 248 links.

Links are undirected, meaning that if A is connected to B, then it is the same as B connected to A.

The report also tells us the graph is not dynamic: it means there is no evolution or chronology, it won’t "move in time".

Click on `OK` to see the graph in Gephi.

[result miserables] | *result_miserables.png*

Figure 3. The network we will use

getting a sense of the attributes in the data laboratory

We can switch to the data laboratory to see the underlying data:

[Switching to the data laboratory] | *Switching-to-the-data-laboratory.png*

Figure 4. Switching to the data laboratory

We see that the nodes of the network have many attributes. In particular, each have a Gender and a measure of how central they are:

[Nodes attributes.] | *Nodes-attributes..png*

Figure 5. Nodes attributes.

This is the list of edges (relations) in the network. Notice that they have a "weight" (a "strength").

[Edges attributes.] | *Edges-attributes..png*

Figure 6. Edges attributes

discovering the filter panel

In the overview, make sure the Filter panel is displayed:

[Making the Filter panel visible.] | *Making-the-Filter-panel-visible..png*

Figure 7. Making the Filter panel visible.

How the Filter panel works:

[Workflow of filters] | *Workflow-of-filters.png*

Figure 8. Workflow of filters

An example: hiding edges with weight lower than 2

[How to use filters.] | *How-to-use-filters..png*

Figure 9. How to use filters.

[filter edge weight 1 en] | *en/filter-edge-weight-1-en.png*

Figure 10. Filtering out edges with weight lower than 2.

view online animation - link: https://tinyurl.com/gephi-tuto-2

When you are finished using a filter in the zone, right click on it and select "remove".

combining 2 filters

One filter is applied AFTER this other:

The first filter to be applied is NESTED (placed inside) the second one as a "subfilter"

Which filter should be placed inside which? Let’s look at different examples:

1. Case when the placement of filters makes no difference

Goal: Keeping on screen only the female characters which have a tie (an edge, a relation) of at least strength 2.

→ place the filter "edge weight" inside the filter "Gender":

[Filter on the Gender attribute] | *Filter-on-the-Gender-attribute.png*

Figure 11. Filter on the Gender attribute

[Filter on edge weight] | *Filter-on-edge-weight.png*

Figure 12. Filter on edge weight

[Keeping only female characters with at least 2 ties] | *Keeping-only-female-characters-with-at-least-*

2-ties.png

Figure 13. Keeping only female characters with at least 2 ties

[filter edge weight gender partition1 en] | *en/filter-edge-weight-gender-partition1-en.png*

Figure 14. Keeping only female characters with at least 2 ties

view online animation - link: https://tinyurl.com/gephi-tuto-1

In this case, it was equivalent to:

- nest the "Gender" filter inside the "Edge weight" filter or
- nest the "Edge weight" filter inside the "Gender" Filter

→ The result was the same (the network on screen is identical in both cases)

2. Case when the placement of filters makes a difference

Here, we want to visualize:

- only the nodes which have **less than** 10 relations <1>
- and among these, only those which form the "main island" of the network (we want to hide small detached groups of nodes) <2>

① in technical terms, nodes with a `degree` of less than 10.

② in technical terms, we are looking for the `giant component`

[filter degree range 1 en] | *en/filter-degree-range-1-en.png*

Figure 15. Filter on degree

[filter giant component 1 en] | *en/filter-giant-component-1-en.png*

Figure 16. Filter on giant component

We will see that the placement on the filters in the zone will make a difference.

First, let us place the filter on giant component **inside** the filter on degree:

[filter order 1 en] | *en/filter-order-1-en.png*

Figure 17. Filters in one configuration

In this first case,

- only the giant component of the network was made visible.

→ Since the network was just one big connected "island" to start with, it did not change a thing.

- then, all characters with more than 10 relations where hidden

→ this hides nodes which were connecting with many others, so that we end up with many groups, disconnected from each others.

Now instead, placing the filter degree **inside** the filter on giant component:

[filter order 2 en] | *en/filter-order-2-en.png*

Figure 18. Same filters in another configuration

In this second case,

- starting from the complete network, all characters with more than 10 relations where deleted.

→ this created a network made of many disconnected groups of nodes

- then the giant component filter is applied,

→ which had for effect to hide small groups, to keep in view only the biggest group of connected nodes.

> ⚠ In summary: be careful how you apply several filters at once, this might have an effect on the logic of filtering.

filter operators

1. The MASK operator

Imagine you are interested in the female characters of the novel "Les Miserables".

- you are interested in female characters and the relations among them
- you are interested in the relations between female characters and male characters
- you are **not** interested in the relations between male characters

How to display this?

The MASK operator applied on the gender partition filter enables you to:

- show all characters
- relations between female characters
- *and relations between male and female characters*
- *but masking male-male relations*

[operator mask 1 en] | *en/operator-mask-1-en.png*

Figure 19. Using the MASK operator

It is also possible to hide / show only some of the directed relations between the visible graph and the filtered out graph:

[operator mask 2 en] | *en/operator-mask-2-en.png*

Figure 20. Parameters of the MASK operator

2. The UNION operator

Imagine you are interested in the characters with names starting with "L" or "J" in "Les Miserables".

How to display only these characters?

We will need to apply filters on the `Label` of the nodes, which contains the names of the characters.

However, looking at the "catalogue" of filters, we see no filter on `Label`. The reason is that `Label` is an internal property of nodes, inaccessible to filters.

So we must first copy the Labels of the nodes in a new attribute, which we will be able to apply a filter on.

Let’s switch to the data laboratory and add this attribute:

[Adding a column for Names] | *Adding-a-column-for-Names.png*

Figure 21. Adding a column for Names

[Copying to this new column] | *Copying-to-this-new-column.png*

Figure 22. Copying to this new column

We now have an attribute called "Name" that we can find in the Filters:

[New filter available] | *New-filter-available.png*

Figure 23. New filter available

This is how the filter on Name and its parameters look like in the zone:

[filter name 1 en] | *en/filter-name-1-en.png*

Figure 24. Name Filter

To recall, we want to show only the characters which name start with "L" or "J". Let's start with the "L" characters.

We need to find the names which match the pattern `Start with an L`. The way to describe a pattern in text is called a "regular expression".

Said differently, *a regular expressions (also called "regex") is a convenient way to express a pattern we search for in a text*.

Regular expressions can become very sophisticated. But here, we need just a simple one:

```
L.*
```

Let's examine what the L, the dot and the star mean.

- the letter "L" means we want names starting with this first letter
- . the dot means: any character
- * the star means: the previous character, repeated any time.

So: "select nodes which have a name starting with L, followed by any character, in any number"

Please note that you need to check the box "regex":

[filter name 2 en] | *en/filter-name-2-en.png*

Figure 25. Using a regular expression in a filter

When the filter is applied, only the characters wit a name starting with L will be displayed:

[filter name 3 en] | *en/filter-name-3-en.png*

Figure 26. Using a regular expression in a filter

How to filter characters with a name starting with the letter "L" or "J"?

We could rely on a more complex regular expression to do this:

```
[LJ].*
```

Meaning: "select nodes which have a name starting with L or J, followed by any characters"

But we can also rely on 2 filters: one for L, one for J. Nesting one inside another would not work, it would mean:

"show nodes which start with an L, and among them, only those which start with a J"

→ no node can meet this condition, so they would all be invisible.

Instead, we should use the `UNION` operator that can be found here:

[filter operator union 1 en] | *en/filter-operator-union-1-en.png*

Figure 27. The UNION operator in filters

Drag it to the zone, and then drag inside it twice the `Attributes → Equal → Name` filter:

[filter operator union 2 en] | *en/filter-operator-union-2-en.png*

Figure 28. The UNION operator and 2 subfilters

In the settings of the first Name filter, put the regular expression:

```
L.*
```

In the second Name filter, put:

```
J.*
```

(make sure the "regex" box is checked in both cases)

As a result, the nodes selected by both filters are added up in the display:

[filter operator union 3 en] | *en/filter-operator-union-3-en.png*

Figure 29. The UNION operator and 2 subfilters

3. The NOT operator

The NOT operator flips the result of a filter: what was hidden becomes visible and vice and versa.

Example: if we want to display all characters except for those returned by a UNION on 2 Name filters on L and J initials:

[filter operator not 3 en] | *en/filter-operator-not-3-en.png*

Figure 30. The NOT nodes operator - 1

Same effect, but applying the NOT operator on single filter using a regex on L or J:

[filter operator not 1 en] | *en/filter-operator-not-1-en.png*

Figure 31. The NOT nodes operator - 2

Same effect again, achieved without using the NOT operator. In regular expressions the ^ sign inside square brackets means "NOT":

```
[^LJ].*
```

[filter operator not 2 en] | *en/filter-operator-not-2-en.png*

Figure 32. Achieving a NOT effect with regex

Tutorials about regular expressions:

- https://regexone.com/
- http://www.themacroscope.org/?page_id=643

And a web page where you can test your regular expressions: http://regexpal.com

more tutorials on using filters in Gephi

- Video on using filters by Jen Golbeck

the end

Visit the Gephi group on Facebook to get help,

or visit the website for more tutorials

[1] D. E. Knuth, The Stanford GraphBase: A Platform for Combinatorial Computing, Addison-Wesley, Reading, MA (1993)