Université catholique de Louvain

Ecole polytechnique de Louvain

PROJECT REPORT

LDATA2010 - Information Visualisation

Glioblastoma Gene Interaction Visualization Dashboard

DIMITRI SCHOROCHOFF - **60241800**
THOMAS JACQUET - **08771800**

Academic year

2021-2022

Contents

1 Introduction

Today, data has a central role in our society. It is collected from many different sources and can be gathered under the form of dataset files. In order to process them efficiently and to extract the necessary informations, it is essential to have a good visualisation of these data. One possibility is to represent them under a network shape. It is not always an easy task, especially when the size of the graph becomes important.

This report aims to detail our approach of a good network representation. Our visualisation software will be explained as well as our different implementation choices.

2 Data analysis

The provided dataset was composed of four different files. We designated the file composed of the genes data as the nodes file of the network. Following the same principle, the file containing interactions information is the edges file. Two additional csv were given. They both give more informations about the different genes. They concern chemicals and PTM.

An important fact to notice is that all the rows of the csv files did not contain each possible characteristics. Some were just associated to a '-' symbol. We decided not to try to complete these missing cells because of the variety of the data. However, even if a line representing a node or an edge is completely empty, the node or edge is taken into account into the network. Indeed, as there are many missing values, it was simply impossible to choose to delete all of them.

3 GraphX

The goal of our software called *GraphX* is to visualize a network in a clear and detailed way. In order to launch the software, it is necessary to execute the python class named *GraphX_start.py*. As the software is based on some dependencies such as PyQt5, pyvis and networkx, some installation of librairies might be necessary.

3.a File selection

Once the application is launched, we arrive on a first screen which ask to select the four files of the dataset (figure 1). These files can either be .txt or .csv. It is important to select a file using each of the button. The type of file required (i.e. genes, PTM, ...) is precised. No miss-matches is allowed for the proper execution of the program. Once it is done, we can press the *Build graph* button.

Even if only two files were required to represent the graph (i.e. the one concerning the nodes and the other about the edges), we decided to include the whole dataset for the sake of completness. These extra informations will be used and displayed in the attribute tab and the node tab, as explained after.

3.b Overall functionning

The MainWindow has now appeared. It is composed in the upper part of a white space where the graph is being shown. As the graph has a quite important size, it can take several

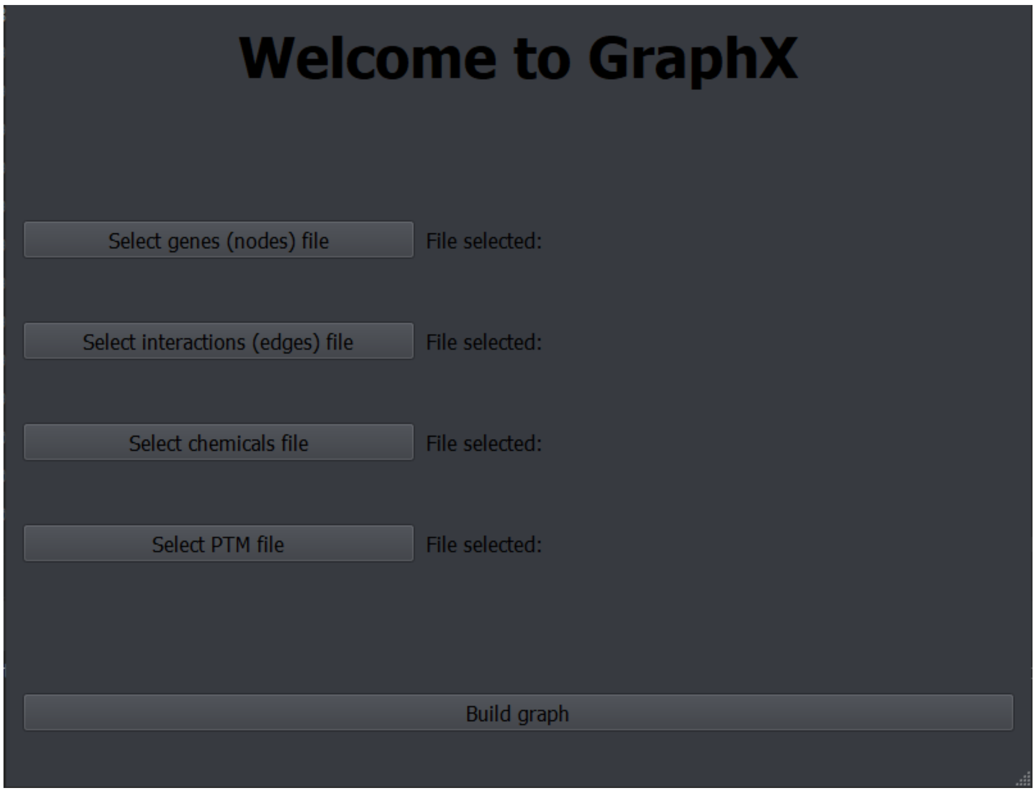


Figure 1: Window to load the files

minutes before being displayed, so be patient. Once we see the network, we can start to test some interesting features. Indeed, we can zoom on a node using the mouse wheel. We can drag the entire graph using the left click and even, if we have zoomed enough on it, move a node by doing a long left click on it.

Let now focus on the lower part of the window. It is composed of four tabs named *Layout*, *Attribute*, *Node* and *Edge*. Each one of these will be detailled below. They all have in common a search bar and a filter button which allows to find easily the element we are looking for. They also all have a big *Build* button on the right. Every time we are doing an operation on the graph (i.e. running an algorithm, changing the color of a node, choosing a layout, ...), we will need to push on this button to rebuild the result graph. If you forget to use this button, the graph will simply stay in its actual form, the one it had before the operation.

Note that you can adjust the ratio between upper and lower part by dragging the 5 five dot on the lower part. Offering a full view on the graph by dragging it all the way to the bottom.

3.c Layout tab

The layout tab contains four different layouts used to show the network with different manner. Each layout contains some editable arguments to allow to see their effects and adapt them to the different encountered situations. Because the dataset is big, the different layouts may take several minutes before being displayed.

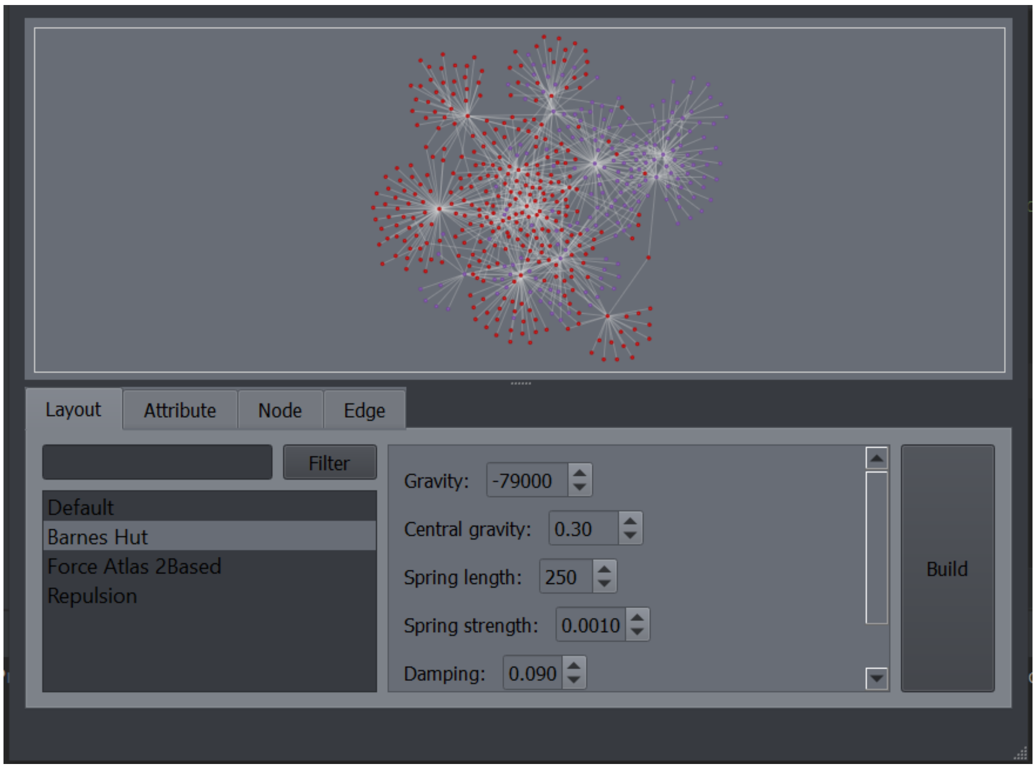
The default layout is a circular layout. We chose this kind of layout because we can clearly see the overall dataset and we already know which nodes have high degrees because they will be in the center of the circle. We thus already have a good global overview of our datas.

The second layout is named *Barnes Hut*. Of the three last layouts, it is the fastest to build.

It allows to see more precisely which nodes are of low degree because they will be in the outer space of the main cluster.

The third layout is the *Force Atlas 2Based*. It has the same properties than the *Barnes Hut* layout excepted the fact that the central gravity model is here independent from the distance and the repulsion force will be linear and not quadratic as before. Knowing that, it is thus interesting the test different argument values to see the differences.

The last layout is called *Repulsion*. It is an interesting one because we can spread even more the different nodes from each others. We see more the effect of this spreading on smaller graphs but as the dataset may vary in size, we made it available.



Gravity: -79000
Central gravity: 0.30
Spring length: 250
Spring strength: 0.0010
Damping: 0.090

Figure 2: Layout tab

3.d Attribute tab

The attribute tab is composed of many elements. Clicking on an attribute will first run the related algorithm on the current graph without considering any filter then let you modify the graph using the attribute's properties. There are two types of attribute. We will detail the different functionalities for both of them.

For numerical attributes (ex: the two first cells of the scrollable list which are *Degree* and *Clustering coefficient*) there is a checkbox allowing to adapt the size of the nodes with respect to the concerned metrics.There is also a double slider named filter with which we can display only the nodes having the metrics in the defined intervals. This is very comfortable to show a smaller number of nodes with defined metrics when working with large datasets. We can study their characteristics more easily. Under this double slider, there are three buttons. The left one is to apply colors to nodes according to the computed metric. The one in the center is to delete all the nodes which were not in the numerical interval defined by the double slider. The button on the righ is to reset the original graph when we have finished to work with the filtered subsample of nodes and edges.

For categorical attributes, the principle is relatively the same. As an example, the third cell *Communities* is a categorical attribute. Each category (i.e. communities for the example) is listed in a scrollable list on the right. We can associate a color to each category. We can also uncheck a specific category if we don't want to view it. The three same buttons as the numerical attributes are at the bottom. As before, there is one to apply the colors, one to delete the not wanted categories and one to reset the original network.

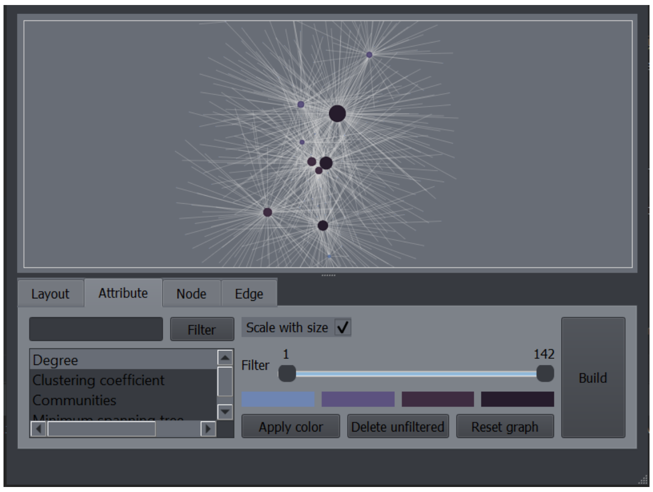


Figure 3: Numerical attribute

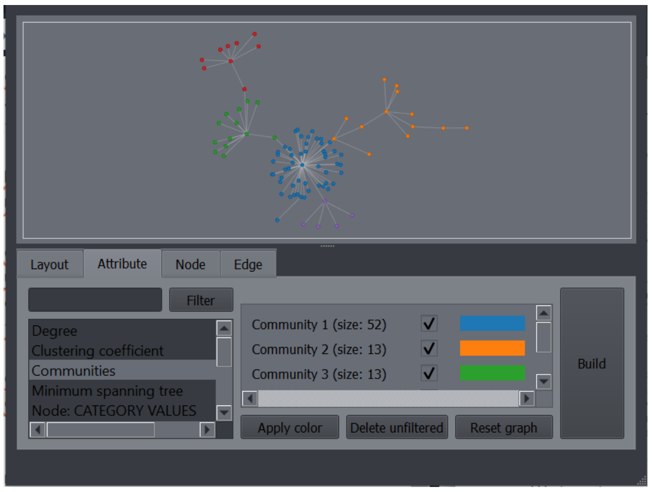


Figure 4: Categorical attribute

It is important to notice that the different filtering methods and criterias can be combined together. For example, it is possible to display only the nodes which are part of community number 2 and which degree $\in [3, 25]$. Or if you want to compute attributes on a smaller graph (e.g the minimum spanning tree of the graph only composed of the community number 7), don't forget to delete unfiltered element beforehand by clicking the "Delete unfiltered" button (in the example, uncheck all other community, then click on "Delete unfiltered" then uncheck the Out edge in the minimum spanning tree attribute). And as for all the features, don't forget to push on the *Build* button. The goal of the attribute tab is really to offer the possibility to the user to work with a smaller instance of the original dataset, determined by defined characteristics. We started from the principle that in order to completely appropriate ourselves a large dataset, it could be interesting to cut it into small pieces. It is indeed easier to analyse fewer informations and then step by step going back to the full graph.

We choose to replace unknow attributes by -1 for numerical ones and None for categoricals ones.

3.e Node tab

The node tab contains three elements. The first one is a scrollable list containing all the different nodes referenced by their biogrid id as well as a special cell mentioning *### All nodes ###*. Then, on the right, there is the possibility to pick a color and a size and changing the graph accordingly. Note that change made to the All nodes cell will be applied to all nodes. After clicking on a specific node, a user can, give a target node value and click on the *Apply* button to color the shortest path from the node which was clicked to the target using the specified color of the node.

If you now double click on an element of the scrollable list, an informative window will appear. It is composed of four tabs containing different data about the node concerning general, chemicals, PTM and attributes information. The three first tabs are characteristics directly

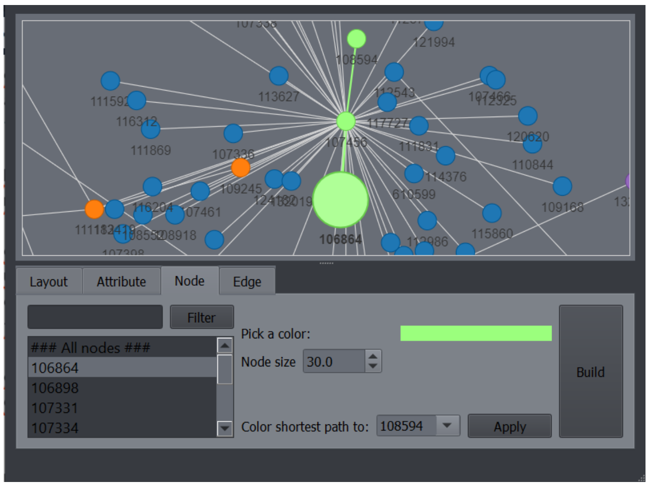
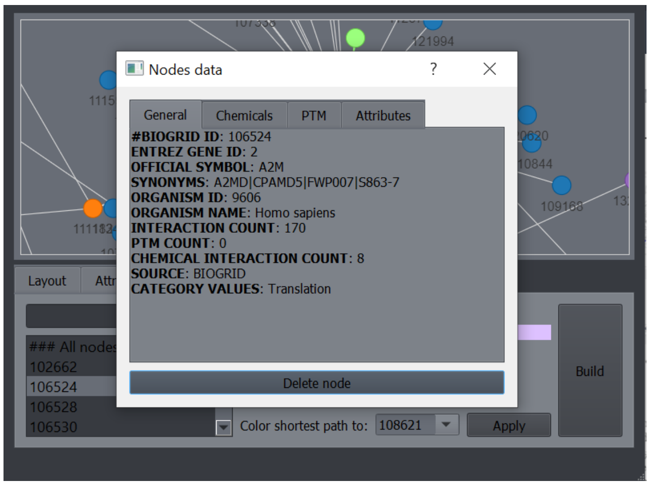


Figure 5: Node tab


Nodes data

General | Chemicals | PTM | Attributes

#BIOGRID ID: 106524
ENTREZ GENE ID: 2
OFFICIAL SYMBOL: A2M
SYNONYMS: A2MD|CPAMD5|FWP007|S863-7
ORGANISM ID: 9606
ORGANISM NAME: Homo sapiens
INTERACTION COUNT: 170
PTM COUNT: 0
CHEMICAL INTERACTION COUNT: 8
SOURCE: BIOGRID
CATEGORY VALUES: Translation

Delete node

Figure 6: Node informative window

extracted from the dataset files. We did not displayed missing information for more clarity. So if a specific label is not on the panel, this means that the corresponding dataset cell was empty for this specific node. The last tab contains information about the different algorithms which were run in the MainWindow attribute tab. Indeed, the node degree is present by default and if for example the algorithm *Communities* has been previously executed, the belonging community of the node will also be precised.

At the bottom of the informative window, there is a button *Delete node* which allows to delete a node from the graph and thus from the scrollable list.

3.f Edge tab

The same principle than above applies for the edge tab. It is possible to give a size and a color for edges represented by a pair of nodes. If we want to apply the color and size change to all edges, we click on *### All edges ###*. If we double click on one element of the scrolling list, we get an informative window with the features of the edge. As for the node, we also have the possibility to delete an edge from the network and from the list by clicking on the *Delete edge* button.

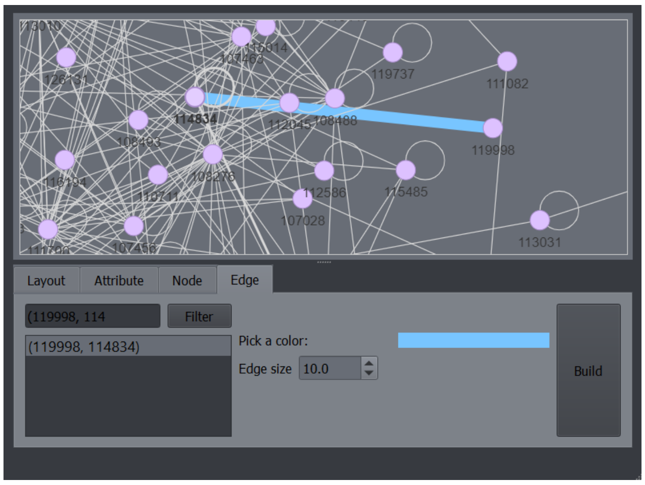
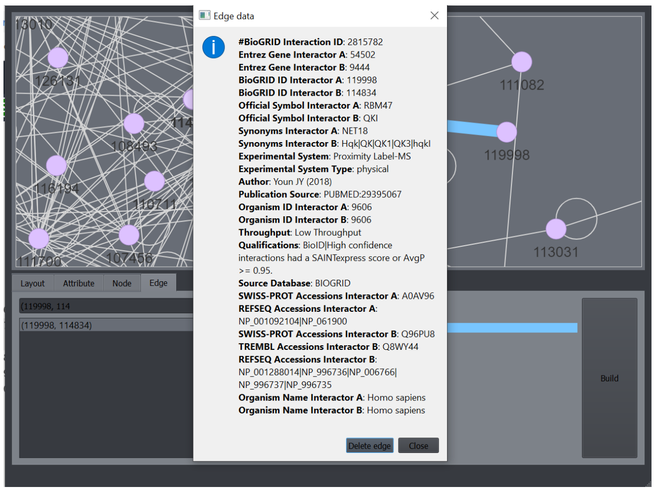


Figure 7: Edge tab


Edge data

#BioGRID Interaction ID: 2815782
Entrez Gene Interactor A: 54502
Entrez Gene Interactor B: 9444
BioGRID ID Interactor A: 119998
BioGRID ID Interactor B: 114834
Official Symbol Interactor A: RBM47
Official Symbol Interactor B: QKI
Synonyms Interactor A: NET18
Synonyms Interactor B: Hqk|QK|QK1|QK3|hqkI
Experimental System: Proximity Label-MS
Experimental System Type: physical
Author: Youn JY (2018)
Publication Source: PUBMED:29395067
Organism ID Interactor A: 9606
Organism ID Interactor B: 9606
Throughput: Low Throughput
Qualifications: BioID|High confidence interactions had a SAINTexpress score or AvgP >= 0.95.
Source Database: BIOGRID
SWISS-PROT Accessions Interactor A: A0AV96
REFSEQ Accessions Interactor A: NP_001092104|NP_061900
SWISS-PROT Accessions Interactor B: Q96PU8
TREMBL Accessions Interactor B: Q8WY44
REFSEQ Accessions Interactor B: NP_001288014|NP_996736|NP_006766|NP_996737|NP_996735
Organism Name Interactor A: Homo sapiens
Organism Name Interactor B: Homo sapiens

Delete edge Close

Figure 8: Edge informative window

4 Possible improvements

The main default of our software is its slowness to display the graph. Despite having tried many different librairies, we didn't find one which was quick enough. If we had more time, we could maybe continue the research for a faster library.

Another thing we could implement to improve the quality of the software would be to add more dialog boxes to notify a user that an algorithm is running or that the graph is being displayed.

It may be a good idea to generalize our software. It is actually designed to process glioblastoma dataset files but it would be possible to treat other type of dataset as long as they contain a node and an edge file.

We could also modify the front end of the software by designing better looking interfaces. Indeed, some label and some buttons don't align well with each other from time to time.

It is also possible to add new features such as new layouts, new algorithms, new metrics,...

5 Conclusion

To finish, we can say that our software is working well. Many features were implemented in order to show the maximum amount of information in a clear and precise way. We can easily combine a wide variety of filter and still have a lot of room the see the resulting graph. As mentionned before, some improvements could be made such as decreasing the reaction time of the graph displayer. However, the most possible improvements would be on the software side (i.e. reduce latency, improve appearance of some layouts) rather than on the visualisation side.