UCLouvain

École polytechnique de Louvain

Information Visualization

LDATA2010 Project

Report

Glioblastoma Gene Interaction Visualization Dashboard

Teacher

Professor John A. Lee

Authors

Atefeh Bahrami
NOMA:05742100
Atefeh.bahrami@student.uclouvain.be

Nima Farnoodian
NOMA: 68372000
Nima.farnoodian@student.uclouvain.be

1. Introduction

In the modern era when data exist everywhere, Information visualization is considered as an important analytical tool to display and process data for different aspects and it is quickly being recognized as an essential part of effective research communication. In this project, we have been asked to provide an Information visualization dashboard to visualize a glioblastoma gene interaction dataset that includes both edges and nodes dataset separately. Furthermore, this interactional dashboard should have designed so that it can satisfy user's needs such as uploading new datasets with a specific format, modifying the color and size of the nodes and edges according to some metrics, depicting multiple views of the graph structure, and so on. In order to achieve this goal, we developed a user-friendly dashboard (called BioVisualizer) with Python 3 and Dash framework for building GUI (Dash Framwork, n.d.). Dash is a powerful framework built specially for creating interactive data visualization apps. In order to create a Network visualization app using Dash, we utilized Dash Cytoscape [(Dash Cytoscape, n.d.)] that is a network visualization component for Dash. Moreover, as we required to apply some network algorithms such as Community Detection, Shortest-path, we used python NetworkX library (NetworkX, n.d.). In the next sections, we will explain how BioVisualizer works and what features it includes.

2. Tutorial

This section gives you an overview of the features BioVisualizer supports and explain how the users can work with them.

2.1. features

- **File Upload**

  With this feature, user can add/upload a new dataset on BioVisualizer to visualize the corresponding network and interact with it. The dashboard supports .csv and .txt file formats provided that the node and edge datasets follow the same structures as the given files in (Themed Curation Project, n.d.). It is worth mentioning that, the given datasets do not match as the edge dataset contains many nodes that do not exist in the node datasets. Indeed, the edge dataset represent a very large multiple-network with 9334/36500 nodes/edges while the node datasets only present 190 identical nodes. Clearly, plotting a very large network is a daunting task and needs so much effort just to fit the network in the plot. After a short discussion with our Teaching Assistant, Mr. Lambert Pierre on November 7, we have decided to focus on the node dataset and create the network based on the nodes that are observed in the node dataset. With this, the resulting multiple-network consists of 190/1840 nodes/edges. Therefore, once a user uploads her/his node and edge datasets, and press **"Build Graph"** button, a multiple-network and a simple network are created with an emphasizing on the node dataset. We should note that multiple-network is used for computations like metrics and simple network is used for the sake of visualizing the network. So, what the user observes is a simple version of the original network. However, all the information shown regarding the network is based on the multiplicity of the network; thus, **the user can have an idea of the real network but with a cleaner visualization.**

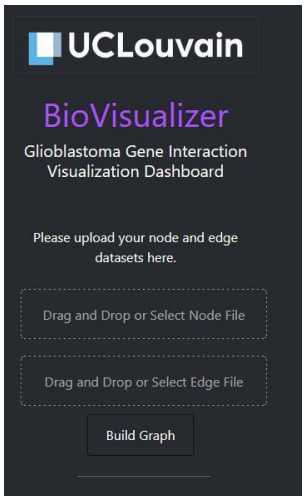


Figure 1 Upload and Build Section

- **Visualizing Network**

  After building the network, the user can draw the network by pressing **"Plot all Connected Component"** button above the visualization frame. After pressing, the caption of the button changes to **"Plot Only Giant Component"**. This allows the user to focus on only the giant component if pressed. It is a useful feature, as we observed that the network includes many isolated nodes. So, in this case, the user can get rid of these nodes and clean the depicted network. Notice that, the user can retrieve the original network by pressing the same button —the button guides the user to change the network version.

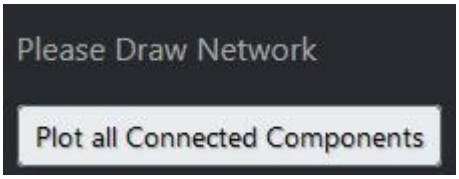


Figure 2 Plot Button for all Components

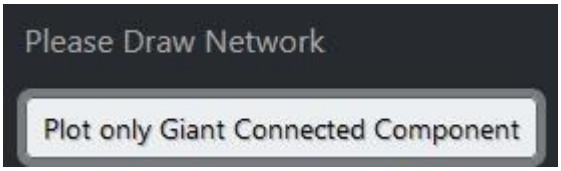


Figure 3 Plot Button for the Gian Component

Below the visualization frame, there is a CheckBox named "Node Label" that removes the labels of the nodes from the visualization if unchecked, otherwise the labels are included in the visualization.

- **Compute Metrics of network**

  One of the primary features for all visualization dashboards is Metrics computation. Therefore, we added this feature to **BioVisualizer** to let user study the network not only statistically but also visually. In the **"compute metrics"** menu, a user is able to compute the following metrics:

  - ✓ Degree Centrality
  - ✓ Betweenness Centrality
  - ✓ Closeness Centrality
  - ✓ Eigenvector Centrality
  - ✓ Clustering Coefficient for nodes

  It is worth noting that, after pressing any of the above metrics (e.g, Betweenness), all the computed information is stored in the memory. This allows the user to prevent from re-computing the same metrics again, which might be heavy. Moreover, once a metric is computed, the metric is appended in the **"Metrics Visualization Mode"** dropdown, which

lets the user to resize the nodes based on the computed metrics. For example, once the user presses **"Betweenness Centrality",** it will be automatically added to the **Metrics Visualization Mode**, and by choosing **Betweenness Centrality** in this dropdown, all the nodes are resized according to their corresponding value.

Figure 4 . An example of resizing the nodes based on betweenness centrality

- **Community Detection**

The Community set is of an important hidden structures that should be detected so as to reveal the node interaction preference. For this reason, in the **"Community Detection"**, the user can detect the communities in the network by pressing **"Detect Communities"** button. We used Louvain Community Detection Algorithm (Python Louvain, n.d.) as our chosen method due to its computational power. Below **"Detect Communities"** button, there is a Power Button named "**Show"** that assign a color to each community if it is on — only top 23 communities are differently colored and the rest is similarly colored. Indeed, the user can decide to color the nodes based on the communities they belong to or assign a unique color to all nodes by turning "**Show"** off. We will discuss coloring features in the coloring sections.

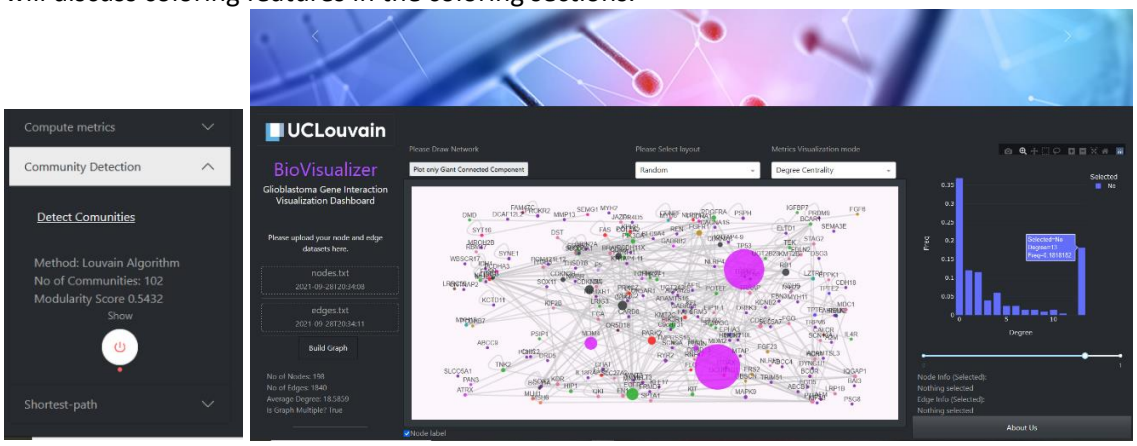


Figure 5 An example of community detection. Each node is colored based on its community

- **Layout**

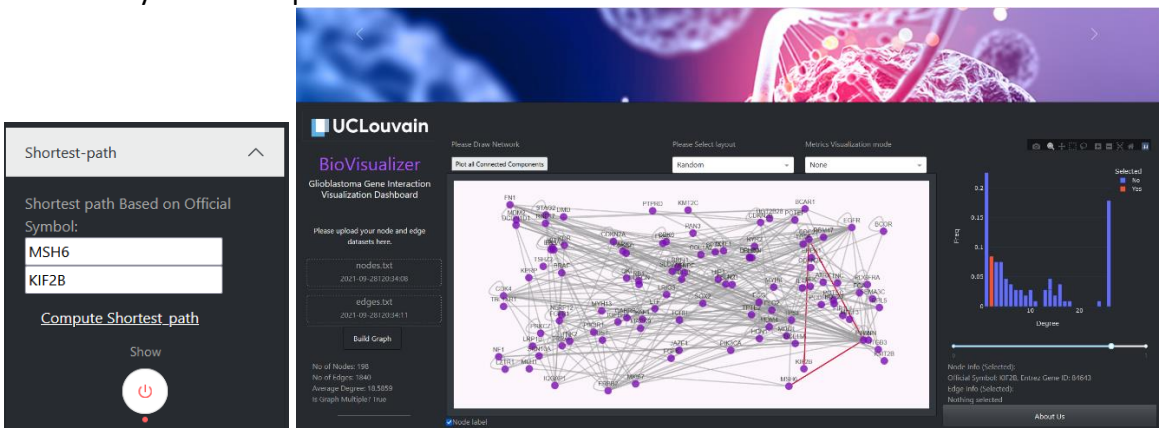
With this feature, the user is able to display multiple views of a network using different layouts such as:

- ✓ Random (default)
- ✓ Grid
- ✓ Circle
- ✓ Concentric
- ✓ Breadth first
- ✓ Cose
- ✓ Cose_bilkent
- ✓ Degre
- ✓ Cola
- ✓ Klay
- ✓ Spread
- ✓ Euler

In the entire of our project, the purpose was to provide an interface for the user with the most usage, accessibility, functionality and flexibility. So, in this section, we provided a list of various layouts to allow user to change the layouts as s/he wishes. The user can simply change the layout by selecting them in the **Layout** Dropdown above the visualization frame. Notice that the default layout is Random as it is very fast, and once the user switches to either **Giant Component** or **All Connected Components**, the default layout is activated.

- **Shortest path**

  BioViusalizer supports shortest path computation very well. A user can find the shortest path between two nodes based on the official names of the nodes. The official names are provided on the right hand-side below the degree distribution bar. Once the user double-clicks a node or edge in the visualization frame, the details will be shown there. After shortest-path computation, the shortest-path will not be immediately depicted. To visualize the shortest-path, the user requires to turn on **Show** power button like community detection power button.



- **Change color and size of Nodes and Edges**

  We purely believe in the freedom in the colors as a good visualization is only achieved by a good color setting. To this end, for nodes, labels, edges, and even background, BioVisualizer provides the user with the color selection. The user can set the colors for all visualization elements only by pressing “Apply Coloring” after setting carefully the colors.

In addition to the color selection, there are two knobs for setting the sizes of nodes and edges separately. These knobs are handful when resizing the nodes based on a metric. The user can make the range smaller or larger. Moreover, two sliders for edge and node opacity are provided as well.

- **Degree Distribution Bar Chart**
  Once a user builds and plots the network, a bar chart automatically will be drawn on the right. This shows the degree distribution of nodes. When a user double-clicks a node in the visualization frame, the corresponding degree of the node is shown as selected in the bar chart. Using this, the user can get notified of where the node stands in the network in terms of degree distribution. The slider below the bar chart changes the quartile. By this, we mean, in the bar char, only the degrees shown that are below the quartile. The rest are shown in the rightmost column. For example, if the quartile is .8, and the last column is selected after double-clicking a node, it means that the degree of the node is more than 80 percent of nodes in the network.

3. User Interface

For designing BioVisualizer dashboard, we tried to consider all aspects of a web product with a good UI. For instance:

- ✓ Access to all elements easily
- ✓ Hierarchy
- ✓ Minimizing memory load
- ✓ Preventing error as much as possible
- ✓ Consistency
- ✓ Allowing users to navigate easily
- ✓ Providing informative feedback
- ✓ Letting user to choose his/her favorite color for changing graph

4. User interactive aspects

In our designed dashboard, we considered almost all interactive aspects in order to make an interactional and **functional** software. BioVisualizaer also supports

- ✓ ability to drag some nodes to change their positions using the mouse
- ✓ ability to depict some node label and edge weight information when selecting the corresponding node or edge with the mouse
- ✓ ability to see the degree of selected node

5. Future works

Although we tried to create a fully functional data visualization app, we are utterly aware that it still needs work to materialize our initial goal. One way to improve it is to enable the app to fully support the network when the edge dataset is only considered. Another possible improvement is to add a feature enabling the user to modify the layout properties freely.

References

Dash Cytoscape. (n.d.). Retrieved from https://dash.plotly.com/cytoscape

Dash Framwork. (n.d.). Retrieved from https://dash.plotly.com/

NetworkX. (n.d.). Retrieved from https://networkx.org/

Python Louvain. (n.d.). Retrieved from https://python-louvain.readthedocs.io/en/latest/

Themed Curation Project. (n.d.). Retrieved from https://thebiogrid.org/project/5/glioblastoma.html