

Community Detection App for Cytoscape

Cytoscape is an open source software platform for visualizing complex networks and integrating these with any type of attribute data. The Community Detection App is a Java plugin for Cytoscape which provides the user interface to execute several community detection algorithms and gene-set enrichment tools. All these algorithms and tools run on a remote server which exposes REST API. The app uses these API to execute the algorithms and tools.

1 Installation

1.1 Cytoscape

- If you don't have Cytoscape installed on your system, please follow the instructions at <https://cytoscape.org/download.html>.
- More information on Cytoscape can be found at <https://cytoscape.org>.

1.2 Community Detection App

1. Download the JAR from https://cytoscape-builds.ucsd.edu/cytoscape-builds/community_detection/.
2. Click on "Apps" from the menu options at the top of the screen → click on "App Manager" → click on "Install from File..." under the "Install Apps" tab → select the JAR → click on "Open" → once the app is installed, close the dialog box.

2 User Options

The community detection app adds menu options in the menu bar and the context menu. A menu bar is the list of options on the top-left panel of Cytoscape window (Figure 1). A context menu gets populated when a user right-clicks on the network view panel (Figure 8). In general, the menu bar options of this app apply to the whole network and those in the context menu apply to the selected node(s). The app currently provides three major functionalities, explained below.

2.1 Community Detection algorithms

There are several algorithms which can be used for community detection of a network. However, most of these algorithms are written in languages other than Java and hence cannot be used via Cytoscape. The CD service exposes REST API to execute community detection algorithms and the application acts as a simplified interface between a user and CD service.

A sample workflow to execute an algorithm can be found in the images below. A user can select one of the algorithms from the menu (Figure 1) and an edge attribute from a drop-down list (Figure 2) which would be used as a weight for the network. The resultant hierarchy network can be seen in Figure 3.

An added feature for the community detection algorithms is the option to tune resolution parameters. Each of these algorithms has a parameter that determines the size of resulting clusters.

The "Settings..." option (Figure 4) opens a dialog box to set the appropriate resolution parameter. Since each algorithm implements the resolution parameter differently, a brief description for each algorithm's resolution parameter will appear when clicked on the parameter name or the "i" icon (Figure 5).

2.2 Gene-set Enrichment

One of the main reasons to apply a community detection algorithm on a protein-protein interaction network is to find out previously unknown interactions in known complexes. In order to find known complexes associated with the communities in a hierarchy network, the app currently supports two enrichment tools called GProfiler and Enrichr. There are two ways of running these tools:

- Menu bar - The menu bar option runs for the whole hierarchy network. Figures 6 and 7 show the menu bar option and GO enrichment result. Depending upon the size of the hierarchy network, GO enrichment could take several minutes or hours to complete.
- Context Menu - Another way to call an enrichment tool is through the context menu. However, this option works for selected nodes. Figure 8 shows the context menu option.

2.3 Interaction Sub-Networks

Another useful feature of this application is to view interactions among the genes of a community. This is achieved by creating a sub-network of the original interaction network. The sub-network contains genes of the selected community node and the edges connecting those genes. Figures 9 and 10 provide a snapshot of the context menu option and the resulting sub-network respectively. The context menu option, however, works only if 1 node is selected. An important point to note about the sub-network creation task is that it depends on network table (CD_OriginalNetwork) and node table (CD_MemberList) entries. To create a correct sub-network both these entries should exist and entry under CD_OriginalNetwork should point to a valid network.

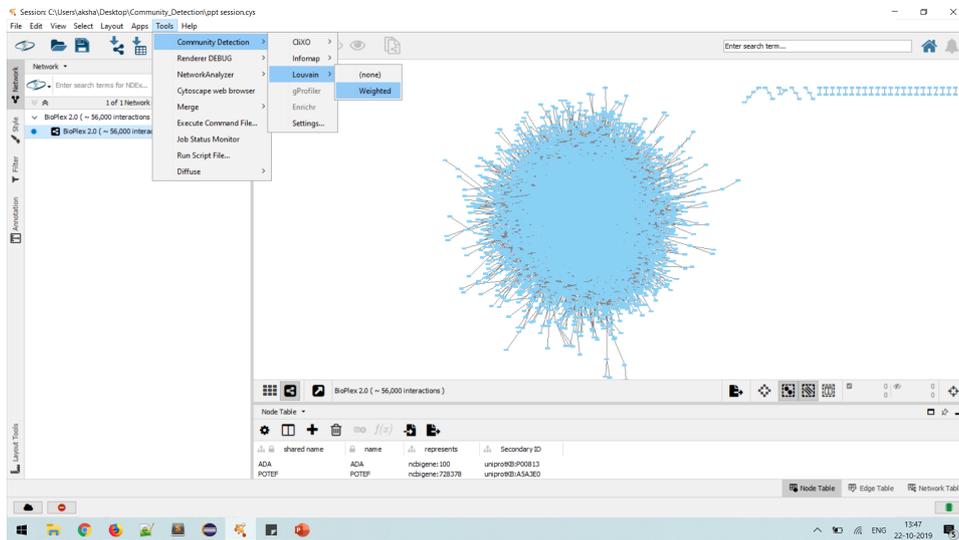


Figure 1: An example to choose a community detection algorithm.

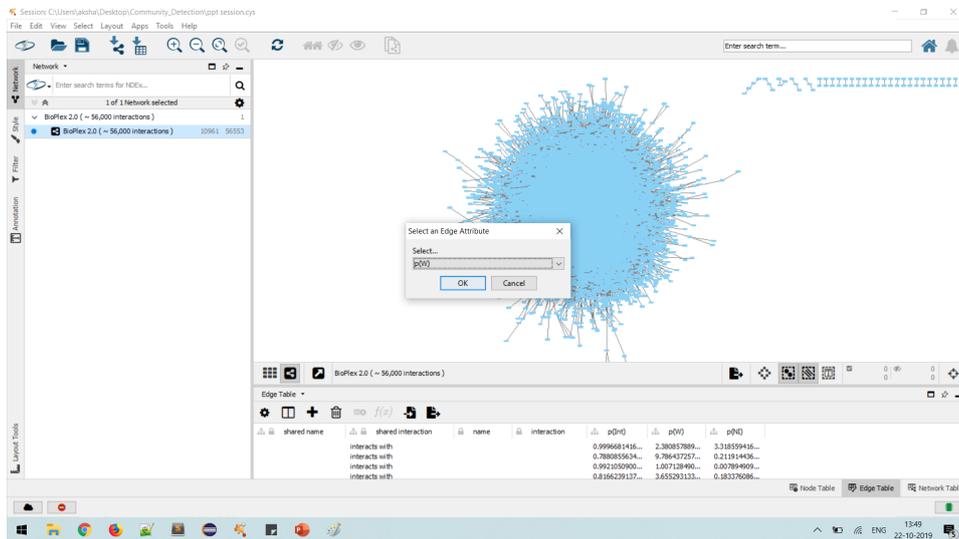


Figure 2: Selecting an edge attribute as a weight.

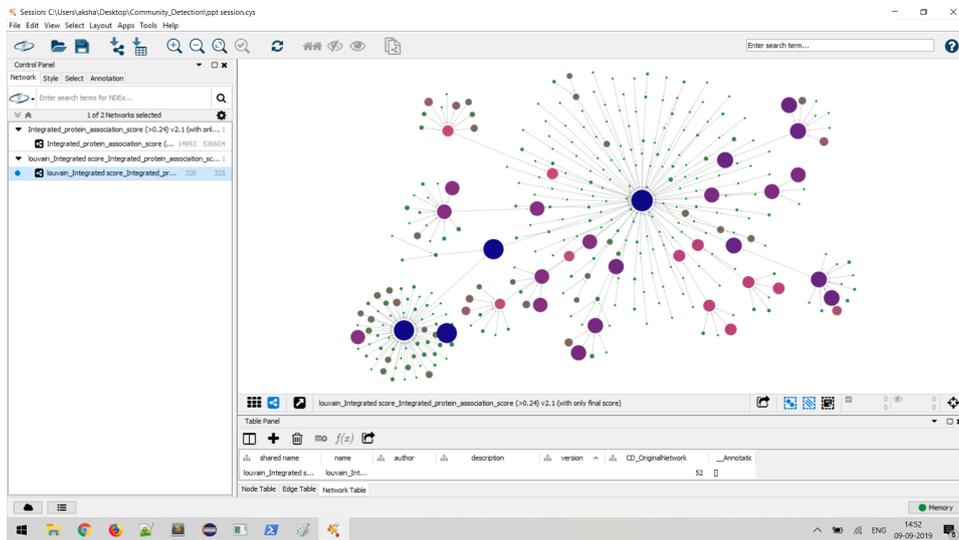


Figure 3: Resultant hierarchy network.

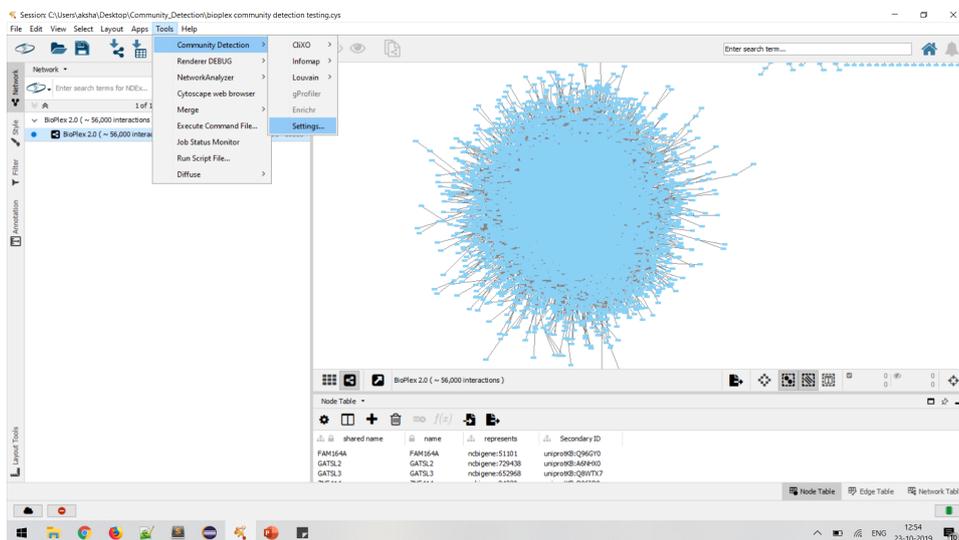


Figure 4: Menu option to select "Settings..."

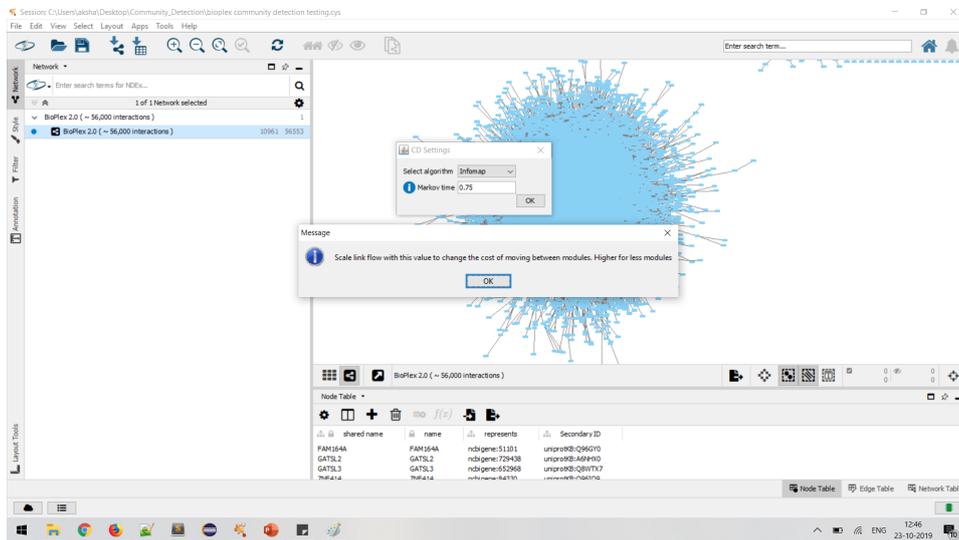


Figure 5: Dialog box with a brief description of the resolution parameter

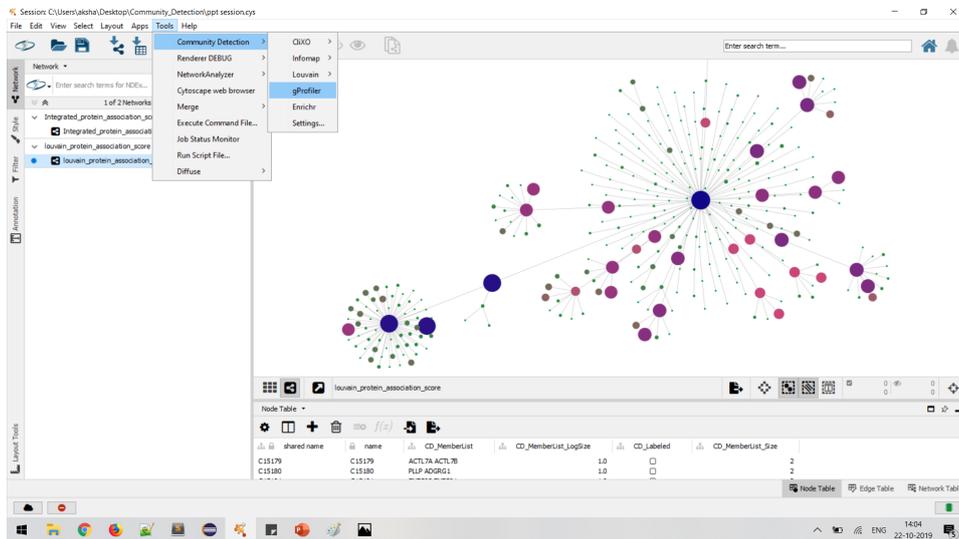


Figure 6: The menu bar option to run GProfiler.

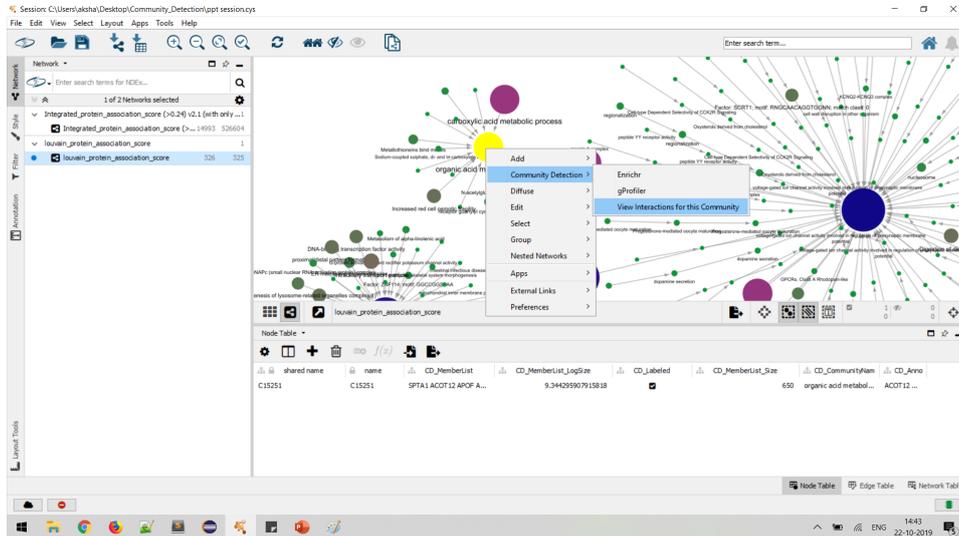


Figure 9: Context menu option to view interactions of the selected community node's genes.

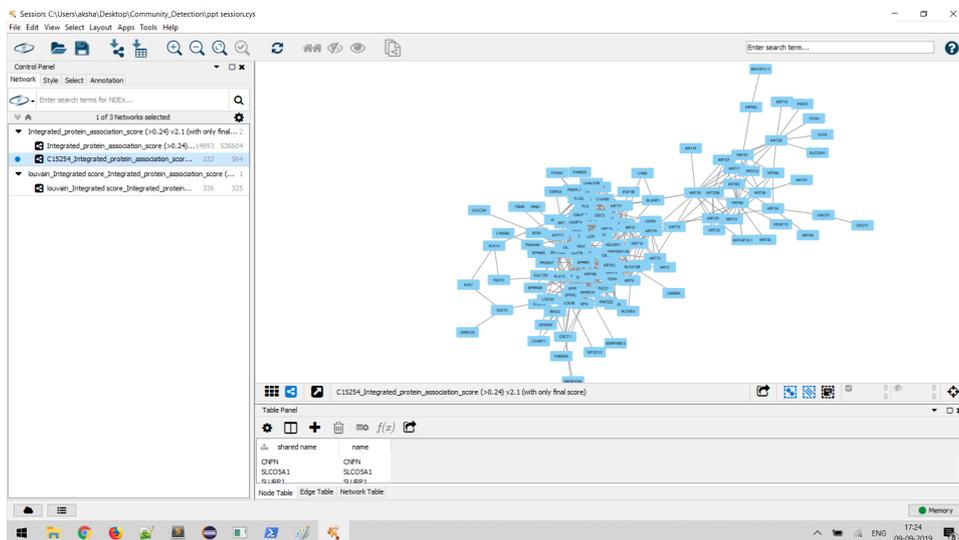


Figure 10: Resultant interaction network as a sub-network of the original interaction network.