# Cytoscape_3/UserManual

# Table of Contents

# Cytoscape 3.0.1 User Manual

# Introduction

Cytoscape 3.0.1 is the latest version of Cytoscape, which introduces a whole new architecture, developer API and set of user controls when compared with the 2.X series. Cytoscape 3.0.1 represents a major redesign from the ground up, toward the goal of providing a stable, extensible, high-performance platform for network visualization and analysis. That being said; if you're familiar with former versions of Cytoscape, this version will feel completely familiar and you'll be set to go directly

*This manual describes the installation and use of Cytoscape. For a more thorough understanding of Cytoscape and its ecosystem, we highly recommend reading the* **Welcome Letter** *accessible on the http://cytoscape.org website.*

This release introduces users to new interfaces and features:

• New welcome screen and toolbar for quick access to common actions

• New Data Table controls to interface with your data

• Integrated support for graphical annotations in networks

• Support for independent sets of networks within a single session

• Advanced network and table import controls

• Improved architecture and API for a more stable experience

In future releases, we will continue to tweak and improve both the software and the documentation. This manual will be updated to reflect all the latest changes.

# Launching Cytoscape

Cytoscape is a Java application verified to run on the Linux, Windows, and Mac OS X platforms. Although not officially supported, other UNIX platforms such as Solaris or FreeBSD may run Cytoscape if Java version 5 or later is available for the platform.

## System requirements

The system requirements for Cytoscape depend on the size of the networks the user wants to load, view and manipulate.

|  | **Small Network Visualization** | **Large Network Analysis/Visualization** |
|---|---|---|
| **Processor** | 1GHz | As fast as possible |
| **Memory** | 512MB | 2GB+ |
| **Graphics Card** | On board Video | Highend Graphics Card |
| **Monitor** | XGA (1024X768) | Wide or Dual Monitor |

*Specific system requirements, limitations, and configuration options apply to each platform, as described in the* **Release Notes** *available on the http://cytoscape.org website.*

# Getting Started

## Install Java

If not already installed on your computer, download and install Java SE 6. Cytoscape functionality has not been verified yet with SE 7. Link to download SE 6: Java SE 6

## Install Cytoscape

### A quick note on upgrading

*If you are installing Cytoscape for the first time on your workstation, skip to the **Download** section below.*

There should really be no issue in upgrading. If you have a previous installation you have two options:

1. Starting with a clean slate. For this you should delete your previous installation directory and the `CytoscapeConfiguration` directory (see below for the location of this directory).

2. Just keep what you have and simply pick a distinct, new directory for installation. In the unlikely event that you should encounter any problem, delete the `.props` files in your `CytoscapeConfiguration` directory. If that doesn't help try deleting the `CytoscapeConfiguration` directory. This latter step will cause you to lose all of the apps that you have installed via the App Store, so only do that if you are having problems or if you don't mind reinstalling your apps. The core apps will not be affected by this step.

### Download

There are a number of options for downloading and installing Cytoscape. See the download page at the http://cytoscape.org website for all options.

- Automatic installation packages exist for Windows, Mac OS X, and Linux platforms.

- You can install Cytoscape from a compressed archive distribution.

- You can build Cytoscape from the source code.

- You can check out the latest and greatest software from our Subversion repository.

Cytoscape installations (regardless of platform) containing the following files and directories:

| File/Directory | Description |
| --- | --- |
| `p/Cytoscape_v3.0.0` | Cytoscape program files, startup scripts, and default location for session files |
| `p/Cytoscape_v3.0.0/Cyto-scape.vmoptions` | Cytoscape memory configuration settings |
| `p/Cytoscape_v3.0.0/sampleData` | Preset networks as described in the embedded README.txt file |
| `p/Cytoscape_v3.0.0/framework` | Cytoscape program files |
| `u/CytoscapeConfiguration` | Cytoscape properties and program cache files |
| `u/CytoscapeConfiguration/cyto-scape3.props` | Cytoscape configuration settings |

The `p/` directory signifies the program directory, which varies from platform to platform.

The `u/` directory signifies the user's home directory, which varies from user to user and from platform to platform.

# Launch the application

As with any application: double-click on the icon created by the installer or by running `cytoscape.sh` from the command line (Linux or Mac OS X) or double-clicking `cytoscape.bat` (Windows).

Alternatively, for more advanced users, you can pass the .jar file to Java directly using the command `java -Xmx512M -jar cytoscape.jar -p plugins`. The `-Xmx512M` flag tells java to allocate more memory for Cytoscape and the `-p plugins` option tells cytoscape to load all of the apps (called *plugins* in previous Cytoscape versions) in the apps directory. Loading the apps is important because many key features like layouts, filters and the attribute browser are included with Cytoscape as apps in the `plugins` directory. See the Command Line chapter for more detail. In Windows, it is also possible to directly double-click the .jar file to launch it. However, this does not allow specification of command-line arguments (such as the location of the apps directory).

After launching Cytoscape a window will appear that looks like this (captured on Mac OS 10.7.4):
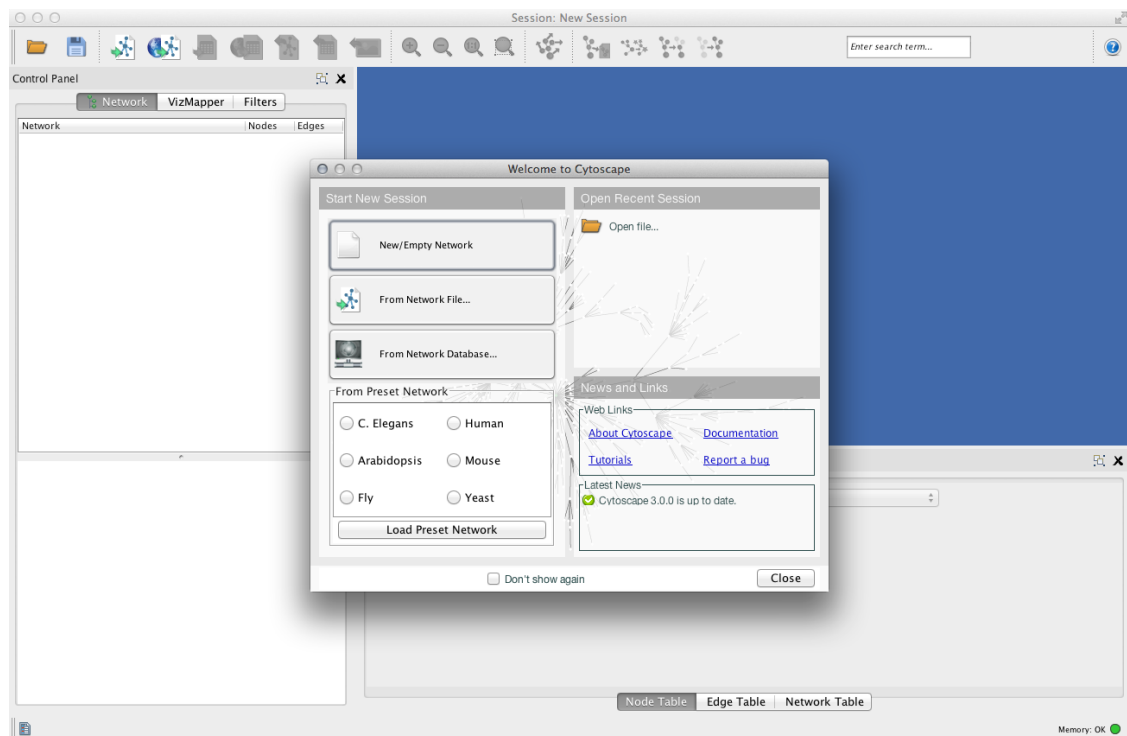


*If your Cytoscape window does not resemble this, further configuration may be required. Consult the **Release Notes** available on the http://cytoscape.org website.*

## Note on Memory Consumption

For most regular users Cytoscape will estimate the proper amount of memory, if so; you can skip this section and go for the Quick Tour directly.

For users interested in loading large networks, the amount of memory needed by Cytoscape has to be increased. Memory usage depends on both number of network objects (nodes+edges) and the number of attributes. Here are some *rough* suggestions for memory allocation:

*Suggested Memory Size **Without** View*

| Number of Objects (nodes + edges) | Suggested Memory Size |
|---|---|
| 0 - 70,000 | 512M (default) |
| 70,000 - 150,000 | 800M |

*Suggested Memory Size **With** View*

| Number of Objects (nodes + edges) | Suggested Memory Size |
|---|---|
| 0 - 20,000 | 512M (default) |
| 20,000 - 70,000 | 800M |
| 70,000 - 150,000 | 1G |

*Specific memory configuration recommendations for each platform may be described in the **Release Notes** available on the http://cytoscape.org website.*

## Overall Memory Size for Cytoscape

To increase the maximum memory size for Cytoscape, you can specify it in a file, residing in the same directory as the Cytoscape executable, called **Cytoscape.vmoptions** with one option per line and lines separated by linefeeds. The last line must also be followed by a linefeed. The one exception to this rule is the MacOS platform if you are launching Cytoscape by clicking on the Cytoscape icon. (In that case you will have to edit the **../Cytoscape.app/Contents/Info.plist** file instead!) For example, if you want to assign 1GB of memory, create the **Cytoscape.vmoptions** file containing the single line (Do not forget the linefeed at the end of the line!):

**-Xmx1GB**

## Stack Size

There is one more option related to memory allocation. Some of the functions in Cytoscape use larger stack space (a temporary memory for some operations, such as Layout). Since this value is set independently from the *Xmx* value above, sometimes Layout algorithms fails due to the *out of memory* error. To avoid this, you can set larger heap size by using the **-Xss** option. If layout fails for large networks, please try the following:

**-Xmx1GB -Xss10M**

The option **-Xss10M** means set the heap size to 10MB. In many cases, this solves *out of memory* error for Layouts.
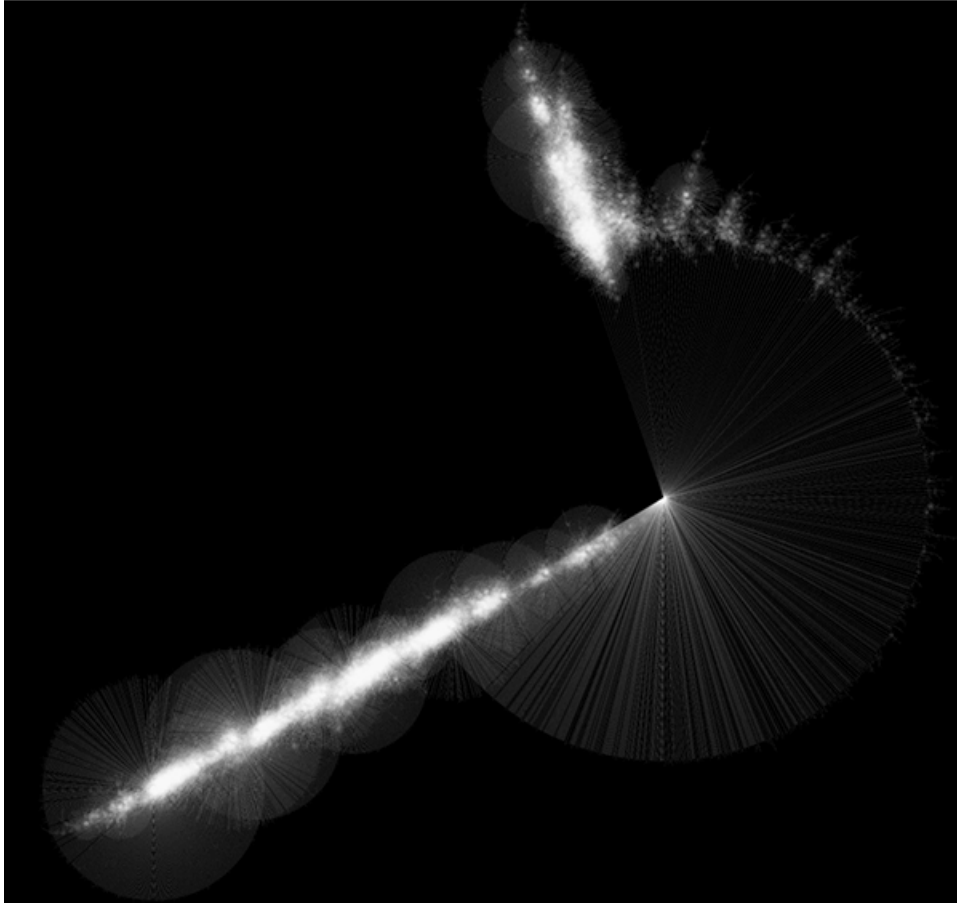
**Randomly generated scale-free network with 500K nodes and 500k edges:** If memory parameters are set properly, you can visualize huge networks. In this example, about 5GB of memory is used by Cytoscape. Stack size is set to 10MB. To use large memory space (4GB+) you need 64-bit version operating system AND 64-bit version Java SE 6.

Note: Some of the web service clients are multi-thread programs and each thread uses the memory size specified by *-Xss* option. If web service clients fails due to the *out of memory* error, please reduce the stack size and try again.

For more details, see How_to_increase_memory_for_Cytoscape.

## Note on Directory Location

For the application to work properly, all files should be left in the directory in which they were unpacked. The core Cytoscape application assumes this directory structure when looking for the various libraries needed to run the application. If you are adventurous, you can get creative with the $CLASSPATH and/or the cytoscape.jar manifest file and run Cytoscape from any location you want.

# Quick Tour of Cytoscape

When a network is loaded, Cytoscape will look something like the image below:
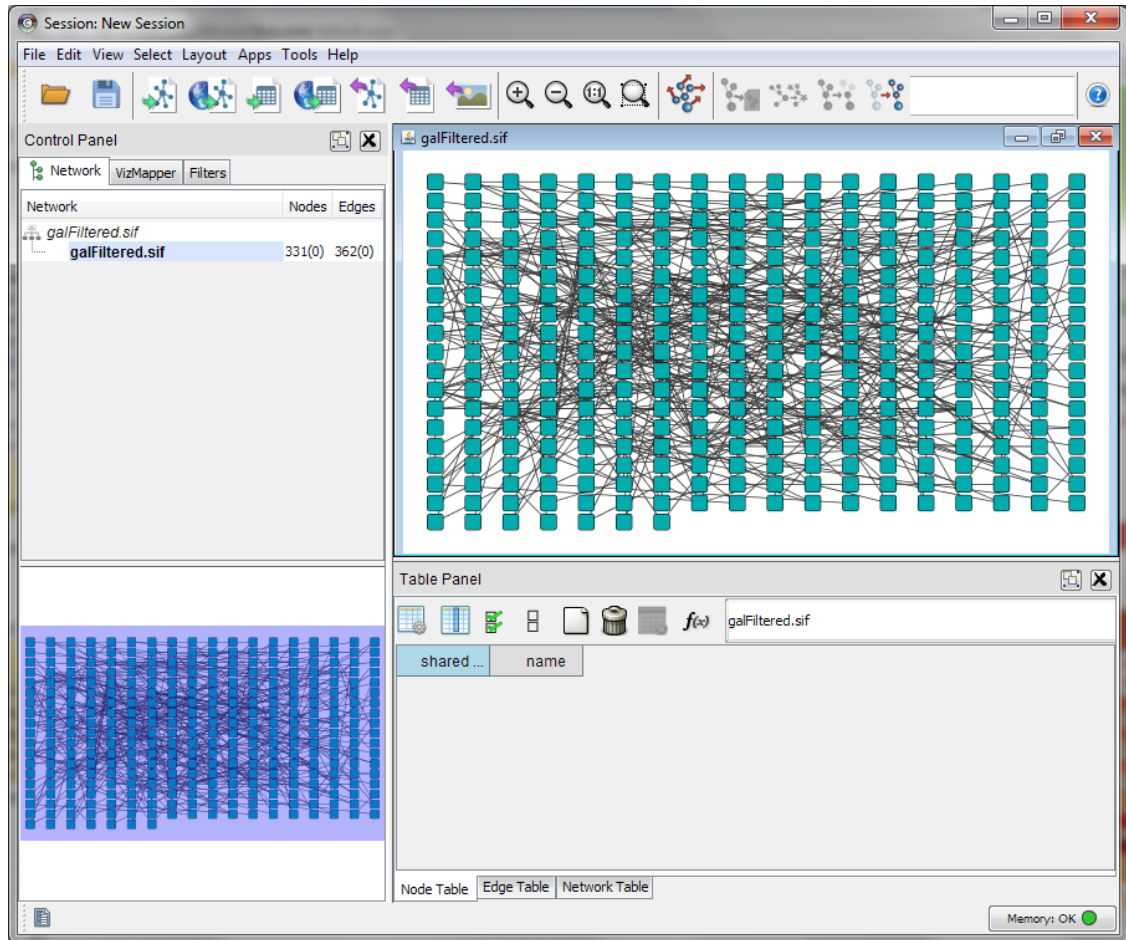
Most functionalities are self evident, but we'll go through a thorough explanation for clarity. The main window here has several components:

- The menu bar at the top (see below for more information about each menu).

- The toolbar, which contains icons for commonly used functions. These functions are also available via the menus. Hover the mouse pointer over an icon and wait momentarily for a description to appear as a tooltip.

- The network management panel (top left panel). This contains an optional network overview pane (shown at the bottom left).

- The main network view window, which displays the network.

- The attribute browser panel (bottom panel), which displays attributes of selected nodes and edges and enables you to modify the values of attributes.

The network management and attribute browser panels are dockable tabbed panels known as CytoPanels. You can undock any of these panels by clicking on the Float Window control
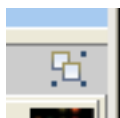


in the upper-right corner of the CytoPanel. This is useful when you want assign the network panel as much

screen   space   as   possible.   Clicking   the   Hide   Panel   icon:
☒
will hide the panel; this can be shown again by choosing View>Show ...panel

If you select this control, e.g. on the attribute browser panel, you will now have two Cytoscape windows, the main window, and a new window labeled !Table Panel, similar to the one shown below. A popup will be displayed when you put the mouse pointer on a cell.



Note that !Table Panel now has a Dock Window control. If you select this control, the window will dock onto the main window.

Cytoscape also has an edit functionality that enables you to build and modify networks interactively within the network canvas. To edit a network, just right click on the open space of network window, select the menu item "Add" -> "Node", a new node will be added to the canvas. To add an edge, right click on a node, choose the menu item "Edit" --> "Add Edge". Then select the target node, a new edge will be added between the two nodes. In a similar way annotation objects can be added; pictures, shapes or textboxes; much like in MS PowerPoint or similar software.
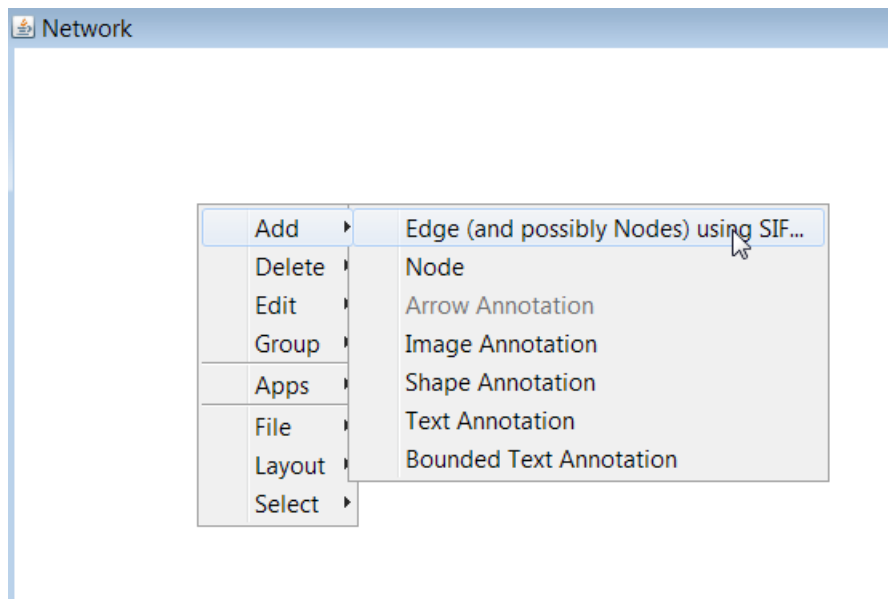
# The Menus

## File

The File menu contains most basic file functionality: File → Open for opening a Cytoscape session file; File → New for creating a new network, either blank for editing, or from an existing network; File → Save for saving a session file; File → Import for importing data such as networks and attributes; and File → Export for exporting data and images. Also, File → Print Current Network... allows printing, while File → Quit closes all windows of Cytoscape and exits the program.

```
File  Edit  View  Select  Layout  Apps  Tools

    Recent Session                    ▶
    New                               ▶
    Open                  Ctrl+O

    Save                  Ctrl+S
    Save As               Ctrl+Shift+S

    Import                            ▶
    Export                            ▶

    Print Current Network...  Ctrl+P

    Quit                  Ctrl+Q
```
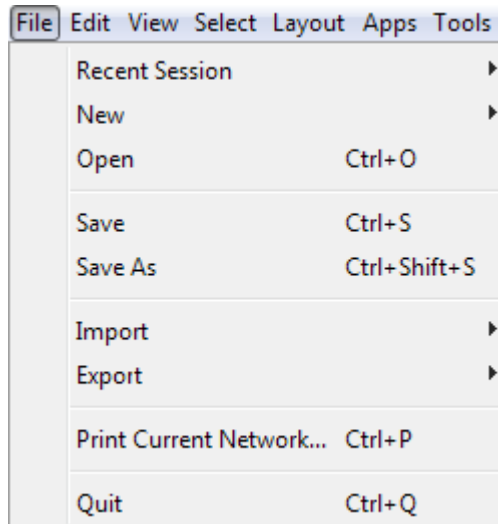
## Edit

The Edit menu contains Undo and Redo functions which undo and redo edits made in the Attribute Browser, the Network Editor and to layout.

There are also options for creating and destroying views (graphical representations of a network) and networks (the raw network data – not yet visualized), as well as an option for deleting selected nodes and edges from the current network. All deleted nodes and edges can be restored to the network via Edit → Undo. Editing preferences for properties and apps is found under Edit → Preferences → Properties... .
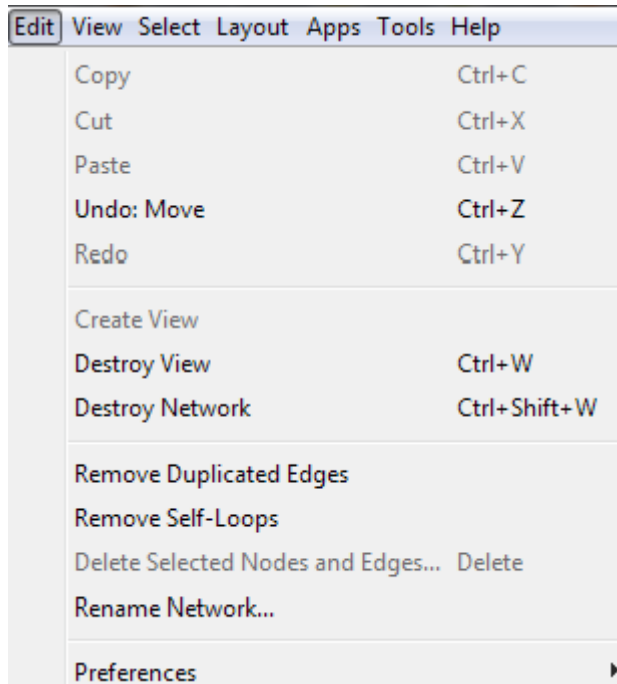
## View

The View menu allows you to display or hide the network management panel (Control panel), the attribute browser (Table panel), Tool panel and the Result Panel. And provides the control of other view related functionality.
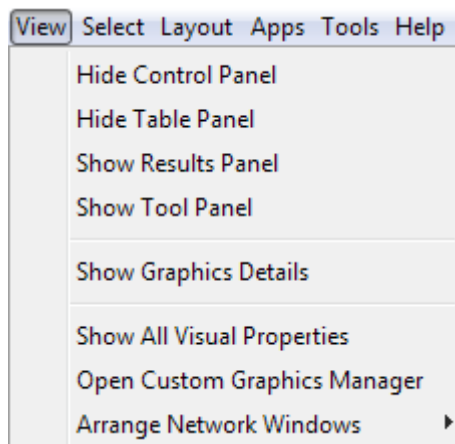


## Select

The Select menu contains different options for selecting nodes and edges. It also contains the Select → Use Filters option, which allows filters to be created for automatic selection of portions of a network whose node or edge attributes meet a filtering criterion.
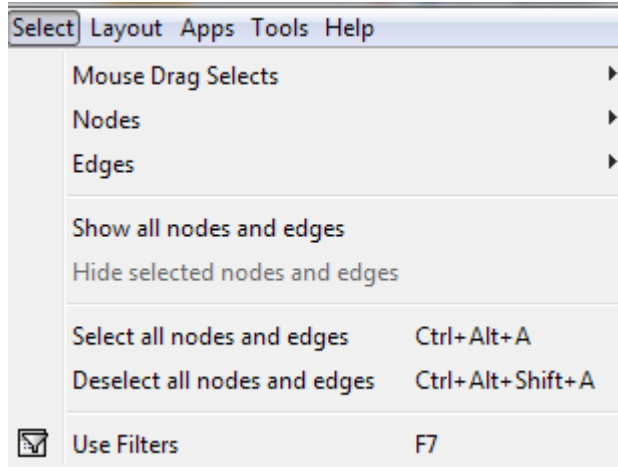
## Layout

The Layout menu has an array of features for visually organizing the network. The features in the top portion of the network (Bundle Edges, Rotate, Scale, Align and Distribute) are tools for manipulating the network visualization. The bottom section of the menu lists a variety of layout algorithms which automatically lay a network out.
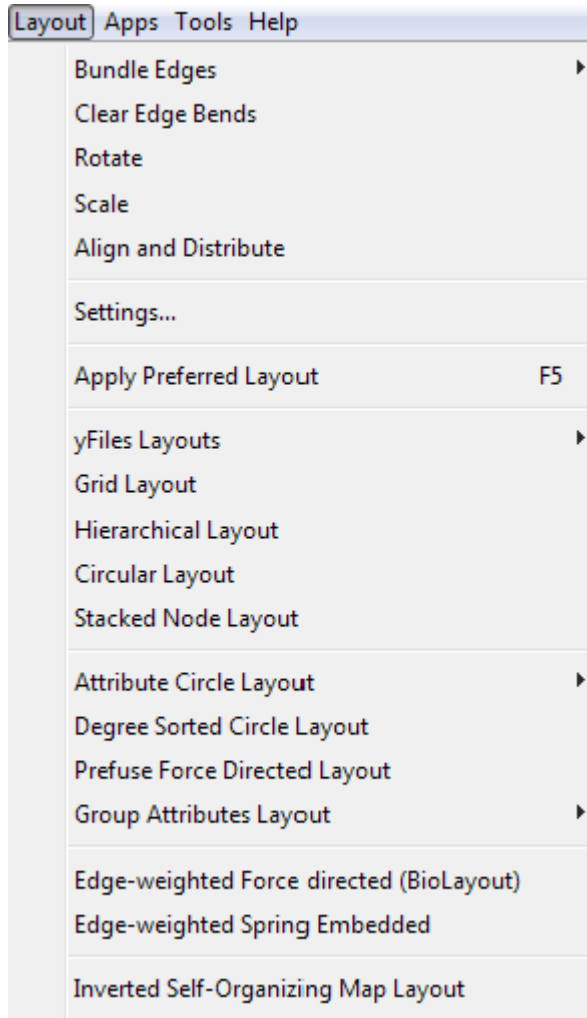
## Apps

The Apps menu contains option for managing (install/update/delete) your apps and may have options added by apps that have been installed. Depending on which apps are loaded, the apps that you see may be different than what appear here.
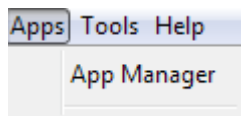


Note: A list of available Cytoscape apps with descriptions is available online at: http://apps.cytoscape.org

*In previous versions of Cytoscape, apps were called plugins and served a similar function. However, apps are not compatible with plugins.*

## Help

The Help menu allows you to launch the online help viewer and browse the table of contents for this manual. The "About…" option displays information about the running version of Cytoscape.
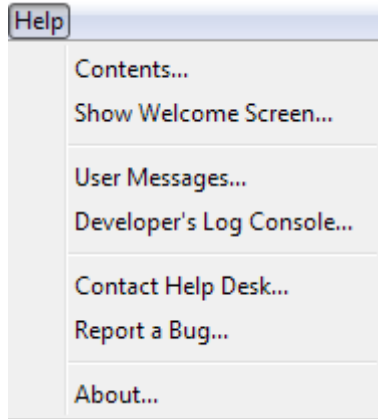
# Network Management

Cytoscape allows multiple networks to be loaded at a time, either with or without a view. A network stores all the nodes and edges that are loaded by the user and a view displays them.

An example where a number of networks have been loaded is shown below:



The network manager (in Control panel) shows the networks that are loaded. Clicking on a network here will make that view active in the main window, if the view exists. Each network has a name and size (number of nodes and edges), which are shown in the network manager. If a network is loaded from a file, the network name is the name of the file.

Some networks are very large (thousands of nodes and edges) and can take a long time to display. For this reason, a network in Cytoscape may not contain a 'view'. Networks that have a view are in normal black font and networks that don't have a view are highlighted in red. You can create or destroy a view for a

network by right-clicking the network name in the network manager or by choosing the appropriate option in the Edit menu. You can also destroy previously loaded networks this way.

Certain operations in Cytoscape will create new networks. If a new network is created from an old network, for example by selecting a set of nodes in one network and copying these nodes to a new network (via the File → New → Network option), it will be shown immediately follows the network that it was derived from.

The available network views are also arranged as multiple overlapping windows in the network view window. You can maximize, minimize, and destroy network views by using the normal window controls for your operating system.

## Arrange Network Windows

When you work on multiple networks, you can arrange the network view windows from *View* → *Arrange Network Windows*. You can re-arrange the network location by these commands.
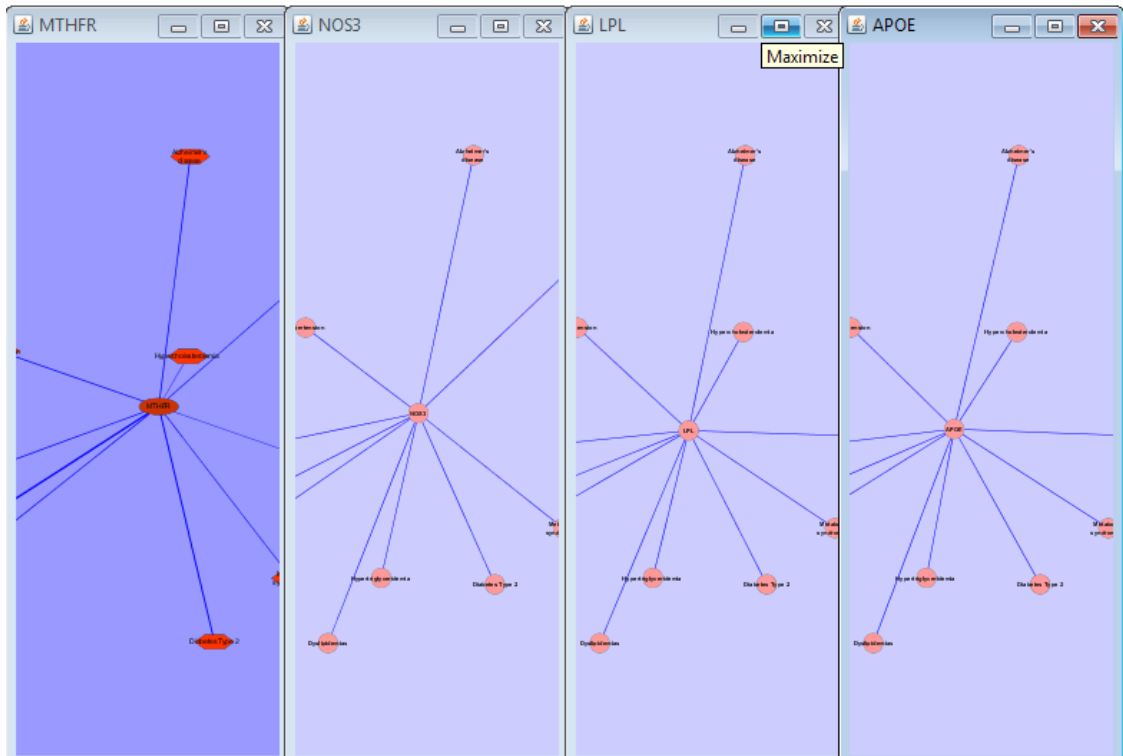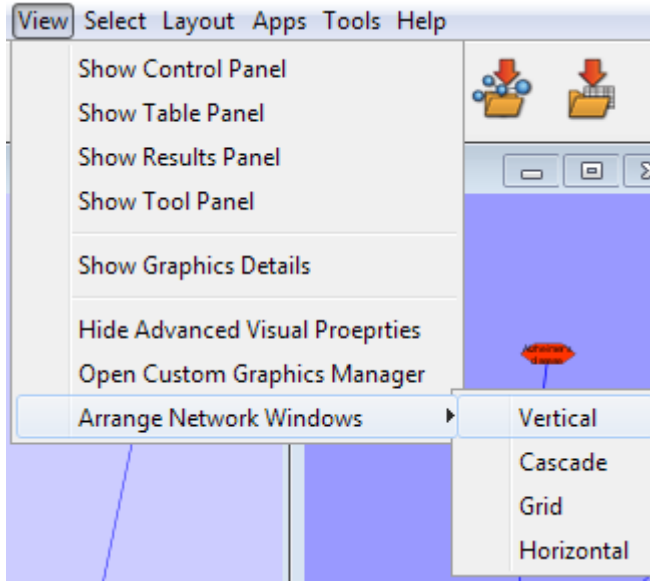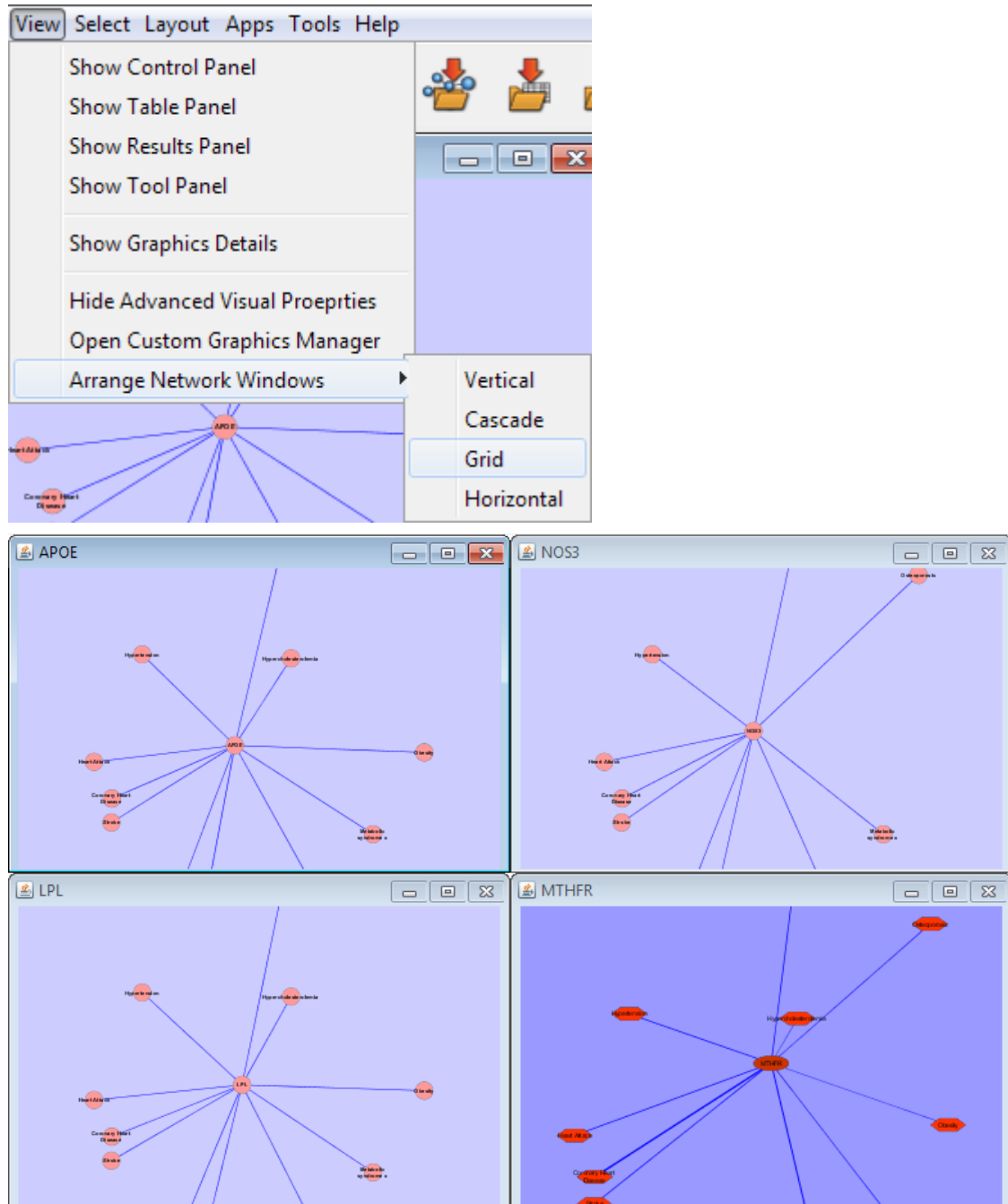
# The Network Overview Window

The network overview window shows an overview (or 'bird's eye view') of the network. It can be used to navigate around a large network view. The blue rectangle indicates the portion of the network currently displayed in the network view window, and it can be dragged with the mouse to view other portions of the network. Zooming in will cause the rectangle to appear smaller and vice versa.
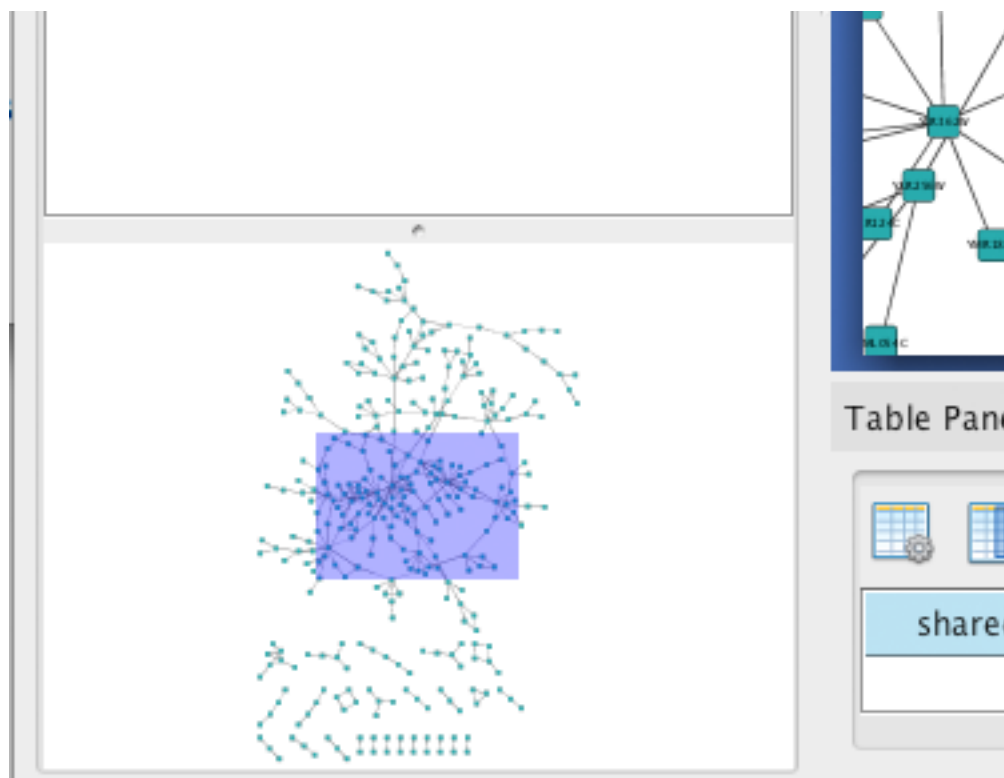
# Command Line Arguments

Cytoscape recognizes a number of optional command line arguments, including run-time specification of network files, attribute files, and session files. This is the output generated when the cytoscape is executed with the "-h" or "--help" flag:

```
usage: cytoscape.{sh|bat} [OPTIONS]
 -h,--help               Print this message.
 -v,--version            Print the version number.
 -s,--session <file>     Load a cytoscape session (.cys) file.
 -N,--network <file>     Load a network file (any format).
 -T,--table <file>       Load a data table file (any table format).
 -p,--plugin <file>      Load a SIMPLIFIED plugin jar file/URL.
 -b,--bundle <file>      Load a BUNDLE plugin jar file or URL.
 -P,--props <file>       Load cytoscape properties file (Java properties
                         format) or individual property: -P name=value.
 -V,--vizmap <file>      Load vizmap properties file (Cytoscape VizMap
                         format).
```

Note: This starts Cytoscape and then displays this message.

Any file specified for an option may be specified as either a path or as a URL. For example you can specify a network as a file (assuming that myNet.sif exists in the current working directory): `cytoscape.sh -N myNet.sif`. Or you can specify a network as a URL: `cytoscape.sh -N http://example.com/myNet.sif`.

| Argument | Description |
|---|---|
|  |  |

| -h,--help | This flag generates the help output you see above and exits. |
|---|---|
| -v,--version | This flag prints the version number of Cytoscape and exits. |
| -s,--session <file> | This option specifies a session file to be loaded. Since only one session file can be loaded at a given time, this option may only specified once on a given command line. The option expects a `.cys` Cytoscape session file. It is customary, although not necessary, for session file names to contain the .cys extension. |
| -N,--network <file> | This option is used to load all types of network files. SIF, GML, and XGMML files can all be loaded using the -N option. You can specify as many networks as desired on a single command line. |
| -e,--edge-attrs <file> | This option specifies an edge attributes file. You may specify as many edge attribute files as desired on a single command line. |
| -n,--node-attrs <file> | This option specifies a node attributes file. You may specify as many node attribute files as desired on a single command line. |
| -m,--matrix <file> | This option specifies a data matrix file. In a biological context, the data matrix consists of expression data. All data matrix files are read into node attributes. You may specify as many data matrix files as desired on a single command line. |
| -p,--plugin <file> | This option specifies a cytoscape plugin (.jar) file to be loaded by Cytoscape. This option also subsumes the previous "resource plugin option". You may specify a class name that identifies your plugin and the plugin will be loaded if the plugin is in Cytoscape's CLASSPATH. For example, assuming that the class MyPlugin can be found in the CLASSPATH, you could specify the plugin like this: `cytoscape.sh -p MyPlugin.class`. A final means of specifying plugins is to specify a file name whose contents contain a list of plugin jar files. |
| -P,--props <file> | This option specifies Cytoscape properties. Properties can be specified either as a properties file (in Java's standard properties format), or as individual properties. To specify individual properties, you must specify the property name followed by the property value where the name and value are separated by the '=' sign. For example to specify the defaultSpeciesName: `cytoscape.sh -P defaultSpeciesName=Human`. If you would like to include spaces in your property, simply enclose the name and value in quotation marks: `cytoscape.sh -P "defaultSpeciesName=Homo Sapiens"`. The property option subsumes previous options -noCanonicalization, -species, and -bioDataServer. Now it would look like: `cytoscape.sh -P defaultSpeciesName=Human -P noCanonicalization=true -P bioDataServer=myServer`. |
| -V,--vizmap <file> | This option specifies a visual properties file. |

All options described above (including plugins) can be loaded from the GUI once Cytoscape is running.

# Cytoscape Preferences

## Managing Properties

The Cytoscape properties editor, accessed via Edit → Preferences → Properties…, is used to specify general and default properties. Any changes made to these properties will be saved in the file `cytoscape3.props` in the `CytoscapeConfiguration` subdirectory of the user's home directory. Properties are now stored in Cytoscape session files, so changes to general properties will be saved as part of the current session.

Cytoscape properties are configurable using the Add, Modify and Delete buttons as seen below.
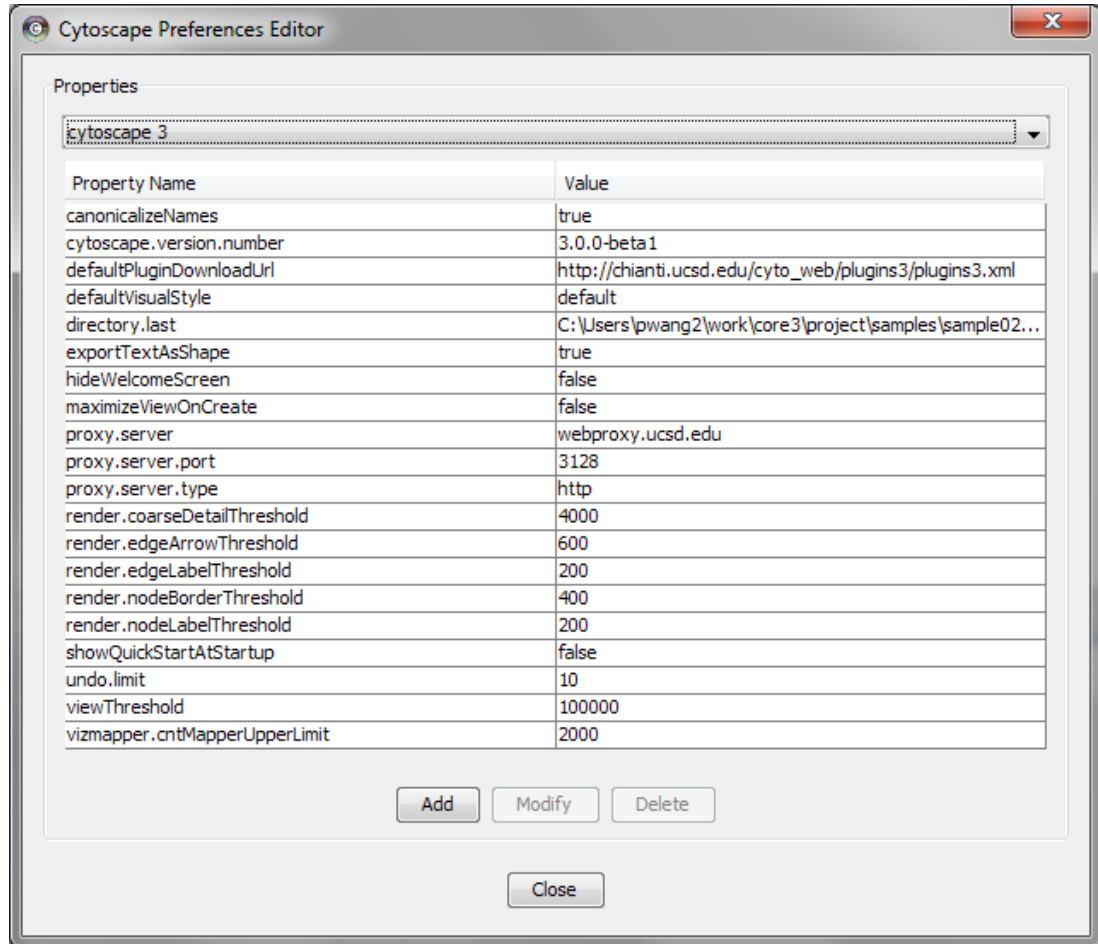
App properties may also be edited in the same way as editing Cytoscape properties. For example, to edit the properties of Linkout, select 'linkout' from the combobox of the Preferences Editor.
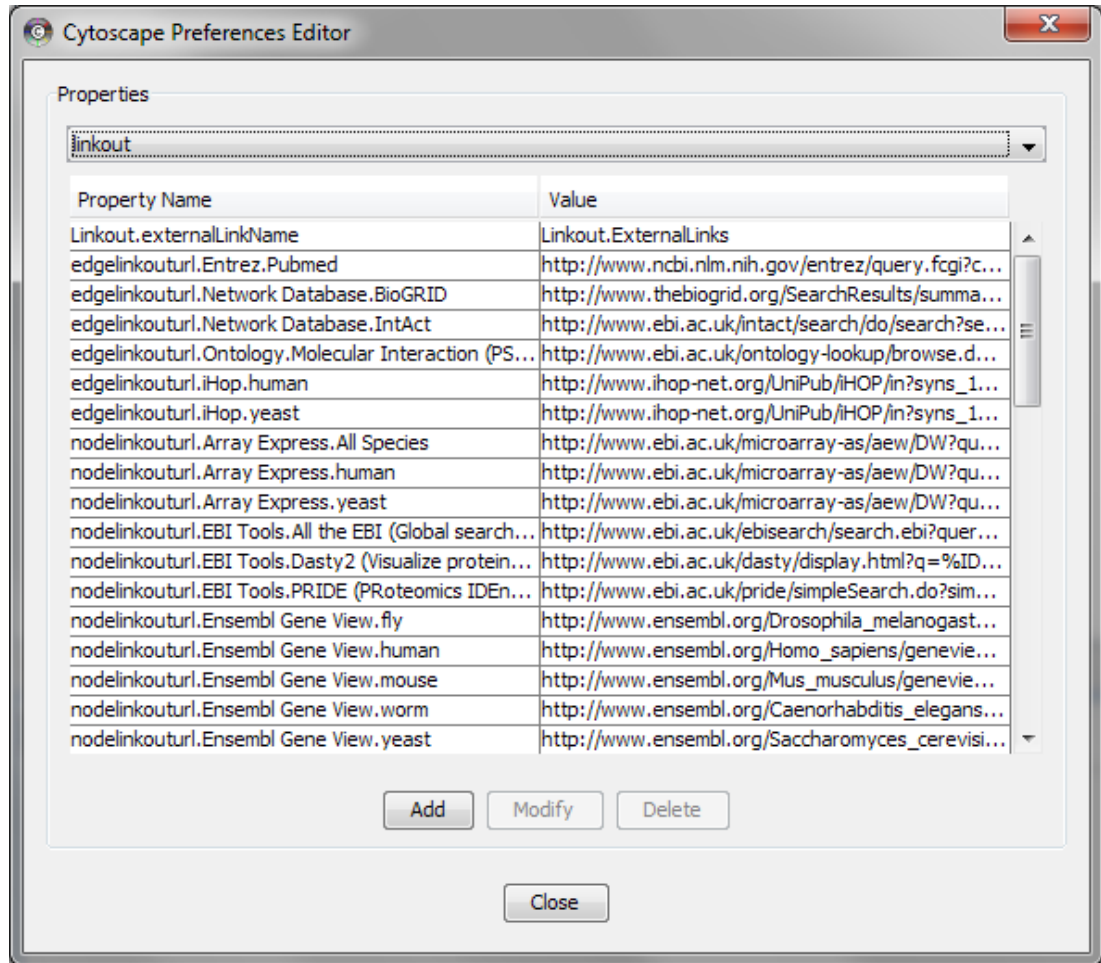
# Managing Bookmarks

Cytoscape contains a pre-defined list of bookmarks, which point to sample network files located on the Cytoscape web server. Users may add, modify, and delete bookmarks through the Bookmark manager, accessed by going to Edit → Preferences → Bookmarks… .
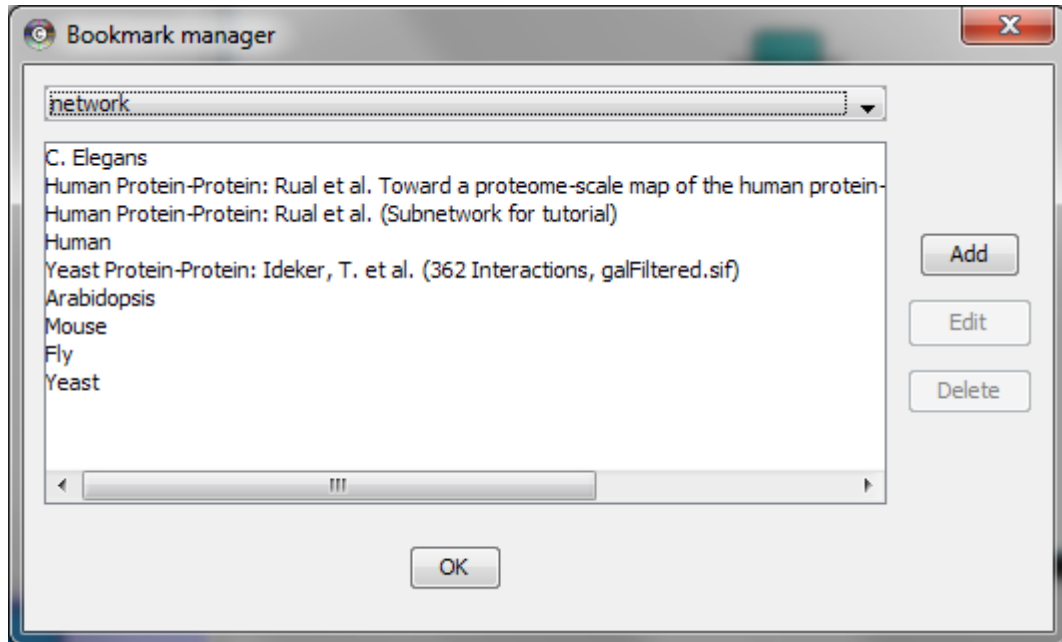
There are currently several types of bookmarks (based on data categories), including network and table. Network bookmarks are URLs pointing to Cytoscape network files. These are normal networks that can be loaded into Cytoscape. Table bookmarks are URLs pointing to data table files.

# Managing Proxy Servers

You can define and configure a proxy server for Cytoscape by going to Edit → Preferences → Proxy Settings... .
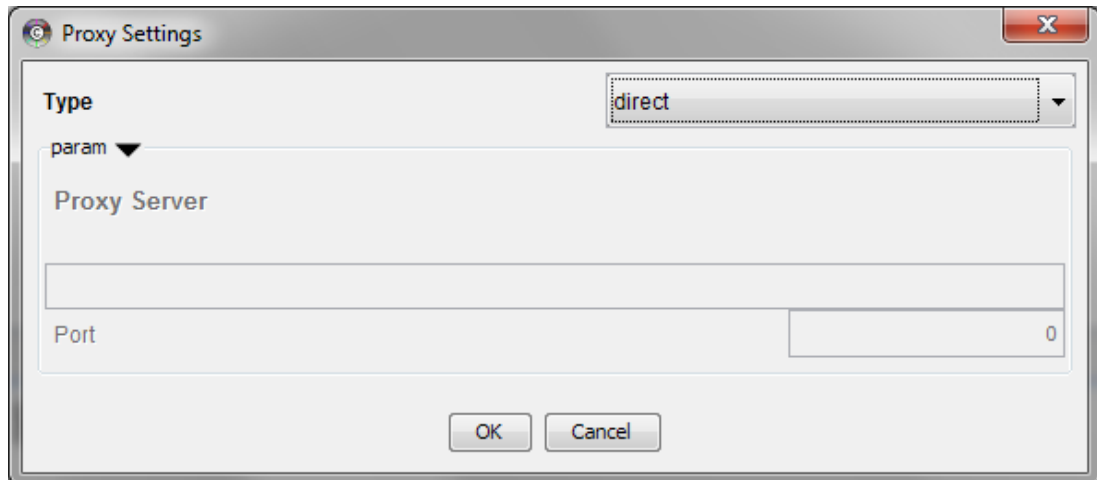


After the proxy server is set, all network traffic related to loading a network from URL will pass through the proxy server. Cytoscape apps use this capability as well. The proxy settings are saved in `cytoscape3.props`. Each time you click the OK button after making a change to the proxy settings, an attempt is made to connect to a well known site on the Internet (e.g., google.com) using your settings. For both success and failure you are notified and for failure you are given an opportunity to change your proxy settings.

If you no longer need to use a proxy to connect to the Internet, simply set the Proxy type to 'direct' and click the OK button.

# Managing Group View

The configuration of Cytoscape group view may also be edited through Edit → Preferences → Group preferences…
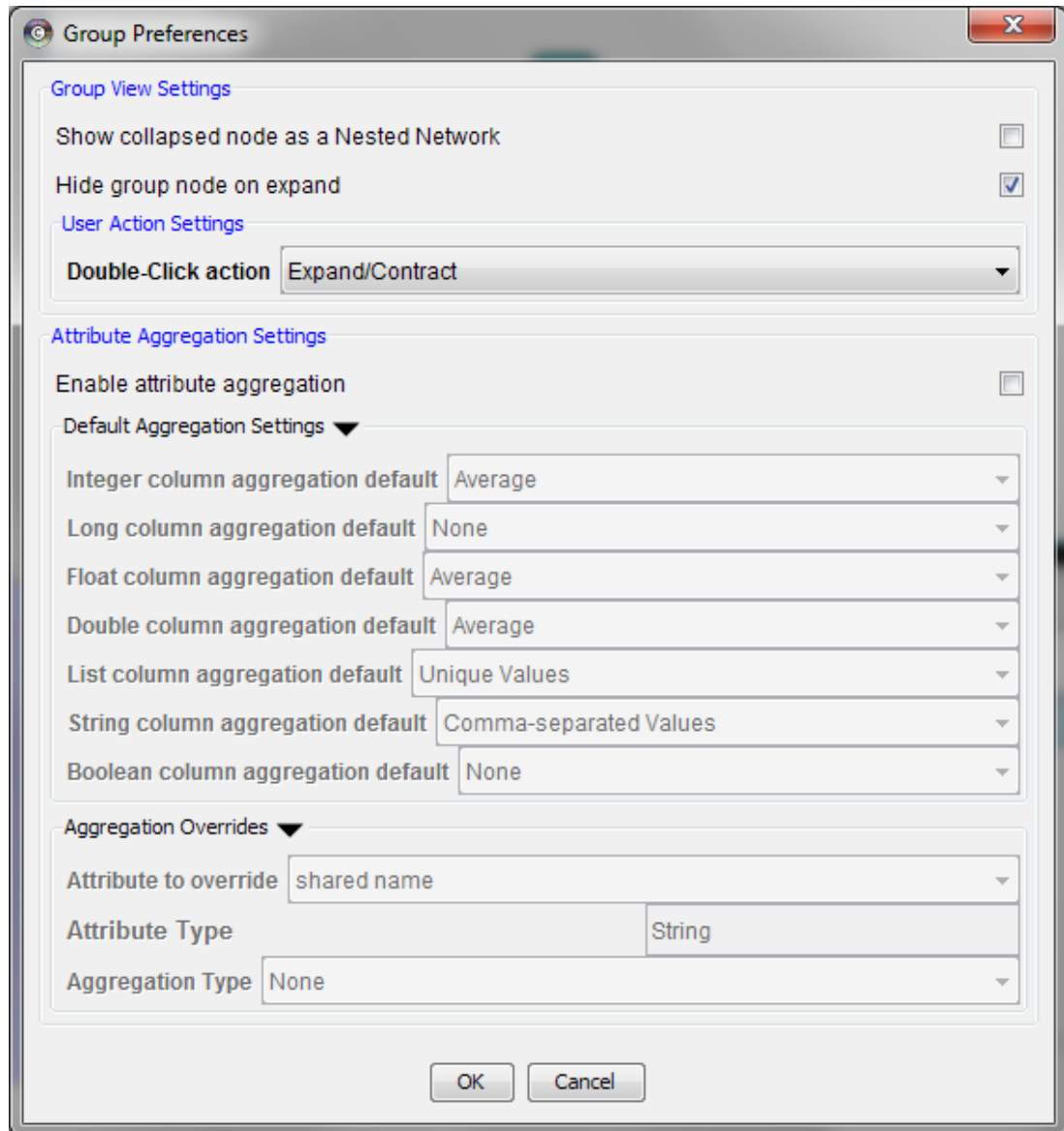


# Creating Networks

There are 4 different ways of creating networks in Cytoscape:

1. Importing pre-existing, fixed-format network files.

2. Importing pre-existing, unformatted text or Excel files.

3. Importing data from from public databases.

4. Creating an empty network and manually adding nodes and edges.

# Import Fixed-Format Network Files

Network files can be specified in any of the formats described in the Supported Network Formats chapter. Networks are imported into Cytoscape through the "Import Network" window, which can be accessed by going to File → Import → Network. The network file can either be located directly on the local computer, or found on a remote computer (in which case it will be referenced with a URL).
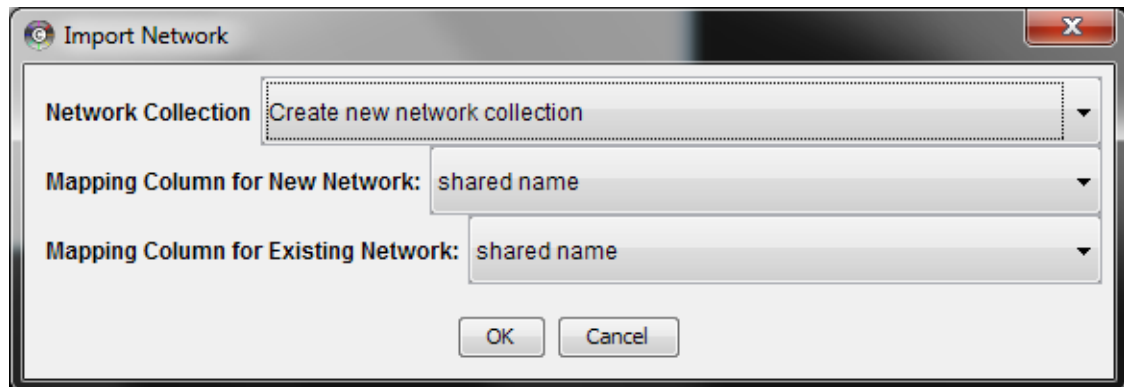
## Load Networks from Local Computer

In order to load network from local files you can select File → Import → Network → File... or click on



on the tool bar. Choose the correct file in the file chooser dialog and press Open. Some sample network files of different types have been included in the sampleData folder in Cytoscape.

After you choose a network file, another dialog will pop up. Here, you can choose either to create a new network collection for the new network, or load the new network into an existing network collection. When you choose the latter, make sure to choose the right mapping column to map the new network to the existing network collection.



Network files in SIF, GML, and XGMML formats may also be loaded directly from the command line using the –N option.

## Load Networks from a Remote Computer (URL import)

To load network from remote files you can select File → Import → Network → URL... or click on



on the tool bar. In the import network dialog, insert the appropriate URL, either manually or using URL bookmarks. Bookmarked URLs can be accessed by clicking on the arrow to the right of the text field (see the Bookmark Manager in Preferences for more details on bookmarks). Also, you can drag and drop links from a web browser to the URL text box. Once a URL has been specified, click on the OK button to load the network.
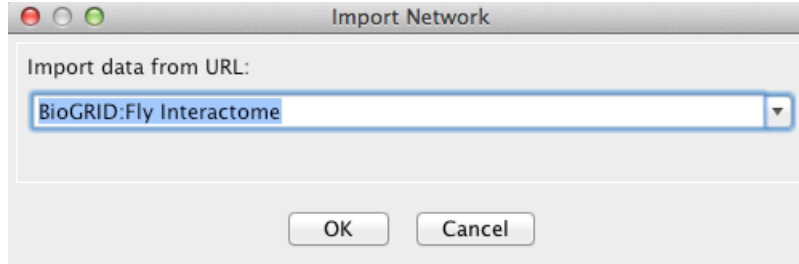
Another issue for network import is the presence of firewalls, which can affect which files are accessible to a computer. To work around this problem, Cytoscape supports the use of proxy servers. To configure a proxy server, go to Edit → Preferences→ Proxy Server... . This is further described in the Preferences chapter.

# Import Unformatted Table Files

Cytoscape supports the import of networks from delimited text files and Excel workbooks using Edit → Import → Network from Table (Text/MS Excel)... . An interactive GUI allows users to specify parsing options for specified files. The screen provides a preview that shows how the file will be parsed given the current configuration. As the configuration changes, the preview updates automatically. In addition to specifying how the file will be parsed, the user must also choose the columns that represent the source and target nodes as well as an optional edge interaction type.
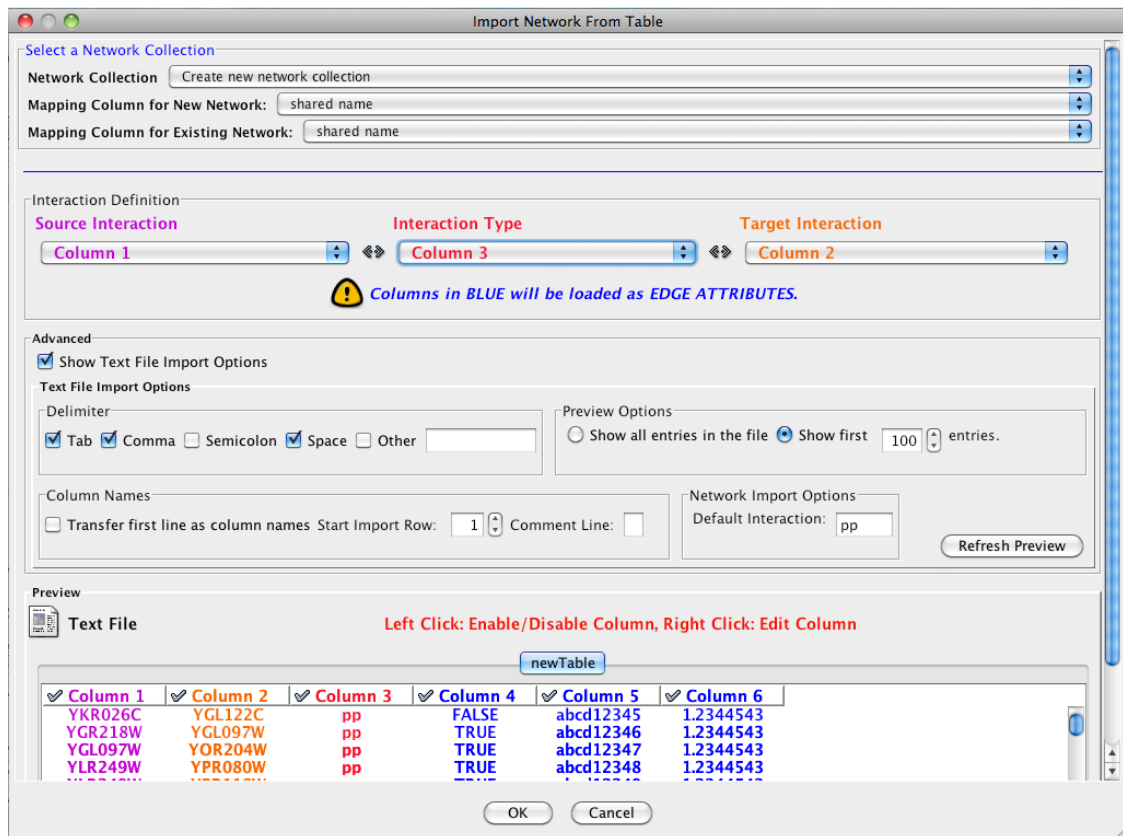
## Supported Files

The "Import Network from Table" function supports delimited text files and single-sheet Microsoft Excel Workbooks. The following is a sample table file:

```
source   target  interaction  boolean attribute  string attribute        floating p
YJR022W YNR053C pp           TRUE     abcd12371         1.2344543
YER116C YDL013W pp           TRUE     abcd12372         1.2344543
YNL307C YAL038W pp           FALSE    abcd12373         1.2344543
YNL216W YCR012W pd           TRUE     abcd12374         1.2344543
YNL216W YGR254W pd           TRUE     abcd12375         1.2344543
```

The network table files should contain at least two columns for creating network with edges. If the file has only one column, the created network will not contain any edges. The interaction type is optional in this format. Therefore, a minimal network table looks like the following:

```
YJR022W YNR053C
YER116C YDL013W
YNL307C YAL038W
YNL216W YCR012W
YNL216W YGR254W
```

One row in a network table file represents an edge and its edge attributes. This means that a network file is considered a combination of network data and edge attributes. A table may contain columns that aren't meant to be edge attributes. In this case, you can choose not to import those columns by clicking on the column header in the preview window. This function is useful when importing a data table like the following (1):

| Unique ID A | Unique ID B | Alternative ID A | Alternative ID B | Aliases A | Aliases B | Interaction detection methods | First author surnames | Pubmed IDs | species A | species B | Interactor types | Source database | Interaction ID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7205 | 5747 | TRIP6 | PTK2 | Q15654 | Q05397-1 | vv\|HPRD | Currently not available | 14688263\|15892868(Marcotte) | Mammalia | Homo sapiens | protein\|protein | HPRD\|Marcotte | 0 |
| 4174 | 7311 | MCM5 | UBA52 | P33992 | P62987 | neighbouring_reaction | Currently not available | 15608231(Reactome) | Homo sapiens | Homo sapiens | protein\|protein | Reactome | 1 |
| 7040 | 7040 | TGFB1 | TGFB1 | P01137 | P01137 | nmr: nuclear magnetic resonance | Currently not available | 8679613 | Homo sapiens | Homo sapiens | protein\|protein | BIND | 2 |

This data file is a tab-delimited text and contains network data (interactions), edge attributes, and node attributes. To import network and edge attributes from this table, you need to choose Unique ID A as source, Unique ID B as target, and Interactor types as interaction type. Then you need to turn off columns used for node attributes (Alternative ID A, species B, etc.). Other columns can be imported as edge attributes.

The network import function cannot import node attributes - only edge attributes. To import node attributes from this table, please see the Attributes section of this manual.

Note (1): This data is taken from the *A merged human interactome* datasets by Andrew Garrow, Yeyejide Adeleye and Guy Warner (Unilever, Safety and Environmental Assurance Center, 12 October 2006). Actual data files are available at http://www.cytoscape.org/cgi-bin/moin.cgi/Data_Sets/

## Basic Operations

To import network text/Excel tables, please follow these steps:

1. Select File → Import → Network → File... or click on

   

   on the tool bar.

2. Select a table file in the file chooser dialog.

3. Define the interaction parameters by specifying which columns of data contain the Source Interaction, Target Interaction, and Interaction Type. Setting the Interaction Type as Default Interaction will result in all interactions being given the value pp; this value can be modified in Advanced Options (below).

4. (Optional) Define edge attribute columns, if applicable. Network table files can have edge attribute columns in addition to network data.

   • Enable/Disable Attribute Column - By *left*-clicking on a column header in the preview table, you can enable/disable edge attributes. If the header is checked and entries are blue, the column will be imported as an edge attribute. For example, the table below shows that columns 1 through 3 will be used as network data, column 4 will not be imported, and columns 5 and 6 will be imported as edge attributes.



   • Change Attribute Name and Data Types - If you *right*-click on a column header in the preview table, you can modify the attribute name and data type. For more detail, see "Modify Attribute Name/Type" below.

5. Click the Import button.

## Import List of Nodes Without Edges

The table import feature supports lists of nodes without edges. If you select only a source column, it creates a network without interactions. This feature is useful with the node expansion function available from some web service clients. Please read the section *Importing Networks from External Database* for more detail.

# Advanced Options



You can select several options by checking the Show Text File Import Options checkbox.

• Delimiter: You can select multiple delimiters for text tables. By default, Tab and Space are selected as delimiters.

- Preview Options: When you select a network table file, the first 100 entries will be displayed in the preview panel. To display more entries, change the value for this option. If you want to show all entries in the file, select "Show all entries in the file". You will need to click the Reload button to update the Preview panel.

- Attribute Names

  - Transfer first line as attribute names: Selecting this option will cause all edge attribute columns to be named according to the first data entry in that column.

  - Start Import Row: Set which row of the table to begin importing data from. For example, if you want to skip the first 3 rows in the file, set 4 for this option.

  - Comment Line: Rows starting with this character will not be imported. This option can be used to skip comment lines in text files.

- Network Import Options: If the Interaction Type is set to Default Interaction, the value here will be used as the interaction type for all edges.

## Modify Attribute Name/Type
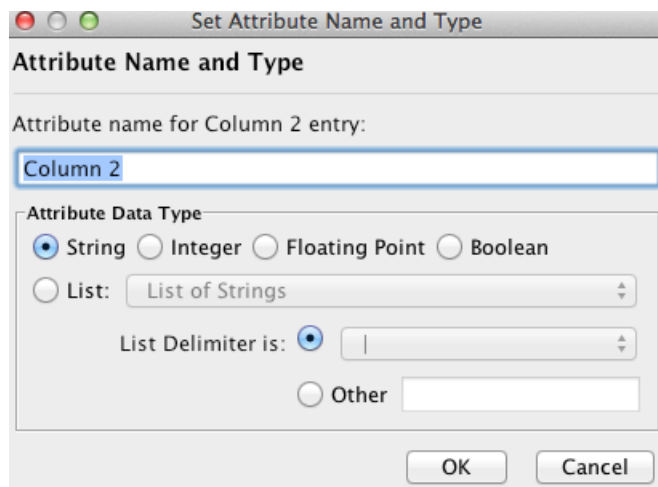


Attribute names and data types can be modified here.

- Modify Attribute Name - just enter a new attribute name and click OK.

- Modify Attribute Data Type - The following attribute data types are supported:

  - String

  - Boolean (True/False)

  - Integer

  - Floating Point

  - List of (one of) String/Boolean/Integer/Floating Point

Cytoscape has a basic data type detection function that automatically suggests the attribute data type of a column according to its entries. This can be overridden by selecting the appropriate data type from the radio

buttons provided. For lists, a global delimiter must be specified (i.e., all cells in the table must use the same delimiter).

# Import Networks from Public Databases

Cytoscape has a feature called **Import Network from Public Databases**. Users can access various kinds of databases through this function, File → Import → Network → from Public Databases...

## Web Service Client Manager

Cytoscape has a feature called **Web Service Client Manager**. This is a framework to manage various kinds of web service clients in Cytoscape. By using web service clients, users can access remote data sources easily.

### What is a Web Service?

A web service is a standardized, platform-independent mechanism for machines to interact over the network. These days, many major biological databases publish their data with web service API:

- List of Biological Web Services: http://taverna.sourceforge.net/services

- Web Services at the EBI: http://www.ebi.ac.uk/Tools/webservices/

This enables developers to write a program to access these services. Cytoscape core developer team have developed several web service clients using this framework. Cytoscape supports many web services including:

- PSICQUIC: Standard web service for biological interaction data sets. The following data providers support PSICQUIC:

  - APID

  - Bar

  - BIND

  - BindingDB

  - GeneMANIA

  - I2Z-IMEX

  - ChEMBL

  - BioGrid

  - InnateDB

  - InnateDB-IMEx

  - Interoporc

  - MB-Info

  - Molcon

- Spike

- TopFind

- uniProt

- VirHostNet

- DIP

- IntAct

- MatrixDB

- MPIDB

- Reactome

- Reactome-FIs

- MINT

- iRefIndex

- STRING

- Pathway Commons: an open source portal, providing access to multiple integrated data sets, including: Reactome, IntAct, HPRD, HumanCyc, MINT, the MSKCC Cancer Cell Map, and the NCI/Nature Pathway Interaction database.

In the following sections, users learn how to import network from external databases.

# Getting Started

To get started, select: File → Import → Network → from Public Databases...
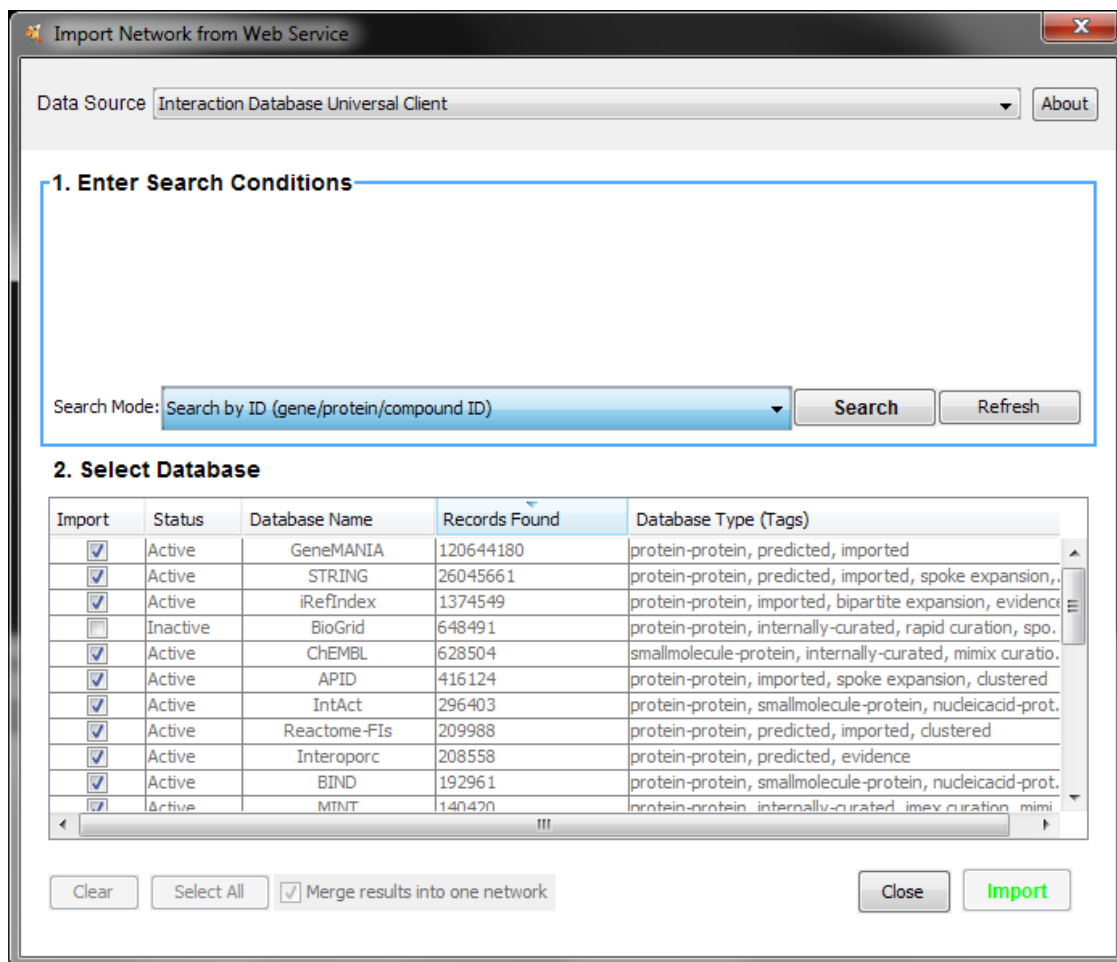
# Example #1: Retrieving Protein-Protein Interaction Networks from IntAct

- Select: File → Import → Network →from Public Databases...

- From the pull-down menu, select the Interaction databases Universal Client.

- Enter one or more search terms, such as BRCA1

- Click the button, Search

- Select Database, intAct

- Click the Import button.
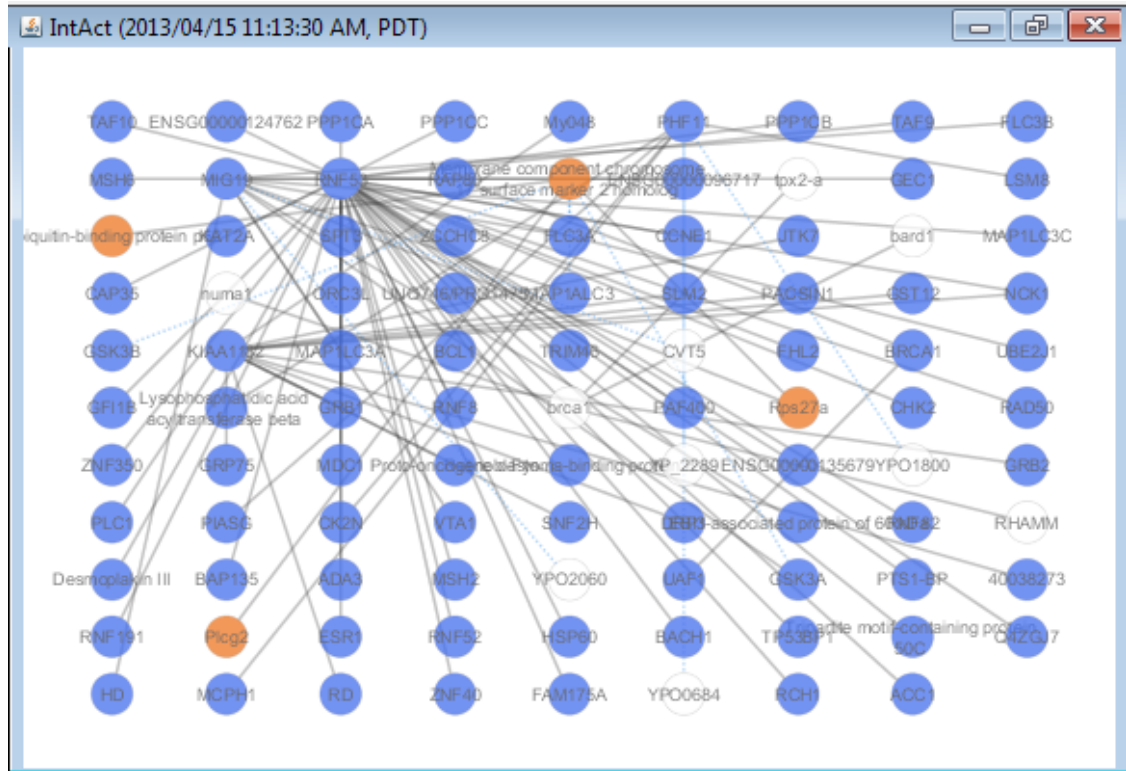
After confirming the download of interaction data, the network of BRCA1 will be imported and visualized.
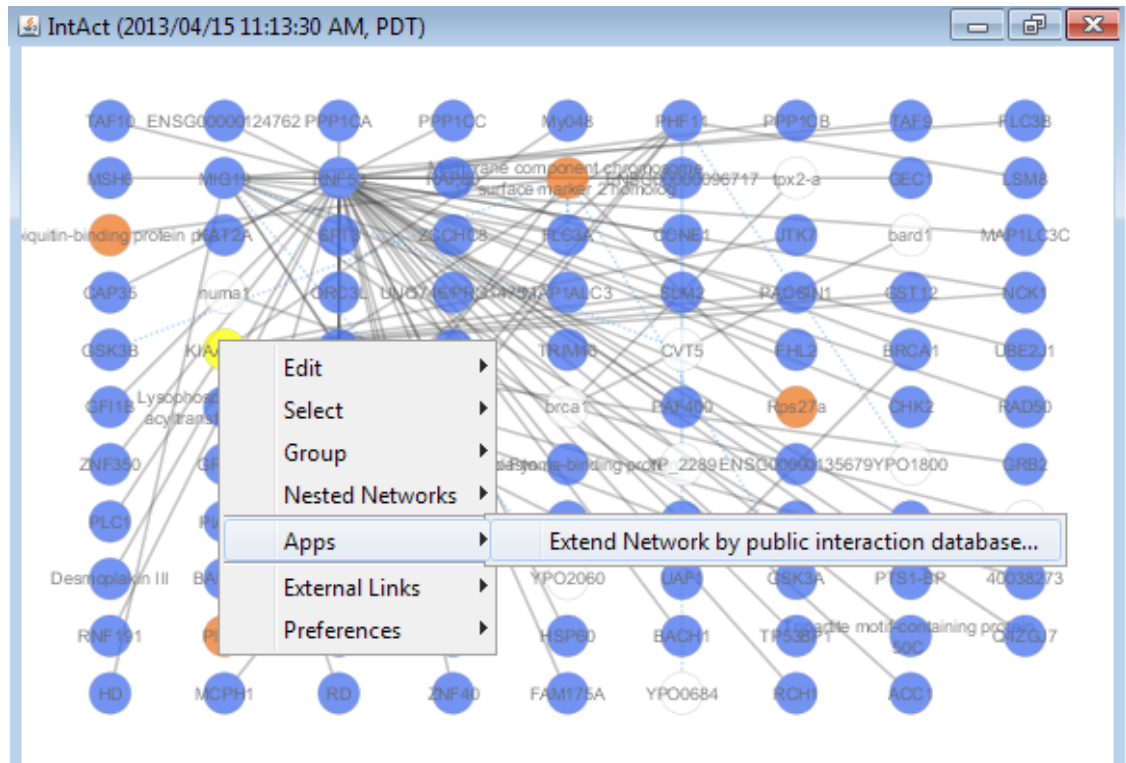
**Tip: Expanding the Network:** Several of the Cytoscape web services provide additional options in the node context menu. To access these options, right-click on a node and select "Apps → Extend Network by public interaction database..." For example, in the screenshot, we have loaded the BRCA1 network from IntAct, and have chosen to merge this node's neighbors into the existing network.
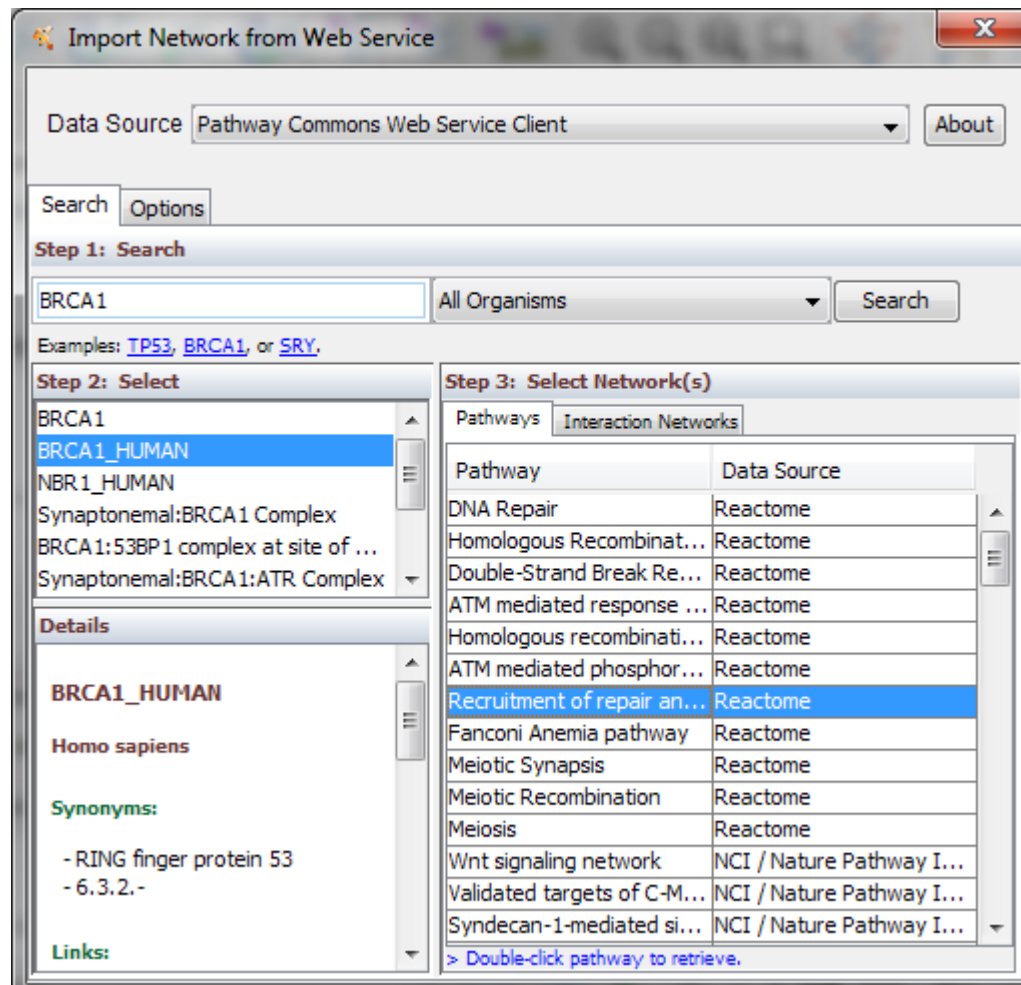
# Example #2: Retrieving Pathways and Networks from Pathway Commons

- Select: File → Import → Network → from public database...

- From the pull-down menu, select the Pathway Commons Web Service Client.

Then, follow the three-step process outlined below:



- Step 1: Enter your search term; for example: BRCA1

- Step 2: Select the protein or small molecule of interest. Full details regarding each molecule is shown in the bottom left panel.

- Step 3: Download a specific pathway or interaction network.

# Downloading Pathways and Interaction Networks

In Step 3, you can simply double-click on a pathway of interest, or click on the Interaction Networks tab. The Interaction Networks tab enables you to filter interactions by data source and/or interaction type. For example, you can choose to restrict your network to direct physical interactions from intAct and MINT only:
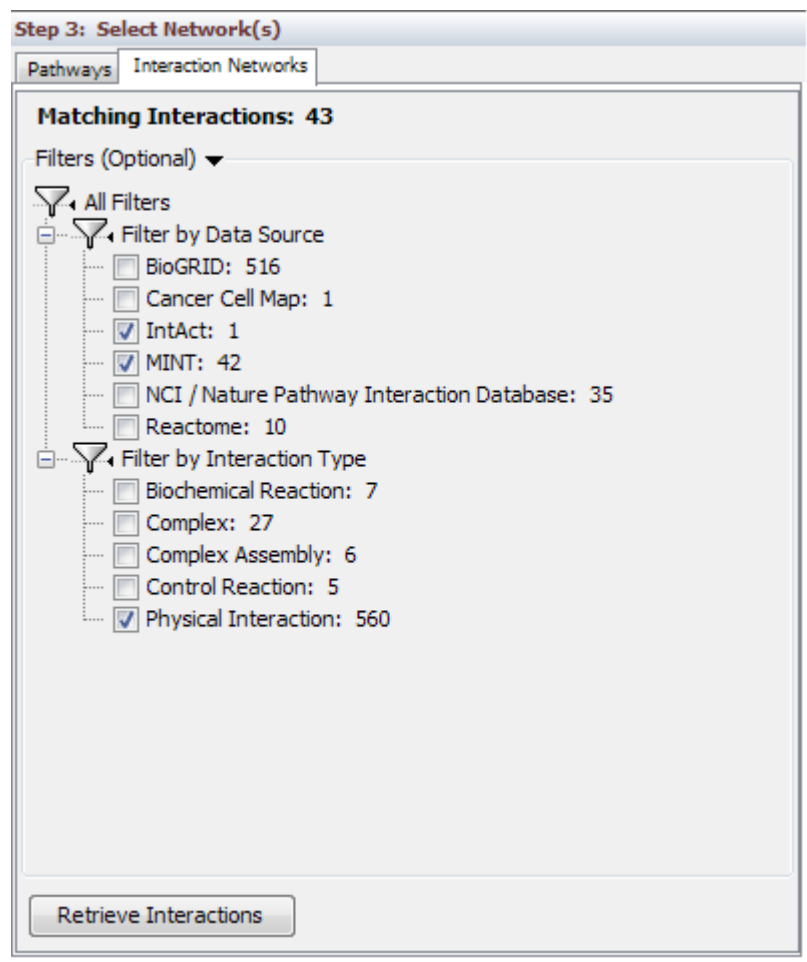


# Pathway Commons Options

You can configure access options from the Options tab. There are two retrieval options:

- Simplified Binary Model: Retrieve a simplified binary network, as inferred from the original BioPAX representation. In this representation, nodes within a network refer to physical entities only, and edges refer to inferred interactions.

- Full Model: Retrieve the full model, as stored in the original BioPAX representation. In this representation, nodes within a network can refer to physical entities and interactions.

By default, the simplified binary model is selected.

## Create a New Network Manually

A new, empty network can also be created and nodes and edges manually added. To create an empty network, go to File → New → Network → Empty Network, and then manually add network components by right clicking on the network canvas or on a node.

# Supported Network File Formats

Cytoscape can read network/pathway files written in the following formats:

- Simple interaction file (SIF or .sif format)

- Nested network format (NNF or .nnf format)

- Graph Markup Language (GML or .gml format)

- XGMML (extensible graph markup and modelling language).

- SBML

- BioPAX

- PSI-MI Level 1 and 2.5

- GraphML

- Delimited text

- Excel Workbook (.xls, .xlsx)

The SIF format specifies nodes and interactions only, while other formats store additional information about network layout and allow network data exchange with a variety of other network programs and data sources. Typically, SIF files are used to import interactions when building a network for the first time, since they are easy to create in a text editor or spreadsheet. Once the interactions have been loaded and network layout has been performed, the network may be saved to GML or XGMML format for interaction with other systems. All file types listed (except Excel) are text files and you can edit and view them in a regular text editor.

## SIF Format

The simple interaction format is convenient for building a graph from a list of interactions. It also makes it easy to combine different interaction sets into a larger network, or add new interactions to an existing data set. The main disadvantage is that this format does not include any layout information, forcing Cytoscape to re-compute a new layout of the network each time it is loaded.

Lines in the SIF file specify a source node, a relationship type (or edge type), and one or more target nodes:

```
nodeA <relationship type> nodeB
nodeC <relationship type> nodeA
nodeD <relationship type> nodeE nodeF nodeB
nodeG
...
nodeY <relationship type> nodeZ
```

A more specific example is:

```
node1 typeA node2
node2 typeB node3 node4 node5
node0
```

The first line identifies two nodes, called node1 and node2, and a single relationship between node1 and node2 of type typeA. The second line specifies three new nodes, node3, node4, and node5; here "node2" refers to the same node as in the first line. The second line also specifies three relationships, all of type typeB and with node2 as the source, with node3, node4, and node5 as the targets. This second form is simply shorthand for specifying multiple relationships of the same type with the same source node. The third line indicates how to specify a node that has no relationships with other nodes. This form is not needed for nodes that do have relationships, since the specification of the relationship implicitly identifies the nodes as well.

Duplicate entries are ignored. Multiple edges between the same nodes must have different edge types. For example, the following specifies two edges between the same pair of nodes, one of type xx and one of type yy:

```
node1 xx node2
node1 xx node2
node1 yy node2
```

Edges connecting a node to itself (self-edges) are also allowed:

```
node1 xx node1
```

Every node and edge in Cytoscape has a name attribute. For a network defined in SIF format, node names should be unique, as identically named nodes will be treated as identical nodes. The name of each node will be the name in this file by default (unless another string is mapped to display on the node using the visual mapper). This is discussed in the section on visual styles. The name of each edge will be formed from the name of the source and target nodes plus the interaction type: for example, `sourceName (edgeType) targetName`.

The tag <edgeType> can be any string. Whole words or concatenated words may be used to define types of relationships, e.g. geneFusion, cogInference, pullsDown, activates, degrades, inactivates, inhibits, phosphorylates, upRegulates, etc.

Some common interaction types used in the Systems Biology community are as follows:

```
  pp .................. protein — protein interaction
  pd .................. protein -> DNA
  (e.g. transcription factor binding upstream of a regulating gene.)
```

Some less common interaction types used are:

```
  pr .................. protein -> reaction
  rc .................. reaction -> compound
  cr .................. compound -> reaction
  gl .................. genetic lethal relationship
  pm .................. protein-metabolite interaction
  mp .................. metabolite-protein interaction
```

## Delimiters

Whitespace (space or tab) is used to delimit the names in the simple interaction file format. However, in some cases spaces are desired in a node name or edge type. The standard is that, if the file contains any tab characters, then tabs are used to delimit the fields and spaces are considered part of the name. If the file contains no tabs, then any spaces are delimiters that separate names (and names cannot contain spaces).

If your network unexpectedly contains no edges and node names that look like edge names, it probably means your file contains a stray tab that's fooling the parser. On the other hand, if your network has nodes whose names are half of a full name, then you probably meant to use tabs to separate node names with spaces.

Networks in simple interactions format are often stored in files with a `.sif` extension, and Cytoscape recognizes this extension when browsing a directory for files of this type.

# NNF

The NNF format is a very simple format that unlike SIF allows the optional assignment of single nested network per node. No other node attributes can be specified. There are only 2 possible line formats:

- A node "node" contained in a "network:"

```
network node
```

- 2 nodes linked together contained in a network:

```
network node1 interaction node2
```

If a network name (first entry on a line) appeared previously as a node name (in columns 2 or 4), the network will be nested in the node with the same name. Also, if a name that has been previously defined as a network (by being listed in the first column), later appears as a node name (in columns 2 or 4), the previously defined network will be nested in the node with the same name. In summary: any time a name is used as both, a network name , and a node name, this implies that the network will be nested in the node of the same name. Additionally comments may be included on all lines. Comments start with a hash mark '#' and continue to the end of a line. Trailing comments (after data lines) and entirely blank lines anywhere are also permissible. Please **note** that if you load multiple NNF files in Cytoscape they will be treated like a single, long concatenated NNF file! If you need to embed spaces, tabs or backslashes in a name, you must escape it by preceding it with a backslash, so that, e.g. an embedded backslash becomes two backslashes, an embedded space a backslash followed by a space etc.

# Examples

## Example 1
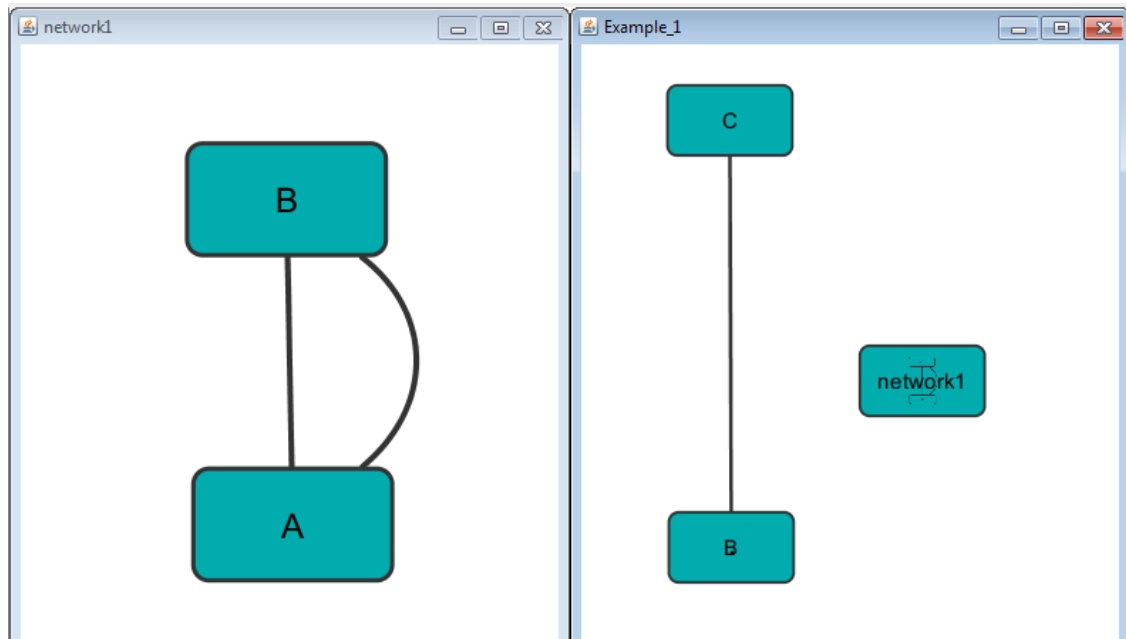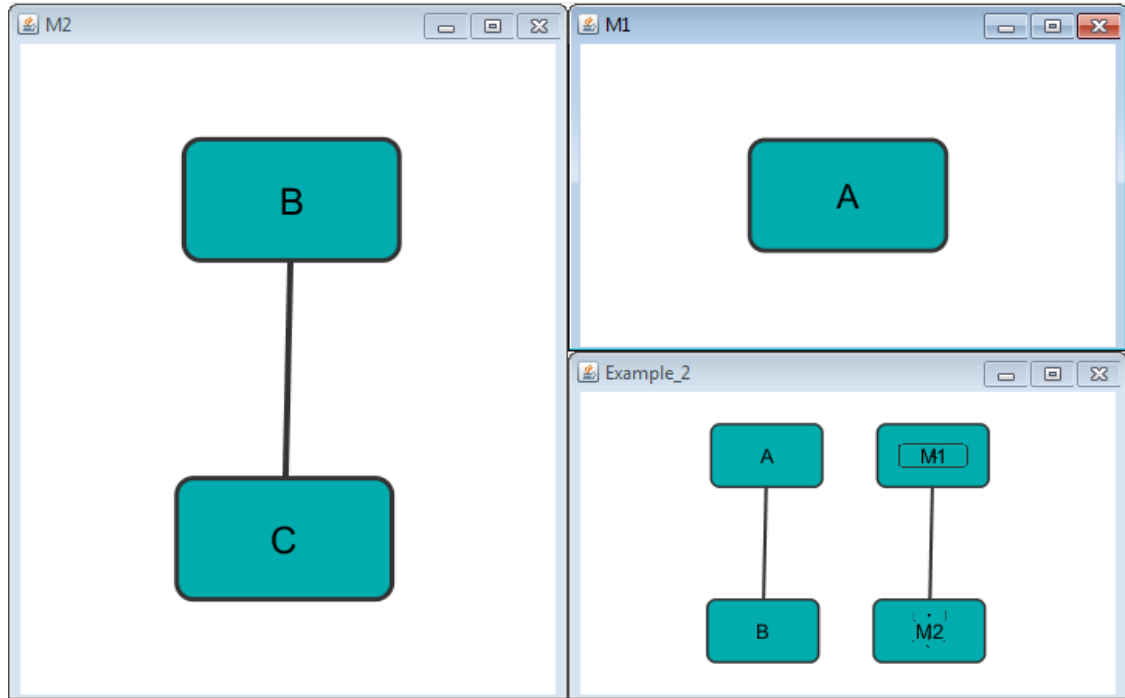


```
Example_1      C
Example_1      network1
network1       A         pp        B
network1       B         pp        A
Example_1      C         pp        B
```
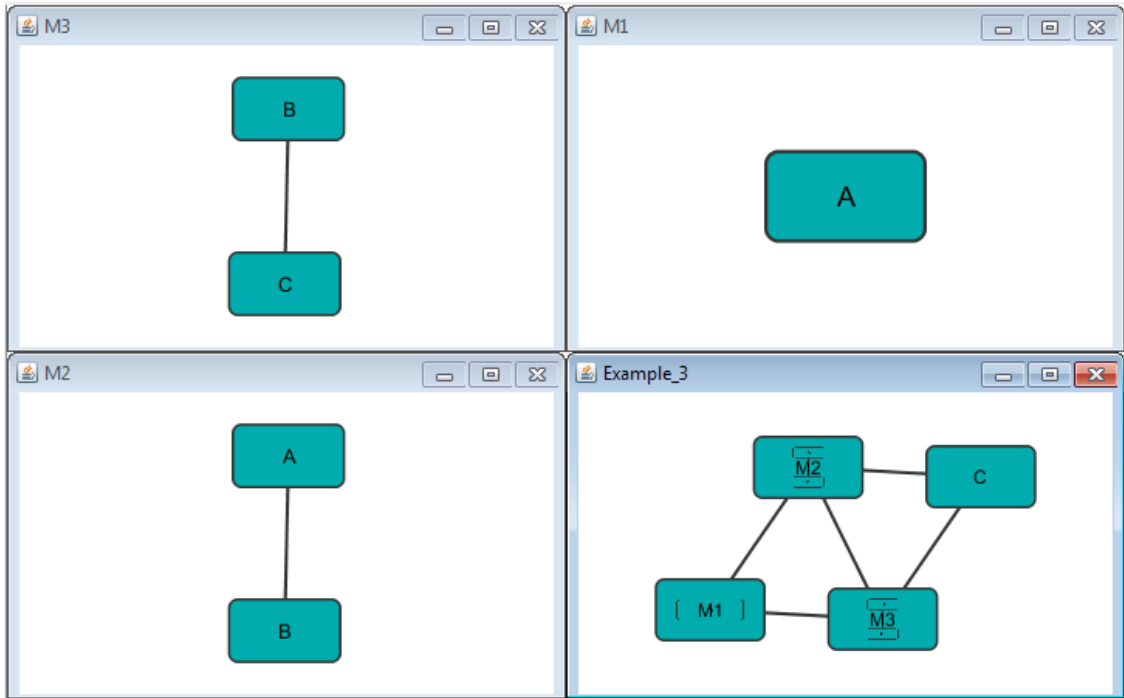
## Example 2



```
Example_2        M1
Example_2        M2
M1               A
M2               B          pp         C
Example_2        A          pp         B
Example_2        M1         im         M2
```
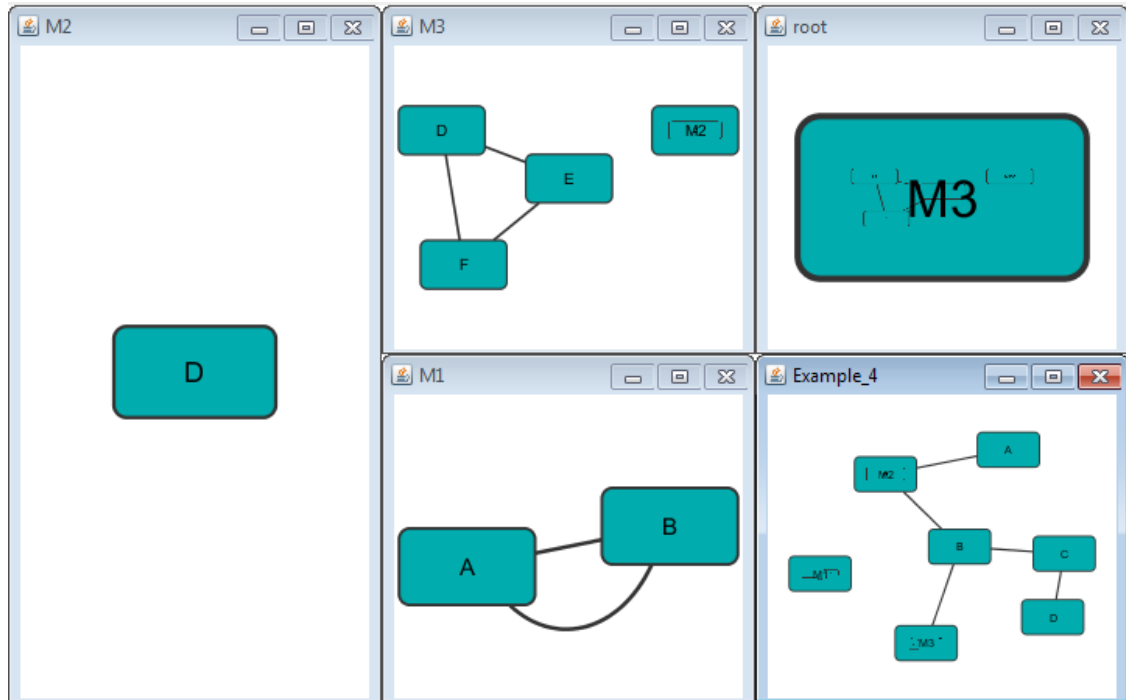
## Example 3



```
Example_3      M1      im      M2
Example_3      M3      im      M1
Example_3      M2      im      M3
Example_3      C       pp      M3
Example_3      M2      pp      C
M1             A
M2             A       pp      B
M3             B       pp      C
```
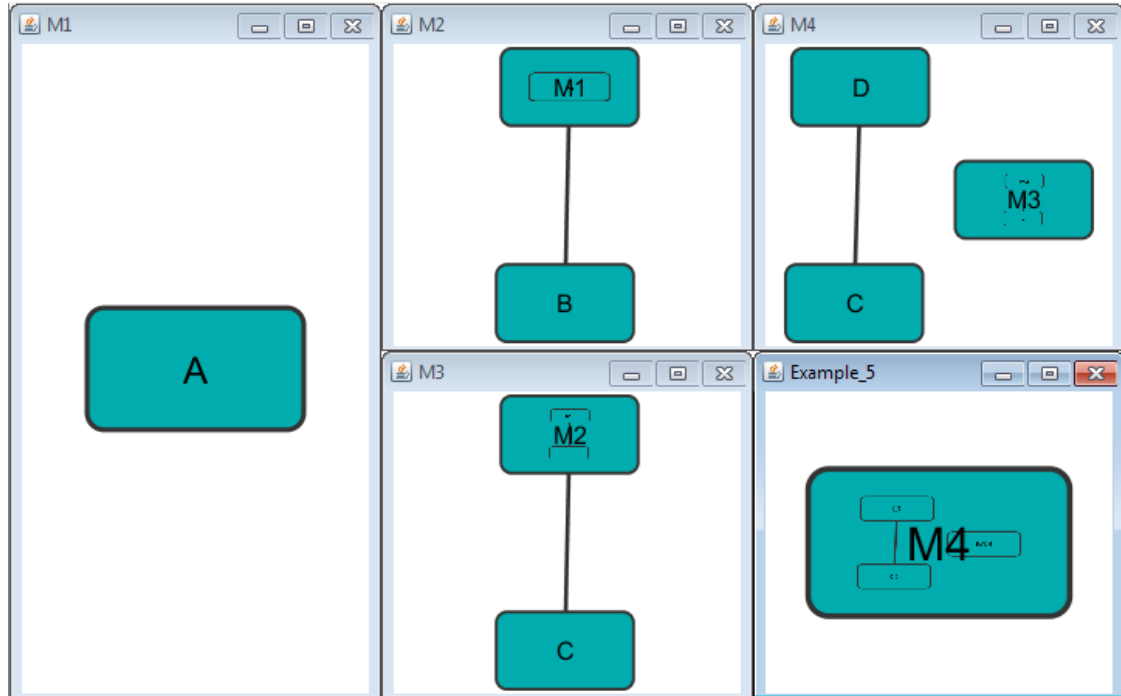
## Example 4



```
Example_4       M1
root            M3
M1              A       pp      B
M1              B       pp      A
Example_4       C       pp      B
M3              M2
M2              D
M3              E       pp      F
M3              D       pp      F
M3              D       pp      E
Example_4       D       pp      C
Example_4       A       pp      M2
Example_4       B       pp      M3
Example_4       M2      pp      B
```

**Example 5**



```
Example_5      M4
M4             D
M4             M3
M3             M2        pp        C
M2             M1        pp        B
M1             A
M4             C         pp        D
```

# GML Format

In contrast to SIF, GML is a rich graph format language supported by many other network visualization packages. The GML file format specification is available at:

http://www.infosun.fmi.uni-passau.de/Graphlet/GML/

It is generally not necessary to modify the content of a GML file directly. Once a network is built in SIF format and then laid out, the layout is preserved by saving to and loading from GML. Visual attributes specified in a GML file will result in a new visual style named `Filename.style` when that GML file is loaded.

# XGMML Format

XGMML is the XML evolution of GML and is based on the GML definition. In addition to network data, XGMML contains node/edge/network attributes. The XGMML file format specification is available at:

http://cgi5.cs.rpi.edu/research/groups/pb/punin/public_html/XGMML/

XGMML is now preferred to GML because it offers the flexibility associated with all XML document types. If you're unsure about which to use, choose XGMML.

There is a java system property "cytoscape.xgmml.repair.bare.ampersands" that can be set to "true" if you have experience trouble reading older files.

This should only be used when an XGMML file or session cannot be read due improperly encoded ampersands, as it slows down the reading process, but this is still preferable to attempting to fix such files using manual editing.

# SBML (Systems Biology Markup Language) Format

The Systems Biology Markup Language (SBML) is an XML format to describe biochemical networks. SBML file format specification is available at:

http://sbml.org/documents/

# BioPAX (Biological PAthways eXchange) Format

BioPAX is an OWL (Web Ontology Language) document designed to exchange biological pathways data. The complete set of documents for this format is available at:

http://www.biopax.org/

# PSI-MI Format

The PSI-MI format is a data exchange format for protein-protein interactions. It is an XML format used to describe PPI and associated data. PSI-MI XML format specification is available at:

http://psidev.sourceforge.net/mi/xml/doc/user/

# GraphML

GraphML is a comprehensive and easy-to-use file format for graphs. It is based on XML. The complete set of documents for this format is available at:

http://graphml.graphdrawing.org/

# Delimited Text Table and Excel Workbook

Cytoscape has native support for Microsoft Excel files (.xls, .xlsx) and delimited text files. The tables in these files can have network data and edge attributes. Users can specify columns containing source nodes, target nodes, interaction types, and edge attributes during file import. Some of the other network analysis tools, such as igraph (http://cneurocvs.rmki.kfki.hu/igraph/), has feature to export graph as simple text files. Cytoscape can read these text files and build networks from them. For more detail, please read the Import Free-Format Tables section of the Creating Networks chapter.

# Node and Edge Data

Interaction networks are useful as stand-alone models. However, they are most powerful for answering scientific questions when integrated with additional information.

Cytoscape allows the user to add arbitrary node, edge and network data to Cytoscape through the data table. This could include, for example, expression data of a gene or confidence values in a protein-protein interaction. In the data table, information is linked to nodes, edges or networks by mapping the columns to one of their identifiers. Through the table panel the values can be further manipulated through the use of attribute functions and equations

A second type of data that can be associated with networks is ontology data: organized sets of controlled vocabulary terms. Because this type of data is mostly hierarchically organized, this requires a special importing facility, described in the second part of this section.

Data associated with the network elements can be visualized in a user-defined way by setting up a mapping from data attributes in the columns to visual attributes (colors, shapes, and so on) in the network on screen. The section on visual styles discusses this in greater detail.

# Table data

Cytoscape supports several tabular formats. Of course the regular text and table formats: *.tsv, .tab, .csv, .txt* (comma, tab or any delimiter separated values file), *.xls, .xlsx* (Microsoft Excel file format). Note that for Excel file formats, only the first sheet of a workbook is currently imported. The legacy native Cytoscape formats: *.attrs* and *.pvals* (Cytoscape expression matrix) are also supported (for a more thorough discussion of these formats consult the 2.x manual). For most users the regular data table importing functionality will be sufficient; in case a format is unknown, renaming the file to a *.txt* extension and toying around with delimiter and header settings will most of the times do the trick.

## Importing data

### Basic table file

The basic file format consists of a table containing at least one column with identifiers (unique names) for the nodes, edges or networks, and one or (any number) more columns with data you want to associate with these network elements.

***Sample Data Table 1***

| Yeast Key | Degree |
|-----------|--------|
| YER054C   | 85     |
| YBL079W   | 7      |
| YLR345W   | 1      |

To import such a file:

1. Select File → Import → Table → File... ( or URL... if your source data file is accessible through the internet)

2. Select a data file in the file chooser panel (or enter the URL in the displayed box). This file can be in any of the accepted formats mentioned above.

3. In the "Import Column from Table" panel, select one of the attribute types. Cytoscape can import node, edge, and network attributes.

4. (Optional) Choose if you would like to import the file for all of the available networks or only selected networks using the check box in the expandable "Network Options" panel (this panel is collapsed by default). Select networks from the list.

5. (Optional) If the table is not properly delimited in the preview panel, change the delimiter in the Text File Import Options panel (the default delimiter is Tab).

6. By default, the first column is designated as the primary key. Change the key column if necessary, see below for an example.

7. The data attribute column name can be changed by right clicking on the header. A dialog appears that enables specification of the datatype and column name.

8. Left clicking selects or deselects the column for importing.

9. Click the Import button.

The user interface of the "Import Column from Table" window is similar to that of the "Import Network from Table" window.
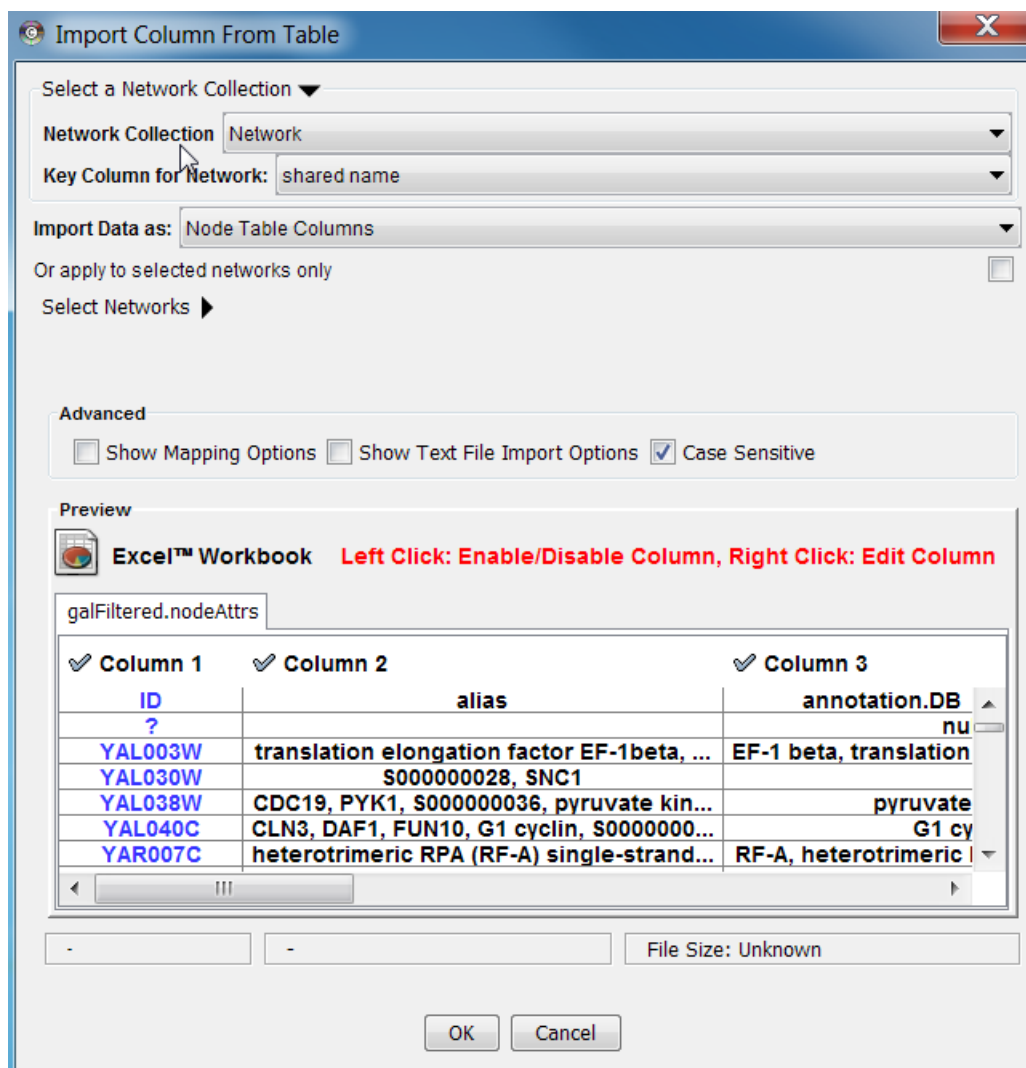


If the data only relates to a specific network select the 'Apply to selected networks only' box and select the specific network from the list (click 'Select Networks' to show this list)
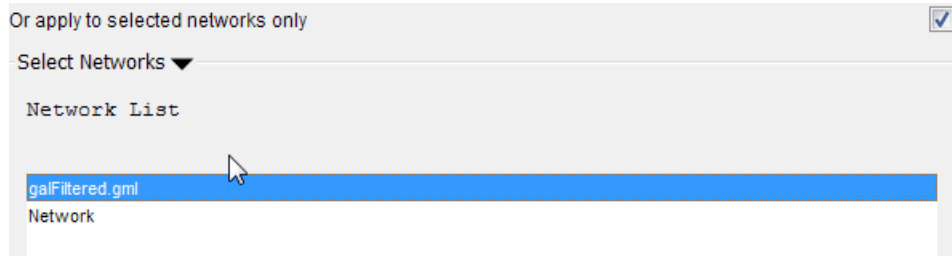
## Mapping data options

In Cytoscape 3.0 data columns with primitive data types (string, boolean, floating point, and integer) can be selected as the key column using the dropdown list provided. Complex data attributes such as lists are not supported as keys.
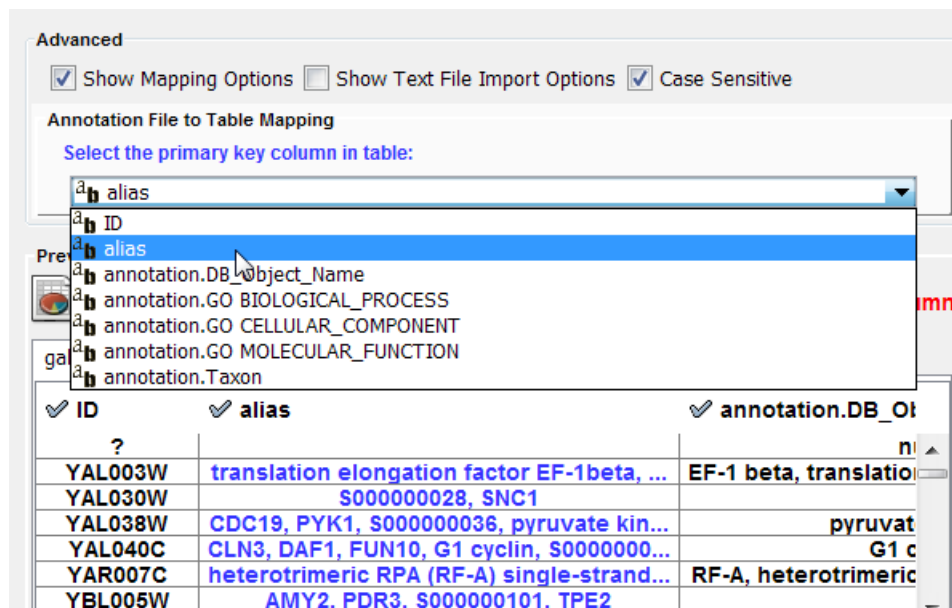
## Text file import options

When the text file import options box is checked, several additional parameters can be selected to tune the way the data file will be imported. The first line can be used to specify column header names. Another data delimiter (default is tab) can be chosen, and the comment prefix (signifying lines to be ignored) can be defined.

## Import from Public Databases

You can import various kinds of ID sets from the BioMart (http://www.biomart.org/index.html) public database. The BioMart web service client is integrated into Cytoscape.
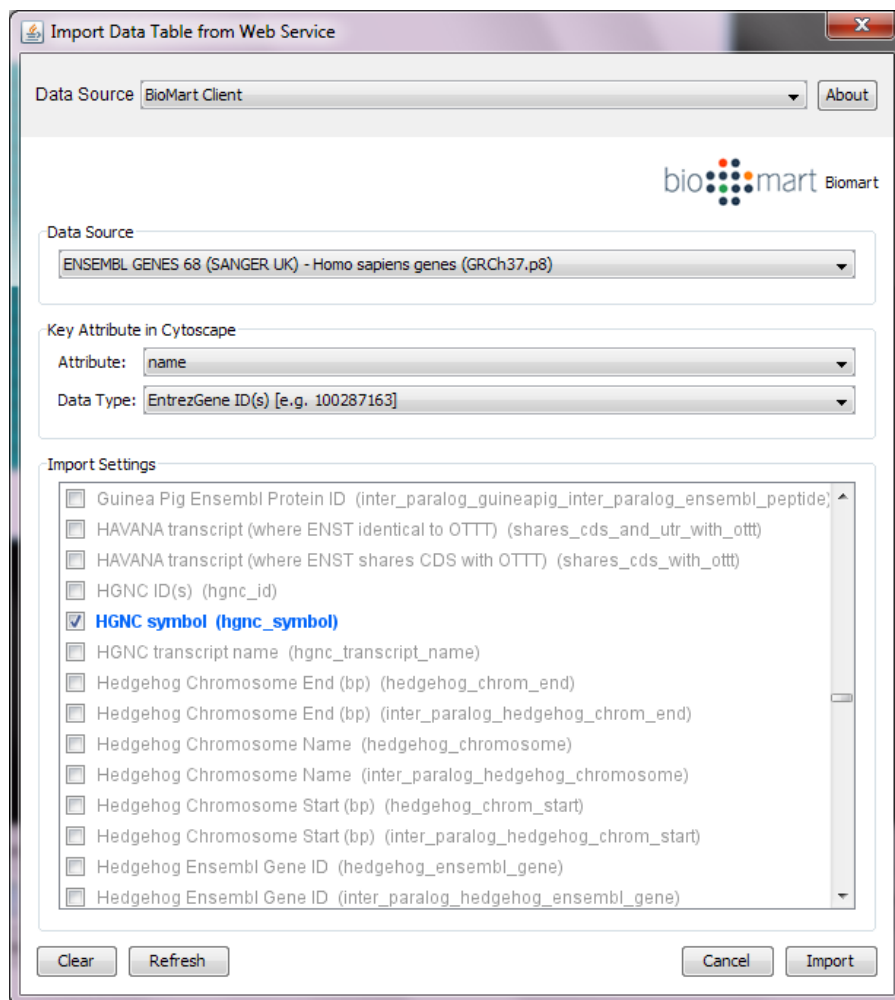
1. Select: **File → Import → Table → Public Databases...**

2. Select a data source. For ID mapping, select one of the *Ensemble Genes* data set. You need to choose the correct species for your network.

3. Select **Attribute**. If you want to import new ID sets matching current node IDs, select *shared name*.

4. Select **Data Type**. This should be the type of ID set selected in **Attribute** list. For example, if you select *shared name* for **Attribute** and your network uses *Entrez Gene ID* for its node ID, you need to select *EntrezGene ID(s)* for **Data Type**.

5. Select new ID sets from the list. Because there is a size limitation on the results returned by the!BioMart server, you can select only 3-5 attributes for each import.

6. Press **Import**.

## Data in complete networks

When importing a network from a table, columns other than the node identifier can be imported as data also. For more detail on these options, please see the "Import Free-Format Table Files" section of the user manual in the Creating Networks chapter.

# Table Panel



When Cytoscape is started, the Table Panel appears in the bottom CytoPanel. This panel can be hidden and restored using the View → Show/Hide Table Panel menu option. Like other CytoPanels, the panel can be undocked by pressing the little icon in the panel's top right corner.

To swap between displaying node, edge, and network tables use the tabs on the bottom of the panel labeled "Node Table", "Edge Table", and "Network Table".

In Cytoscape 3.0 there are two display modes for the table: show selected nodes/edges only and show all rows. This configuration can be set using button (the left most) in the figure. The Table Panel displays data belonging to the currently selected network.

Using the three buttons (left 2nd to 4th) in the figure, it is possible to make some or all columns visible and hide others or all of them. Also, a new column can be created by pressing button the (left 5th) or mutable columns can be deleted by button (left 6th). Button f(x) is for writing equations to manipulate the data which is further explained in the section attribute functions and equations.

Most data values can be edited by double-clicking on their table cell (only the SUID cannot be edited). Newline characters can be inserted into String attributes either by pressing *Enter* or by typing "\n". Once finished editing, click outside of the editing cell in the Table Panel or press *Shift-Enter* to save your edits. Pressing *Esc* while editing will undo any changes.

Table rows in the browser can be sorted by a specific column by clicking on a column heading. A new column can be created using the Create New Column button (left 5th), and must be one of four types – integer, string, real number (floating point), or boolean. Attributes can be deleted using the Delete Attributes button (left 6th, trash can icon). **NOTE: Deleting attributes removes them from Cytoscape, not just the attribute browser!** To remove attributes from the browser without deleting them, simply unselect the attribute using the Select Column button (left 3rd).

The right-click menu on the Table Panel has several functions, such as exporting attribute information to spreadsheet applications. For example, use the right-click menu to select the data and copy for use in a spreadsheet application.

# Ontologies

Another type of data that can be associated with network elements are ontologies. An ontology consists of an organized set of controlled vocabulary terms that annotate the objects. Most ontologies in science

are organized in a hierarchical way. In biology for example, using the Gene Ontology, the Saccharomyces Cerevisiae **CDC55** gene has a biological process described as "protein biosynthesis", to which GO has assigned the number 6412 (a GO ID).

```
GO 8150 biological_process
 GO 7582 physiological processes
   GO 8152 metabolism
    GO 44238 primary metabolism
      GO 19538 protein metabolism
        GO 6412 protein biosynthesis
```

**Graphical View of GO Term 6412: protein biosynthesis**
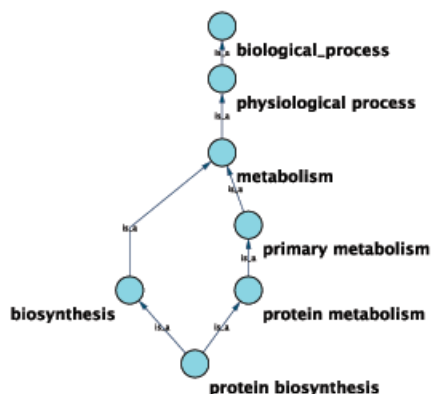


Cytoscape can use this ontology DAG (Directed Acyclic Graph) to annotate objects in networks. The Ontology Server is a Cytoscape feature which allows you to load, navigate, and associate ontology terms to nodes and edges in a network. Cytoscape has an GUI for loading ontology and associating it with the network elements, enabling you to load both local and remote files.

# Ontology and Association File Format

The standard file formats used in Cytoscape Ontology Server are OBO and Gene Association. The GO website details these file formats:

- Ontologies and Definitions: http://www.geneontology.org/GO.downloads.shtml#ont

- Current Associations: http://www.geneontology.org/GO.current.annotations.shtml

## OBO File

An OBO file is the ontology DAG itself. This file defines the relationships between ontology terms. Cytoscape can load all ontology files written in OBO format. The full listing of ontology files are available from the Open Biomedical Ontologies (OBO) website:

- OBO Ontology Browser: http://obo.sourceforge.net/browse.html

Sample OBO File - gene_ontology.obo: http://www.geneontology.org/ontology/gene_ontology_edit.obo

```
format-version: 1.2
date: 27:11:2006 17:12
saved-by: midori
auto-generated-by: OBO-Edit 1.002
```

```
subsetdef: goslim_generic "Generic GO slim"
subsetdef: goslim_goa "GOA and proteome slim"
subsetdef: goslim_plant "Plant GO slim"
subsetdef: goslim_yeast "Yeast GO slim"
subsetdef: gosubset_prok "Prokaryotic GO subset"
default-namespace: gene_ontology
remark: cvs version: $Revision: 5.49 $

[Term]
id: GO:0000001
name: mitochondrion inheritance
namespace: biological_process
def: "The distribution of mitochondria, including the mitochondrial genome, into d
synonym: "mitochondrial inheritance" EXACT []
is_a: GO:0048308 ! organelle inheritance
is_a: GO:0048311 ! mitochondrion distribution

[Term]
id: GO:0000002
name: mitochondrial genome maintenance
namespace: biological_process
def: "The maintenance of the structure and integrity of the mitochondrial genome."
is_a: GO:0007005 ! mitochondrion organization and biogenesis
```

## Default List of Ontologies

Cytoscape provides a list of ontologies available in OBO format. If an Internet connection is available, Cytoscape will import ontology and associatation files directly from the remote source. The table below lists the included ontologies.

| Ontology Name | Description |
|---|---|
| Gene Ontology Full | This data source contains a full-size GO DAG, which contains all GO terms. This OBO file is written in version 1.2 format. |
| Generic GO slim | A subset of general GO Terms, including higer-level terms only. |
| Yeast GO slim | A subset of GO Terms for annotating Yeast data sets maintained by SGD. |
| Molecule role (INOH Protein name/family name ontology) | A structured controlled vocabulary of concrete and abstract (generic) protein names. This ontology is a INOH pathway annotation ontology, one of a set of ontologies intended to be used in pathway data annotation to ease data integration. This ontology is used to annotate protein names, protein family names, and generic/concrete protein names in the INOH pathway data. INOH is part of the BioPAX working group. |
| Event (INOH pathway ontology) | A structured controlled vocabulary of pathway-centric biological processes. This ontology is a INOH pathway annotation ontology, one of a set of ontologies intended to be used in pathway data annotation to ease data integration. This ontology is used to annotate biological processes, pathways, and sub-pathways in the INOH pathway data. INOH is part of the BioPAX working group. |
| Protein-protein interaction | A structured controlled vocabulary for the annotation of experiments concerned with protein-protein interactions. |
| Pathway Ontology | The Pathway Ontology is a controlled vocabulary for pathways that provides standard terms for the annotation of gene products. |

| PATO | PATO is an ontology of phenotypic qualities, intended for use in a number of applications, primarily phenotype annotation. For more information, please visit the PATO wiki (http://www.bioontology.org/wiki/index.php/PATO:Main_Page). |
|---|---|
| Mouse pathology | The Mouse Pathology Ontology (MPATH) is an ontology for mutant mouse pathology. This is Version 1. |
| Human disease | This ontology is a comprehensive hierarchical controlled vocabulary for human disease representation. For more information, please visit the Disease Ontology website (http://diseaseontology.sourceforge.net/). |

Although Cytoscape can import all kinds of ontologies in OBO format, association files are associated with specific ontologies. Therefore, you need to provide the correct ontology-specific association file to annotate nodes/edges/networks in Cytoscape. For example, while you can annotate human network data using the GO Full ontology with human Gene Association files, you cannot use a combination of the human Disease Ontology file and human Gene Association files, because the Gene Association file is only compatible with GO.

## Visualize and Browse Ontology DAG (for Advanced Users)

Relationships between ontology terms are usually represented as Directed Acyclic Graphs (DAGs). This is a special case of a network (or graph), where nodes are ontology terms and edges are relationships between terms. Ontology data is stored in the same data structure as normal networks. This enables users and App writers to visualize, browse and manipulate ontology DAGs just like other networks. The following is an example of visualization of an ontology DAG (Generic GO Slim):
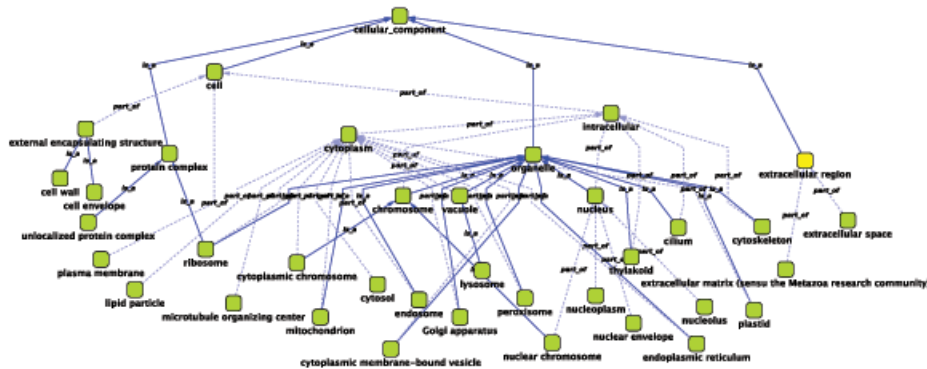


Every ontology term and relationship can have attributes. In the OBO files, ontology terms have optional fields such as definition, synonyms, comments, or cross-references. These fields will be imported as node attributes. To browse those attributes, please use the attribute browser (see the example below):

- Note 1: Some ontologies have a lot of terms. For example, the full Gene Ontology contains more than 20,000 terms. If you need to save memory, you can remove this ontology DAG from Network Panel (right-click on the ontology name at the left-hand side of the screen and select Destroy Network).

- Note 2: All ontology DAGs will be saved in the session file. To minimize the session file size, you can delete the Ontology DAG before saving session.

## Gene Association File

The Gene Association (GA) file provides annotation only for the Gene Ontology. It is a species-specific association file for GO terms. **Gene Association files will only work with Gene Ontologies and NOT others!**

Sample Gene Association File (gene_association.sgd - association file for yeast):

```
SGD     S000003916     AAD10     GO:0006081     SGD_REF:S000042151|PMID:10
SGD     S000005275     AAD14     GO:0008372     SGD_REF:S000069584     ND
```

If you have a network file and an association file, they should have a common key to map attributes onto network data. If those two do not have a common key, you need to do an extra step to add a shared key by mapping the current key to a common key as described above (Node Name Mapping).

# Import Gene Ontology and Gene Association Files

For convenience, Cytoscape has a list of URLs for commonly used ontology data and a complete set of Gene Association files. To import Gene Ontology and Gene Association files for the loaded networks, please follow these steps:

*Important: All data sources in the preset list are remote URLs, meaning a network connection is required!*

## Step 1. Select an Annotation File
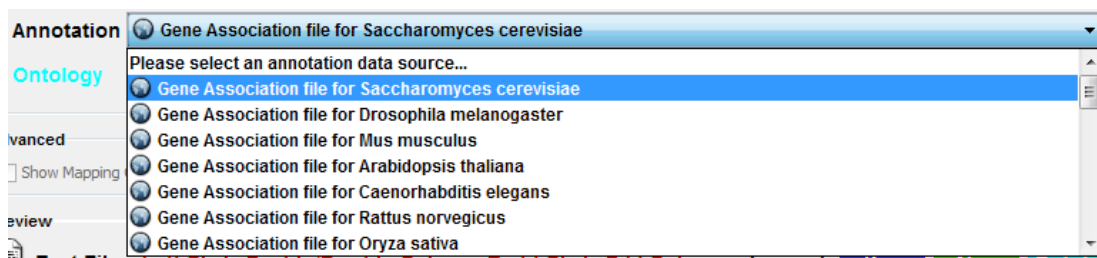
Select File → Import → Ontology and Annotation... to open the "Import Ontology and Annotation" window. From the Annotation dropdown list, select a gene association file for your network. For example, if you want to annotate the yeast network, select "Gene Association file for Saccharomyces cerevisiae".

## Step 2. Select an Ontology File
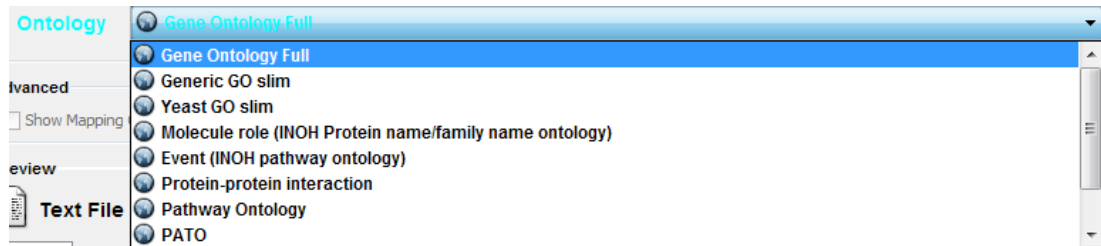


Select an Ontology data (OBO file) from the Ontology dropdown list. If the file is not loaded yet, it will be shown in red. The first three files are Gene Ontology files. You can load other ontologies, but you need your own annotation file to annotate networks.

## Step 3. Import the files

Once you click the Import button, Cytoscape will start loading OBO and Gene Association files from the remote sources. If you choose GO Full it may take a while since it is a large data file.

# Attribute Functions and Equations

## Attribute Formulas

## Introduction

Attribute values may be formulas. A typical example is **=ABS($otherAttrib + LOG(10.2))**. Formulas are modelled after Excel™ but only support references to other attributes at the same node, edge or network. Since Cytoscape attribute names may contain embedded spaces, optional braces around the attribute name (required if the name is not simply a letter followed by one or more letters or digits) is allowed e.g. **${a name with spaces}**. Backslashes, opening braces and dollar signs in attribute names have to be escaped with a leading backslash. For example the attribute name **ex$am{p\le** would have to be written as **${ex\$am\{p\\le}**. Finally, attribute names are case sensitive.

String constants are written with double-quotes **"**. In order to embed a double-quote or a backslash in a string they have to be escaped with a leading backslash, therefore the string **"\** must be written as **"\"\\"**. Formula results must be compatible with the type of the attribute that they have been assigned to. The rules are rather lax though, for example anything can be interpreted as a string and all numeric values will be accepted for a boolean (or logical) attribute where non-zero will be interpreted as **true** and zero as **false**. For integer attributes floating point values will be converted using the rules of the Excel™ **INT** function. Parentheses can be used for grouping and to change evaluation order. The operator precedence rules follow those of standard arithmetic.

## Operators

Currently supported operators are the four basic arithmetic operators and the **^** exponentiation operator. **+**, **-**, **\***, and **\** are left-associative and **^** is right-associative. The string concatenation operator is **&**. Supported boolean or logical operators are the comparison operators **<**, **>**, **<=**, **>=**, **=**, and **<>** (not equal).

# Supported Functions

Currently we support the following functions:

## *Cytoscape-specific functions*

- *Degree -- the degree of a node.*

- *InDegree -- the indegree of a node.*

- *OutDegree -- the outdegree of a node.*

- *SourceID -- the ID of the source node of an edge.*

- *TargetID -- the ID of the target of an edge.*

## *Numeric Functions*

- *Abs -- Returns the absolute value of a number.*

- *ACos -- Returns the arccosine of a number.*

- *ASin -- Returns the arcsine of a number.*

- *ATan2 -- Returns the arctangent of two numbers x and y.*

- *Average -- Returns the average of a group of numbers.*

- *Cos -- Returns the cosine of an angle given in radians.*

- *Cosh -- Returns the hyperbolic sine of its argument.*

- *Count -- Returns the number of numeric values in a list.*

- *Degrees -- Returns its argument converted from radians to degrees.*

- *Exp -- Returns e raised to a specified number.*

- *Ln -- Returns the natural logarithm of a number.*

- *Log -- Returns the logarithm of a number to a specified base.*

- *Max -- Returns the maximum of a group of numbers.*

- *Median -- Returns the median of a list of numbers.*

- *Min -- Returns the minimum of a group of numbers.*

- *Mod -- Calculates the modulus of a number.*

- *Pi -- Returns an approximation of the value of →*

- *Radians -- Returns its argument converted from degrees to radians.*

- *Round -- Rounds a number to a specified number of decimal places.*

- *Sin -- Returns the sine of an angle given in radians.*

- *Sinh -- Returns the hyperbolic sine of its argument.*

- *Sqrt -- Calculates the square root of a number.*

- *Tan -- returns the tangent of its argument in radians.*

- *Tanh -- returns the hyperbolic tangent of its argument in radians.*

- *Trunc -- Truncates a number.*

## String Functions

- *Concatenate -- Concatenates two or more pieces of text.*

- *Left -- Returns a prefix of s string.*

- *Len -- Returns the length of a string.*

- *Lower -- Converts a string to lowercase.*

- *Mid -- Selects a substring of some text.*

- *Right -- Returns a suffix of a string.*

- *Substitute -- Replaces some text with other text.*

- *Text -- Format a number using the Java DecimalFormat class' conventions.*

- *Upper -- Converts a string to uppercase.*

- *Value -- Converts a string to a number.*

## Logical/Boolean Functions

- *And -- Returns the logical conjunction of any number of boolean values.*

- *Not -- Returns the logical negation of a boolean value.*

- *Or -- Returns the logical disjunction of any number of boolean values.*

## List Functions

- *First -- Returns the first entry in a list.*

- *Last -- Returns the last entry in a list.*

- *Nth -- Returns the n-th entry in a list.*

## Statistical Functions

- *Largest -- the kth largest value in a list.*

- *GeoMean -- the geometric mean of a set of numbers.*

- *HarMean -- the harmonic mean of a set of numbers.*

- *Mode -- the mode of a set of numbers.*

- *NormDist -- Returns the pdf or CDF of the normal distribution.*

- *Permut -- Returns the number of permutations for a given number of objects.*

- *StDev - sample standard deviation.*

- *Var -- sample variance.*

### Miscellaneous Functions

- *Combin - Returns the number of combinations for a given number of objects.*

- *If -- Returns one of two alternatives based on a boolean value.*

- *ListToString -- Returns a string representation of a list.*

- *Now -- Returns a string representation of the current date and time.*

- *Today -- returns a string representation of the current date.*

# Pitfalls

The possibly biggest problem is the referencing of other attributes that have null values. This is not allowed and leads to errors. In order to mitigate this problem we support the following optional syntax for attribute references: **${attribName:defaultValue}**. The interpretation is that if **attribName** is null, then the default value will be used, otherwise the value of the referenced value will be used instead. The referenced attribute must still be a defined attribute and not an arbitrary name! The other potential problem is when there are circular attribute reference dependencies. Circular dependencies will be detected at formula evaluation time and lead to a runtime error.

# Useful Tips

When working with formulas it can be very helpful to open the Developer's Log Console. Formula evaluation errors will be logged there.

# The Formula Builder

In order to ease the creation of formulas as well as to facilitate discovery of built-in functions we provide a *formula builder* in our attribute browser. After selecting a non-list attribute cell, you can invoke it by c       l       i       c       k       i       n       g                                                   o       n

                                                                                           .

This should bring up the Formula Builder which looks like this:
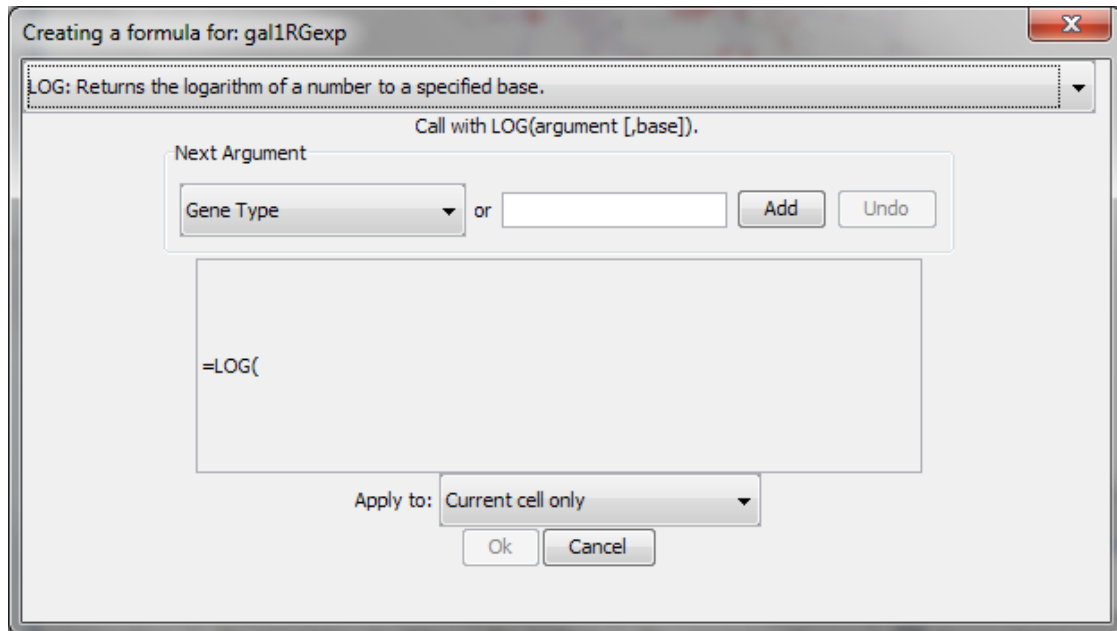
The Formula Builder is a useful tool for discovery of the list of built-in functions, which has the return type matching the data type of the column. Arguments can either be selected from a list of named attributes or constant values can be entered in a text entry field. A major shortcoming at this time is that the Formula Builder won't let you compose functions with function calls as arguments. If you need the most general functionality, please type the expression directly into a cell.

# A Note for App Writers

It is relatively easy to add your own built-in formula functions. A simple function can probably be implemented in 15 to 20 minutes. It can then be registered via the parser and becomes immediately available to the user. It will of course also show up in the drop-down list in the formula builder.

# Navigation and Layout

## Basic Network Navigation

Cytoscape uses a Zoomable User Interface for navigating and viewing networks. ZUIs use two mechanisms for navigation: zooming and panning. Zooming increases or decreases the magnification of a view based on how much or how little a user wants to see. Panning allows users to move the focus of a screen to different parts of a view.

## Zoom

Cytoscape provides two mechanisms for zooming: either using mouse gestures or buttons on the toolbar. Use the zooming buttons located on the toolbar to zoom in and out of the interaction network shown in the current network display. Zoom icons are detailed below:



From Left to Right:

- Zoom In

- Zoom Out

- Zoom Out to Display all of Current Network

- Zoom Selected Region

You can also zoom in/out by holding down the right mouse button and moving the mouse to the right (zoom in) or left (zoom out).

# Pan

There are several ways to pan the network:

- Middle Click and Drag - You can pan the network image by holding down the middle mouse button and moving the mouse.

- Command Key + Drag (Mac only) - If you use Mac without middle button, you can pan the view by holding down Command key and then drag.

- Dragging Box on Network Overview - You can also pan the image by holding down the left mouse button over the blue box in the overview panel in the lower part of the Network tab in the Control Panel.

# Other Mouse Behaviors

## Select

Click the left mouse button on a node or edge to select that object. You can hold down the Shift key to select more than one node/edge or you can hold down the left mouse button and drag the mouse to select groups of nodes/edges.

## Context

Click the right mouse button on a node/edge to launch a context-sensitive menu with additional information about the node/edge.

## Node Context Menu

This menu can change based on the current context. For nodes, it typically shows:

- Edit

- Select

- Group

- Nested Networks

- Apps

- External Links

- Preferences

Edges usually have the following menu:

- Edit

- External Links

Apps can contribute their own items into node and edge context menus. These additions usually appear in the *Apps* menu.

### Nested Network Node Context Submenu

!!! menu items changed names

- Add Nested Network

   Lets the user select any network in Cytoscape as the current node's nested network. If the current node already has a nested network it will be replaced.

- Remove Nested Network

   After being selected the current node will no longer have a nested network associated with it. The associated network is not deleted. Only the association between the node and the network is removed.

- Go to Nested Network

   After selecting this the network view of the current node's nested network will be the current network view and have the focus. Should a network view for the nested network not exist, it will be created.

# Automatic Layout Algorithms

The Layout menu has an array of features for organizing the network visually according to one of several algorithms, aligning and rotating groups of nodes, and adjusting the size of the network. Cytoscape layouts have three different sources, which are reflected in the *Layout* menu.

## yFiles Layouts

**yFiles** layouts are a set of commercial layouts which are provided courtesy of yWorks. Due to license restrictions, the detailed parameters for these layouts are **not** available (there are no **yFiles** entries in the **Settings...** dialog). The main layout algorithms provided by yFiles are:

## yFiles Circular Layout
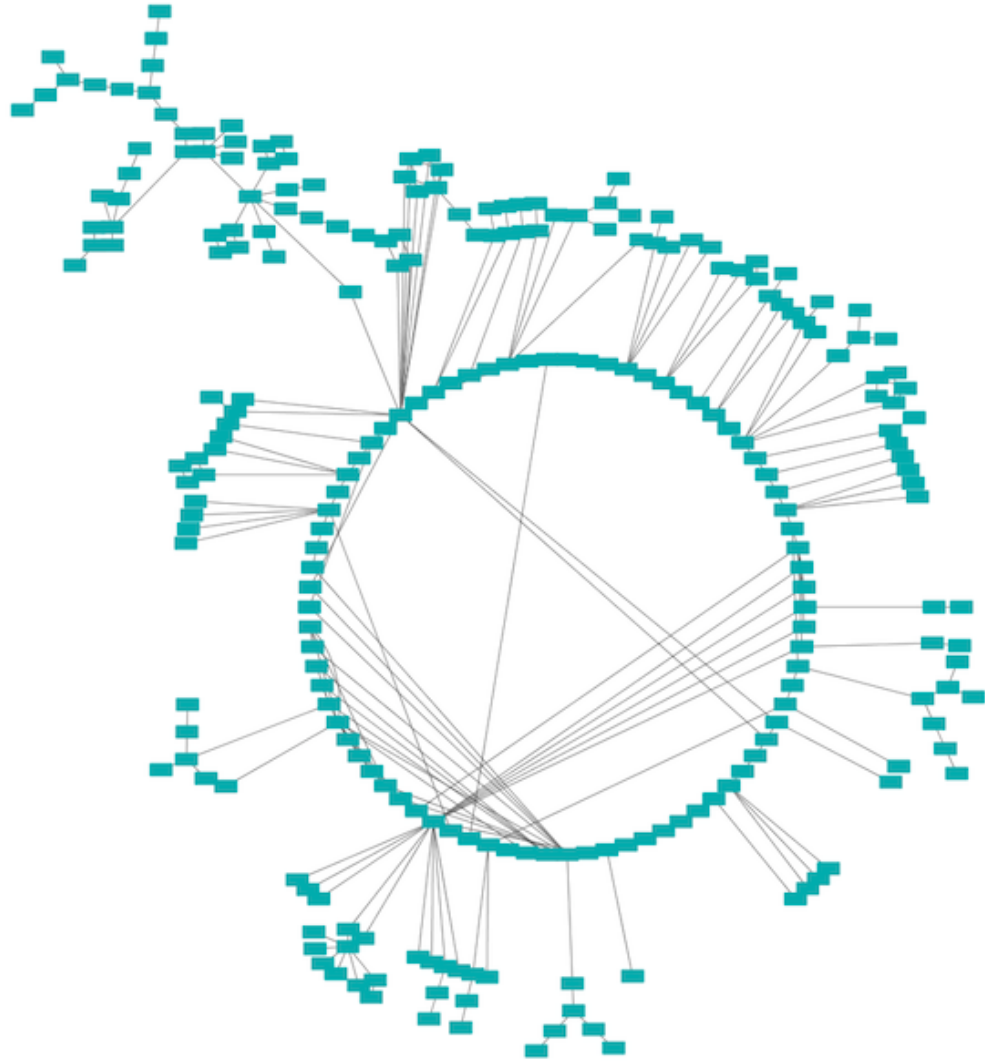


This algorithm produces layouts that emphasize group and tree structures within a network. It partitions the network by analyzing its connectivity structure, and arranges the partitions as separate circles. The circles themselves are arranged in a radial tree layout fashion. This algorithm is available by selecting **Layout → yFiles Layouts → Circular**.

## yFiles Hierarchical Layout
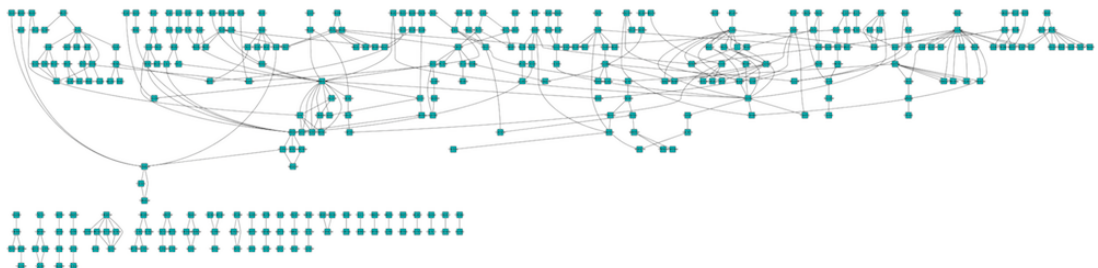
The hierarchical layout algorithm is good for representing main direction or "flow" within a network. Nodes are placed in hierarchically arranged layers and the ordering of the nodes within each layer is chosen in such a way that minimizes the number of edge crossings. This algorithm is available by selecting **Layout → yFiles Layouts → Hierarchical**.

# yFiles Organic Layout
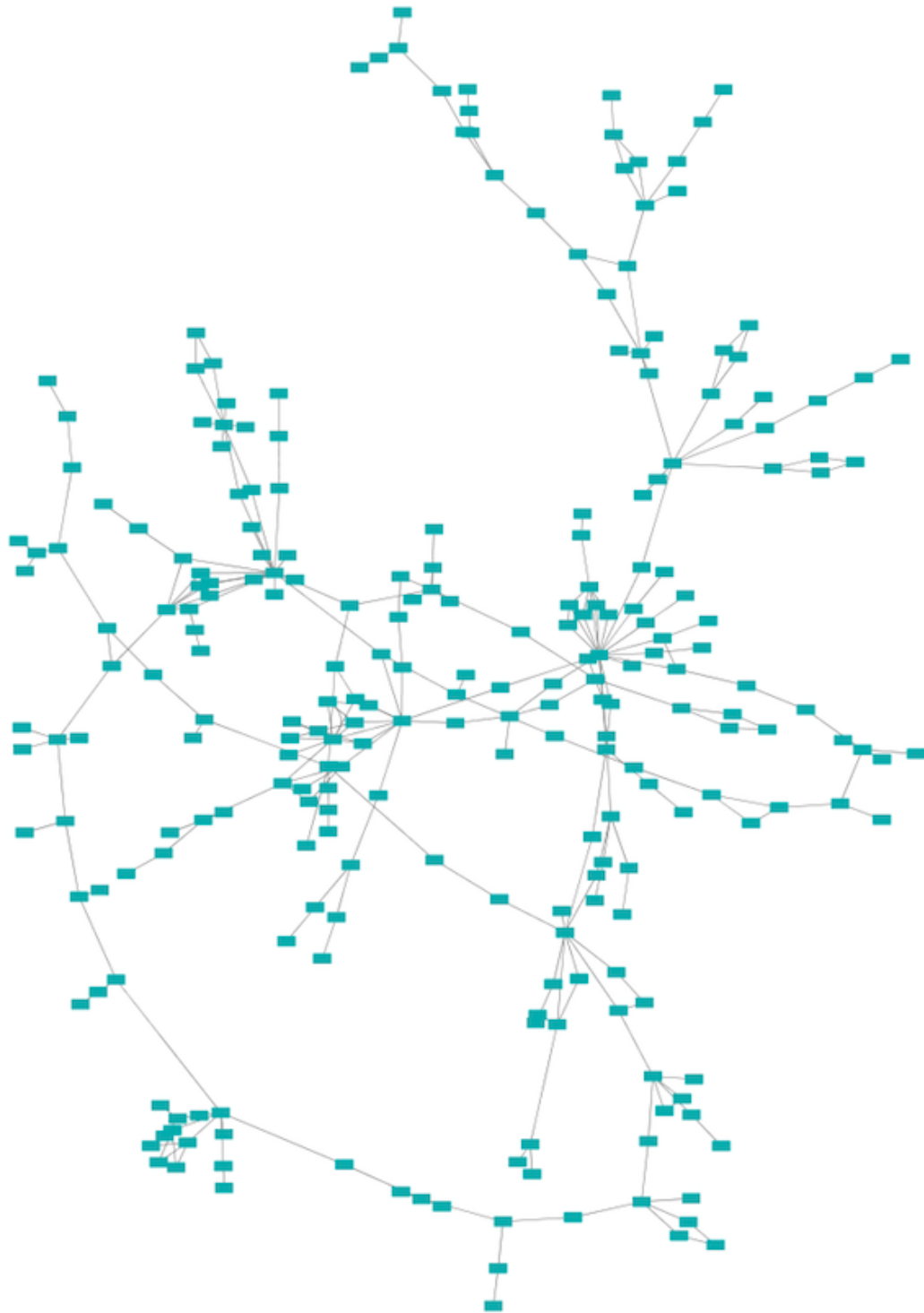
The organic layout algorithm is a kind of spring-embedded algorithm that combines elements of the other algorithms to show the clustered structure of a graph. This algorithm is available by selecting **Layout → yFiles Layouts → Organic**.

# Cytoscape Layouts

**Cytoscape Layouts** are those layouts that have been written or integrated by Cytoscape developers. These layouts are fully integrated with Cytoscape. All Cytoscape Layouts have the option to operate on only the selected nodes, and all provide a **Settings...** panel to change the parameters of the algorithm. Most of the Cytoscape layouts also partition the graph before performing the layout. In addition, many of these layouts include the option to take either node or edge attributes into account. Some of these layouts are:

## Grid Layout
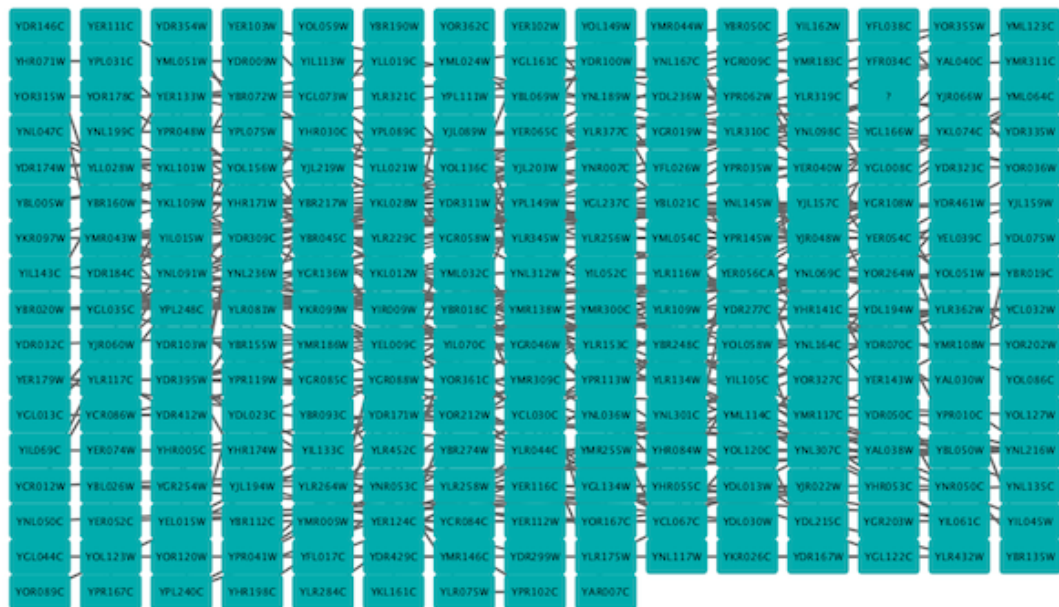


The grid layout is a simple layout the arranges all of the nodes in a square grid. This is the default layout and is always available as part of the Cytoscape core. It is available by selecting **Layout → Grid Layout**. A sample screen shot is shown above.

## Spring-Embedded Layout
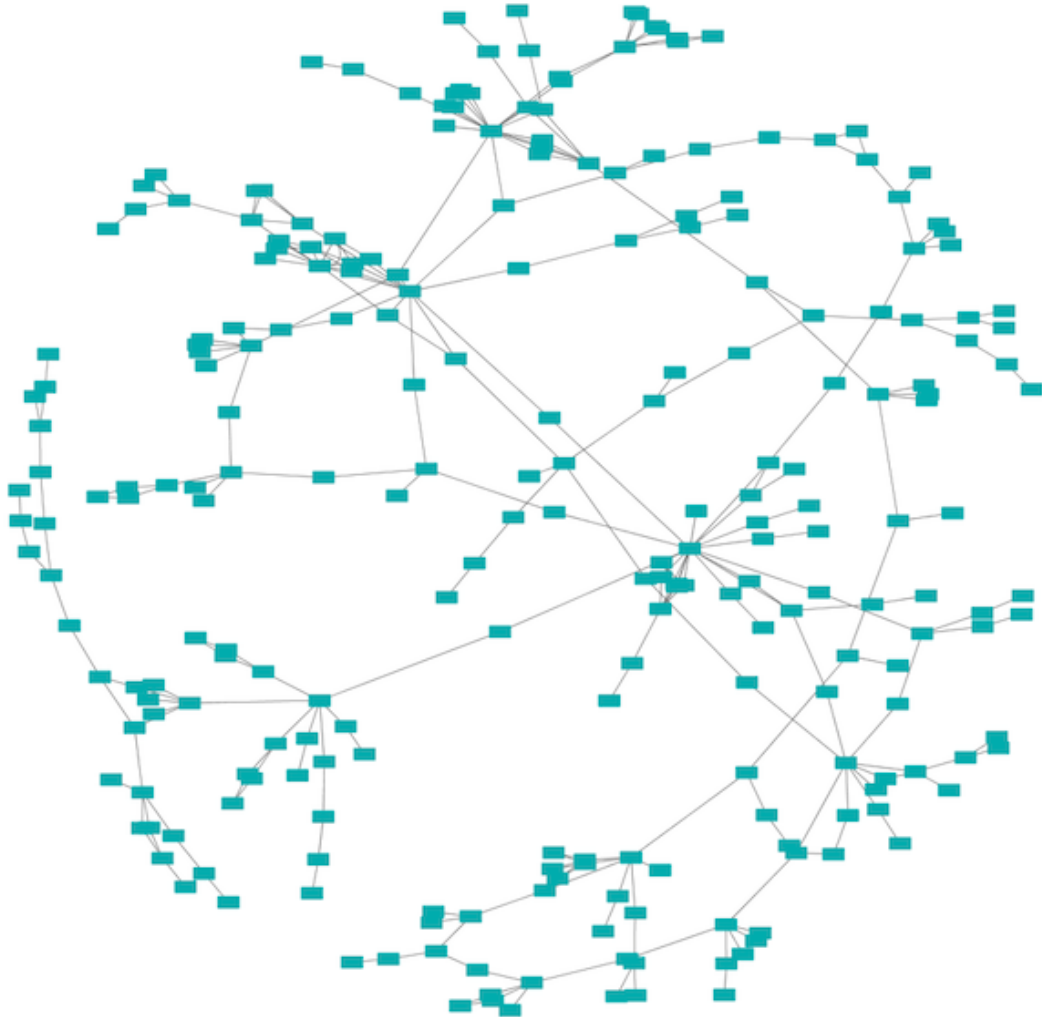
The spring-embedded layout is based on a "force-directed" paradigm as implemented by Kamada and Kawai (1988). Network nodes are treated like physical objects that repel each other, such as electrons. The connections between nodes are treated like metal springs attached to the pair of nodes. These springs repel or attract their end points according to a force function. The layout algorithm sets the positions of the nodes in a way that minimizes the sum of forces in the network. This algorithm can be applied to the entire network or a portion of it by selecting the appropriate options from **Layout → Edge-weighted Spring Embedded**.

## Attribute Circle Layout
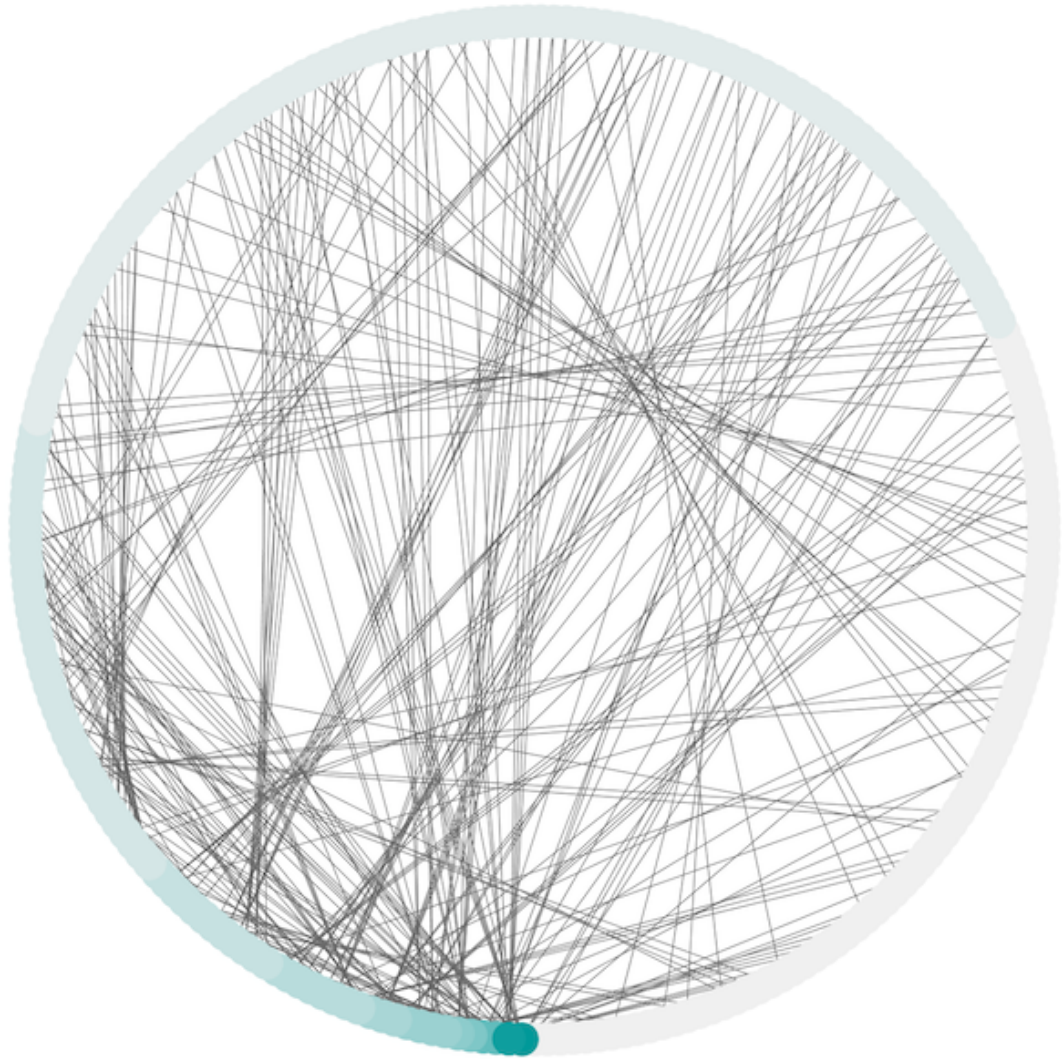


The attribute circle layout is a quick, useful layout, particularly for small networks, that will locate all of the nodes in the network around a circle. The node order is determined by a user-selected node attribute. The result is that all nodes with the same value for that attribute are located together around the circle. Using **Layout → Attribute Circle Layout →** *attribute* to put all nodes around a circle using *attribute* to position them. The sample screen shot above shows the a subset of the galFiltered network organized by node degree.

## Group Attributes Layout
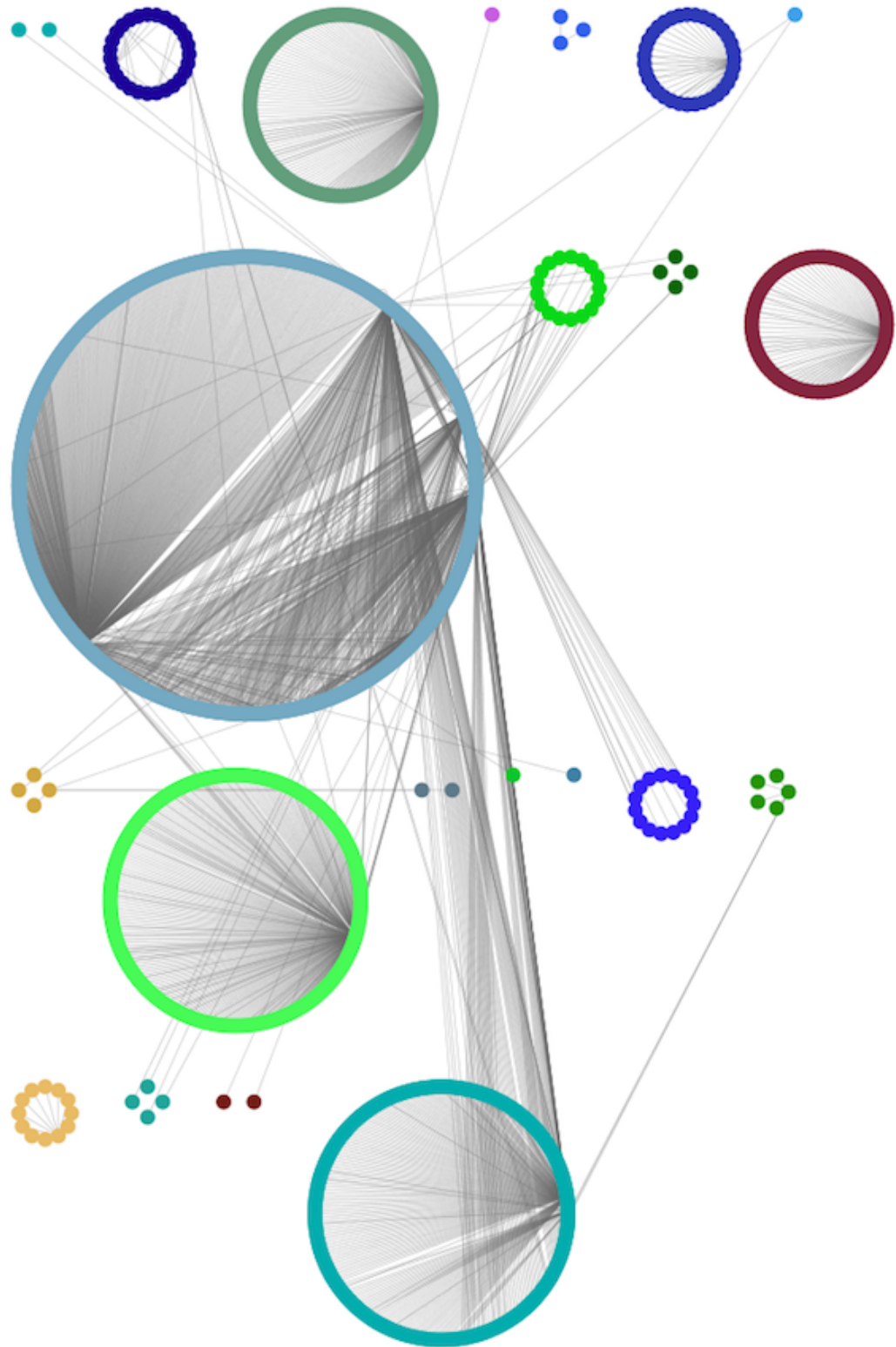
The group attributes layout is similar to the attribute circle layout described above except that instead of a single circle with all of the nodes, each set of nodes that share the same value for the attribute are laid out in a separate circle. The same network shown above (network generated by PSICQUIC Client) is shown above, using **Layout → Group Attributes Layout → taxonomy**.

# Prefuse Layouts

## Force Directed Layout
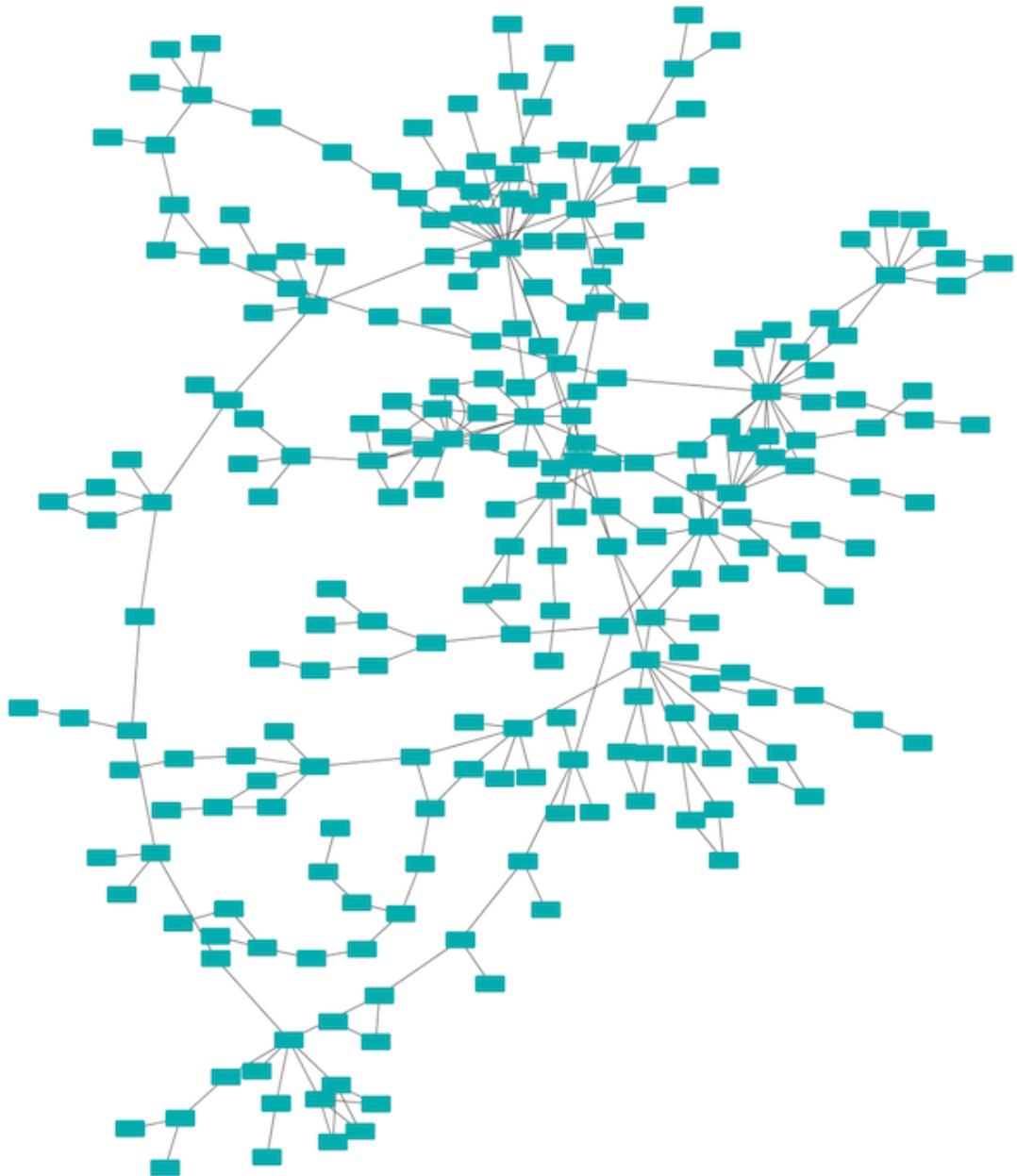


The force-directed layout is a layout based on the "force-directed" paradigm. This layout is based on the algorithm implemented as part of the excellent prefuse toolkit provided by Jeff Heer. The algorithm is very fast and with the right parameters can provide a very pleasing layout. The **Force Directed Layout** will also accept a numeric edge attribute to use as a weight for the length of the spring, although this will often

require more use of the **Settings...** dialog to achieve the best layout. This algorithm is available by selecting **Layout → Force-Directed Layout → (unweighted)** or the edge attribute you want to use as a weight. A sample screen shot showing a portion of the galFiltered network provided in sample data is provided above.

## Layout Parameters

Many layouts have adjustable parameters that are exposed through the Layouts → Settings... menu option. This will pop up the following dialog, which allows you to choose which layout algorithm settings to adjust. The settings presented vary by algorithm and only those algorithms that allow access to their parameters will appear in the drop-down menu at the top of the dialog. Once you've modified a parameter, clicking the "Execute Layout" button will apply the layout. This be repeated until a useful layout is achieved.
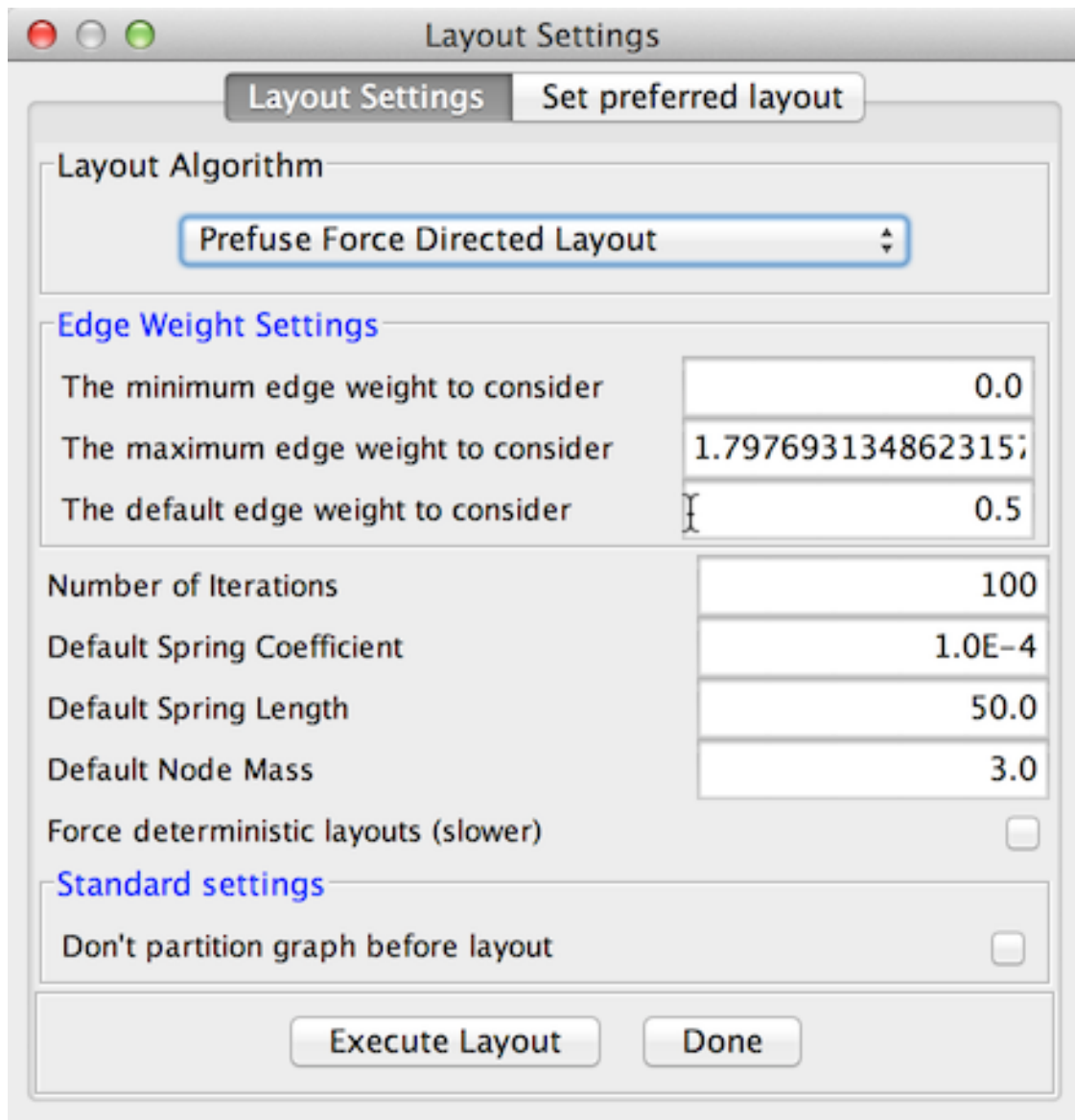
# Edge Bend and Automatic Edge Bundling
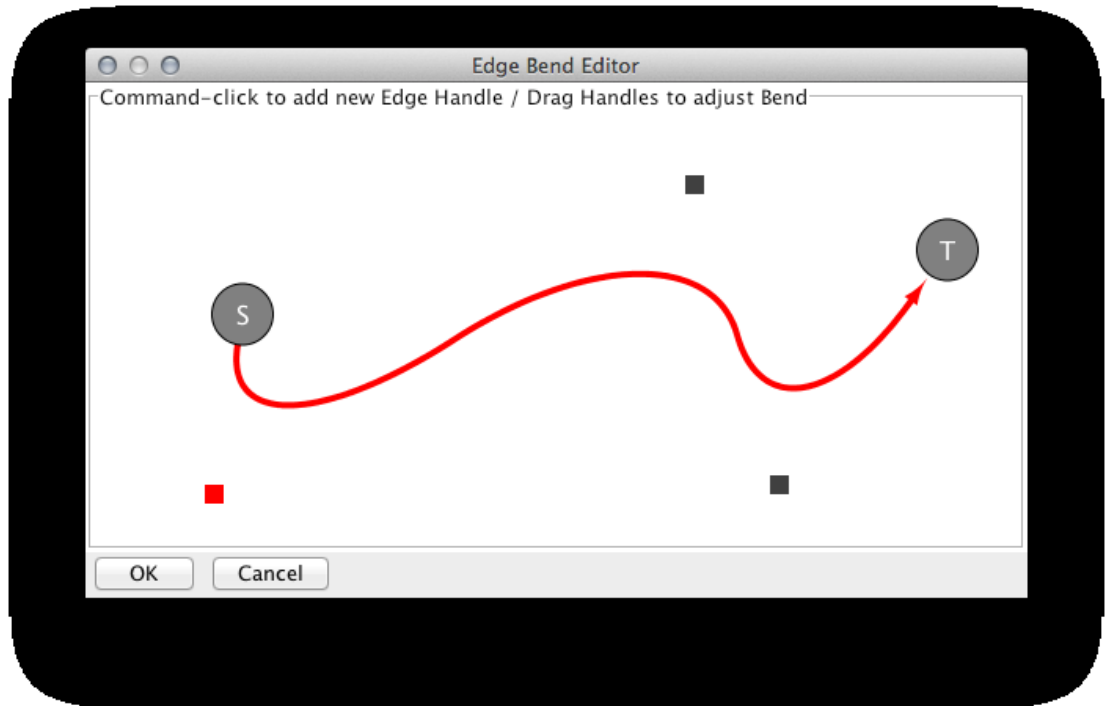


From Cytoscape 3.0, *Edge Bend* is a regular Visual Property and can be used as a part of Visual Style. In the Edge Bend editor, you can add as many handles as you want to the edge by ALT-Click (For Mac, COMMAND-click).
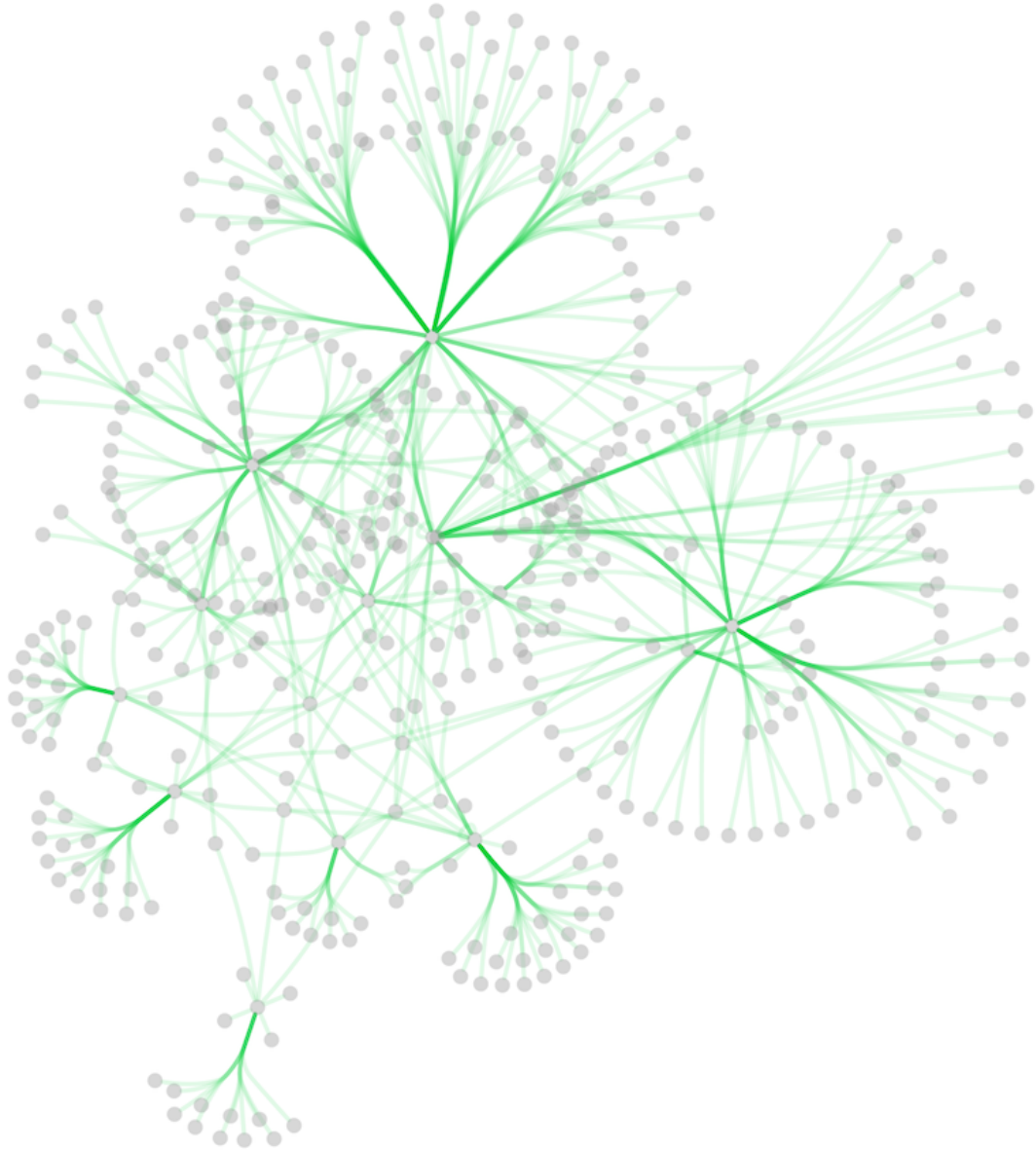
In addition to adding handles manually, you can use *Edge Bundle* function to bundle all or selected edges automatically.



1. Select **Layout --> Bundle Edges --> All Nodes and Edges**

2. Set parameters.

- Details of the algorithm is described in this paper.

3. Press OK to run. It may take a long time if the number of edges is large.

- If it takes too long, you can decrease *Maximum Iterations*.

- For large, dense networks, try Maximum Iterations = 500 - 1000.

Note: The handle locations will be optimized for current location of nodes. If you move node positions, you need to run the function again to get proper result.

# Manual Layout

The simplest method to manually organize a network is to click on a node and drag it. If you select multiple nodes, all of the selected nodes will be moved together.

## Rotate
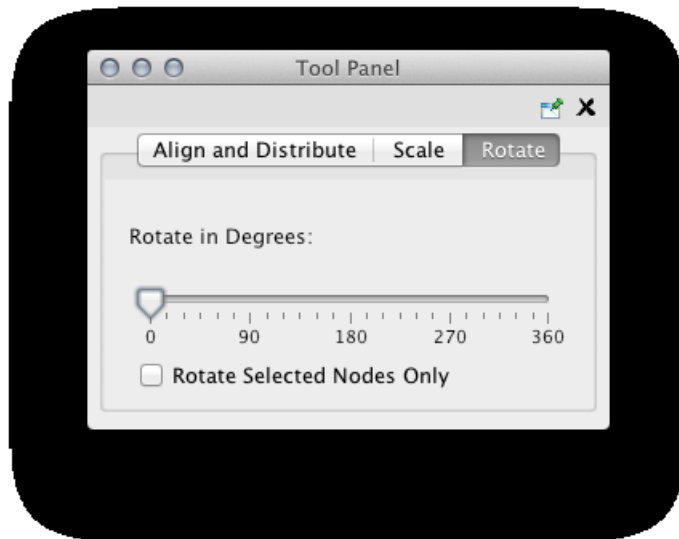


Selecting the Layout → Rotate option will show the Rotate window in the Tool Panel. This function will either rotate the entire network or a selected portion of the network. The image below shows a network with selected nodes rotated.

Before
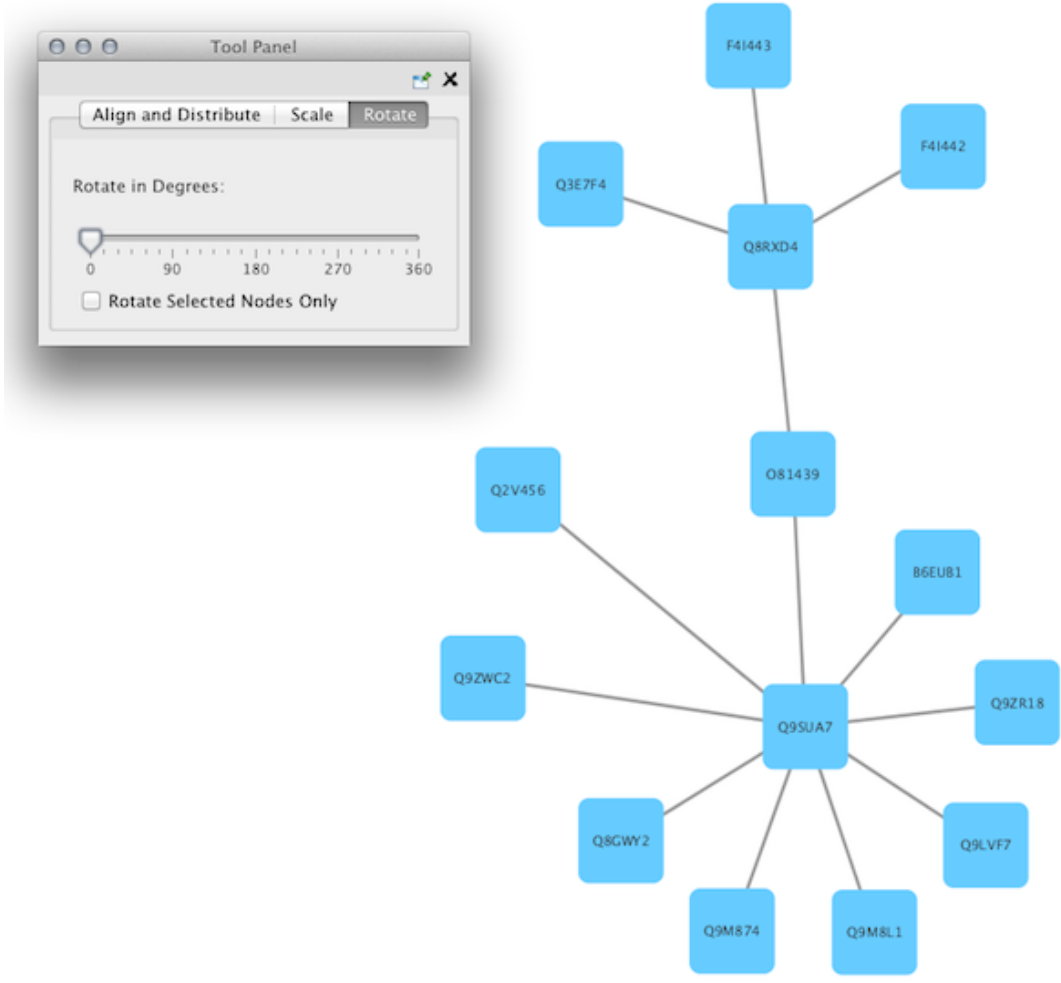
After
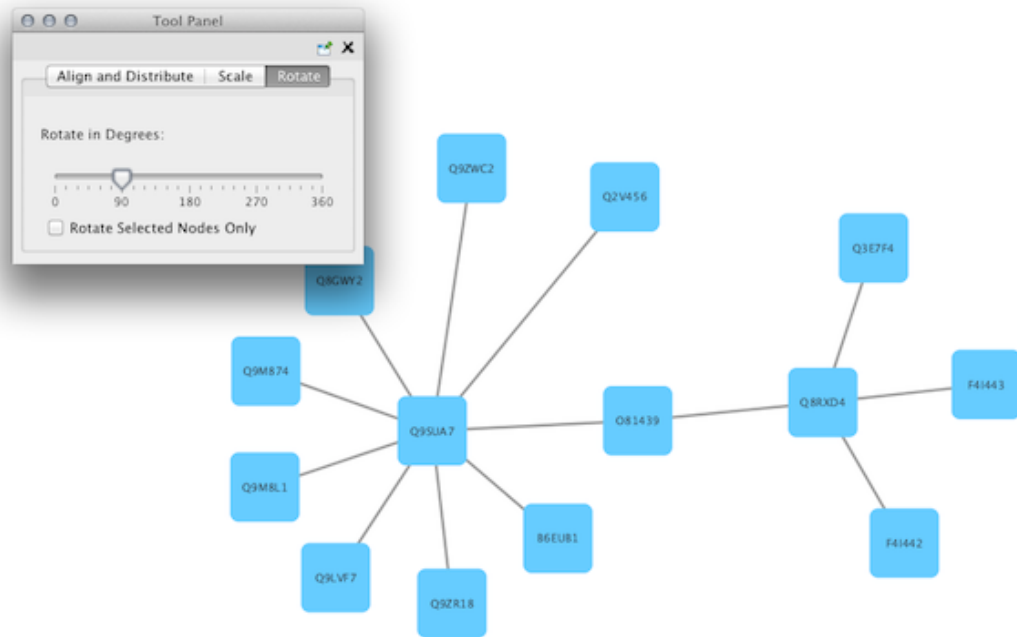
## Scale
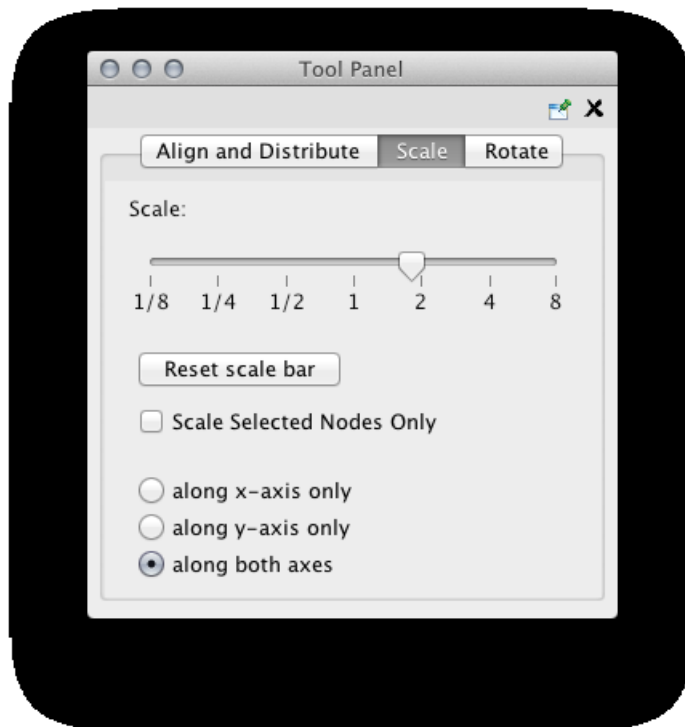


Selecting the Layout → Scale option will open the Scale window in the Tool Panel. This function will scale the position of the entire network or of the selected portion of the network. Note that only the position

of the nodes will be scaled, not the node sizes. Node size can be adjusted using the VizMapper. The image below shows selected nodes scaled.

Before
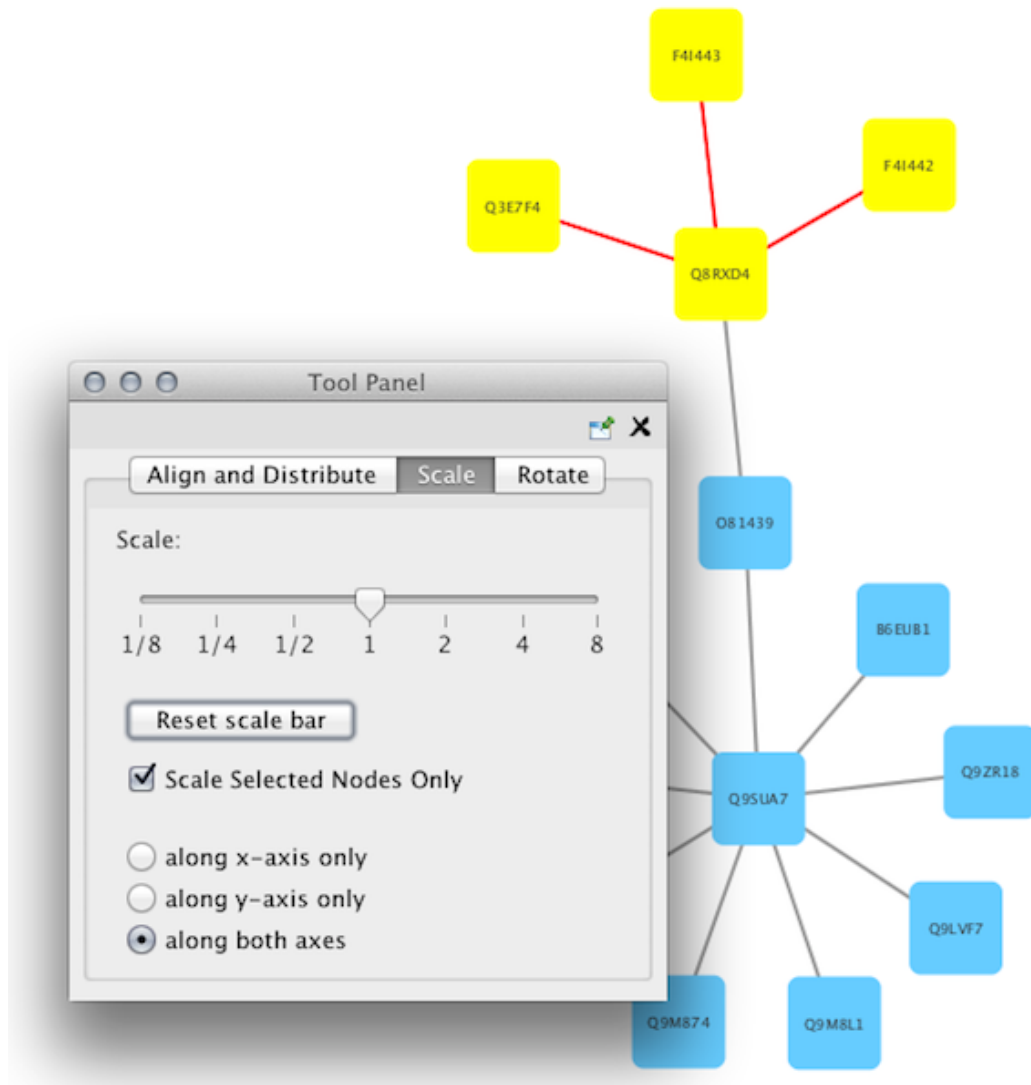


After
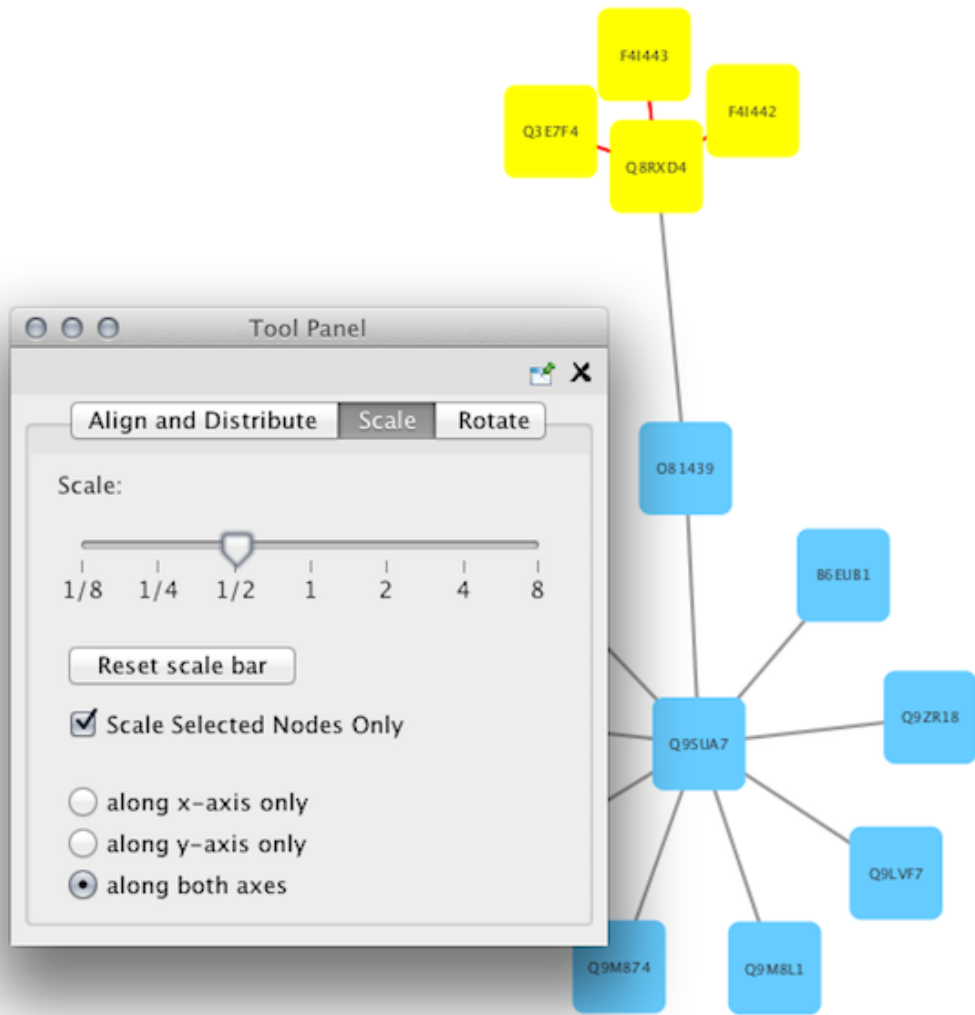
# Align/Distribute/Stack



Selecting the Layout → Align/Distribute option will open the Align/Distribute/Stack window in the Tool Panel. The Align buttons provide different options for either vertically or horizontally aligning selected nodes against a line. The differences are in what part of the node gets aligned, e.g. the center of the node, the top of the node, the left side of the node. The Distribute buttons evenly distribute selected nodes between the two most distant nodes along either the vertical or horizontal axis. The differences are again a function what part of the node is used as a reference point for the distribution. And the Stack buttons vertically or horizontally stack selected nodes with the full complement of alignment options. The table below provides a decription of what each button does.

| Button | Before | After | Description of Align Options |
|---|---|---|---|
| | | | Vertical Align Top - The tops of the selected nodes are aligned with the top-most node. |
| | | | Vertical Align Center - The centers of the selected nodes are aligned along a line defined by the midpoint between the top and bottom-most nodes. |
| | | | Vertical Align Bottom - The bottoms of the selected nodes are aligned with the bottom-most node. |
| | | | Horizontal Align Left - The left hand sides of the selected nodes are aligned with the left-most node. |
| | | | Horizontal Align Center - The centers of the selected nodes are aligned along a line defined by the midpoint between the left and right-most nodes. |
| | | | Horizontal Align Right - The right hand sides of the selected nodes are aligned with the right-most node. |

| Button | B e - fore | After | Description of Distribute Options |
|---|---|---|---|
| | | | |

| | | | |
|---|---|---|---|
| | | | Vertical Distribute Top - The tops of the selected nodes are distributed evenly between the top-most and bottom-most nodes, which should stay stationary. |
| | | | Vertical Distribute Center - The centers of the selected nodes are distributed evenly between the top-most and bottom-most nodes, which should stay stationary. |
| | | | Vertical Distribute Bottom - The bottoms of the selected nodes are distributed evenly between the top-most and bottom-most nodes, which should stay stationary. |
| | | | Horizontal Distribute Left - The left hand sides of the selected nodes are distributed evenly between the left-most and right-most nodes, which should stay stationary. |
| | | | Horizontal Distribute Center - The centers of the selected nodes are distributed evenly between the left-most and right-most nodes, which should stay stationary. |
| | | | Horizontal Distribute Right - The right hand sides of the selected nodes are distributed evenly between the left-most and right-most nodes, which should stay stationary. |

| Button | Before | After | Description of Stack Options |
|---|---|---|---|
| | | | Vertical Stack Left - Vertically stacked below top-most node with the left-hand sides of the selected nodes aligned. |
| | | | Vertical Stack Center - Vertically stacked below top-most node with the centers of selected nodes aligned. |
| | | | Vertical Stack Right - Vertically stacked below top-most node with the right-hand sides of the selected nodes aligned. |
| | | | Horizontal Stack Top - Horizontally stacked to the right of the left-most node with the tops of the selected nodes aligned. |
| | | | Horizontal Stack Center - Horizontally stacked to the right of the left-most node with the centers of selected nodes aligned. |
| | | | Horizontal Stack Bottom - Horizontal Stack Center - Horizontally stacked to the right of the left-most node with the bottoms of the selected nodes aligned. |

# Node Movement and Placement

In addition to the ability to click on a node and drag it to a new position, Cytoscape now has the ability to move nodes using the arrow keys on the keyboard. By selecting one or more nodes using the mouse and clicking one of the arrow keys ($\leftarrow$, $\rightarrow$, $\uparrow$, $\downarrow$) the selected nodes will move one pixel in the chosen direction. If an arrow key is pressed while holding the Shift key down, the selected nodes will 15 pixels in the chosen direction.

# Visual Styles

## What is a Visual Style?

One of Cytoscape's strengths in network visualization is the ability to allow users to encode any table data (name, type, degree, weight, expression data, etc.) as a Visual Property (such as color, size of node, transparency, or font type). A set of these encoded or mapped table data sets is called a *Visual Style* and can be created or edited using the Cytoscape *VizMapper*. With the VizMapper, the visual appearance of your network is easily customized. For example, you can:

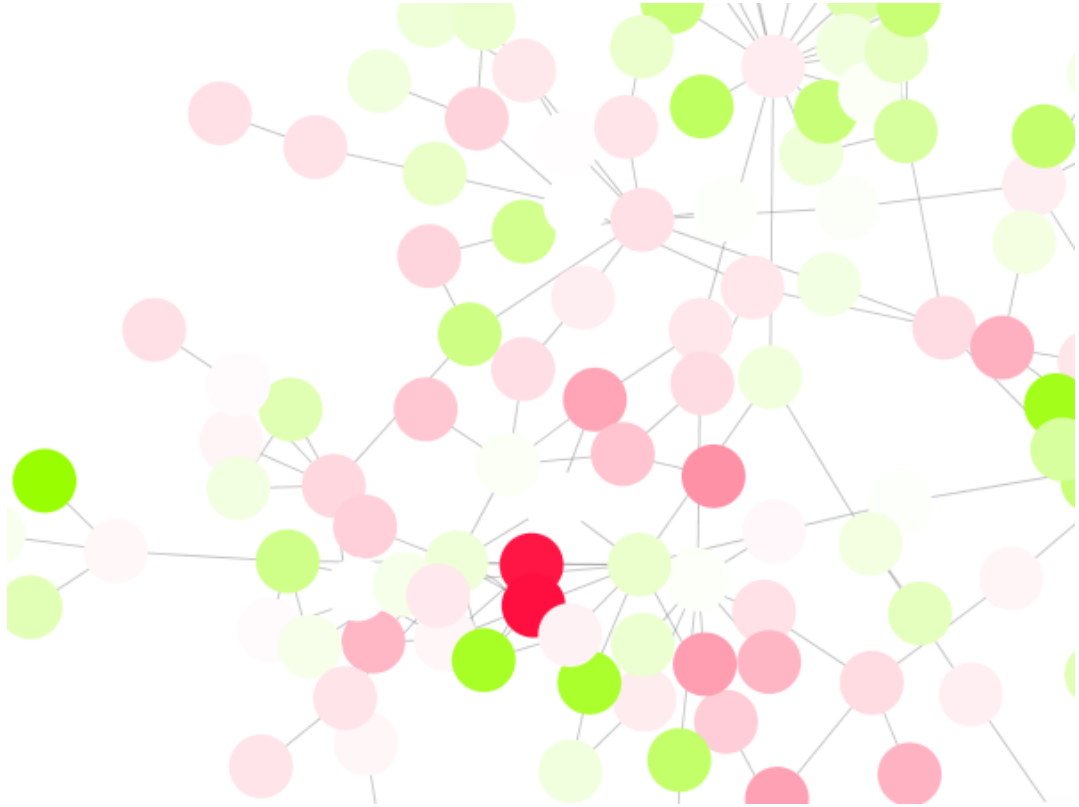• Specify a default color and shape for all nodes.
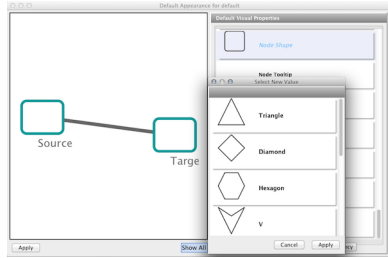
- Set node sizes based on the degree of connectivity of the nodes. You can visually see the hub of a network...
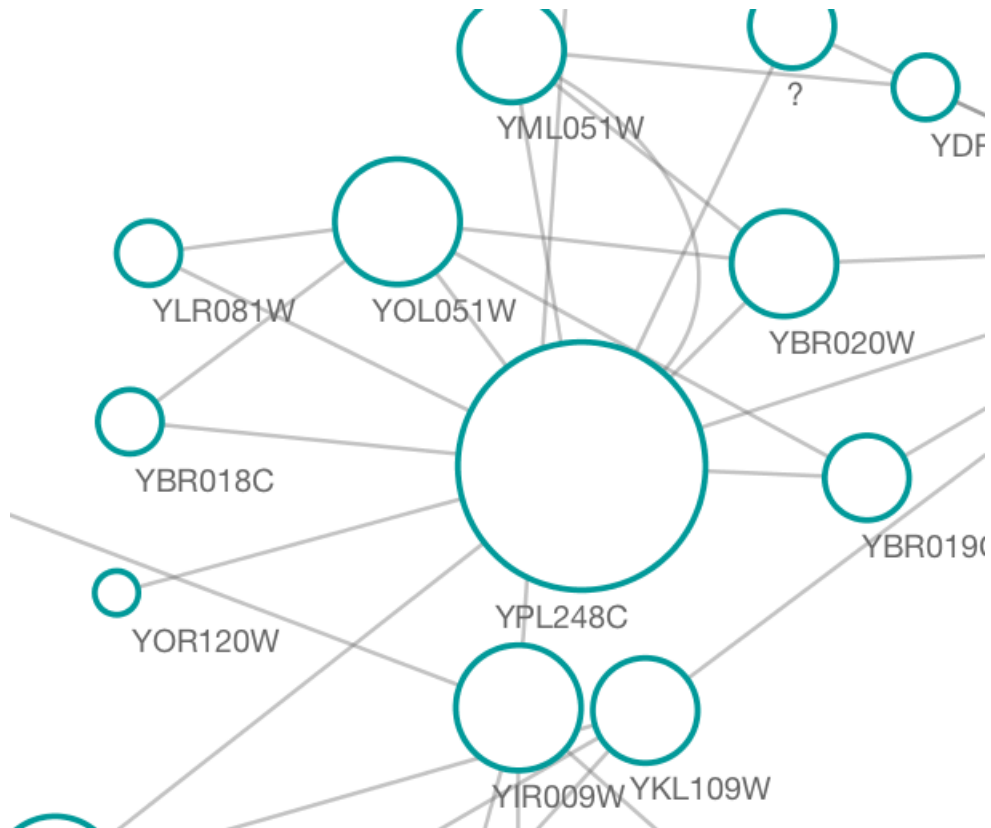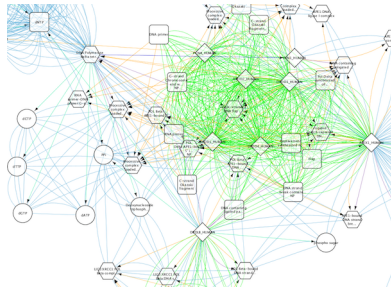


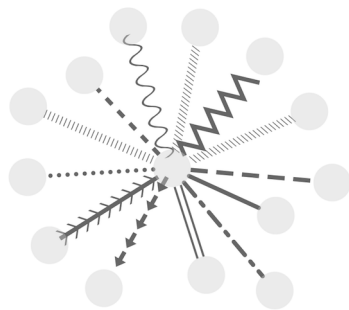- ...or, set the font size of the node labels instead.



- Visualize gene expression data along a color gradient.

- Encode specific physical entities as different node shapes.



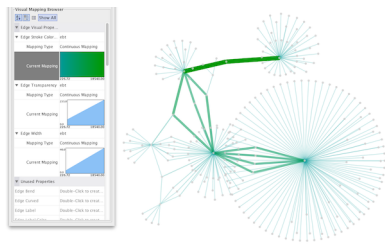- Use specific line types to indicate different types of interactions.


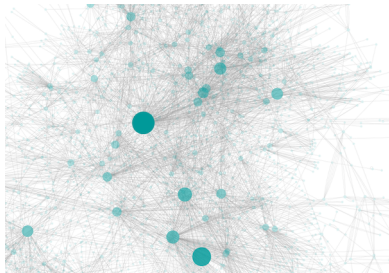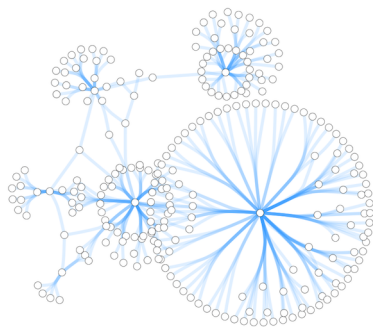
- Control edge transparency (opacity) using edge weights.
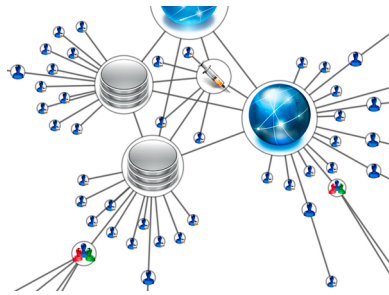
- Control multiple edge visual properties using edge score.



- Browse extremely-dense networks by controlling the opacity of nodes.



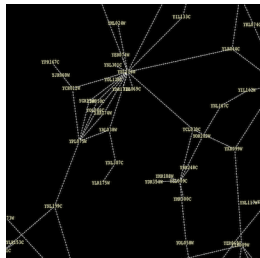- Show highly-connected region by edge bundling and opacity.



- Add photo/image/graphics on top of nodes.
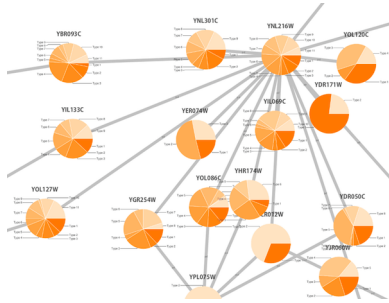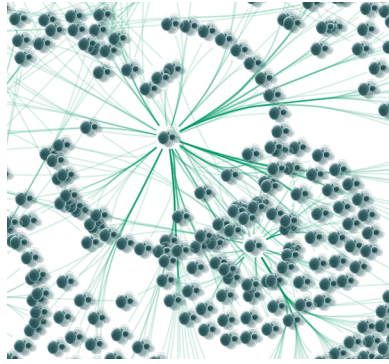
Cytoscape 3 has several sample Visual Styles. You can try those to examine how Visual Styles change the appearence of a network. The following is a list of network views based upon sample styles applied to the *galFiltered.sif* network :
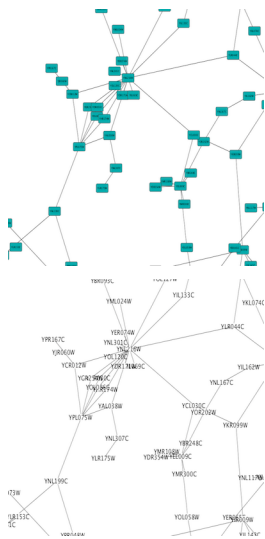
VizMapper is located in a tab on the Control Panel at the left-hand side of the screen.

# Introduction to the **VizMapper User Interface**

There are two main control panels in the VizMapper:

1. Main Panel

- This panel allows you to create/delete/view/switch between different visual styles using the Current Visual Style options. The Visual Mapping Browser at the bottom displays the mapping details for a given visual style and is used to edit these details as well.

2. Default Appearance Editor



- Clicking on the section labelled "Defaults" on the Main Panel will bring up the Default Appearance Editor, which allows you to visually edit the default appearance of nodes and edges for the selected visual style.

These editors will be discussed in further detail below.

# Introduction to Visual Styles

The Cytoscape distribution includes several predefined visual styles to get you started. To examine these styles, try out the following example:

**Step 1. Load some sample data**

- Load a sample session file: From the main menu, select *File → Open*, and select the file *sampleData/galFiltered.cys*.

- The session file includes a network, some annotations, and sample visual styles. By default, the **Sample for galFiltered** style is selected. Gene expression values for each node are colored along a color gradient between red and green (where red represents a low expression ratio and green represents a high expression ratio, using thresholds set for the gal4RGexp experiment bundled with Cytoscape in the *sampleData/galExpData.csv* file). Also, node size is mapped to the degree of the node (number of edges connected to the node) and you can see the hubs of the network as larger nodes. See the sample screenshot below:
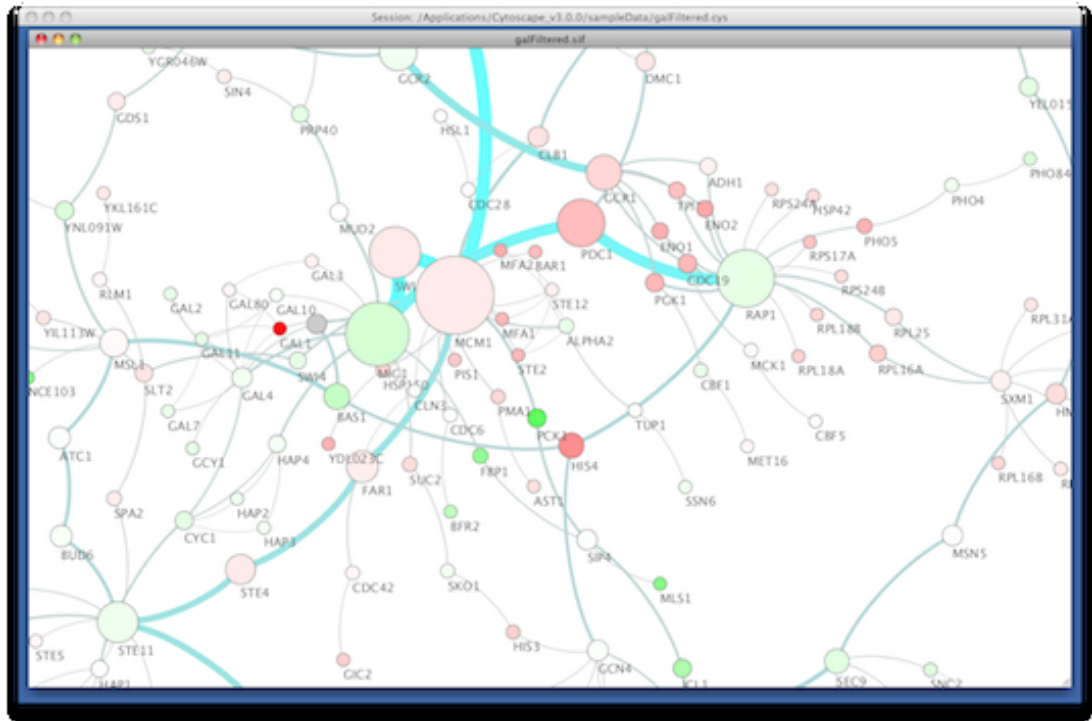
**Step 2. Switch between different Visual Styles**

You can change visual styles by making a selection from the Current Visual Style dropdown list (found at the top of the VizMapper Main Panel).

For example, if you select **Sample1**, a new visual style will be applied to your network, and you will see a white background and round blue nodes. If you zoom in closer, you can see that protein-DNA interactions (specified with the label "pd") are drawn with dashed edges, whereas protein-protein interactions (specified with the label "pp") are drawn with solid edges (see sample screenshot below).
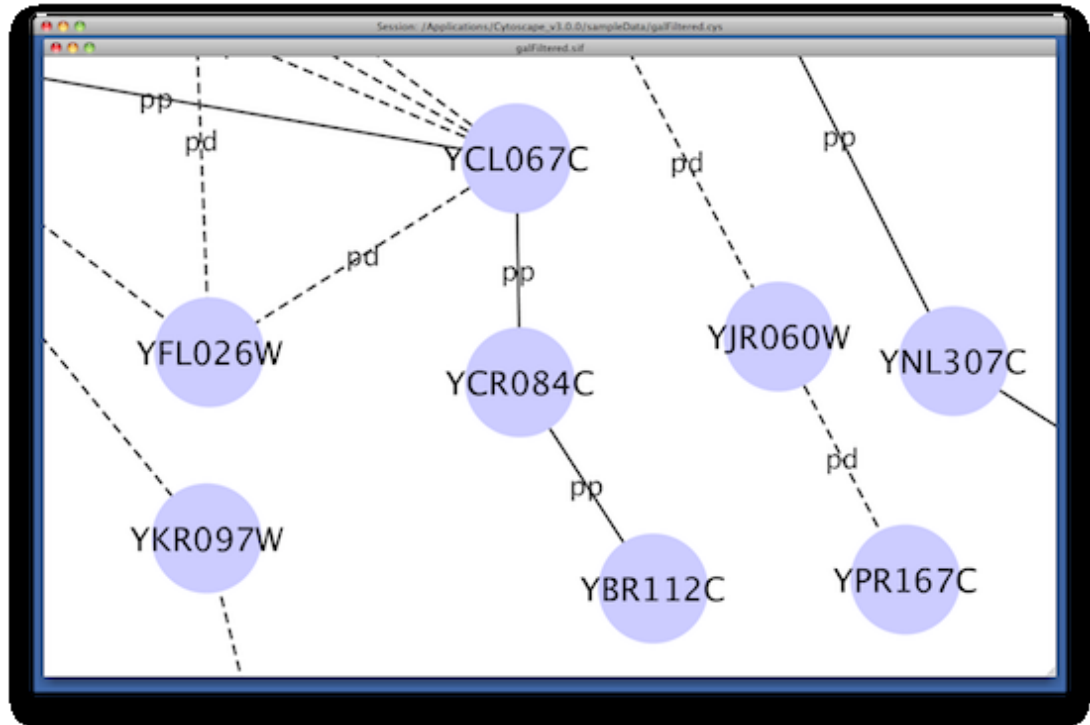
Finally, if you select **Solid**, you can see the graphics below:
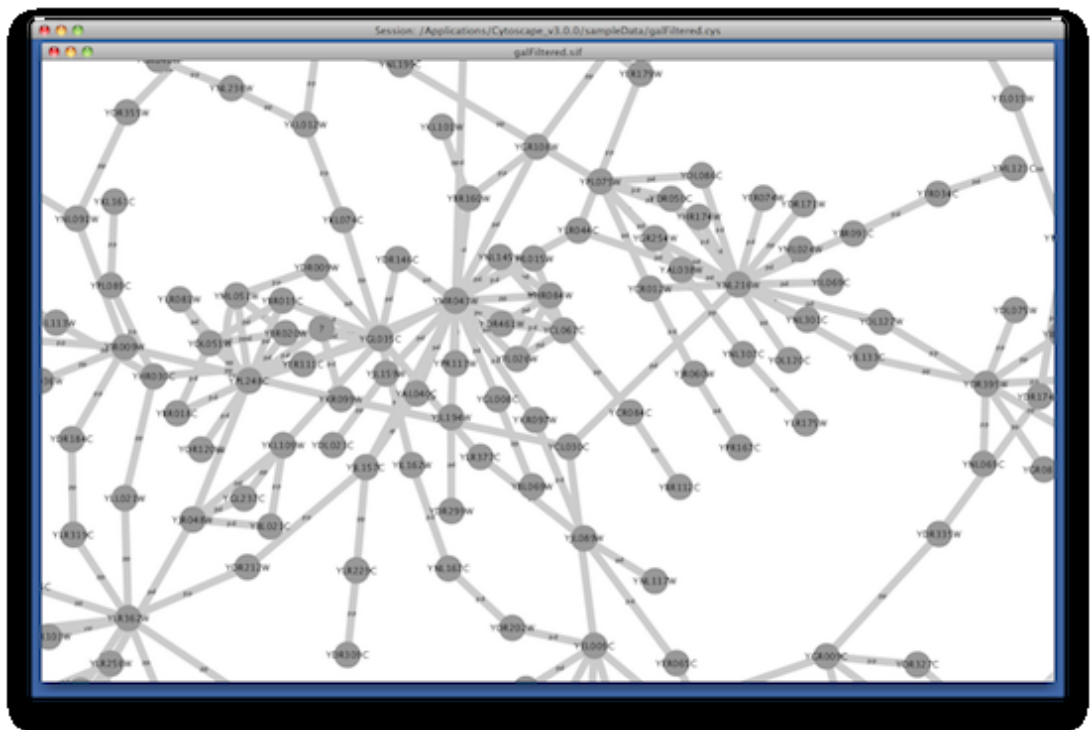


This Visual Style does not have mappings except node/edge labels, but you can modify the network graphics by editing *Default View*.

Additional sample styles are available in the `sampleStyles.xml` file in the *sampleData* directory. You can import the sample file from ***File → Import → Vizmap File…***

# Visual Attributes, Graph Attributes and Visual Mappers

The Cytoscape VizMapper uses three core concepts:

- A *visual attribute* is any visual setting that can be applied to your network. For example, you can change all nodes from circles to squares by changing the node shape visual attribute.

- A *network attribute* is any data attribute associated with a node or an edge. For example, each edge in a network may be associated with a label, such as "pd" (protein-DNA interactions), or "pp" (protein-protein interactions).

- A *visual mapper* maps network attributes to visual attributes. For example, a visual mapper can map all protein-DNA interactions to the color blue, and all protein-protein interactions to the color red.

Cytoscape allows a wide variety of visual attributes to be controlled. These are summarized in the tables below.

| Visual Attributes Associated with Nodes | Description |
|---|---|
| *Node Shape* | The shape of the node. |
| *Node Fill Color* | The color of the node. |
| *Node Selected Paint* | The color of the node when selected. |
| *Node Transparency* | The opacity of the color of the node. Zero means totally transparent, and 255 is most opaque. |
| *Node Border Paint* | The color of the border of the node. |
| *Node Border Transparency* | The opacity of the color of the border of the node. Allows for transparency. |
| *Node Border Line Type* | The type of line used for the border of the node. |
| *Node Border Width* | The width of the node border. |
| *Node Label Font Face* | The font used for the node label. |
| *Node Label Font Size* | The size of the font used for the node label. |
| *Node Label* | The text used for the node label. |
| *Node Label Color* | The color of the node label. |
| *Node Label Transparency* | The transparency of the color of the node label. Allows for transparency. |
| *Node Label Position* | The position of the node label relative to the node. |
| *Node Label Width* | The maximum width of the node label. If the node label is wider than the specified width, Cytoscape will automatically wrap the label on space characters. Cytoscape will not hyphenate words, meaning that if a single word (i.e. no spaces) is longer than maximum width, the word will be displayed beyond the maximum width. |
| *Node Tooltip* | The text of the tooltip that appears when a mouse hovers over the node. |
| *Node Size* | The size of the node. Width and height will be equal. This visual property is mutually exclusive of *Node Height* and *Node Width*. |

| | |
|---|---|
| *Node Height* | The height of the node. Height will be independent of width. This visual property is mutually exclusive of *Node Size*. |
| *Node Width* | The width of the node. Width will be independent of height. This visual property is mutually exclusive of *Node Size*. |
| *Node Custom Graphics 1 - 9* | Custom Graphics objects on the node. These are selected from the *Custom Graphics Manager* |
| *Node Custom Graphics Position 1 - 9* | Position of Custom Graphics relative to the node. By default, this value is set to center. |
| *Node X Location* | X location of the node. Default value of this will be ignored. The value will be used only when mapping function is defined. |
| *Node Y Location* | Y location of the node. Default value of this will be ignored. The value will be used only when mapping function is defined. |
| *Node Visible* | Node is visible or not. By default, this value is set to *true*. |
| *Nested Network Image Visible* | A boolean value that indicates whether a nested network should be visualized (assuming a nested network is present for the specified node). |

| **Visual Attributes Associated with Edges** | **Description** |
|---|---|
| *Edge Stroke Color (Unselected)* | The color of the edge. |
| *Edge Stroke Color (Selected)* | The color of the edge when selected. |
| *Edge Transparency* | The opacity of the color of the edge. Allows for transparency. |
| *Edge Line Type* | The type of stoke used to render the line (solid, dashed, etc.) |
| *Edge Width* | The width of the line. |
| *Edge Label* | The text used for the edge label. |
| *Edge Label Color* | The color of the edge label. |
| *Edge Label Transparency* | The opacity of the color of the edge label. Allows for transparency. |
| *Edge Label Font Face* | The font used for the edge label. |
| *Edge Label Font Size* | The size of the font used for the edge label. |
| *Edge Tooltip* | The text of the tooltip that appears when a mouse hovers over the edge. |
| *Edge Source Arrow Unselected Paint* | The color of the arrow on the source node end of the edge. |
| *Edge Source Arrow Selected Paint* | The selected color of the arrow on the source node end of the edge. |
| *Edge Source Arrow Shape* | The shape of the arrow on the source node end of the edge. |
| *Edge Target Arrow Unselected Paint* | The color of the arrow on the target node end of the edge. |
| *Edge Target Arrow Selected Paint* | The selected color of the arrow on the target node end of the edge. |
| *Edge Target Arrow Shape* | The shape of the arrow on the target node end of the edge. |
| *Edge Bend* | The edge bend. Defines how the edge is rendered. Users can add multiple handles to define how to bend the edge line. |

| | |
|---|---|
| *Edge Curved* | If *Egde Bend* is defined, edges will be rendered as straight or curved lines. If this value is set to *true*, edges will be drawn as curved lines. |
| *Edge Visible* | Edge is visible or not. By default, this value is set to *true*. |

| Network Visual Properties | Description |
|---|---|
| *Network Background Paint* | The background color of the network view. |
| *Network Title* | The title of the network view. |
| *Network Height* | The height of the network view. |
| *Network Width* | The width of the network view. |
| *Network Scale Factor* | The zoom level of the network view. |
| *Network Center X Location* | The X location of network view center. |
| *Network Center Y Location* | The Y location of network view center. |
| *Network Edge Selection* | Edges are selectable or not. If this is *false*, users cannot select edges. |
| *Network Node Selection* | Nodes are selectable or not. If this is *false*, users cannot select nodes. |

| Available Shapes and Line Styles | Sample |
|---|---|
| *Node Shapes* | |
| *Edge Line Types* | |
| *Arrow Shapes* | |

For each visual attribute, you can specify a default value or define a dynamic visual mapping. Cytoscape currently supports three different types of visual mappers:

1. **Passthrough Mapper**

   • The values of network attributes are passed directly through to visual attributes. A passthrough mapper is typically used to specify node/edge labels. For example, a passthrough mapper can label all nodes with their common gene names.

2. **Discrete Mapper**

   • Discrete data attributes are mapped to discrete visual attributes. For example, a discrete mapper can map different types of molecules to different node shapes, such as rectangles for gene products and ellipses for metabolites

3. **Continuous Mapper**

   • Continuous data are mapped to visual attributes. Depending on the visual attribute, there are three kinds of continuous mappers:

     i. **Continuous-to-Continuous Mapper**: for example, you can map a continuous numerical value to a node size.

     ii. **Color Gradient Mapper**: This is a special case of continuous-to-continuous mapping. Continuous numerical values are mapped to a color gradient.

     iii. **Continuous-to-Discrete Mapper**: for example, all values below 0 are mapped to square nodes, and all values above 0 are mapped to circular nodes.

- However, note that there is no way to smoothly morph between circular nodes and square nodes.

The table below shows visual mapper support for each visual property.

**Legend**

| Symbol | Description |
|--------|-------------|
| - | Mapping is not supported for the specified visual property. |
| + | Mapping is fully supported for the specified visual property. |
| o | Mapping is partially supported for the specified visual property. Support for "continuous to continuous" mapping is not supported. |

**Node Visual Mappings**

| Node Visual Properties | | Passthrough Mapper | Discrete Mapper | Continuous Mapper |
|---|---|---|---|---|
| Color | Node Color | o | + | + |
| | Node Transparency | o | + | + |
| | Node Border Color | o | + | + |
| | Node Border Transparency | o | + | + |
| | Node Label Color | o | + | + |
| | Node Label Transparency | o | + | + |
| Numeric | Node Size/Width/Height | o | + | + |
| | Node Font Size | o | + | + |
| | Node Line Width | o | + | + |
| | Node Label Width | o | + | + |
| Other | Node Border Line Type | o | + | o |
| | Node Shape | o | + | o |
| | Node Label | + | + | o |
| | Node Tooltip | + | + | o |
| | Node Font Family | o | + | o |
| | Node Label Position | o | + | o |
| | Node Show Nested Network | o | + | o |

**Edge Visual Mappings**

| Edge Properties | | Passthrough Mapper | Discrete Mapper | Continuous Mapper |
|---|---|---|---|---|
| Color | Edge Color | o | + | + |
| | Edge Transparency | o | + | + |
| | Edge Target Arrow Color | o | + | + |
| | Edge Source Arrow Color | o | + | + |
| | Edge Label Color | o | + | + |
| | Edge Label Transparency | - | + | + |
| Numeric | Edge Line Width | - | + | + |

| Other | | | | |
|---|---|---|---|---|
| | *Edge Font Size* | - | + | + |
| | *Edge Label Width* | - | + | + |
| Other | *Edge Line Type* | o | + | o |
| | *Edge Source Arrow Shape* | o | + | o |
| | *Edge Target Arrow Shape* | o | + | o |
| | *Edge Label* | + | + | o |
| | *Edge Tooltip* | + | + | o |
| | *Edge Font Family* | - | + | o |

# Text Passthrough Mapper (New feature for version 2.8)

In Cytoscape 2.8.0 and later versions, the Passthrough Mapper can recognize some text representations of values. This means, if you have an integer attribute named *Node Size Values*, you can directly map those values as the Node Size by setting *Node Size Values* as controlling attribute name for Node Size Passthrough mapping. The following value types are supported:

- Color: Standard color names supported by all browsers or RGB representation in hex

- Numerical Values: Automatically mapped to the specified Visual Property.

- Custom Graphics: URL String. If the URL is valid and an actual image data exists there, Cytoscape automatically downloads the image and maps it to the node.

## Examples

1. Color Passthrough Mapping



2. Node Size Passthrough Mapping

| Node Size | node size |
| --- | --- |
| Mapping Type | Passthrough Mapping |
| YOL016C | 20 |
| YBR109C | 100 |
| YFR014C | 120 |
| YGL106W | 200 |
| YOR326W | 50 |
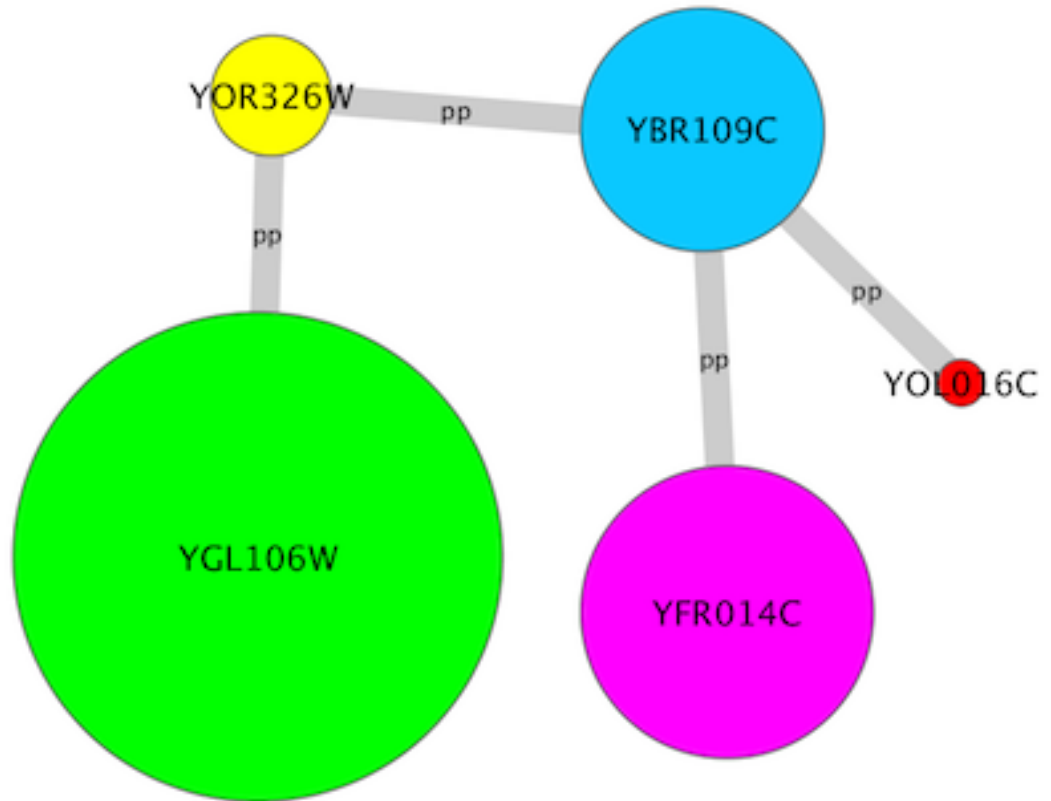


3. Custom Graphics Passthrough Mapping

| ▼ Node Custom Graphics 1 | Graphics Location URL as Strings |
|---|---|
| Mapping Type | Passthrough Mapping |
| YJR066W | http://www.cytoscape.org/images/logo.png |
| YKL074C | http://www.cytoscape.org/images/logo/ucsd_logo... |
| YKL012W | |
| YLR116W | |
| YGR046W | |
| YNL236W | |
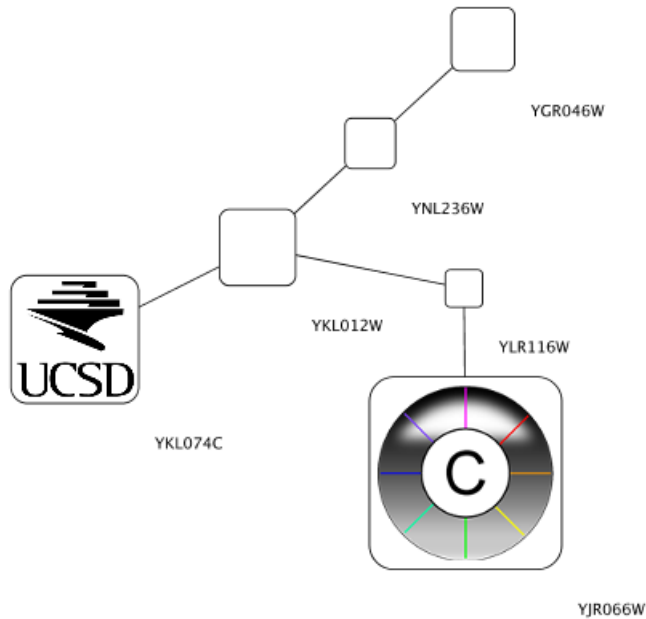
# Custom Graphics Manager
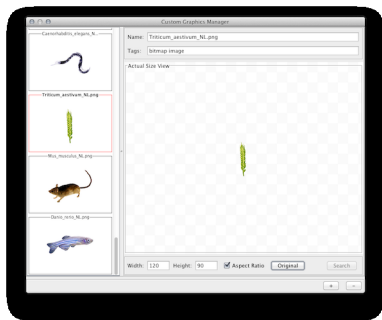


For Cytoscape 2.8.0 and later versions, Cytoscape supports *Custom Graphics* for nodes. You can add all kinds of bitmap images, such as jpg, png, or gif, on the top of network nodes. From the user's point of view, this is simply an addition to the Visual Properties, and you can use same VizMap user interface to map Custom Graphics to nodes. However, before mapping images, you need to prepare Custom Graphics in the *Custom Graphics Manager*. It is a simple GUI component to add/remove images to or from a Cytoscape session.

Taxonomy Icon set used in this section is created by Database Center for Life Science (DBCLS) and is distributed under Creative Commons License (CC BY 2.1.)

## Adding New Images to Cytoscape

The *Custom Graphics Manager* supports drag and drop for image files and URLs. If you want to add images from a web browser or local file system, you can drag images from them and drop those images onto the list of images on the left.
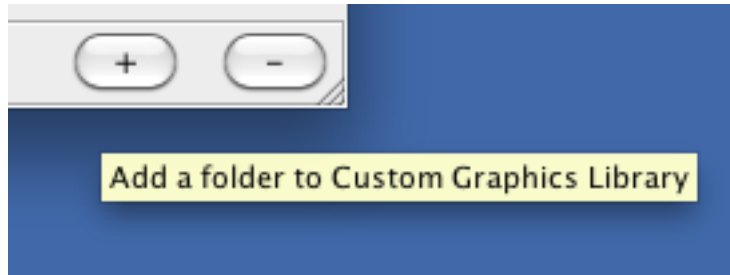
If you want to add all images in a folder, press the **+** button on the bottom of the Custom Graphics Manager window.

**Note: When you drag and drop images from web browser, make sure that you are actually dragging the URL for the image. In some cases, images are linked to an HTML page or scripts, and in such cases, this drag and drop feature may not work.**

# Removing Images

To remove images from the current session's Custom Graphics library, simply select graphics from the list and press the **-** button.

# Resizing Images

You can resize images by typing width/height in the text box. If *Keep Aspect Ratio* box is checked, the width-height ratio is always synchronized. Any time, you can resize the image to the original size by pressing **Original** button.

# Using Custom Graphics in the VizMapper
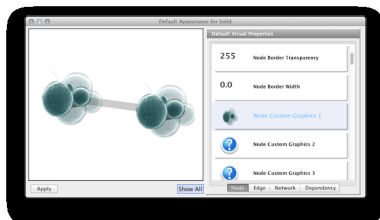


**Custom Graphics** is a new type of Visual Property and you can use them from the standard VizMapper user interface. There are nine Custom Graphics Visual Properties (**Node Custom Graphics 1 - 9**) and they will be displayed in the standard VizMapper Default View Editor or Mapping Editors.

## Custom Graphics Positions

Each Custom Graphics Visual Property is associated with a position. You can edit their positions by using same UI as Label Position.

## Z-Ordering

This number that appears with the Custom Graphics Visual Property represents an ordering of layers. *Basic node color and shape are always rendered first, then Node Custom Graphics 1, 2, ..., through 9.*

## Synchronize Custom Graphics to Node Size
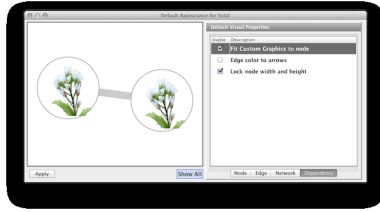


By default, Custom Graphics objects are automatically resized to be consistent with the Node Size Visual Property. To control Custom Graphics size separately, uncheck the *Fit Custom Graphics to node* dependency. You can find this check box in the Default Editor's **Dependencies** panel.

**Custom Graphics Selector**

To select a Custom Graphics, just click one of them and press **Apply**. To remove a value from mappings/defaults, select the **Remove Graphics** option from the Custom Graphics list.

# Saving and Loading Custom Graphics

In general, saving and loading Custom Graphics is automatic. When you quit Cytoscape, all of the Custom Graphics in the manager will be saved automatically. There are two types of saving:

1. To session file

   When you save the current session to a file, the Custom Graphics **used in Visual Styles** will be saved to that file. For example, if you have a Visual Style with a discrete mapping for *Custom Graphics*, all Custom Graphics used in the style will be saved to the session file. Other graphics will **not** be saved in your session file. This is because your image library can be huge when you add thousands of images to the Custom Graphics Manager and it takes very long time to save and load the session file.

2. Automatic saving to `CytoscapeConfiguration/images3` directory

   When you select *File → Quit*, all of Custom Graphics in the Manager will be saved automatically to your Cytoscape setting directory. Usually, it's `YOUR_HOME_DIRECTORY/CytoscapeConfiguration/images3`.

In any case, Custom Graphics will be saved automatically to your system or session and will be restored when you restart Cytoscape or load a session.

# Visual Styles Tutorials

The following tutorials demonstrate some of the basic VizMapper features. Each tutorial is independent of the others.

## Tutorial 1: Create a Basic Visual Style and Set Default Values

The goal of this tutorial is to learn how to create a new Visual Style and set some default values.

**Step 1. Load a sample network.** From the main menu, select *File ← Import ← Network ← File...*, and select `sampleData/galFiltered.sif`.

**Step 2. Create some node/edge statistics by Network Analyzer.** Network Analyzer calculates some basic statistics for nodes and edges. From the main menu, select *Tools ← Network Analyzer ← Network Analysis ← Analyze Network*, and click *OK*. Once the result is displayed, simply close the window. All statistics are stored as a regular table data.

**Step 3. Open the VizMapper.** Select the *VizMapper* tab in the Control Panel at the left of the screen. You will now see the VizMapper main panel, as shown below.

**Step 4. Create a new visual style.** Click the Options
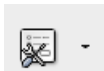


button, and select *Create New Visual Style*. Then enter a name for your new visual style when prompted. You will see an empty visual style in the VizMapper Main Panel, as shown below.
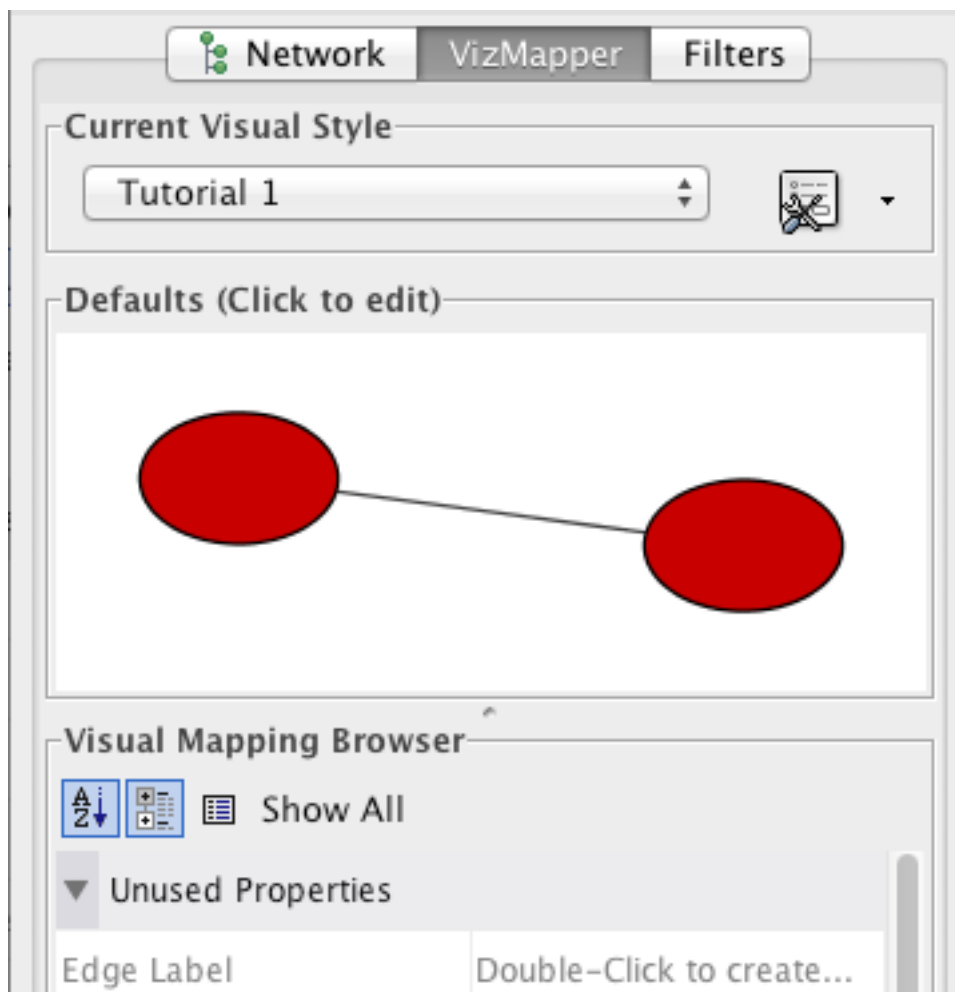


Since no mappings are set up yet, all visual attributes are listed in the *Unused Properties* category. From this panel, you can create node/edge mappings for all visual properties.

**Step 5. Edit default values.** Open the Default Appearance Editor by clicking on the *Defaults* graphics window (shown below) in the *VizMapper* Main Panel.
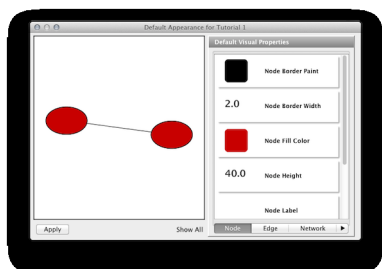
**Step 6. Change the default node color and shape.** To set the default node shape to triangles, click *Node Shape* in the Default Visual Properties list. A list of available node shapes will be shown. Click on the *Triangle* icon and then click the *Apply* button. The Default Appearance Editor will be automatically updated. You can edit other default values by clicking on visual attribute names on the list. In the example shown below, the node shape is set to *Round Rectangle*, while the **Node Fill Color** is set to white.
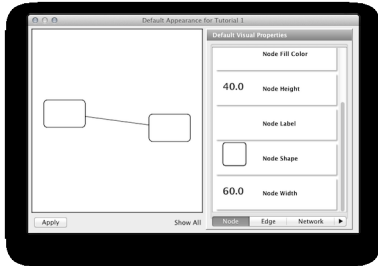


**Step 7. Apply your settings.** When you finish editing, click the *Apply* button at the bottom of the editor. Your new Visual Style will be applied to the current network, as shown below.
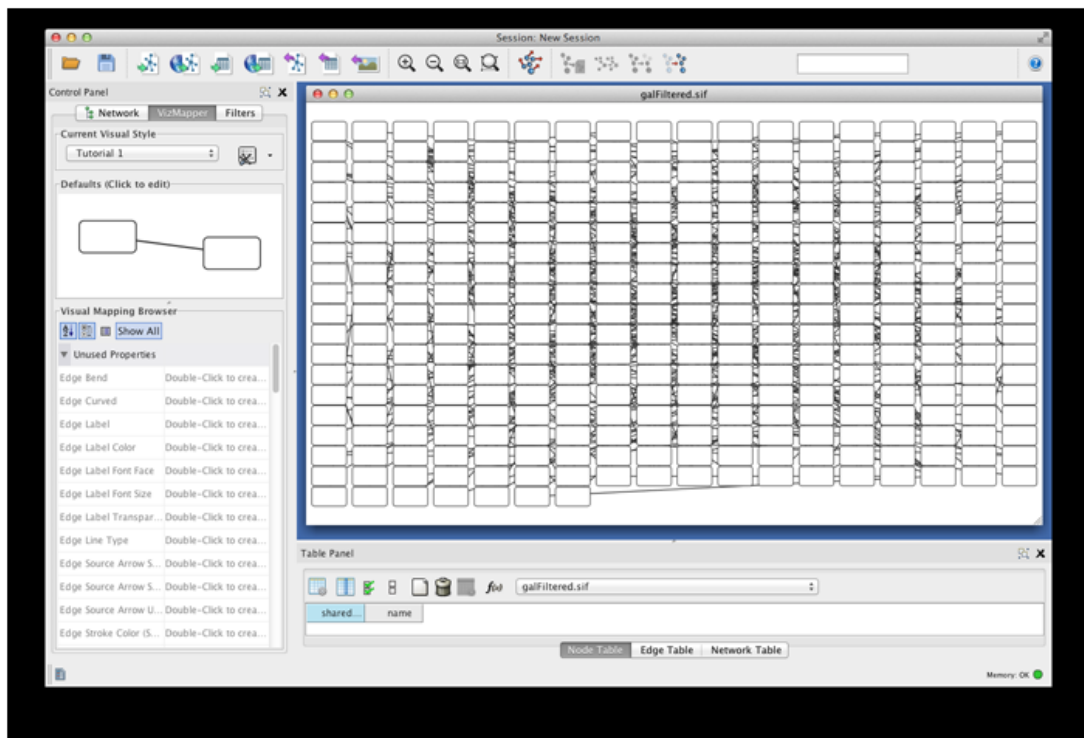


# Tutorial 2: Creating a New Visual Style with a Discrete Mapper

Now you have a network with new Visual Style. The following section demonstrates how to create a new visual style using a discrete mapper. The goal is to draw protein-DNA interactions as dashed lines, and protein-protein interactions as solid lines.

**Step 1. Choose a visual attribute.** Double click the **Edge Stroke Color (Unselected)** entry listed in *Unused Properties*. It will now appear at the top of the list, under the *Edge Visual Property* category (as shown below).

**Step 2. Choose an edge column.** Click on the cell to the right of the *Edge Stroke Color* entry and select "interaction" from the dropdown list that appears.

**Step 3. Choose a mapping type.** Set the "Discrete Mapping" option as the *Mapping Type*. All available attribute values for "interaction" will be displayed, as shown below.



**Step 4. Set the mapping relationship.** Click the empty cell next to "pd" (protein-DNA interactions). On the right side of the cell, **...** and **X** buttons will appear. Click on the **...** button. A popup window will appear; select green or similar, and the change will immediately appear on the network window.



Repeat step 4 for "pp" (protein-protein interactions), but select a darker color as edge stroke color. Then repeat steps 3 through 4 for the *Edge Line Type* attribute. You can select the correct line style ("Dash" or "Solid") from the list.
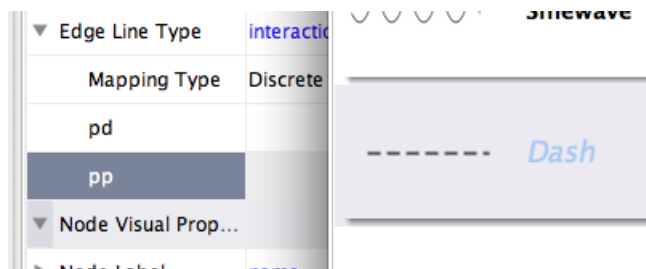


Now your network should show "pd" interactions as dashed green lines and "pp" interactions as solid lines. A sample screenshot is provided below.
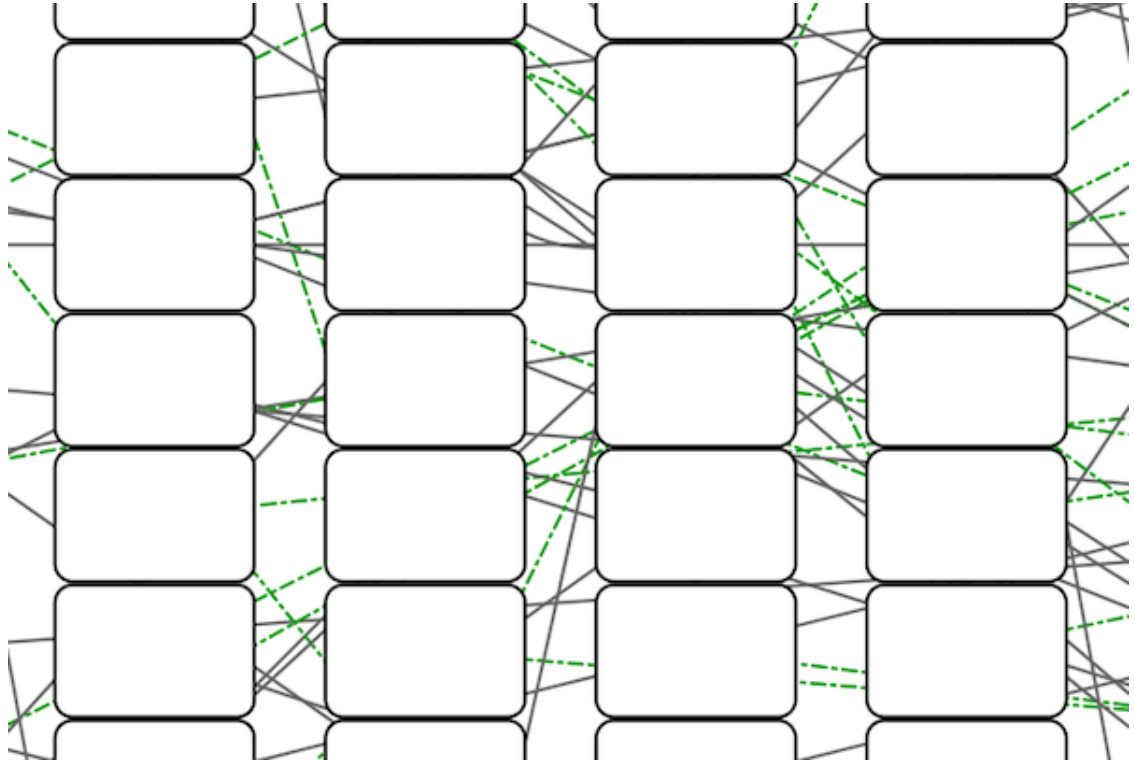
## Tutorial 3: Creating a New Visual Style with a Continuous Mapper

At this point, you have a network with some edge visual mappings. Next, let's create mappings for nodes. The following section demonstrates how to create a new visual style using a continuous mapper. The goal is to superimpose node statistics (in this example, node degree) onto a network and display it along a color gradient.

**Step 1. Choose a visual attribute.** Double click the '**Node Fill Color**' entry listed in *Unused Properties*. Node Fill Color will now appear at the top of the list, under the *Node Visual Property* category.

**Step 2. Choose a node table column.** Click on the cell to the right of the *Node Fill Color* entry and select "Degree" from the dropdown list that appears.

**Step 3. Choose a mapping type.** Set the "Continuous Mapping" option as the *Mapping Type*. This automatically creates a default mapping



**Step 4. Define the points where colors will change.** Double-click on the black-and-white gradient rectangle next to Graphical View to open the Color Gradient Mapper. Click and drag one to left, and move the second point to right.
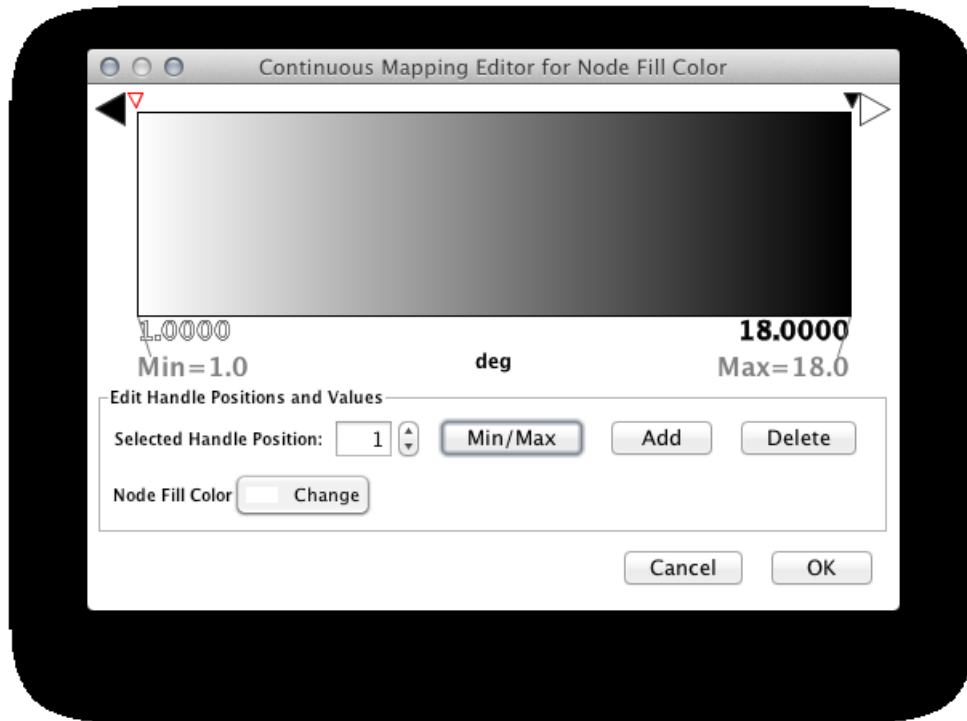
**Step 5. Define the colors between points.** Double-click on the leftmost triangle (facing left) and a color palette will appear. Choose a shade of yellow and click OK. Double-click on the triangle at left and set the color white. For the triangle at the right, set its color to green.
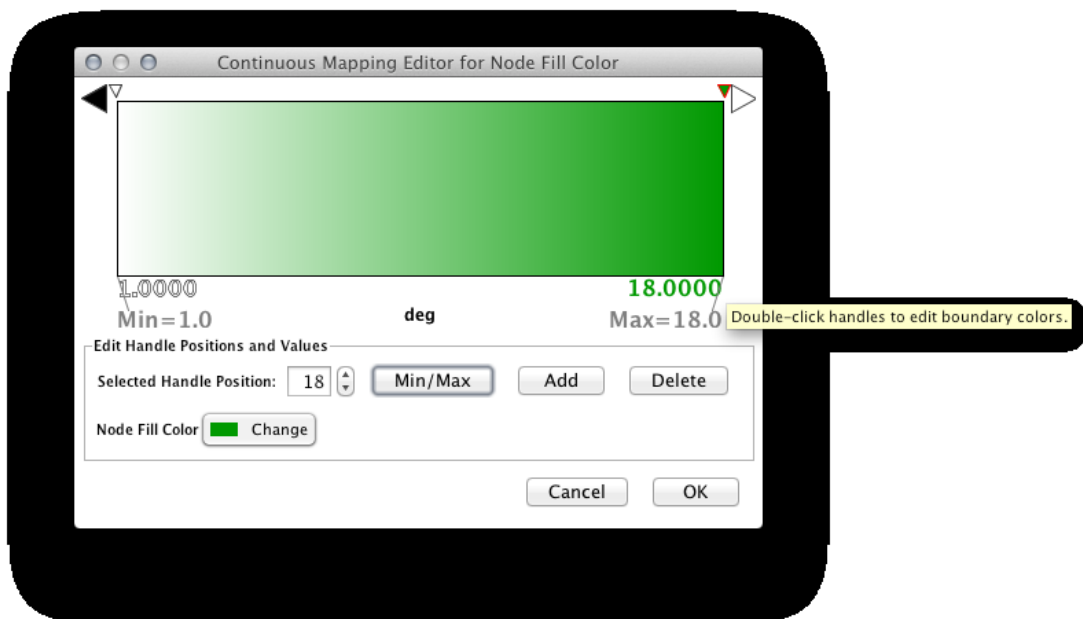


The color gradients will immediately appear in the network window. All nodes with degree 1 will be set to white, and all values between 1 and 18 will be painted with a white/green color gradient. A sample screenshot is below.
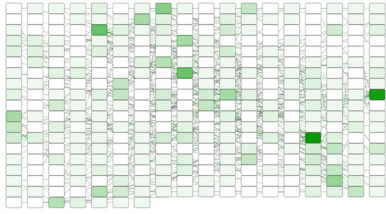
**Step 6. Repeat for other Visual Properties.** You can create some more mappings for other numeric table data. For example, edge data table column "EdgeBetweenness" is a number, so you can use it for continuous mapping. The following is an example visualization by mapping *Edge Width* to "EdgeBetweenness".
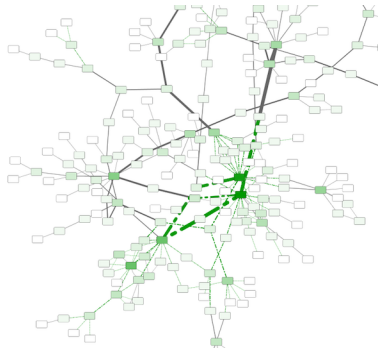


# Tutorial 4: How to Use Utilities for Discrete Mappers

The following tutorial demonstrates utilities for editing discrete mappings. The goal of this section is learning how to set and adjust values for discrete mappings automatically.

1. Switch the Visual Style to *Minimal*. Now your network looks like the following:

6. Cretate a **Discrete Node Color Mapping**. Select "AverageShortestPathLength" (generated by Network Analyzer) as controlling attribute.

7. Click the **Node Fill Color** cell, then right-click it to show the popup menu. Select *Mapping Value Generators ← Rainbow*. It generates different colors for different table values as shown below:
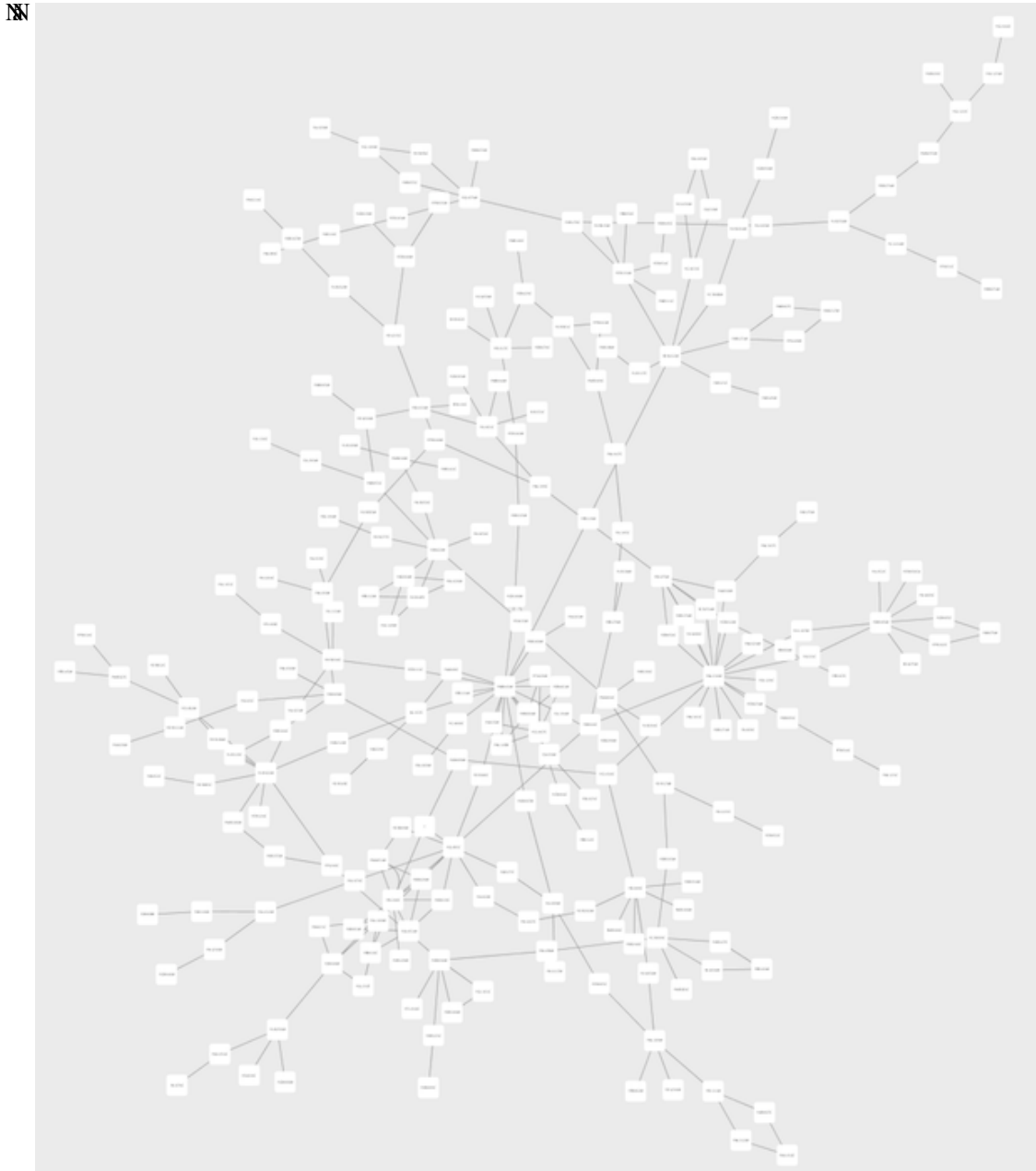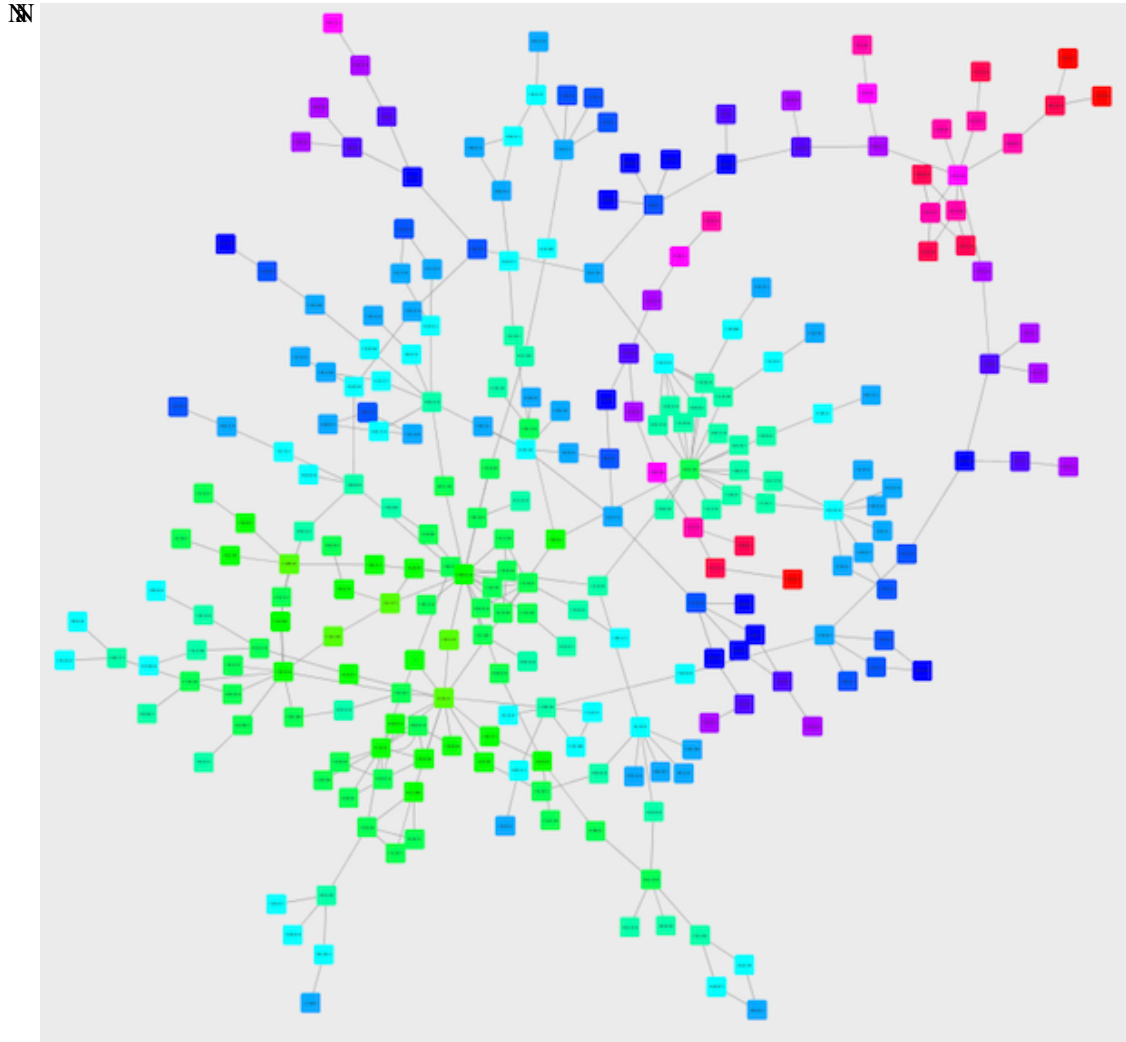
- Create a **Discrete Node Label Font Size Mapping**. Select "AverageShortestPathLength" as controlling attribute.

- Click the **Node Label Font Size** cell, then right-click it to show the popup menu. Select *Mapping Value Generators ← Number Series*. Type *3* for the first value and click OK. Enter *3* for increment.

- Apply *Layout ← yFiles Layouts ← Organic*. The final view is shown below:

This mapping generator utility is useful for categorical data. The following example is created by adding node color discrete mapping from species column to color.

# Tutorial 5: Using Custom Graphics in Visual Styles

This tutorial is a quick introduction to Custom Graphics feature. You can assign up to nine images per node as a part of Visual Style.

1. Prepare images. These can be any type of bitmap graphics. In this tutorial, we are going to use the Crystal Project's PNG icons. You can download it from here.

2. Start Cytoscape and select *View ← Open Custom Graphics Manager*. Cytoscape 3 has some preset images and in this example, we use these presets. You can close the Custom Graphics Manager window for now.
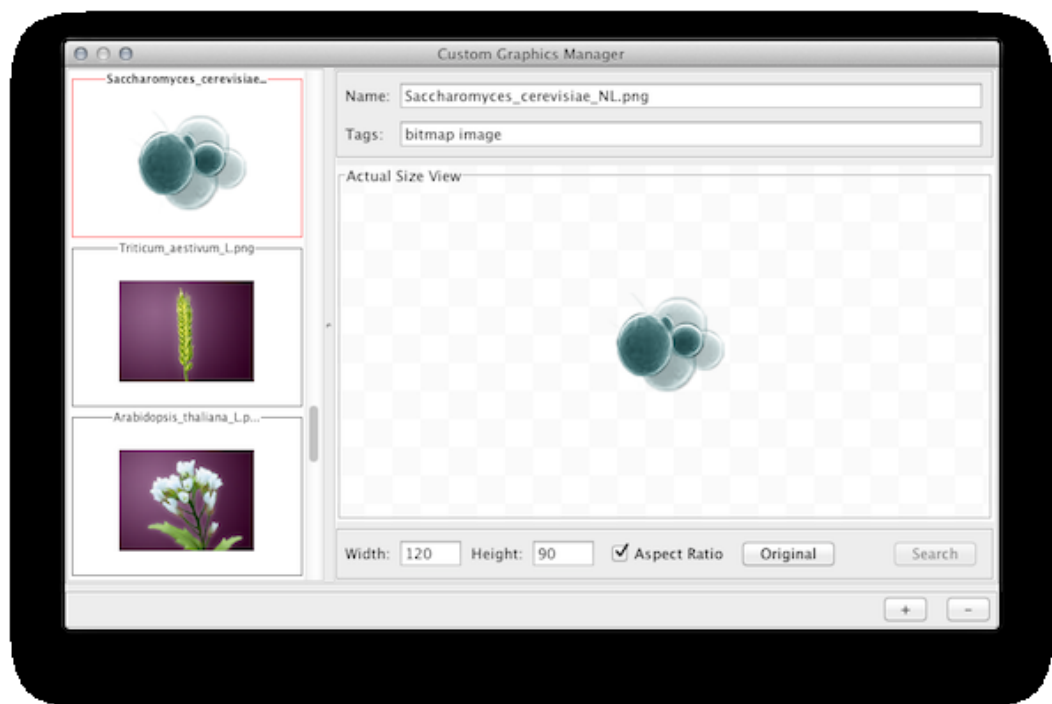


3. Load a network and run the Network Analyzer (*Tools ← Network Analyzer ← Network Analysis ← Analyze Network*). This creates several new table columns (statistics for nodes and edges).

4. Click the *VizMapper* tab in the *Control Panel*, and select the *Solid* style.

5. Click the *Defaults* section to open the *Default Appearance Editor*.

6. Click **Node Custom Graphics 1** and select any of the custom graphics from the list. Click **Node Transparency** and set the value to zero.
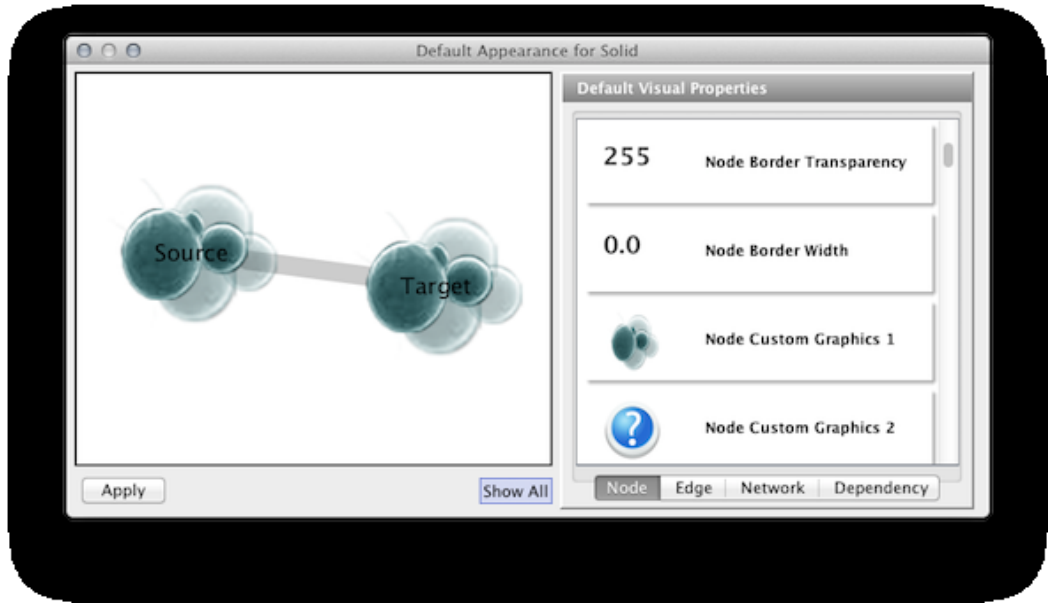
7. Press *Apply*. Now your network looks like the following:

8. Open the Custom Graphics Manager again. Drag and Drop this



icon to the image list. It automatically adds it to the manager.
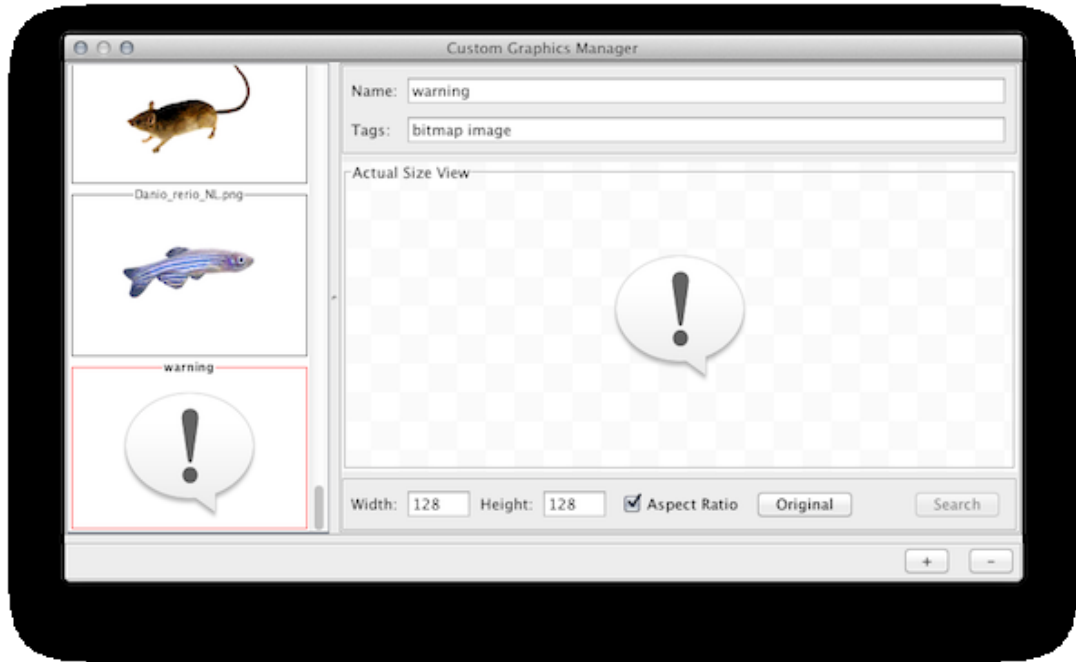


1. Create a *Continuous Mapping* for **Node Custom Graphics 2**. Select "BetweennessCentrality" as controlling column and move the handle to *0.2*. Double click the region over *0.2* and set the new icon you have just added in the last step.
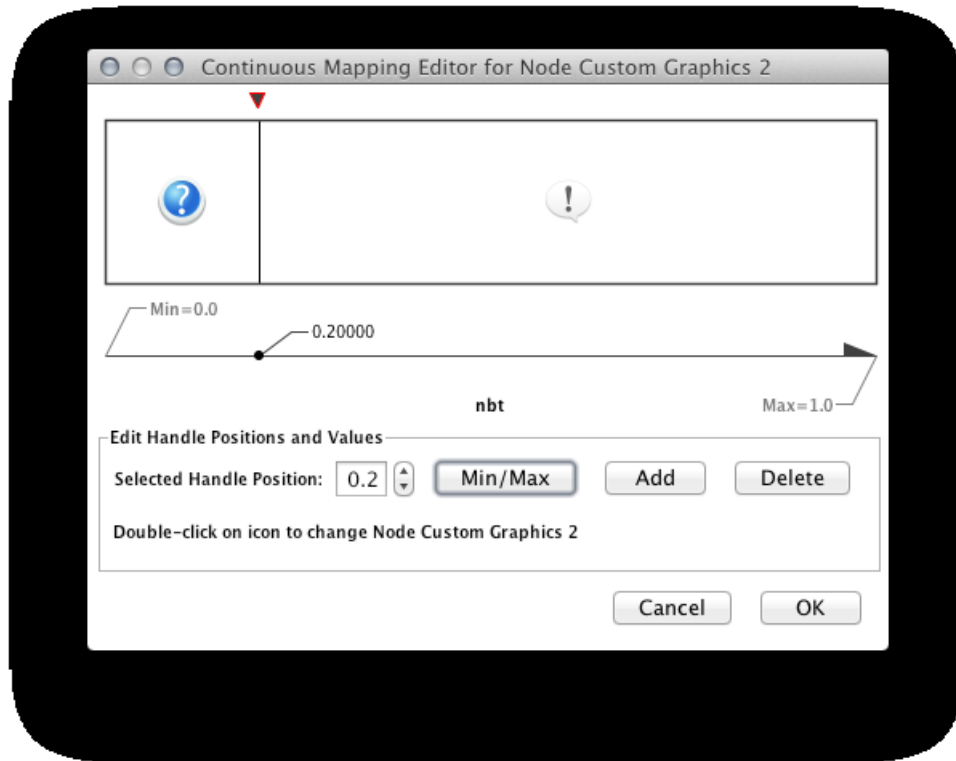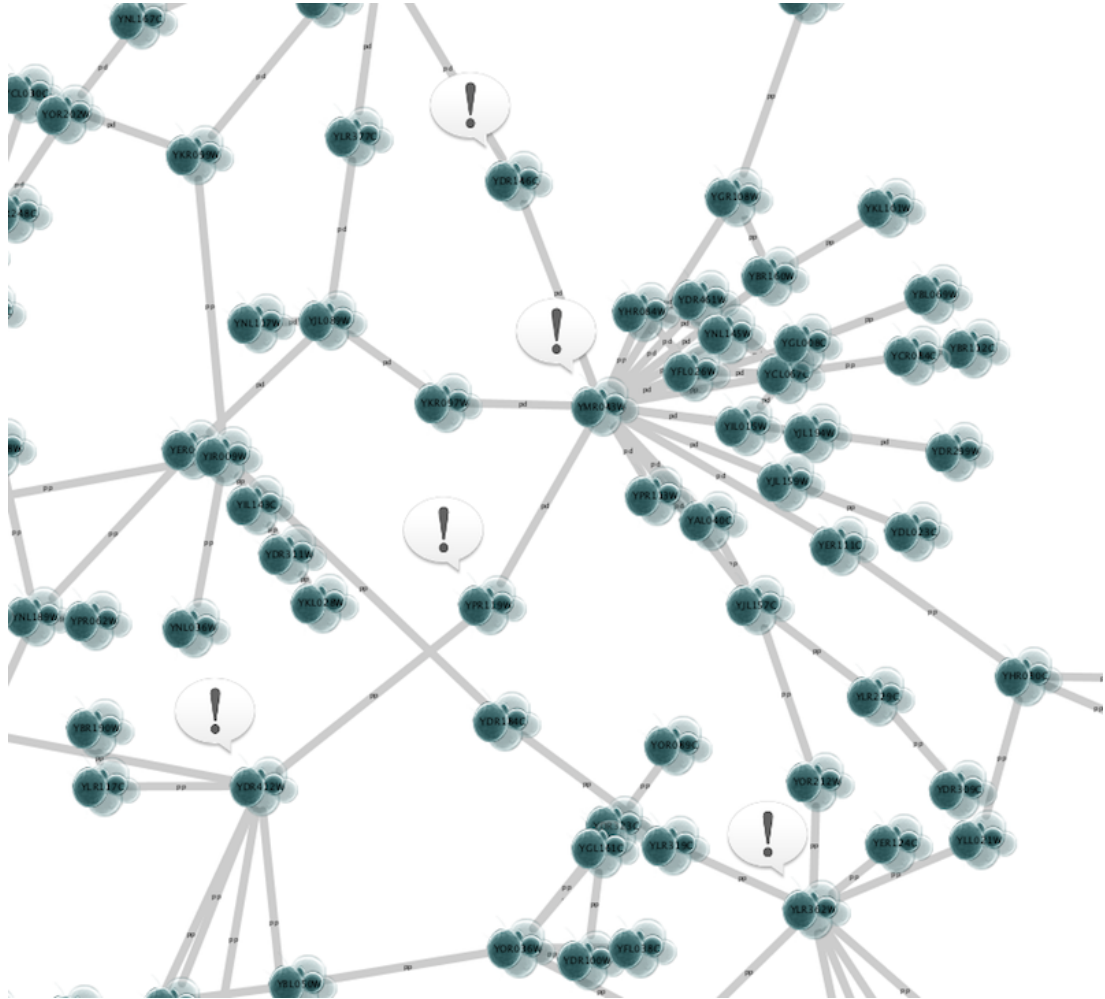
1. Press *Apply* and now open the *Default Appearance Editor* again. Click **Node Custom Graphics Position 2** and move the position of the graphics to upper left.
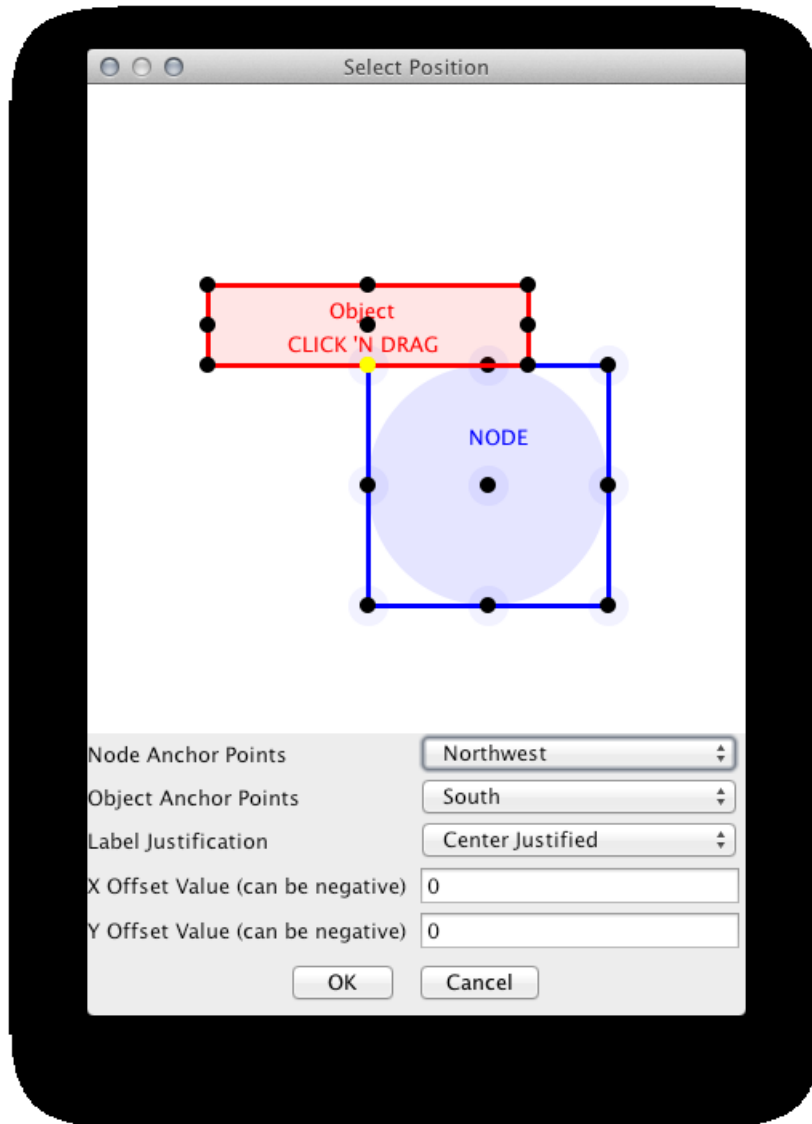
1. Press *Apply*. Now the important nodes in the network (nodes with high betweenness centrality) is annotated with the icon.

# Advanced Topics

## Editing Discrete Mappings

Several utility functions are available for Discrete Mappings. You can use those functions by right clicking anywhere on the *Visual Mapping Browser* (shown below.)
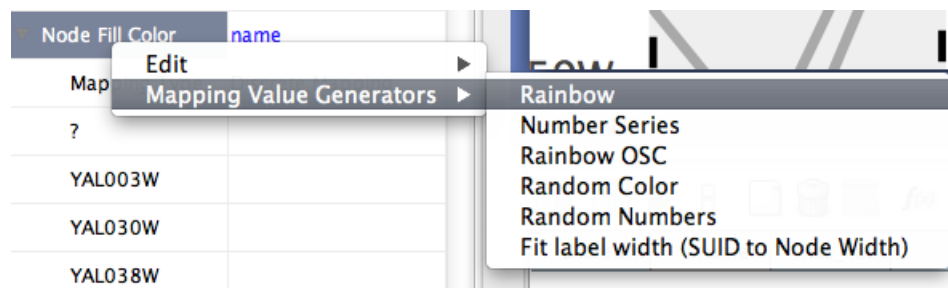
## Automatic Value Generators

- *Mapping Value Generators* - Functions in this menu category are value generators for discrete mappings. Users can set values for discrete mappings automatically by these functions.

  - **Rainbow** and **Rainbow OSC** - These functions try to assign as diverse a set of colors as possible for each data value.



  - **Random Numbers** and **Random Colors** - Randomize numbers and colors. If you use this function for numerical values (node size, opacity, etc.) you need to specify a range. For example, if you want to set values from 1 to 100, you need to type *1-100* in the dialog.

  - **Number Series** - Set a series of numbers to the specified mapping.
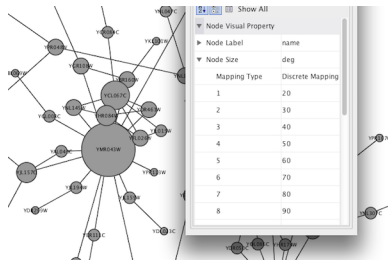


  - **Fit label width** - This function is only for node width and height. When the node size is *unlocked* AND *Node Width/Height* discrete mappings are available, you can fit the size of each node to its label automatically by selecting this function. See the example below:
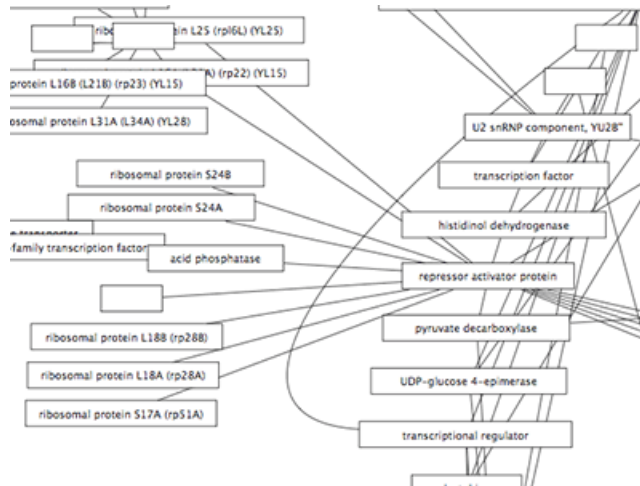
## Edit Selected Values at Once

You can set multiple values at once. First, you need to select rows in which you want to change values then select *Edit all selected rows*... under *Edit* right-click menu. A dialog pops up and you can enter the new value for the selected rows.

## Visual Property Dependencies

The fourth tab in the Default Editor is the list of available *Visual Property Dependencies*. A Visual Property Dependency can be established between different visual properties. Currently there are three dependencies: **Lock Node with and height**, **Edge color to arrows**, and **Fit Custom Graphics to node**.
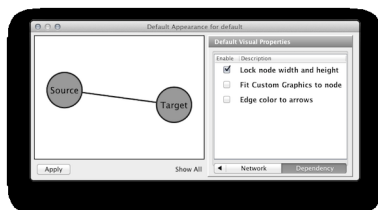


- **Lock Node with and height** - If this menu item is checked, *Node Width* and *Node Height* mappings are ignored and *Node Size* overrides them. If you want to use *Fit node size to label* function, you need to unlock this.

- **Edge color to arrows** - If this menu item is checked then *Edge Source Arrow Color* and *Edge Target Arrow Color* are overridden and *Edge Color* is used in both cases.

- **Fit Custom Graphics to node** - If a Custom Graphics is assigned to a node and this option is enabled, Custom Graphics will be resized to fit in the node.

# Working with Continuous Mapping Editors

There are three kinds of Continuous Mapping Editors. Each of them are associated with a specific visual attributes:

| *Editor Type* | *Supported Data Type* | *Visual Attributes* |
|---|---|---|
| **Color Gradient Editor** | Color | node/edge/border/label colors |

| Continuous-Continuous Editor | Numbers | size/width/opacity |
|---|---|---|
| Continuous-Discrete Editor | All others | font/shape/text |

## Range Setting Panel



Each editor has a common section named ***Range Setting***.

1. **Handle Value Box** - This box displays the current value for the selected slider handle. Also you can directly type the value in this box to move the slider to an exact location.

2. **Min/Max Button** - Set the overall range of this editor. First time you open the editor, the *Min* and *Max* values are set by the range of attribute you selected, i.e., minimum and maximum value of the attribute will be set to the range of this editor. You can change this range anytime you want by pressing this button.

3. **Add Handle Button** - Add a new handle to the editor.

4. **Delete Handle Button** - Delete the selected handle from the slider widget.

5. **Handle Value Editor Button** - Edit value assigned to the selected handle.

## Gradient Editor
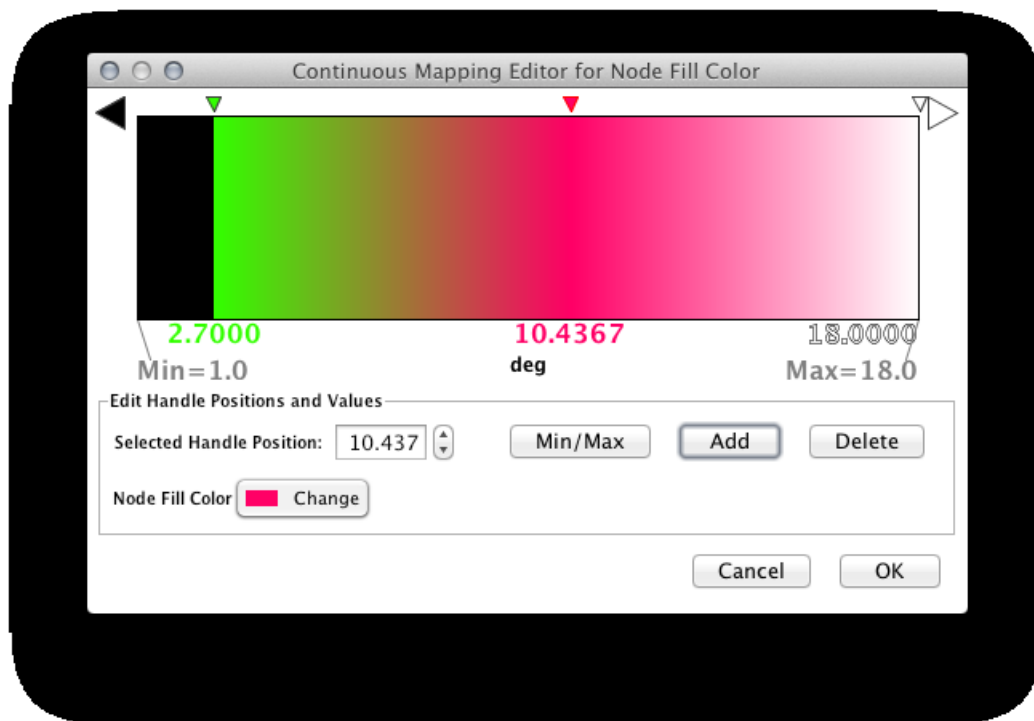
The Gradient Editor is an editor for creating continuous mappings for colors. To change the color of each region, just double click the handles (small triangles on the top). A Color gradient will be created only when the editor has two or more handles (see the example below).

| 1 handle (no gradient) | 2 handles |
|---|---|
| | |

## Continuous-Continuous Editor
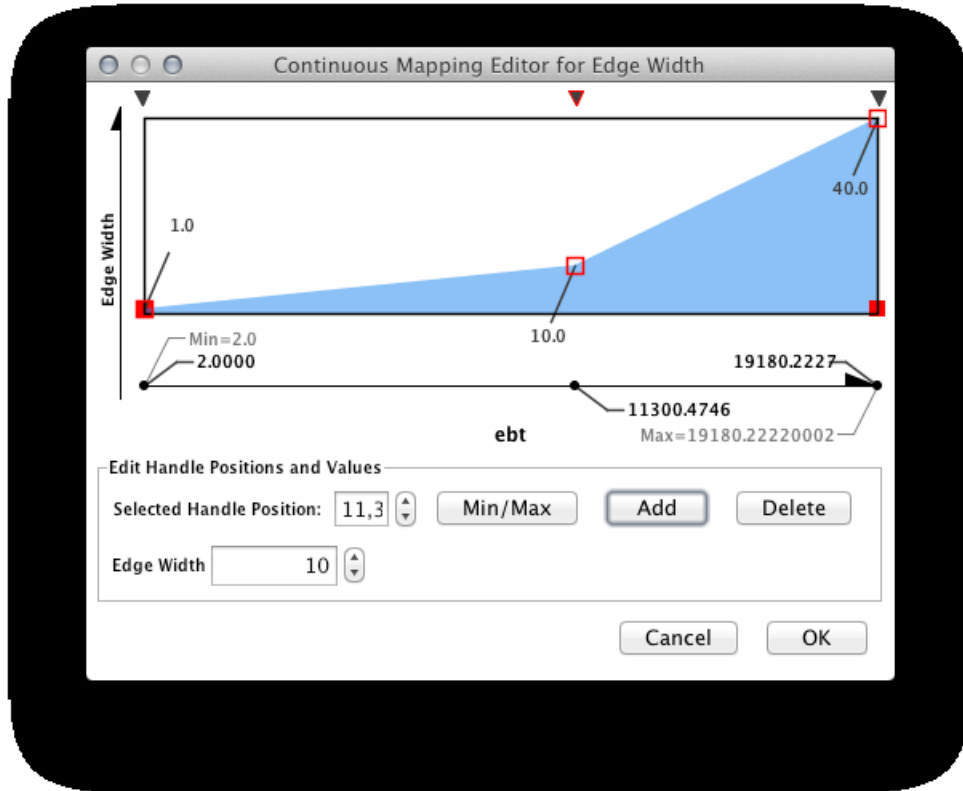


The Continuous-Continuous Editor is for creating mappings between numerical attributes and numerical visual properties (size/opacity). To change the value assigned on Y-axis (the visual property shown in the example above is node size), drag the red squares or double click on the squares to directly type an exact value.

**Continuous-Discrete Editor**



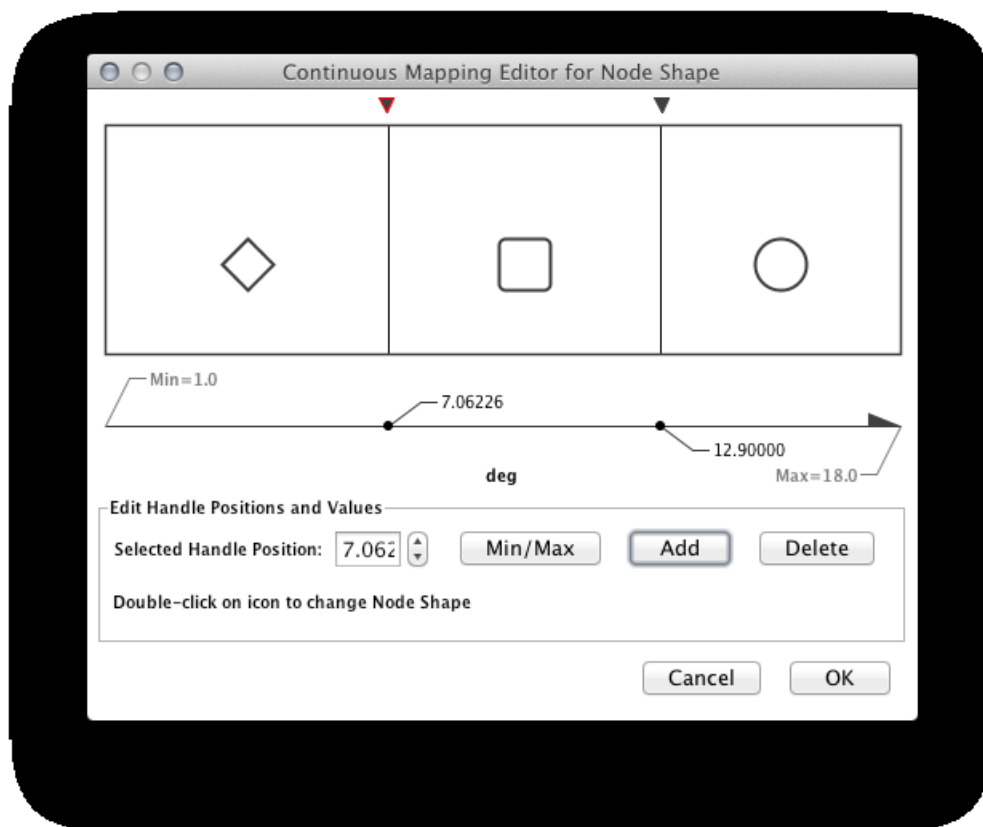The Continuous-Discrete Editor is used to create mappings from numerical attribute values to discrete visual properties, such as font, shape, or line style. To edit a value for a specific region, double click on the icon on the track.

# Managing Visual Styles

All Cytoscape Visual Style settings are initially loaded from a default file that cannot be altered by users. When users make changes to the visual properties, a `session_vizmap.xml` file is saved in the session file. This means that if you save your session, you will not lose your visual properties. No other vizmap files are saved during normal operation.

## Saving Visual Styles

Visual styles are automatically saved with the session they were created in. Before Cytoscape exits, you will be prompted to make sure you save the session before quitting. It is also possible to save your visual styles in a file separate from the session file. To do this, navigate to the *File ← Export ← Vizmap...* menu option and save the properties as an XML file. This feature can be used to share visual styles with other users.

## Importing Visual Styles

To import existing visual styles, navigate to the *File ← Import ← Vizmap File...* menu option and select a `vizmap.props` (Cytoscape 2 format) or `vizmap.xml` (Cytoscape 3 format) file. Imported properties

will supplement existing properties or override existing properties if the properties have the same name. You can also specify a visual properties file using the -V command line option (`cytoscape.sh -V myVizmap.props`). Visual properties loaded from the command line will override any default properties.

# Bypassing Visual Styles

Cytoscape has a feature that allows users to override visualizations created by the VizMapper for individual nodes and edges. This feature is available by right-clicking on a node or edge and then clicking on *Edit* ⇐ *Bypass Visual Style*.
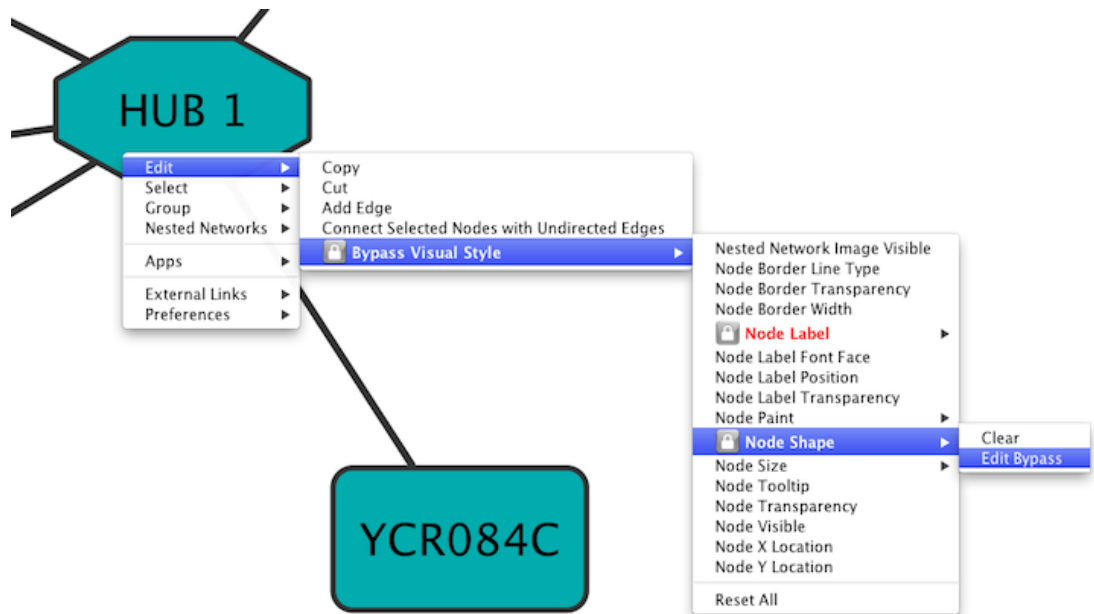


Each visual property of the node or edge is displayed. When a property is overridden, a lock icon



appears next to the property and *Clear* / *Edit Bypass* menu options appear. By clicking the *Clear* option, the bypass will be removed and the attribute will be displayed as defined by the VizMapper. At the bottom of the menu a *Reset All* option appears. When clicked, this will remove all bypasses for the specified node or edge. In the example above, you can see the selected node size, color, and shape have been overridden. This is apparent in the appearance of the node itself and by the check marks in the popup menu.

It is important to realize that the Visual Mapping Bypass only works for individual nodes and edges and not for all nodes or edges of a specific type. Using the bypass function is not particularly resource intensive, you can use it as much as you like. However, if you find yourself repeating the same bypasses, then you should consider using the VizMapper instead.

The bypass values will persist between sessions only as long as you save your session. If you don't save your session, you will lose whatever bypass values you set.

# Visual Property Dependencies

In some cases, you want to use the same value for multiple Visual Properties. For example, if you want to use the same value for *Node Width* and *Node Height*, you have to use the ***Visual Property Dependency*** feature in VizMapper. Currently, Cytoscape supports the following Visual Property Dependencies:

- **Edge color to arrows** - Assign same color for edge and arrows.

- **Fit Custom Graphics to node** - Adjust the size of Custom Graphics based on *Node Size*.

- **Lock node width and height** - Assign the same value to node width and height.

# Example: Unlocked
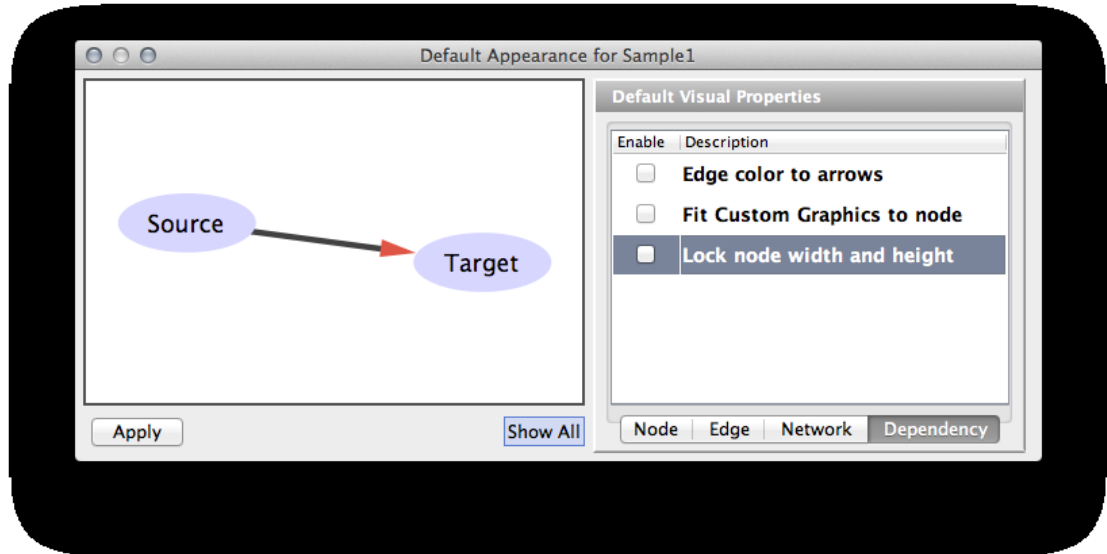


# Example: Locked
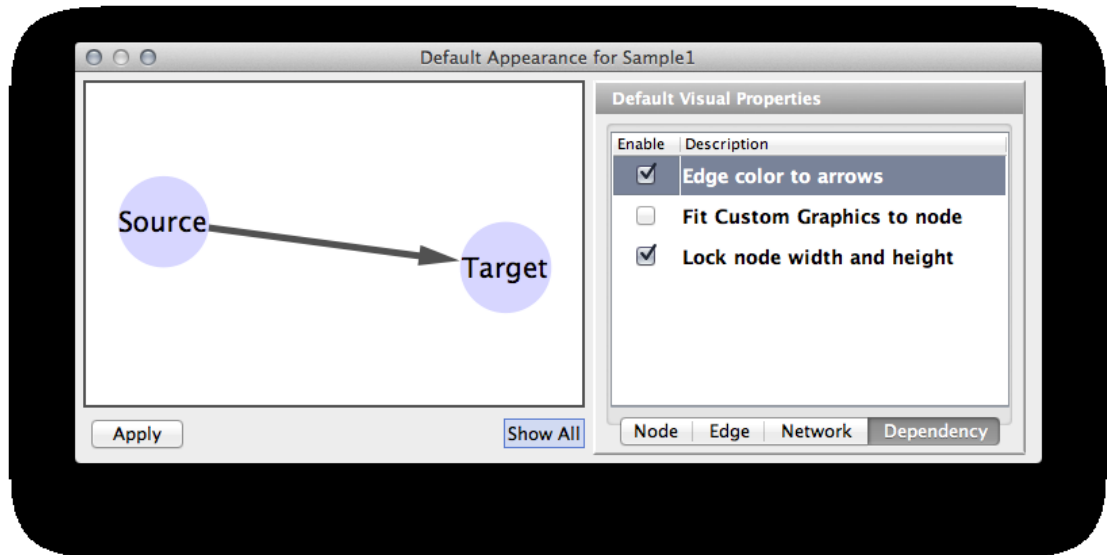
# Finding and Filtering Nodes and Edges

## Enhanced Search

Cytoscape includes a search feature, which enables you to quickly find nodes and edges. The search box is located at the right side in the icon bar and functions as a regular search box.



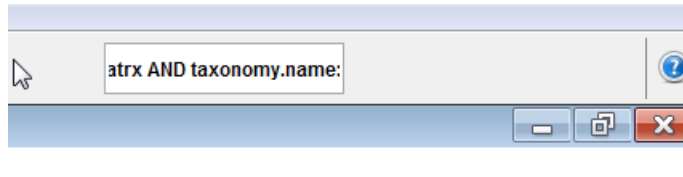For example, type in "ATRX*" in the search box and hit enter. This will find all the nodes/edges with string "ATRX" as prefix in any of the columns. Type in "taxonomy.name:homo*" will restrict the search to column named 'taxonomy.name' only. You can also use 'and', 'or' to make the complex search. The searching is actually based on Apache Lucene. Please look at this page for more about search syntax.

## Filters

Filters allow you to quickly select multiple nodes or edges of interest by comparing node and edge column values to properties you specify. For example, you can select all the nodes whose name contains a specific pattern, or whose numeric column attribute value falls within a certain range. You can also perform complex selection by defining filters and performing the boolean operation between them. Filters are located under the filters tab on the Control Panel to the left. You can also access the filters by using the Select menu pull-down and choose "Use Filters" menu item. The filters panel initially looks like this:
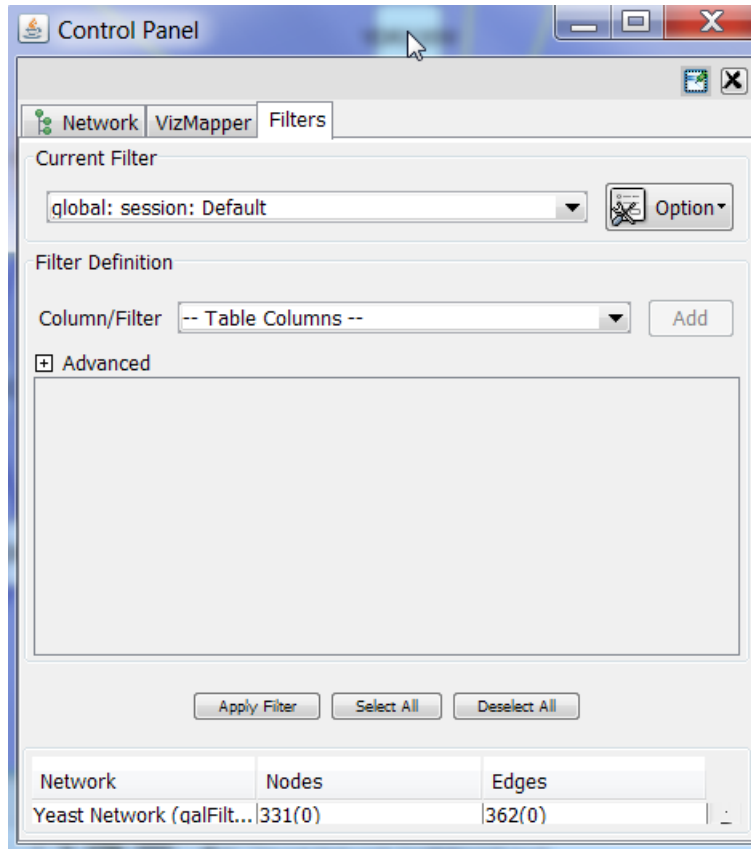
## 1. Filter Definition

To create a new filter, click the option button and select "Create new filter" from the list provided. Enter a name for the new filter.
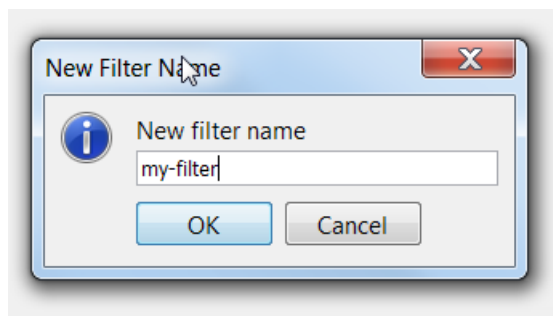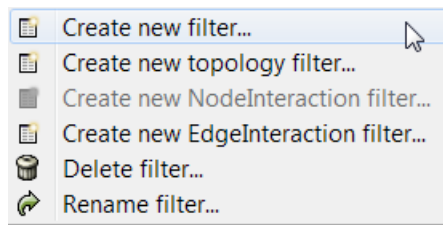




*Definition of simple filter*

When a new filter is created, it is empty initially. It can be defined by choosing columns (one at a time) in the Column/Filter comboBox and clicking the Add button. Note that in the comboBox each column has a prefix, either node. or edge., denoting what type of attribute it is. After add button is clicked, a widget, depended on the item selected, will be 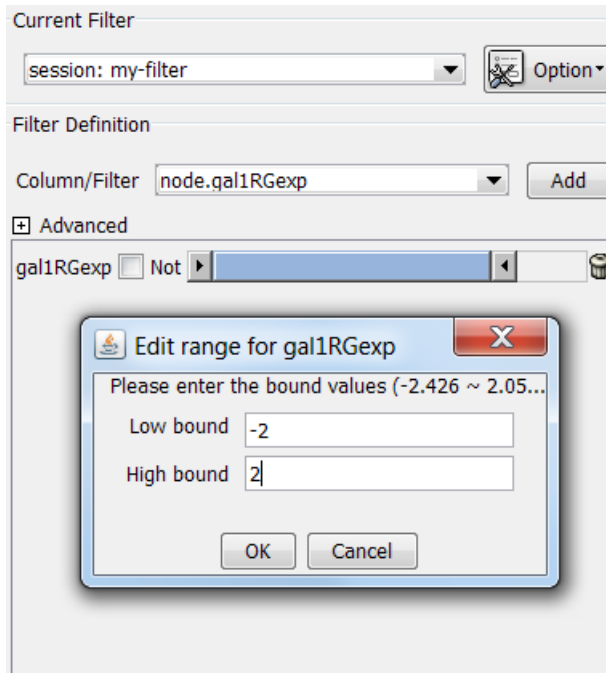added to the definition panel. If the column type is text-based (or String), the widget will be a indexed-text-box widget, if numeric attribute, the widget will be a range slider; clicking the slider enables specific definition of the values.



For each widget, the name of the column it represents is on the left. A NOT checkbox gives the option to get the negation of selection for the widget. There is a trash-can icon on the right. Clicking on the trash-can icon will delete the widget. In this way, the filter definition can be modified after it is defined.

Note that if more than one widget is added on the filter definition panel, the relationship between them is "AND" by default. This relationship can be changed to "OR" by selecting the OR relation in Advanced Panel.

The Advanced panel can be opened by clicking on the plus (+) sign.
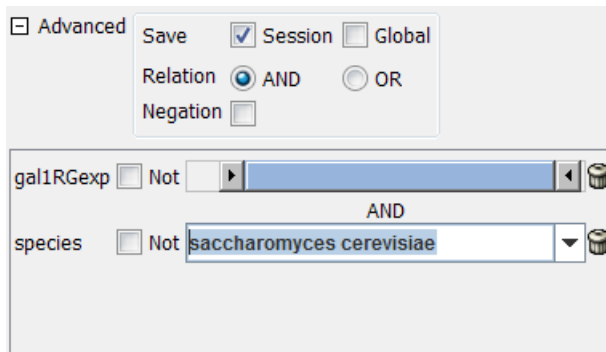


There are three rows in the advanced panel:

1. The first row, labeled Save. The two checkboxes (Global and Session) will determine where the filter is saved. By default, filters are saved in individual sessions. If the Global checkbox is checked, the filter will be saved in the global properties file. Note also that the prefix of the filter name in the Current Filter dropdown list reflects where it is saved.

2. The second row, Relation, will determine what Boolean operation (AND or OR) will be applied to each individual widget.

3. Negation checkbox. If this checkbox is checked, the result of the filter will be negated.

*Definition of complex filter*

In the pull-down list of Column/Filter comboBox, there are two sections, the top one is the list of columns, the bottom one is the list of previously defined filter. Those previously defined filters can also be used as components of other filters. By combinational use of AND, OR, NOT and pre-defined filters, filters with arbitrary complexity can be defined.

**3. Apply the filter**

If a network is small (number of nodes or edges less than 1000), the filter will be applied dynamically when a widget is added or any value is adjusted. If the network is large, then Apply button should be clicked to apply the filter.

**4. Other filters**

In the option menu pulldown, there are menu items "Create new topology filter", "Create new NodeInteraction filter" and "Create new EdgeInteraction filter".

*Topology Filter*

Topology filter will select nodes based on the properties of its near-by nodes (neighbors). To create a topology filter, choose the menu item "Create new topology filter" from the option menu. See below,
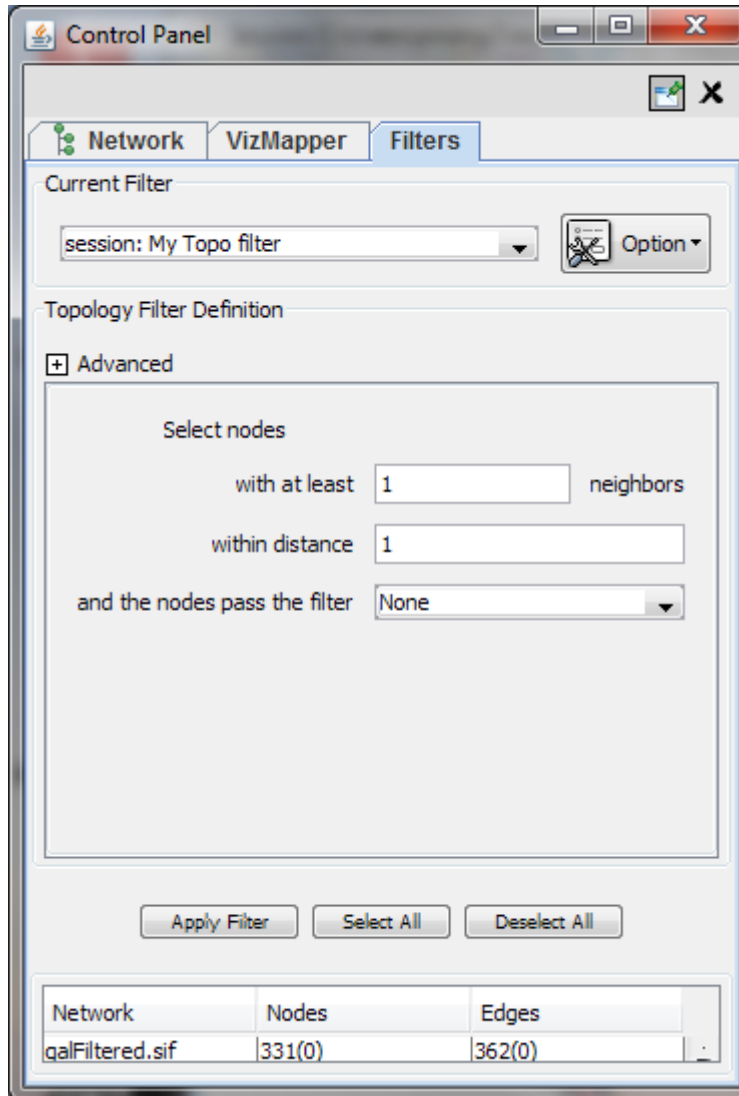
*Interaction filter*

Interaction filters are used to select nodes/edges based on the properties of their neighboring edges/nodes. See below for a edge interaction filter.
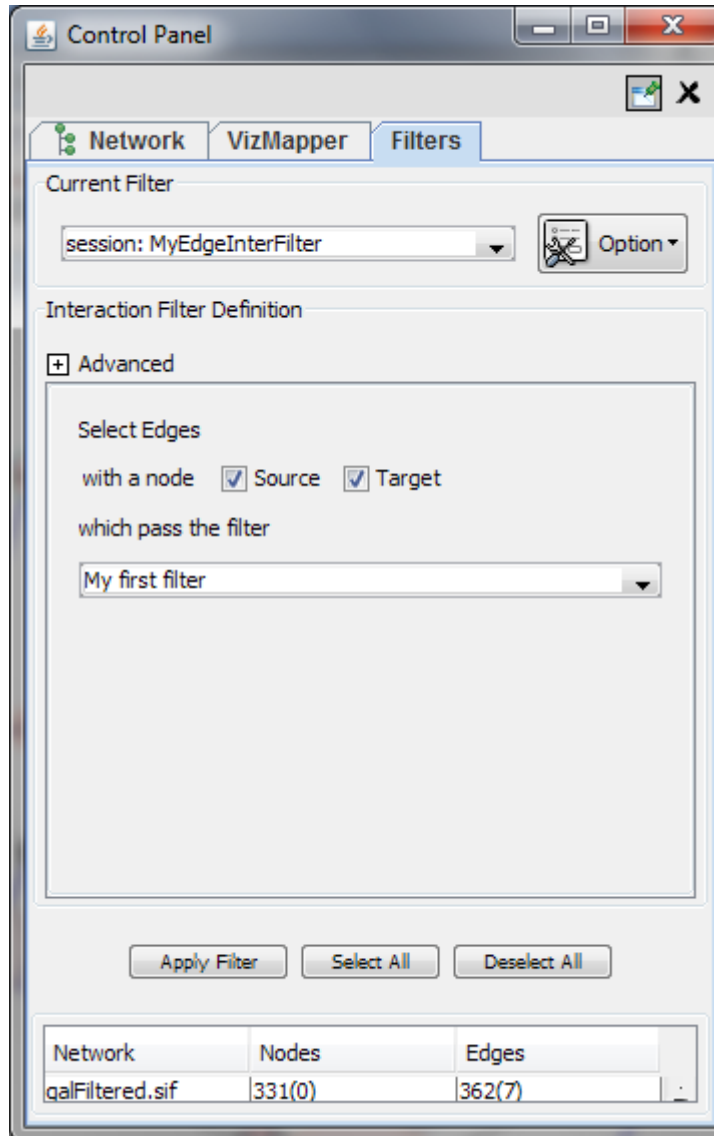
# The Select Menu

The Select → Nodes and Select → Edges menus provide several mechanisms for selecting nodes and edges. Most options are fairly straightforward; however, some need extra explanation.

Select → Nodes → From ID List File... selects nodes based on node identifiers found in a specified file. The file format is simply one node id per line:

```
Node1
Node2
Node3
...
```

# Editing Networks

Using Cytoscape you can edit networks, add nodes and edges. In Cytoscape 3.x editing can be done using right click menus on the network view panel. To start editing a new network, create a new network by going to File → New → Network → Empty Network.

## Adding Node

To add a new node, **right click** on an empty space of the network view panel. Select ***Add → Node*** item from the poped up menu. Image below shows the right click menu for creating a new node.
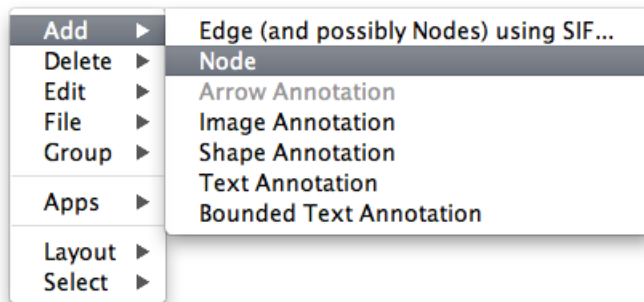


## Adding Edge

For adding an edge to connect to nodes, you need to ***right click on the source node***. Select ***Edit → Add Edge*** from the popped up menu. Next, move pointer and click on the target node. Images below show the two steps for drawing an edge between two nodes. You can abort the drawing of the edge by pressing Esc key.
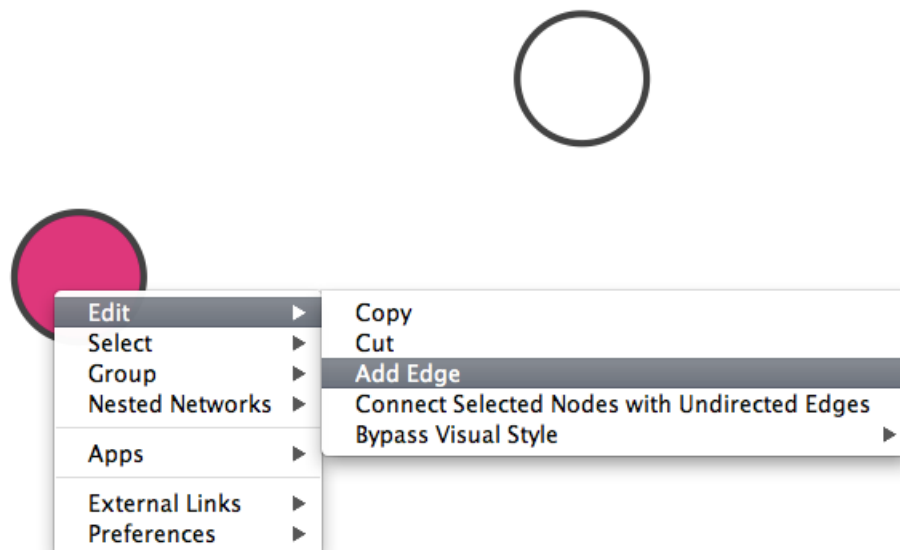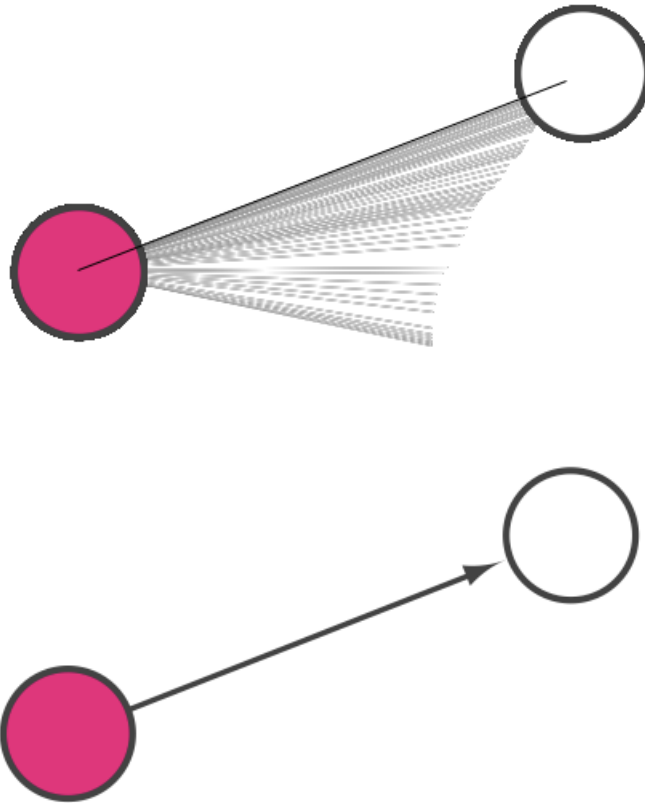
You can delete nodes and edges by selecting a number of nodes and edges, then selecting Edit → Delete Selected Nodes and Edges. You can recover any nodes and edges deleted from a network by going to Edit → Undo.

# CytoPanels

## What are CytoPanels?

*CytoPanels* are floatable/dockable panels designed to cut down on the number of pop-up windows within Cytoscape and to create a more unified user experience. They are named based on their functions -- *Control panel*, *Table panel*, *Tool panel* and *Result panel*. The following screenshot shows the file `gal-Filtered.sif` loaded into Cytoscape, performed Force-Directed layout, enable Align and Distribute tools, and then run Tools -> Network Analysis -> Analyze Network for the network. In *Control Panel* (at the left-hand side of the screen), the Network Manager, Network Overview, VizMapper and Filters have been loaded. On the bottom of the panel, there is another CytoPanel called *Tool Panel*. In the *Table Panel*, the Attribute Browser has been loaded. In addition, Result of the analysis is shown in *Result Panel* (at the right-hand side).
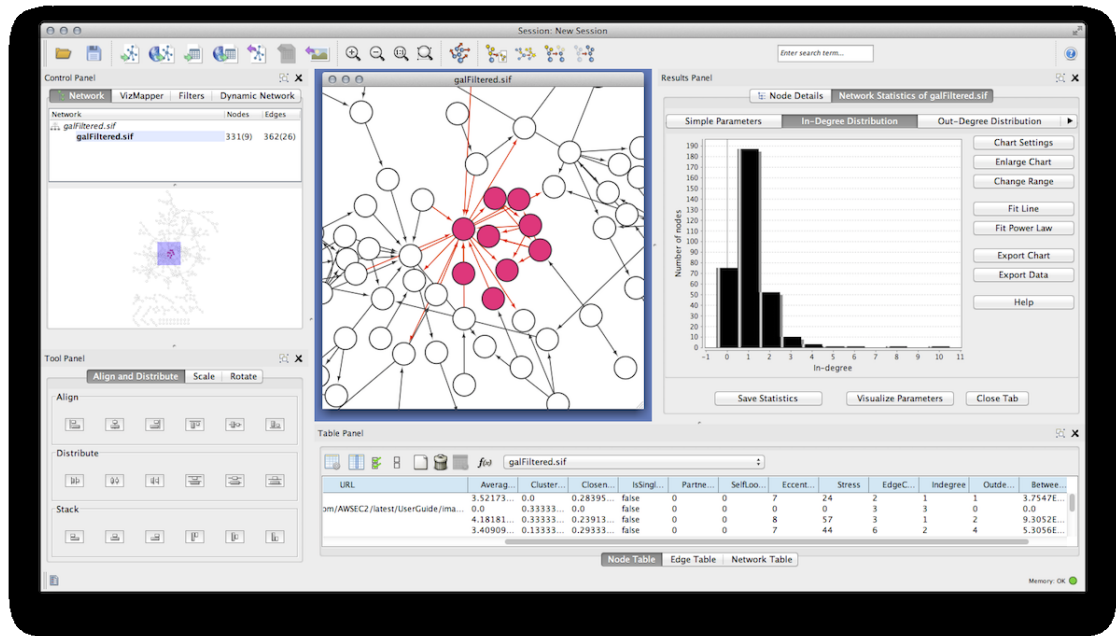
The user can then choose to resize, hide or float CytoPanels. For example, in the screenshot below, the user has chosen to float all panels and toolbar:
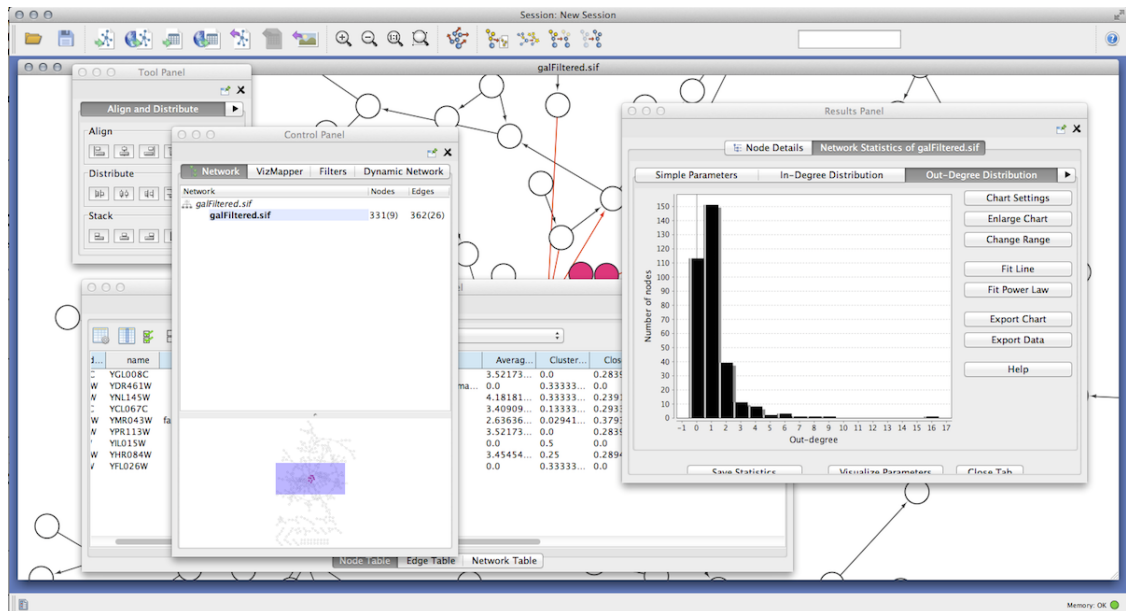


# Basic Usage

Cytoscape includes four CytoPanels: **Control Panel** on the left, **Tool Panel** on the bottom of *Control Panel*, **Table Panel** on the bottom, and **Result Panel** on the right. By default, *Control Panel* and *Data Panel* will appear. *Result Panel* may appear, depending on the mix of Cytoscape apps that you currently have installed. *Tool Panel* will appear when you select the following commands under the *Layout* menu: *Rotate*, *Scale*, and *Align and Distribute*.

All panels can be shown or hidden using the View → Show/Hide functions.
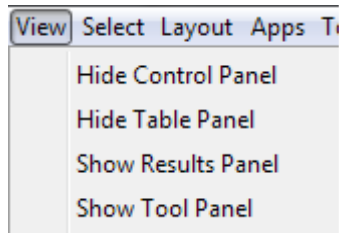
In addition, CytoPanels can be floated or docked by selecting the icon at the top right corner of each CytoPanel. The icon and tooltip will change based on the CytoPanel state. If the CytoPanel is docked, clicking on the icon will float the CytoPanel, as indicated by the "Float Window" tooltip. Alternatively, if the CytoPanel is floating, clicking on the icon will dock the CytoPanel, as indicated by the "Dock Window" tooltip.

# Rendering Engine

## What is Level of Detail (LOD)?

Cytoscape 3.0 retains the rendering engine found in version 2.8. It is to be able to display large networks (>10,000 nodes), yet retain interactive speed. To accomplish this goal, a technique involving *level of detail (LOD)* is being used. Based on the number of objects (nodes and edges) being rendered, an appropriate *level of detail* is chosen. For example, by default, node labels (if present) are only rendered when less than 200 nodes are visible because drawing text is a relatively expensive operation. This can create some unusual behavior. If the screen currently contains 198 nodes, node labels will be displayed. If you pan across the network, such that now 201 nodes are displayed, the node labels will disappear. As another example, if the sum of rendered edges and rendered nodes is greater than or equal to a default value of 4000, a very coarse level of detail is chosen, where edges are always straight lines, nodes are always rectangles, and no antialiasing is done. The default values used to determine these thresholds can be changed by setting properties under Edit → Preferences → Properties... .

**Low LOD vs High LOD**

| Network with **Low** LOD | Network with **High** LOD |
|---|---|
|  |  |

With low LOD values, all nodes are displayed as square and antialiasing is turned off. With high LOD values, antialiasing is turned on and nodes are displayed as actual shape user specified in the Visual Style.

## Parameters for Controlling LOD

**NOTE:** The greater these thresholds become, the slower performance will become. If you work with small networks (a few hundred nodes), this shouldn't be a problem, but for large networks it will produce noticeable slowing. The various thresholds are described below.

| | |
|---|---|
| render.coarseDetail-Threshold | If the sum of *rendered* nodes and *rendered* edges equals to or exceeds this number, a very coarse level of detail will be chosen and all other detail parameters will be ignored. If the total number of nodes and edges is below this threshold, anti-alias will be turned on; this value defaults to 4000. |
| render.nodeBorder-Threshold | If the number of *rendered* nodes equals to or exceeds this number, node borders will not be rendered; this value defaults to 400. |
| render.nodeLa-belThreshold | If the number of *rendered* nodes equals to or exceeds this number, node labels will not be rendered; this value defaults to 200. |
| render.edgeArrow-Threshold | If the number of *rendered* edges equals to or exceeds this number, edge arrows will not be rendered; this value defaults to 600. |
| render.edgeLa-belThreshold | If the number of *rendered* edges equals to or exceeds this number, edge labels will not be rendered; this value defaults to 200. |

When printing networks or exporting to formats such as PostScript, the highest level of detail is always chosen, regardless of what is currently being displayed on the screen.

## Force to Display Detail

If you want to display every detail of the network regardless of LOD values, you can toggle to full details mode by **View → Show Graphics Details (or CTR+SHIFT+F on Windows/Linux, Command+SHIFT+F for Mac)**. This option forces the display of all graphics details. If the network is large, this option slows down rendering speed. To hide details, select the menu item again **(View → Hide Graphics Details)**.

# Linkout

!Linkout provides a mechanism to link nodes and edges to external web resources within Cytoscape. Right-clicking on a node or edge in Cytoscape opens a popup menu with a list of web links.

The external links are specified in a `linkout.props` file which is internal to Cytoscape. The defaults include a number of links such as Entrez, SGD, iHOP, and Google, as well as a number of species-specific links. In addition to the default links, users can customize the LinkOut menu and add (or remove) links by editing the linkout properties (found under Edit → Preferences → Properties...).

External links are listed as *'key'-'value'* pairs in the `linkout.props` file where *key* specifies the name of the link and *value* is the search URL. The LinkOut menus are organized in a hierarchical structure that is specified in the key. Linkout key terms specific for nodes start with the keyword `nodelinkouturl`, for edges this is `edgelinkouturl`.

For example, the following entry:

`nodelinkouturl.Model Organism DB.SGD (yeast)=http://www.yeastgenome.org/cgi-bin/lo`

places the SGD link under the Model Organism DB submenu. This link will appear in Cytoscape as:
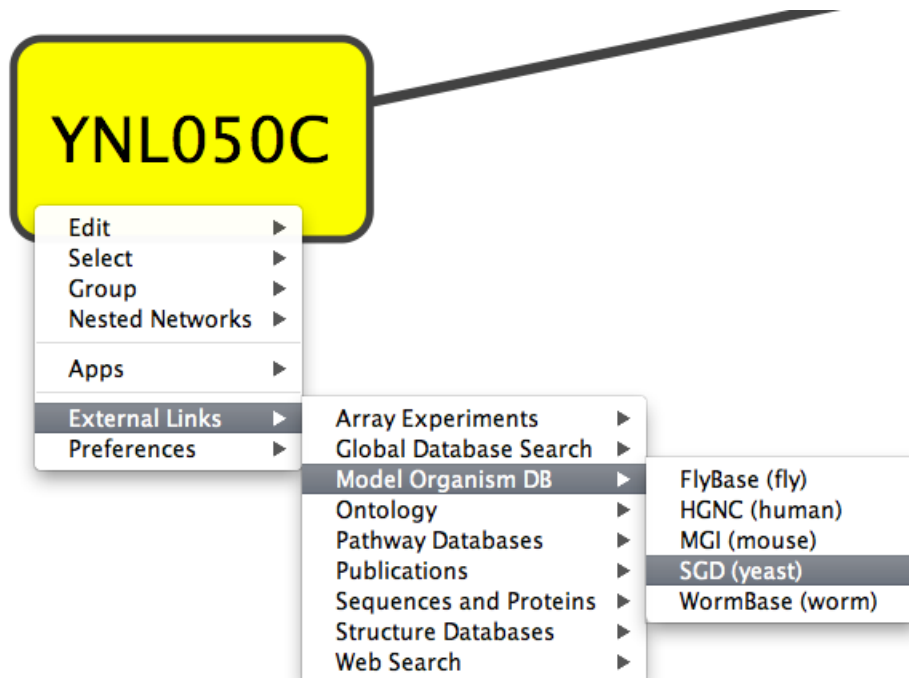
In a similar fashion one can add new submenus.

The `%ID%` string in the URL is a place-holder for the node label. When the popup menu is generated this marker is substituted with the node label. In the above example, the generated SGD link for the YNL050C protein is:

```
http://www.yeastgenome.org/cgi-bin/locus.fpl?locus=YNL050C
```

If you want to query based on a different attribute, you need to specify a different node label using the VizMapper.

For edges the mechanism is much the same; however here the placeholders `%ID1%` and `%ID2%` reflect the source and target node label respectively.

Currently there is no mechanism to check whether the constructed URL query is correct and if the node label is meaningful. Similarly, there is no ID mapping between various identifiers. For example, a link to NCBI Entrez from a network that uses Ensembl gene identifiers as node labels will produce a link to Entrez using the Ensembl ID, which results in an incorrect link. It is the user's responsibility to ensure that the node label that is used as the search term in the URL link will result in a meaningful link.

# Adding or Removing Links

The default links are defined in a `linkout.props` file contained inside the Linkout JAR bundle under the framework/system/org/cytoscape/linkout-impl subdirectory of the Cytoscape installation. These links are normal Java properties and can be edited by going to Edit → Preferences → Properties... and selecting linkout from the box (shown below). Linkouts can be modified, added or removed using this dialog; however, note that the modifications would not be stored in the file. To change a URL permanently, you would need to edit the linkout.props file directly.
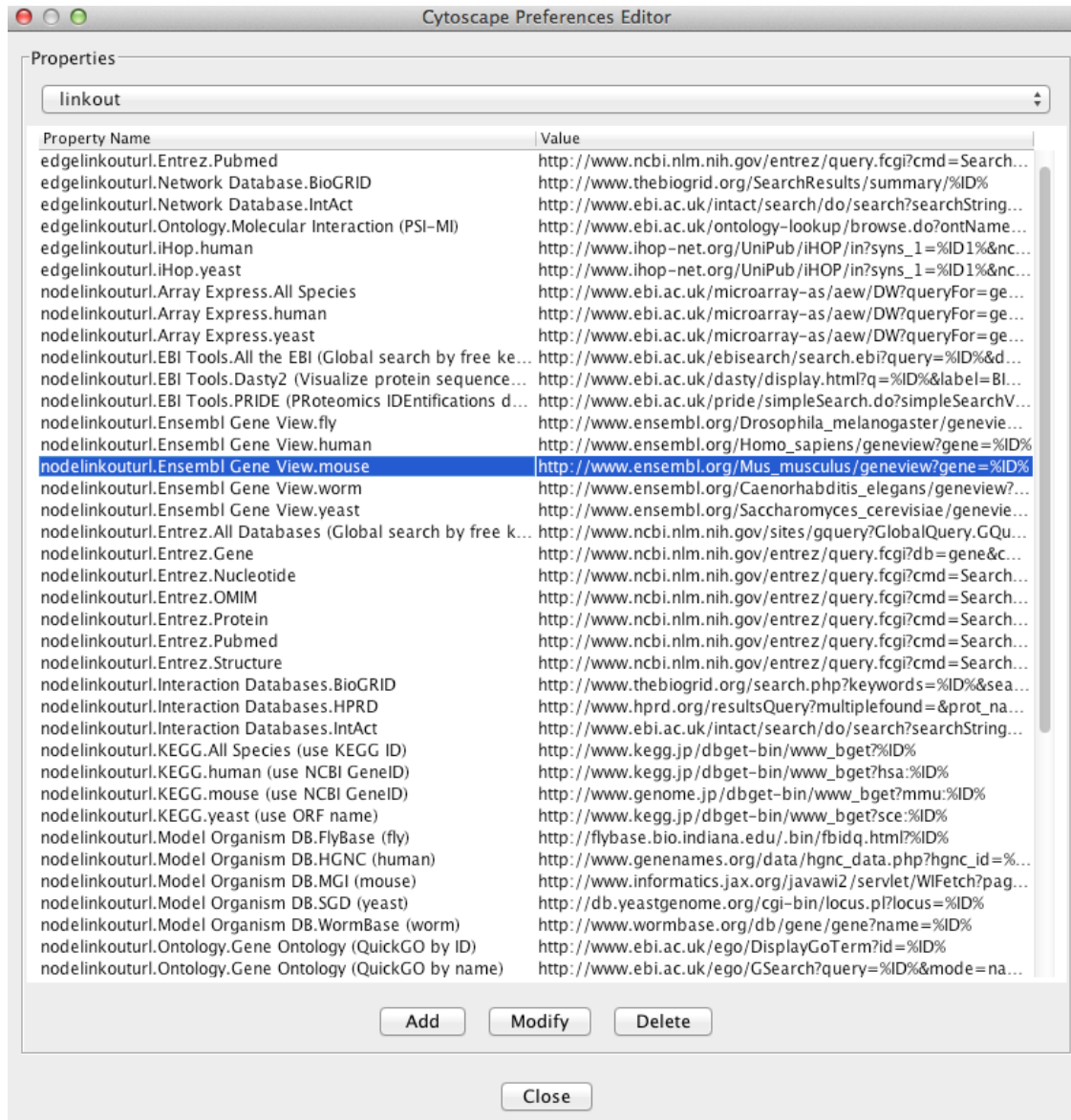
In addition, new links can be defined when starting Cytoscape from command line by specifying individual properties. The formatting of the command is `cytoscape.sh -P [context_menu_defini-tion]=[link]` . *context_menu_definition* specifies the context menu for showing the linkout menu item. The structure of this definition is "." separated and the first item needs to be either *nodelinkouturl* or *edgelinkouturl*. The former will add the linkout item as a node context menu and the latter will add it as an edge context menu. The rest of the definition would define the hierarchy of the menu.

For instance this command:

```
cytoscape.sh -P nodelinkouturl.yeast.SGD=http://db.yeastgenome.org/cgi-bin/locus.p
```

will add this menu item:
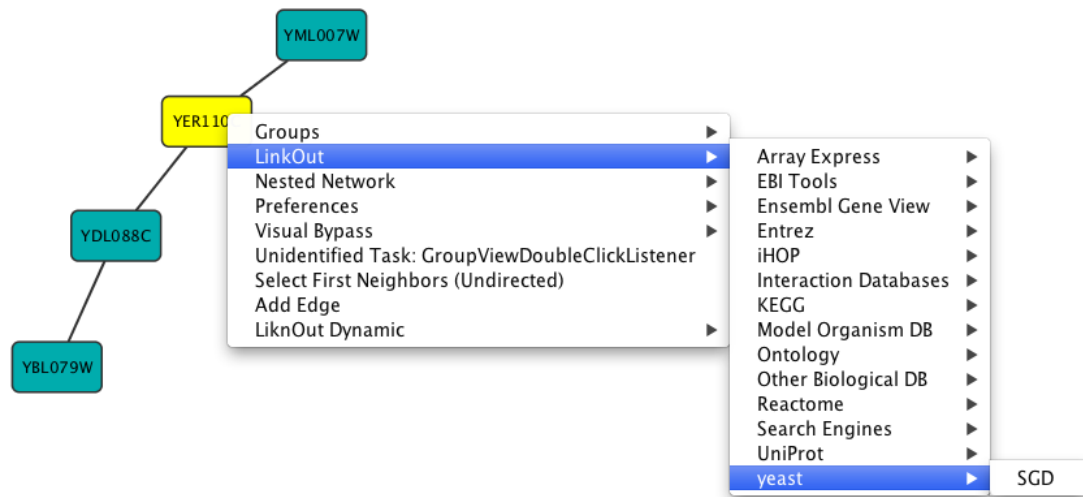
To remove a link from the menu, simply delete the property using Edit → Preferences → Properties... and selecting **commandline**. Linkouts added in the command line will be available for the running instance of Cytoscape.

# App Manager

## What are apps?

Cytoscape's capabilities are not fixed. They can be expanded with *apps*. They can extend Cytoscape in a variety of ways. One app can have the ability to import data from an online database. Another app could provide a new method for analyzing networks. You can install apps after you have installed Cytoscape. Most apps were made by Cytoscape users like you.

If you're familiar with Cytoscape 2.x, you probably know that Cytoscape apps were called *plugins*. Starting in Cytoscape 3.0, we are calling them *apps*. Cytoscape 2.x plugins cannot be used in Cytoscape 3.0.

## Installing Apps

You can install apps through the App Store or within Cytoscape. In this section, we'll talk about installing apps through Cytoscape. You can learn how to install apps through the App Store here.

To install apps within Cytoscape, go to the menu bar and choose **Apps** > **App Manager**. At the top of the App Manager window, make sure you have the **Install** tab selected.

There are three ways you can find apps:

- If you know the name of an app you're looking for, enter it in the **Search** field. The App Manager will list the apps whose names or descriptions match the **Search** field in the middle panel.

- If you have a general idea of what sort of app you're looking for, double-click on the **apps by tag** folder, then click on one of the tags that interests you. The apps with that tag are listed in the middle panel.

- If you're not sure what sort of app you and want to see everything, click the **all apps** folder. In the middle pane, you will see a list of all the apps.

When you click on an app in the middle panel, the App Manager shows its short description and icon in the right panel. If you want more information about the app, click the **View on App Store** button on the bottom-right. If you want to go ahead and install the app, click the **Install** button.

If you've downloaded an app to your computer, you can install it by clicking the **Install from File** button on the bottom-left.

# Managing your Installed Apps

You can see a list of all apps you have installed by clicking the **Currently Installed** tab at the top. When you click on an app in the list, you'll see a description of your app at the bottom. At the bottom, you'll see a couple buttons where you can:

- **Uninstall** an app. This deletes the app from your computer. If you want to reinstall the app, you will have to find it again in the **Install from App Store** tab or in the App Store site and reinstall it there.

- **Disable** an app. This temporarily disables the app. The app stays on your computer, but Cytoscape does not load it. You can enable the app by first selecting the disabled app in the list, then click **Enable**.