
Cytoscape User Manual

Release 3.6.0

The Cytoscape Consortium

Nov 16, 2017

Contents

1	Introduction	3
2	Launching Cytoscape	5
2.1	System Requirements	5
2.2	Getting Started	5
3	Command Line Arguments	9
4	Quick Tour of Cytoscape	11
4.1	Starter Panel	11
4.2	Basic Features	12
4.3	The Menus	14
4.4	Network Management	21
4.5	The View Navigator	24
5	Creating Networks	25
5.1	Import Fixed-Format Network Files	25
5.2	Import Networks from Unformatted Table Files	26
5.3	Import Networks from Public Databases	31
5.4	Create a New Network or Edit One Manually	37
6	Nested Networks	43
6.1	Creating Nested Networks	43
6.2	Visualization of Nested Networks	43
7	Supported Network File Formats	45
7.1	SIF Format	45
7.2	NNF	47
7.3	GML Format	51
7.4	XGMML Format	51
7.5	SBML (Systems Biology Markup Language) Format	52
7.6	BioPAX (Biological PAthways eXchange) Format	52
7.7	PSI-MI Format	52
7.8	GraphML	52
7.9	Delimited Text Table and Excel Workbook	52
7.10	Cytoscape.js JSON	53
7.11	Cytoscape CX	57

8 Node and Edge Column Data	59
8.1 Import Data Table Files	59
8.2 Legacy Cytoscape Attributes Format	61
8.3 Import Data Table from Public Databases	64
8.4 Mapping Identifiers	66
9 Column Data Functions and Equations	69
9.1 Column Formulas	69
9.2 The Formula Builder	72
9.3 A Note for App Writers	73
10 Finding and Filtering Nodes and Edges	75
10.1 Search Bar	75
10.2 Filters	76
10.3 Diffusion	84
10.4 The Select Menu	85
11 Navigation and Layout	87
11.1 Basic Network Navigation	87
11.2 Other Mouse Behaviors	88
11.3 Manual Layout	89
11.4 Node Layout Tools	89
11.5 Edge Bend and Automatic Edge Bundling	92
11.6 Automatic Layout Algorithms	94
11.7 yFiles Layouts	103
11.8 Layout Parameters	106
12 Styles	109
12.1 What are Styles?	109
12.2 Introduction to the Style Interface	116
12.3 Introduction to Style	118
12.4 Images, Charts and Gradients	121
12.5 Styles Tutorials	128
12.6 Advanced Topics	156
12.7 Managing Styles	162
13 App Manager	165
13.1 What are Apps?	165
13.2 Installing Apps	165
13.3 Managing your Installed Apps	167
14 Command Tool	169
15 Merge	171
15.1 Merge Networks	171
15.2 Merge Tables	172
16 NetworkAnalyzer	173
16.1 Network Analysis	173
16.2 Subnetwork Creation	177
16.3 NetworkAnalyzerDemo: Computation and Visualization of Topological Parameters and Centrality Measures for Biological Networks	178
17 Cytoscape Preferences	181
17.1 Managing Properties	181

17.2	Managing Bookmarks	183
17.3	Managing Proxy Servers	184
17.4	Managing Group Settings	185
17.5	Managing OpenCL Settings	187
18	Linkout	189
18.1	Adding and Removing Links	190
19	Panels	193
19.1	Basic Usage	195
20	Rendering Engine	197
20.1	What is Level of Detail (LOD)?	197
21	Export Your Data	199
21.1	Raw Data Export	199
21.2	Exporting for Publication	203
22	Cytoscape and OpenCL (Computing on the GPU)	215
22.1	What it is for?	215
22.2	Setup & Configuration	215
22.3	OpenCL/GPU Troubleshooting	216
23	Cytoscape.js and Cytoscape	217
23.1	What is Cytoscape.js?	217
23.2	Data Exchange between Cytoscape and Cytoscape.js	220
23.3	Export Styles to Cytoscape.js	221
23.4	Build Your Own Web Application with Cytoscape.js	221
24	Cytoscape Automation	223
24.1	Background	223
24.2	Commands	225
24.3	CyREST Interface Layer	225
24.4	cyREST and R/Python	229
25	Cytoscape Privacy Policy	231
26	Basic Expression Analysis Tutorial	233
26.1	Loading Network	233
26.2	Loading Expression Data	234
26.3	Visualizing Expression Data	236

The Cytoscape User Manual copyright is owned by The Cytoscape Consortium, and is made available under the same GPL license as Cytoscape itself: LGPL 2.1, the GNU Lesser General Public License, version 2.1, February 1999 available in text at <http://www.gnu.org/licenses/lgpl-2.1.html>.

Copyright (c) 2001-2017 The Cytoscape Consortium

Table of Contents

CHAPTER 1

Introduction

This version of Cytoscape builds upon the new 3.x architecture, developer API and set of user controls established. If you're familiar with former versions of Cytoscape, this version will feel completely familiar and you'll be all set to go. 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.

If you would prefer to learn by doing, consider starting with the [Basic Expression Analysis Tutorial](#) chapter.

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.

CHAPTER 2

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 8 is available for the platform.

2.1 System Requirements

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

Note that as of Cytoscape v3.2, networks are loaded faster and in less memory than with previous versions. While this is good news, networks created on v3.2 on a given memory configuration (e.g., 1GB) may not be loadable by prior Cytoscape versions on the same memory configuration.

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

2.2 Getting Started

2.2.1 Install Java

Cytoscape requires Java 8.

- While Cytoscape versions prior to v3.2 run on Java 6, Oracle and other JVM suppliers have dropped Java 6 support. Consequently, Cytoscape v3.2 and later don't support Java 6 either. With v3.3, we have also dropped support for Java 7 for the same reason.
- We recommend a 64 bit Java Runtime Environment (JRE). While Cytoscape runs with 32 bit Java versions, using a 64 bit Java allows the largest networks to be loaded and enables the fastest network processing. For Windows, the default JRE download provided at java.com is 32 bits regardless of the Windows version. While Cytoscape will run with a 32 bit JRE, it will be limited to loading only small networks. We recommend downloading and installing a 64 bit JRE.

- We currently recommend only Java 8.

For additional information, select the Release Notes button on the Cytoscape web site.

2.2.2 Install Cytoscape

Downloading and Installing

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 – best for most users.
- 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 [Git](#) repository (<https://github.com/cytoscape/cytoscape>).

Unattended Installation

The easiest and most common way to install Cytoscape is by executing an [automatic installation package](#) downloaded from the Cytoscape web site. This will bring up a wizard that will lead you through the process, presenting choices for the installation directory, license agreement, file associations and privacy settings.

The installation process can be automated and made silent by executing the installation package with the “-q” command parameter (e.g., “Cytoscape_3_6_0-RC1_windows_64bit.exe -q”) from a command line or script. For this to succeed, your execution environment must already have sufficient privileges to install software (e.g., for Windows: administrator privileges). With a “-q” parameter, the installation package will automatically choose all default settings.

More automation flexibility is available using other settings and pre-programmed response files, as described in Appendix A of the [Install4j](#) manual (<http://resources.ej-technologies.com/install4j/help/doc/help.pdf>).

Cytoscape Directories

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

The *p/* directory signifies the program directory, which varies from platform to platform. For Cytoscape 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.

The *u/* directory signifies the user’s home directory, which varies from user to user and from platform to platform. To change the user home directory from the default, one can set the Java environment variable *user.home* to the desired directory – this is useful when Cytoscape is installed on a workstation, but the home directory is stored on a central file server. *user.home* can be set by adding the following option to the *Cytoscape.vmoptions* file or the *_JAVA_OPTIONS* environment variable, substituting the desired path as appropriate:

-Duser.home=/path/to/desired/home

Your operating system may have other mechanisms for setting environment variables – see your operating system documentation for further details.

A quick note on upgrading your Cytoscape installation

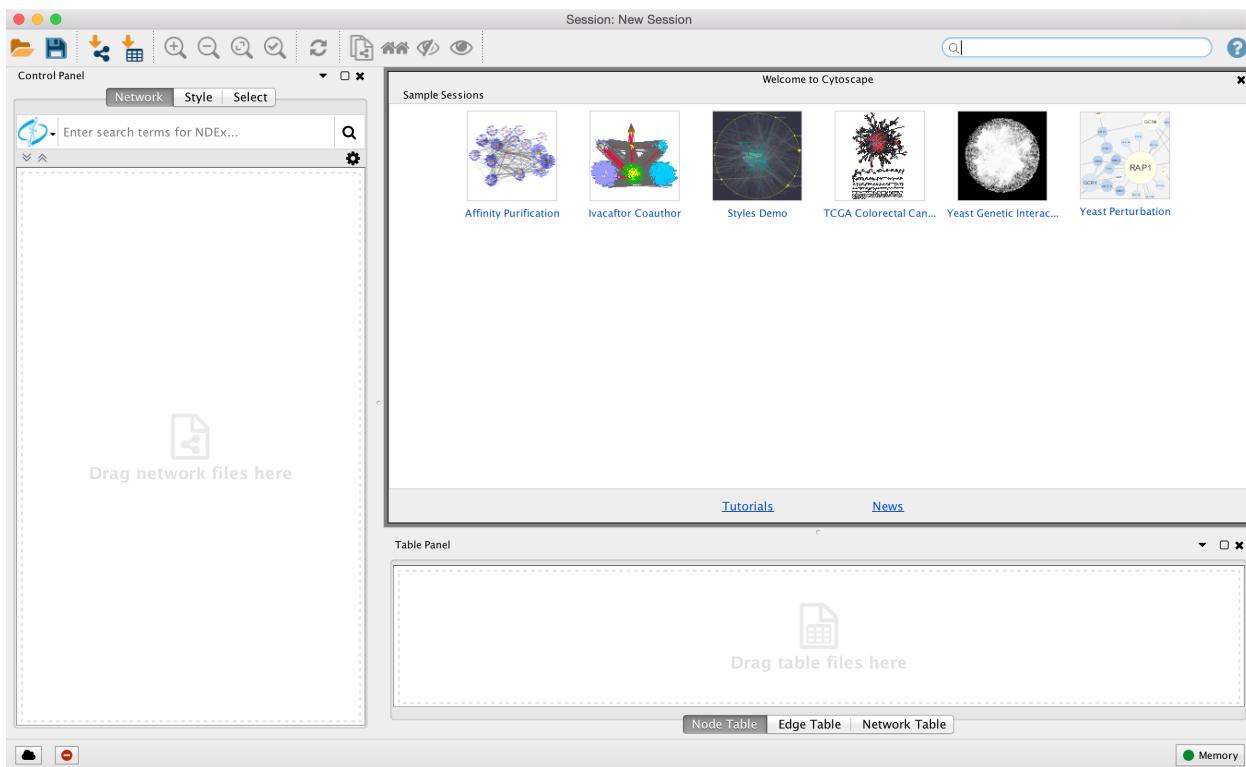
If you have a previous Cytoscape installation you have two options:

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.

2.2.3 Launch the Application

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

After launching Cytoscape a window will appear that looks like this:



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

2.2.4 Note on Memory Consumption

For most regular users, Cytoscape will estimate and reserve the proper amount of memory. An incorrect estimate may result in Cytoscape hanging at startup or Cytoscape unable to load your network. Unless Cytoscape fails to start or open your network, it has likely estimated the available memory correctly, and you can continue to the [Quick Tour](#). If Cytoscape misjudges the memory size or can't allocate enough memory, it could be that you're running with a 32 bit JRE and could get better results by installing a 64 bit JRE – see the Install Java section above.

When Cytoscape starts, it displays the current memory usage in the lower right corner of the main interface. You can click on the **Memory** button at any time to access an option to **Free Unused Memory**. While most users won't need

to use this option, it can be useful for users who have multiple large networks loaded.

Overall Memory Size for Cytoscape

By default, Cytoscape uses an estimate for initial and maximum memory allocation based on your operating system, system architecture (32 or 64 bit), and installed memory. You can change Cytoscape's initial and/or maximum memory size by editing the Cytoscape.vmoptions file, which resides in the same directory as the Cytoscape executable. The file contains one option per line, with each line terminated by a linefeed, and an extra linefeed at the end of the file. Note that for the MacOS platform, the situation is slightly different – if you are launching Cytoscape by clicking on the Cytoscape icon, you must edit the .../Cytoscape.app/Contents/vmoptions.txt file instead. To access this in Finder, you will need to right-click the Cytoscape app icon and select “Show Package Contents”, which will display the Contents subdirectory that contains vmoptions.txt.

For example, if you want Cytoscape to initially allocate 2GB of memory and use up to a maximum of 4GB, edit the Cytoscape.vmoptions file to contain the following lines (... do not forget the linefeed at the end of each line, and an extra linefeed at the end of the file!):

-Xms2GB

-Xmx4GB

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 values above, sometimes layout algorithms fail due to an *out of memory* error. To avoid this, you can set a larger heap size for Cytoscape tasks by using the **taskStackSize** option in the **cytoscape3.props** file (located in the **CytoscapeConfiguration** directory). This can be edited within Cytoscape using the Preferences Editor (**Edit → Preferences → Properties...***) - look for taskStackSize. The value should be specified in bytes.

CHAPTER 3

Command Line Arguments

Cytoscape recognizes a number of optional command line arguments, including run-time specification of network files, node and edge data files, and session files. This is the output generated when 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).
-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).
-S, --script <file>          Execute commands from script file.
-R, --rest <port>            Start a rest service.
```

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`.

Note: if there are spaces in the file path, be sure to put quotes around it: `cytoscape.bat -N "C:\Program Files\Cytoscape\sampleData\galFiltered.sif"`.

Or you can specify a network as a URL: `cytoscape.sh -N http://example.com/myNet.sif`.

All options described above (except for starting a REST service) can be accessed from the menu once Cytoscape is running.

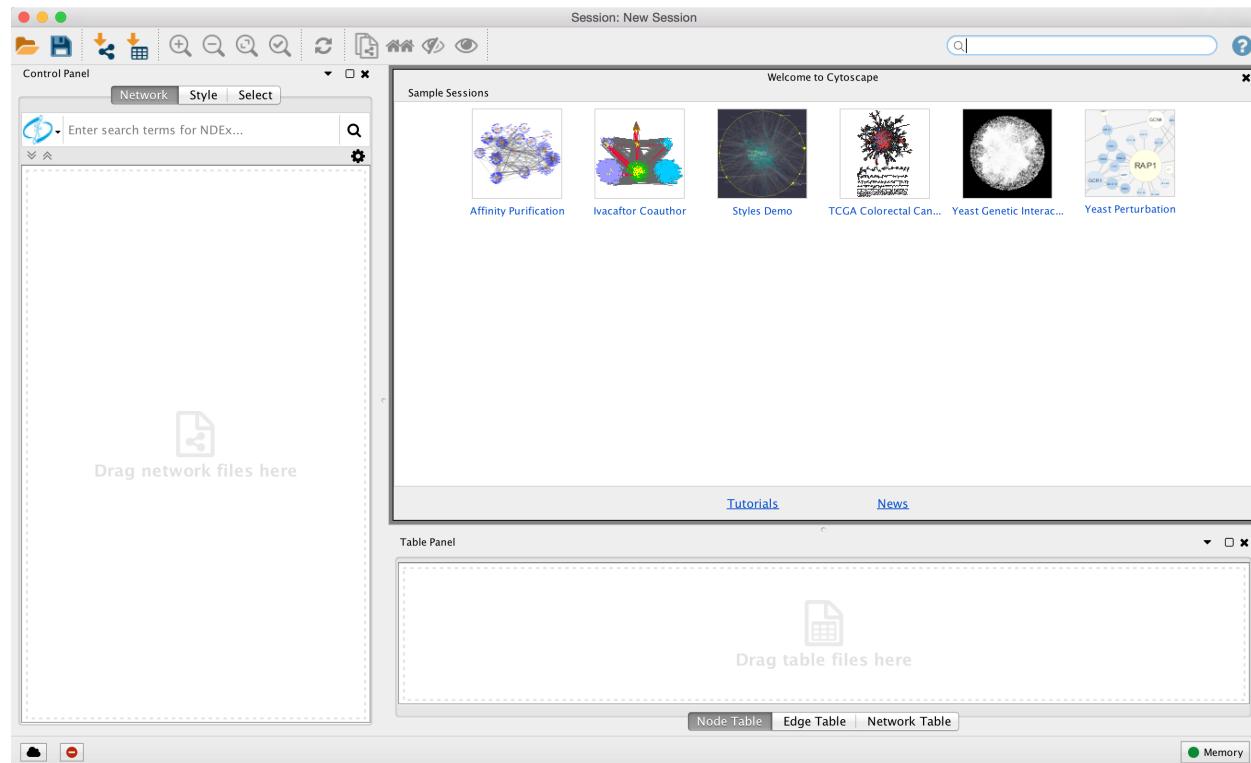
CHAPTER 4

Quick Tour of Cytoscape

This chapter describes the basic layout and mechanics of using Cytoscape. If you would prefer to learn by doing, consider starting with the [Basic Expression Analysis Tutorial](#) chapter.

4.1 Starter Panel

When you start Cytoscape, you can access basic functions from the **Starter Panel**:



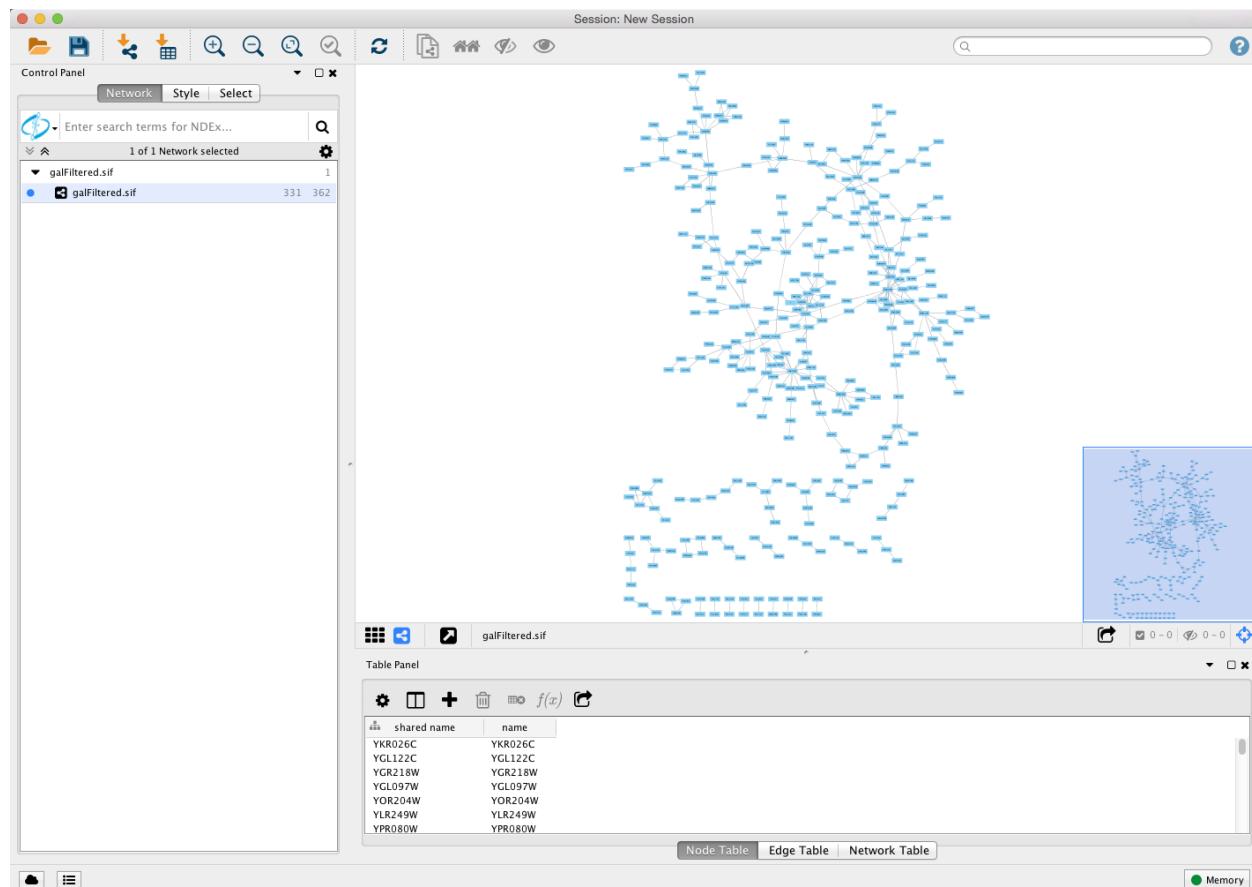
The **Starter Panel** is designed to give you quick access to a set of sample session files, as well as tutorials and Cytoscape news. The sample session files include a broad range of networks to give a sense of the diversity of interaction types, visualization styles and biological applications (see Network Table properties for more information about each sample session network). The **Starter Panel** will also present your own most recent sessions for quick access.

When you load a session file, the **Starter Panel** will disappear. You can display it at any time by choosing **View → Show Starter Panel**.

For information on user privacy, see the [Cytoscape Privacy Policy](#).

4.2 Basic Features

When a network is loaded, Cytoscape will look similar to the image below:



Most functionalities are self-explanatory, but we'll go through a concise 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 Tool Bar, 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. Right-clicking on the tool bar allows the contents to be customized.
- The Network Panel (Network tab of the Control Panel, top left). At the top of the Network Panel is a Search bar, with direct access to multiple external resources. The network panel also contains an optional network overview pane (shown at the bottom left).

- The main Network View Window, which displays the network.
- The Table Panel (bottom right panel), which displays columns of selected nodes and edges and enables you to modify the values of column data.

The Network Panel and Table Panel are dockable tabbed Panels. 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

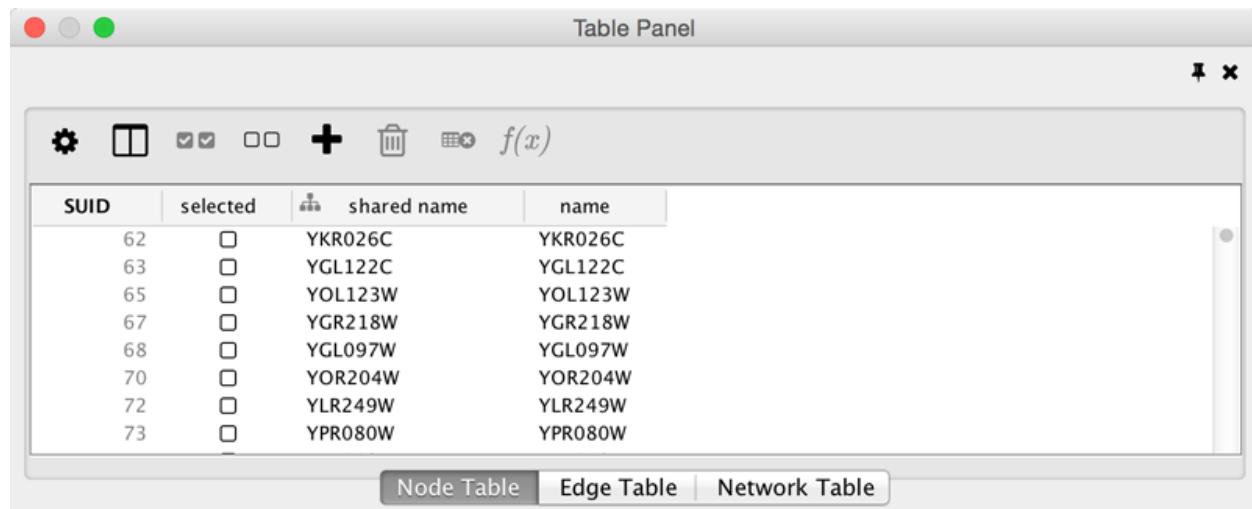


the network panel as much screen space as possible. To dock the window again, click the **Dock Window** icon



Clicking the **Hide Panel** icon will hide the panel; this can be shown again by choosing **View → Show** and selecting the relevant panel.

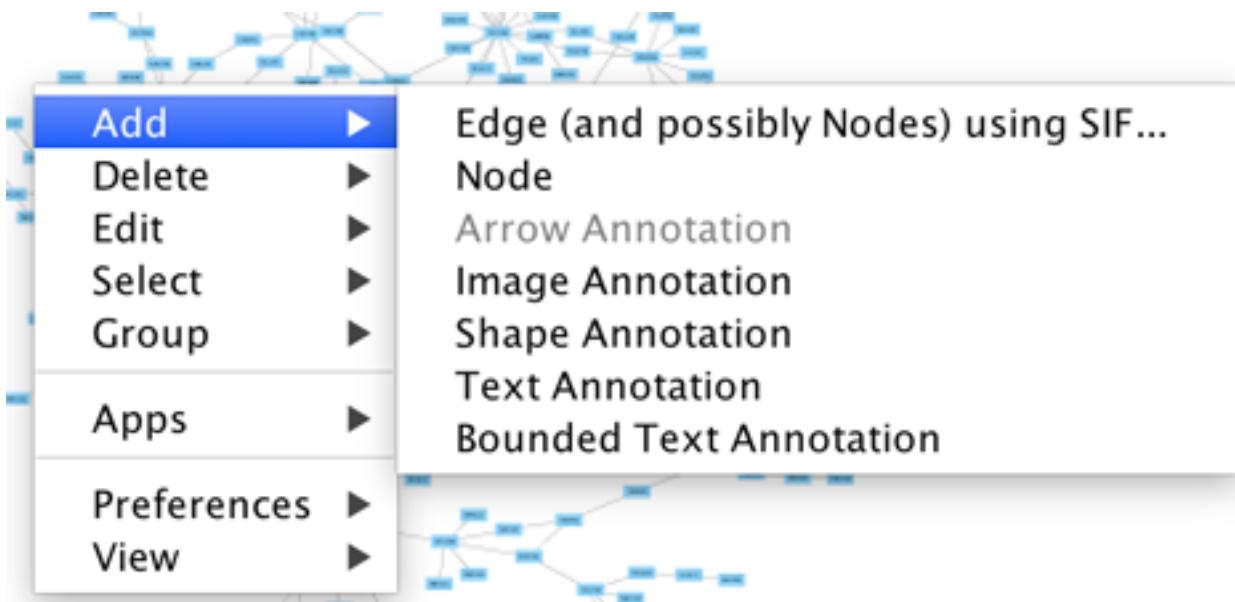
If you click this control, for example on the Table Panel, you will now have two Cytoscape windows, the main window, and a new window labeled Table Panel, similar to the one shown below. A popup will be displayed when you put the mouse pointer on a cell.



Note that Table Panel now has a Dock Window control. If you click this control, the window will dock onto the main window. For more information on the panels in Cytoscape, see the **Panels** section.

4.2.1 Network Editing

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. Detailed information on network editing can be found in the **Editing Networks** section.

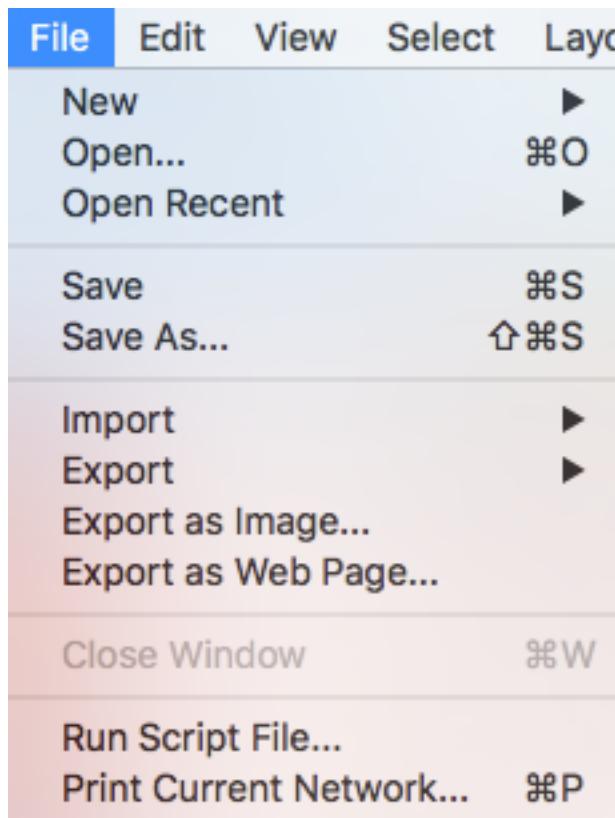


4.3 The Menus

4.3.1 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 tables; and **File** → **Export** for exporting data. **File** → **Export as Image** lets you export the network in either JPEG, PDF, PNG, PostScript or SVG format. **File** → **Export as Web Page** lets you export the entire session or a single network as a Web page using Cytoscape.js.

File → **Recent Session** will list recently opened session files for quick access. **File** → **Run** allows you to specify a Cytoscape script file to run, and **File** → **Print Current Network...** allows printing.



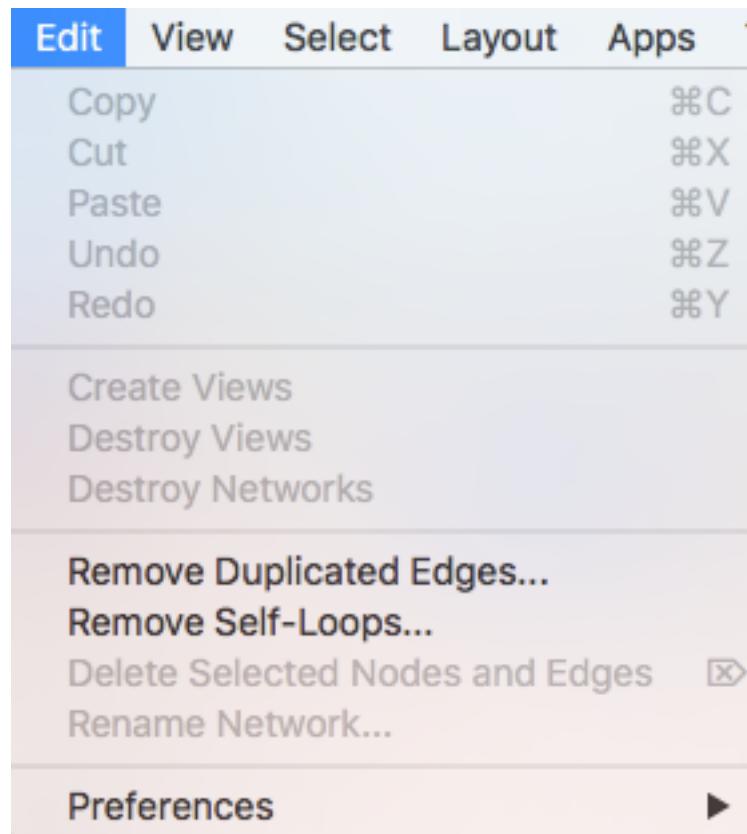
4.3.2 Edit

The **Edit** menu contains **Cut**, **Copy** and **Paste** functions, as well as **Undo** and **Redo** functions which undo and redo edits made in the **Table Panel**, 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**.

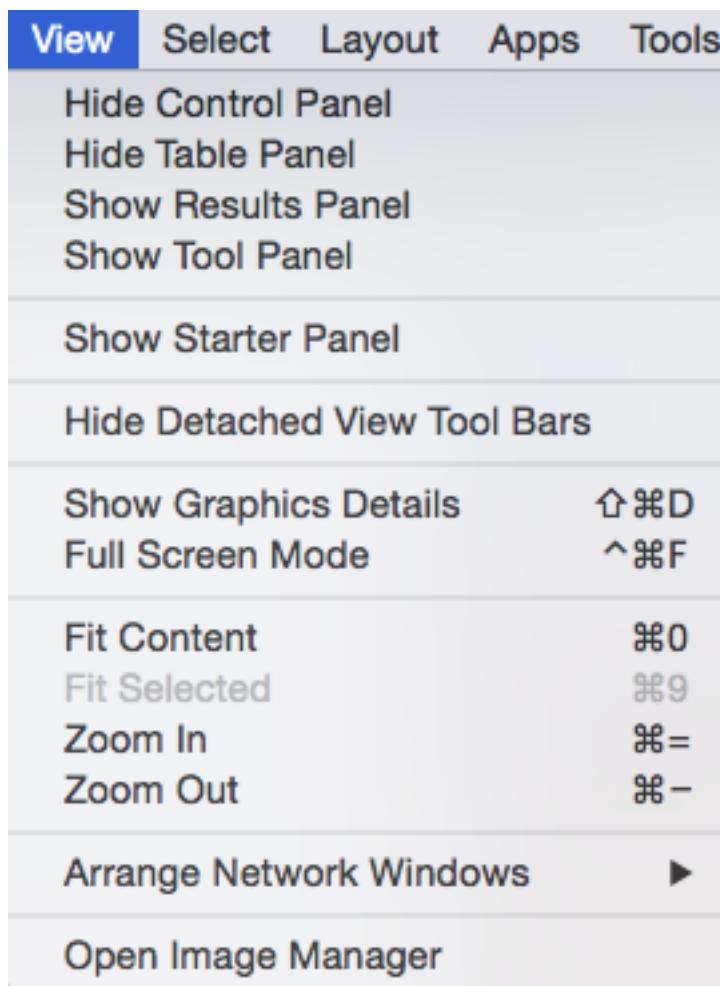
There are also other editing options; **Remove Duplicated Edges** will delete edges that are duplicates (having the same source and target nodes), keeping one edge, **Remove Self-Loops** removes edges that have the same source and target node, and **Delete Selected Nodes and Edges...** deletes a selected subset of the network. **Rename Network...** allows you to rename the currently selected network.

Editing preferences for properties and apps is found under **Edit** → **Preferences** → **Properties...**. More details on how to edit preferences can be found [here](#).



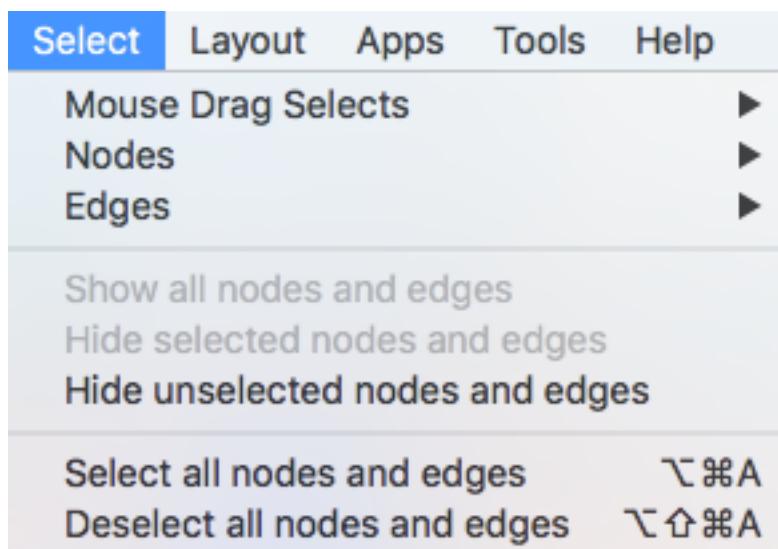
4.3.3 View

The **View** menu allows you to display or hide the **Control Panel**, **Table Panel**, **Tool Panel** and the **Result Panel**. It also provides the control of other view-related functionality.



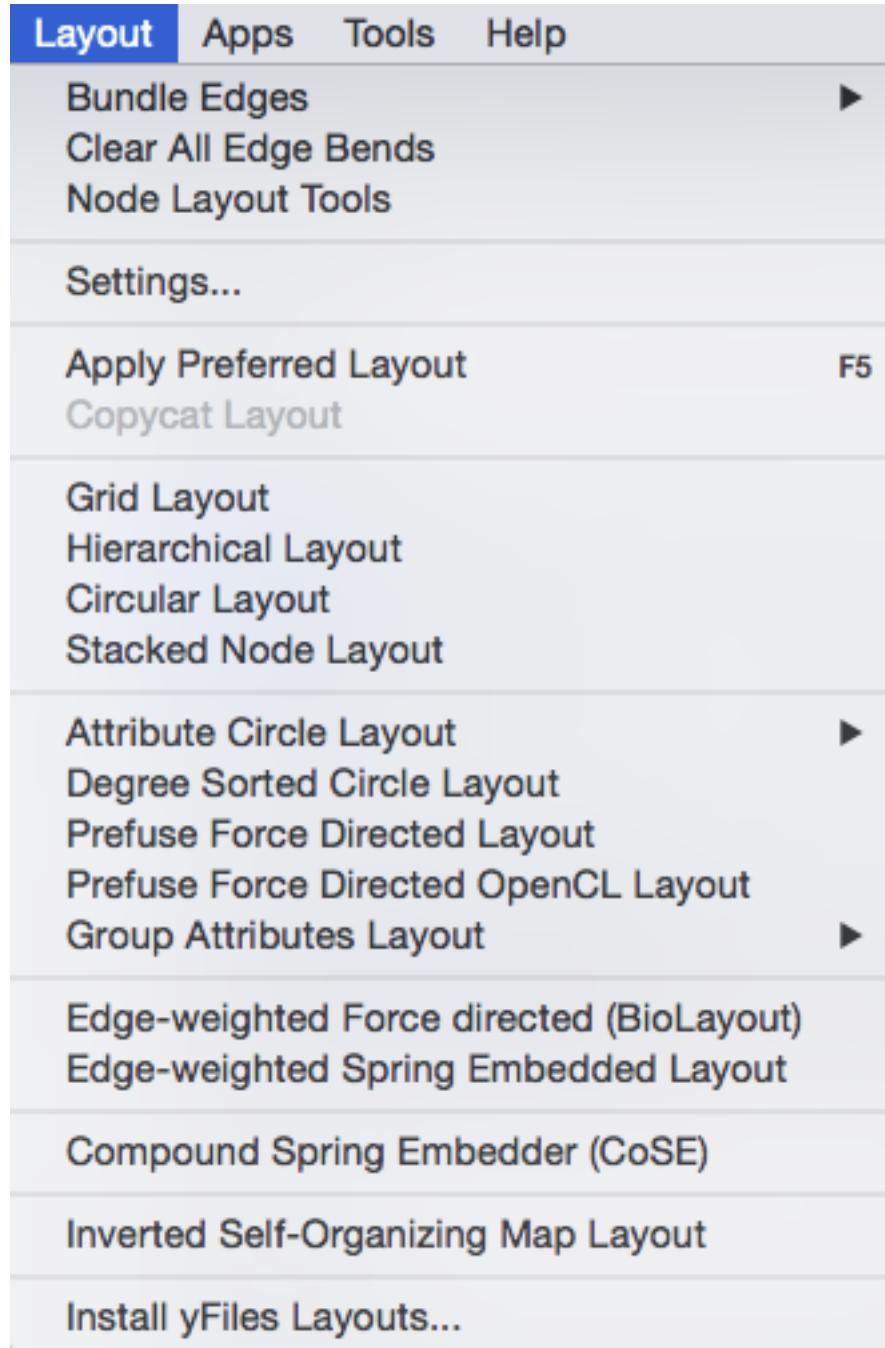
4.3.4 Select

The **Select** menu contains different options for selecting nodes and edges.



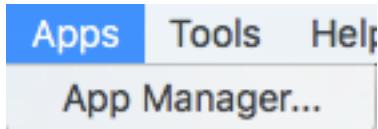
4.3.5 Layout

The **Layout** menu has an array of features for visually organizing the network. The features in the top portion of the network (**Bundle Edges**, **Clear Edge Bends**, **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.



4.3.6 Apps

The **Apps** menu gives you access to the **App Manager** (**Apps** → **App Manager**) 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. The below picture shows a Cytoscape installation without installed apps.

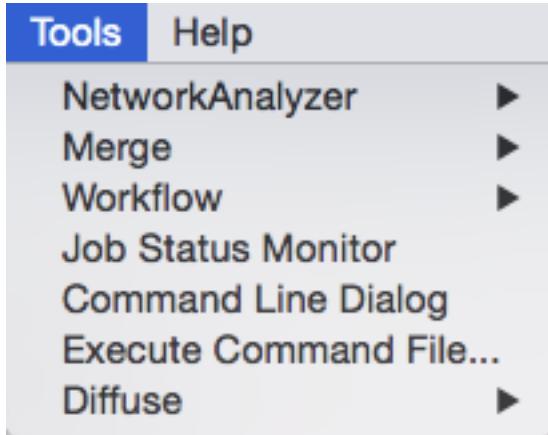


Note: A list of available Cytoscape apps with descriptions is available online at: <http://apps.cytoscape.org>

In previous versions of Cytoscape, apps were called plugins and served a similar function.

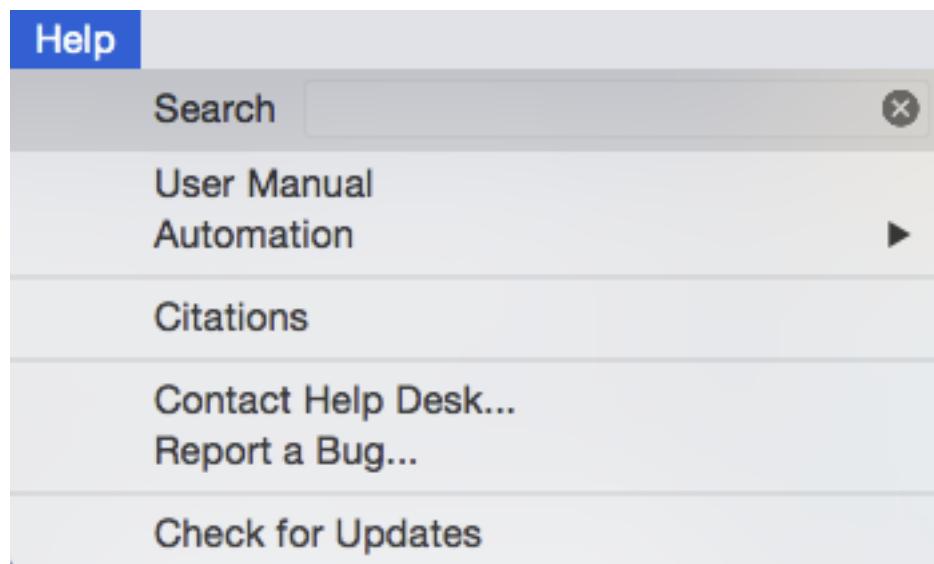
4.3.7 Tools

The **Tools** menu contains advanced features like the **Command Line Dialog**, **Network Analyzer**, **Network Merge** and **Workflow**.



4.3.8 Help

The **Help** menu allows you to launch the online help viewer and browse the table of contents for this manual (**Contents**).



The **Citations** option displays the main literature citation for Cytoscape, as well as a list of literature citations for installed apps. The list will be different depending on the set of apps you have installed.

The 'Citations' section displays the following publications:

- Cytoscape**
Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T. *Cytoscape: a software environment for integrated models of biomolecular interaction networks*. Genome Res, 13:11 (2498-504). 2003 Nov. PubMed ID: 14597658.
[Open in PubMed →](#)
- GeneMANIA**
Montojo J, Zuberi K, Rodriguez H, Kazi F, Wright G, Donaldson SL, Morris Q, Bader GD. *GeneMANIA Cytoscape plugin: fast gene function predictions on the desktop*. Bioinformatics, 26:22 (2927-8). 2010 Nov 15. PubMed ID: 20926419.
[Open in PubMed →](#)
- MCODE**
Bader GD, Hogue CW. *An automated method for finding molecular complexes in large protein interaction networks*. BMC Bioinformatics, 4: (2). 2003 Jan 13. PubMed ID: 12525261.
[Open in PubMed →](#)
- WikiPathways**
Kutmon M, Lotia S, Evelo CT, Pico AR. *WikiPathways App for Cytoscape: Making biological pathways amenable to network analysis and visualization*. F1000Res, 3: (152). 2014. PubMed ID: 25254103.
[Open in PubMed →](#)

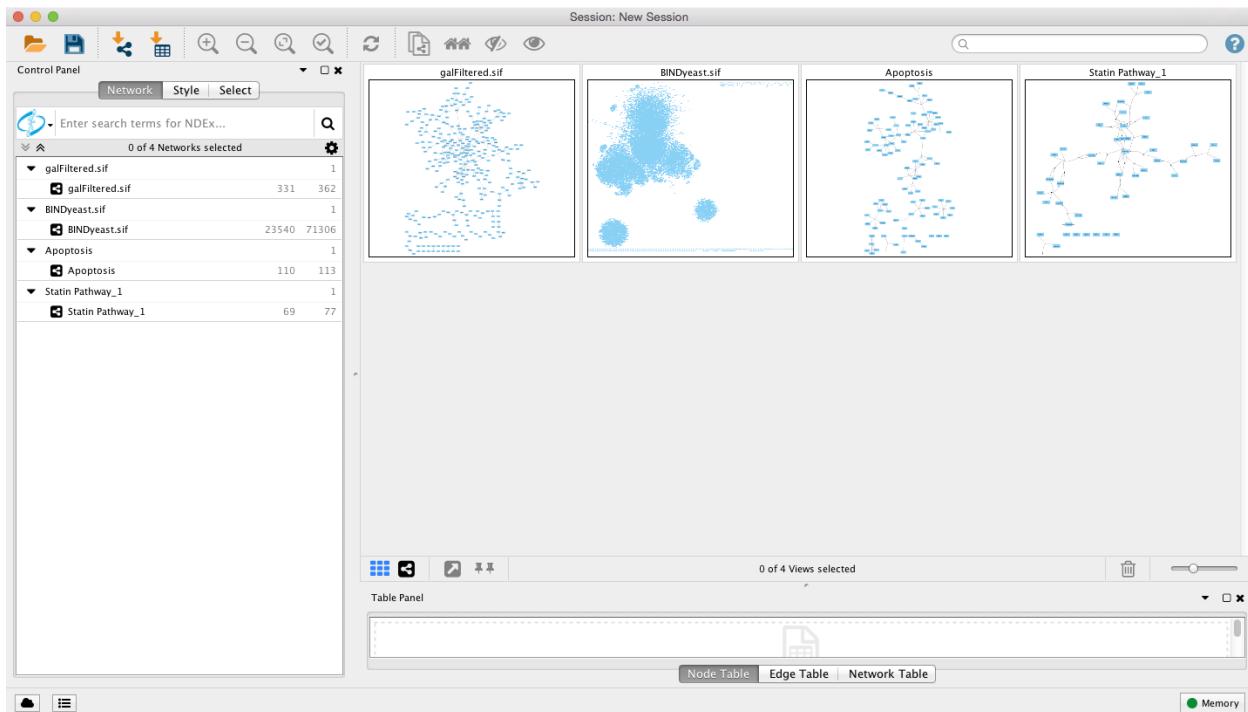
The **Update Organism Preset Networks** option updates the preset networks available on the **Welcome** screen to the latest version.

The **Help** menu also allows you to connect directly to Cytoscape Help Desk and the Bug Report interface.

4.4 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.

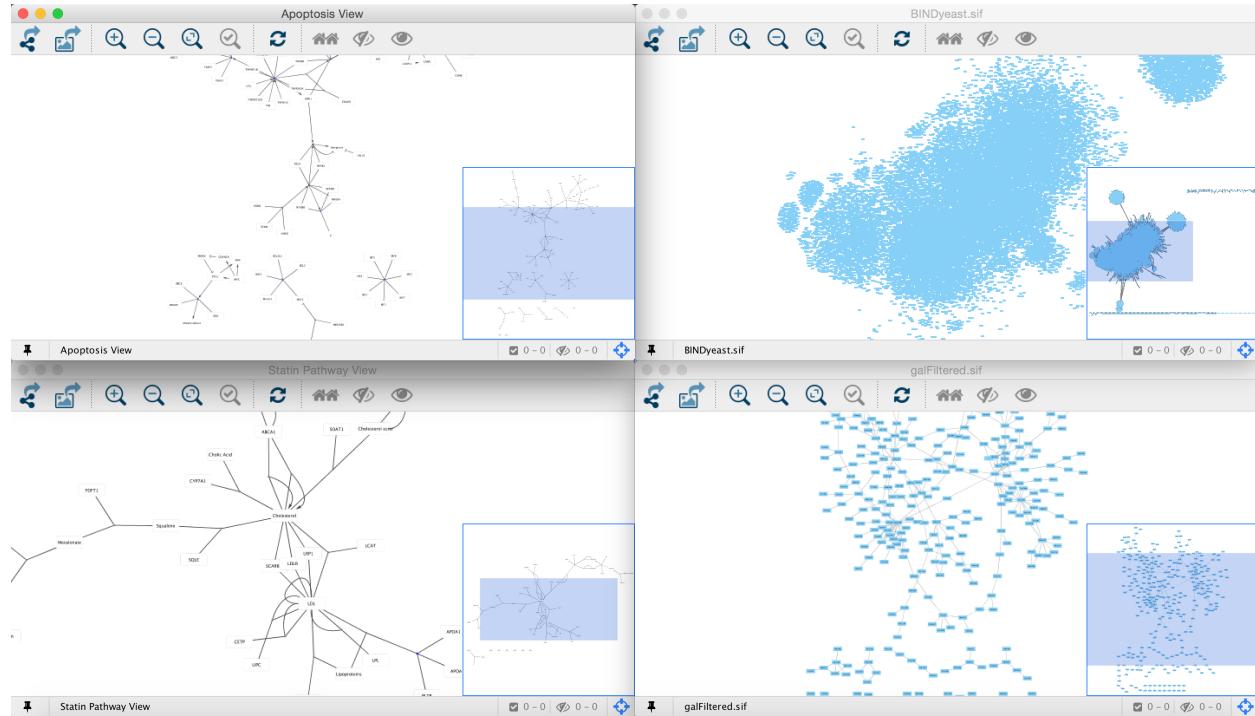
Certain operations in Cytoscape will create new networks. If a new network is created from an old network, for example by selecting a set of nodes in one network and copying these nodes to a new network (via the **File** → **New** → **Network** option), it will be shown immediately follows the network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

Network views can also be detached (undocked) from the main Cytoscape window. When detached, a view window can be dragged to another monitor, resized, maximized and minimized by using the normal window controls for your operating system. Notice, however, that closing a view window does not destroy it, but simply reattaches it to the Cytoscape window.

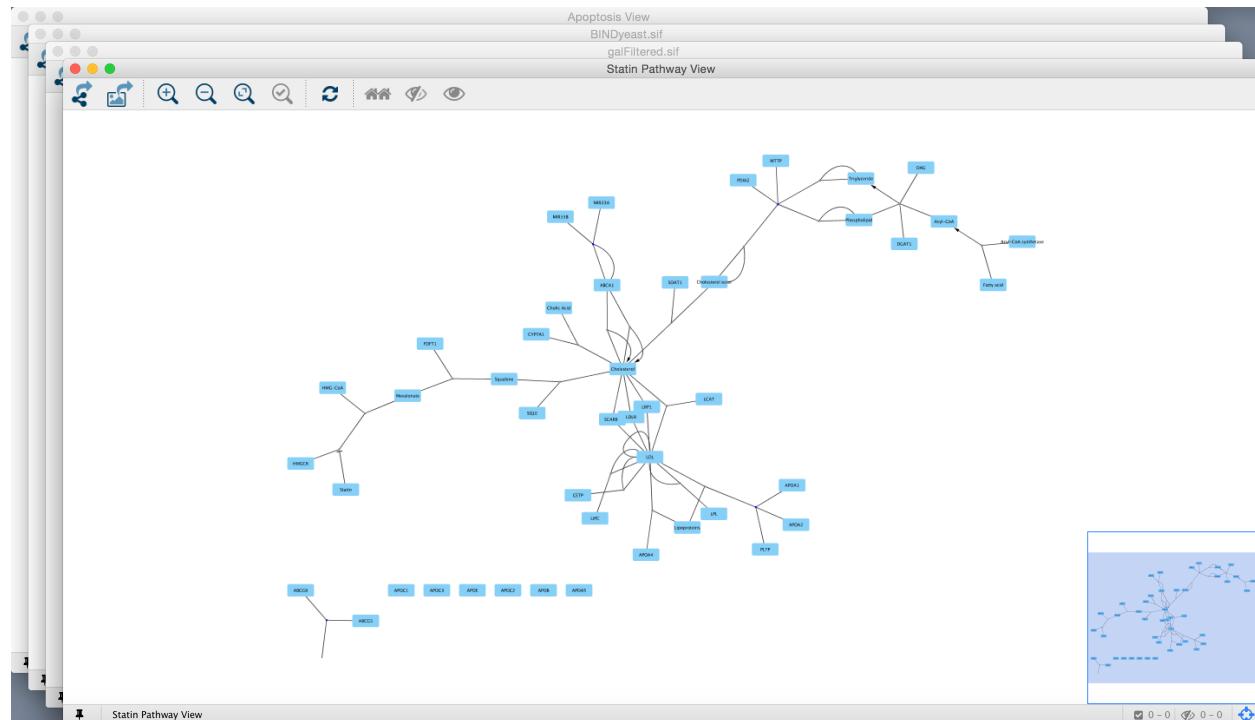
4.4.1 Arrange Network Windows

When you have detached network view windows, you can arrange them by selecting one of these options under **View** → **Arrange Network Windows**:

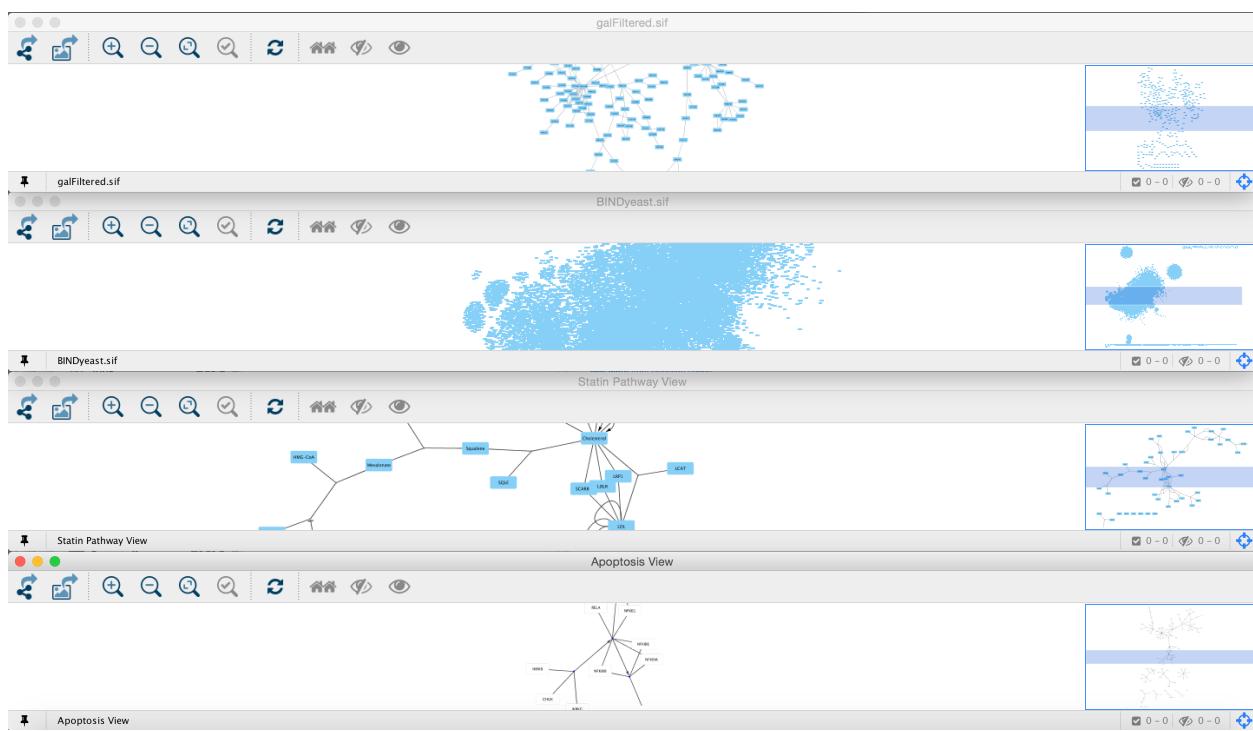
Grid



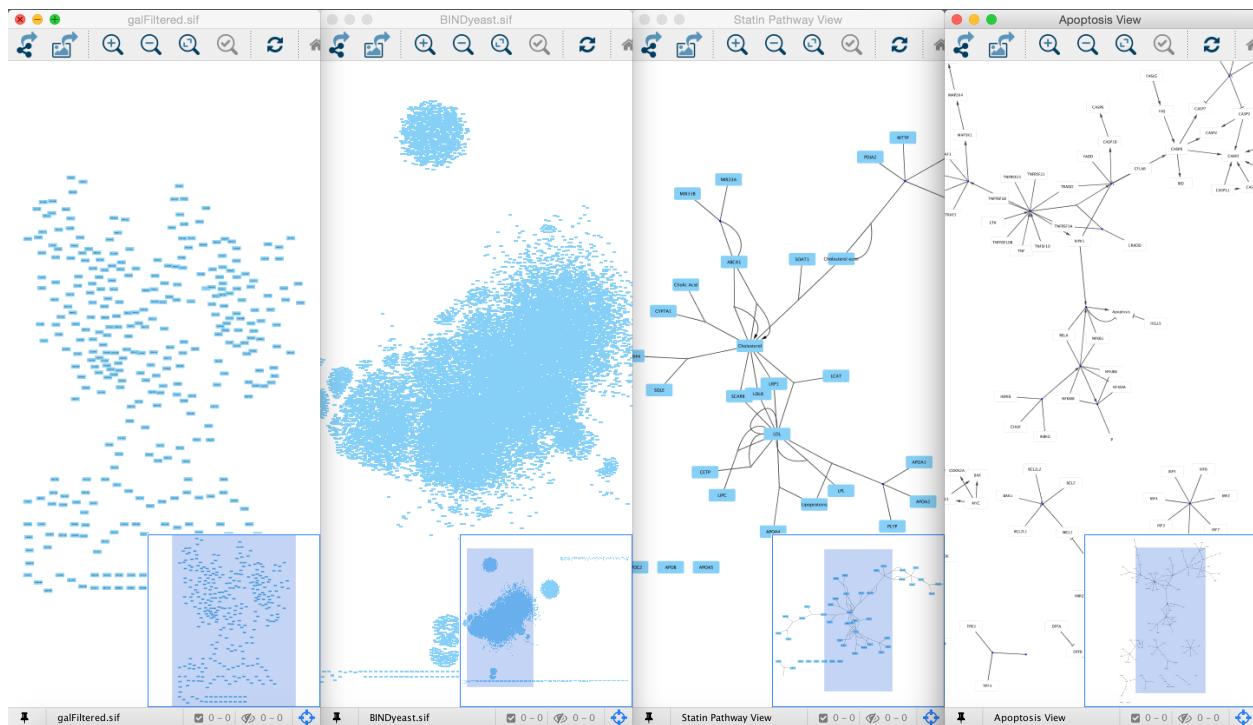
Cascade



Vertical Stack



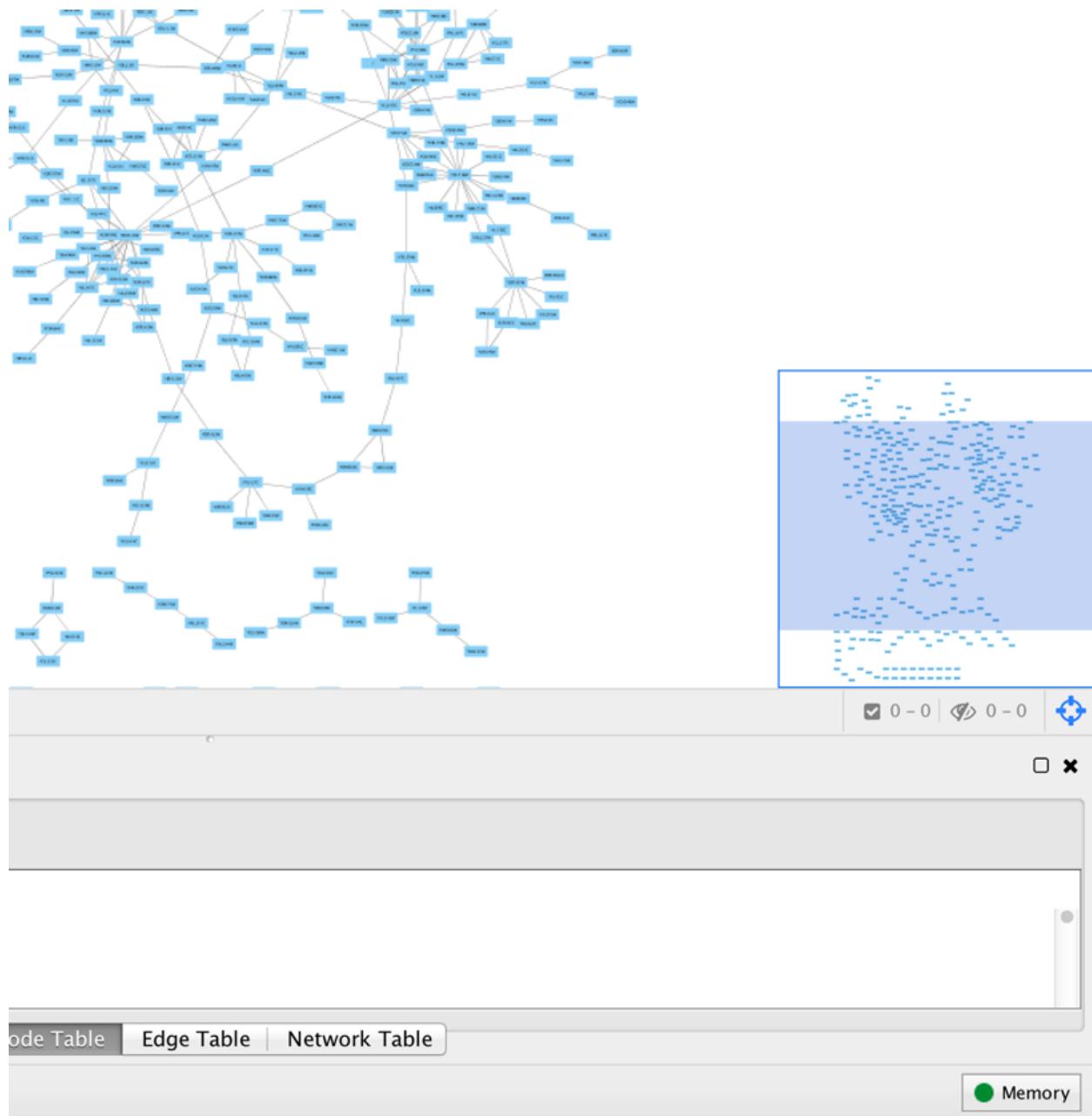
Side by Side



4.5 The View Navigator

The View Navigator (or “bird’s eye view”) is a control that shows an overview 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. You can show or hide it by clicking the “Show/Hide

Navigator” button  or pressing the key “N”.



CHAPTER 5

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

5.1 Import Fixed-Format Network Files

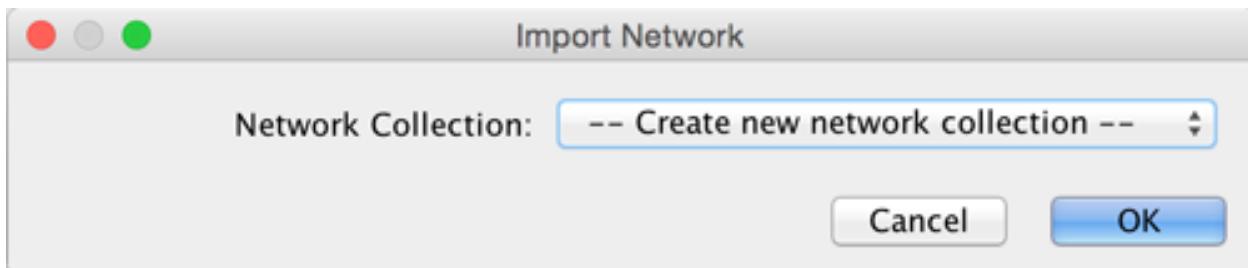
Network files can be specified in any of the formats described in the **Supported Network Formats** section. Networks are imported into Cytoscape through the **File → Import → Network** menu. The network file can either be located directly on the local computer, or found on a remote computer (in which case it will be referenced with a URL).

5.1.1 Load Networks from Local Computer



In order to load a network from a local file 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.

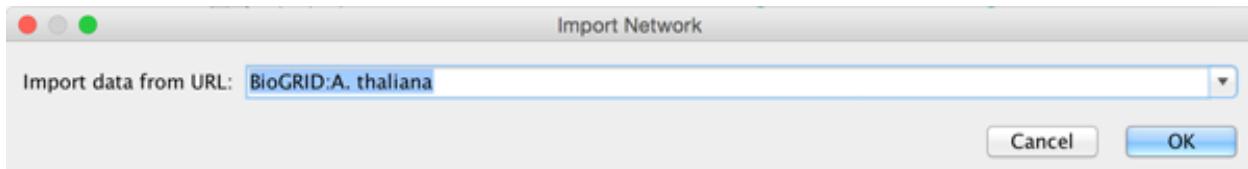


Alternatively, you can simply drag and drop a network file from the desktop into the *Network* list (**Control Panel**), rather than selecting the file from the menu option.

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

5.1.2 Load Networks from a Remote Computer (URL import)

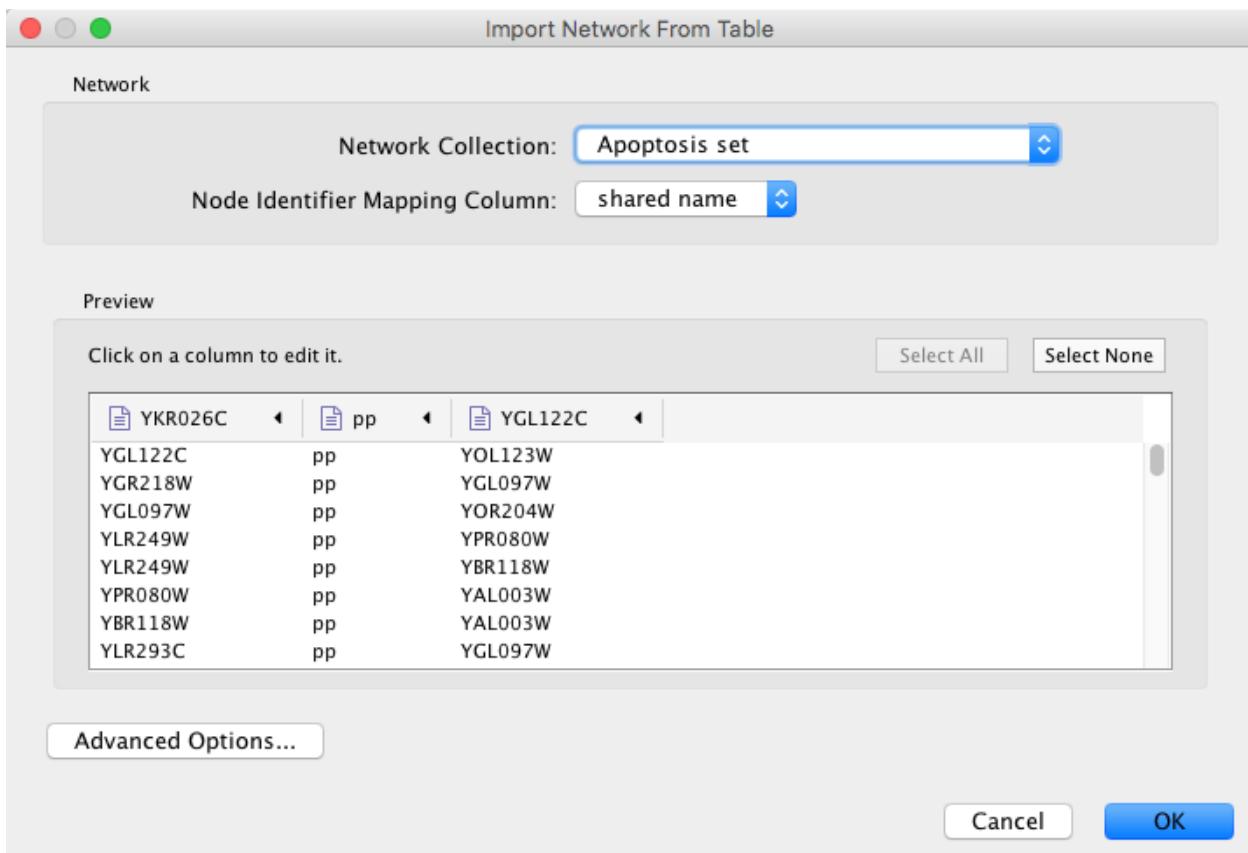
To load a network from a remote file, you can select **File → Import → Network → URL....**. In the import network dialog, insert the appropriate URL, either manually or using URL bookmarks. Bookmarked URLs can be accessed by clicking on the arrow to the right of the text field (see the Bookmark Manager 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 Settings....** This is further described in the [Preferences](#) section.

5.2 Import Networks from Unformatted Table Files

Cytoscape supports the import of networks from delimited text files and Excel workbooks using **File → Import → Network → File....** 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.



5.2.1 Supported Files

The **Import Network from Table** function supports delimited text files and Microsoft Excel Workbooks. For Excel Workbooks with multiple sheets, one sheet can be selected for import at a time. The following is a sample table file:

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:

One row in a network table file represents an edge and its edge data columns. This means that a network file is considered a combination of network data and edge column data. A table may contain columns that aren't meant to be edge data. 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	Q5397-1	vv HPRD	Currently not available	14682623 15892868(Marcotte)	Mammalia	Homo sapiens	protein protein	HPRD Marcotte	0	
4174	7311 MCMS	UBA52	P33992	P62987	neighbouring_reaction	Currently not available	15608231(Reactome)	Homo sapiens	Homo sapiens	protein protein	Reactome	1	
7040	7040 TGFBI	TGFBI	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 file and contains network data (interactions), edge data, and node data. To import network and edge data from this table, choose Unique ID A as source, Unique ID B as target, and Interactor types as interaction type. Next, turn off columns used for node data (Alternative ID A, species B, etc.). Other columns can be imported as edge data.

The network import function cannot import node table columns - only edge table columns. To import node table columns from this table, please see the **Node and Edge Column Data** section of this manual.

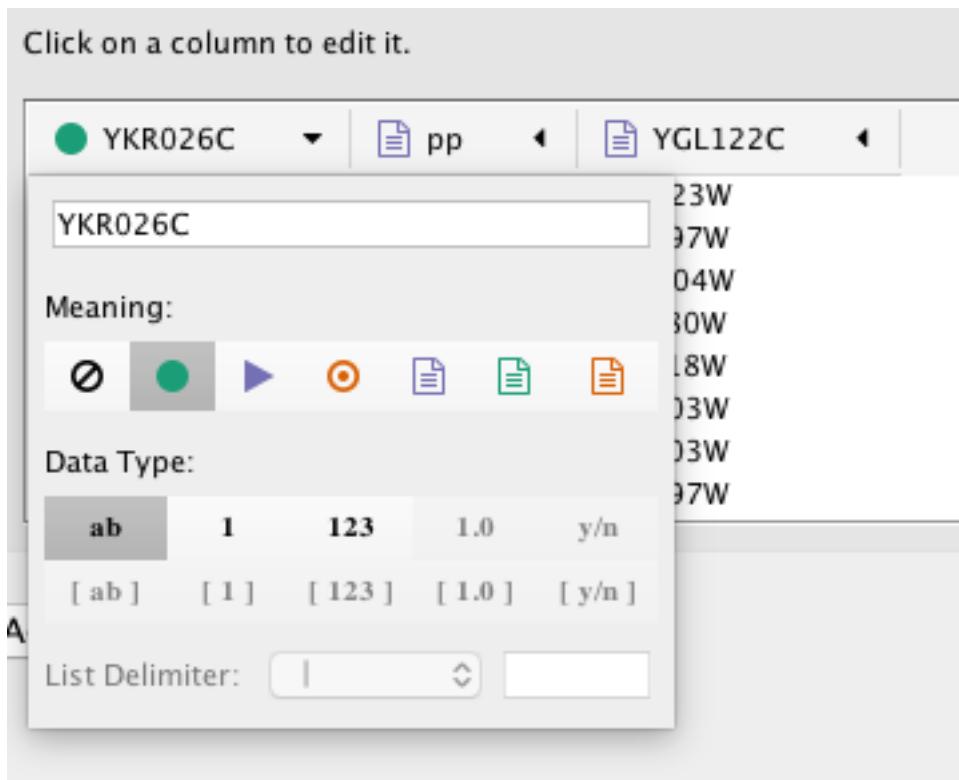
Note (1): This data is taken from the *A merged human interactome* datasets by Andrew Garrow, Yeyejide Adeleye and Guy Warner (Unilever, Safety and Environmental Assurance Center, 12 October 2006). Actual data files are available at http://wiki.cytoscape.org/Data_Sets/.

5.2.2 Basic Operations

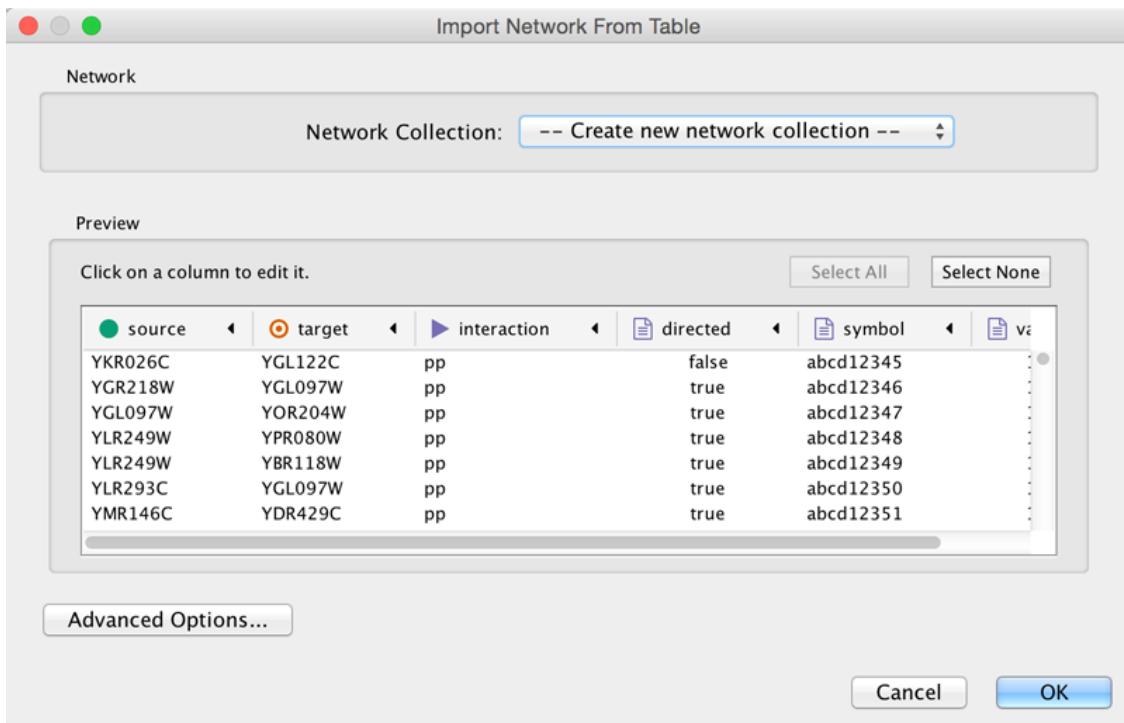
To import network from text/Excel tables, please follow these steps:



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. Clicking on any column header will bring up the interface for selecting source, interaction and target:



4. (Optional) Define edge table columns, if applicable. Network table files can have edge table columns in addition to network data.
 - Enable/Disable Table Columns: You can enable/disable column data by selecting the [attachment:disablecolumn.png] symbol in the column editor.



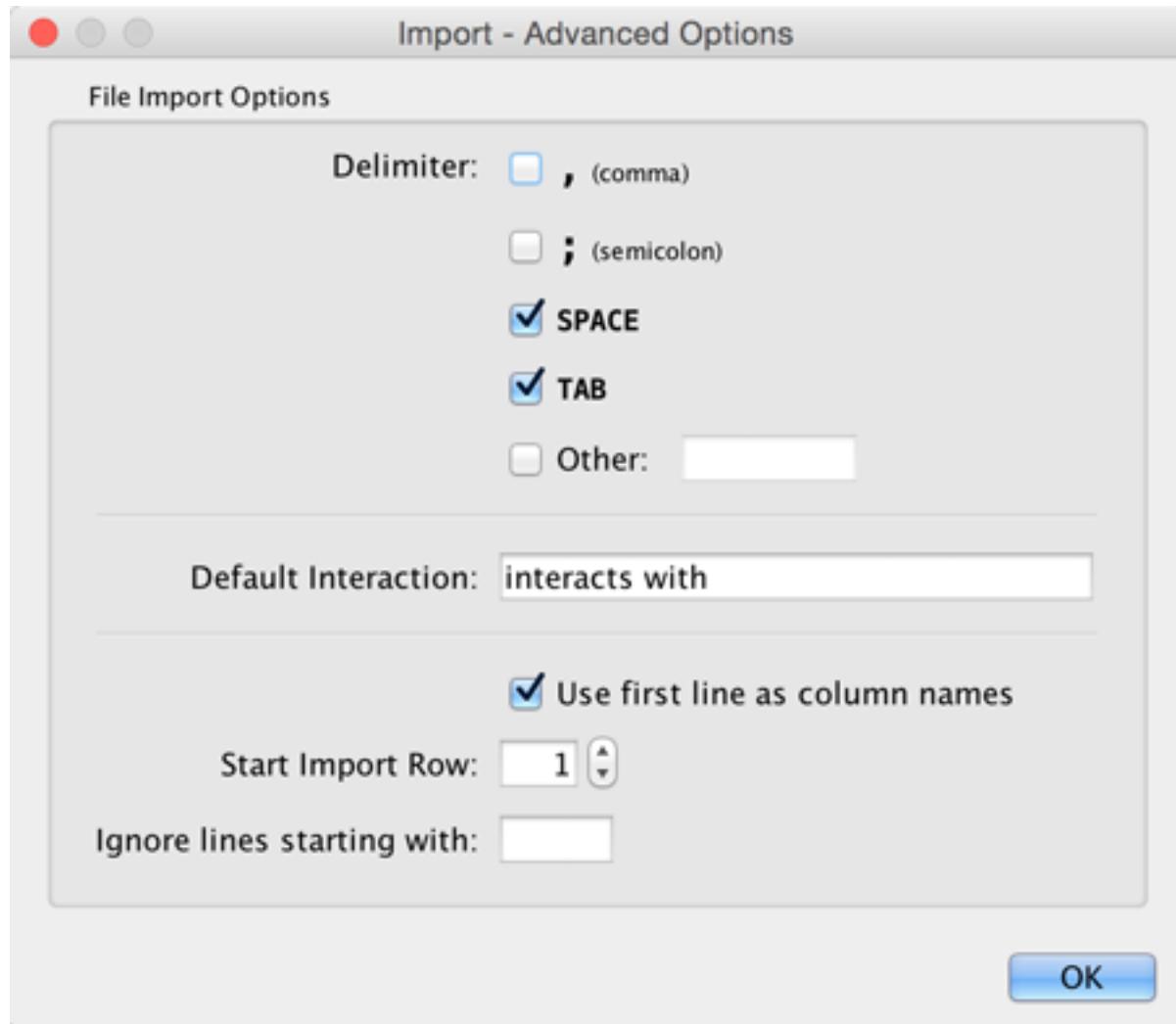
- Change Column Name and Data Types: You can also modify the column name and data type in the column editor. For more detail, see [Modify Column Name/Type](#) below.

5. Click the OK 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.

5.2.3 Advanced Options

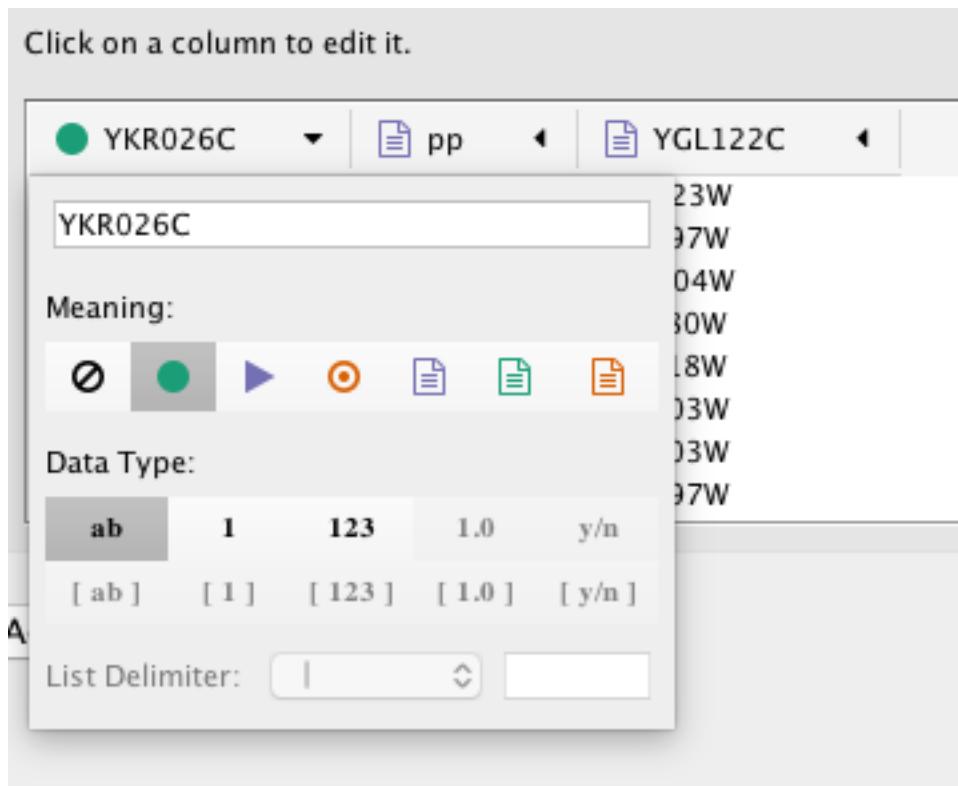


You can select several options by clicking the **Advanced Options** button in the main import interface.

- Delimiter: You can select multiple delimiters for text tables. By default, Tab and Space are selected as delimiters.
- Default Interaction
- Transfer first line as column names: Selecting this option will cause all edge 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.
- Ignore lines starting with: Rows starting with this character will not be imported. This option can be used to skip comment lines in text files.

5.2.4 Modify Column Name/Type

In the **Import Network from Table** interface, you can change the name and data type of column by clicking on any column header:



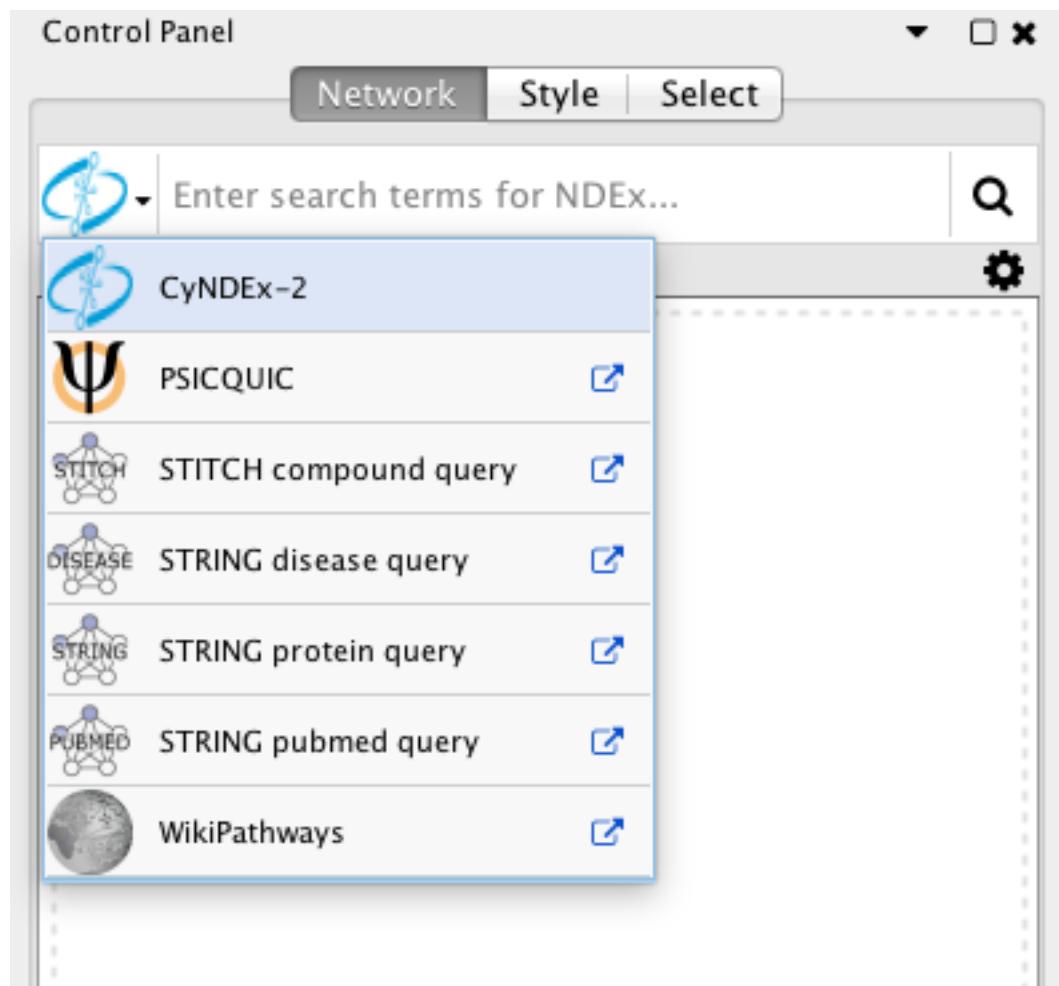
Column names and data types can be modified here.

- Modify Column Name - just enter a new column name.
- Modify Column Data Type - The following column 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 column data type 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).

5.3 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, under **File → Import → Network → Public Databases....**. As of Cytoscape 3.6, a new **Search bar** is also available at the top of the **Network** panel in the **Control Panel**. From the search bar, you can directly access several public databases.



5.3.1 What is a Web Service?

A web service is a standardized, platform-independent mechanism for computers to interact over the internet. These days, many major biological databases publish their data with a 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/>

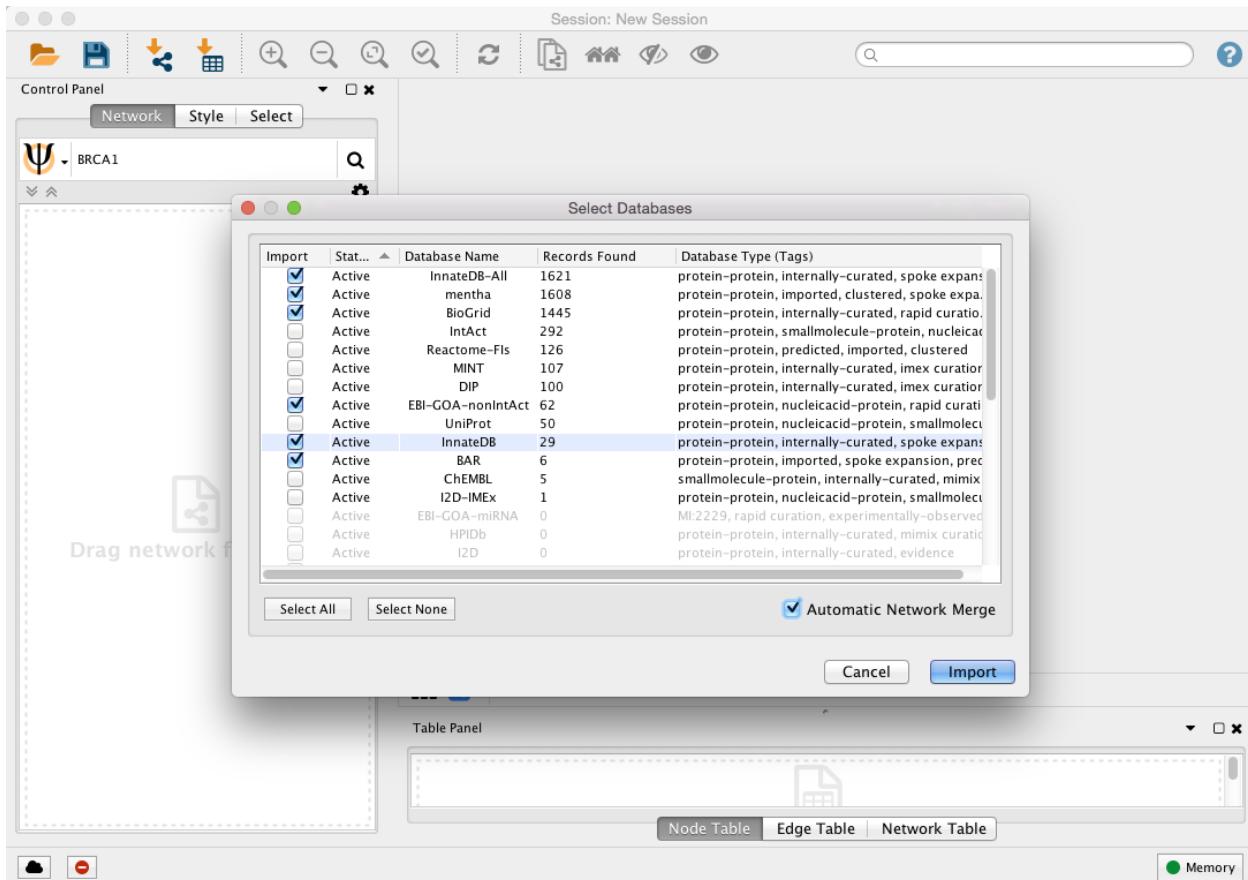
Cytoscape core developer team have developed several web service clients using this framework. Cytoscape supports many web services including:

- **PSICQUIC**: Standard web service for biological interaction data sets. The full list of PSICQUIC-compatible databases is available [here](#). PSICQUIC is available from the **Search bar**.
- **Pathway Commons**: Integrated data from pathway and network resources. The full list of supported resources is available [here](#). Pathway Commons is available from **File → Import → Network → Public Databases....**
- **STITCH** and **STRING**: STITCH is a database of known and predicted interactions between chemicals and proteins. STRING is a database of known and predicted protein-protein interactions. STITCH and STRING are available from the **Search bar** and from **File → Import → Network → Public Databases....**
- **NDEEx**: The Network Data Exchange (NDEEx) Project provides an open-source framework where scientists and organizations can share, store, manipulate, and publish biological network knowledge. NDEEx is available from the **Search bar**.

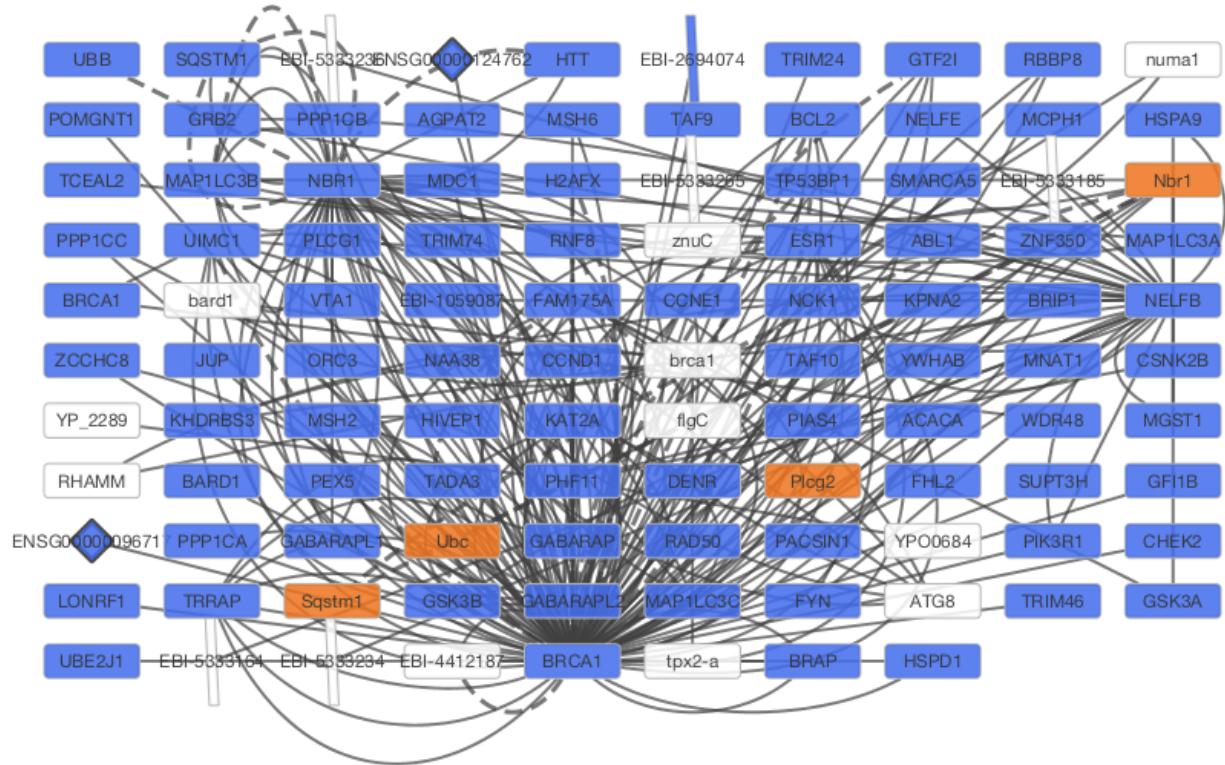
The following sections describe how to import network from external databases.

5.3.2 Example: Retrieving Protein-Protein Interaction Networks from Multiple Databases

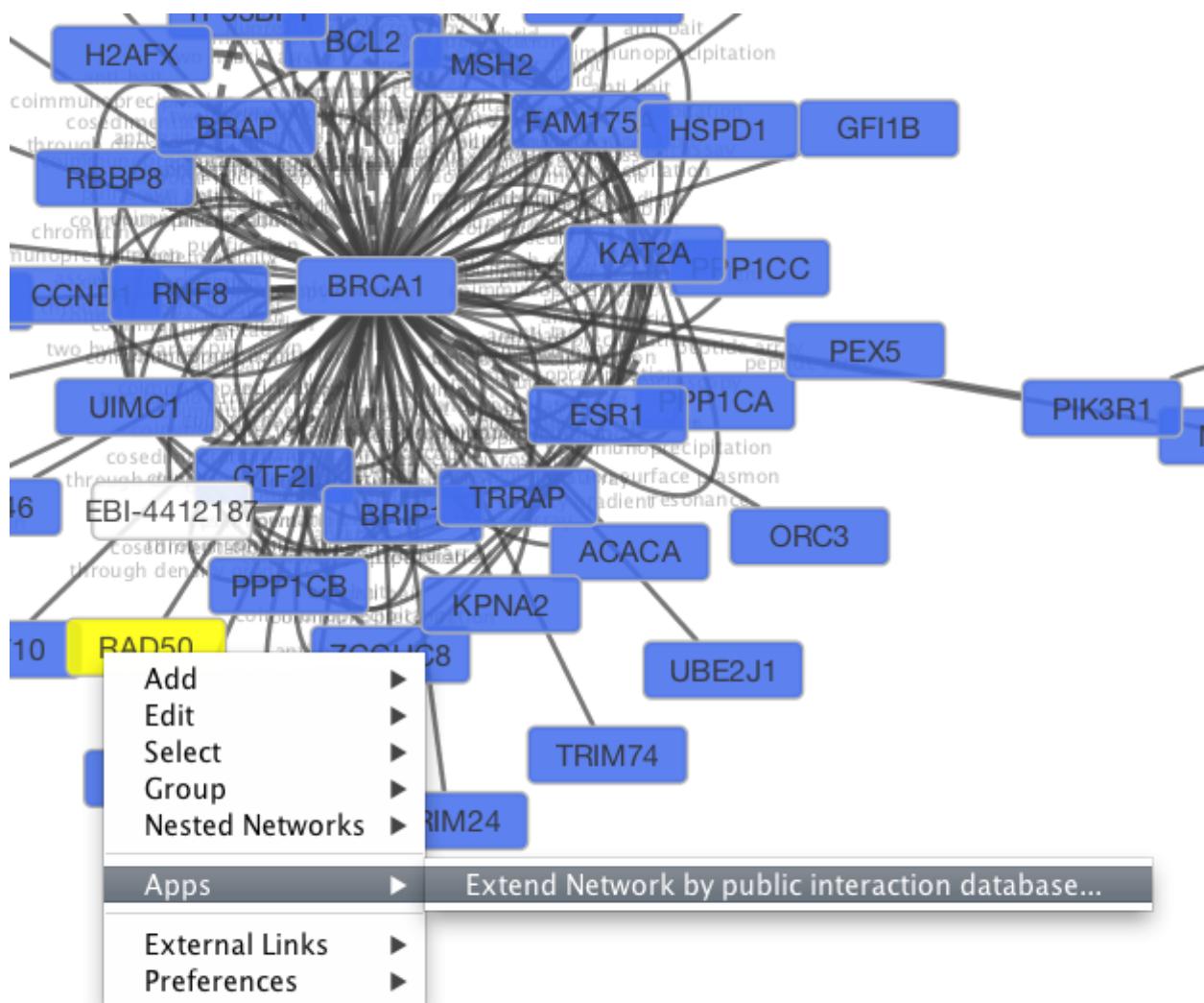
- In the **Search bar**, select PSICQUIC from the drop-down menu and type in one or more search terms, such as BRCA1.
- Click **Enter** to start the search.
- Select databases from the hits.



- Click the **Import** button to import selected network data.



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.

PSICQUIC Options

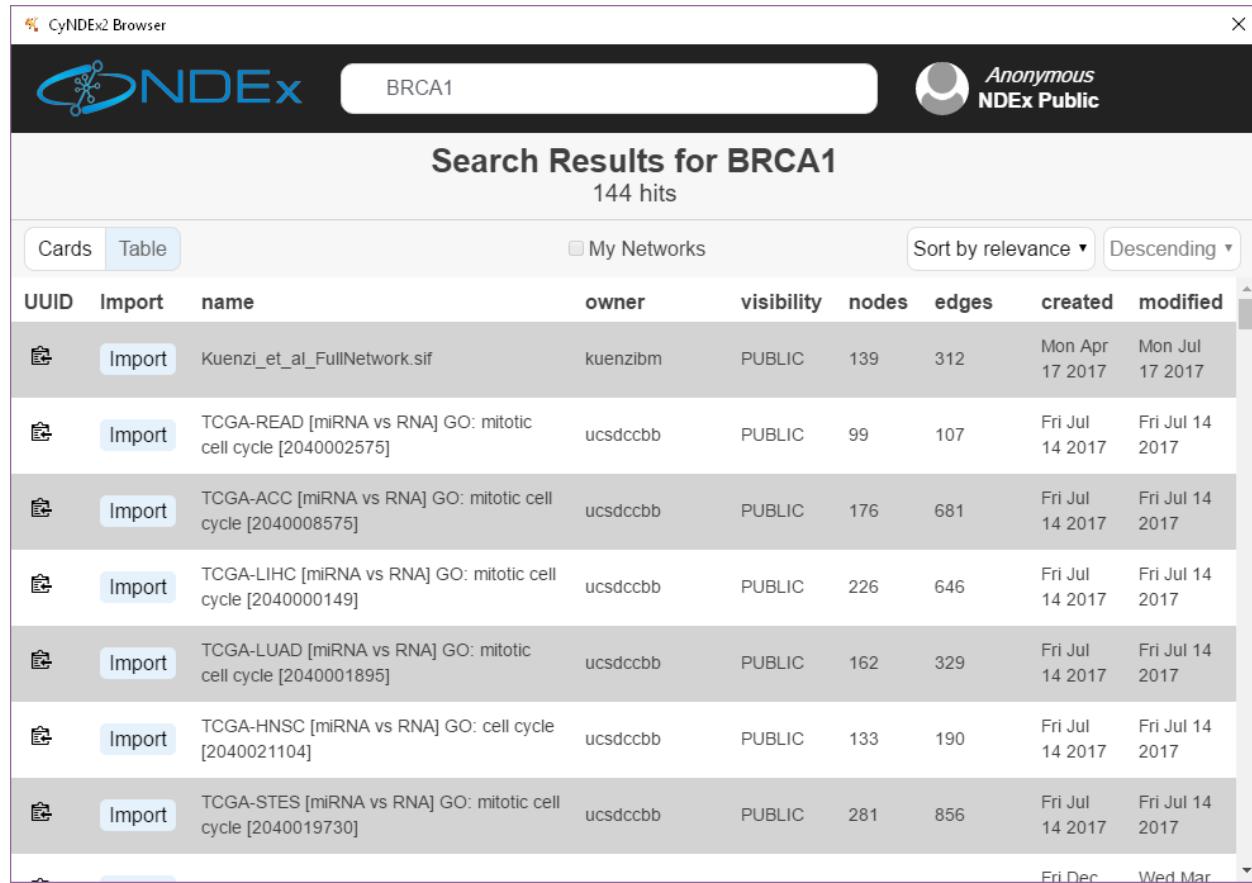
PSICQUIC Web Service Client has three search modes:

- Search by ID
- Search by MIQL
- Search by Species

By default, search mode is set to **Search by ID**. You can search all databases by ID, such as gene symbol, Uniprot ID, or NCBI gene ID. If the search mode is set to MIQL, you can use [MIQL](https://psicquic.github.io/MiqlDefinition.html) (<https://psicquic.github.io/MiqlDefinition.html>) for search. If you want to search interactions by keywords or specific functions, this is the powerful query language to filter the result. The last option is for importing all interactions for the species (i.e., interactome).

5.3.3 Example: Retrieving Networks from NDEx

- In the **Search bar**, select CyNDEEx-2 from the drop-down menu and type in one or more search terms, such as BRCA1.
- Click **Enter** to start the search.
- In the CyNDEEx-2 Browser dialog, click a network's **Import** button to load it into Cytoscape and visualize it. When you have finished loading networks, dismiss the dialog box.



The screenshot shows the CyNDEEx-2 Browser interface. At the top, there is a search bar containing "BRCA1". To the right of the search bar is a "My Networks" checkbox, a "Sort by relevance" dropdown set to "Descending", and an "Anonymous NDEx Public" button. Below the search bar, the title "Search Results for BRCA1" is displayed, followed by "144 hits". The main area is a table with the following columns: UUID, Import, name, owner, visibility, nodes, edges, created, and modified. The table lists seven network entries, each with an "Import" button. The first entry is "Kuenzi_et_al_FullNetwork.sif" (owner: kuenzibm, PUBLIC, 139 nodes, 312 edges, created: Mon Apr 17 2017, modified: Mon Jul 17 2017). The other six entries are TCGA-related networks (all owned by ucsdccbb, PUBLIC, various node and edge counts, all created and modified on Fri Jul 14 2017).

UUID	Import	name	owner	visibility	nodes	edges	created	modified
	Import	Kuenzi_et_al_FullNetwork.sif	kuenzibm	PUBLIC	139	312	Mon Apr 17 2017	Mon Jul 17 2017
	Import	TCGA-READ [miRNA vs RNA] GO: mitotic cell cycle [2040002575]	ucsdccbb	PUBLIC	99	107	Fri Jul 14 2017	Fri Jul 14 2017
	Import	TCGA-ACC [miRNA vs RNA] GO: mitotic cell cycle [2040008575]	ucsdccbb	PUBLIC	176	681	Fri Jul 14 2017	Fri Jul 14 2017
	Import	TCGA-LIHC [miRNA vs RNA] GO: mitotic cell cycle [2040000149]	ucsdccbb	PUBLIC	226	646	Fri Jul 14 2017	Fri Jul 14 2017
	Import	TCGA-LUAD [miRNA vs RNA] GO: mitotic cell cycle [2040001895]	ucsdccbb	PUBLIC	162	329	Fri Jul 14 2017	Fri Jul 14 2017
	Import	TCGA-HNSC [miRNA vs RNA] GO: cell cycle [2040021104]	ucsdccbb	PUBLIC	133	190	Fri Jul 14 2017	Fri Jul 14 2017
	Import	TCGA-STES [miRNA vs RNA] GO: mitotic cell cycle [2040019730]	ucsdccbb	PUBLIC	281	856	Fri Jul 14 2017	Fri Jul 14 2017

You can browse networks in a number of ways:

- Sort by column (via the **Sort by** dropdown) or order the sorted column (via the **Descending/Ascending** dropdown)
- Use your NDEx user account to enable searching within your network list
 - Create an NDEx account by visiting the [NDEx Public Server](#) website
 - Add a profile to the CyNDEEx-2 Browser (by filling out the profile form reachable by clicking on the *Anonymous* credential in the upper right)
 - Click on the **My Networks** checkbox at the top of the network table
- Enter a new query into the search bar at the top of the dialog
 - Find a network via a UUID sent to you by using the *uuid:* selector in your search (e.g., `uuid:50e3dff7-133e-11e6-a039-06603eb7f303`)
- View network images by clicking the **Cards** button at the top of the network table

Cards **Table** My Networks **Sort by owner** **Descending**

rasmachine PUBLIC

The Heart Machine
Last updated Fri Nov 10 2017

3681 nodes 7189 edges

The Heart Machine reads new literature (using the REACH system) on the molecular mechanisms governing cardiac toxicity caused by kinase inhibitors, and extracts and assembles mechanisms into a computational model using INDRA.

Import

rasmachine PUBLIC

The Diabetes Machine
Last updated Fri Nov 10 2017

567 nodes 1148 edges

The Diabetes Machine reads new literature on the molecular mechanisms around insulin secretion in pancreatic beta cells, and extracts and assembles mechanisms into a computational model using INDRA.

Import

rasmachine PUBLIC

The RAS Machine
Last updated Fri Nov 10 2017

1063 nodes 4216 edges

The RAS Machine reads new literature on RAS daily, and extracts and assembles mechanisms into a computational model using INDRA.

Import

CyNDEx-2, like NDEx, uses standard [Lucene Syntax](#) as its network search language. For additional information, see [Searching Networks in NDEx](#). More information about CyNDEx-2, see the [CyNDEx-2 App Store page](#).

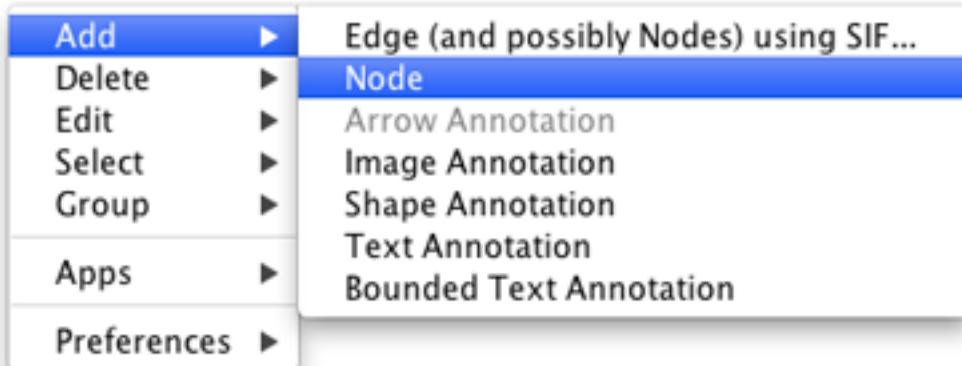
Note that you can save networks back to the NDEx database by using [Export](#) options.

5.4 Create a New Network or Edit One 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. You can edit an existing network using the same process.

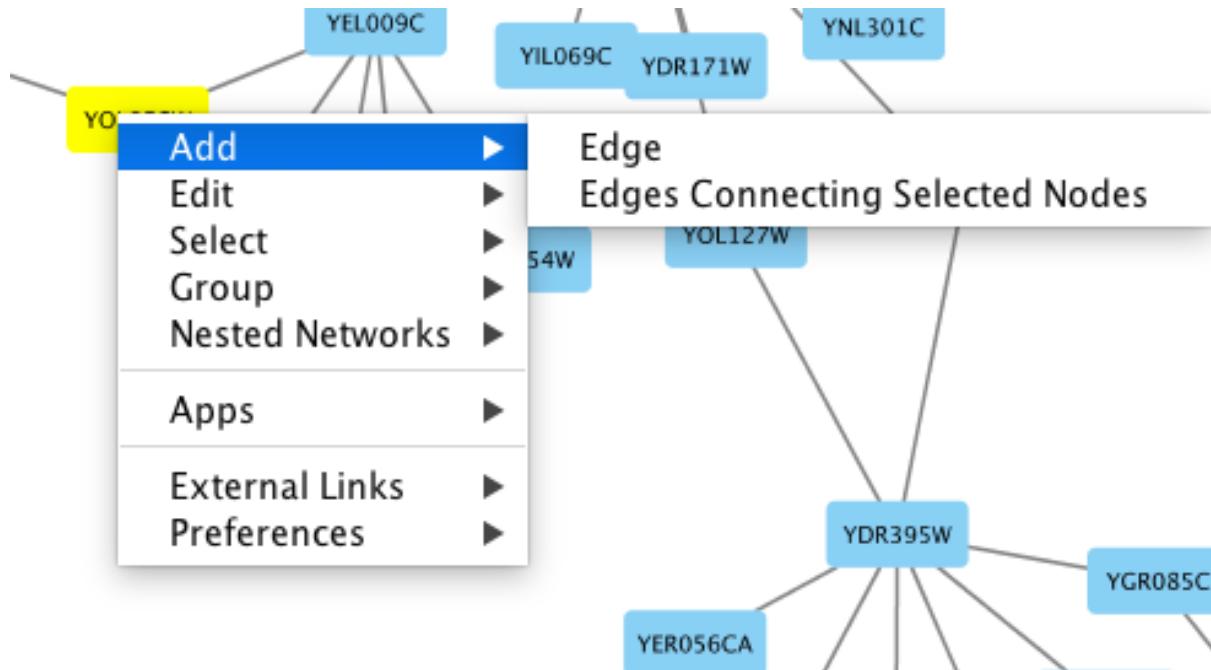
5.4.1 Adding a Node

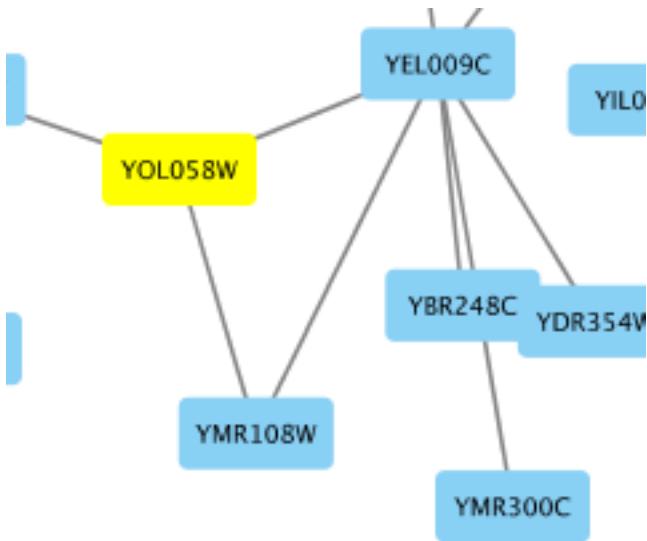
To add a new node, right-click on an empty space of the network view panel. Select **Add** → **Node** item from the pop-up menu.



5.4.2 Adding an Edge

To add an edge to connect nodes, right-click on the source node. Select **Edit** → **Add Edge** from the pop-up menu. Next, click on the target node. The 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 also select two or more nodes to connect and in the right-click menu select **Add** → **Edges Connecting Selected Nodes** to create edges connecting all selected nodes.



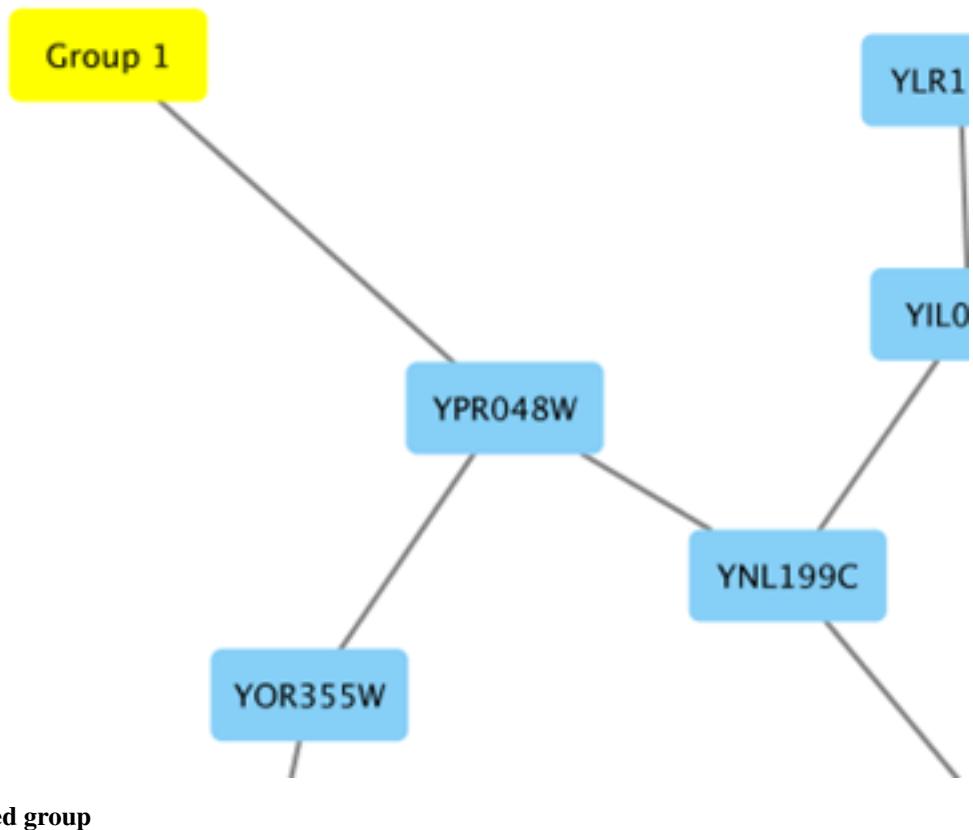


You can delete nodes and edges by selecting a number of nodes and edges, then selecting **Edit → Cut**. You can also delete selected nodes and edges from the **Edit** menu, under **Edit → Delete Selected Nodes and Edges....**. You can recover any nodes and edges deleted from a network by going to **Edit → Undo**.

5.4.3 Grouping Nodes

Any number of nodes can be grouped together and displayed as either one group node or as the individual nodes. To create a group, select two or more nodes and right-click to select **Group → Group Selected Nodes**. You will be prompted to select a name for the group node. Once a group is created, you can use the right-click menu to collapse or expand the group. You can also quickly collapse/expand a group by double-clicking on the group node or any of its children to toggle back and forth.

Collapsed group

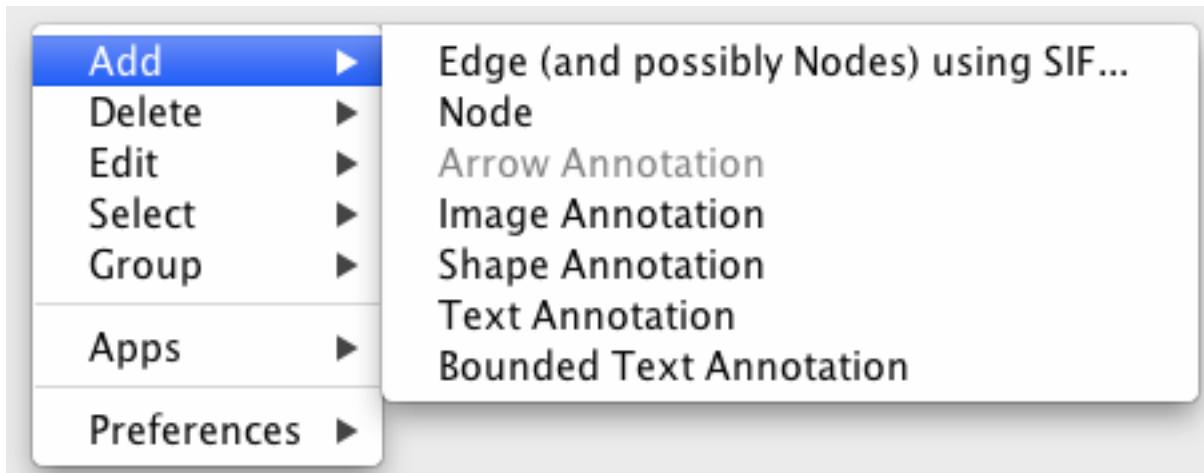


The appearance and behavior of grouped nodes depends on the [group settings](#) in effect when the group is created.

Settings can be managed for the entire Cytoscape session (via **Edit** → **Preferences** → **Group Preferences...**) or for a specific group (right-click **Preferences** → **Group Preferences...**).

5.4.4 Adding Network Annotations

Annotations in the form of text, images or shapes can be added to the network canvas by right-clicking anywhere on the canvas and selecting one of the Annotation choices in the **Add** menu. You can add an image of your own, choose from a shapes library or add either plain or bounded text. Shapes and text are customizable and any added annotations can be edited from the right-click context menu.



CHAPTER 6

Nested Networks

Cytoscape has the ability to associate a **Nested Network** with any node. A nested network is any other network currently defined in Cytoscape. This capability allows for creation of network hierarchies as well as circular relationships. For example, various module-finding plugins make use of nested networks in the overview network that they generate. There each node representing a module contains a nested network.

Note that **Nested Networks** are similar in concept to **Groups**, but with very different consequences. A Nested Network incorporates an independent network that does not share nodes or attributes with the current network – it can be edited only as a separate network collection or in a separate Cytoscape session. Its nodes cannot be connected to nodes in the current network and cannot be found or filtered using Cytoscape functions or apps. Conversely, nodes in a Group are part of the current network, can have edges to nodes outside of the group, and can be found or filtered using Cytoscape functions or apps.

Most often, **Groups** are more functional and easier to use.

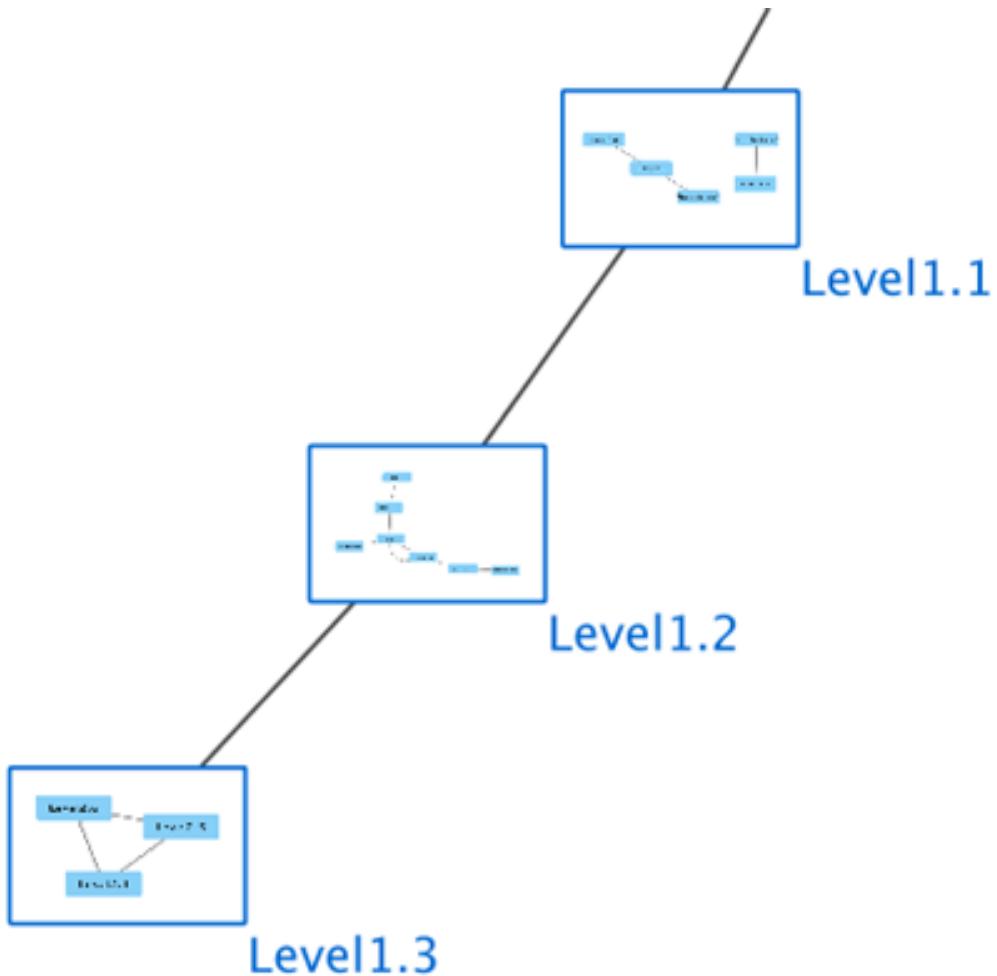
6.1 Creating Nested Networks

There are currently two ways in which a user can create nested networks.

- By importing a Nested Network Format (NNF) file. (See: [NNF Network Format](#)).
- By manually constructing networks and assigning nested networks to individual nodes through the right-click node context menu. (See [Nested Network Node Context Menu](#)).

6.2 Visualization of Nested Networks

Nodes containing nested networks that are zoomed in sufficiently display an image for the nested network. If no current network view exists for the nested network the image will be a default icon, otherwise it will be a low-resolution rendering of the nested network's current layout.



CHAPTER 7

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)
- [Cytoscape.js JSON](#)
- [Cytoscape CX](#)

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.

7.1 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 column. 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 styles). This is discussed in the section on [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 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
```

7.1.1 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.

7.2 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 columns can be specified. There are only 2 possible line formats:

- A node “node” contained in a “network:”

```
network node
```

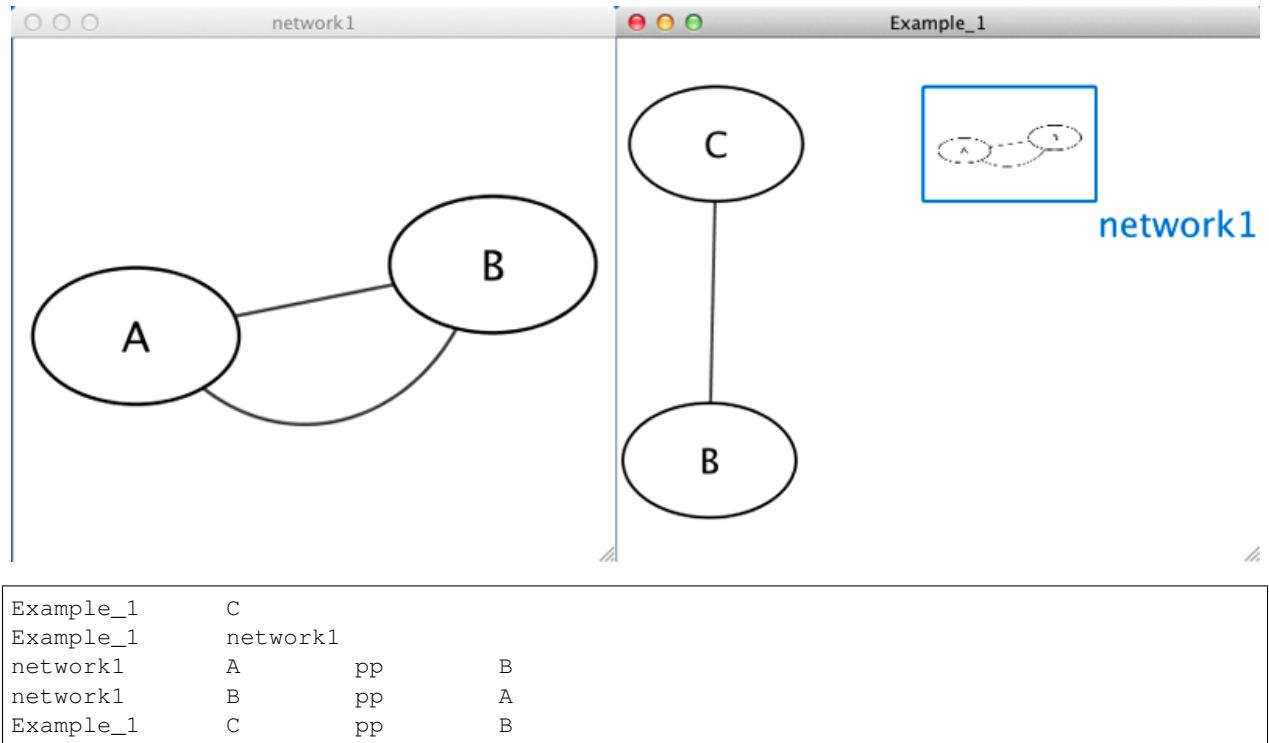
- 2 nodes linked together contained in a network:

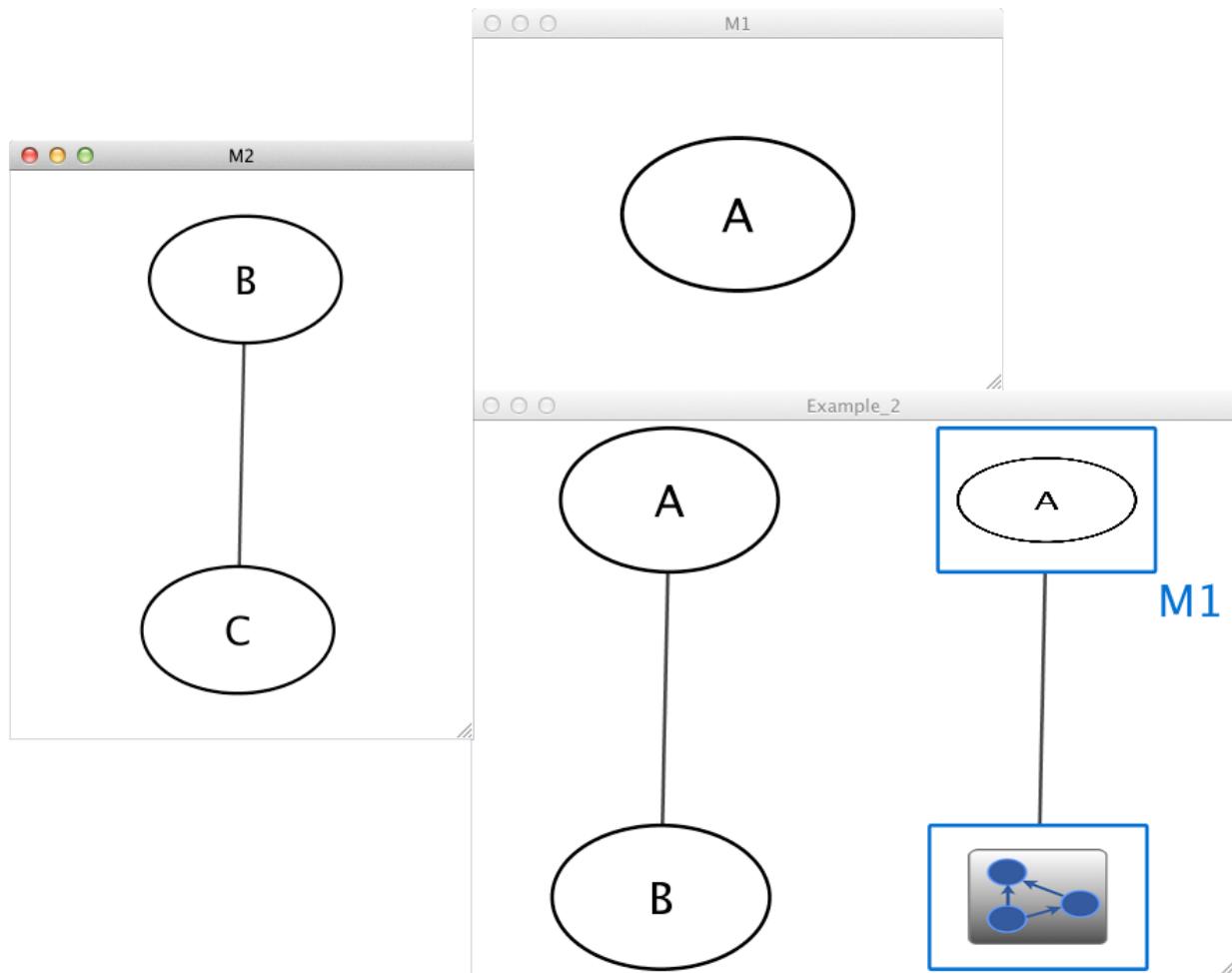
```
network node1 interaction node2
```

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.

7.2.1 Examples

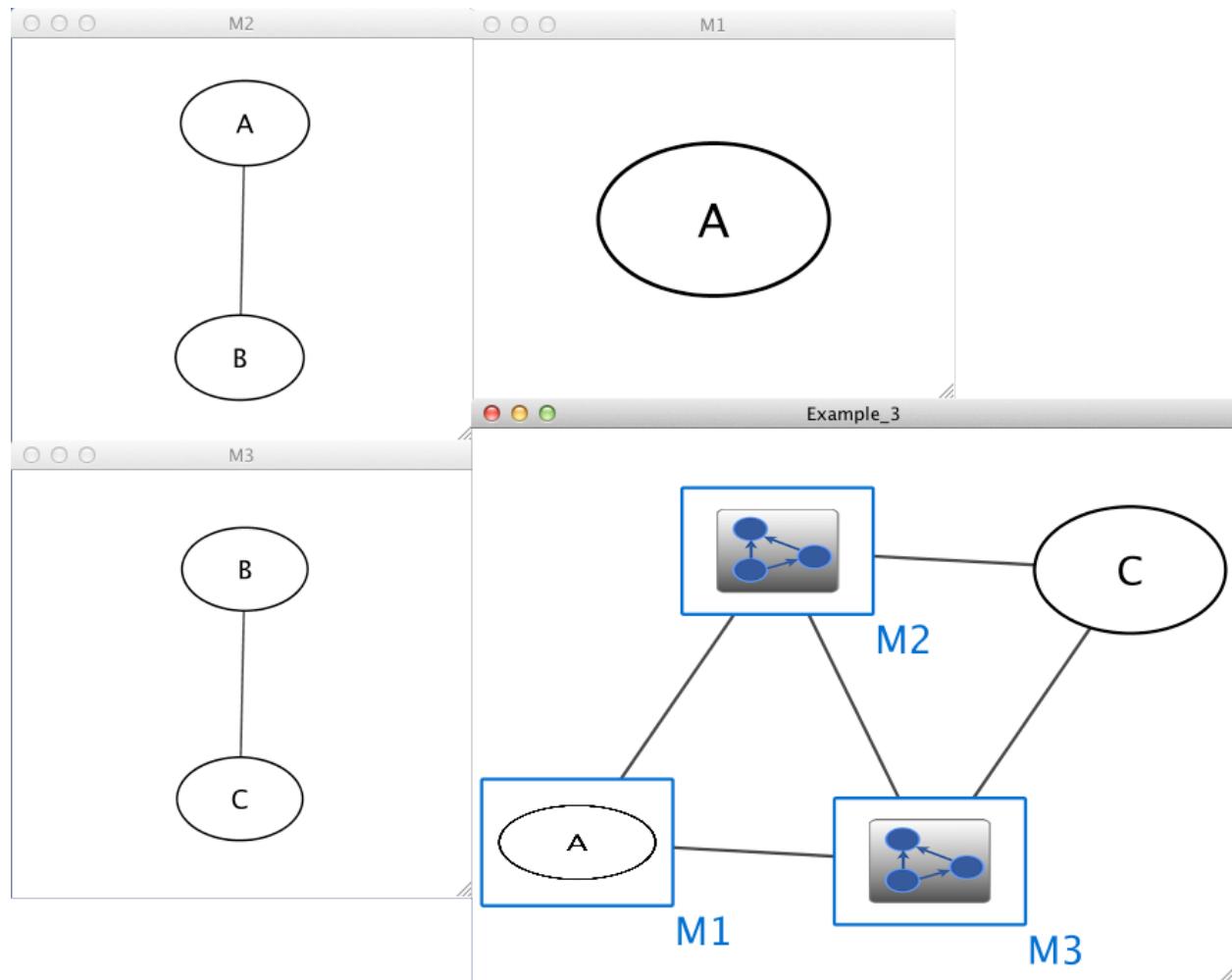
Example 1



Example 2

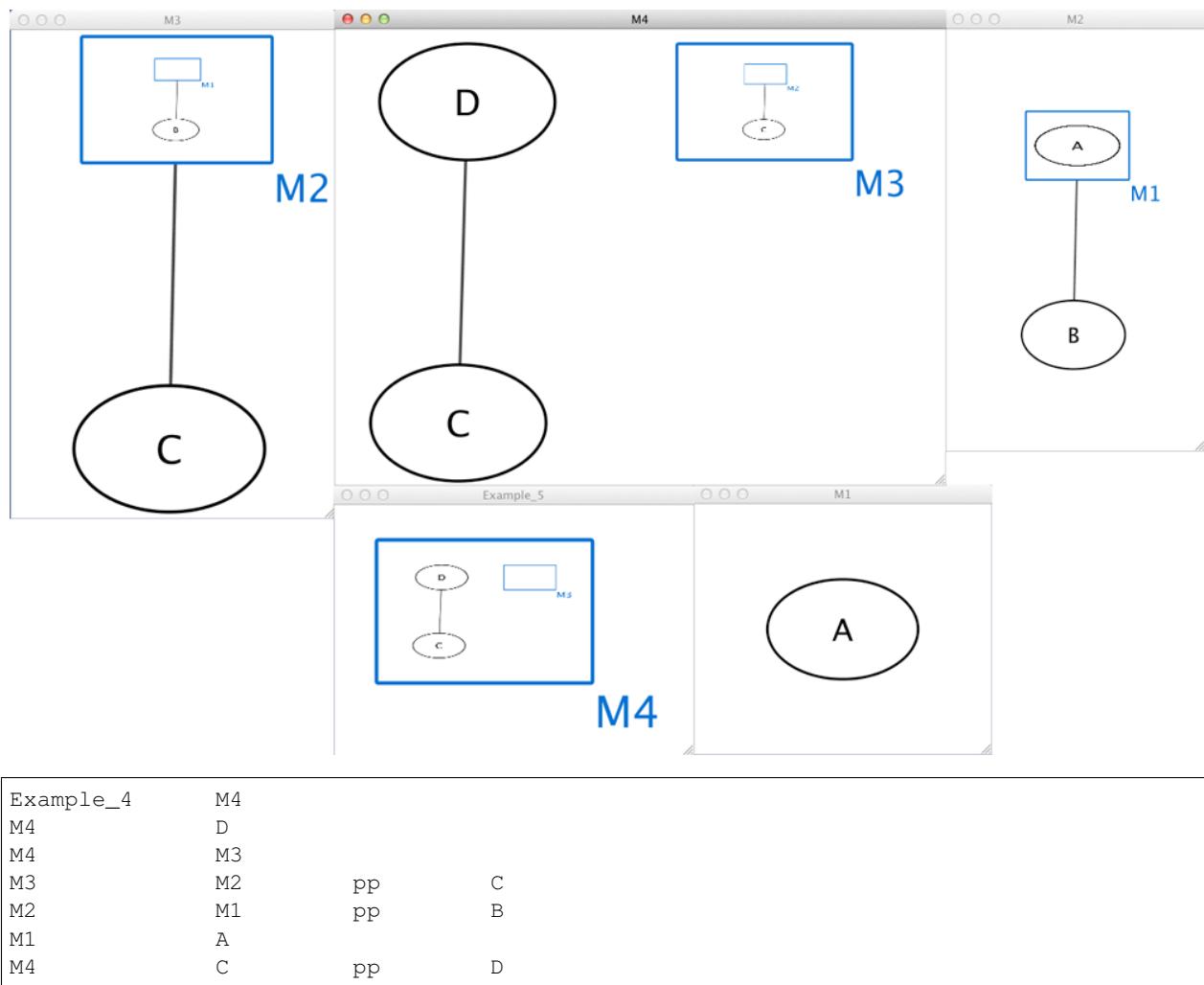
Example_2	M1
Example_2	M2
M1	A
M2	B pp C
Example_2	A pp B
Example_2	M1 im M2

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



7.3 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. Properties specified in a GML file will result in a new style named `Filename.style` when that GML file is loaded.

7.4 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 column data. 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.

7.5 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/>

7.6 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/>

7.7 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/>

7.8 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:

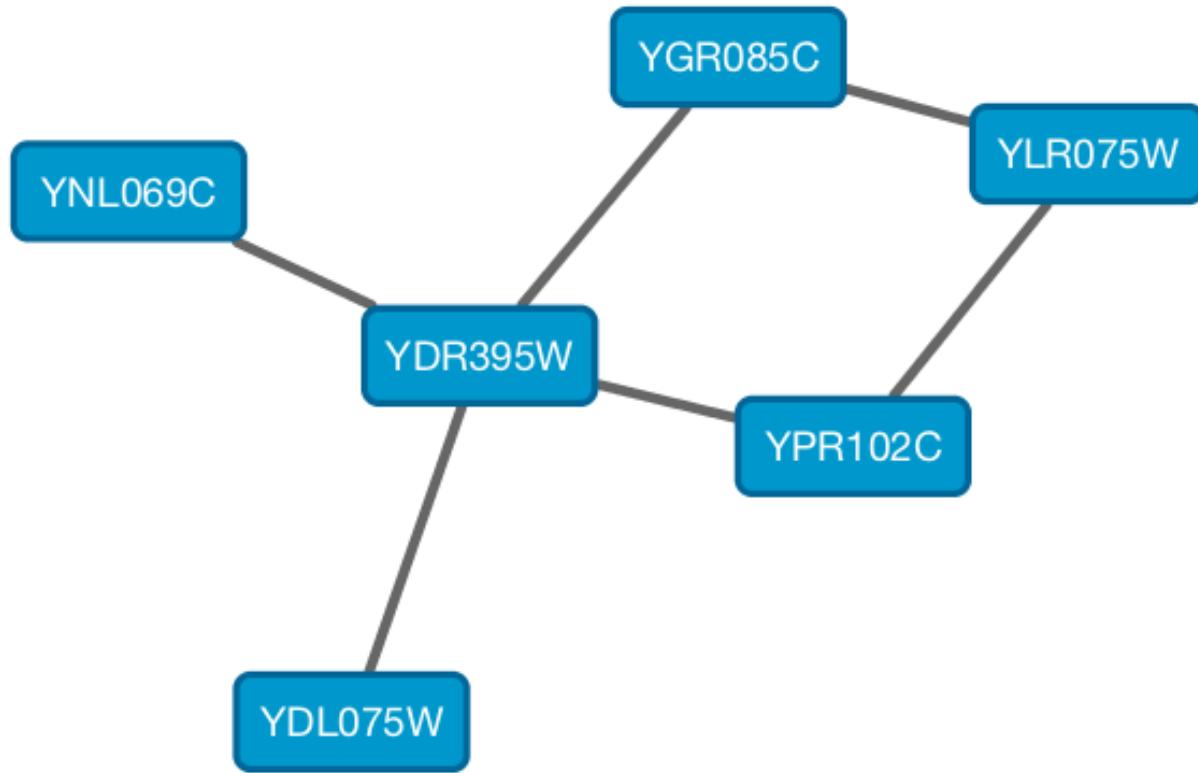
<http://graphml.graphdrawing.org/>

7.9 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 columns. Users can specify columns containing source nodes, target nodes, interaction types, and edge columns 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](#) section.

7.10 Cytoscape.js JSON

From Cytoscape 3.1.0 on, Cytoscape supports Cytoscape.js JSON files. You can use this feature to export your network visualizations to web browsers. Cytoscape.js has two ways to represent network data, and currently both reader and writer support only the array style graph notation. For example, this network in Cytoscape:



will be exported to this JSON:

```
{
  "elements": {
    "nodes": [ {
      "data": {
        "id": "723",
        "selected": false,
        "annotation_Taxon": "Saccharomyces cerevisiae",
        "alias": [ "RPL31A", "RPL34", "S000002233", "ribosomal protein L31A (L34A)" ↴(YL28) ],
        "shared_name": "YDL075W",
        "SUID": 723,
        "degree_layout": 1,
        "name": "YDL075W"
      },
      "position": {
        "x": 693.0518315633137,
        "y": -49.47506554921466
      },
      "selected": false
    }, {
      "data": {
        "id": "724",
        "selected": false,
        "annotation_Taxon": "Saccharomyces cerevisiae",
        "alias": [ "RPL31B", "RPL34B", "S000002234", "ribosomal protein L31B (L34B)" ↴(YL29) ],
        "shared_name": "YDR395W",
        "SUID": 724,
        "degree_layout": 5,
        "name": "YDR395W"
      },
      "position": {
        "x": 493.0518315633137,
        "y": -49.47506554921466
      },
      "selected": false
    }, {
      "data": {
        "id": "725",
        "selected": false,
        "annotation_Taxon": "Saccharomyces cerevisiae",
        "alias": [ "RPL31C", "RPL34C", "S000002235", "ribosomal protein L31C (L34C)" ↴(YL30) ],
        "shared_name": "YGR085C",
        "SUID": 725,
        "degree_layout": 2,
        "name": "YGR085C"
      },
      "position": {
        "x": 793.0518315633137,
        "y": -49.47506554921466
      },
      "selected": false
    }, {
      "data": {
        "id": "726",
        "selected": false,
        "annotation_Taxon": "Saccharomyces cerevisiae",
        "alias": [ "RPL31D", "RPL34D", "S000002236", "ribosomal protein L31D (L34D)" ↴(YL31) ],
        "shared_name": "YLR075W",
        "SUID": 726,
        "degree_layout": 2,
        "name": "YLR075W"
      },
      "position": {
        "x": 893.0518315633137,
        "y": -49.47506554921466
      },
      "selected": false
    }, {
      "data": {
        "id": "727",
        "selected": false,
        "annotation_Taxon": "Saccharomyces cerevisiae",
        "alias": [ "RPL31E", "RPL34E", "S000002237", "ribosomal protein L31E (L34E)" ↴(YL32) ],
        "shared_name": "YDL075W",
        "SUID": 727,
        "degree_layout": 1,
        "name": "YDL075W"
      },
      "position": {
        "x": 593.0518315633137,
        "y": -49.47506554921466
      },
      "selected": false
    }, {
      "data": {
        "id": "728",
        "selected": false,
        "annotation_Taxon": "Saccharomyces cerevisiae",
        "alias": [ "RPL31F", "RPL34F", "S000002238", "ribosomal protein L31F (L34F)" ↴(YL33) ],
        "shared_name": "YPR102C",
        "SUID": 728,
        "degree_layout": 1,
        "name": "YPR102C"
      },
      "position": {
        "x": 703.0518315633137,
        "y": -49.47506554921466
      },
      "selected": false
    } ]
  }
}
```

```
        "id" : "726",
        "selected" : false,
        "annotation_Taxon" : "Saccharomyces cerevisiae",
        "alias" : [ "RP23", "RPL16B", "S000005013", "ribosomal protein L16B (L21B) ↵
↳(rp23) (YL15)" ],
        "shared_name" : "YNL069C",
        "SUID" : 726,
        "degree_layout" : 1,
        "name" : "YNL069C"
    },
    "position" : {
        "x" : 627.3147710164387,
        "y" : -205.99251969655353
    },
    "selected" : false
}, {
    "data" : {
        "id" : "658",
        "selected" : false,
        "annotation_Taxon" : "Saccharomyces cerevisiae",
        "alias" : [ "RPL11B", "S000003317", "ribosomal protein L11B (L16B) (rp39B) ↵
↳(YL22)" ],
        "shared_name" : "YGR085C",
        "SUID" : 658,
        "degree_layout" : 2,
        "name" : "YGR085C"
    },
    "position" : {
        "x" : 804.3092778523762,
        "y" : -245.6235926946004
    },
    "selected" : false
}, {
    "data" : {
        "id" : "660",
        "selected" : false,
        "annotation_Taxon" : "Saccharomyces cerevisiae",
        "alias" : [ "KAP108", "S000002803", "SXM1" ],
        "shared_name" : "YDR395W",
        "SUID" : 660,
        "degree_layout" : 8,
        "name" : "YDR395W"
    },
    "position" : {
        "x" : 730.8733342488606,
        "y" : -157.50702317555744
    },
    "selected" : false
}, {
    "data" : {
        "id" : "579",
        "selected" : false,
        "annotation_Taxon" : "Saccharomyces cerevisiae",
        "alias" : [ "RPL11A", "S000006306", "ribosomal protein L11A (L16A) (rp39A) ↵
↳(YL22)" ],
        "shared_name" : "YPR102C",
        "SUID" : 579,
        "degree_layout" : 2,
```

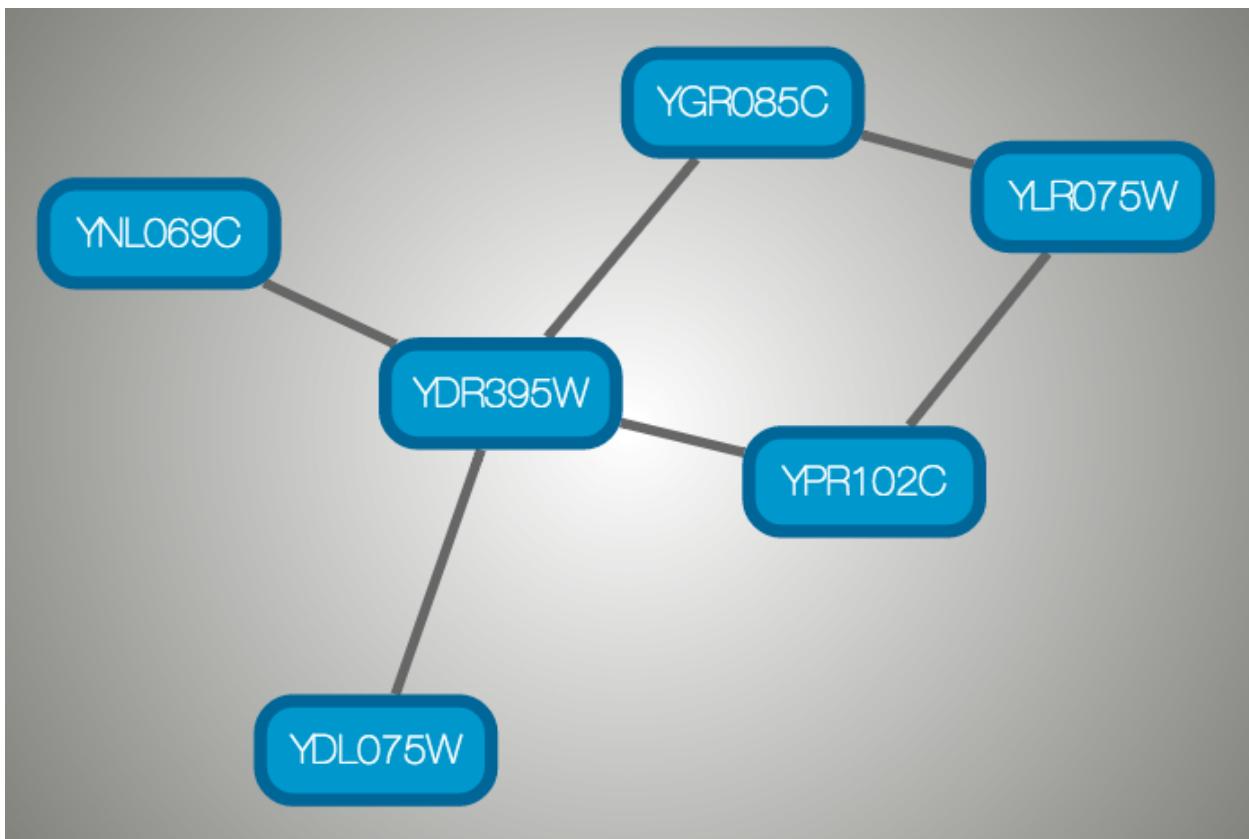
```

        "name" : "YPR102C"
    },
    "position" : {
        "x" : 841.1395696004231,
        "y" : -130.77909119923908
    },
    "selected" : false
}, {
    "data" : {
        "id" : "578",
        "selected" : false,
        "annotation_Taxon" : "Saccharomyces cerevisiae",
        "alias" : [ "GRC5", "QSR1", "RPL10", "S000004065", "ribosomal protein L10" ],
        "shared_name" : "YLR075W",
        "SUID" : 578,
        "degree_layout" : 2,
        "name" : "YLR075W"
    },
    "position" : {
        "x" : 910.3755162556965,
        "y" : -217.0562556584676
    },
    "selected" : false
} ],
"edges" : [ {
    "data" : {
        "id" : "659",
        "source" : "658",
        "target" : "578",
        "selected" : false,
        "interaction" : "pp",
        "shared_interaction" : "pp",
        "shared_name" : "YGR085C (pp) YLR075W",
        "SUID" : 659,
        "name" : "YGR085C (pp) YLR075W"
    },
    "selected" : false
}, {
    "data" : {
        "id" : "661",
        "source" : "658",
        "target" : "660",
        "selected" : false,
        "interaction" : "pp",
        "shared_interaction" : "pp",
        "shared_name" : "YGR085C (pp) YDR395W",
        "SUID" : 661,
        "name" : "YGR085C (pp) YDR395W"
    },
    "selected" : false
}, {
    "data" : {
        "id" : "724",
        "source" : "660",
        "target" : "723",
        "selected" : false,
        "interaction" : "pp",
        "shared_interaction" : "pp",
        "shared_name" : "YGR085C (pp) YDR395W",
        "SUID" : 724,
        "name" : "YGR085C (pp) YDR395W"
    },
    "selected" : false
}
]

```

```
        "shared_name" : "YDR395W (pp) YDL075W",
        "SUID" : 724,
        "name" : "YDR395W (pp) YDL075W"
    },
    "selected" : false
}, {
    "data" : {
        "id" : "733",
        "source" : "660",
        "target" : "579",
        "selected" : false,
        "interaction" : "pp",
        "shared_interaction" : "pp",
        "shared_name" : "YDR395W (pp) YPR102C",
        "SUID" : 733,
        "name" : "YDR395W (pp) YPR102C"
    },
    "selected" : false
}, {
    "data" : {
        "id" : "727",
        "source" : "660",
        "target" : "726",
        "selected" : false,
        "interaction" : "pp",
        "shared_interaction" : "pp",
        "shared_name" : "YDR395W (pp) YNL069C",
        "SUID" : 727,
        "name" : "YDR395W (pp) YNL069C"
    },
    "selected" : false
}, {
    "data" : {
        "id" : "580",
        "source" : "578",
        "target" : "579",
        "selected" : false,
        "interaction" : "pp",
        "shared_interaction" : "pp",
        "shared_name" : "YLR075W (pp) YPR102C",
        "SUID" : 580,
        "name" : "YLR075W (pp) YPR102C"
    },
    "selected" : false
} ]
}
}
```

And this is a sample visualization in Cytoscape.js:



7.10.1 Important Note

Export network and table to Cytoscape.js feature in Cytoscape creates a JSON file **WITHOUT** style. This means that you need to export the style in a separate JSON file if you apply style to your network. Please read the [Style](#) section for more details.

7.11 Cytoscape CX

CX is a JSON-based transfer format that enables diverse Cytoscape Cyberinfrastructure (CI) services to exchange networks while preserving all network-related information. It is designed for flexibility, modularity, and extensibility, and as a message payload in common CI REST protocols. It enables applications to standardize on core aspects of networks, coordinate on more specific or unique standards, and to ignore or omit irrelevant aspects. It is not intended as an optimized format for storage or for specific functionality in applications.

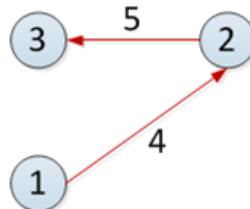
CX is an Aspect-Oriented Network Interchange Format, where the base information is a list of nodes. Independent data structures (called aspects) organize and elaborate on nodes and each other. The core of CX defines five aspects, though a more comprehensive [CX document](#) describes many more aspects.

The “nodes” aspect contains only the identifiers of the network’s nodes, The “edges” aspect contains identifiers for each edge along with the identifiers of the nodes the edge connects. The “networkAttributes” aspect contains name-value pairs describing the network. The “nodeAttributes” and “edgeAttributes” aspects contain name-value pairs attached to specifically identified nodes and edges.

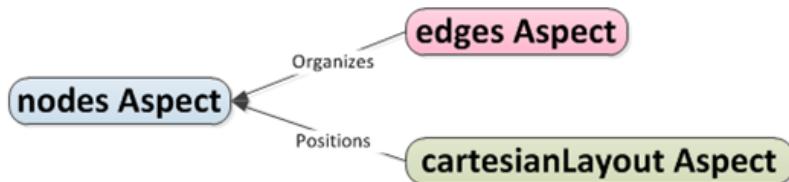
Critically, applications are free to add and maintain their own aspects without coordinating or negotiating with disinterested applications.

As an illustration using the picture below, a three node network can be described as a list of nodes (“nodes Aspect”) and edges that link them (“edges Aspect”). If the network has been laid out, a separate aspect (“cartesianLayout Aspect”) can describe the position of each node. More concretely, a CX encoding would have three nodes in the “nodes Aspect”, each with unique IDs. The “edges Aspect” references each node by ID, with each edge having its own ID. Finally, the “cartesianLayout Aspect” ties coordinates to nodes by ID. In fact, a network may have many aspects, describing node and edge attributes, subnetworks, visual properties, groups and so on.

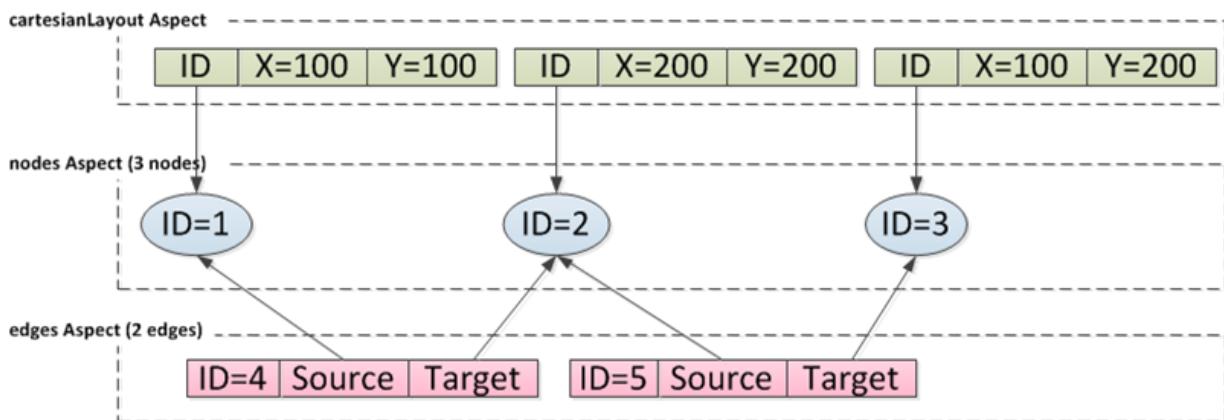
Example Graph



Aspect Relationships



CX Encoding



The actual JSON encoding for a CX stream is described in the [CX document](#). It would appear something like this:

```
{
  "nodes": [{"@id": 1}, {"@id": 2}, {"@id": 3}],
  "edges": [{"s": 1, "@id": 4, "t": 2},
            {"s": 2, "@id": 5, "t": 3}],
  "cartesianLayout": [{"x": 100, "node": 1, "y": 100},
                      {"x": 200, "node": 2, "y": 200},
                      {"x": 100, "node": 3, "y": 200}]
}
```

CHAPTER 8

Node and Edge Column 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 information to Cytoscape as node/edge/network data columns. This could include, for example, annotation data on a gene or confidence values in a protein-protein interaction. These column data can then be visualized in a user-defined way by setting up a mapping from columns to network properties (colors, shapes, and so on). The section on [Styles](#) discusses this in greater detail.

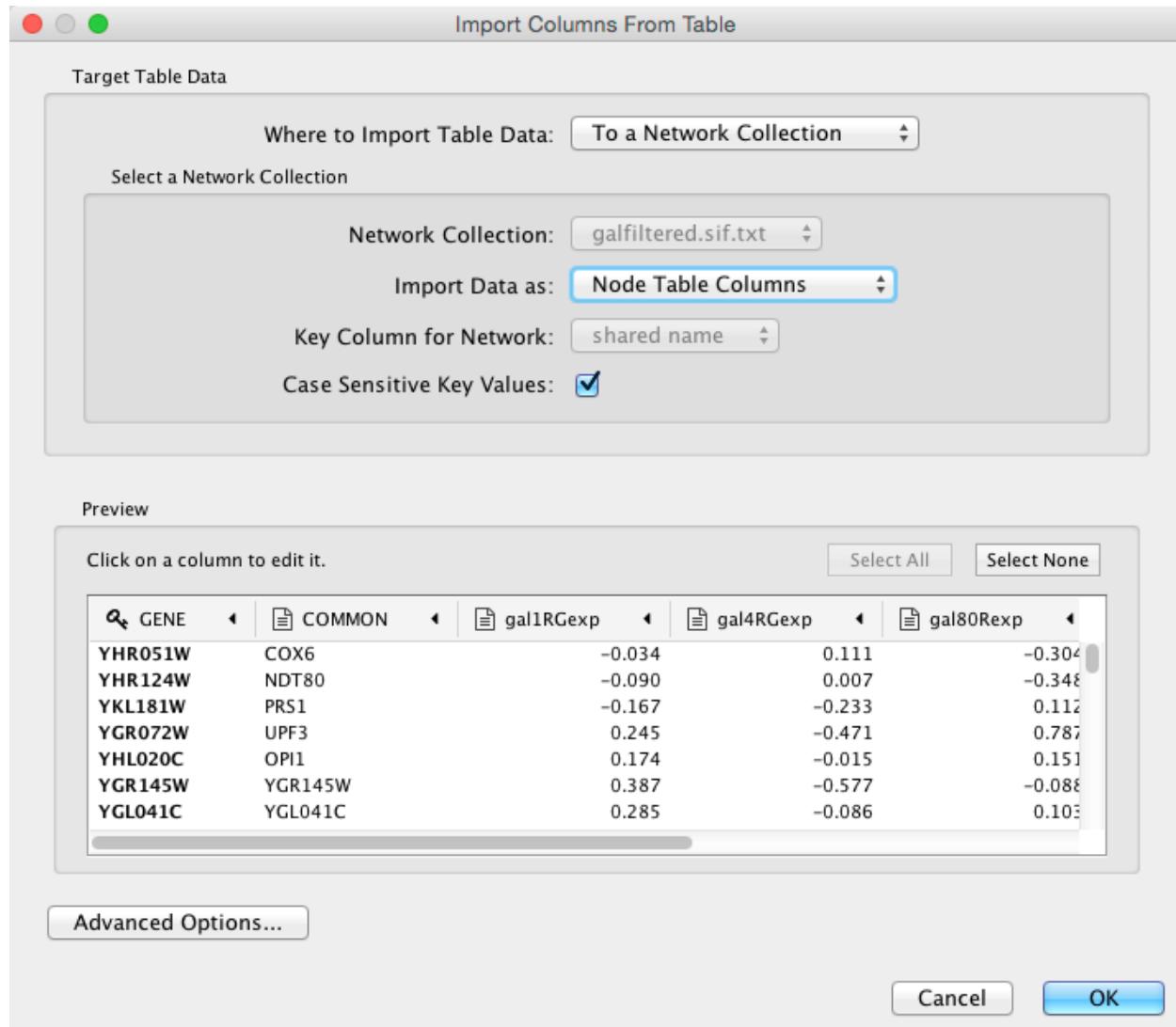
8.1 Import Data Table Files

Cytoscape offers support for importing data from delimited text and MS Excel data tables.

Sample Data Table 1

The data table file should contain a primary key column and at least one data column. The maximum number of data columns is unlimited. The **Alias** column is an optional feature, as is using the first row of data as column names. Alternatively, you can specify each column name from the **File → Import → Table → File...** user interface.

8.1.1 Basic Operation



1. Select **File → Import → Table → File....**
2. Select a data file. The file can be either a text or Excel (.xls/.xlsx) file. Alternatively, you can simply drag and drop the file from the desktop into the **Table Panel**.
3. In the **Target Table Data** section, choose where to import the data to. You can choose an existing network collection, a specific network only, or you can choose to import the data to an **Unassigned Table** (described below).
4. Depending on what you choose in the **Where to import Table Data** drop-down, you will need to select a network collection or specific network. You will also need to select **Importing Type**, that is whether the data is node, edge or network table columns.
5. If the table is not properly delimited in the preview panel, change the delimiter in the **Advanced Options** panel. The default delimiter is tab. This step is not necessary for Excel Workbooks.
6. By default, the first column is designated as the primary key. Change the key column if necessary.
7. Click **OK** to import.

Unassigned Table

As of Cytoscape 3.1, it is possible to import data tables without assigning them to existing networks, meaning the data doesn't have to correspond to any nodes/edges currently loaded. If a data table is imported as unassigned and a network is later imported that maps to the data in terms of nodes or edges, the data will link automatically. This is useful when loading a large dataset (for example expression data), defining a **Style** for visualizing the data on networks and later loading individual networks to view the data, for example from an online database. This feature allows the data to be automatically linked to any network that is applicable, without having to load the data for each network.

8.2 Legacy Cytoscape Attributes Format

In addition to tabular data, the simple attribute file format used in previous versions of Cytoscape is still supported. Node and edge data files are simply formatted: a node data file begins with the name of the column on the first line (note that it cannot contain spaces). Each following line contains the name of the node, followed by an equals sign and the data value. Numbers and text strings are the most common data types. All values for a given column must have the same type. For example:

```
FunctionalCategory
YAL001C = metabolism
YAR002W = apoptosis
YBL007C = ribosome
```

An edge data file has much the same structure, except that the name of the edge is the source node name, followed by the interaction type in parentheses, followed by the target node name. Directionality counts, so switching the source and target will refer to a different (or perhaps non-existent) edge. The following is an example edge data file:

```
InteractionStrength
YAL001C (pp) YBR043W = 0.82
YMR022W (pd) YDL112C = 0.441
YDL112C (pd) YMR022W = 0.9013
```

Since Cytoscape treats edge data as directional, the second and third edge data values refer to two different edges (source and target are reversed, though the nodes involved are the same).

Each data column is stored in a separate file. Node and edge data files use the same format, and have the suffix ".attrs".

Node and edge data may be loaded via the **File → Import → Table** menu, just as data table files are.

When expression data is loaded using an expression matrix, it is automatically loaded as node data unless explicitly specified otherwise.

Node and edge data columns are attached to nodes and edges, and so are independent of networks. Data values for a given node or edge will be applied to all copies of that node or edge in all loaded network files, regardless of whether the data file or network file is imported first.

8.2.1 Detailed file format (Advanced users)

Every data file has one header line that gives the name of the data column, and optionally some additional meta-information about that data column. The format is as follows:

```
columnName (class=JavaClassName)
```

The first field is always the column name: it cannot contain spaces. If present, the class field defines the name of the class of the data values. For example, java.lang.String or String for Strings, java.lang.Double or Double for floating point values, java.lang.Integer or Integer for integer values, etc. If the value is actually a list of values, the

class should be the type of the objects in the list. If no class is specified in the header line, Cytoscape will attempt to guess the type from the first value. If the first value contains numbers in a floating point format, Cytoscape will assume `java.lang.Double`; if the first value contains only numbers with no decimal point, Cytoscape will assume `java.lang.Integer`; otherwise Cytoscape will assume `java.lang.String`. Note that the first value can lead Cytoscape astray: for example,

```
floatingPointDataColumn
firstName = 1
secondName = 2.5
```

In this case, the first value will make Cytoscape think the values should be integers, when in fact they should be floating point numbers. It's safest to explicitly specify the value type to prevent confusion. A better format would be:

```
floatingPointDataColumn (class=Double)
firstName = 1
secondName = 2.5
```

or

```
floatingPointDataColumn
firstName = 1.0
secondName = 2.5
```

Every line past the first line identifies the name of an object (a node in a node data file or an edge in a edge data file) along with the String representation of the data value. The delimiter is always an equals sign; whitespace (spaces and/or tabs) before and after the equals sign is ignored. This means that your names and values can contain whitespace, but object names cannot contain an equals sign and no guarantees are made concerning leading or trailing whitespace. Object names must be the Node ID or Edge ID as seen in the left-most column of the Table Panel if the data column is to map to anything. These names must be reproduced exactly, including case, or they will not match.

Edge names are all of the form:

```
sourceName (edgeType) targetName
```

Specifically, that is

```
sourceName space openParen edgeType closeParen space targetName
```

Note that tabs are not allowed in edge names. Tabs can be used to separate the edge name from the “=” delimiter, but not within the edge name itself. Also note that this format is different from the specification of interactions in the SIF file format. To be explicit: a SIF entry for the previous interaction would look like

```
sourceName edgeType targetName
```

or

```
sourceName whiteSpace edgeType whiteSpace targetName
```

To specify lists of values, use the following syntax:

```
listDataColumnName (class=java.lang.String)
firstObjectName = (firstValue::secondValue::thirdValue)
secondObjectName = (onlyOneValue)
```

This example shows a data column whose value is defined as a list of text strings. The first object has three strings, and thus three elements in its list, while the second object has a list with only one element. In the case of a list every data value uses list syntax (i.e. parentheses), and each element is of the same class. Again, the class will be inferred if it is not specified in the header line. Lists are not supported by Styles and so can't be mapped to network properties.

8.2.2 Newline Feature

Sometimes it is desirable to for data values to include linebreaks, such as node labels that extend over two lines. You can accomplish by inserting into the data value. For example:

```
newlineDataColumn
YJL157C = This is a long\nline for a label.
```

8.2.3 Table Panel

SUID	selected	shared name	name	COMMON	gal1RGexp	gal4RGexp	gal80Rexp	gal1RGsig	gal4RGsig	gal80Rsig
62	<input type="checkbox"/>	YKR026C	GCN3		-0.154	-0.501	0.292	9.117E-4	3.5692E-6	0.011229
63	<input type="checkbox"/>	YGL122C	NAB2		0.174	0.02	0.187	8.7295E-4	0.61707	0.0059966
65	<input type="checkbox"/>	YCR218W	CRM1		-0.018	-0.001	-0.018	0.61381	0.9794	0.80969
66	<input type="checkbox"/>	YGL097W	SRM1		0.16	-0.23	0.008	0.0021913	0.0022461	0.93826
68	<input type="checkbox"/>	YOR204W	DED1		-0.033	-0.056	-0.91	0.39944	0.31268	8.349E-16
70	<input type="checkbox"/>	YLR249W	YEF3		-0.39	-0.394	-0.769	2.713E-8	0.04747	0.035939
71	<input type="checkbox"/>	YPR080W	TEF1		-0.138	0.009	-0.278	9.8725E-4	0.89728	6.7798E-4
73	<input type="checkbox"/>	YBR118W	YBR118W		-0.074	-0.063	0.044	0.053125	0.15497	0.54556
75	<input type="checkbox"/>	YLR293C	GSP1		-0.242	-0.247	-0.128	1.7601E-5	6.7968E-4	0.012703
77	<input type="checkbox"/>	YMR146C	YMR146C		-0.05	-0.143	-0.151	0.30844	0.0085173	0.072007
78	<input type="checkbox"/>	YMR146C	TIF34		0.078	-0.290	0.044	0.073444	0.0011734	1.6641E-4

When Cytoscape is started, the **Table Panel** appears in the bottom right of the main Cytoscape window. This browser can be hidden and restored using the F5 key or the **View → Show/Hide Table Panel** menu option. Like other Panels, the browser can be undocked by pressing the little icon in the top right corner.

To swap between displaying node, edge, and network Data Tables use the tabs on the bottom of the Table Panel. By default, the Table Panel displays columns for all nodes and edges in the selected network. To display columns for only

selected nodes/edges, click the **Change Table Mode** button at the top left. To change the columns that are

displayed, click the **Show Column** button and choose the columns that are to be displayed (select various columns by clicking on them, and then click elsewhere on the screen to close the column list).

Most column values can be edited by double-clicking the cell (only the ID cannot be edited). Newline characters can be inserted into String columns 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.

Rows in the panel can be sorted alphabetically by specific column by clicking on a column heading. A new column



can be created using the **Create New column** button, and must be one of four types - integer, string, real



number (floating point), or boolean. Columns can be deleted using the **Delete Columns...** button. **NOTE: Deleting columns removes them from Cytoscape, not just the Table Panel!** To remove columns from the panel



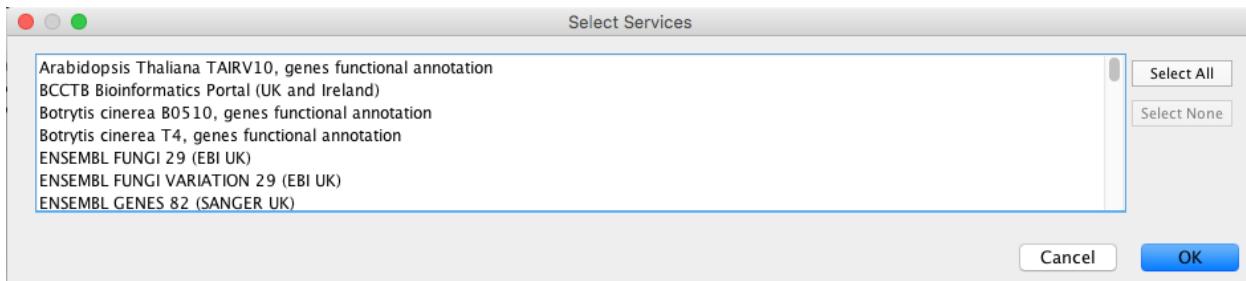
without deleting them, simply unselect the column using the **Select Columns** button.

8.3 Import Data Table from Public Databases

It is also possible to import node data columns from public databases via web services, for example from BioMart (<http://www.biomart.org>).

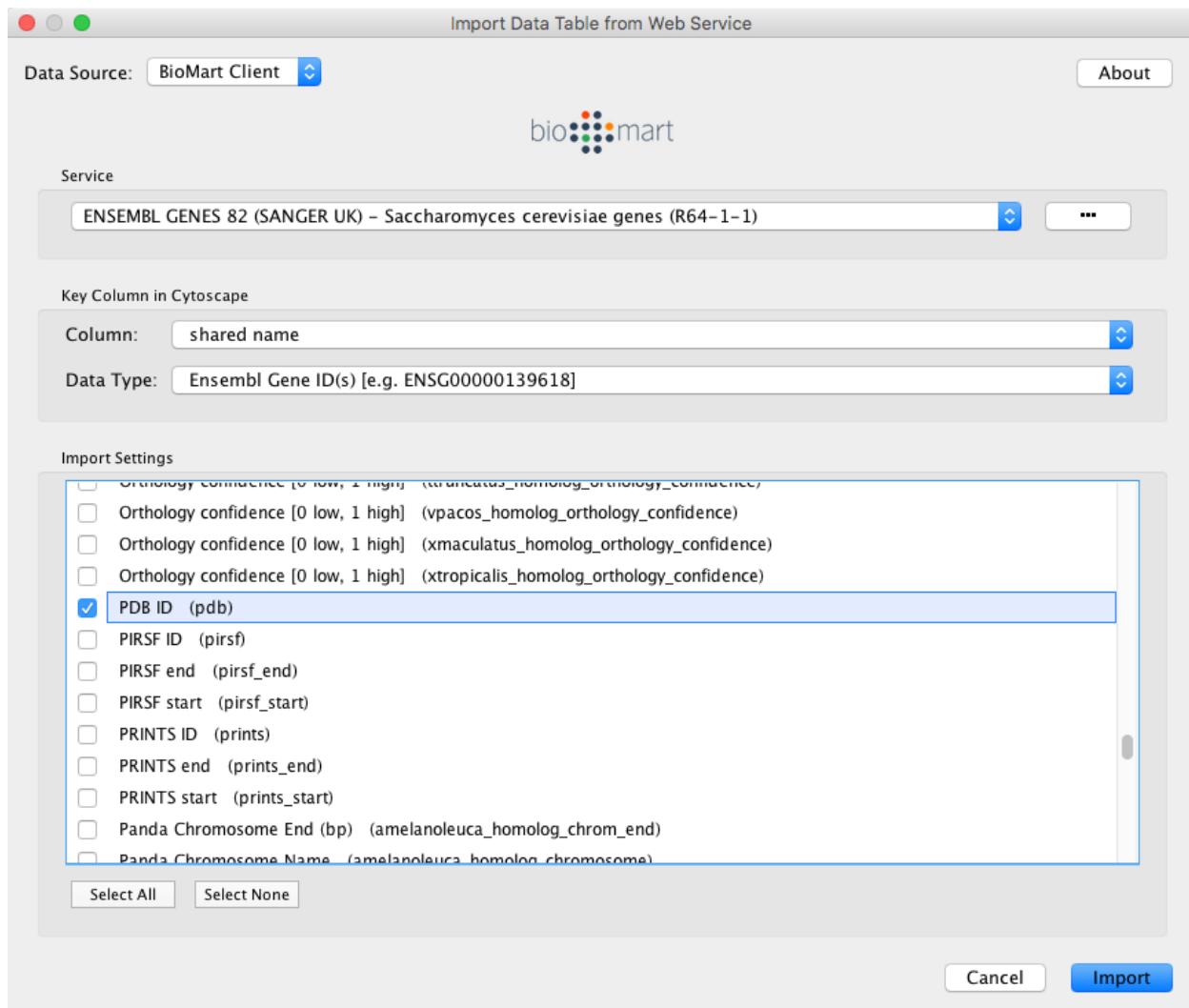
8.3.1 Basic Operation

1. Load a network, for example galFiltered.sif.
2. Select **File → Import → Table → Public Databases...**
3. You will first be asked to select from a set of web services. For this example, we will choose **ENSEMBL GENES 73 (SANGER UK)**.



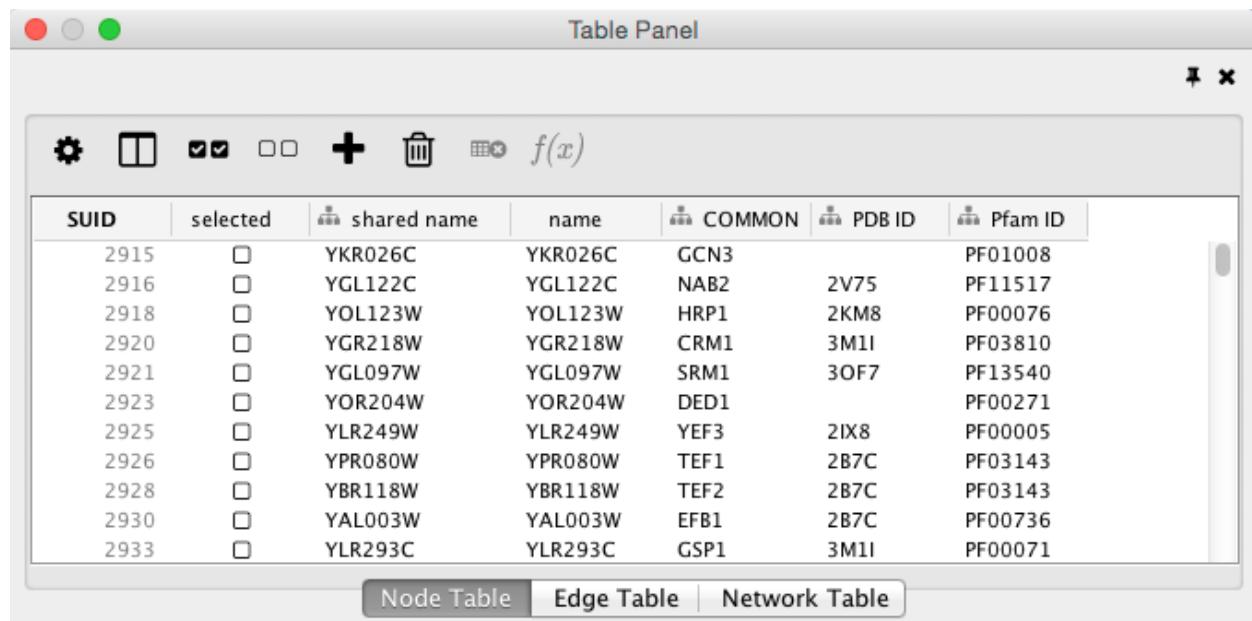
1. In the **Import Data Table from Web Services** dialog, select a **Data Source**. Since galFiltered.sif is a yeast network, select **ENSEMBL GENES - SACCHAROMYCES CEREVISIAE**.
2. For **Key Column in Cytoscape**, select *shared name* for **Column** and *Ensembl Gene ID* for **Data Type**.

The type of identifier selected under **Data Type** must match what is used in the selected **Column** in the network.



1. Select the data columns you want to import.
2. Select **Import**.

When import is complete, you can see the newly imported data columns in the Table Panel.

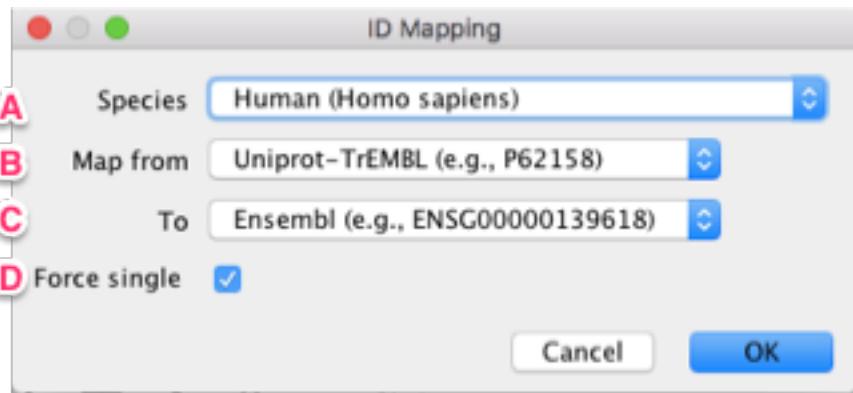


8.4 Mapping Identifiers

A common problem in integrating multiple data sources is differences in terminology. Standardized identifiers have many dialects. Some databases are specific to a particular organism or genome. Others span those boundaries. **BridgeDB** is a web service devoted to solving the id mapping problem. Cytoscape contains simple access to BridgeDB. If you need more esoteric species or data sources, there is an [installable BridgeDB app](#) to access a fuller feature set.

Ensembl is used as the canonical taxonomy. A translation between two arbitrary data sources is generally achieved by mapping each through Ensembl.

To map an identifier from one source to another, right click on the column header of your identifier. Select the option to **Map Column...**



A. The mapping is always constrained by species to prevent senseless matches across species. You must choose a species (once) for this feature to function properly. The choices for the Data Source and Target are determined by the species.

B. Based on the items in the column you chose we can make a reasonable assumption as to the database as to use as the source for your mapping. If this is not the source, you can override it in the Map from field.

C. The target database of the identifier mapping. This list is filtered by species and curated down to the most common options. A full list of supported targets is [here](#).

D. In some cases where there are multiple answers, the **Force Single** option limits the result to the first answer returned by the service. If the option is off, a list of results will appear in the column.

CHAPTER 9

Column Data Functions and Equations

9.1 Column Formulas

9.1.1 Introduction

Column data values may be formulas. A typical example is `=ABS($otherColumn + LOG(10.2))`. Formulas are modeled after Excel(tm) but only support references to other columns at the same node, edge or network. Since Cytoscape column names may contain embedded spaces, optional braces around the column name (required if the name is not simply a letter followed by one or more letters or digits) is allowed e.g. `${a name with spaces}`. Backslashes, opening braces and dollar signs in column names have to be escaped with a leading backslash. For example the column name `ex$am{p\le` would have to be written as `${ex$am{p\\le}`. Finally, column 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 “\\“W“. Formula results must be compatible with the type of the column 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) column data where non-zero will be interpreted as **true** and zero as **false**. For integer columns, floating point values will be converted using the rules of the Excel(tm) **INT** function. Parentheses can be used for grouping and to change evaluation order. The operator precedence rules follow those of standard arithmetic.

9.1.2 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).

9.1.3 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 p.
- 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.

9.1.4 Pitfalls

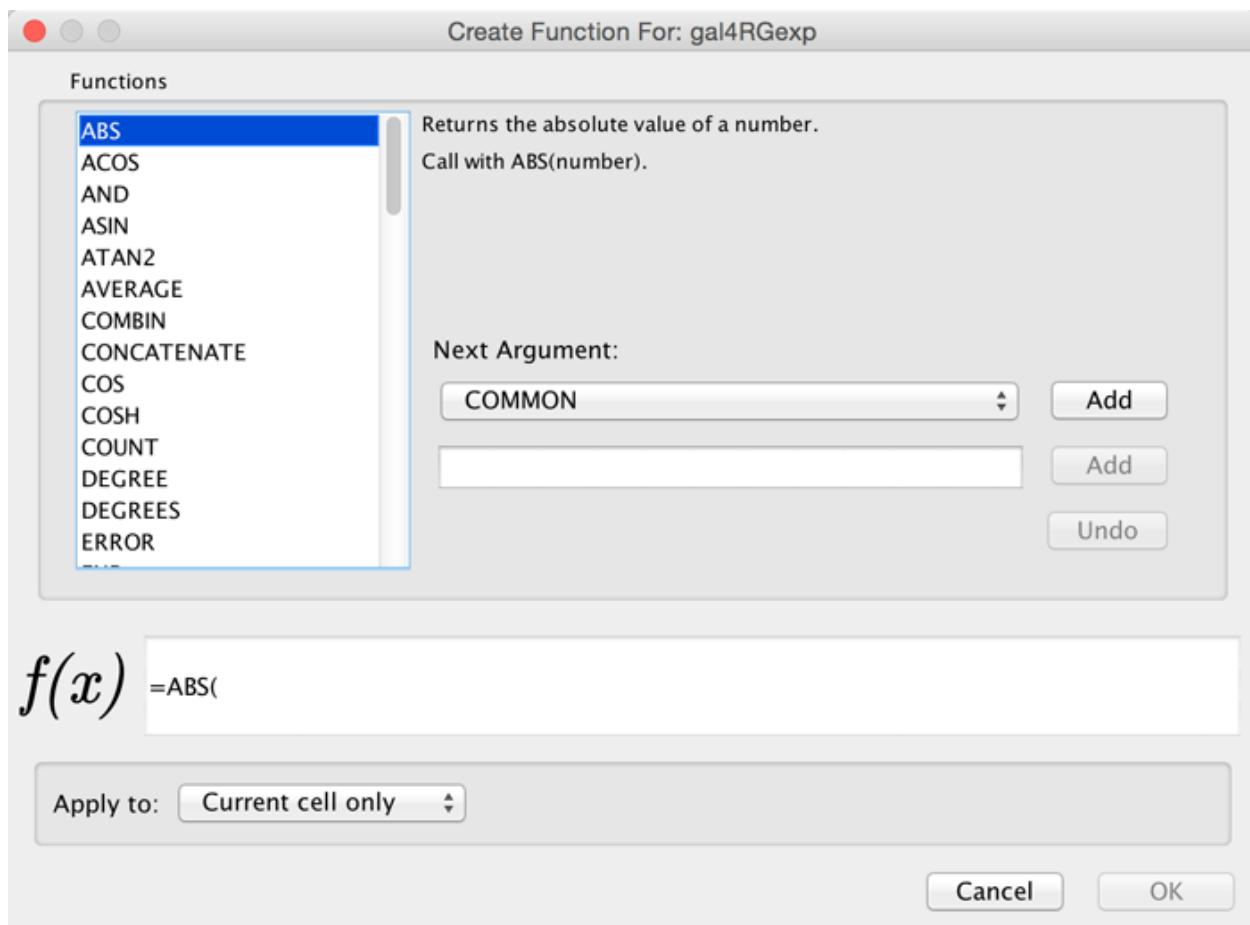
The possibly biggest problem is the referencing of other columns that have null values. This is not allowed and leads to errors. In order to mitigate this problem we support the following optional syntax for column references: `${columnName:defaultValue}` . The interpretation is that if `columnName`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

9.1.5 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.

9.2 The Formula Builder

In order to ease the creation of formulas as well as to facilitate discovery of built-in functions we provide a **Function Builder** in the Table Panel. After selecting a non-list column cell, you can invoke it by clicking on  . This should bring up the Function Builder which looks like this:



Select a function on the left hand side of the dialog - here, we've selected the ABS function. Next to the list of functions, you can specify one or more arguments. This can either be a column (selected from the drop-down list) or a constant specified in the box below. If you select a column, the value of that column (in the row containing the formula) will be used, and the function result will be updated dynamically when that value changes. Click **Add** to add an argument - you can add one or more depending on how many arguments the function accepts. At the bottom of the dialog is a preview of the current formula. Under **Apply to**, you can select whether the formula will apply to the current cell only, the cell selection, or the entire column. Click **OK** when you are satisfied with the result, or **Cancel** to discard any changes.

The Function 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 columns, 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.

9.3 A Note for App Writers

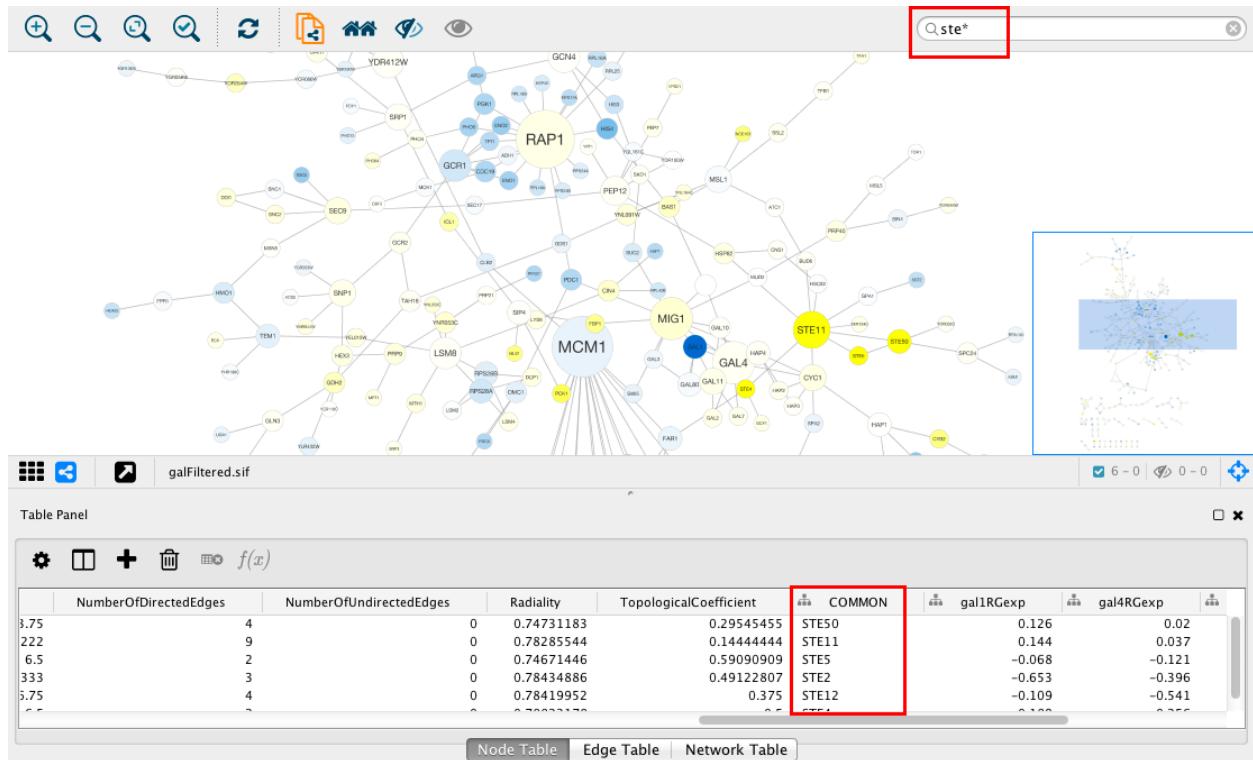
It is relatively easy to add your own built-in formula functions. A simple function can probably be implemented in 15 to 20 minutes. It can then be registered via the parser and becomes immediately available to the user. It will of course also show up in the drop-down list in the Function Builder.

CHAPTER 10

Finding and Filtering Nodes and Edges

10.1 Search Bar

You can search for nodes and edges by column value directly through Cytoscape's tool bar. For example, to select nodes or edges with a column value that starts with "STE", type `ste*` in the search bar. The search is case-insensitive. The * is a wildcard character that matches zero or more characters, while "?" matches exactly one character. So `ste?` would match "STE2" but would not match "STE12". Searching for `ste*` would match both.



To search a specific column, you can prefix your search term with the column name followed by a `:`. For example, to select nodes and edges that have a “COMMON” column value that starts with “STE”, use `common:ste*`. If you don’t specify a particular column, all columns will be searched.

Columns with names that contain spaces, quotes, or characters other than letters and numbers currently do not work when searching a specific column. This will be fixed in a future release.

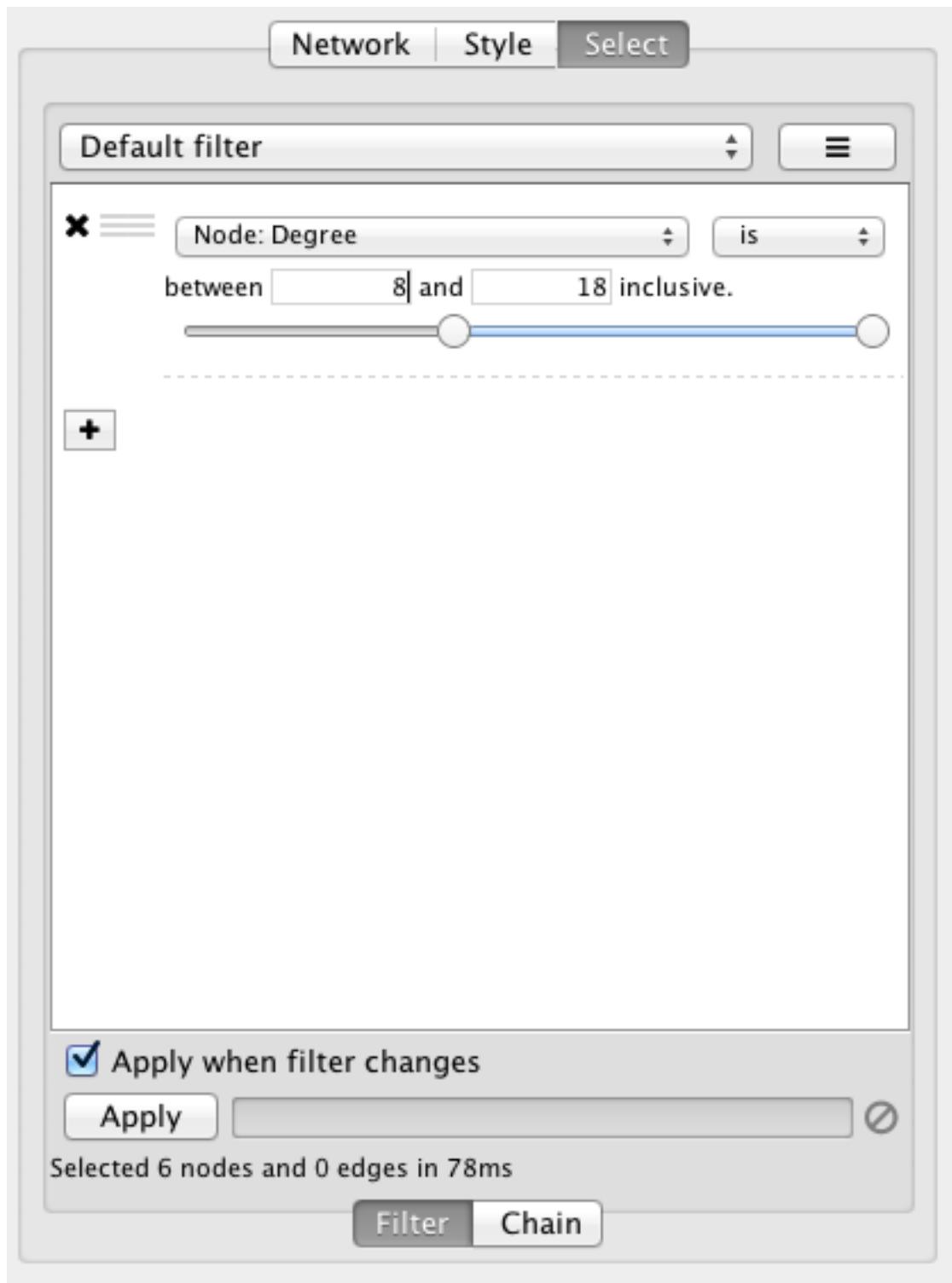
To search for column values that contain special characters you need to escape those characters using a “\”. For example, to search for “GO:1232”, use the query `GO\ : 1232`. The complete list of special characters is:

```
+ - & | ! ( ) { } [ ] ^ " ~ * ? : \
```

Note: Escaping characters only works when searching all columns. It currently does not work for column-specific searching. This will be fixed in a future release.

10.2 Filters

Cytoscape 3 provides a new user interface for filtering nodes and edges. These tools can be found in the **Select** panel:



There are two types of filters. On the **Filter** tab are *narrowing* filters, which can be combined into a tree. On the **Chain** tab are *chainable* filters, which can be combined in a linear chain.

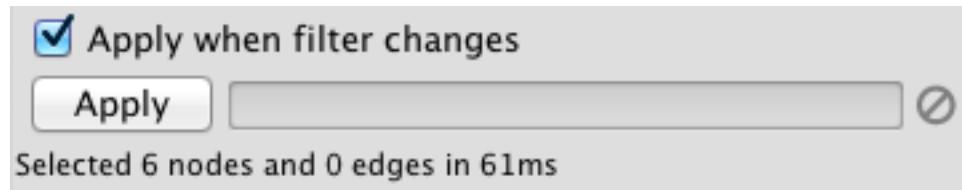
10.2.1 Narrowing Filters

Narrowing filters are applied to the entire network, and are used to select a subset of nodes or edges in a network based on user-specified constraints. For example, you can find edges with a weight between 0 and 5.5, or nodes with degree less than 3. A filter can contain an arbitrary number of sub-filters.

To add a filter click on the “+” button. To delete a filter (and all its sub-filters) click the “x” button. To move a filter grab the handle  with the mouse and drag and drop the filter on its intended destination. Dropping a filter on top of another filter will group the filters into a composite filter.

Interactive Filter Application Mode

Due to the nature of narrowing filters, Cytoscape can apply them to a network efficiently and interactively. Some filters even provide slider controls to quickly explore different thresholds. This is the default behavior on smaller networks. For larger networks, Cytoscape automatically disables this interactivity. You can override this by manually checking the **Apply when filter changes** box above the **Apply** button:

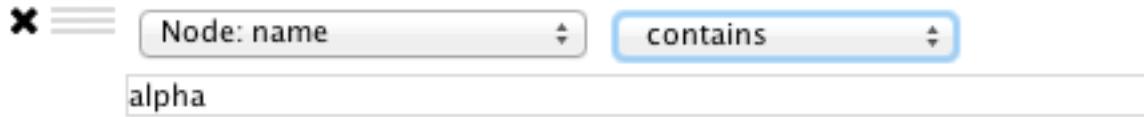


Cytoscape comes packaged with the following narrowing filters:

Column Filter

This filter will match nodes or edges that have particular column values. For numeric columns sliders are provided to set minimum and maximum values, or the values may be entered manually.

From string columns, a variety of matching options are provided:



For example, column values can be checked to see if they contain or match exactly the text entered in the text box. More complex matching criteria can be specified by using a Java-style regular expression.

By default string matching is case insensitive. Case sensitive matching requires the use of a regular expression that starts with “(?-i)”. For example to match the text “ABC” in a case sensitive way use the following regular expression: “(?-i)ABC”.

Cytoscape uses Java regular expression syntax.

Degree Filter

The degree filter matches nodes with a degree that falls within the given minimum and maximum values, inclusive. You can choose whether the filter operates on the in-degree, out-degree or overall (in + out) degree.

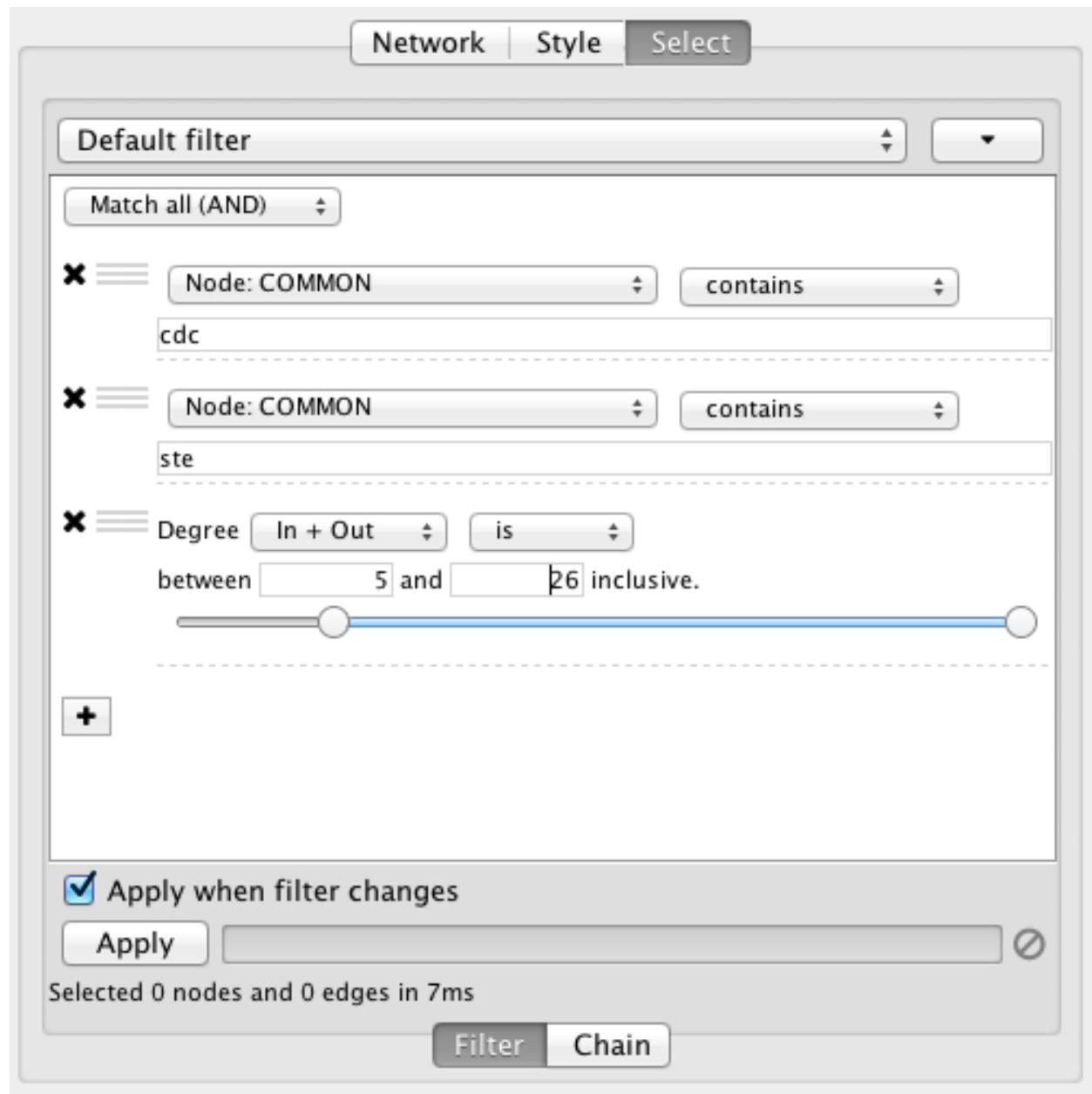
Topology Filter

The topology filter matches nodes having a certain number of neighbors which are within a fixed distance away, and which match a sub-filter. The thresholds for the neighborhood size and distance can be set independently, and the sub-filter is applied to each such neighbor node.

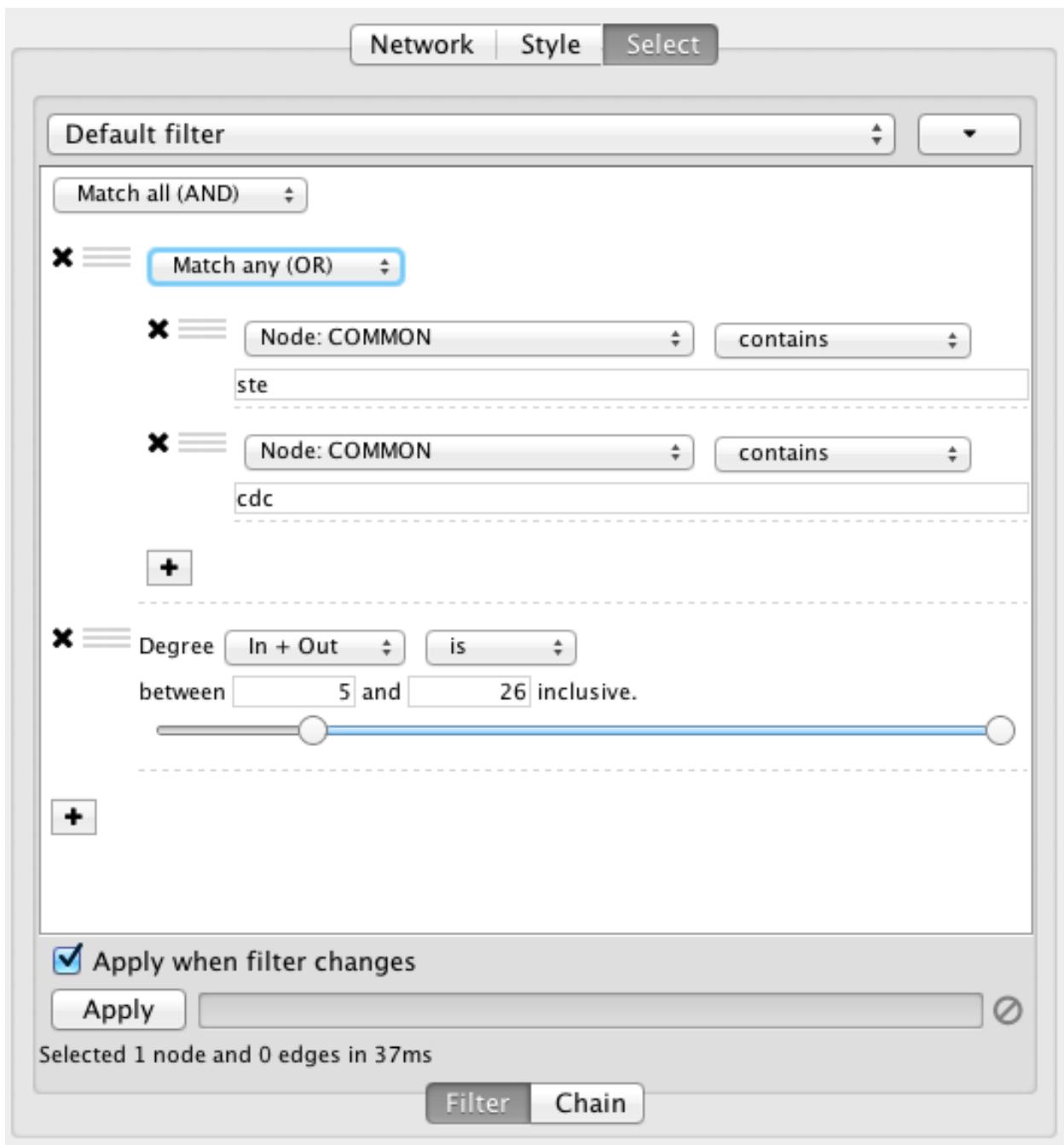
The topology filter will successfully match a node if the sub-filter matches against the required number of neighbor nodes.

Grouping and Organizing Filters

By default, nodes and edges need to satisfy the constraints of all your filters. You can change this so that instead, only the constraints of at least one filter needs to be met in order to match a node or edge. This behavior is controlled by the **Match all/any** drop-down box. This appears once your filter has more than one sub-filter. For example, suppose you wanted to match nodes with column *COMMON* containing `ste` or `cdc`, but you only want nodes with degree 5 or more, you'd first construct a filter that looks like this:



This filter will match nodes where **COMMON** contains **ste** **and** **cdc**. To change this to a logical **or** operation, drag either of the column filters by its handle onto the other column filter to create a new group. Now change the group's matching behavior to **Match any**:

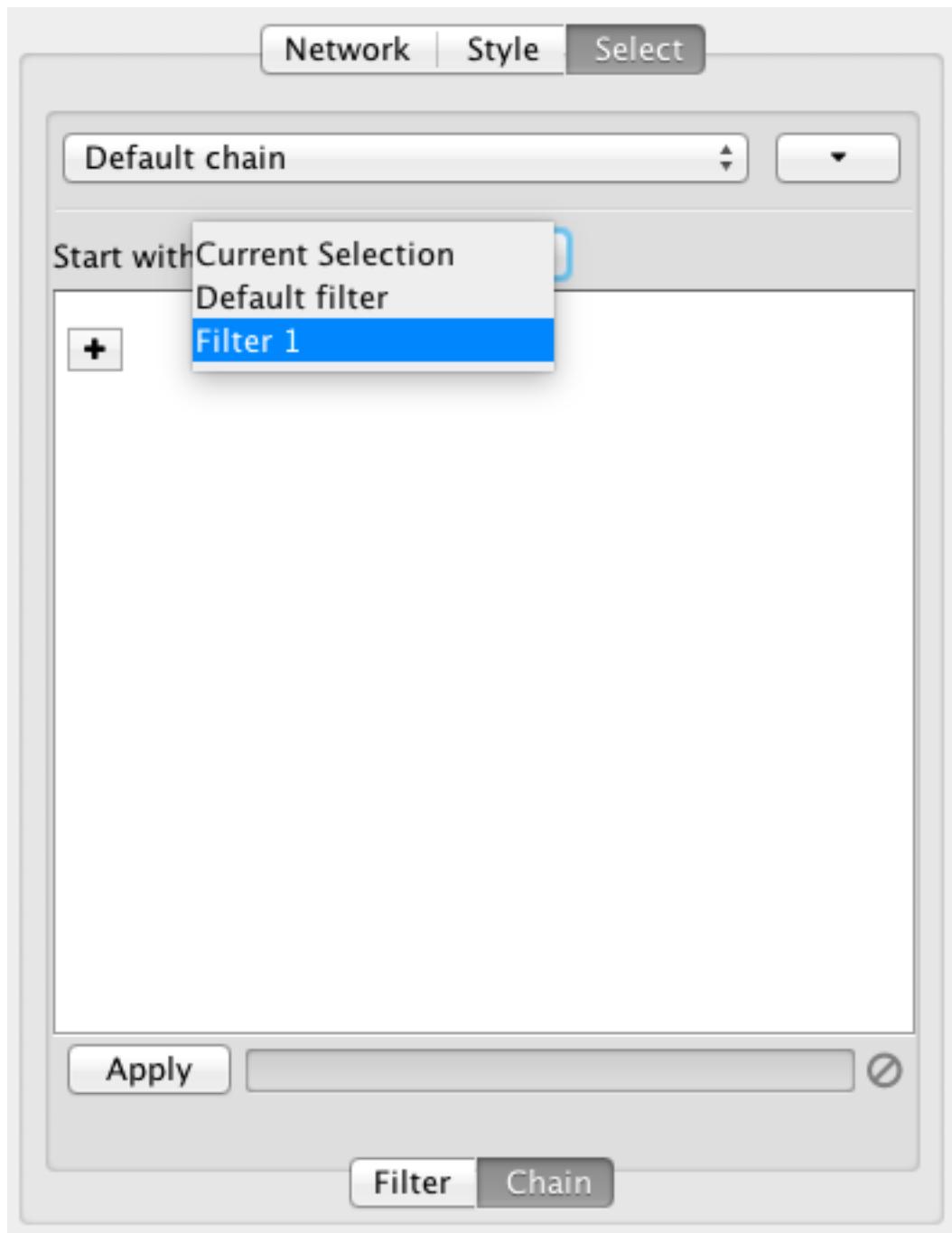


You can also reorder filters by dropping them in-between existing filters.

10.2.2 Chainable Filters

Chainable filters are combined in an ordered list. The nodes and edges in the output of a filter become the input of the next filter in the chain. The first filter in the chain gets its input from the current selection or from a filter on the **Filter** tab. The output of the last filter becomes the new selection.

You can specify the input to the first filter in the chain by selecting a **Start with**, where **Current selection** refers to the nodes and edges currently selected. You can also choose a narrowing filter, which produces a different set of selected nodes and edges.



Chainable filters can be reordered by dragging one by the handle and dropping it between existing filters.

Cytoscape currently bundles the following chainable filters:

Edge Interaction Transformer

This transformer will go through all the input edges and selectively add their source nodes, target nodes, or both, to the output. This is useful for adding nodes that are connected to edges that match a particular filter.

Output options:

- Add (default): Automatically includes all input nodes and edges in the output, and adds source or target nodes from input edges to the output.
- Replace with: Does not automatically include input nodes and edges in the output. Only outputs nodes that match the filter.

A sub-filter may be added as well. When a sub-filter is present the source/target nodes must match the filter to be included in the output.

Node Adjacency Transformer

This transformer is used to add nodes and edges that are adjacent to the input nodes. A sub-filter may be specified as well.

Note that pressing the **Apply** button repeatedly may cause the selection to continuously expand. This allows adjacent nodes that are at greater distances to be added.

Output options:

- Add (default): Automatically includes all input nodes and edges in the output, and adds selected adjacent nodes and edges.
- Replace with: Only outputs the adjacent nodes/edges.

Select options:

- Adjacent nodes: Output nodes that are adjacent to the input nodes.
- Adjacent edges: Output edges that are adjacent (incident) to the input nodes.
- Adjacent nodes and edges (default): Output both nodes and edges that are adjacent to the input nodes.

Edge direction options. (Hidden by default, click the small arrow icon to reveal.):

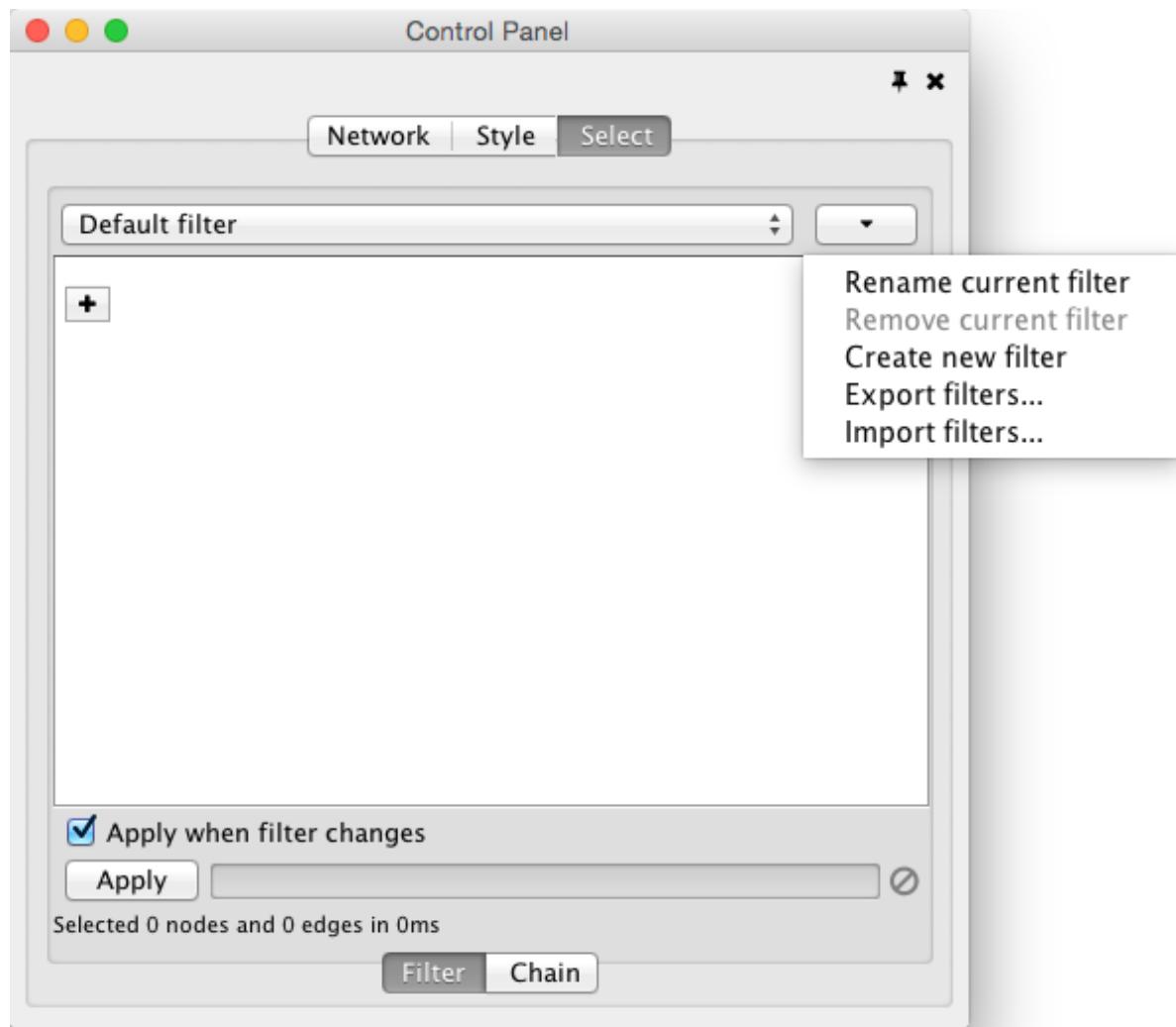
- Incoming: Only include adjacent nodes/edges when the adjacent edge is incoming.
- Outgoing: Only include adjacent nodes/edges when the adjacent edge is outgoing.
- Incoming and Outgoing (default): Ignore the directionality of adjacent edges.

Sub-filter options. (Available when a sub-filter has been added.):

- Adjacent nodes (default): The sub-filter is only applied to adjacent nodes. (Edges to the adjacent nodes are still included in the output.)
- Adjacent edges: The sub-filter is only applied to adjacent edges. (Nodes connected to the adjacent edges are still included in the output.)
- Adjacent nodes and edges: Both the adjacent edge and its connected node must match the filter. Note that for a filter to match an edge and a node at the same time it should be a compound filter that is set to “Match any (OR)”.

10.2.3 Working with Narrowing and Chainable Filters

The name of active filter appears in the drop-down box at the top of **Select** panel. Beside this is the options button which will allow you to rename, remove or export the active filter. It also lets you create a new filter, or import filters.



At the bottom of the **Select** panel, there is an **Apply** button that will re-apply the active filter. On the opposite side of the progress bar is the cancel button, which will let you interrupt a long-running filter.

10.3 Diffusion

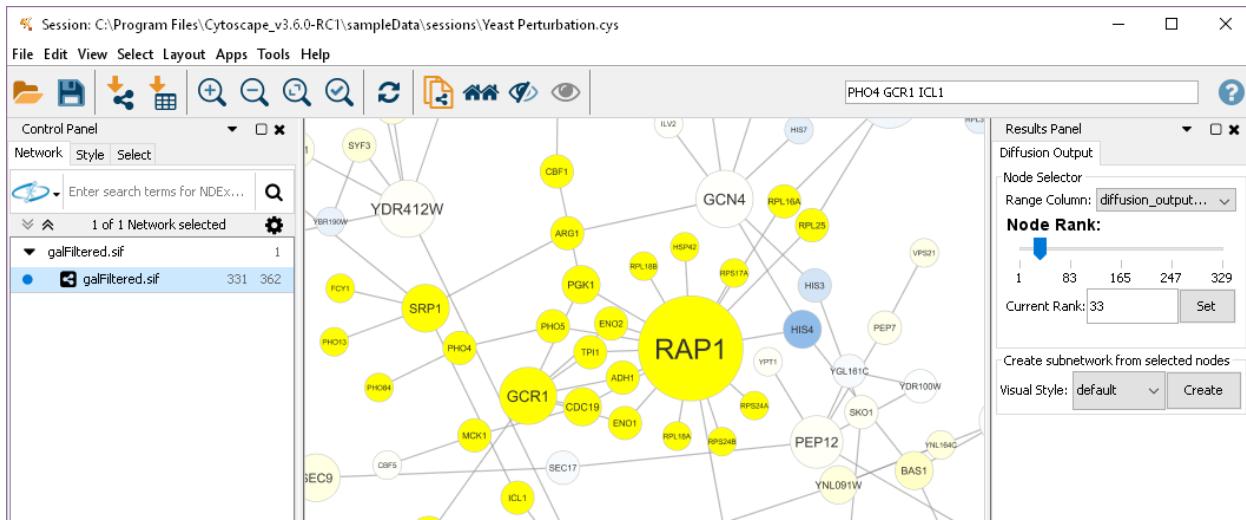
Cytoscape's Diffusion algorithm attempts to use a set of nodes and an entire interaction network to find the nodes most relevant to the original set. Conceptually, Diffusion applies heat to each node in the set, and lets the heat flow through connecting edges to adjacent nodes. It then produces a list of nodes ranked by the heat they accumulated. A node with many connections will tend to have a higher ranking, and an isolated node will tend to have low rank (and thus be excluded from the resulting node set).

By default, Diffusion uses the set of selected nodes as the heat sources, with each node having the same initial heat. At the end of a Diffusion, Cytoscape leaves the top 90th percentile of hot nodes selected. It allows you to use the Results panel to select a higher or lower percentile dynamically. It also stores the node's initial heat as a node attribute in the "diffusion_input" column, and returns the heat and ranking values in the "diffusion_output_heat" and "diffusion_output_rank" columns.

An advanced Diffusion option allows you to specify initial heat values for each node via its "diffusion_input" attribute.

This figure shows the result of selecting the PHO4, GCR1 and ICL1 genes (via the search bar) and performing a Diffusion by either selecting **Tools** → **Diffuse** → **Selected Nodes** or right-clicking to **Diffuse** → **Selected Nodes**.

Diffusion calculated the heat ranking of all 331 nodes in the network, and then selected the top 33.



To select more than 33 nodes, move the **Node Rank** slider in the Diffusion Output Results Panel to the right or enter a number greater than 33 in the **Current Rank** field. You can also select nodes using a heat value cutoff by using the **Range Column** to select a column containing heat values. Finally, you can use the **Visual Style** chooser and **Create** button to extract the selected nodes into a new network.

You can execute Diffusion multiple times on a network, thereby creating multiple heat, output_heat and output_rank columns.

10.4 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
...
```


Navigation and Layout

11.1 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.

11.1.1 Zoom

Cytoscape provides four mechanisms for zooming: toolbar buttons, menu options, keyboard shortcuts and the scroll wheel.

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**
- Menu option: **View → Zoom In**
- Keyboard shortcut: **Ctrl-Plus** (**Command-Plus** on Mac OS X)
- **Zoom Out**
- Menu option: **View → Zoom Out**
- Keyboard shortcut: **Ctrl-Minus** (**Command-Minus** on Mac OS X)
- **Zoom Out to Display all of Current Network**
- Menu option: **View → Fit Content**

- Keyboard shortcut: `Ctrl-0` (`Command-0` on Mac OS X)
- **Zoom Selected Region**
- Menu option: **View → Fit Selected**
- Keyboard shortcut: `Ctrl-9` (`Command-9` on Mac OS X)

Using the scroll wheel, you can zoom in by scrolling up and zoom out by scrolling downwards. These directions are reversed on Macs with natural scrolling enabled (the default for Mac OS X Lion and newer versions).

11.1.2 Pan

There are two ways to pan the network:

- Left-Click and Drag - You can pan the network view by holding down the left mouse button and moving the mouse.
- Dragging Box on **Network Overview** - You can also pan the view by left-clicking and dragging the blue box in the overview panel in the lower part of the view.

11.2 Other Mouse Behaviors

11.2.1 Select

- Click the left mouse button on a node or edge to select that element.
- Hold down the `Shift` or `Ctrl` key (`Command` on Macs) and left-click a node or edge to add it to the selection. Doing the same on a selected element unselects it.
- Hold down the left mouse button on the canvas background and drag the mouse while holding down the `Shift` or `Ctrl` key (`Command` on Macs) to select groups of nodes/edges.

11.2.2 Context

Click the right mouse button (or `Ctrl+left` mouse button on Macs) on a node/edge to launch a context-sensitive menu with additional information about the node/edge.

11.2.3 Node Context Menu

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

- Add
- Diffuse
- Edit
- Select
- Group
- Nested Networks
- Apps
- External Links

- Preferences

Edges usually have the following menu:

- Add
- Delete
- Diffuse
- Edit
- Select
- Group
- Apps
- Preferences

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

Nested Network Node Context Menu

- **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:** Removes the currently associated nested network from a node. The associated network is not deleted. Only the association between the node and the network is removed.
- **Go to Nested Network:** 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.

More information about nested networks can be found in the [Nested Networks](#) section.

11.3 Manual Layout

The simplest method to manually organize a network is to click on a node and drag it. All of the selected nodes are moved together.

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 , \uparrow , \rightarrow , \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 move 15 pixels in the chosen direction.

11.4 Node Layout Tools

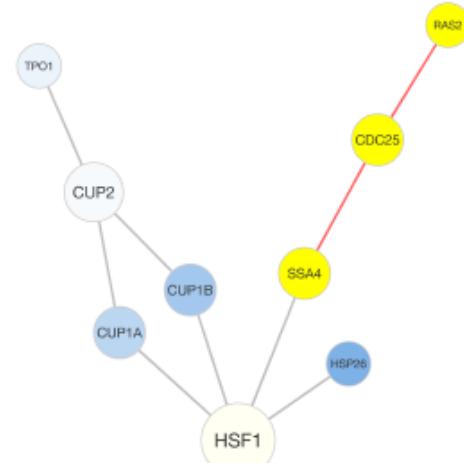
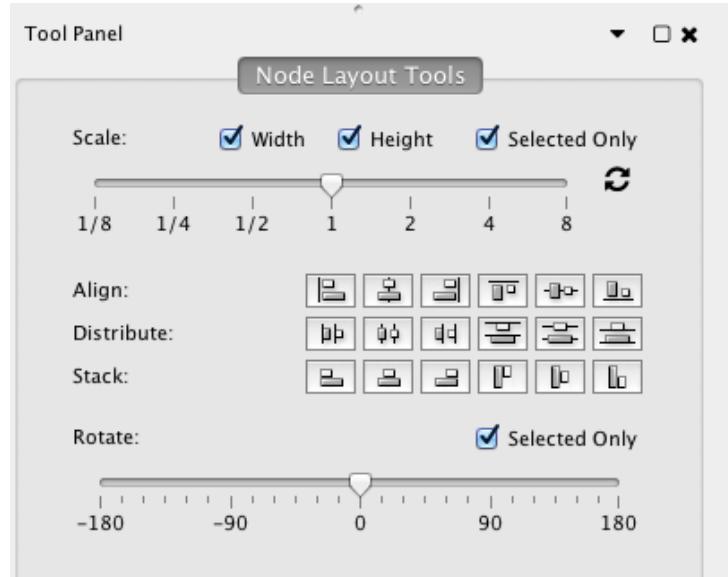
The Tool Panel is available via the menu command: **View → Show Tool Panel**.

It contains several **Node Layout Tools** that can help to automate or fine tune a layout.

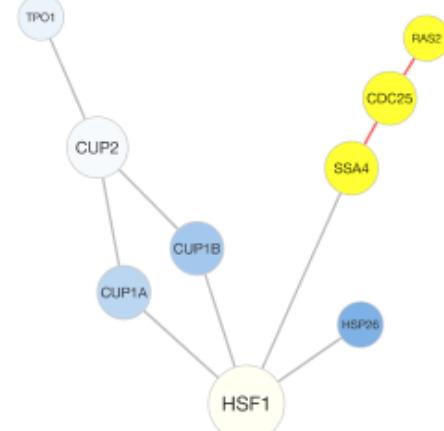
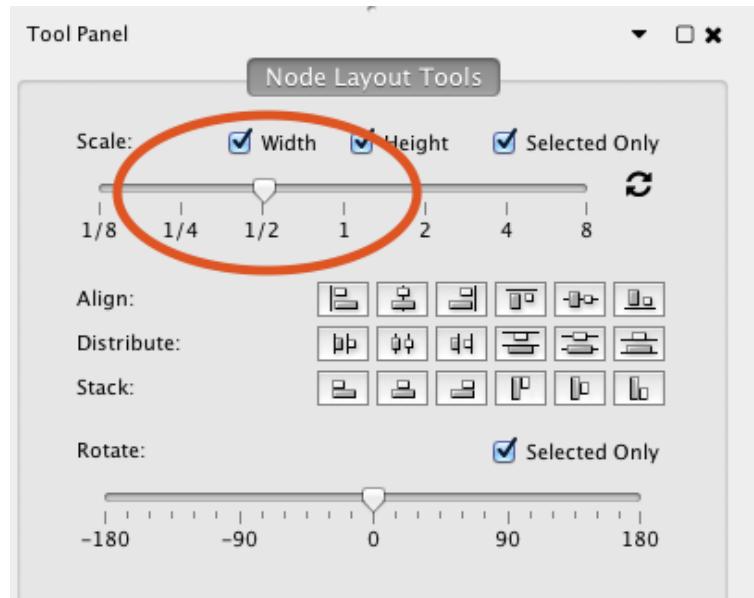
11.4.1 Scale

Adjust the Scale slider to change the length of edges. The position of the nodes will be scaled, not the node sizes. Node size can be adjusted using [Styles](#). The images below show selected (yellow) nodes scaled to 50% of the default value.

Before



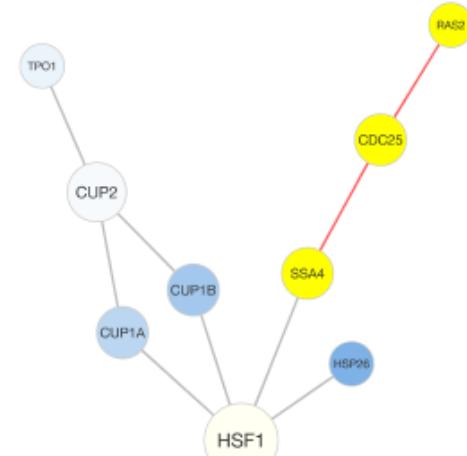
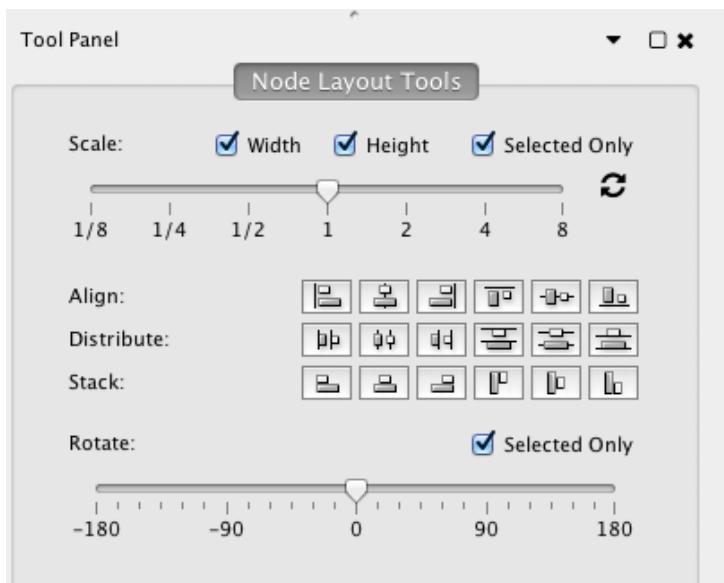
After



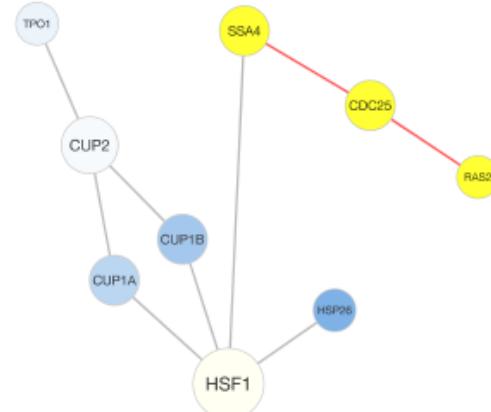
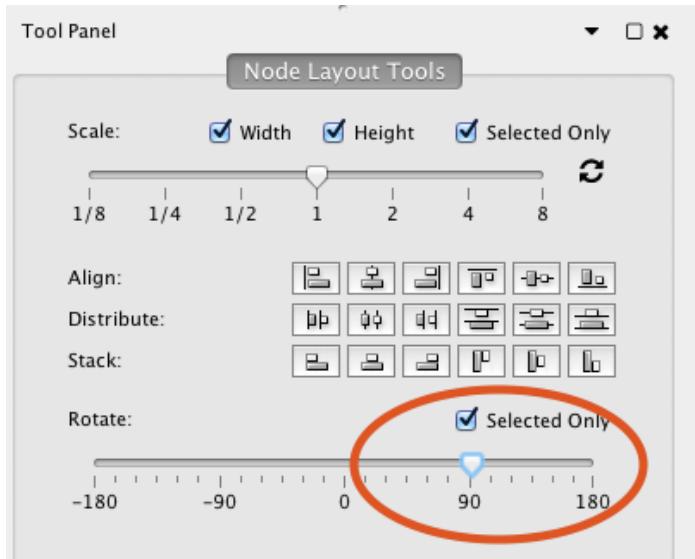
11.4.2 Rotate

The Rotate function will change the orientation of the entire network or a selected portion of the network. The images below show a network with selected nodes rotated 90 degrees.

Before



After



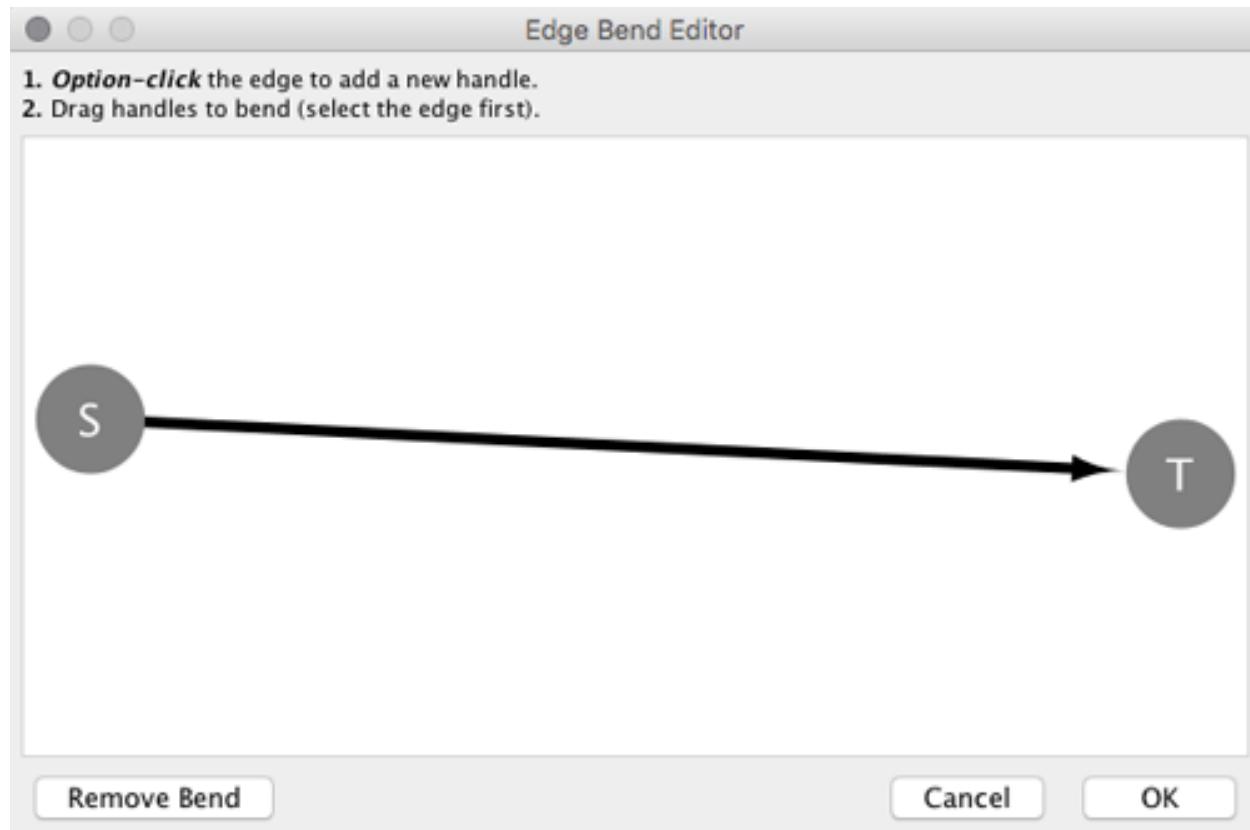
11.4.3 Align, Distribute and Stack

The **Tool Panel** contains buttons to set relative positions of nodes.

Align provides 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. **Distribute** evenly distributes 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. **Stack** vertically or horizontally stacks selected nodes with the full complement of alignment options. The table below provides a description of what each button does.

11.5 Edge Bend and Automatic Edge Bundling

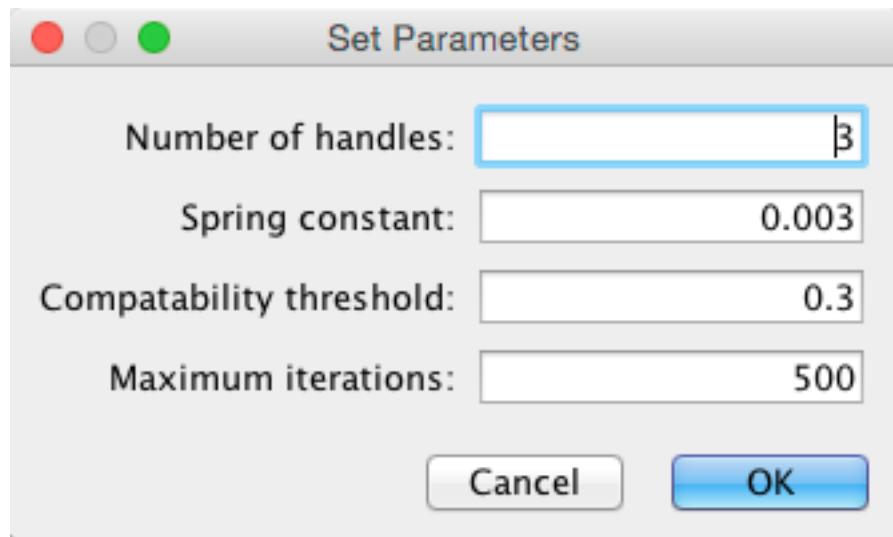
From Cytoscape 3.0, **Edge Bend** is a regular edge property and can be used as a part of a **Style**. Just like any other edge property, you can select a Default Value, a Mapping and use Bypass for select nodes. In the Styles tab, select the **Bend** property from the **Properties** drop-down and click on either the Default Value, Mapping or Bypass cell to bring up the **Edge Bend Editor**. In the editor, you can add as many handles as you want to the edge using Alt-Click on Windows, Option-Click on Mac, or Ctrl-Alt-Click on Linux.





To clear all edge bends, select **Layout → Clear All Edge Bends**.

In addition to adding handles manually, you can use the **Bundle Edges** 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](http://www.win.tue.nl/~dholten/papers/forcebundles_eurovis.pdf) (http://www.win.tue.nl/~dholten/papers/forcebundles_eurovis.pdf).
3. Press OK to run. Edge bundling may take a long time if the number of edges is large.
 - If it takes too long, try decreasing **Maximum Iterations**.
 - For large, dense networks, try setting **Maximum iterations** in the range of 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.

11.6 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.

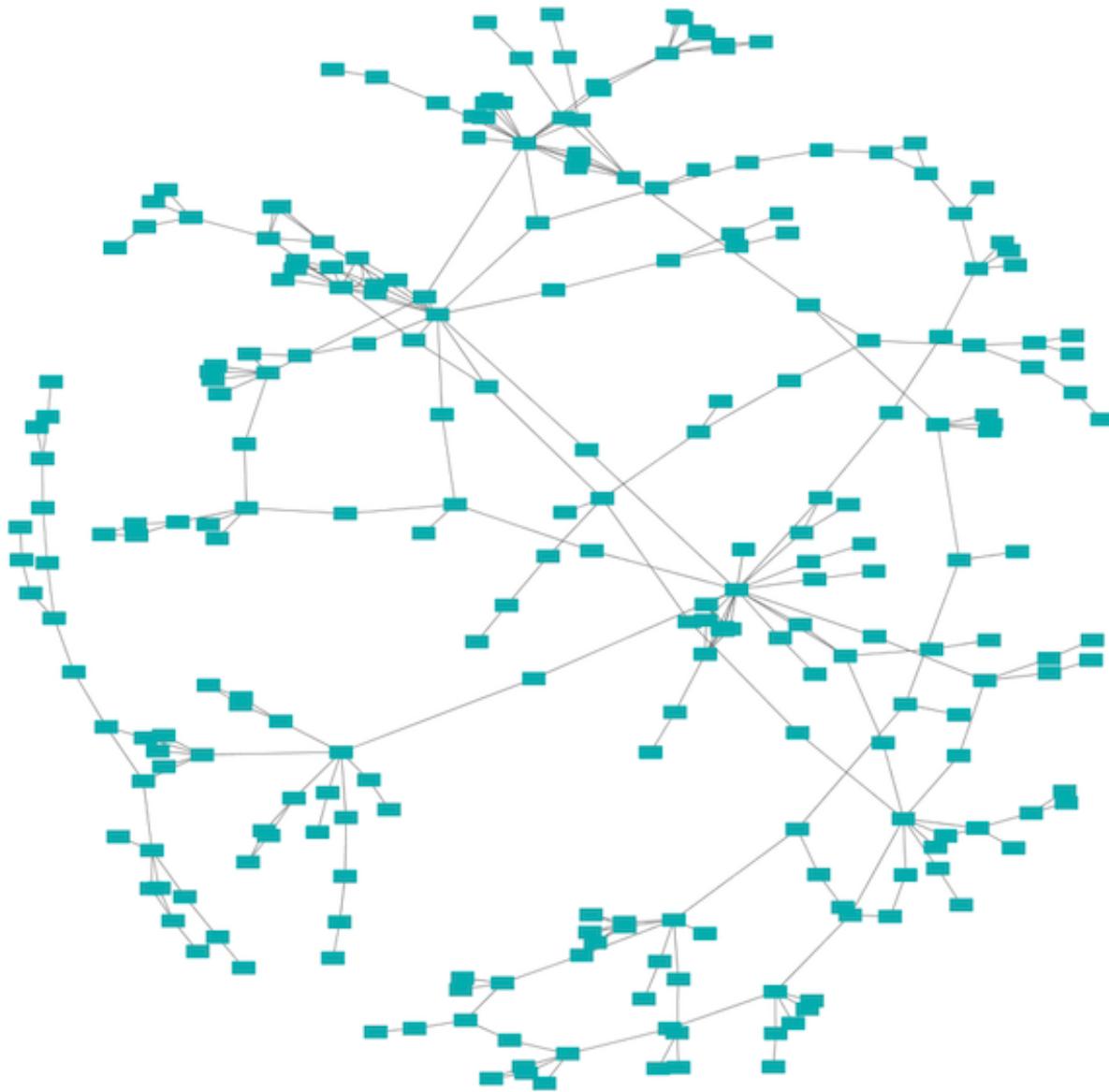
Cytoscape Layouts have the option to operate on only the selected nodes, and all provide a **Settings...** panel to change the parameters of the algorithm. Most of the Cytoscape layouts also partition the graph before performing the layout. In addition, many of these layouts include the option to take either node or edge columns into account. A few of the layout algorithms are:

11.6.1 Grid Layout

YDR146C	YER111C	YDR354W	YER103W	YOLO59W	YBR190W	YOR362C	YER102W	YOL149W	YMR044W	YBR050C	YIL162W	YFL038C	YOR355W	YML123C
YHR071W	YPL031C	YML051W	YDR003W	YIL113W	YLL019C	YML024W	YGL163C	YDR100W	YNL157C	YGR009C	YMR183C	YFR034C	YAL040C	YMR311C
YOR315W	YOR117C	YER113W	YBR072W	YCL073W	YLR321C	YPL111W	YBL069W	YNL189W	YDL236W	YPR062W	YLR319C	?	YJR066W	YML064C
YNL047C	YNL119C	YPR048W	YPL075W	YHR030C	YPL089C	YJL089W	YER065C	YLR377C	YCR019W	YLR310C	YNL098C	YGL166W	YKL074C	YDR335W
YDR174W	YLL028W	YKL101W	YOL156W	YJL219W	YLL021W	YOL136C	YJL203W	YNR002C	YFL026W	YPR015W	YER040W	YGL008C	YDR323C	YOR36W
YBL005W	YBR160W	YKL109W	YHR171W	YBR217W	YKL028W	YDR311W	YPL149W	YCL237C	YBL021C	YNL145W	YJL157C	YCR108W	YDR461W	YJL159W
YKR097W	YMR043W	YIL015W	YDR309C	YBR045C	YLR229C	YGR058W	YLR345W	YLR256W	YML054C	YPR145W	YJR048W	YER054C	YEL039C	YDL075W
YIL143C	YDR184C	YNL091W	YNL236W	YCR136W	YKL012W	YML032C	YNL312W	YIL052C	YLR116W	YER056CA	YNL069C	YOR264W	YOL051W	YBR019C
Y8R020W	YGL035C	YPL248C	YLR082W	YKR099N	YR009W	YBR018C	YMR138W	YMR300C	YLR109W	YDR277C	YHR141C	YDL194W	YLR362W	YCL032W
YDR032C	YJL060W	YDR103W	YBR155W	YMR180W	YEL009C	YIL070C	YGR046W	YLR153C	YBR248C	YOL058W	YNL164C	YDR070C	YMR108W	YDR202W
YER179W	YLR117C	YDR395W	YPR119W	YGR085C	YGR088W	YOR361C	YMR309C	YPR113W	YLR134W	YIL105C	YOR327C	YER143W	YAL030W	YOL086C
YCL013C	YCR086W	YDR412W	YDL023C	YBR093C	YDR171W	YOR212W	YCL030C	YNL036W	YNL301C	YML114C	YMR117C	YDR050C	YPR010C	YOL127W
YIL069C	YER074W	YHR005C	YHR174W	YLL133C	YR452C	YBR274W	YLR044C	YMR255W	YHR084W	YOL120C	YNL307C	YAL038W	YBL050W	YNL216W
YCR012W	YBL026W	YGR254W	YJL194W	YLR264W	YNR053C	YLR258W	YER116C	YGL134W	YHR055C	YDL013W	YJR022W	YHR053C	YNR050C	YNL135C
YNL050C	YER052C	YEL015W	YBR112C	YMR009W	YER124C	YCR084C	YER112W	YOR167C	YCL067C	YDL030W	YDL215C	YGR203W	YL061C	YIL045W
YCL044C	YOL123W	YOR120W	YPR041W	YJL017C	YDR429C	YMR146C	YDR299W	YLR175W	YNL117W	YKR026C	YDR167W	YCL122C	YLR432W	YBR135W
YOR089C	YPR167C	YPL240C	YHR198C	YLR284C	YKL161C	YLR075W	YPR102C	YAR007C						

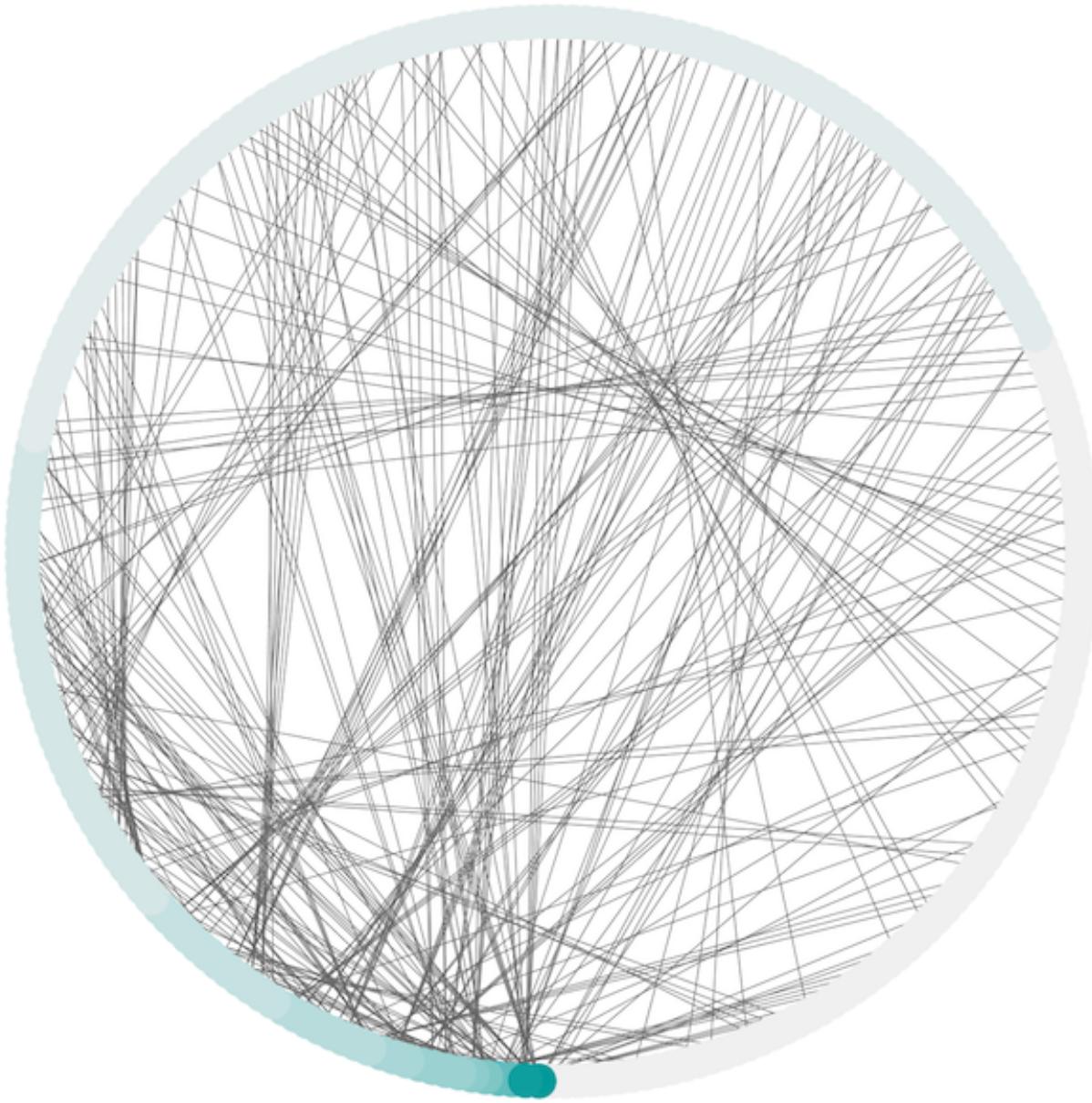
The grid layout is a simple layout that arranges all of the nodes in a square grid. This is the default layout and is always available as part of the Cytoscape core. It is available by selecting **Layout → Grid Layout**. A sample screen shot is shown above.

11.6.2 Edge-weighted Spring-Embedded Layout



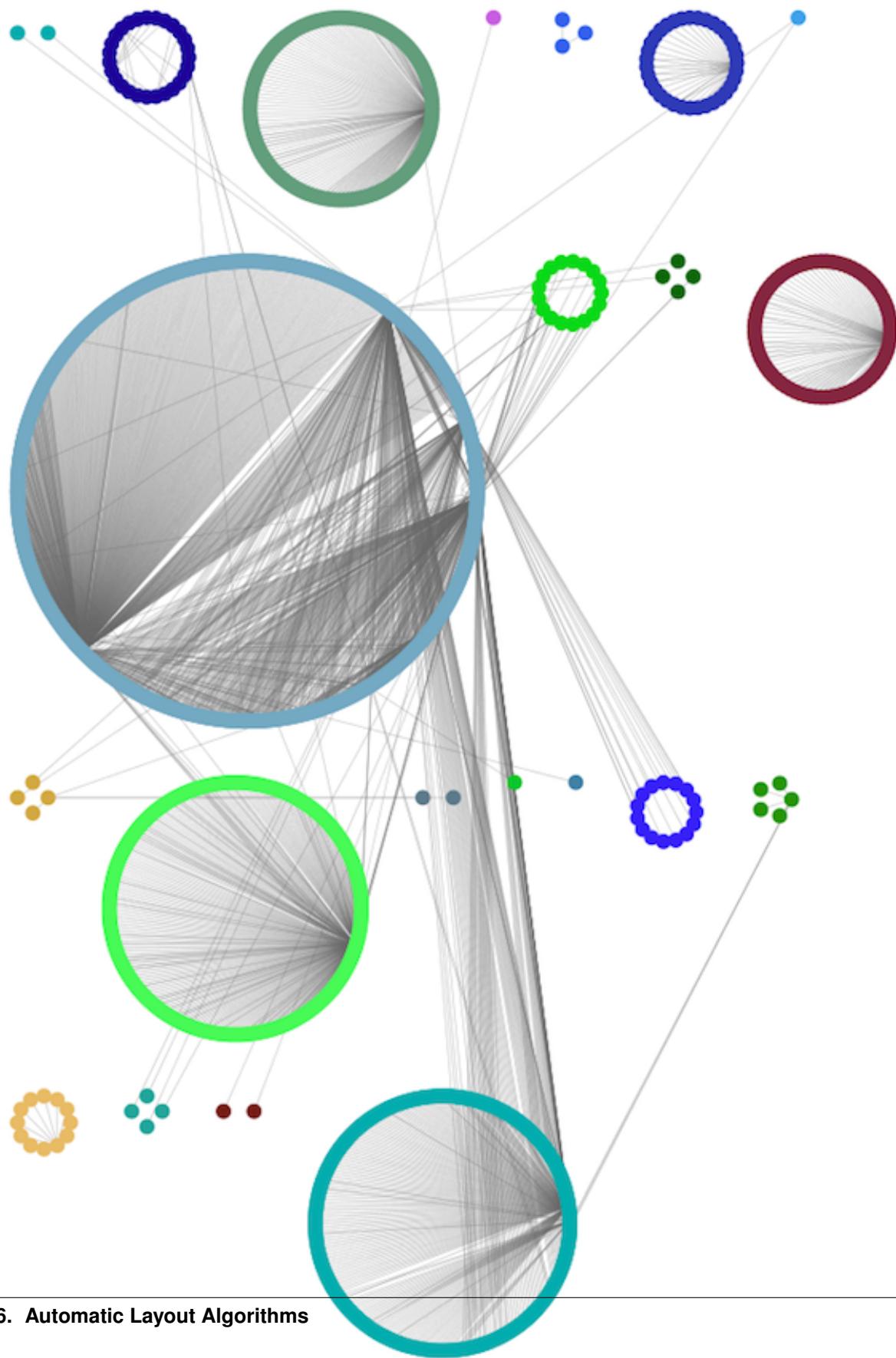
The spring-embedded layout is based on a “force-directed” paradigm as implemented by Kamada and Kawai (1988). Network nodes are treated like physical objects that repel each other, such as electrons. The connections between nodes are treated like metal springs attached to the pair of nodes. These springs 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**.

11.6.3 Attribute Circle Layout



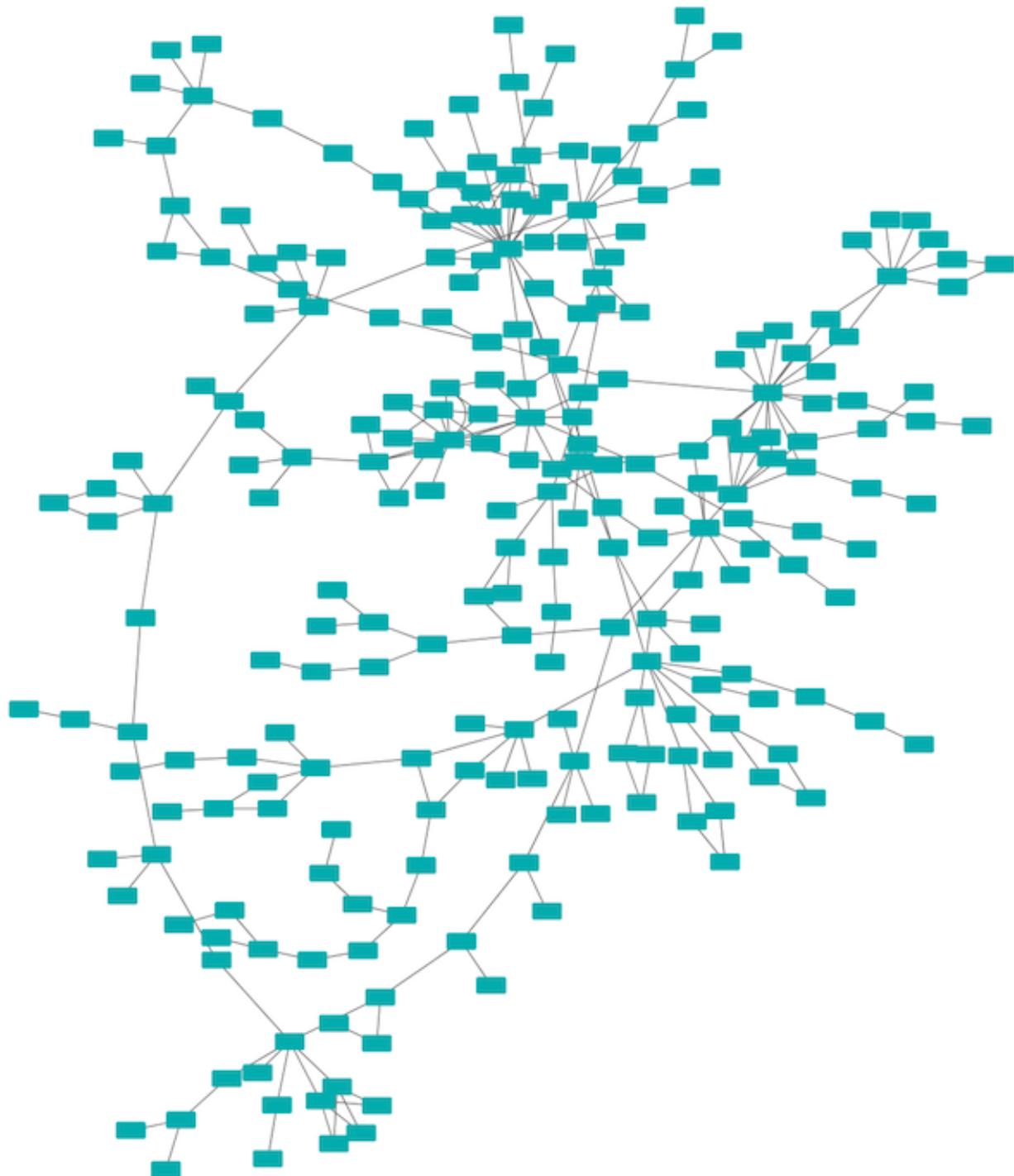
The **Attribute Circle** layout is a quick, useful layout, particularly for small networks, that will locate all of the nodes in the network around a circle. The node order is determined by a user-selected node column. The result is that all nodes with the same value for that column are located together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

11.6.4 Group Attributes Layout



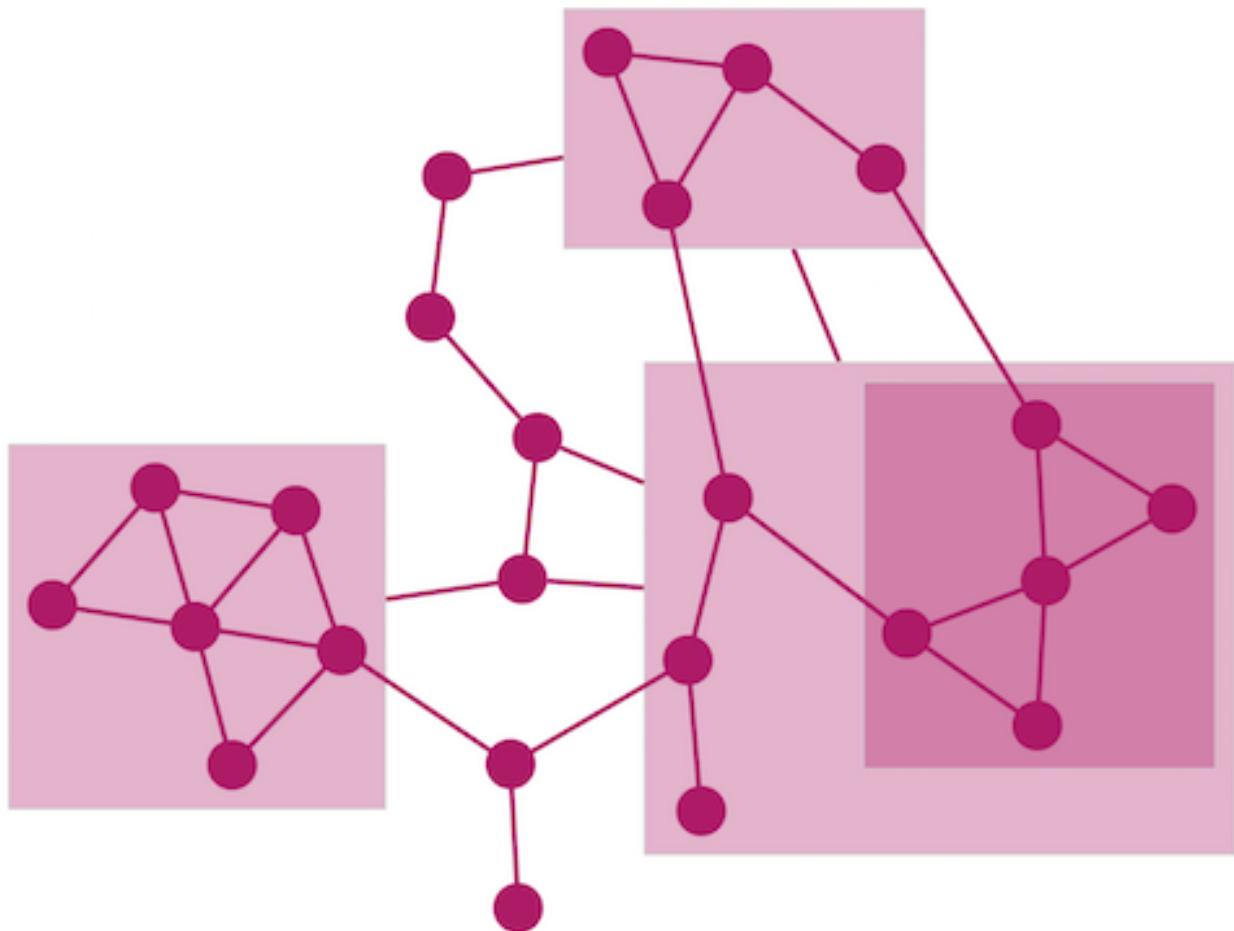
The **Group Attributes** layout is similar to the **Attribute Circle** layout described above except that instead of a single circle with all of the nodes, each set of nodes that share the same value for the column are laid out in a separate circle. The same network shown above (network generated by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

11.6.5 Prefuse Force Directed Layout



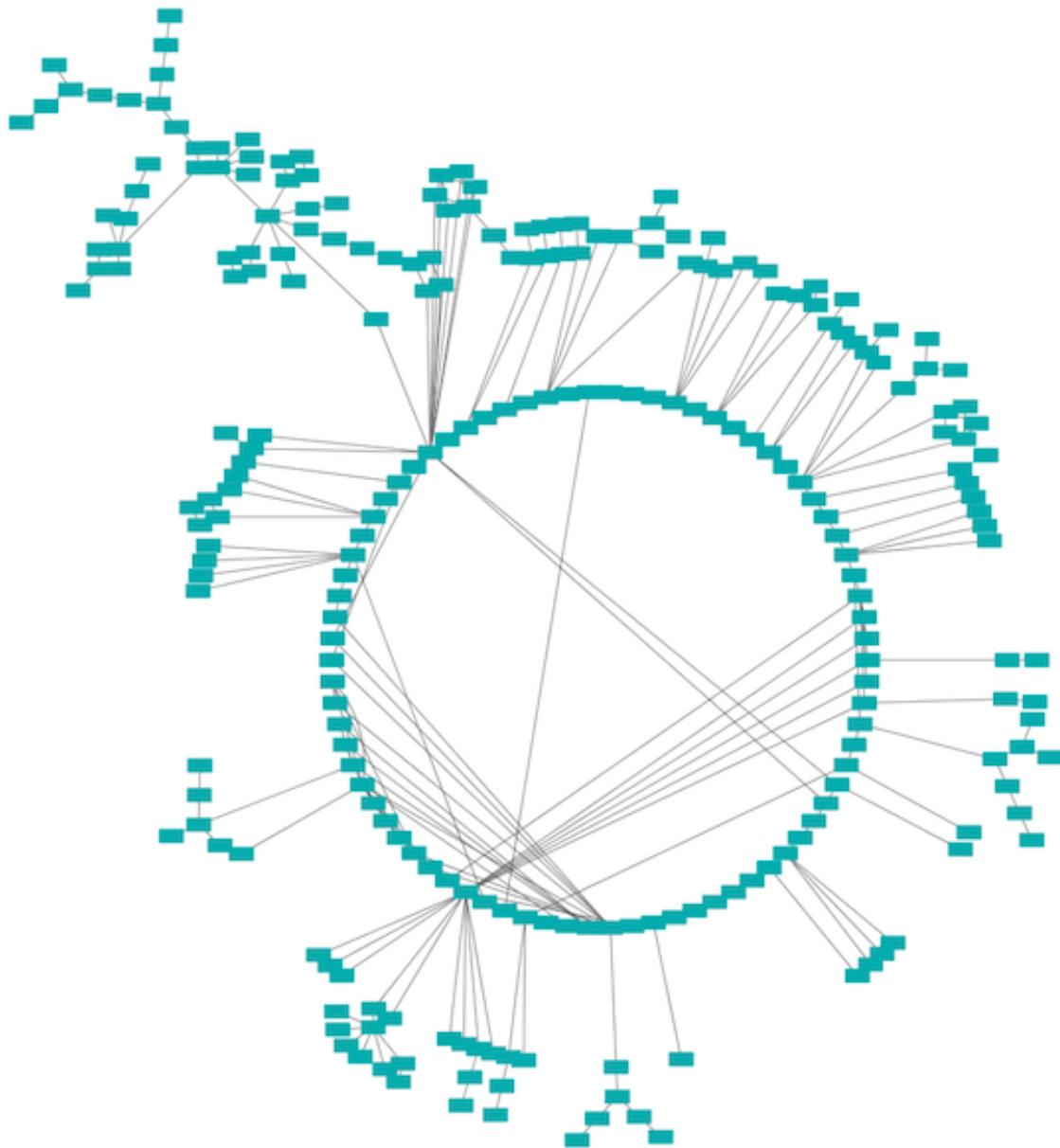
The force-directed layout is a layout based on the “force-directed” paradigm. This layout is based on the algorithm implemented as part of the [prefuse toolkit](http://www.prefuse.org/) (<http://www.prefuse.org/>) provided by Jeff Heer. The algorithm is very fast and with the right parameters can provide a very visually pleasing layout. The **Force Directed Layout** will also accept a numeric edge column 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 → Prefuse Force-Directed Layout → (unweighted)** or the edge column 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.

11.6.6 Compound Spring Embedder Layout



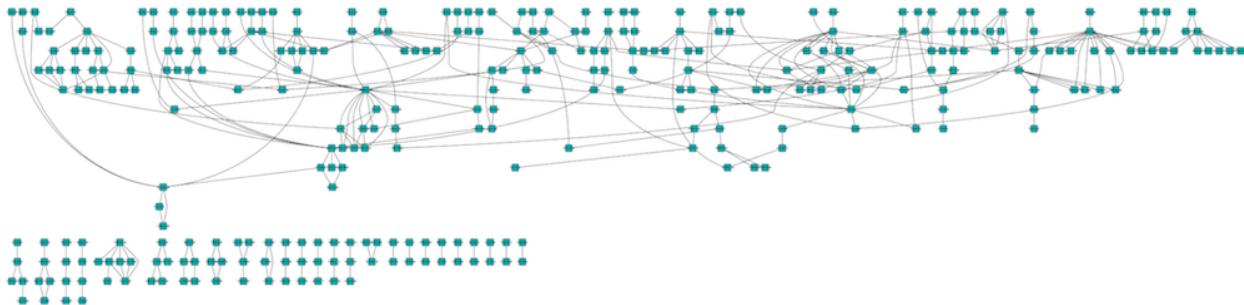
The **Compound Spring Embedder (CoSE)** layout is based on the traditional force-directed layout algorithm with extensions to handle multi-level nesting (compound nodes), edges between nodes of arbitrary nesting levels and varying node sizes. It is the suggested Cytoscape layout for compound graphs, although it also works very well with noncompound graphs. It is available by selecting **Layout → Compound Spring Embedder (CoSE)**.

11.6.7 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 → Circular Layout**.

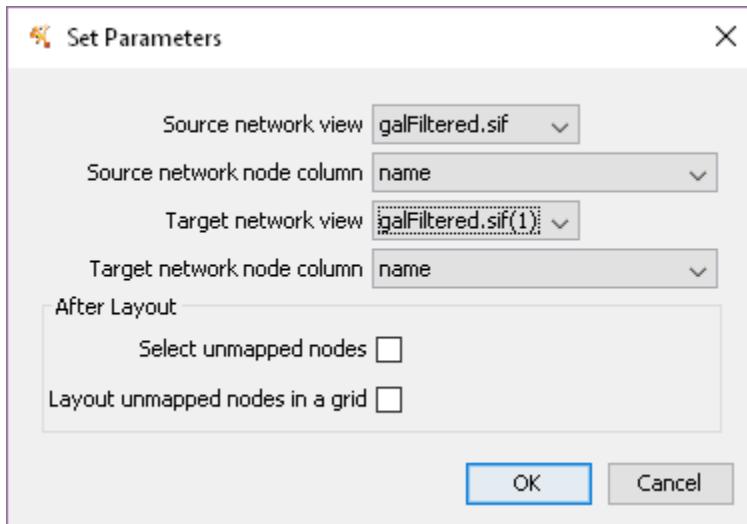
11.6.8 Hierarchical Layout



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 → Hierarchical Layout**.

11.6.9 Copycat Layout

The Copycat layout uses node positions in one network to lay out nodes in another network. Selecting **Layout → Copycat Layout** displays a dialog box that allows you to select the source network (which is already laid out) and the target network (which needs layout). By default, Copycat matches nodes in the source and target networks by node name, but you can choose any node attribute for this match.



As options, you can use **Select unmapped nodes** to cause Copycat to select target nodes that were not matched by nodes in the source network. This allows you to move them and possibly lay them out using other layout algorithms. You can use the **Layout unmapped nodes in a grid** option to preemptively move the unmatched target nodes away from the laid out target network.

11.7 yFiles Layouts

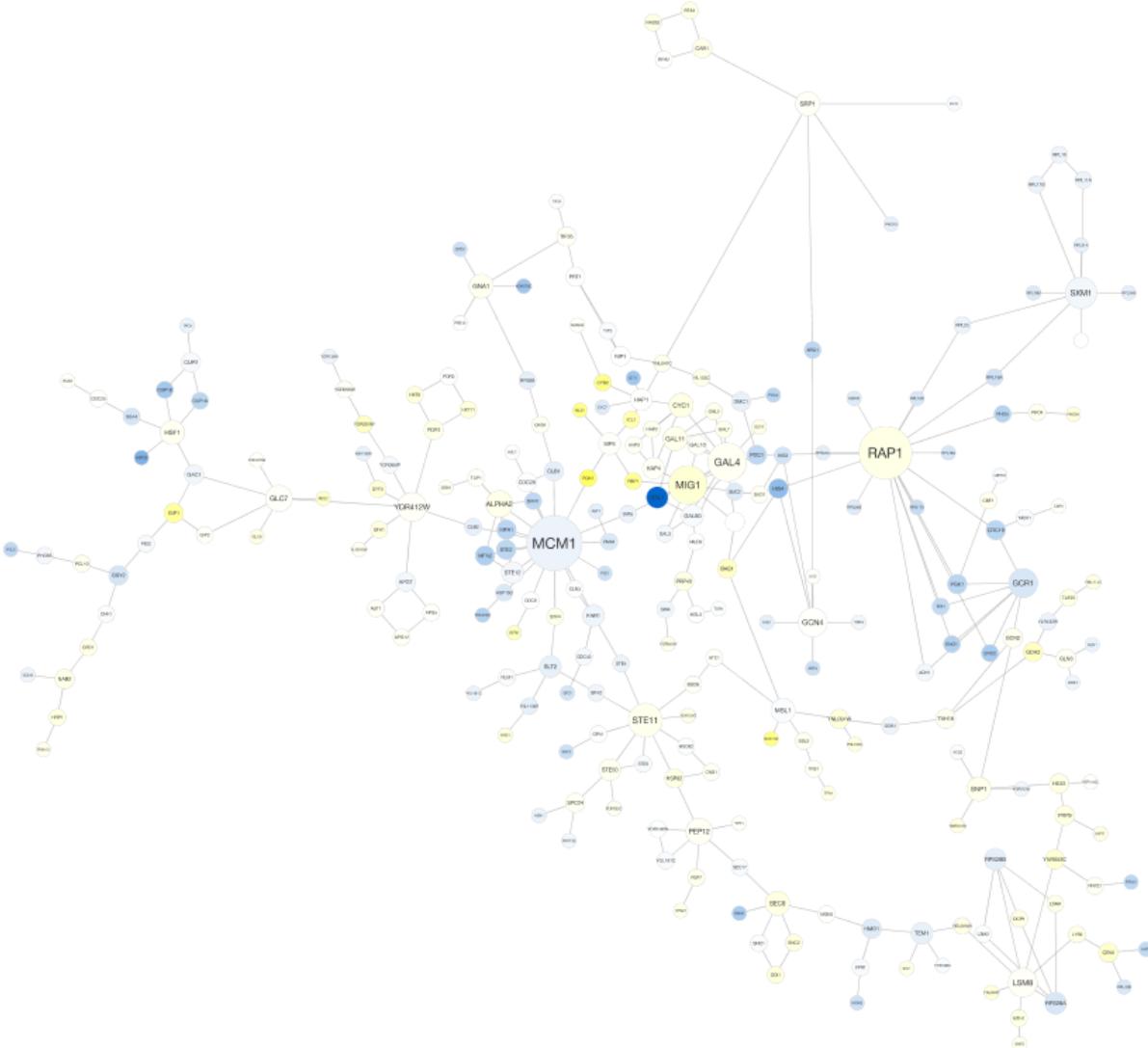
Cytoscape offers a set of layout algorithms based on the yFiles library. As of Cytoscape 3.6, the yFiles algorithms are available directly through the [Cytoscape App Store](#). To install them, go to **Layout → Install yFiles**, which will direct you to the App Store. Click the **Install** button to proceed. You will be directed to a license agreement, and once yFiles is installed, the yFiles layouts will be available in the **Layout** menu.

The layout algorithms included in yFiles are:

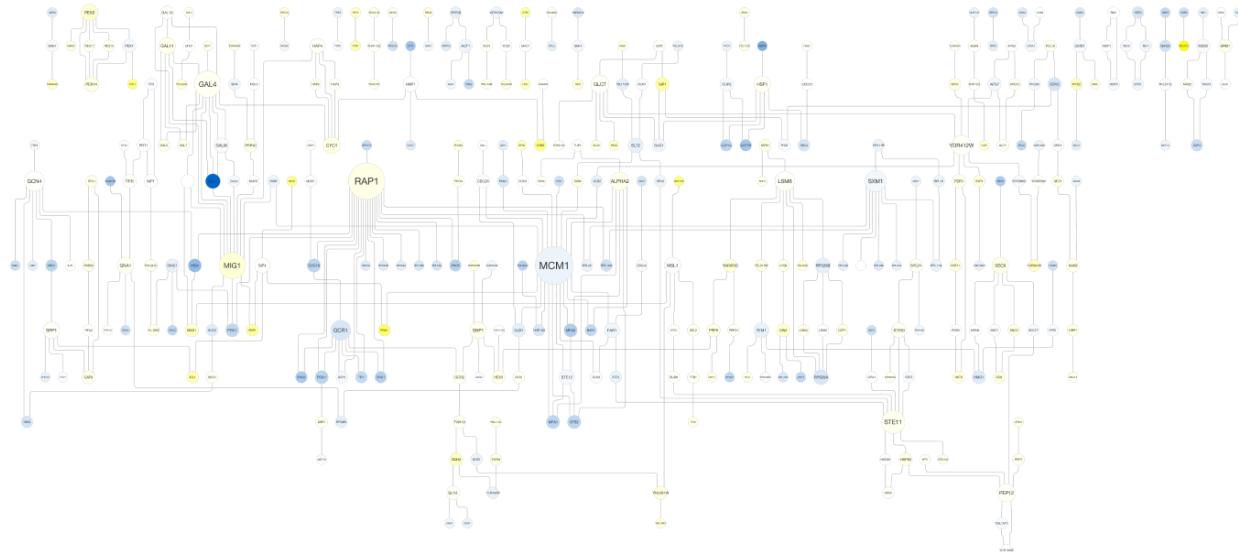
- Circular Layout
- Hierarchic Layout
- Hierarchic Layout Selected Nodes
- Organic Layout
- Orthogonal Layout
- Radial Layout
- Tree Layout
- Orthogonal Edge Router
- Organic Edge Router

Below are a few examples of yFiles layouts in Cytoscape, with more available on the [yFiles web page](#).

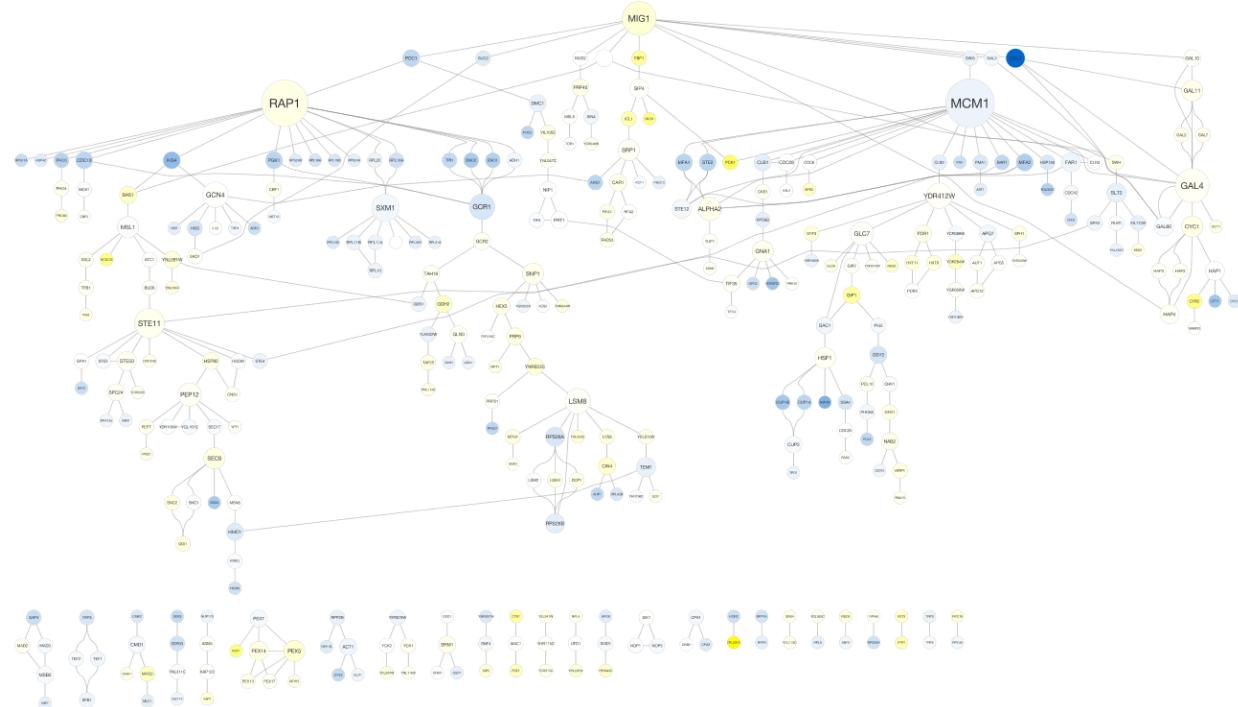
11.7.1 yFiles Organic Layout



11.7.2 yFiles Hierarchic Layout



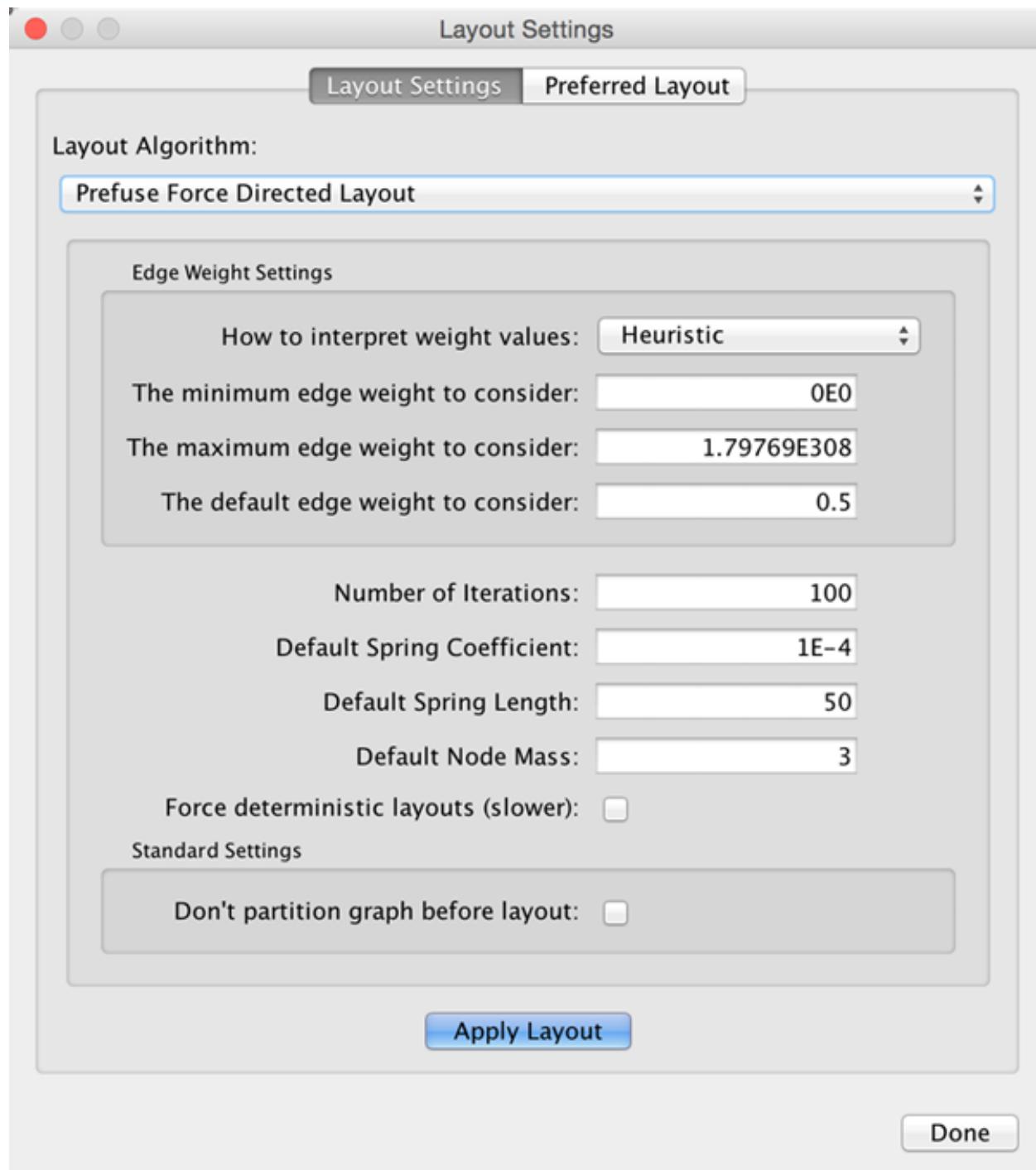
11.7.3 yFiles Tree Layout



11.8 Layout Parameters

Many layouts have adjustable parameters that are exposed through the **Layouts → Settings...** menu option. The **Layout Settings** dialog, which allows you to choose which layout algorithm settings to adjust, is shown below. 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.



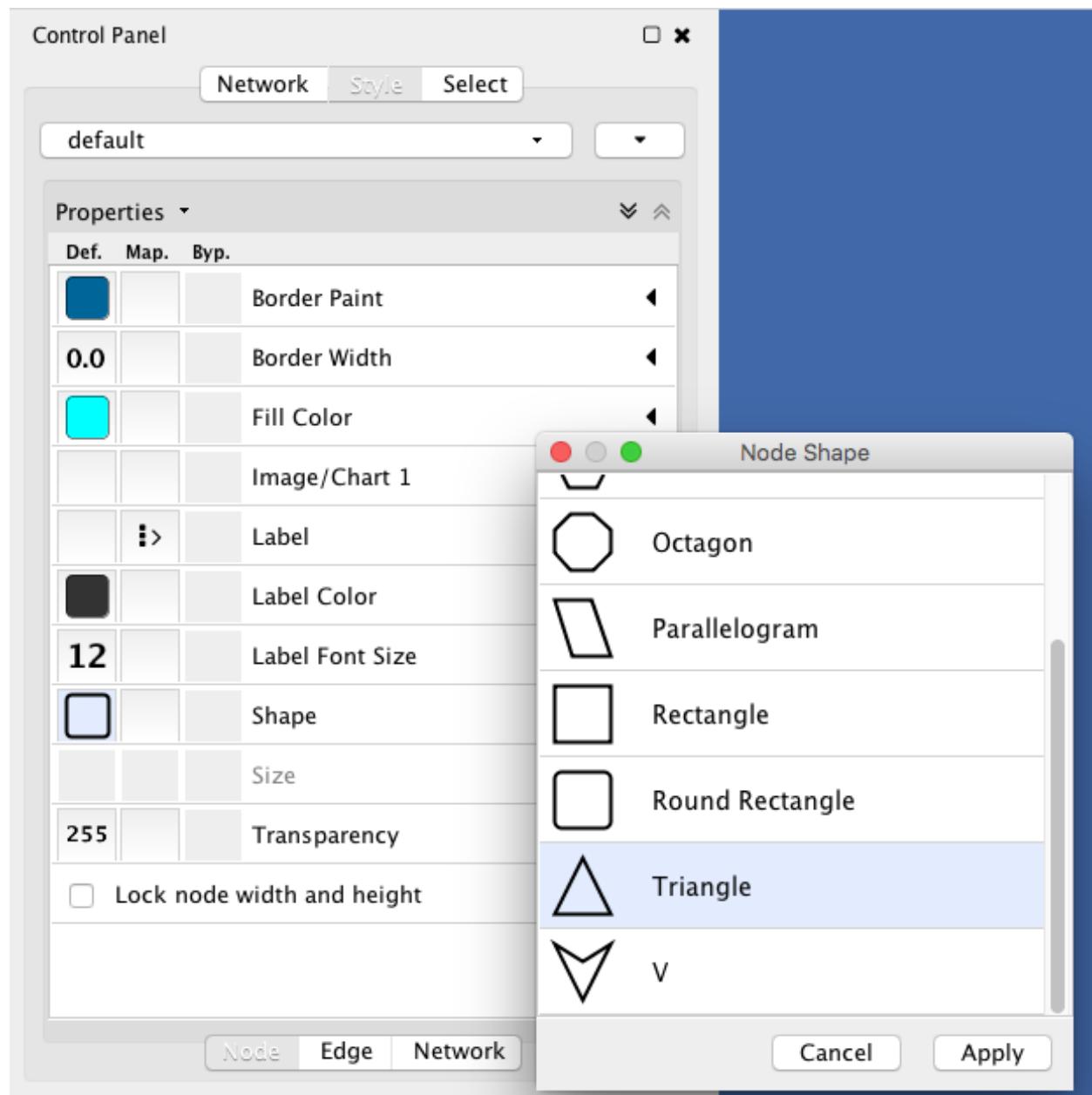
CHAPTER 12

Styles

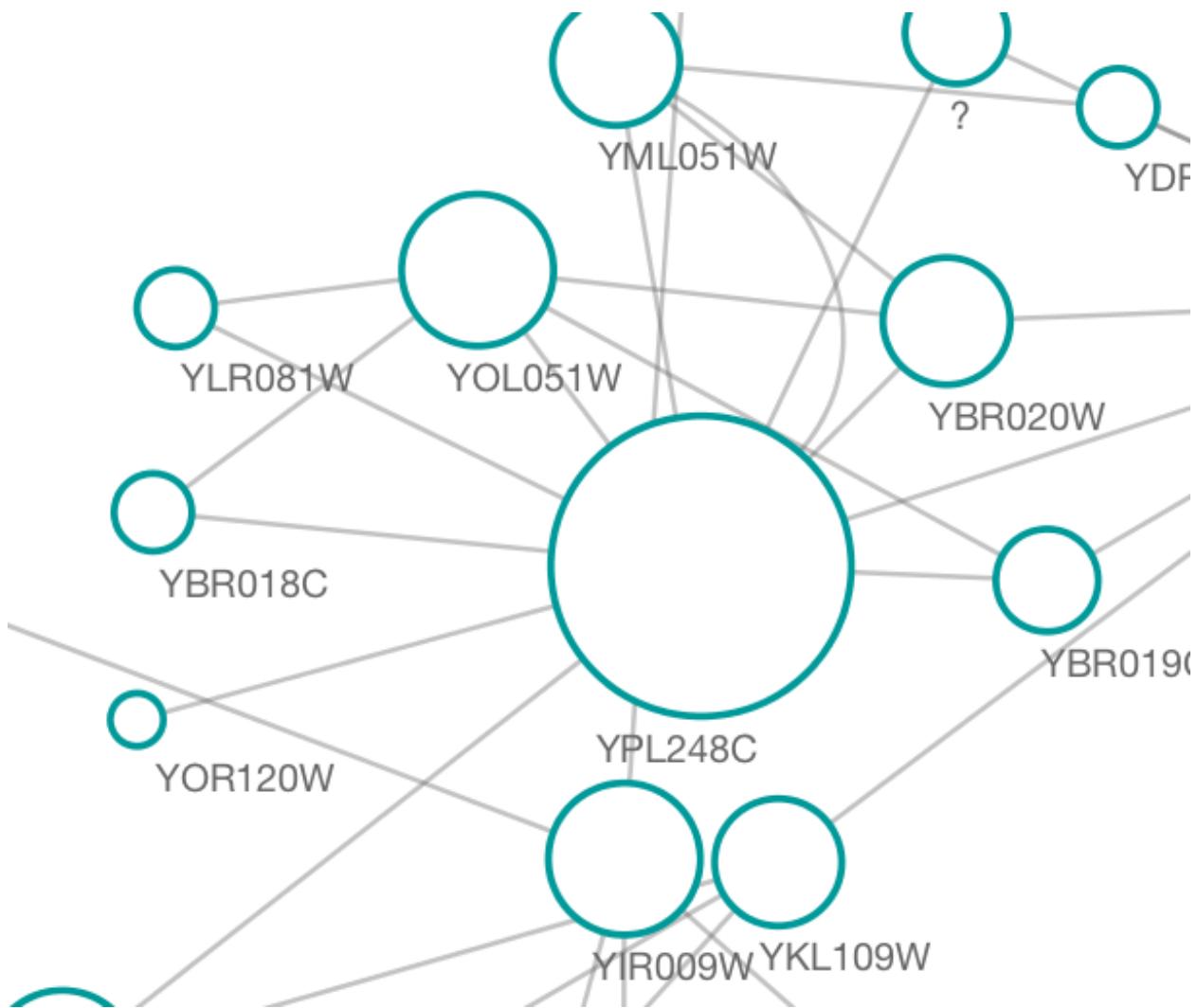
12.1 What are Styles?

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 property (such as color, size of node, transparency, or font type) of the network. A set of these encoded or mapped table data sets is called a **Style** and can be created or edited in the **Style** panel of the **Control Panel**. In this interface, the appearance of your network is easily customized. For example, you can:

Specify a default color and shape for all nodes.



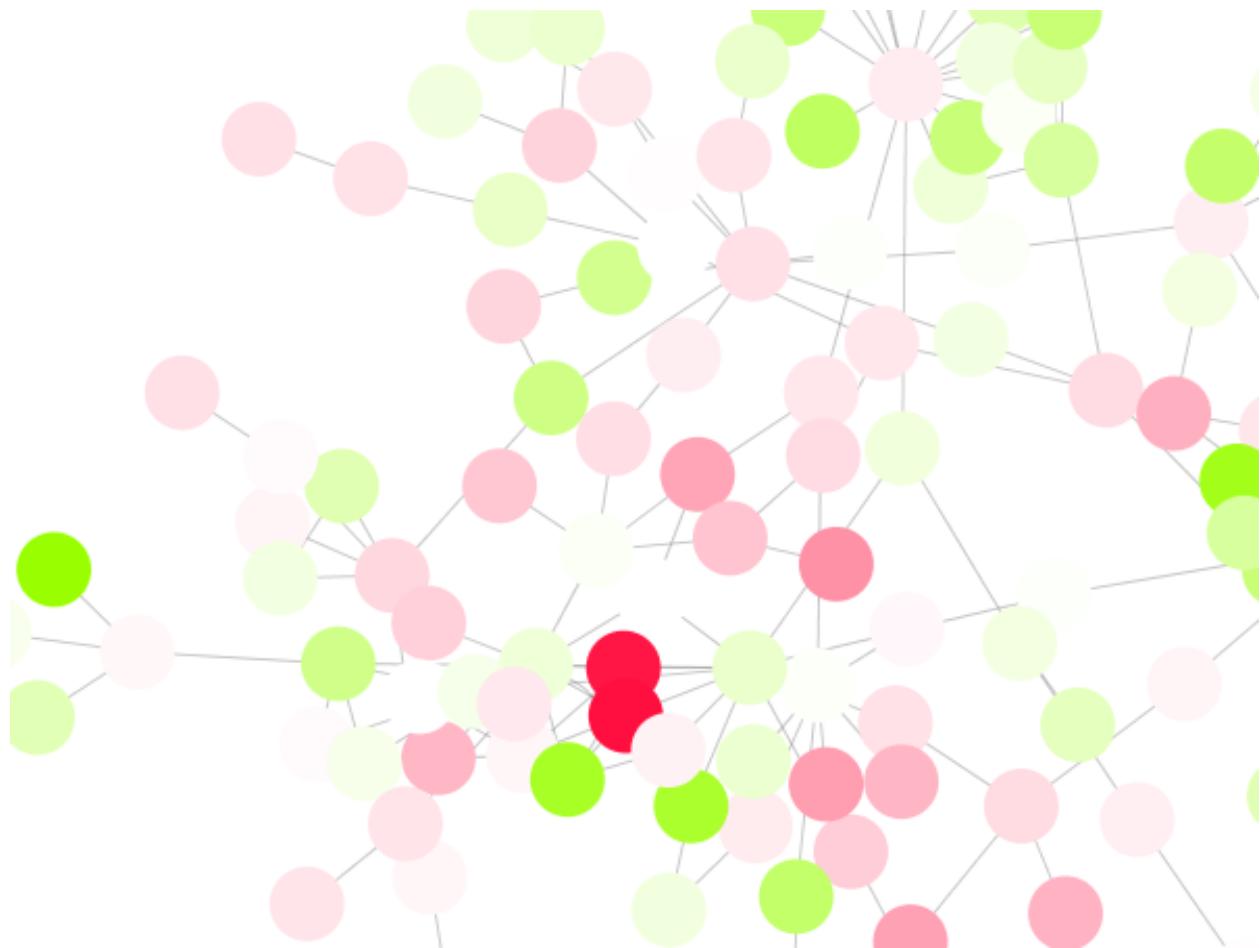
Set node sizes based on the degree of connectivity of the nodes. You can visually see the hub of a network...



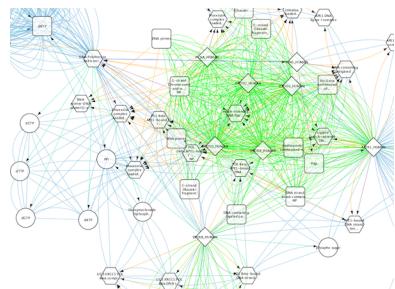
...or, set the font size of the node labels instead.



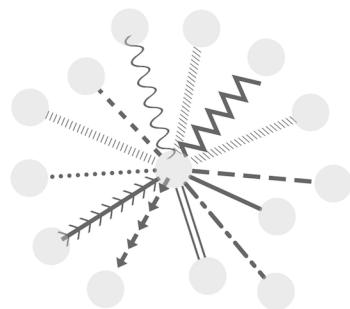
Visualize gene expression data along a color gradient.



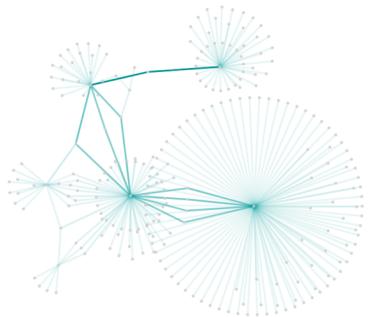
Encode specific physical entities as different node shapes.



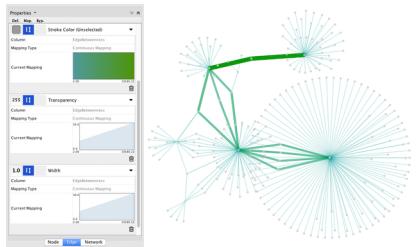
Use specific line types to indicate different types of interactions.



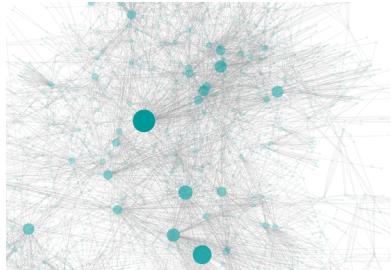
Control edge transparency (opacity) using edge weights.



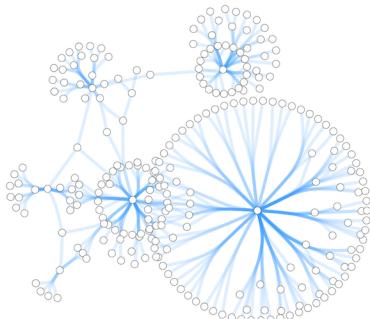
Control multiple edge properties using edge score.



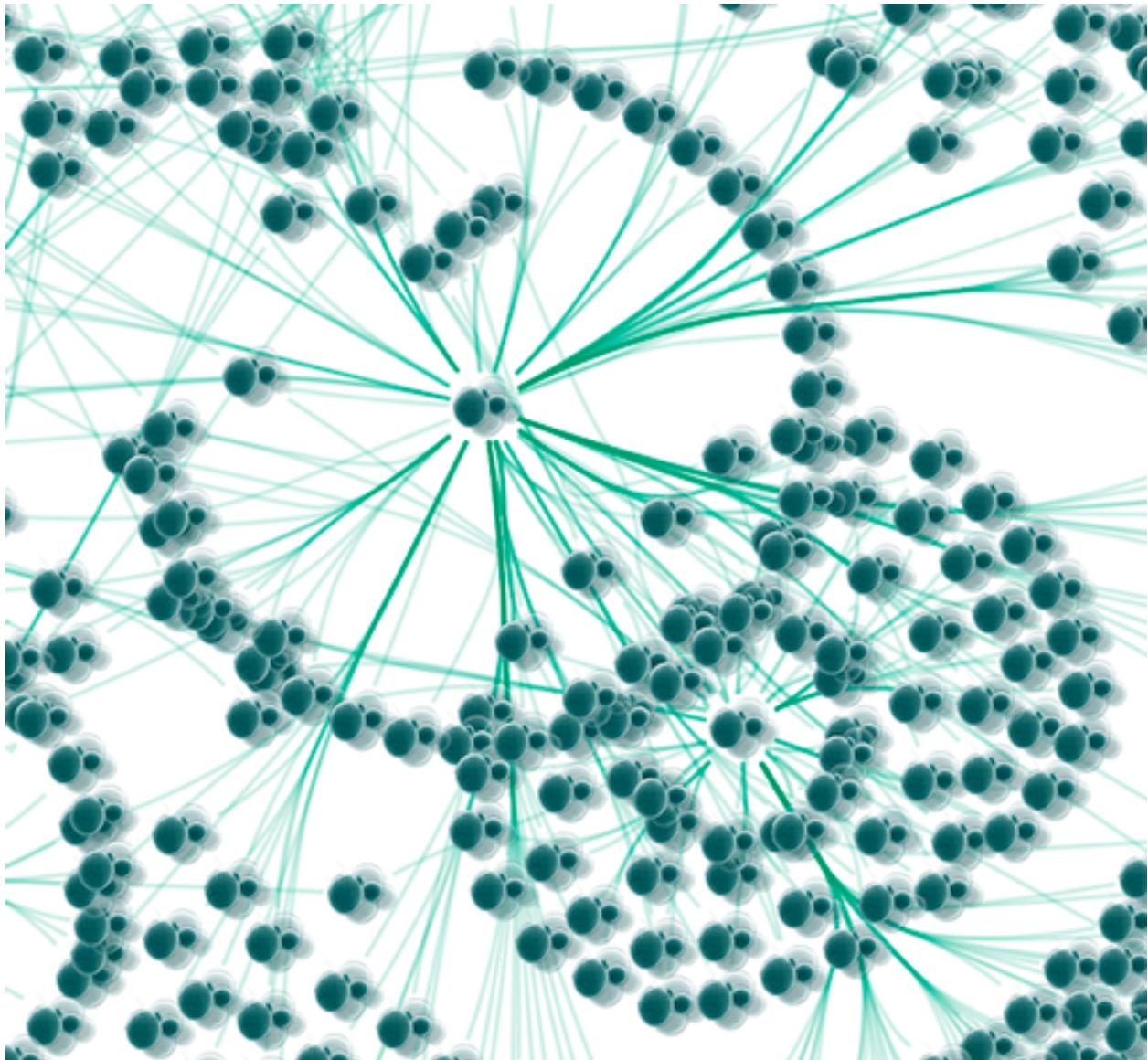
Browse extremely-dense networks by controlling the opacity of nodes.

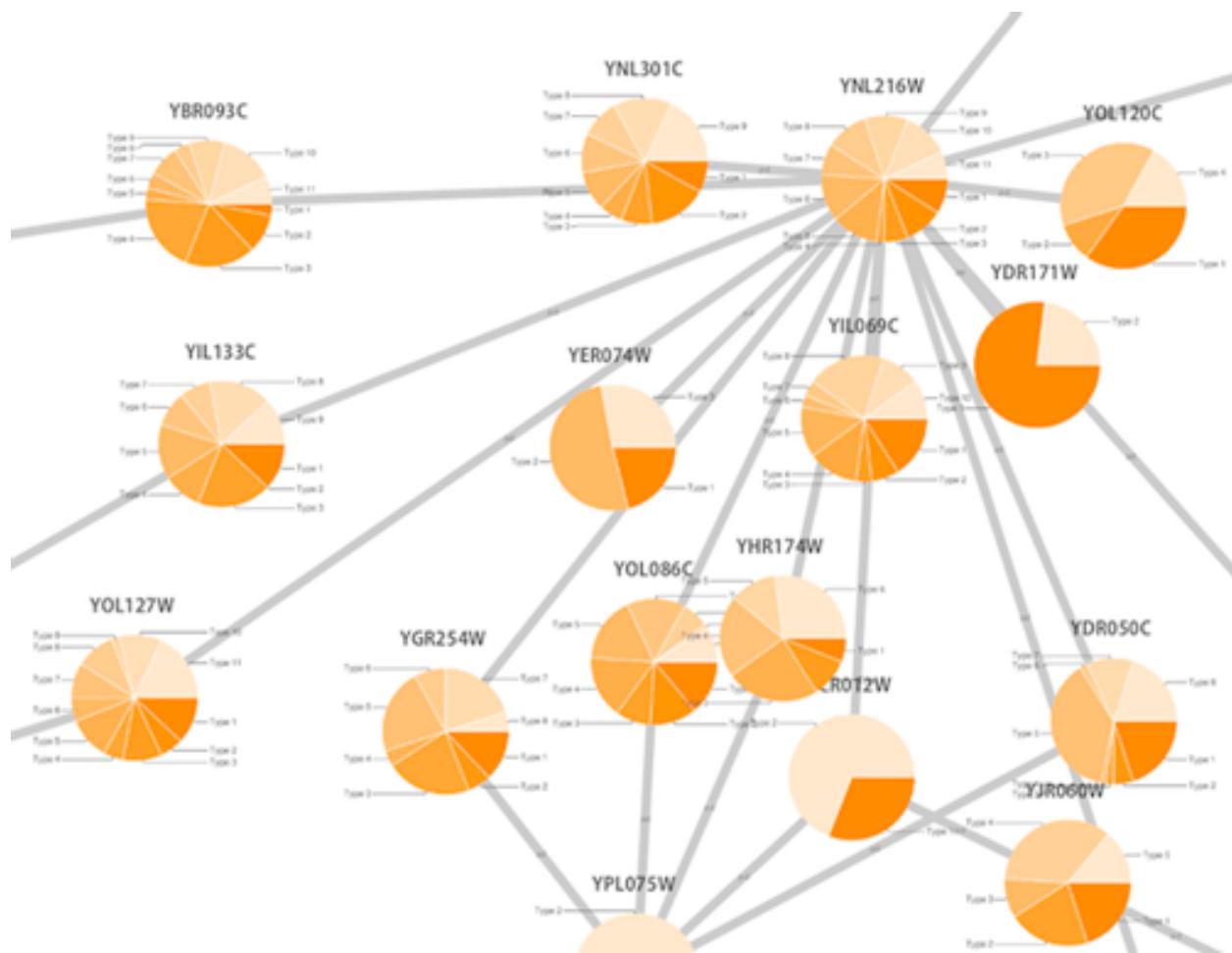


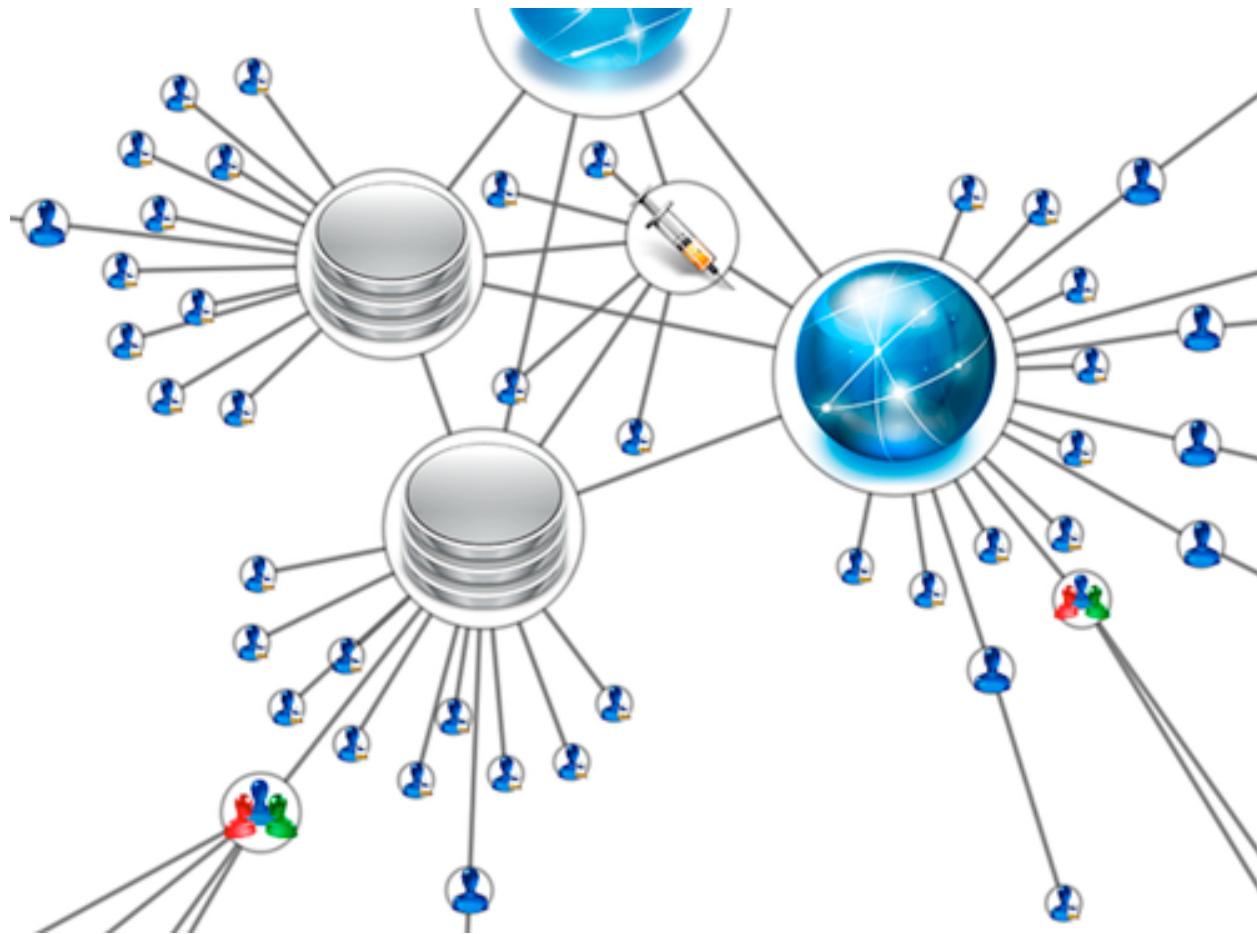
Show highly-connected region by edge bundling and opacity.



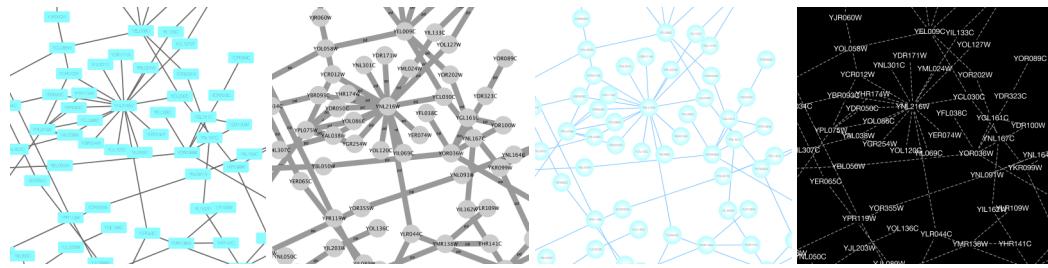
Add photo/image/graphics on top of nodes.





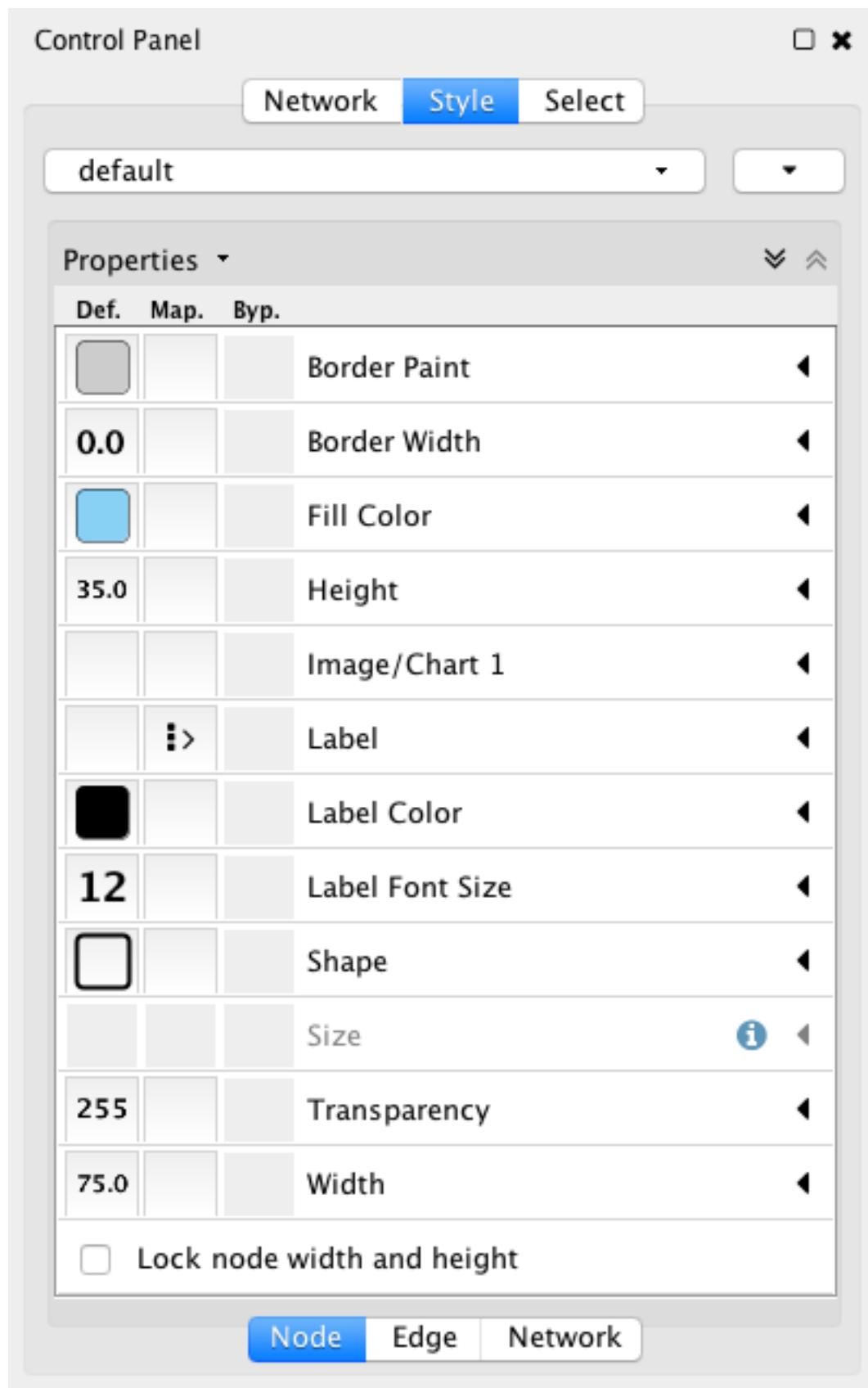


Cytoscape 3 has several sample styles. Below are a few examples of these applied to the *galFiltered.sif* network :



12.2 Introduction to the Style Interface

The **Style** interface is located under the **Style** panel of the **Control Panel**.



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

- At the top of the interface, there is a drop-down menu for selecting a pre-defined style. There is also an **Options** drop-down with options to rename, remove, create and copy a Style, and an option to create a legend for the selected Style.
- The main area of the interface is composed of three tabs, for Node, Edge and Network.
- Each tab contains a list of properties relevant to the current style. At the top of the list a **Properties** drop-down allows you to add additional properties to the list.
- Each property entry in the list has 3 columns:
 - The **Default Value** shows just that, the default value for the property. Clicking on the **Default Value** column for any property allows you to change the default value.
 - **Mapping** displays the type of mapping currently in use for the property. Clicking on the **Mapping** column for any property expands the property entry to show the interface for editing the mapping. Details on the mapping types provided [here](#).
 - **Bypass** displays any style bypass for a selected node or edge. Note that a node/edge or subset of nodes/edges must be selected to activate the **Bypass** column. Clicking on the **Bypass** column for selected node(s)/edge(s) allows you to enter a bypass for that property for selected node(s)/edge(s).

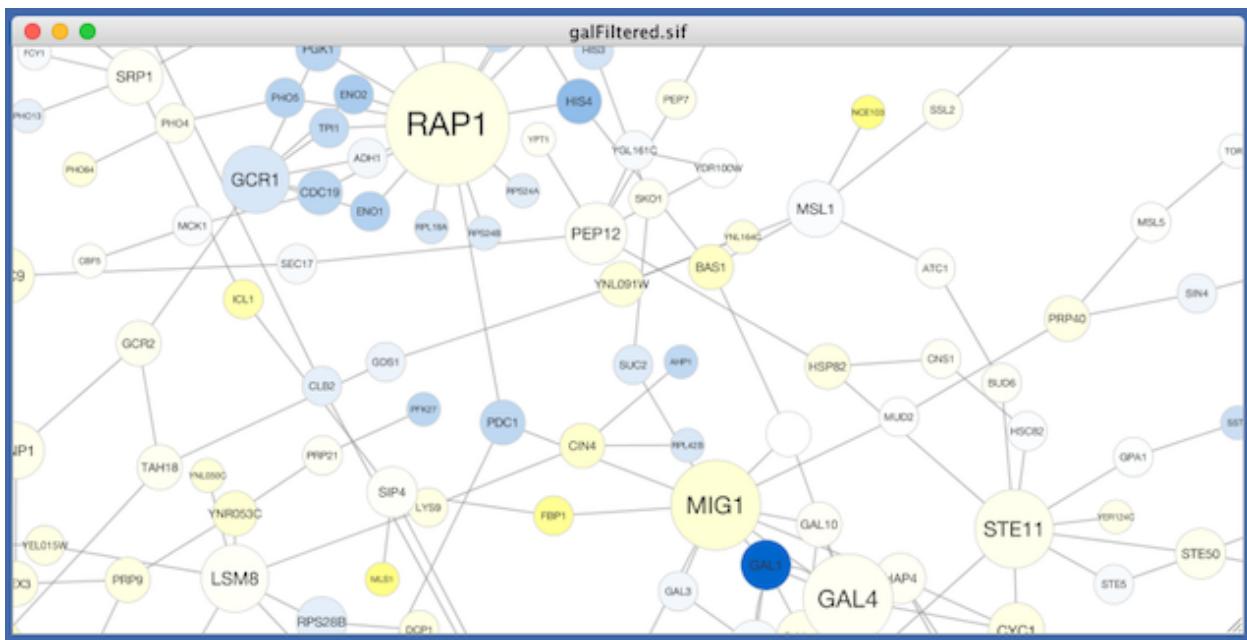
The **Default Value** is used when no mapping is defined for a property, or for nodes/edges not covered by a mapping for a particular property. If a **Mapping** is defined for a property, this defines the style for all or a subset of nodes/edges, depending on how the mapping is defined. A **Bypass** on a specific set of nodes/edges will bypass and override both the default value and defined mapping.

12.3 Introduction to Style

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

Step 1. Load some sample data

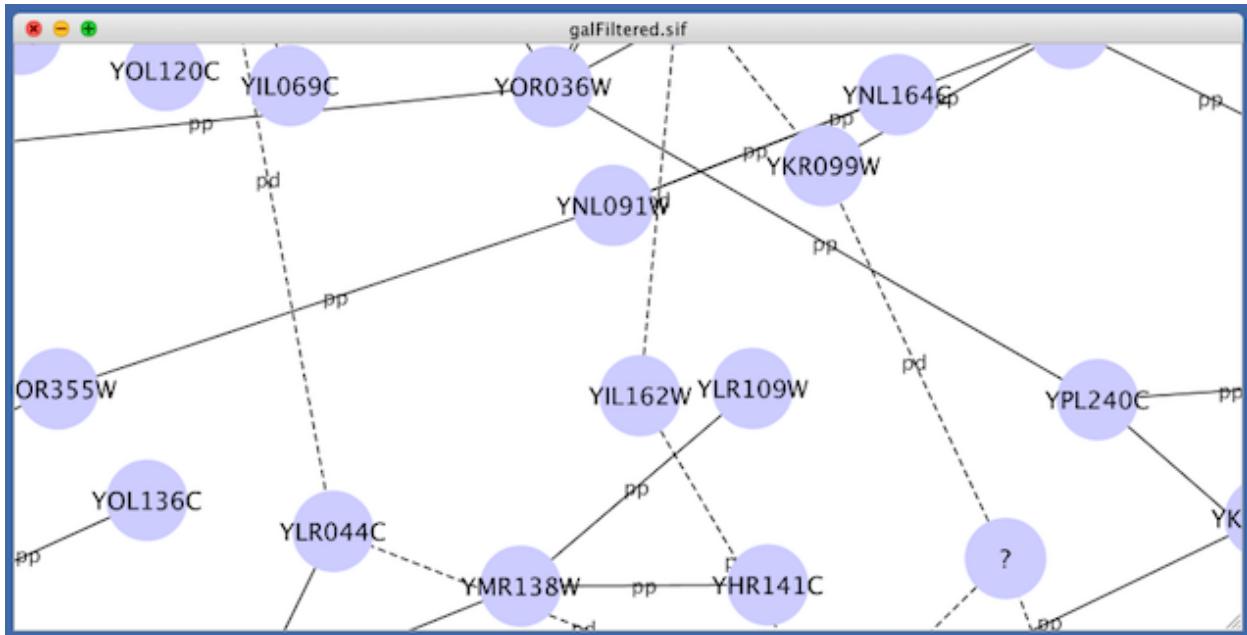
- Load a sample session file: From the main menu, select **File** → **Open...**, and select the file *sampleData/galFiltered.cys*.
- The session file includes a network, some annotations, and sample styles. By default, the style **galFiltered Style** is selected. Gene expression values for each node are colored along a color gradient between blue and yellow (where blue represents a low expression ratio and yellow represents a high expression ratio, using thresholds set for the **gal1RGexp** 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:



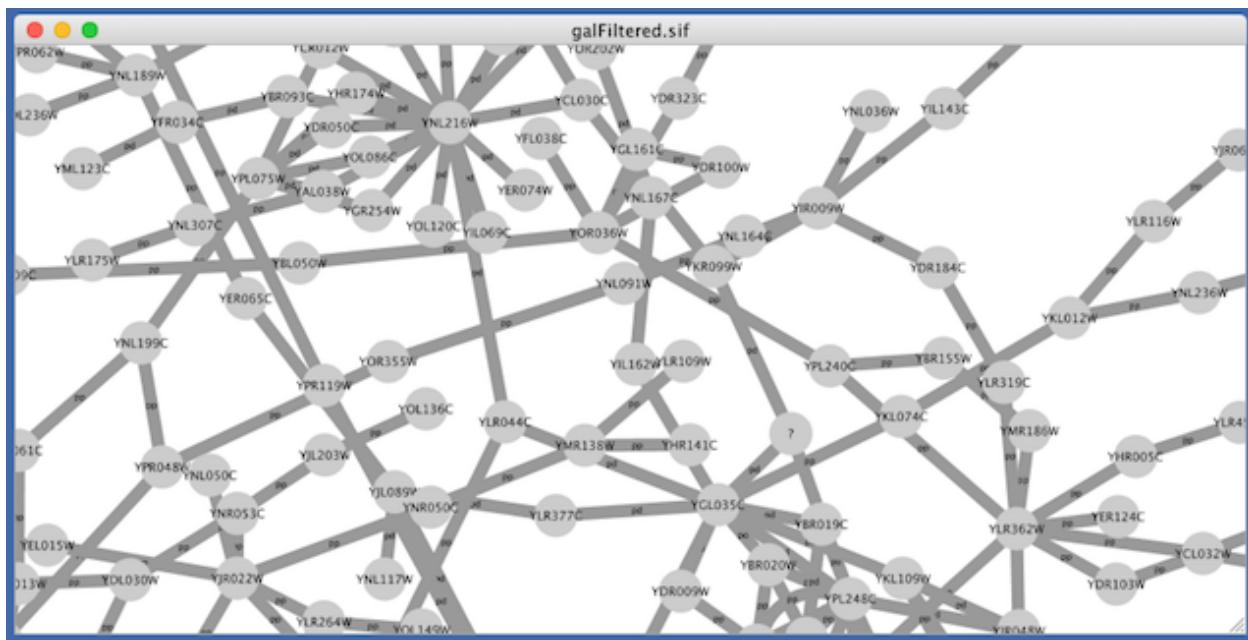
Step 2. Switch between different styles

You can change the style by making a selection from the **Current Style** drop-down list, found at the top of the **Style** panel.

For example, if you select **Sample1**, a new 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 style does not have mappings except node/edge labels, but you can modify the network graphics by editing the *Default Value* for any property.

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

12.3.1 List of Node, Edge and Network Properties

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

12.3.2 Available Shapes and Line Styles

12.3.3 How Mappings Work

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

1. Passthrough Mapping

- The values of network column data are passed directly through to properties. A passthrough mapping is typically used to specify node/edge labels. For example, a passthrough mapping can label all nodes with their common gene names.

2. Discrete Mapping

- Discrete column data are mapped to discrete properties. For example, a discrete mapping can map different types of molecules to different node shapes, such as rectangles for gene products and ellipses for metabolites.

3. Continuous Mapping

- Continuous data are mapped to properties. Depending on the property, there are three kinds of continuous mapping:

- i. **Continuous-to-Continuous Mapping:** for example, you can map a continuous numerical value to node size.
- ii. **Color Gradient Mapping:** This is a special case of continuous-to-continuous mapping. Continuous numerical values are mapped to a color gradient.
- iii. **Continuous-to-Discrete Mapping:** 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 mapping support for each property.

Legend

Node Mappings

Edge Mappings

Text Passthrough Mapping

In Cytoscape 2.8.0 and later versions, the Passthrough Mapping can recognize some text representations of values. This means, if you have a string column named *Node Size Values*, you can directly map those values as the Node Size by setting “Node Size Values” as controlling column with **Node Size** “Passthrough Mapping”. The following value types are supported:

- **Colors:** Standard color names supported by all browsers or RGB representation in hex
- **Numerical Values:** Automatically mapped to the specified property.
- **Images:** 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

Color Passthrough Mapping

Node Size Passthrough Mapping

Image Passthrough Mapping

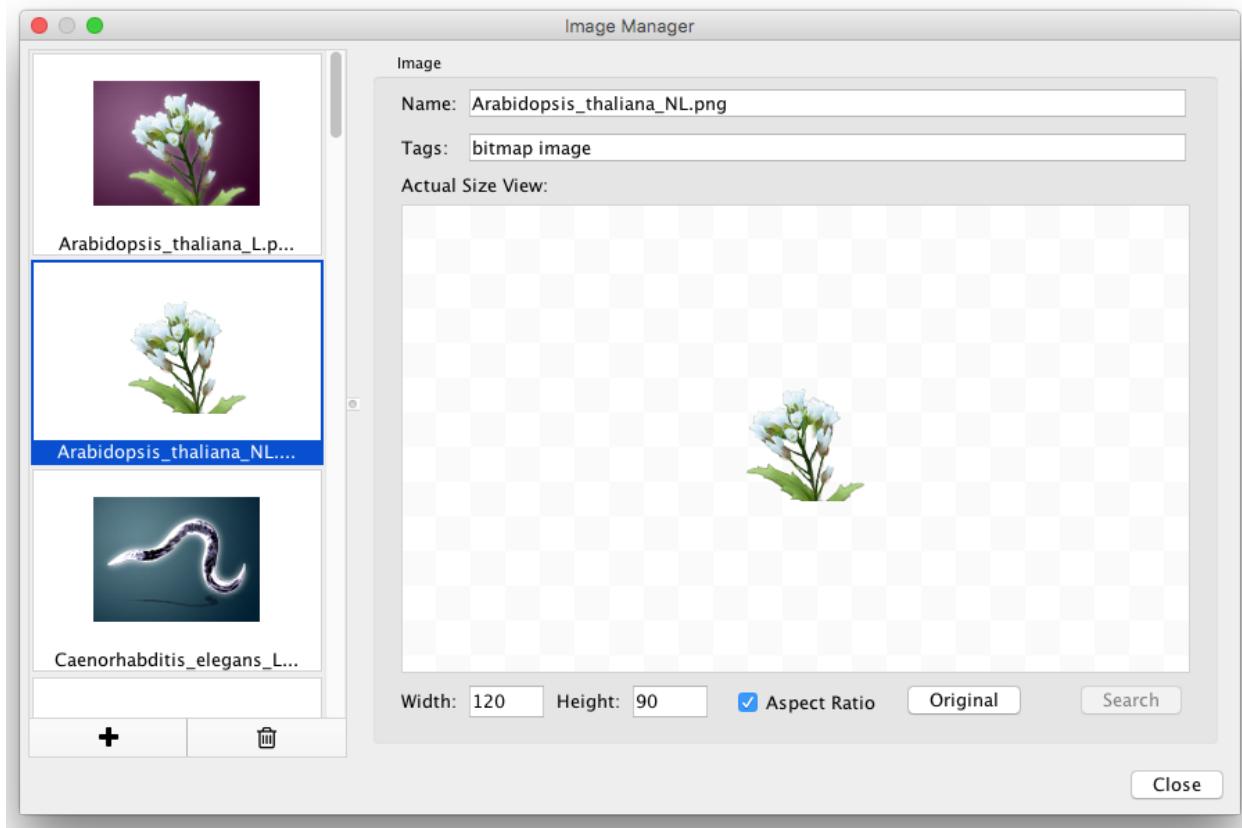
12.4 Images, Charts and Gradients

Cytoscape allows you to set custom graphics to nodes. Using the Style interface, you can map **Image/Chart** properties to nodes like any other property. Cytoscape provides a set of images and you can also add your own images in the **Image Manager**, as well as remove or modify existing ones.

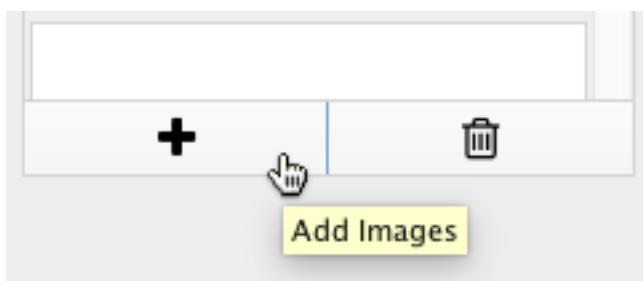
Taxonomy Icon (http://biosciencedbc.jp/taxonomy_icon/taxonomy_icon.cgi?lng=en) set used in this section is created by Database Center for Life Science (DBCLS) and is distributed under Creative Commons License (CC BY 2.1.)

12.4.1 Managing Images

The **Image Manager** is available under the menu option **View → Open Image Manager...**

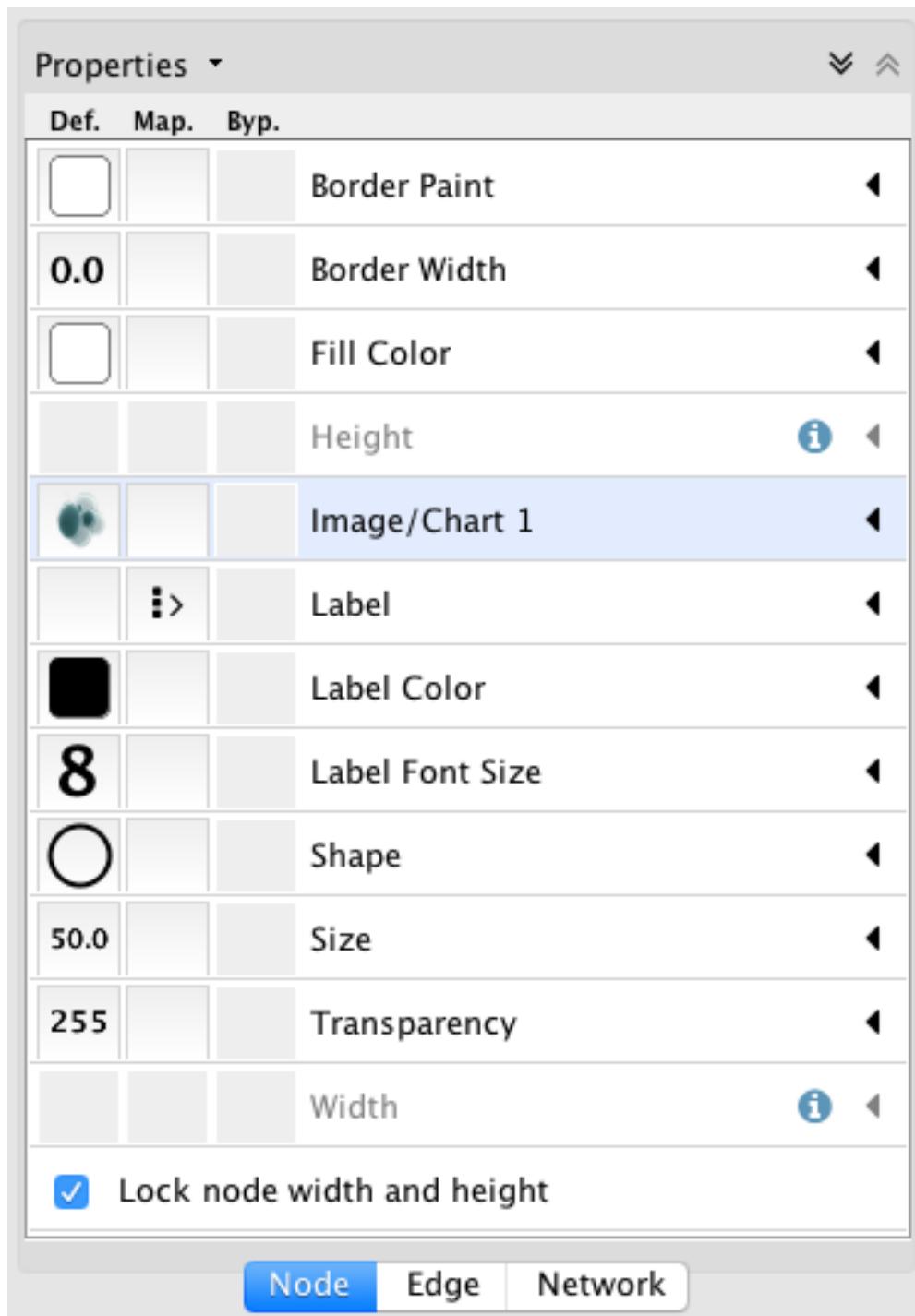


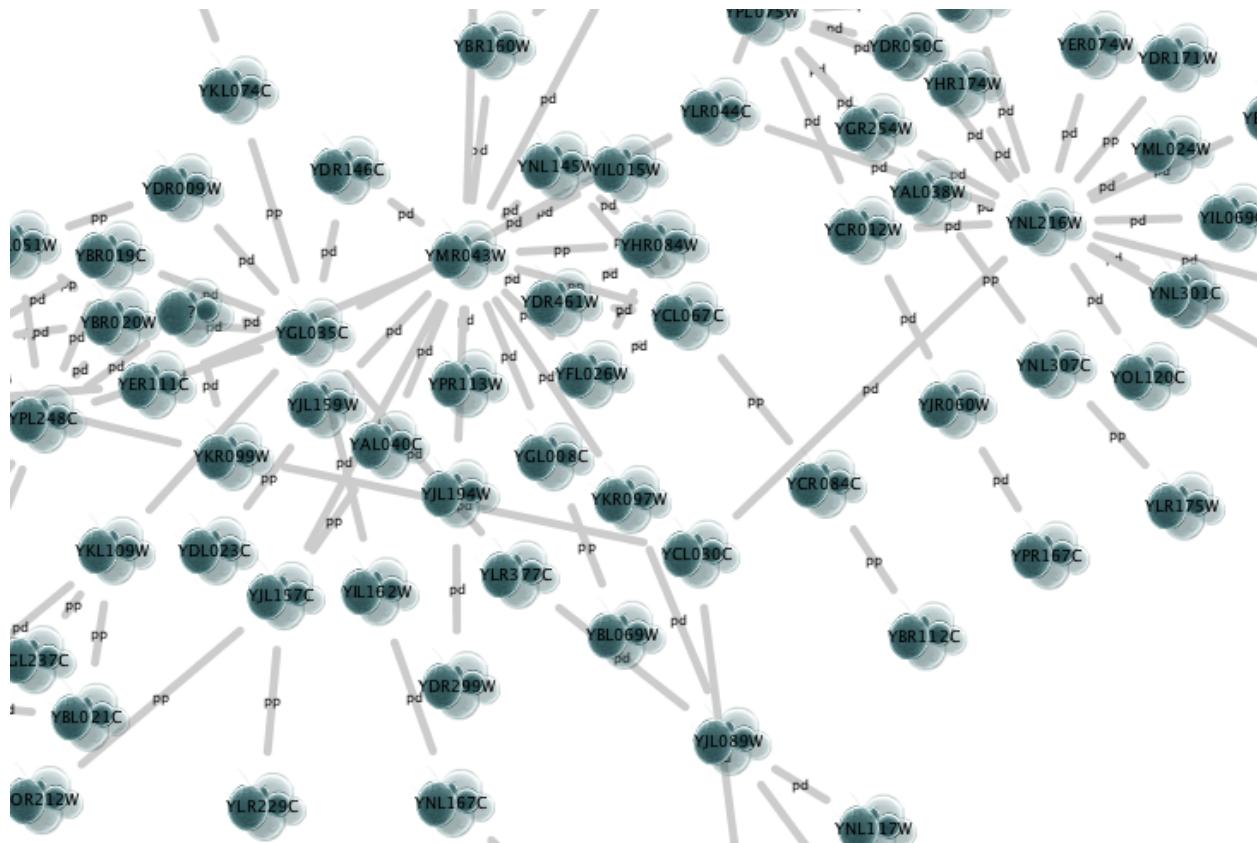
- You can add images by drag-and-drop of image files and URLs. If you want to add images from a web browser or local file system, you can drag images from them and drop those images onto the list of images on the left.
 - **Note:** When you drag and drop images from web browser, make sure that you are 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.
- If you want to add one or more images from a folder, press the + button on the bottom of the Image Manager window and then select the images you want to add.



- To remove images from the current session's image library, simply select one or more images from the list and press the **Remove Selected Images** button (trash icon).
- Images can be resized by defining specific **Width** and **Height** values. If the **Aspect Ratio** box is checked, the width-height ratio is always synchronized. You can resize the image to the original size by pressing the **Original** button.

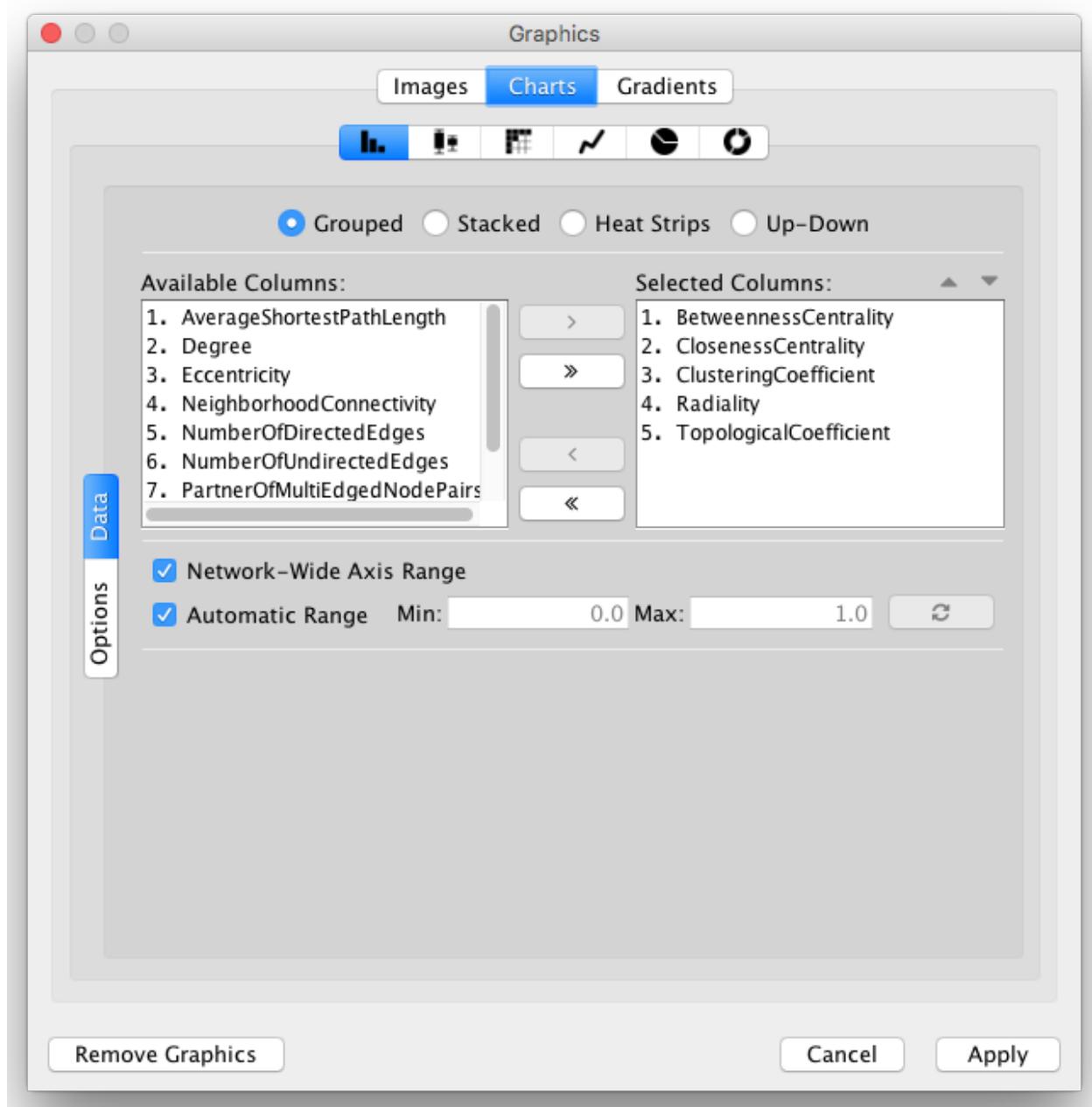
12.4.2 Using Graphics in Styles

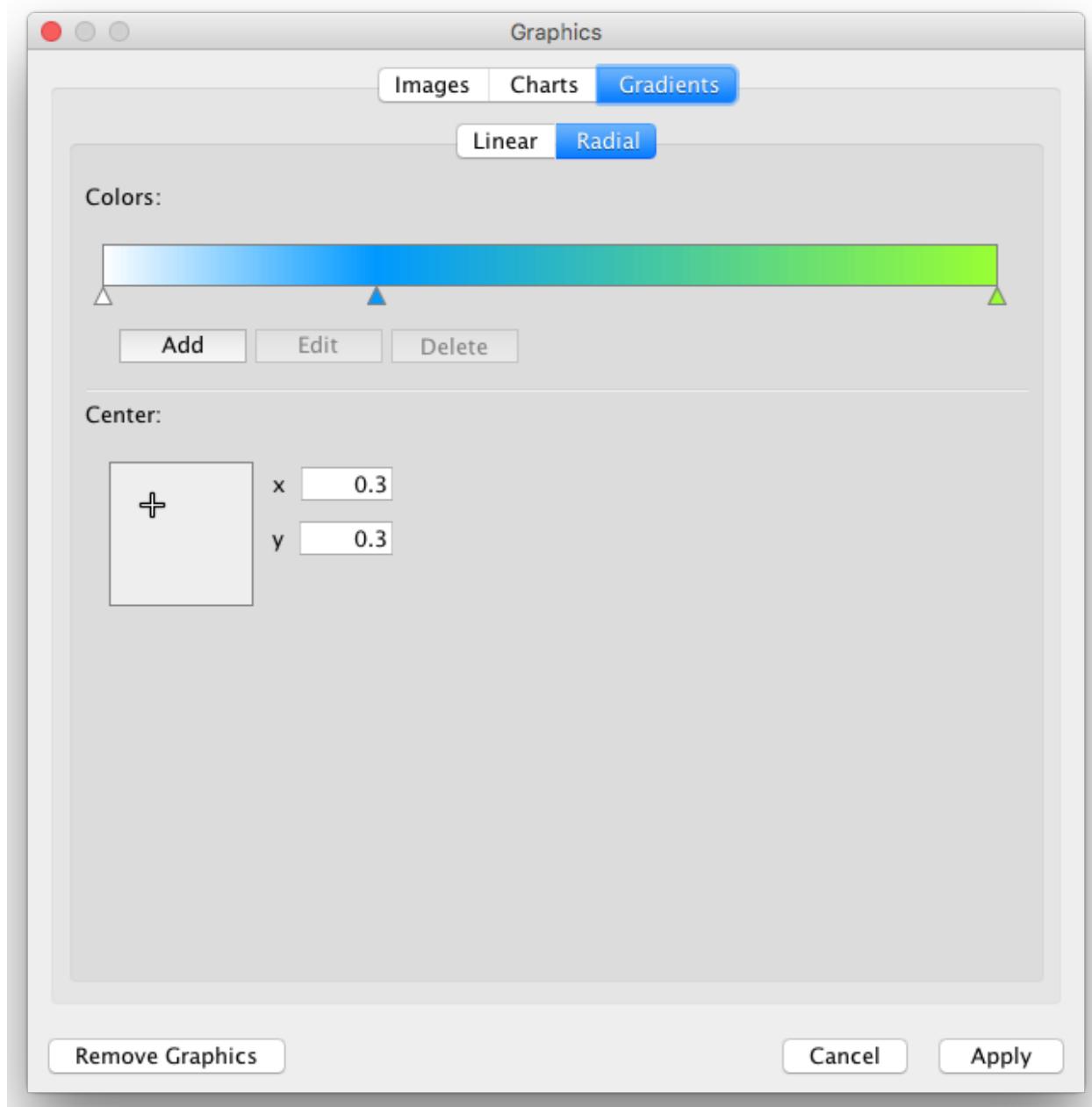




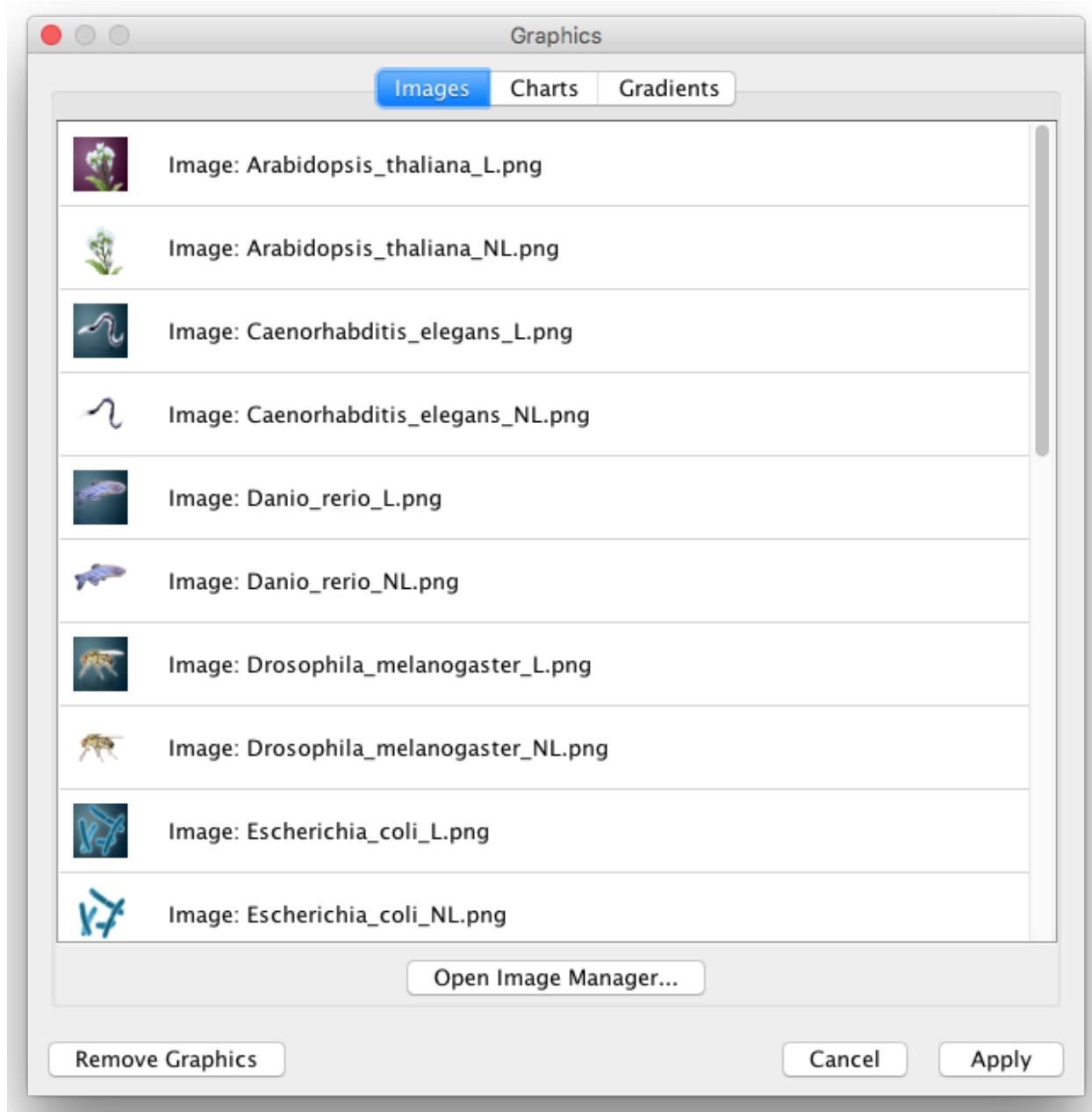
Node graphics are used and defined like any other property, through the **Style** interface. There are nine **Image/Chart** properties.

- Cytoscape provides three kinds of graphics (selectable via tabs on the **Graphics** dialog):
 - **Images:** You can select one of the provided images or add your own (click the **Open Image Manager...** button to add more images to the list).
 - **Charts:** The following chart types are available: Bar, Box, Heat Map, Line, Pie, Ring.
 - **Gradients:** You can also set *Linear* and *Radial* gradients to nodes.





- To add a graphic, first add one **Image/Chart** property to the properties list in the **Style** interface (on the **Node** tab, select **Properties** → **Paint** → **Custom Paint n** → **Image/Chart n**). Next, click the **Default Value** column of the **Image/Chart** property to bring up the **Graphics** dialog. Select an image, a chart or a gradient and then click **Apply**.
 - By default, graphics are automatically resized to be consistent with the **Node Size** property.
- To remove an image, chart or gradient, click the **Remove Graphics** button on the **Graphics** dialog.



Graphics Positions

Each **Image/Chart** property is associated with a position. You can edit its position by using the UI available in the **Default Value** column for the **Image/Chart Position** property that has the same number. For instance, the **Image/Chart Position 2** value modifies the position of **Image/Chart 2**.

- **Note:** Setting graphics positions for *Linear* or *Radial* gradients has no effect, as they are always centered on the node.

Z-Ordering

The number that appears with the **Image/Chart** property represents an ordering of layers. Basic node color and shape are always rendered first, then node Image/Chart 1, 2, ..., through 9.

12.4.3 Saving and Loading Images

In general, saving and loading images is automatic. When you quit Cytoscape, all of the images in the **Image Manager** will be saved automatically. There are two types of saving:

1. To a session file
 - When you save the current session to a file, the images used in the current styles will be saved to that file. For example, if you have a style with a discrete mapping for **Image/Chart 1**, all images used in the style will be saved to the session file. Other images 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 Image Manager and it takes a very long time to save and load the session file.
2. Automatic saving to CytoscapeConfiguration/images3 directory
 - When you select **File** → **Quit** (Windows and Linux) or **Cytoscape** → **Quit Cytoscape** (Mac OS X), all of the images in the Image Manager will be saved automatically to your Cytoscape settings directory. Usually, they are saved in `YOUR_HOME_DIRECTORY/CytoscapeConfiguration/images3`.

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

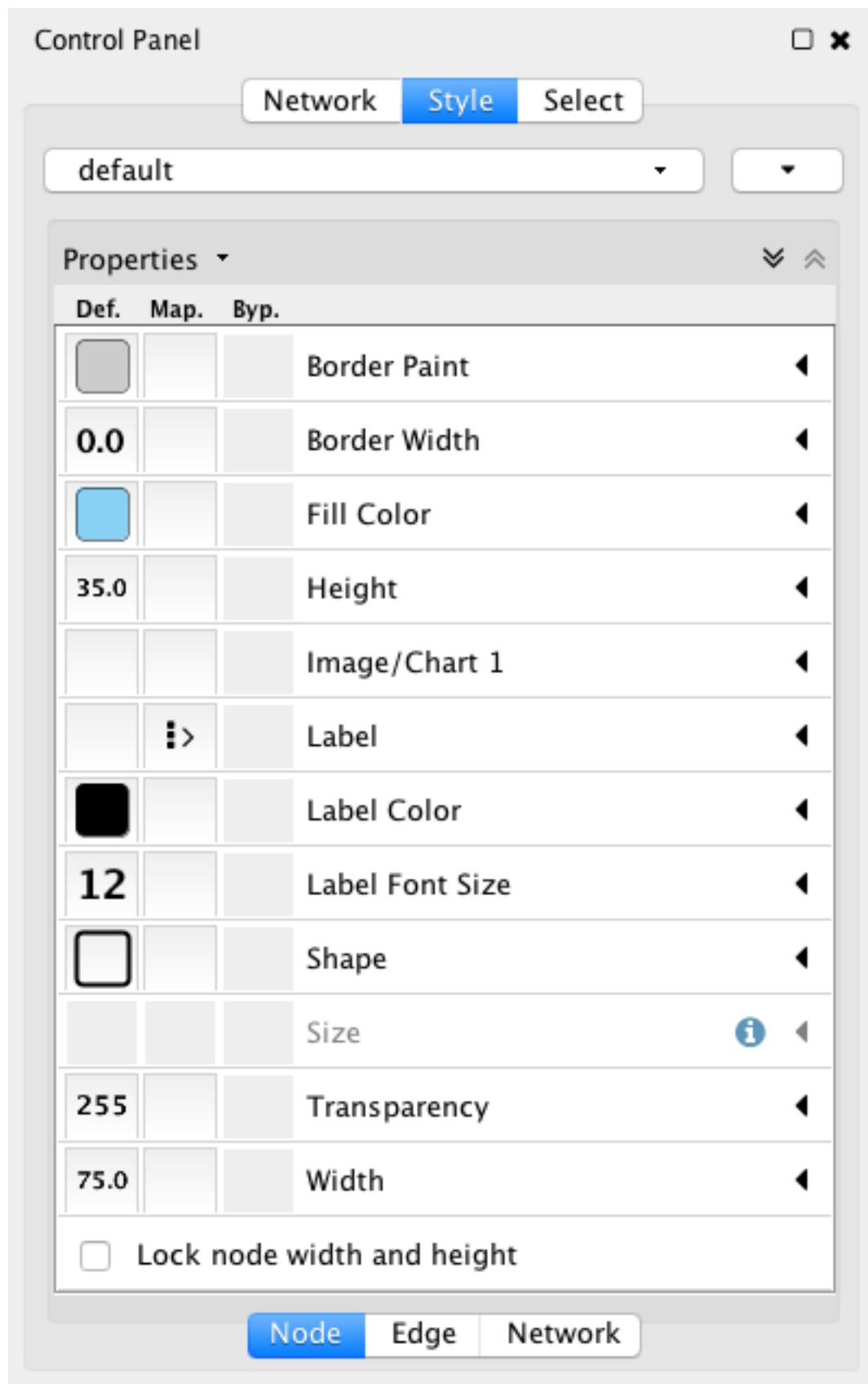
12.5 Styles Tutorials

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

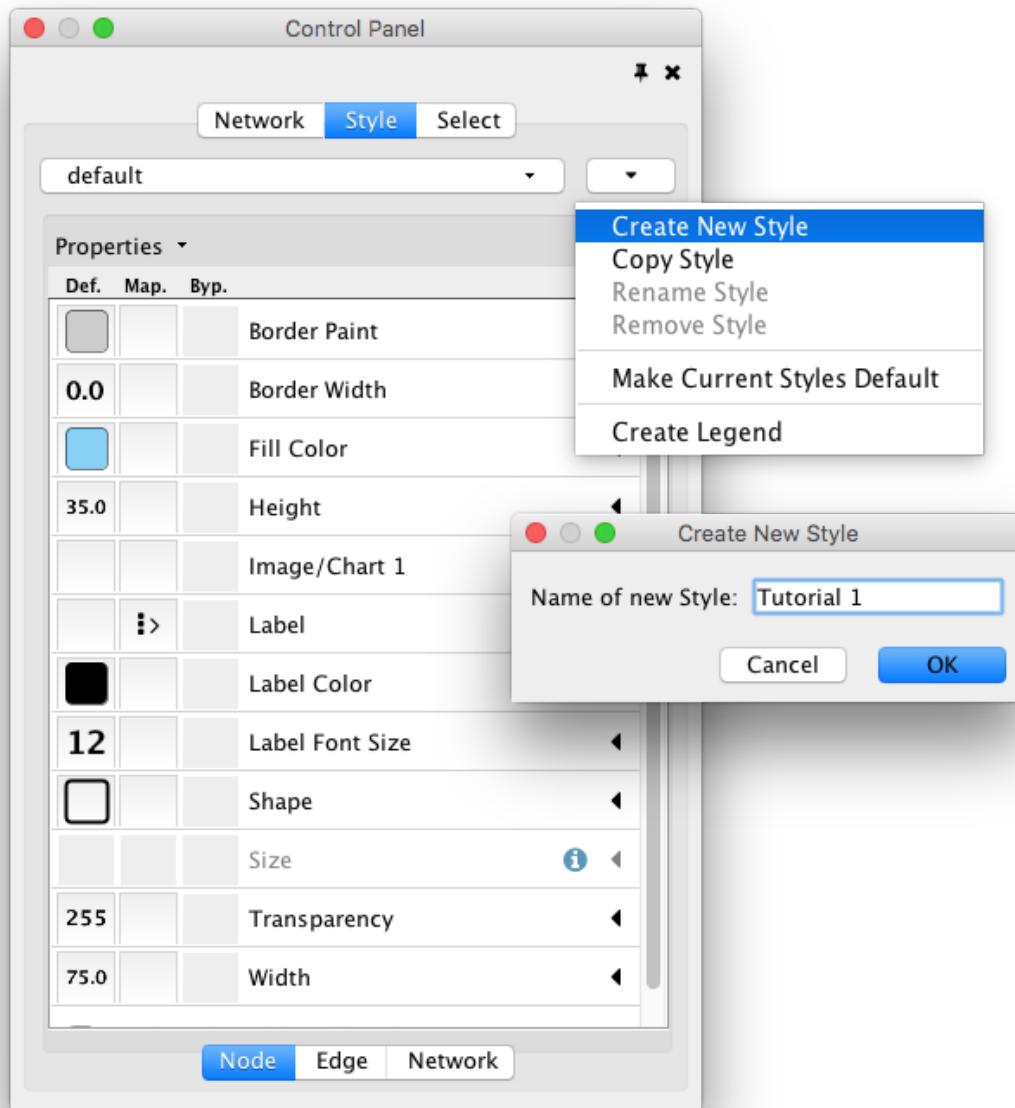
12.5.1 Tutorial 1: Creating a Basic Style and Setting Default Values

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

1. **Load a sample network:** From the main menu, select **File** → **Import** → **Network** → **File...**, and select `sampleData/galFiltered.sif`.
2. **Create some node/edge statistics:** The **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 regular table data.
3. Select the **Style** panel in the Control Panel.

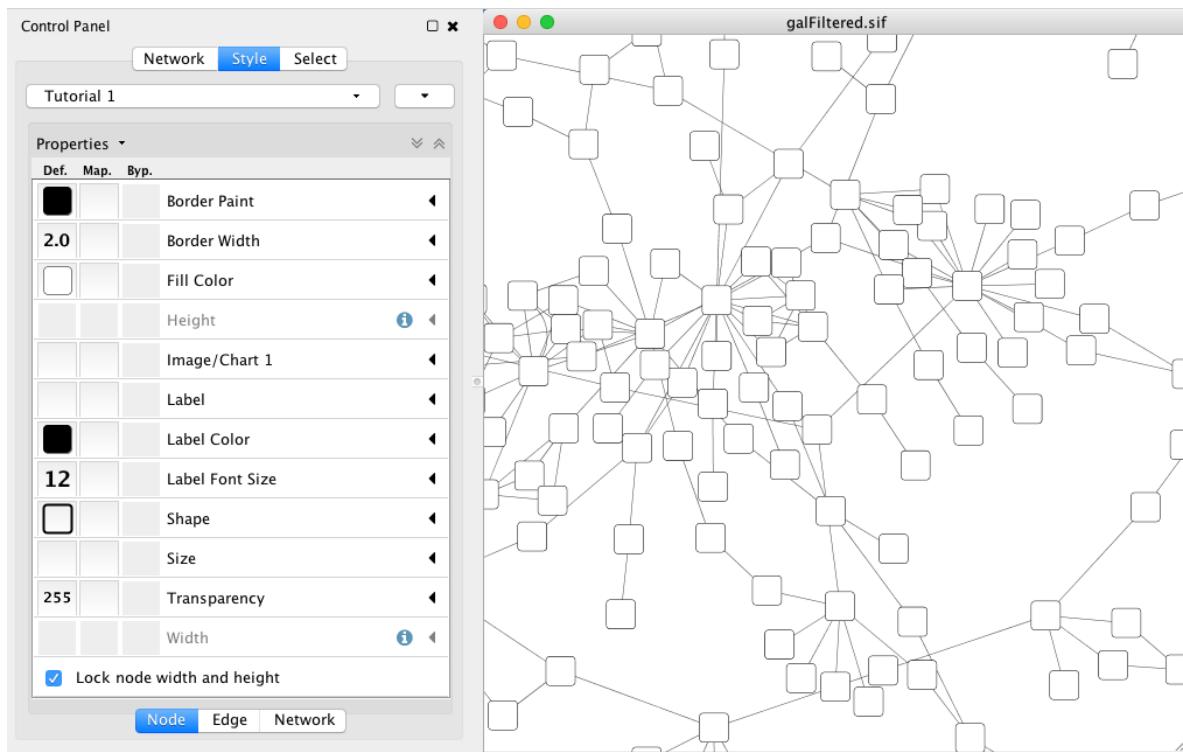


4. **Create a new style:** Click the Options  drop-down, and select **Create New Style**. Enter a name for your new style when prompted.



Since no mappings are set up yet, only default values are defined for some of the properties. From this panel, you can create node/edge mappings for all properties.

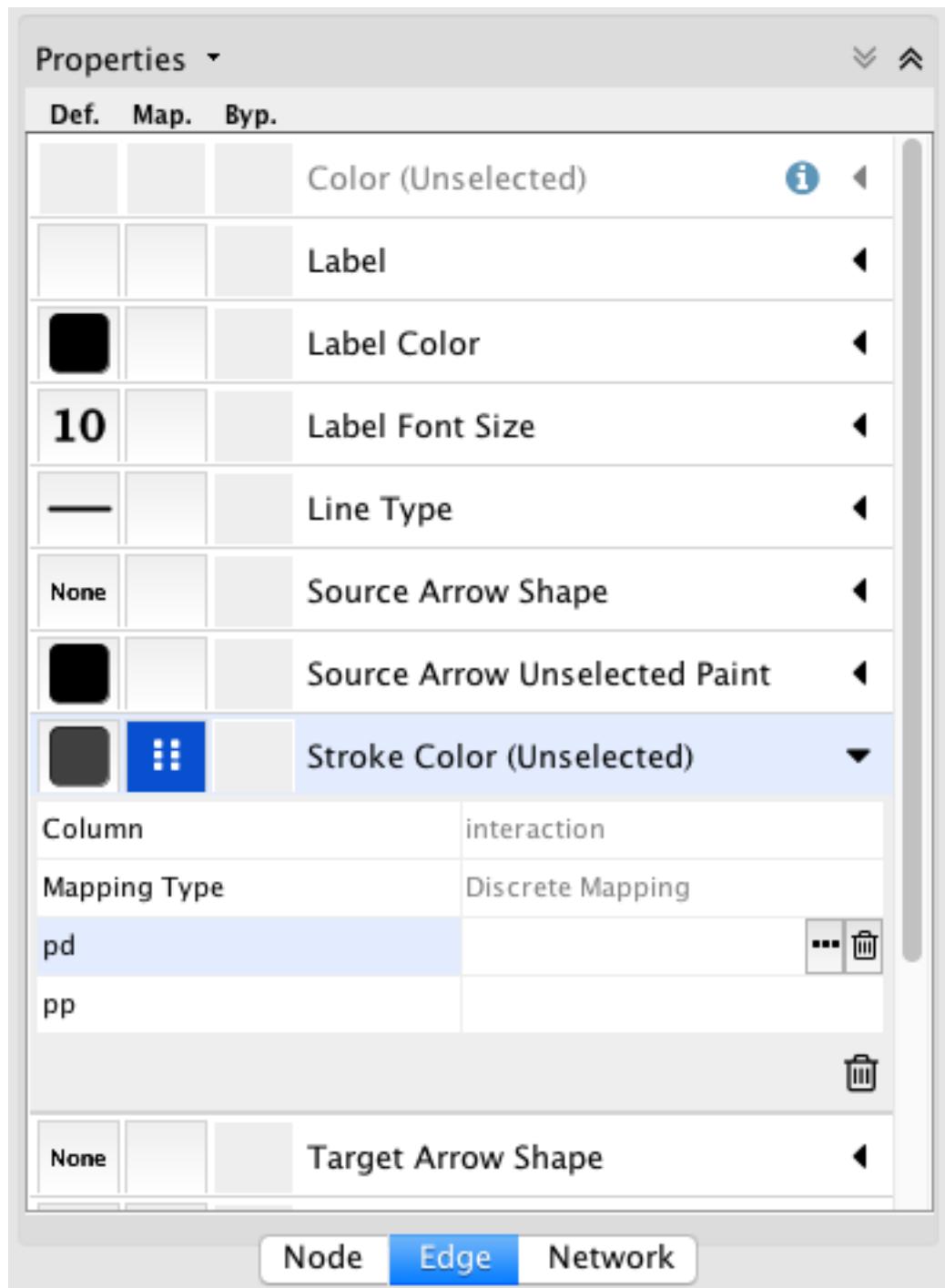
1. **Change the default node color and shape:** To set the default node shape to triangles, click the **Default Value** column for the **Shape** property. A list of available node shapes will be shown. Select the **Triangle** item and click the **Apply** button. You can edit other default values in the same way. In the example shown below, the node shape is set to **Round Rectangle**, while **Fill Color** is set to white. The new Style is automatically applied to the current network, as shown below.



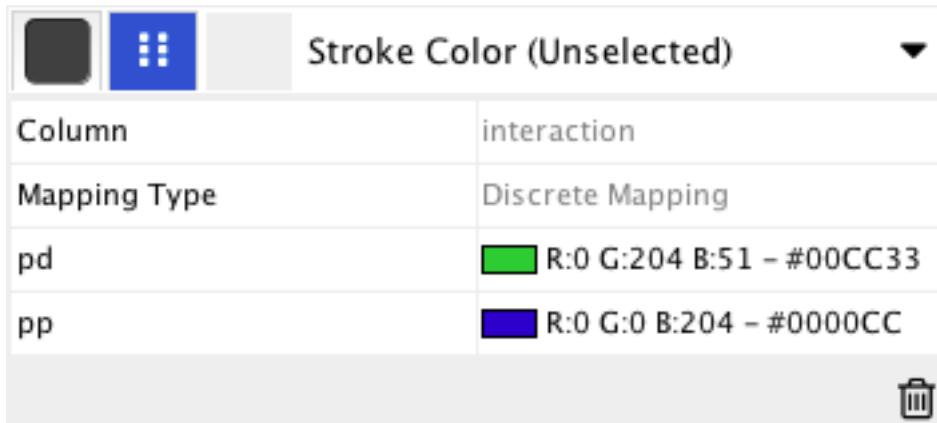
12.5.2 Tutorial 2: Creating a New Style with a Discrete Mapping

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

- Find the property:** In the Edge tab of the Style panel, find the **Stroke Color (Unselected)** property. If it is not already visible in the properties sheet, add it by selecting the drop-down item **Properties** → **Paint** → **Color (Unselected)** → **Stroke Color (Unselected)**.
- Choose a data column to map to:** Expand the entry for **Stroke Color (Unselected)** by clicking the arrow icon on the right. Click the **Column** entry and select “interaction” from the drop-down list that appears.
- Set the mapping type:** Under **Mapping Type**, select “Discrete Mapping”. All available column values for “interaction” will be displayed, as shown below.

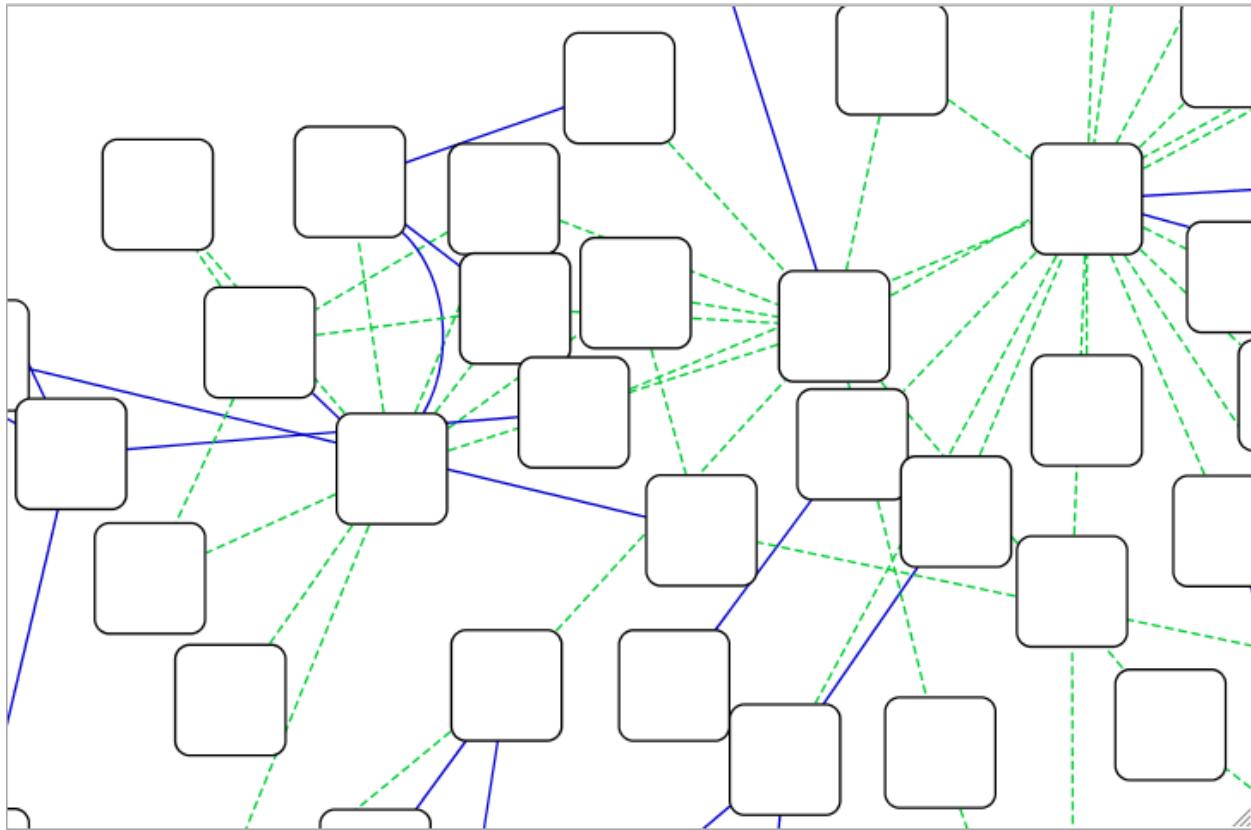


4. **Set the mapped values:** Click the empty cell next to “pd” (protein-DNA interactions). On the right side of the cell, click on the ... button that appears. A popup window will appear; select green or similar, and the change will immediately appear on the network window.



Repeat step 4 for “pp” (protein-protein interactions), but select a darker color. Then repeat steps 3 through 4 for the *Line Type* property, by selecting 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.



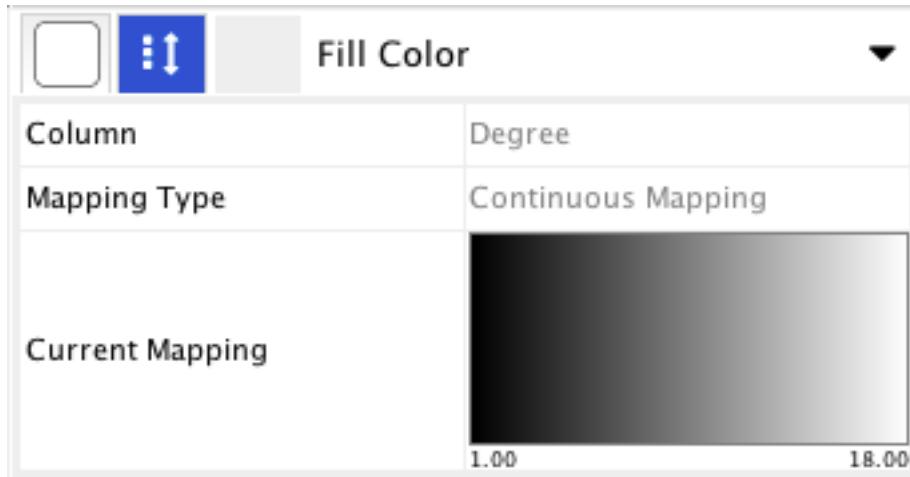
12.5.3 Tutorial 3: Creating a New Style with a Continuous Mapping

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

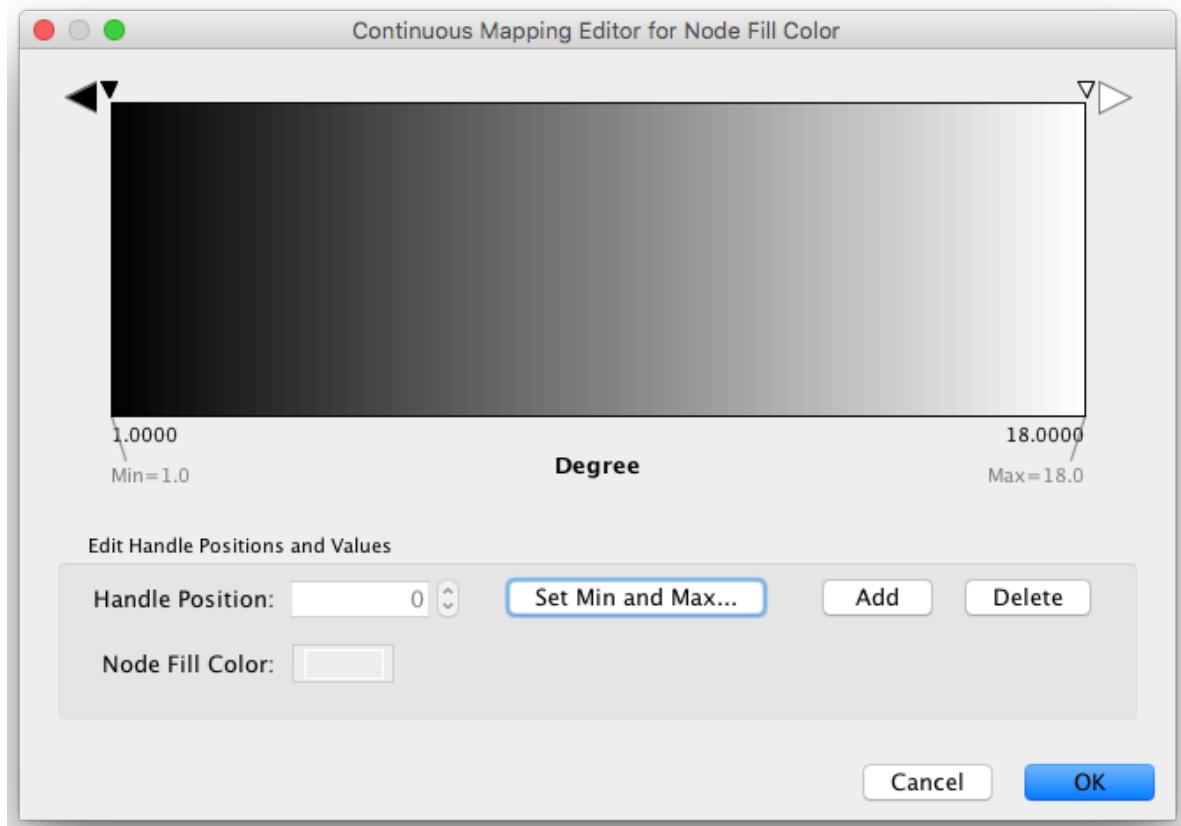
1. **Find the property:** In the Node tab of the Style panel, find the **Fill Color** property. If it is not already visible

in the properties sheet, add it by selecting the drop-down item **Properties** → **Paint** → **Fill Color**.

2. **Set the node table column:** Expand the entry for **Fill Color** by clicking the arrow icon on the right. Click the **Column** entry and select “Degree” from the drop-down list that appears.
3. **Set the mapping type:** Set the “Continuous Mapping” option as the **Mapping Type**. This automatically creates a default mapping.



4. **Define the points where colors will change:** Double-click on the black-and-white gradient rectangle next to **Current Mapping** to open the **Continuous Mapping Editor**. Note the two smaller triangles at the top of the gradient.

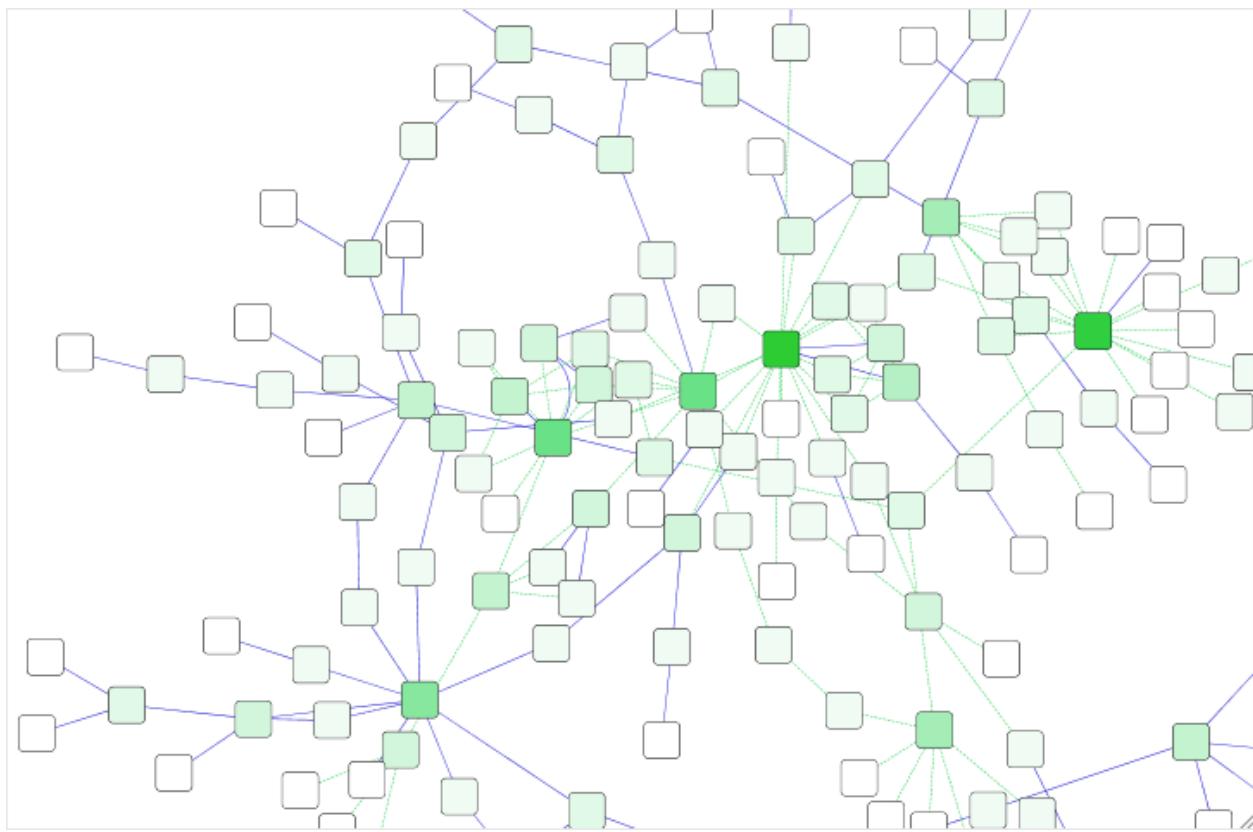


5. **Define the colors between points:** Double-click on the larger leftmost triangle (facing left) and a color palette

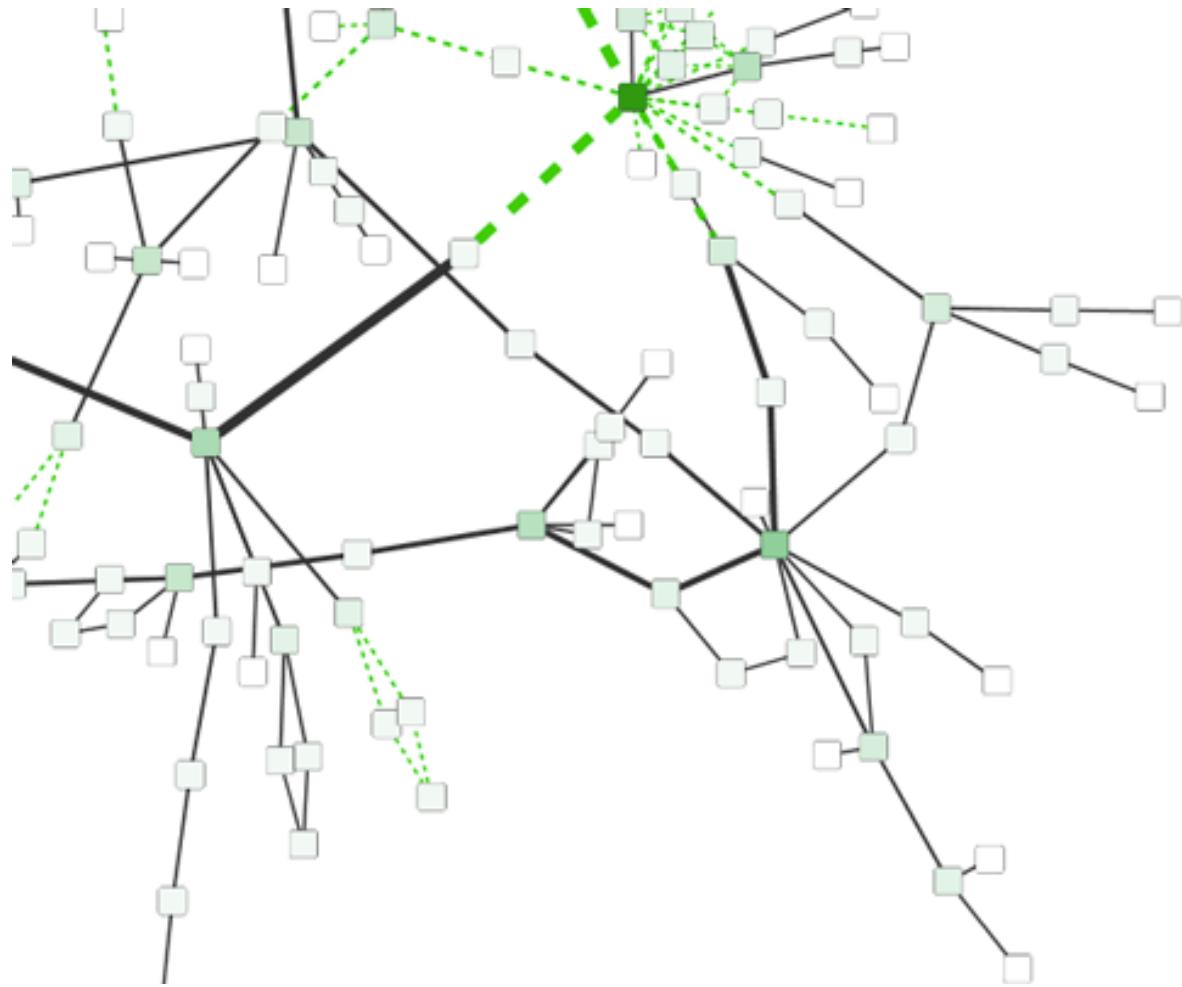
will appear. Set the color white and repeat for the smaller left-side triangle. For the triangle on the right, set the color green and then choose the same color for the smaller right-side triangle.



The color gradients will immediately appear on the network. All nodes with degree I will be set to white, and all values between I and 18 will be painted with a white/green color gradient (see the sample screenshot below).



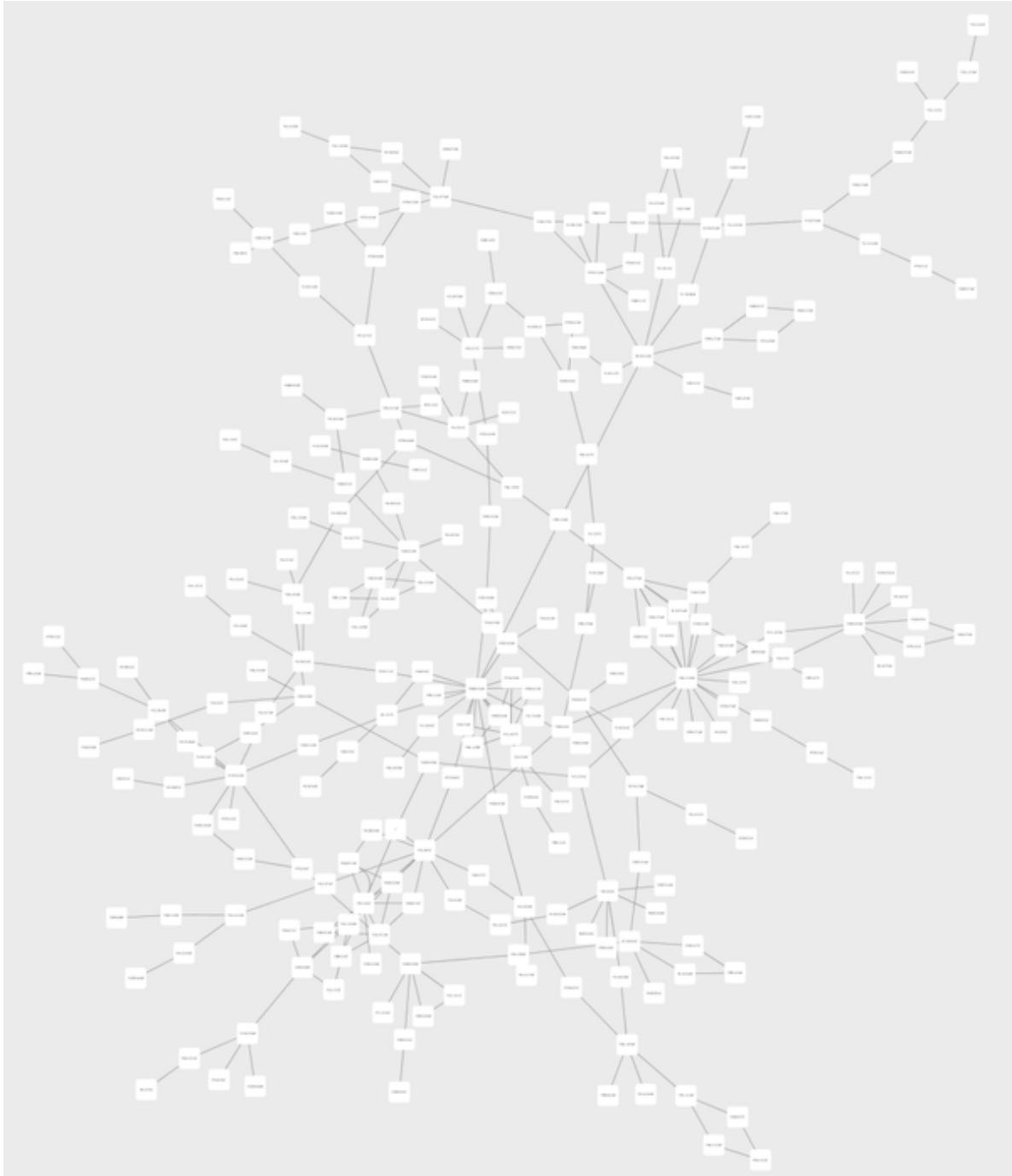
- **Repeat for other properties:** You can create more continuous mappings for other numeric table data. For example, edge data table column “EdgeBetweenness” is a number, so you can use it for continuous mapping. The following is an example visualization which maps Edge Width to “EdgeBetweenness”.



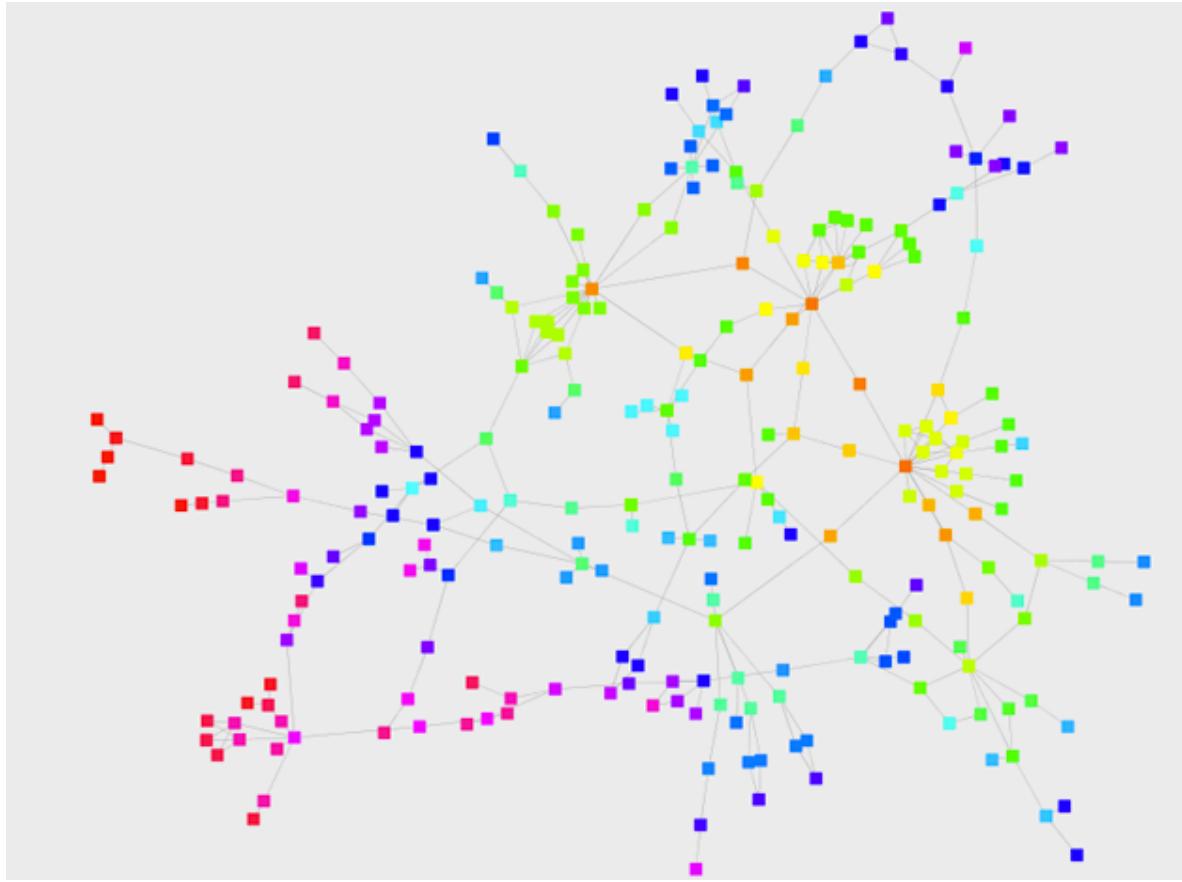
12.5.4 Tutorial 4: Setting Automatic Values to a Discrete Mapping

The goal of this section is to learn how to generate automatic values for discrete mappings.

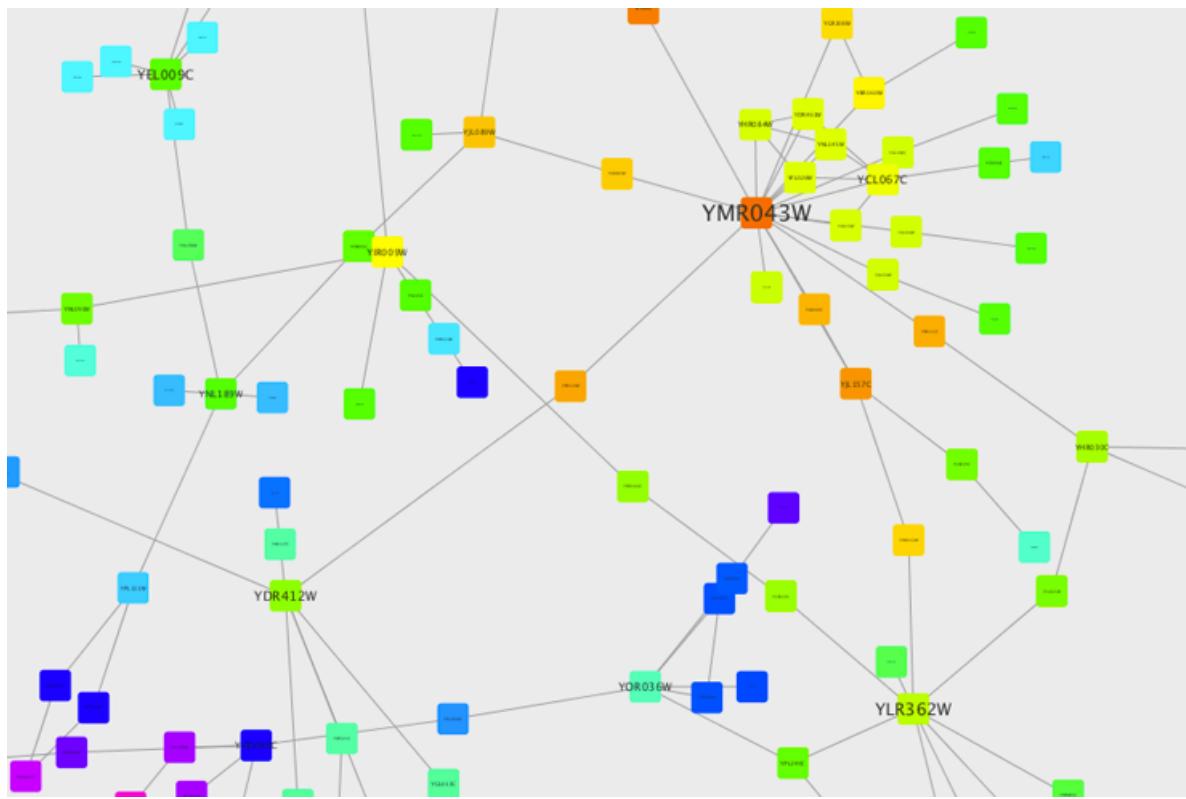
1. Switch the Current Style to **Minimal**. Now your network looks like the following:



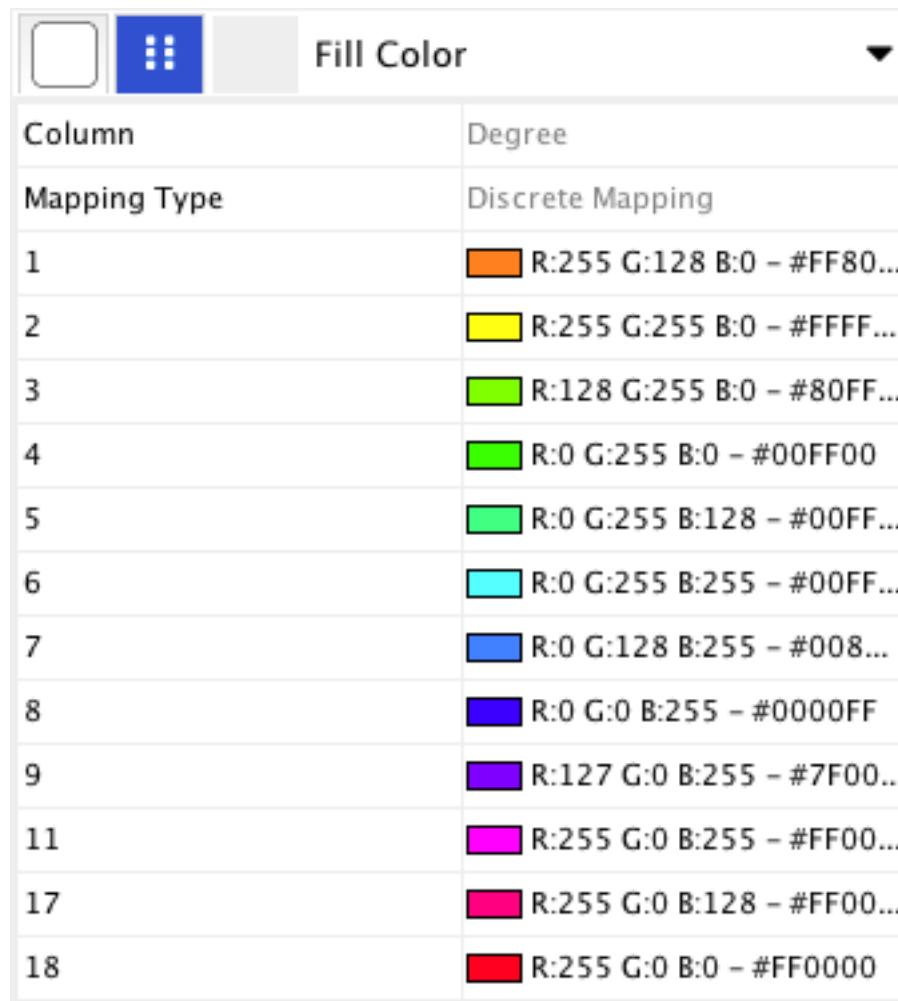
2. Create a discrete mapping for **Fill Color**. Select “AverageShortestPathLength” (generated by the Network Analyzer) as the controlling property.
3. Right-click the **Fill Color** cell, then select **Mapping Value Generators → Rainbow**. Cytoscape will automatically generate different colors for all the property values as shown below:



4. Create a discrete mapping for **Label Font Size**. Select “AverageShortestPathLength” as controlling property (**Column** field).
5. Right-click the **Label Font Size** cell, then select **Mapping Value Generators → Number Series**. Type **3** for the first value and click OK. Enter **3** for increment.
6. Apply **Layout → yFiles Organic Layout**. The final view is shown below:



This mapping generator utility is useful for categorical data. The following example shows a discrete mapping that maps the species column to node color.

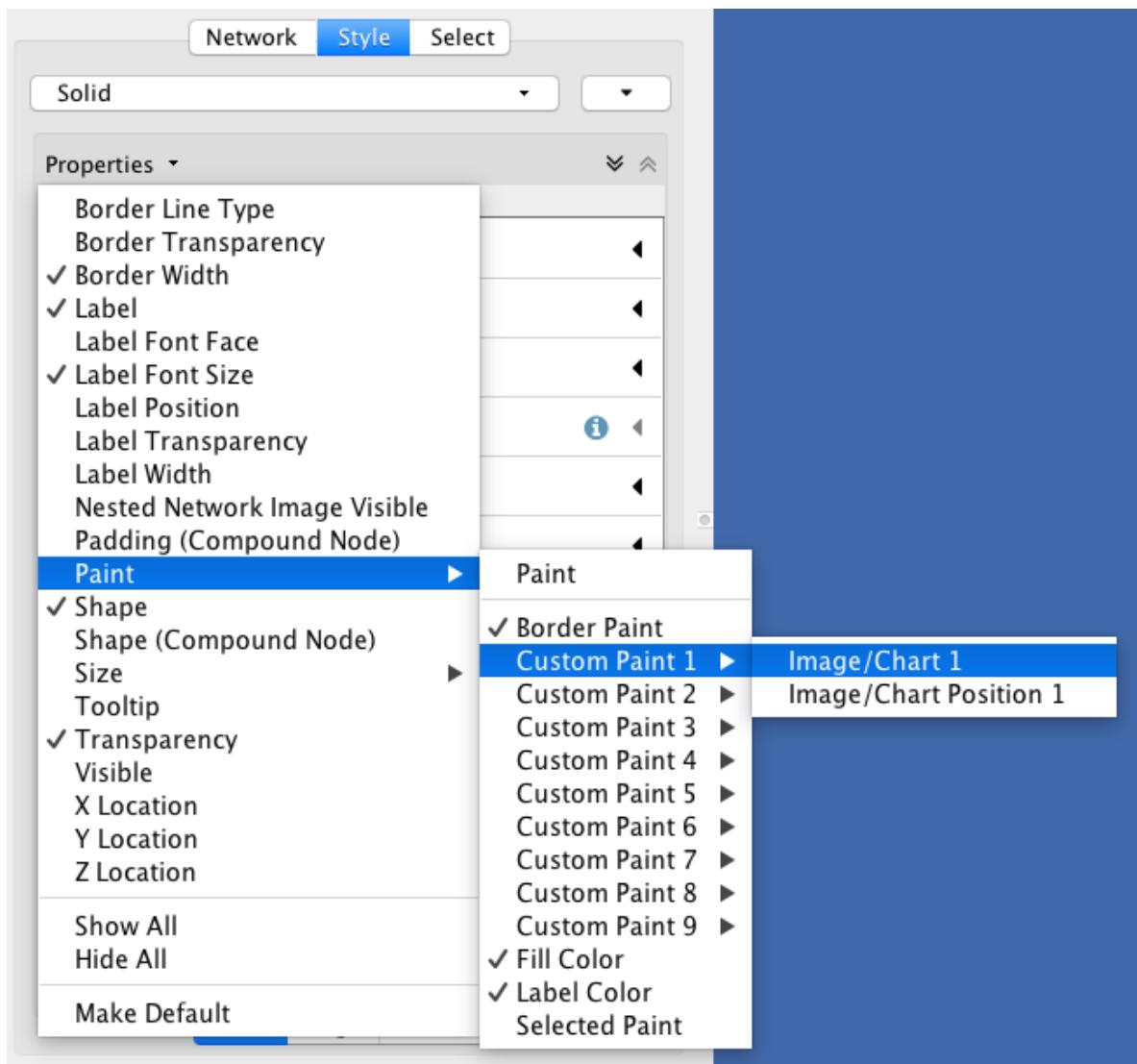


Column	Degree
Mapping Type	Discrete Mapping
1	 R:255 G:128 B:0 – #FF8000
2	 R:255 G:255 B:0 – #FFFF00
3	 R:128 G:255 B:0 – #80FF00
4	 R:0 G:255 B:0 – #00FF00
5	 R:0 G:255 B:128 – #00FF80
6	 R:0 G:255 B:255 – #00FFFF
7	 R:0 G:128 B:255 – #0080FF
8	 R:0 G:0 B:255 – #0000FF
9	 R:127 G:0 B:255 – #7F00FF
11	 R:255 G:0 B:255 – #FF00FF
17	 R:255 G:0 B:128 – #FF0080
18	 R:255 G:0 B:0 – #FF0000

12.5.5 Tutorial 5: Using Images in Styles

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

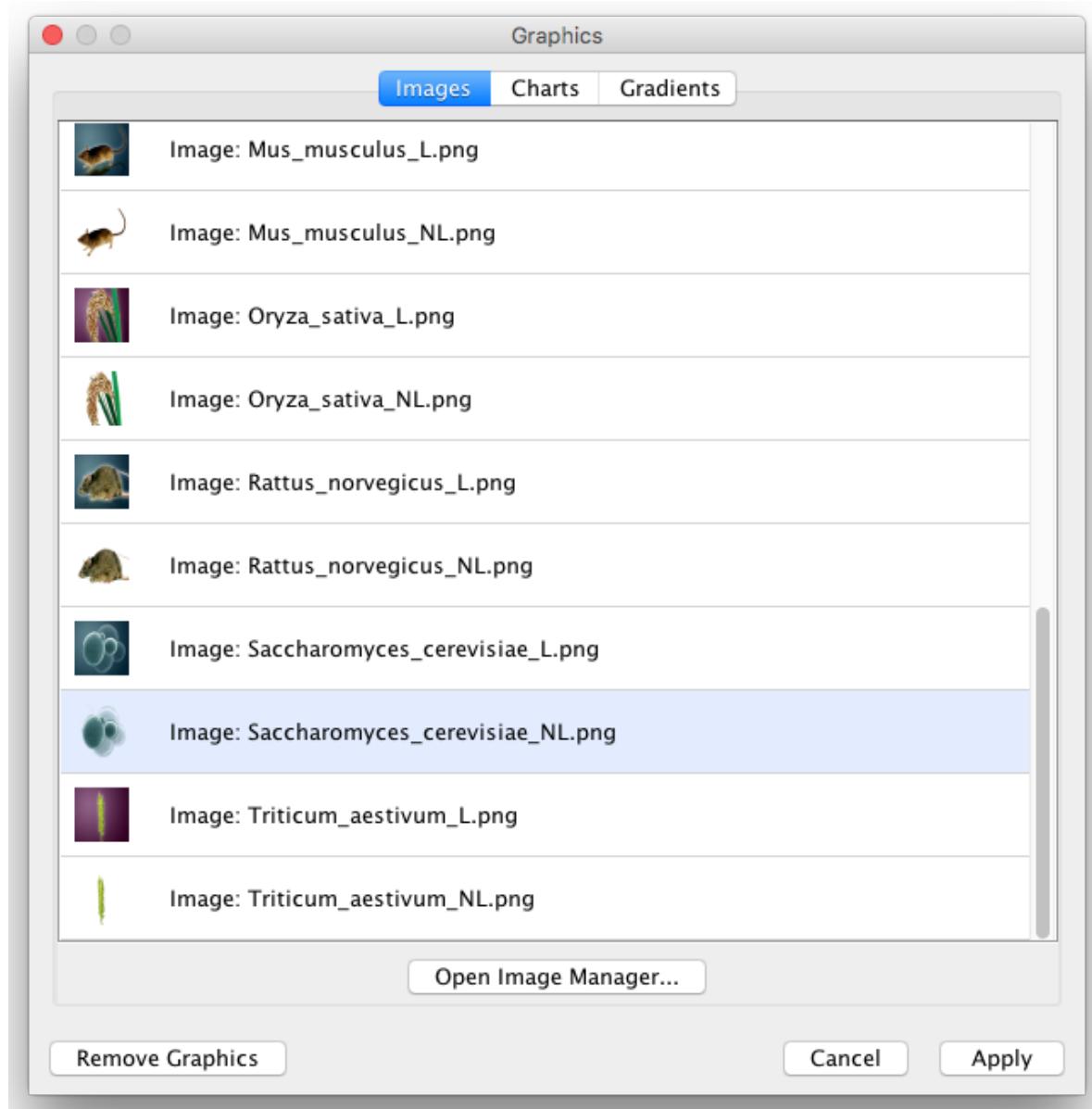
1. In this first example, we will use the presets that Cytoscape 3 has. In general, you can use any type of bitmap graphics. User images should be added to the **Image Manager** before using them in a Style.
2. If you are continuing from the previous tutorials, skip to the next step. Otherwise, load 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).
3. Click the **Style** panel in the **Control Panel**, and select the **Solid** style.
4. If the property **Image/Chart 1** is not in the properties sheet yet, add it from the drop-down **Properties** → **Paint** → **Custom Paint 1** → **Image/Chart 1**.



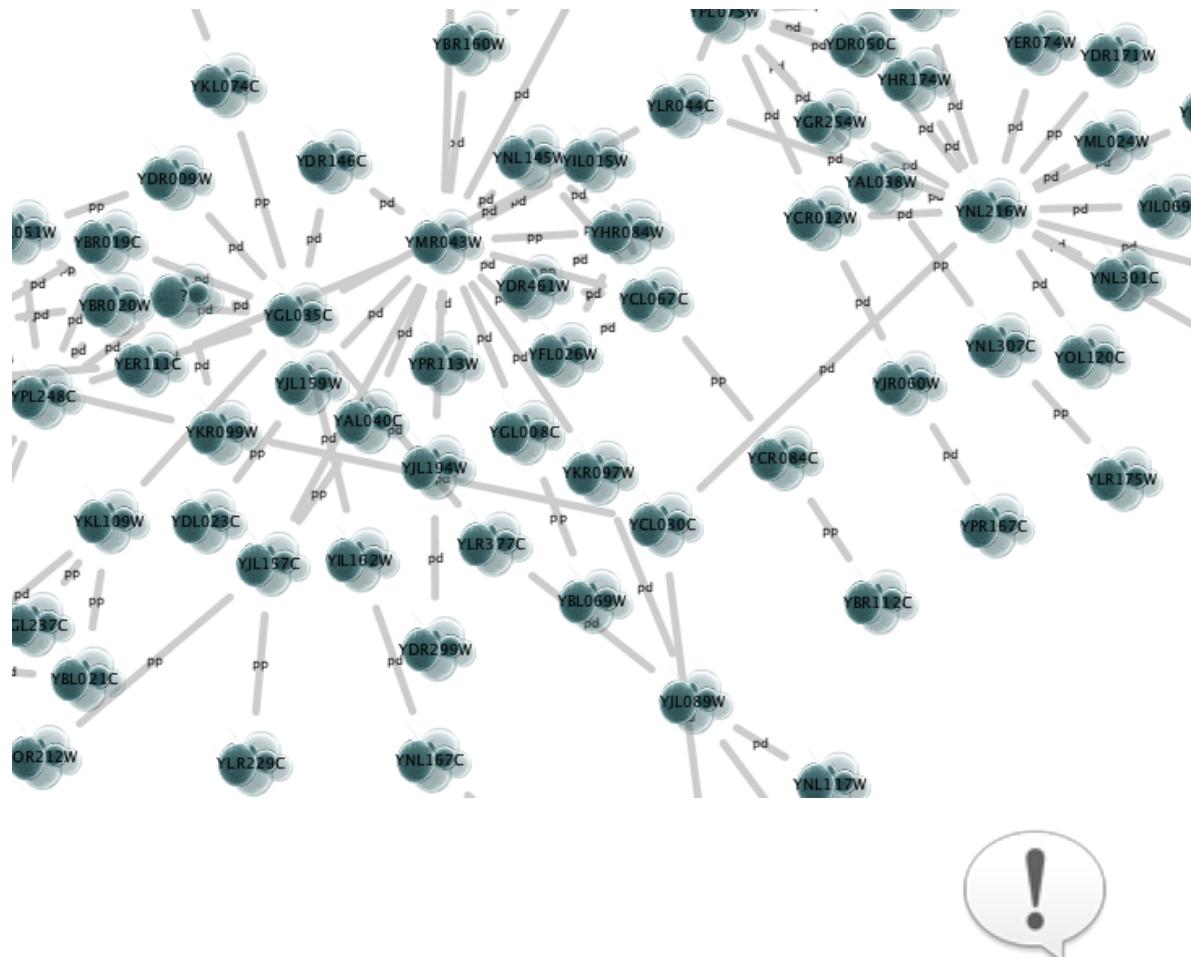
5. Click the **Default Value** cell of the **Image/Chart 1** entry in order to open the **Graphics** dialog.



6. Select any of the images from the list and click **Apply**.

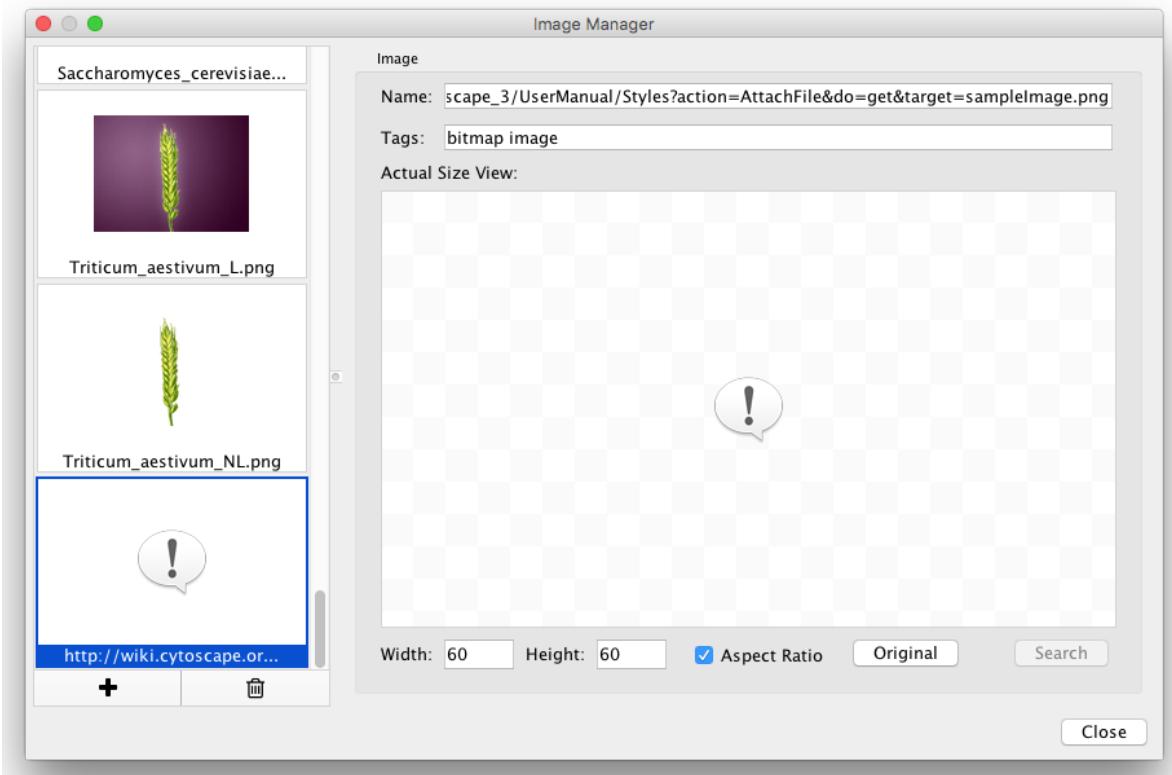


7. Click the **Default Value** cell of node **Transparency** and set the value to *zero*.
8. Set the **Default Value** of node **Size** to *80*.
9. Set the **Default Value** of node **Label Font Size** to *10*, to increase readability.
10. Also change the edge **Width** to *6*. Now your network looks like the following:

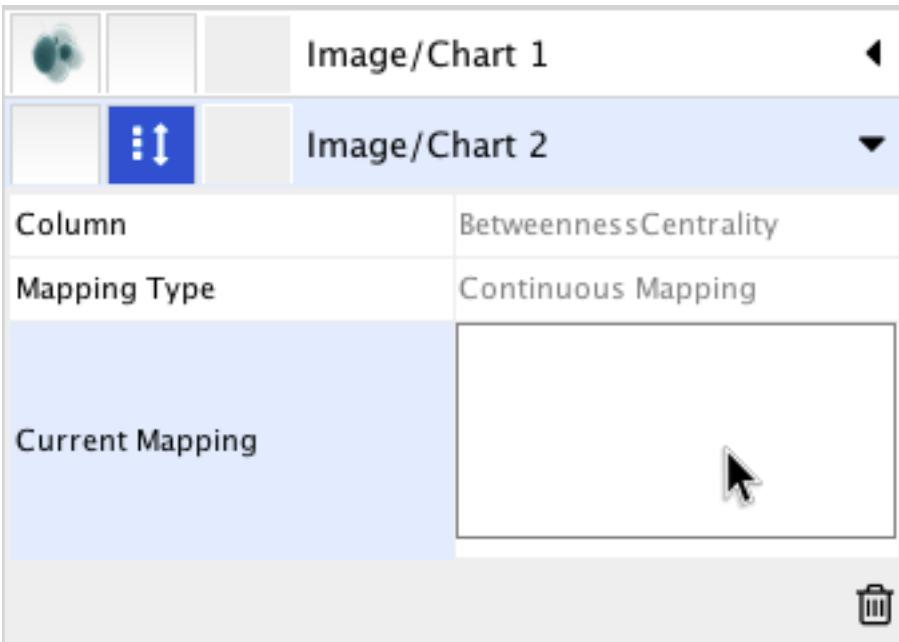


11. Open the Image Manager under **View → Open Image Manager...**. Drag and Drop this icon to

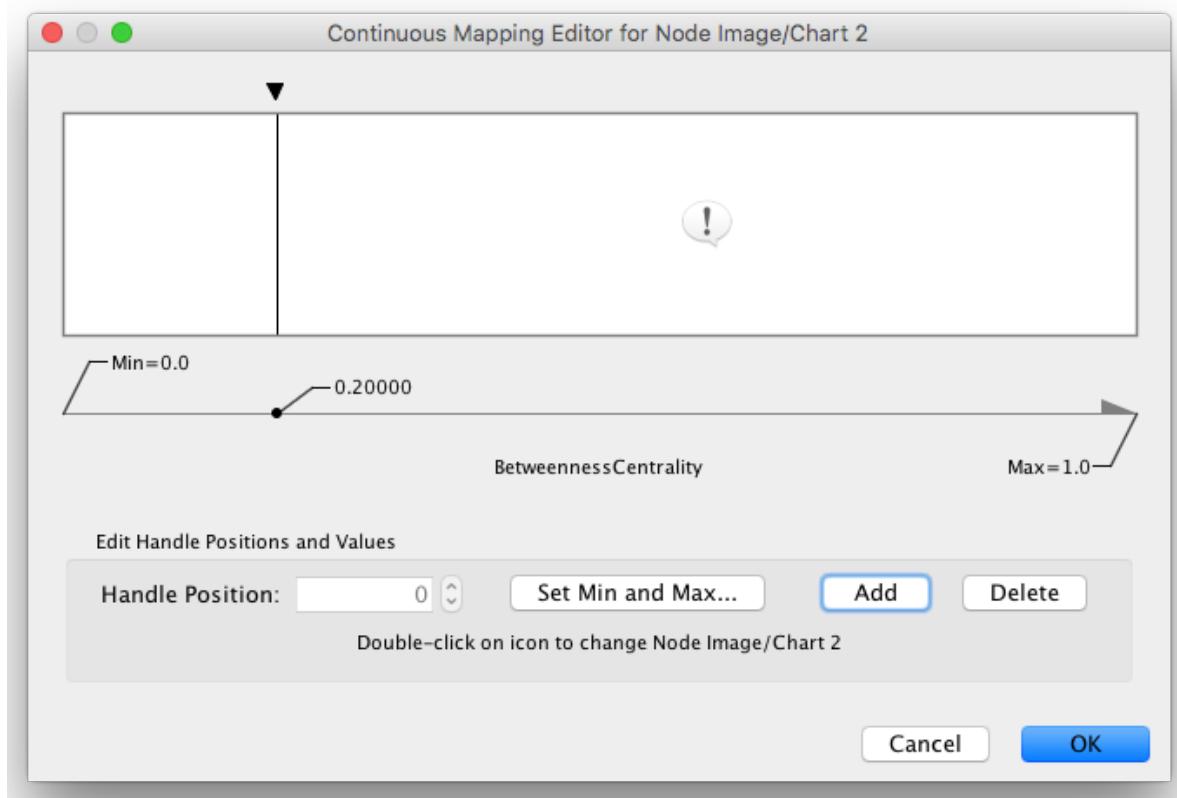
the image list which automatically adds it to the manager.



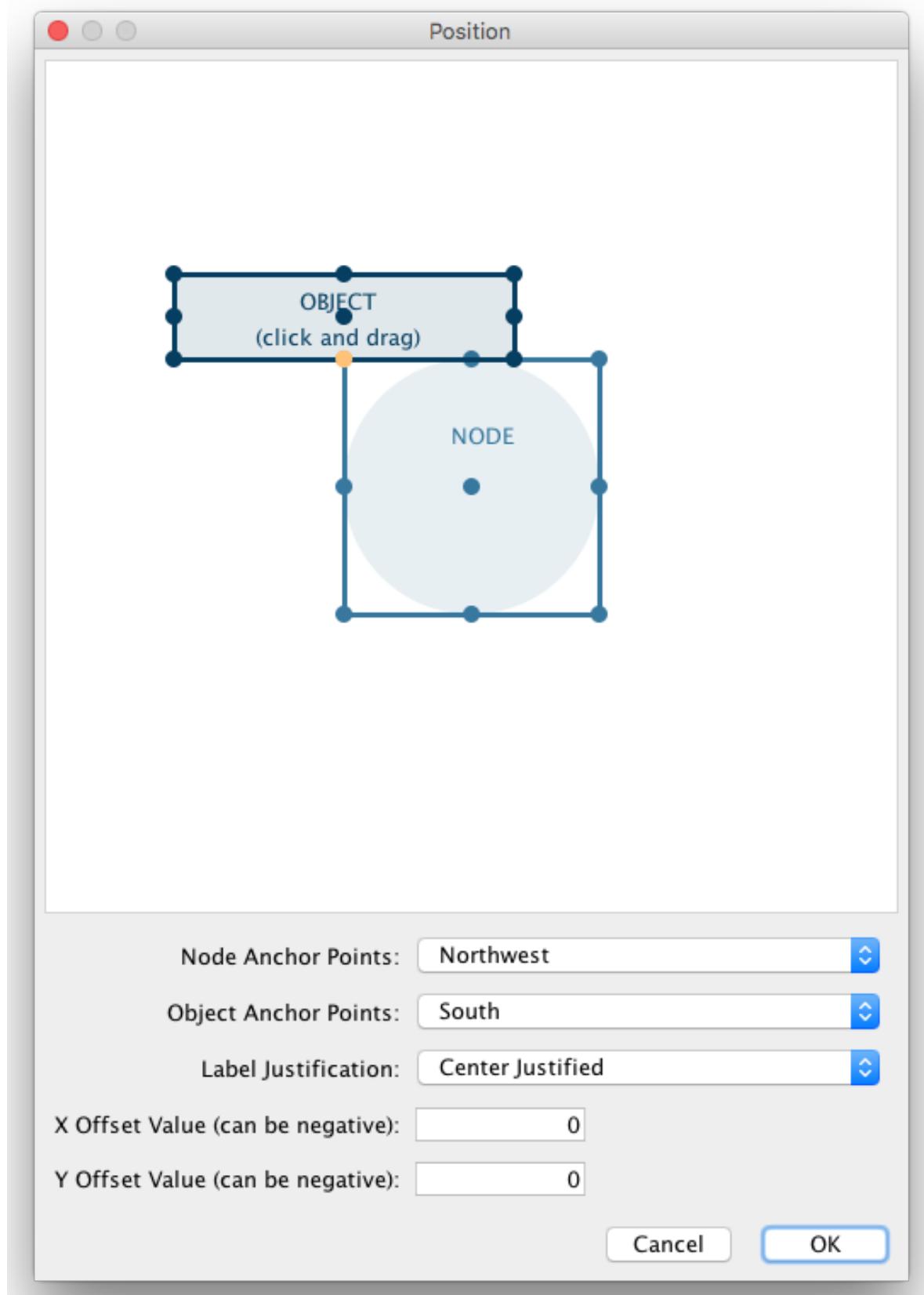
12. Create a Continuous Mapping for **Image/Chart 2** and select “BetweennessCentrality” as its controlling property. Double-click the **Current Mapping** value cell to open the Continuos Mapping Editor.



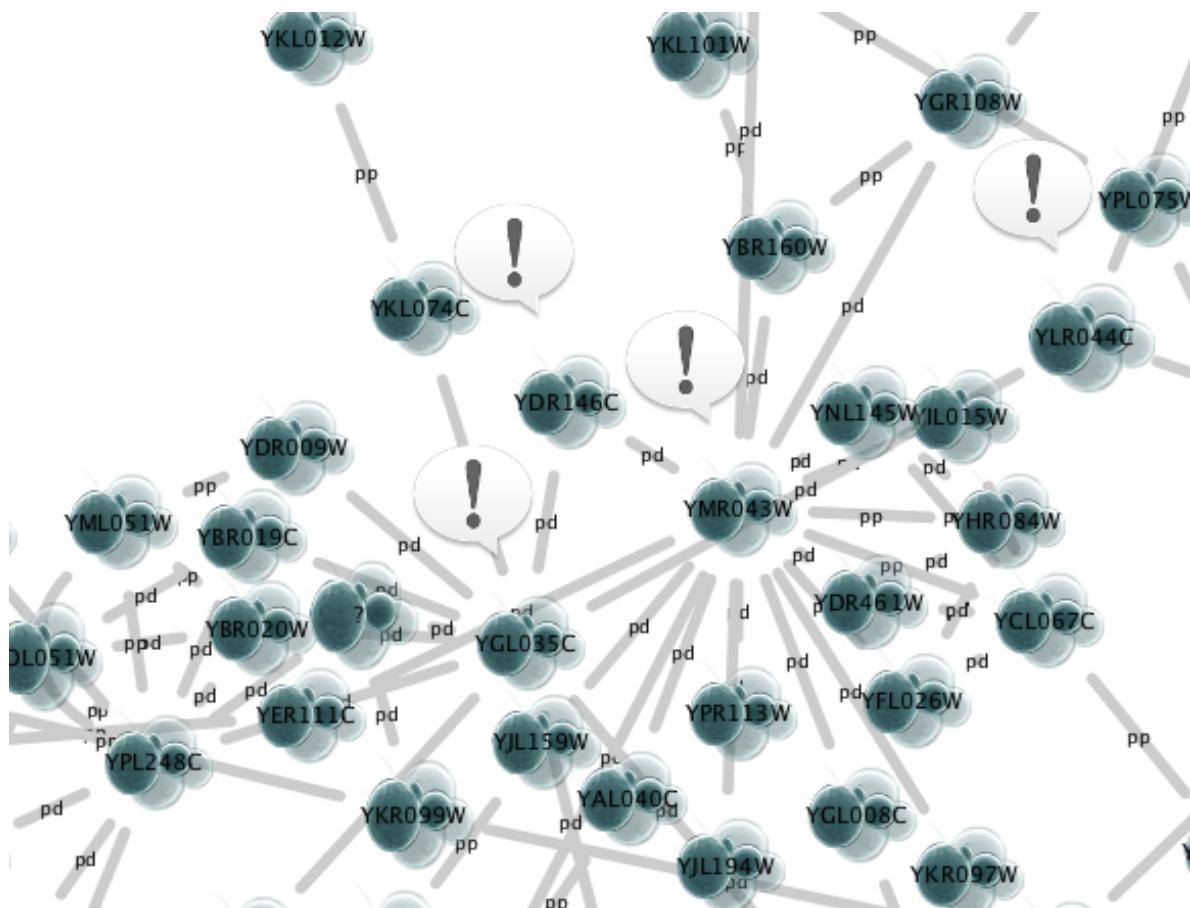
13. In the **Continuos Mapping Editor**, add a handle position by clicking in the **Add** button, and move the handle to **0.2**. Double-click the region over **0.2** and set the new icon you have just added in the last step.



14. Add the property **Image/Chart Position 2** from the drop-down option **Properties → Paint → Custom Paint 2 → Image/Chart Position 2**. Click its **Default Value** cell to move the position of the graphics to upper left.



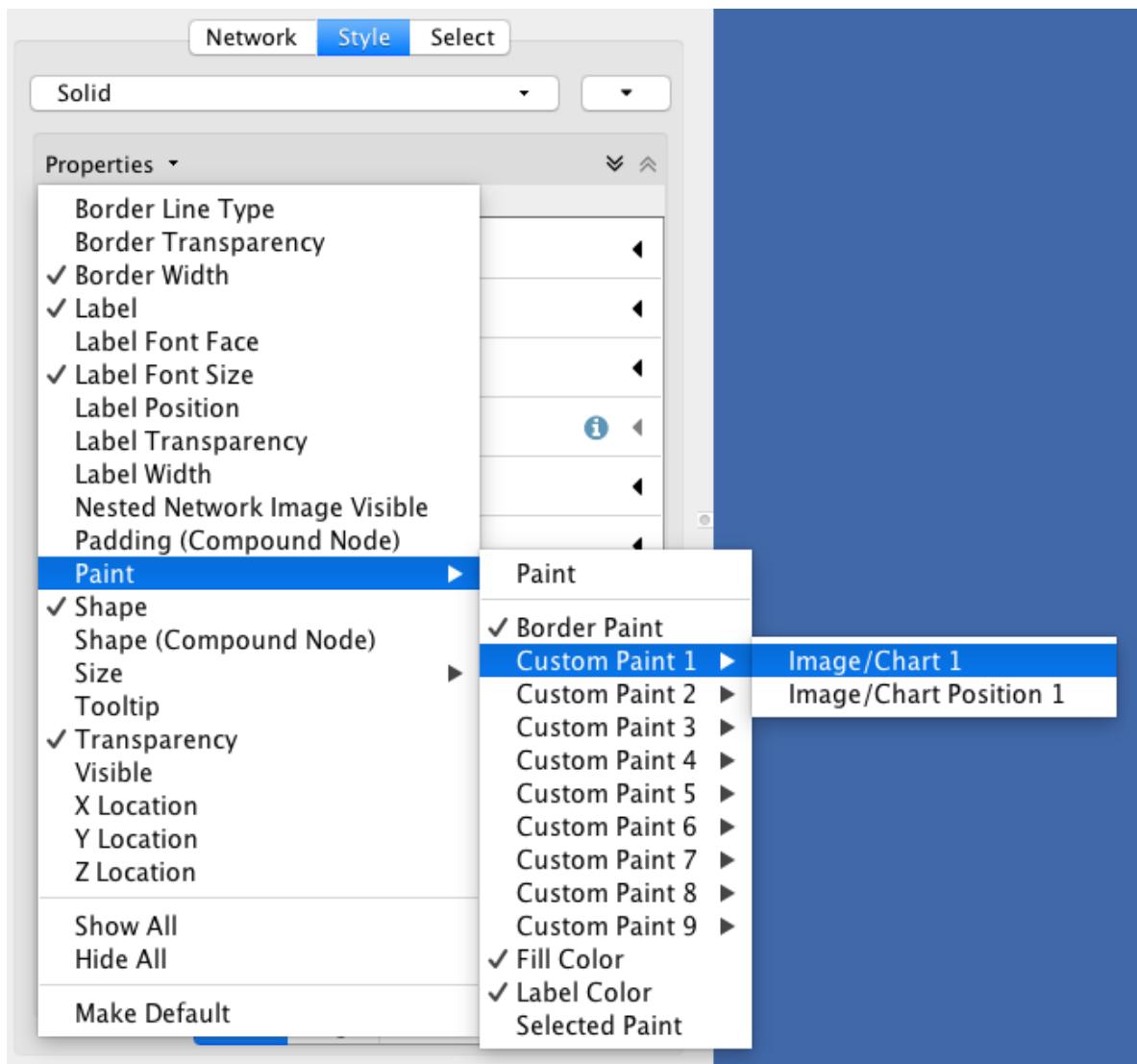
Now the important nodes in the network (nodes with high [betweenness centrality](#)) are annotated with the icon:



12.5.6 Tutorial 6: Creating Node Charts

The goal of this tutorial is to learn how to create and customize node charts from data stored in the Node tables.

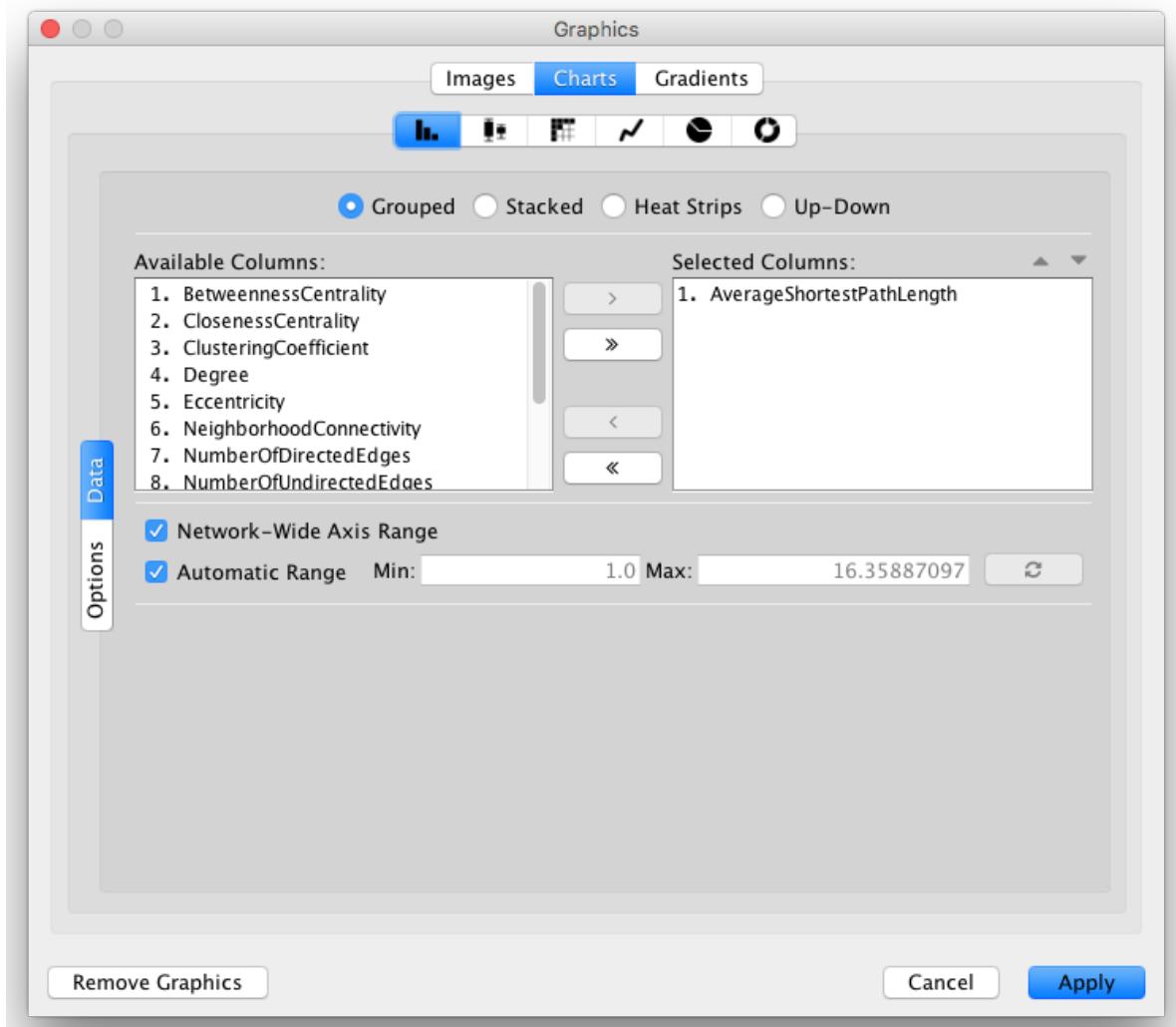
1. Start a new session and load a sample network. From the main menu, select **File** → **Import** → **Network** → **File...**, and select `sampleData/galFiltered.sif`.
2. Create some node/edge statistics using the **Network Analyzer**. Network Analyzer calculates some basic statistics for nodes and 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 regular table data.
3. Select the **Style** panel in the Control Panel.
4. **Create a new style:** Click the **Options** drop-down, and select **Create New Style**. Enter a name for your new style when prompted.
5. If the property **Image/Chart 1** is not in the properties sheet yet, add it from the drop-down **Properties** → **Paint** → **Custom Paint 1** → **Image/Chart 1**.



6. Click the **Default Value** cell of the **Image/Chart 1** entry in order to open the **Graphics** dialog.



7. Click the **Charts** tab and make sure the **Bar Chart** option is selected.

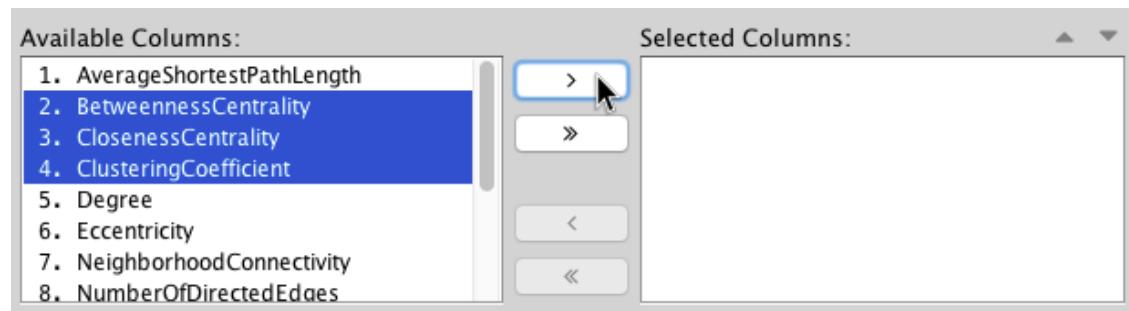


8. **Select data columns:** Now you have to choose the columns in the Node table that contains the data you want to be displayed as charts. The **Available Columns** list displays all node columns that can be used as chart data (i.e. *single* or *list* columns of numerical types).

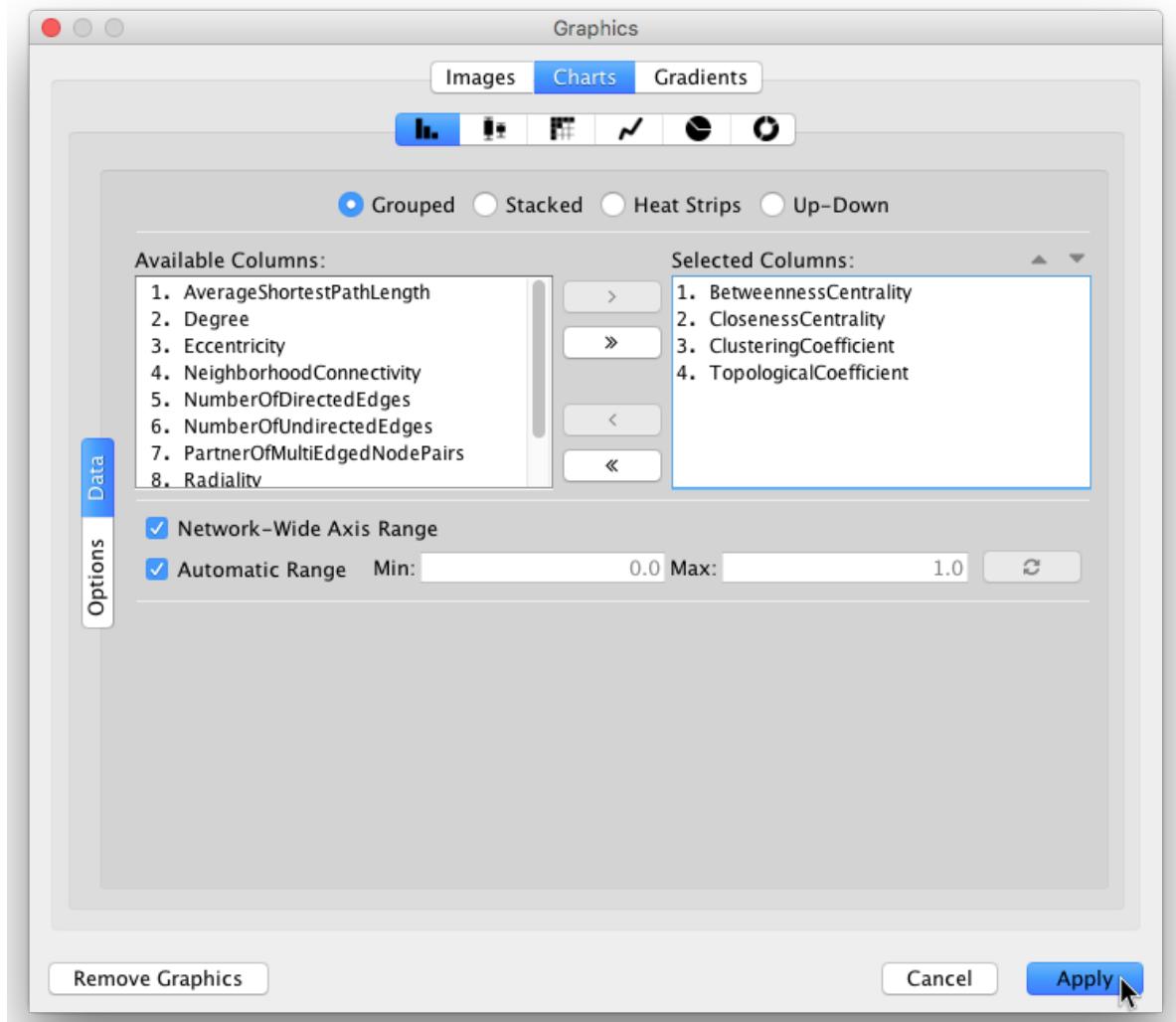
- First click the **Remove All** button to remove the current selected columns (by default, Cytoscape selects the first column in the **Available Columns** list).

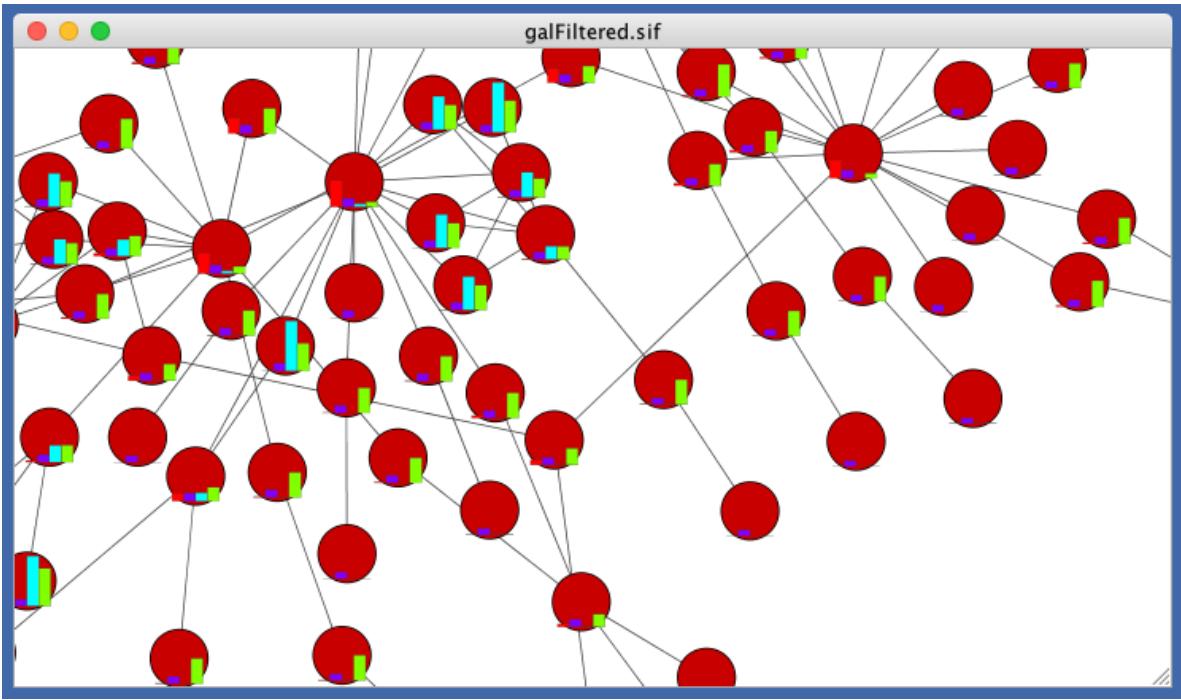


- Now select all *centrality* and *coefficient* columns from **Available Columns** list and click the **Add Selected** button.

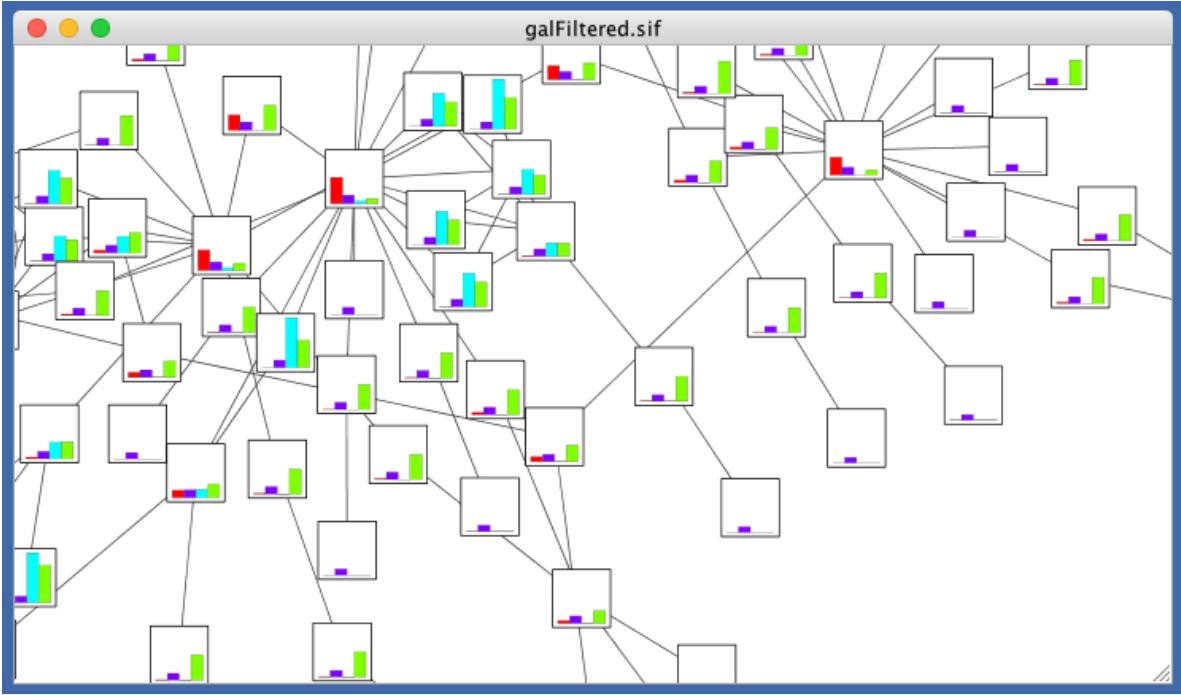


9. Click the **Apply** button to create bar charts with the selected data columns and default options.

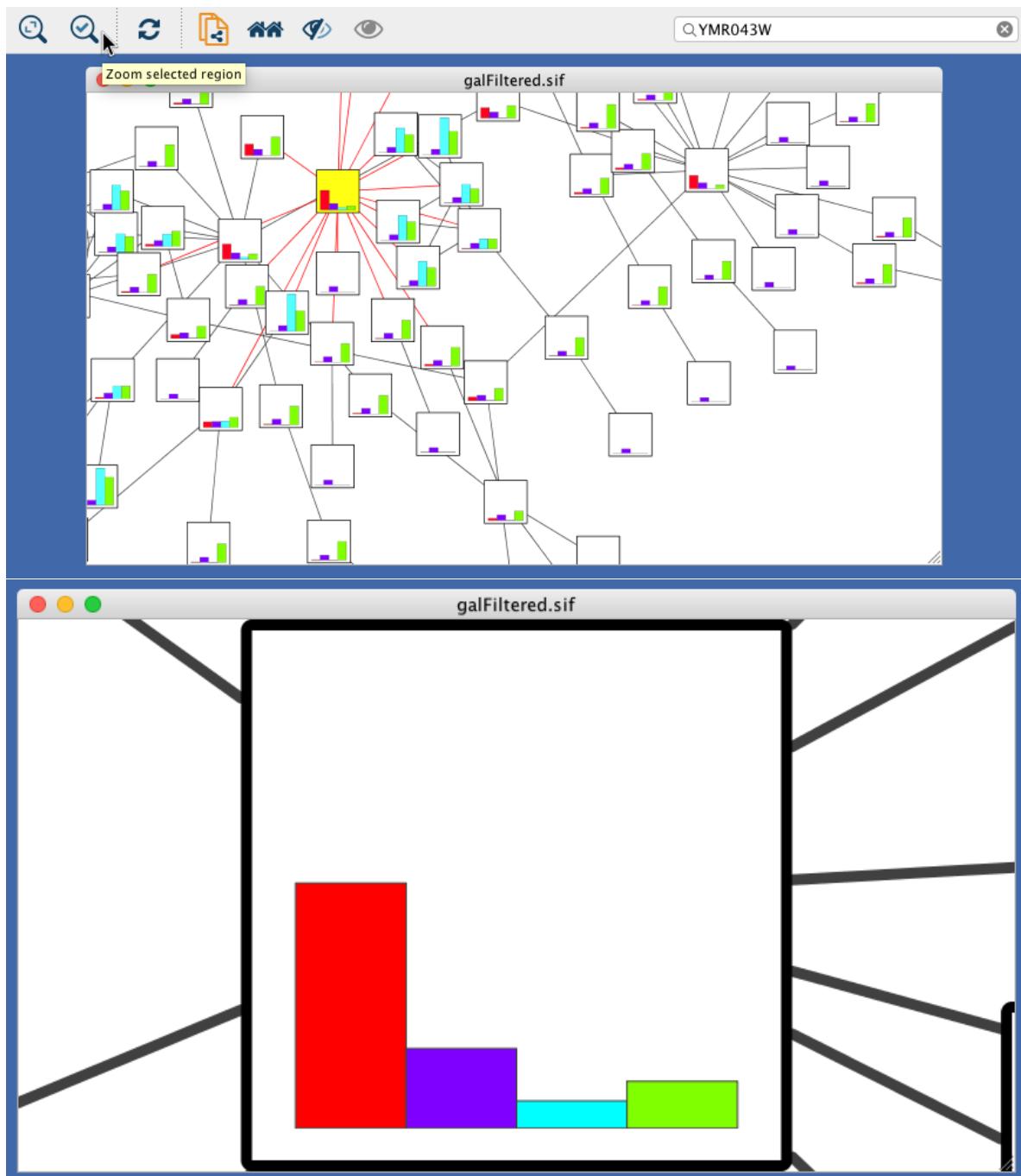




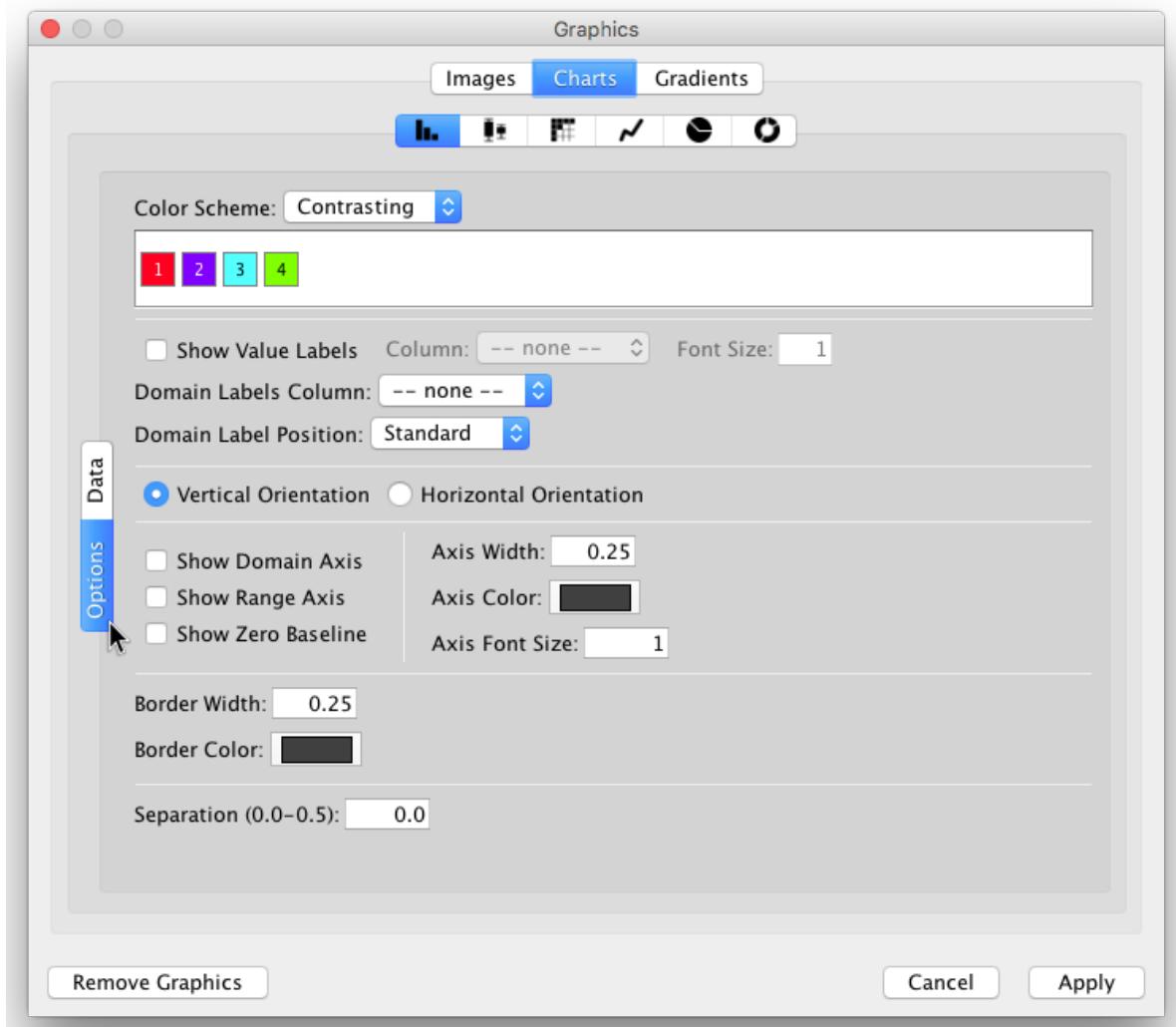
10. The network view doesn't look so good yet, so let's make a few changes to its Style before we continue. In the example shown below, the node **Shape** is set to *Rectangle*, while the node **Fill Color** is set to *white*.



11. Focus on one node to see the chart details. For example search for and then focus on node "YMR043W".



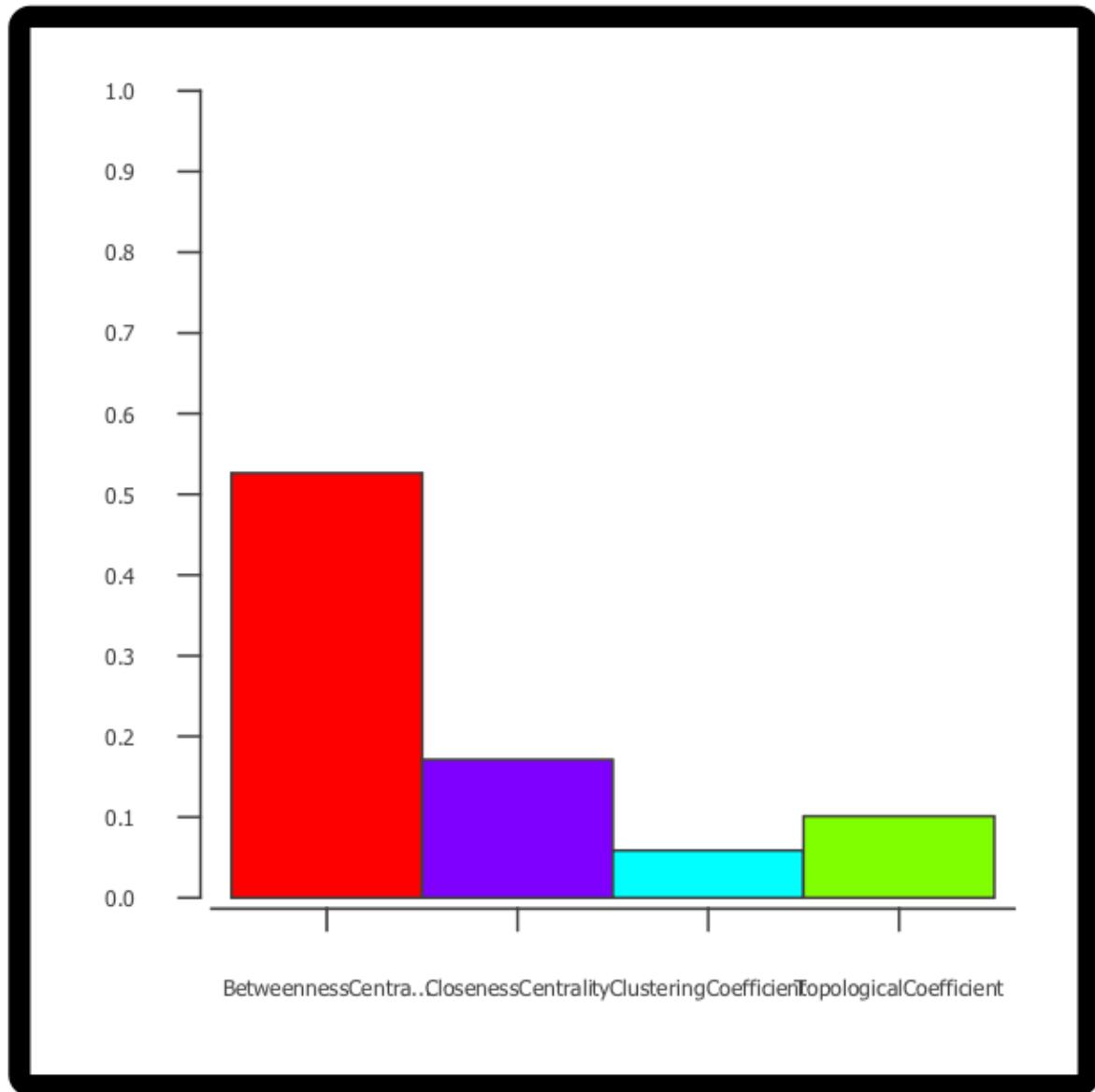
12. **Change other chart options:** Click the **Default Value** cell of the **Image/Chart 1** property again in order to open the **Graphics** dialog, and then select the **Options** tab on the **Bar Chart** editor.



On this panel, you can:

- Choose another **Color Scheme** or set all the colors individually (just click on each color).
- Show/Hide Value and Domain **Labels** and also set the **Domain Label Position**.
- Change the chart **Orientation**.
- Show/Hide Axes.
- Change the line width and color for axes and bars.
- Increase or reduce the separation between bars (up to 50% of the total chart width).
- **Note:** Other charts provide different options.

13. Check both **Show Domain Axis** and **Show Range Axis** and apply the graphics again. Now the node chart should look like this:



14. The default domain labels are not very useful, so let's set better labels:

- On the **Node Table** (Table Panel), create a new *List Column* of type *String* and name it “domain_labels”.
- Edit any of the cells of the created column (double-click it) and type `["Bet. Cent.", "Closen. Cent", "Clust. Coeff.", "Topol. Coeff."]`.
- Right-click the same cell and select the option **Apply to entire column**.

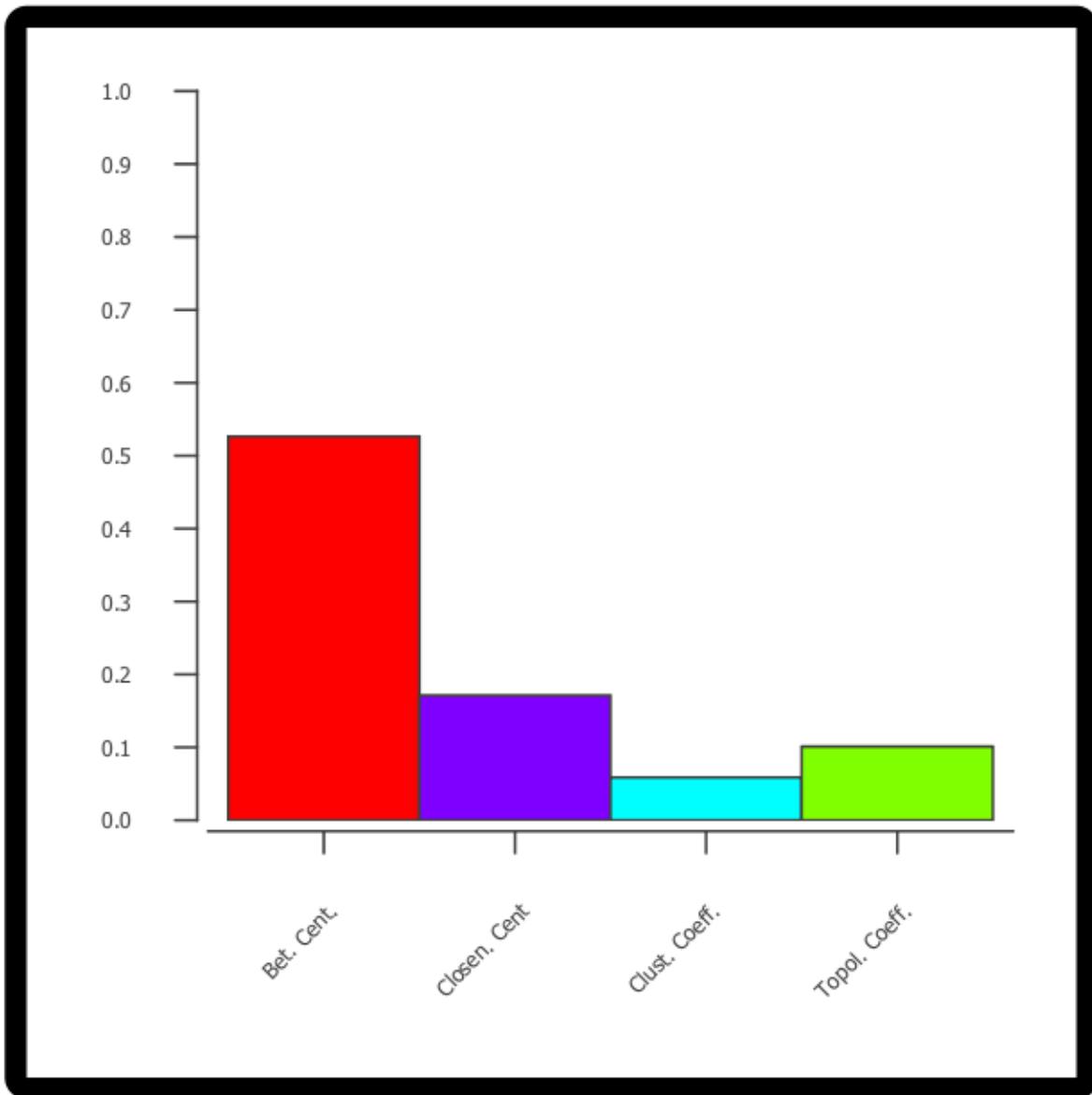
Table Panel

The screenshot shows the Cytoscape Table Panel with a Node Table selected. The table has the following structure:

y	NumberOfDirectedEdges	NumberOfUndirectedEdges	Radiality	TopologicalCoefficient	domain_labels
0	1	0	0.46774...	0.0	[Bet. Cent., Closen. Cent, Clust. Coeff., Topol. Coeff.]
7	3	0	0.50462...	0.33333333	[Bet. Cent., Closen. Cent, J ust. Coeff., Topol. Coeff.]
0	1	0	0.66666...	0.0	[Bet. Cent., Closen. Cent, Clust. Coeff., Topol. Coeff.]
0	3	0	1.0	0.0	[Bet. Cent., Closen. Cent, Clust. Coeff., Topol. Coeff.]
0	1	0	0.66666...	0.0	[Bet. Cent., Closen. Cent, Clust. Coeff., Topol. Coeff.]
0	2	0	0.83333...	1.0	[Bet. Cent., Closen. Cent, Clust. Coeff., Topol. Coeff.]
0	2	0	0.83333...	1.0	[Bet. Cent., Closen. Cent, Clust. Coeff., Topol. Coeff.]
0	1	0	0.66666...	0.0	[Bet. Cent.. Closen. Cent. Clust. Coeff.. Topol. Coeff.]

Below the table are three tabs: Node Table (selected), Edge Table, and Network Table.

- Open the chart editor again and select the **Options** panel.
- Select “domain_labels” on the **Domain Labels Column** drop-down button.
- Select “Up 45 $^{\circ}$ ” on the **Domain Labels Position** drop-down button. The labels should look like this now:



12.6 Advanced Topics

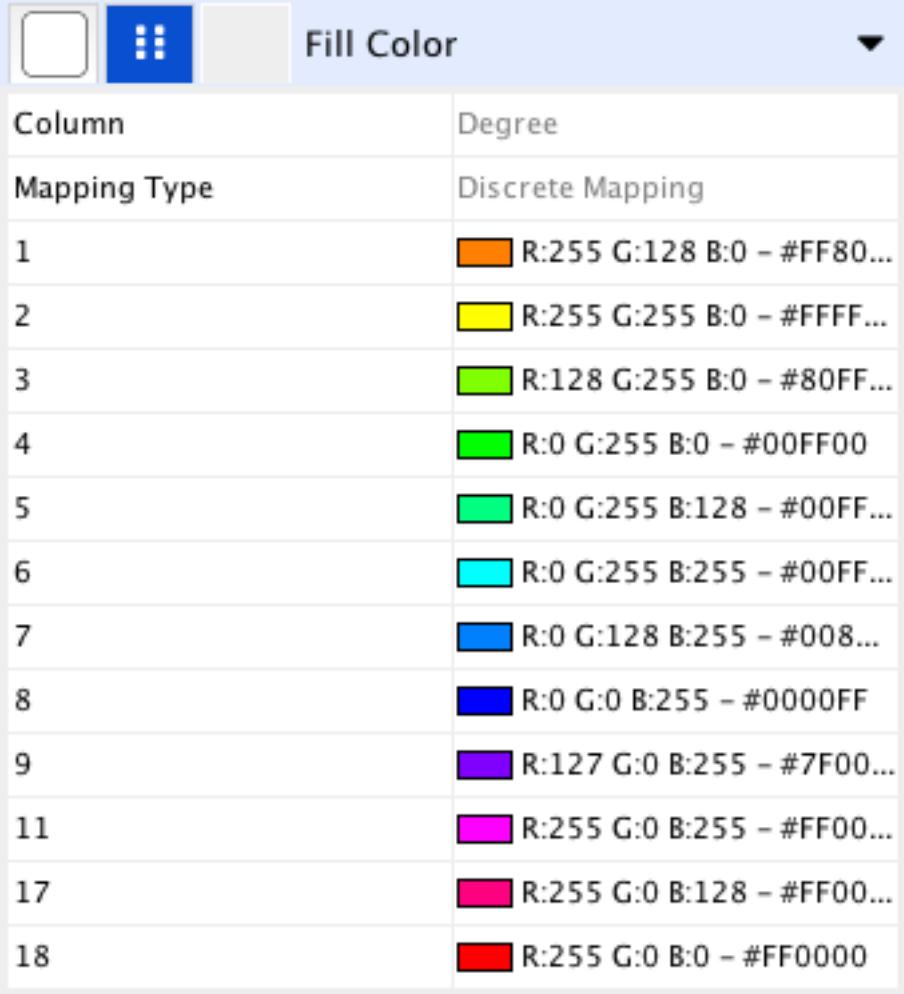
12.6.1 Discrete Mappings

Several utility functions are available for Discrete Mappings. You can use these functions by right-clicking on any property entry (shown below).



Automatic Value Generators

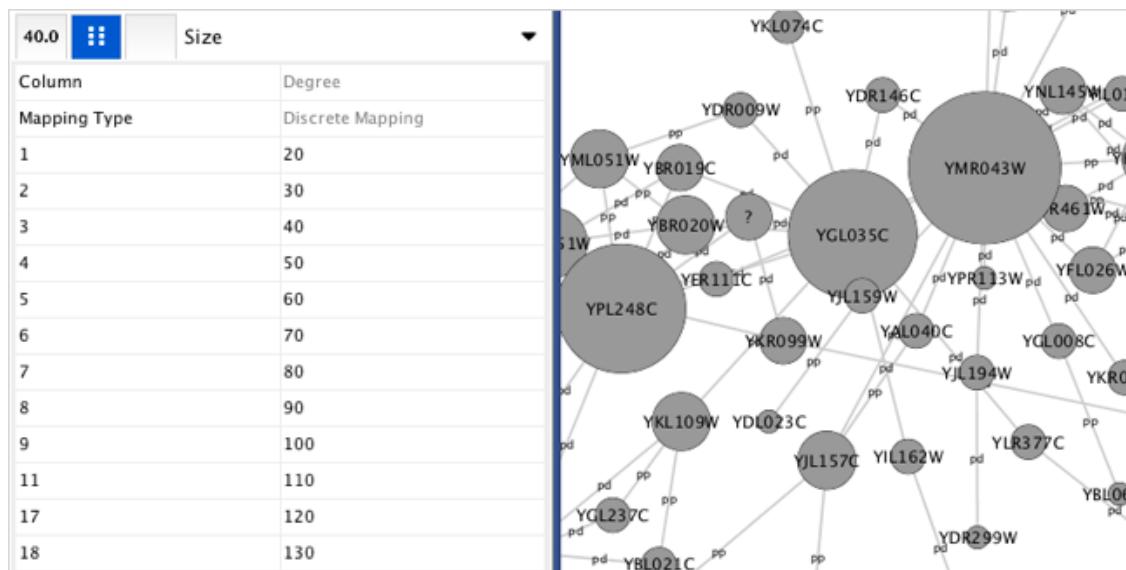
- **Mapping Value Generators** - Functions in this menu category are value generators for discrete mappings. Users can set values for discrete mappings automatically by selecting these functions.
 - **Rainbow** and **Rainbow OSC** - These functions try to assign as diverse a set of colors as possible for each data value.



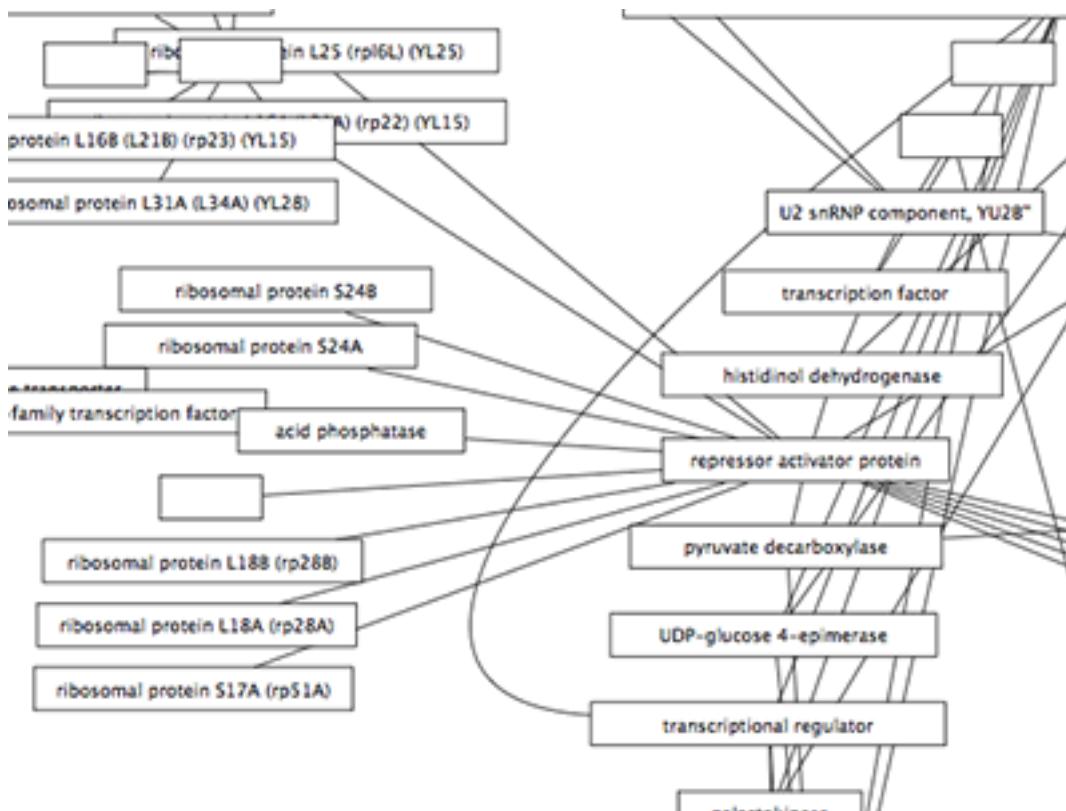
The screenshot shows the 'Fill Color' dialog in Cytoscape. At the top, there are two icons: a white square with a black outline and a blue square with a white grid pattern. To the right of these is the text 'Fill Color' and a dropdown arrow. Below this is a table with two columns: 'Column' and 'Degree'. The 'Degree' column lists values 1 through 18, each associated with a color swatch and its corresponding RGB hex code.

Column	Degree
Mapping Type	Discrete Mapping
1	R:255 G:128 B:0 – #FF8000
2	R:255 G:255 B:0 – #FFFF00
3	R:128 G:255 B:0 – #80FF00
4	R:0 G:255 B:0 – #00FF00
5	R:0 G:255 B:128 – #00FF80
6	R:0 G:255 B:255 – #00FFFF
7	R:0 G:128 B:255 – #0080FF
8	R:0 G:0 B:255 – #0000FF
9	R:127 G:0 B:255 – #7F00FF
11	R:255 G:0 B:255 – #FF00FF
17	R:255 G:0 B:128 – #FF0080
18	R:255 G:0 B:0 – #FF0000

- **Random Numbers** and **Random Colors** - Randomized numbers and colors.
- **Number Series** - Set a series of numbers to the specified mapping. Requires a starting number and increment.

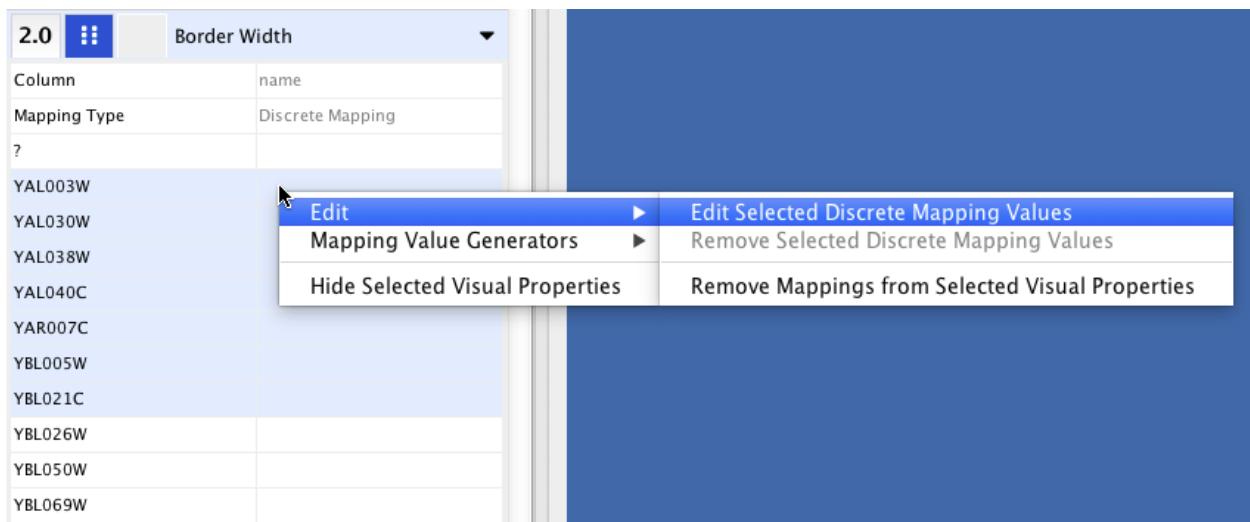


- **Fit label width** - This function is only for node **Width** and **Size**. When a discrete mapping for node **Width** or **Size** is available, you can fit the size of each node to its label automatically by selecting this function. See the example below:



Edit Selected Values at Once

You can set multiple values at once. First, you need to select discrete mapping rows in which you want to change values then right-click and select **Edit → Edit Selected Discrete Mapping Values**. A dialog pops up and you can enter the new value for the selected rows.



12.6.2 Working with Continuous Mapping Editors

There are three kinds of **Continuous Mapping Editors**. Each of them are associated with a specific property type:

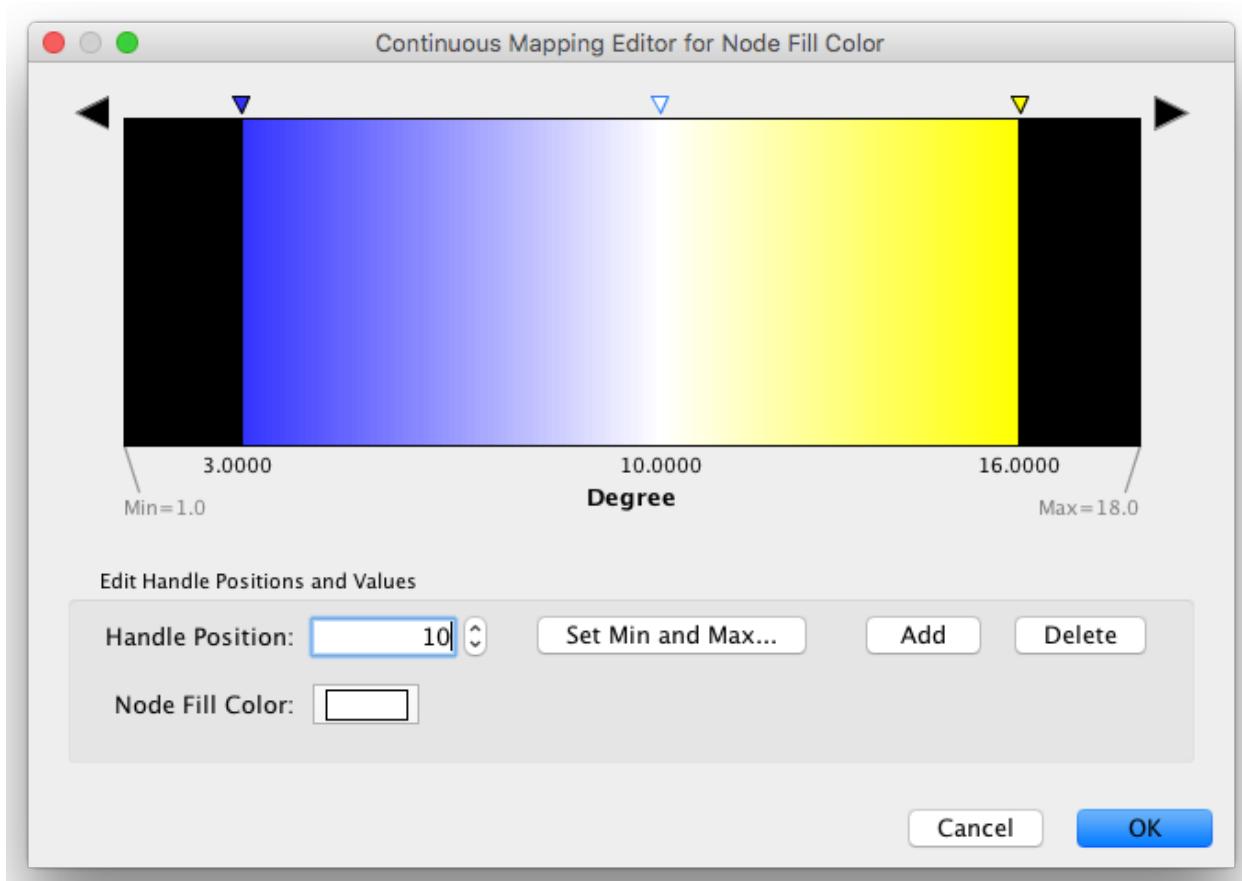
Range Settings Panel



Each continuous mapping editor has a range settings section (labelled **Edit Handle Positions and Values**) with the following fields and buttons.

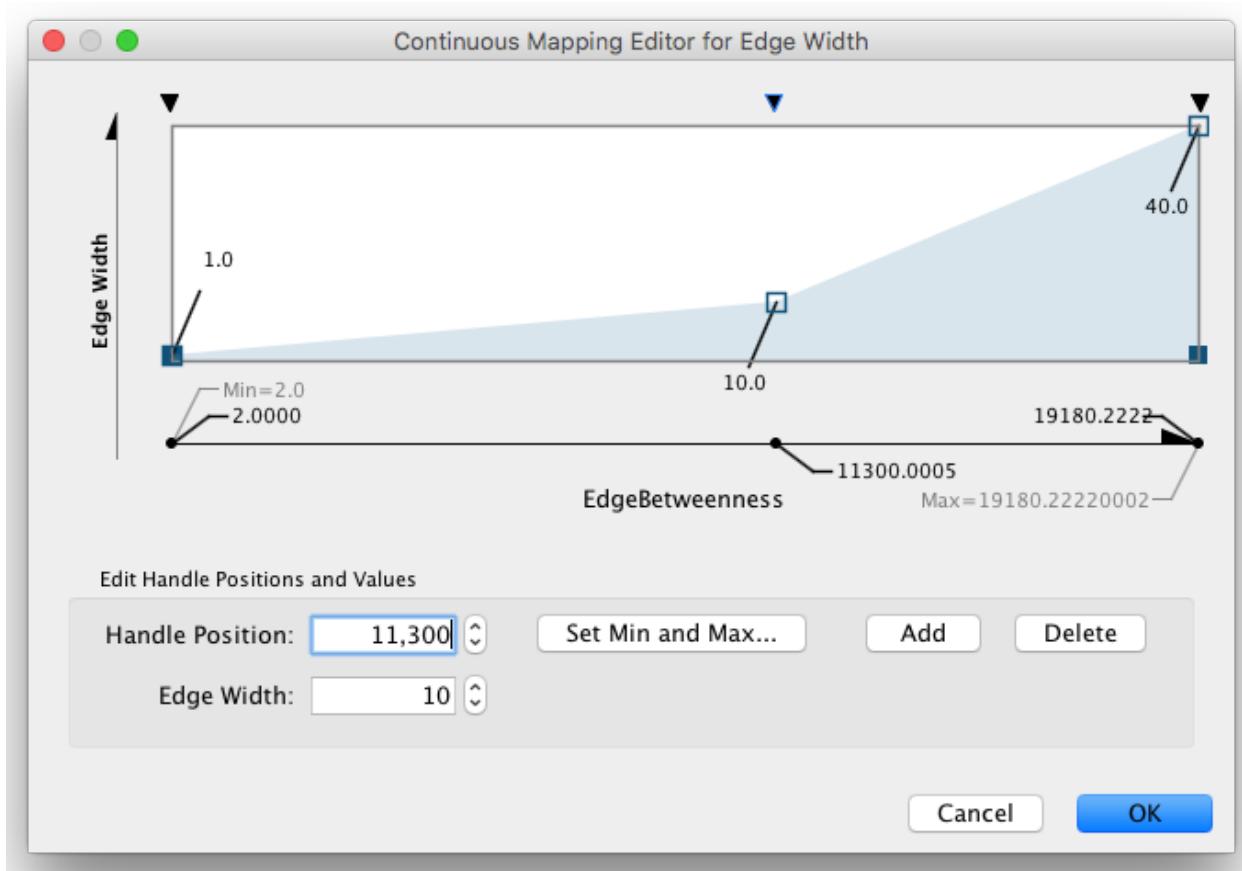
1. **Handle Position** - This box displays the current value for the selected slider handle. You can also directly type the value in this box to move the slider to an exact location.
2. **Set Min and Max...** - Click this button to set the overall range of this editor. The first time you open the editor, the *Min* and *Max* values are set by the range of the data column you selected (i.e. the minimum and maximum values of the mapped column). In the pop-up dialog you can manually enter the Min and Max values, or they can be set to the current minimum and maximum values of the data column by clicking the *Reset* button.
3. **Add** - Adds a new handle to the editor.
4. **Delete** - Deletes the selected handle from the slider widget.
5. **Handle Value (e.g. Node Fill Color)** - Click this button to edit the value (e.g. a color) assigned to the selected handle.

Gradient Editor



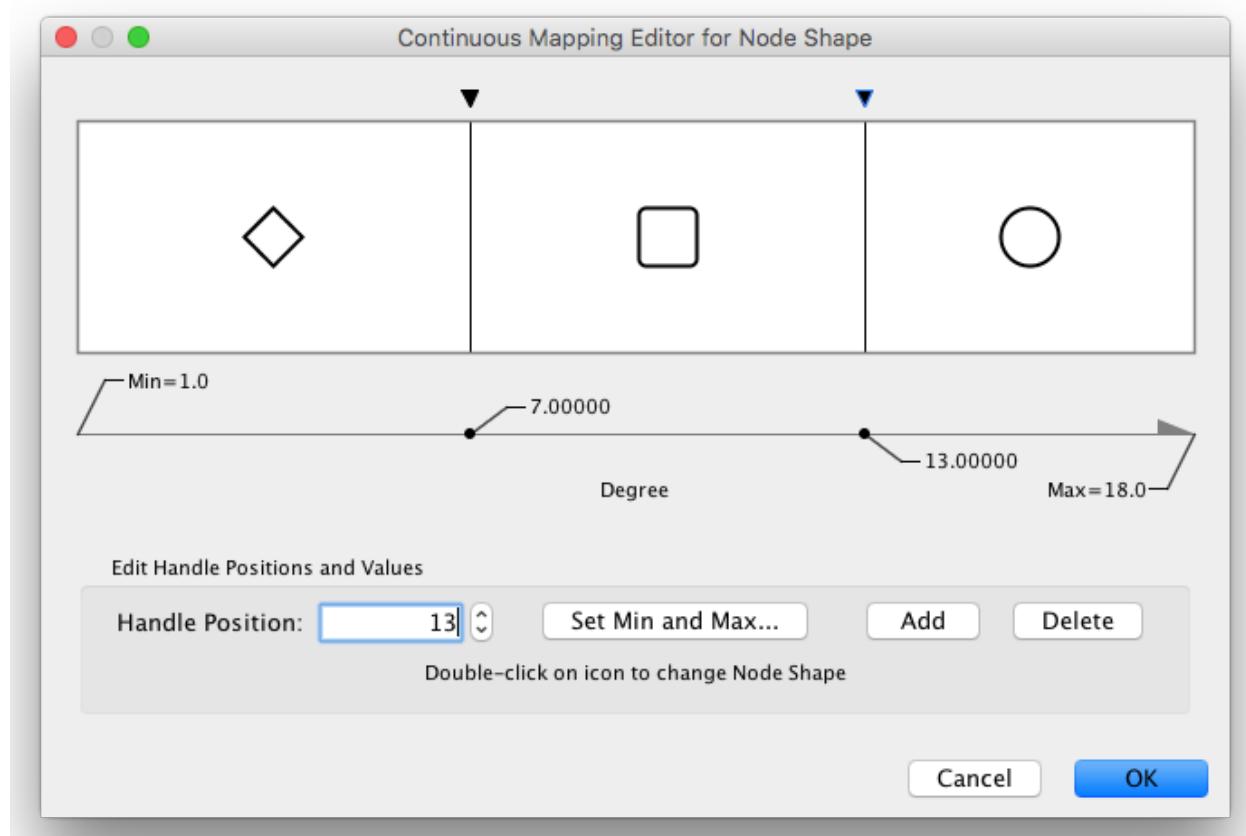
The Gradient Editor is an editor for creating continuous mappings for colors. To change the color of each region, just double-click the handles (small triangles on the top). A Color gradient will be created only when the editor has two or more handles (see the example below).

Continuous-Continuous Editor



The Continuous-Continuous Editor is for creating mappings between numerical data and numerical properties (e.g. size, transparency). To change the value assigned on the Y-axis (the property shown in the example above is edge **Width**), drag the small squares or double-click them to directly type an exact value.

Continuous-Discrete Editor



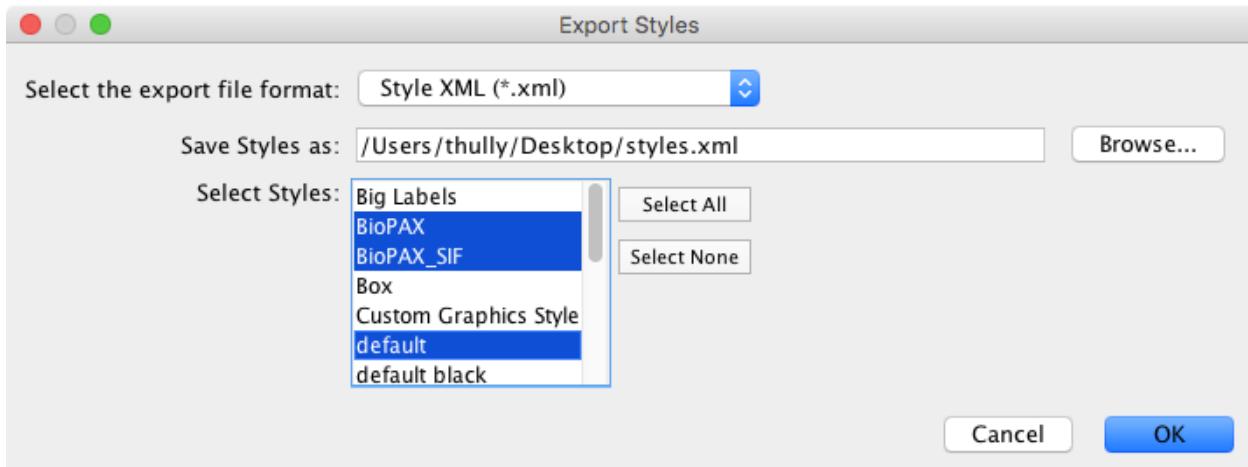
The Continuous-Discrete Editor is used to create mappings from numerical column values to discrete properties, such as fonts, shapes, or line styles. To edit a value for a specific region, double-click the icon on the track.

12.7 Managing Styles

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

12.7.1 Saving Styles

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 styles in a file separate from the session file. To do this, navigate to the menu option **File** → **Export** → **Styles...**, and save the selected styles to a file. This feature can be used to share styles with other users.



You can also change the default list of styles for all future sessions of Cytoscape. To do this, click the **Options**

 drop-down in the **Style** section, and select **Make Current Styles Default**. This will save the current styles as a `default_vizmap.xml` file to your `CytoscapeConfiguration` directory (found in your home directory). These styles will then be loaded each time Cytoscape is started.

Style File Formats

The Cytoscape-native Style format is *Style XML*. If you want to share Style files with other Cytoscape users, you need to export them to this format.

From version 3.1.0 on, Cytoscape can also export [Cytoscape.js](#) compatible JSON file. Since Cytoscape.js is an independent JavaScript library, and there are some differences between Cytoscape and Cytoscape.js, not all properties are mapped to JSON. The following properties are not supported by the exporter:

- Custom Graphics and their locations
- Edge Bends
- Nested Networks
- Network Background (Note: This can be set manually as standard CSS in Cytoscape.js)

The Continuous-Discrete Editor is used to create mappings from numerical data values to discrete properties, such as fonts, shapes, or line styles. To edit a value for a specific region, double-click on the icon on the track.

12.7.2 Importing Styles

To import existing styles, navigate to the menu option **File → Import → Styles...** and select a `styles.xml` (Cytoscape 3 format) file. Imported properties will supplement existing properties or override existing ones if the properties have the same name. You can also specify a style file using the `-V` command line option. Properties loaded from the command line will override any default properties. Note that legacy `.props` files can only be loaded via the **File → Import → Styles...** menu, but not by command line.

CHAPTER 13

App Manager

13.1 What are Apps?

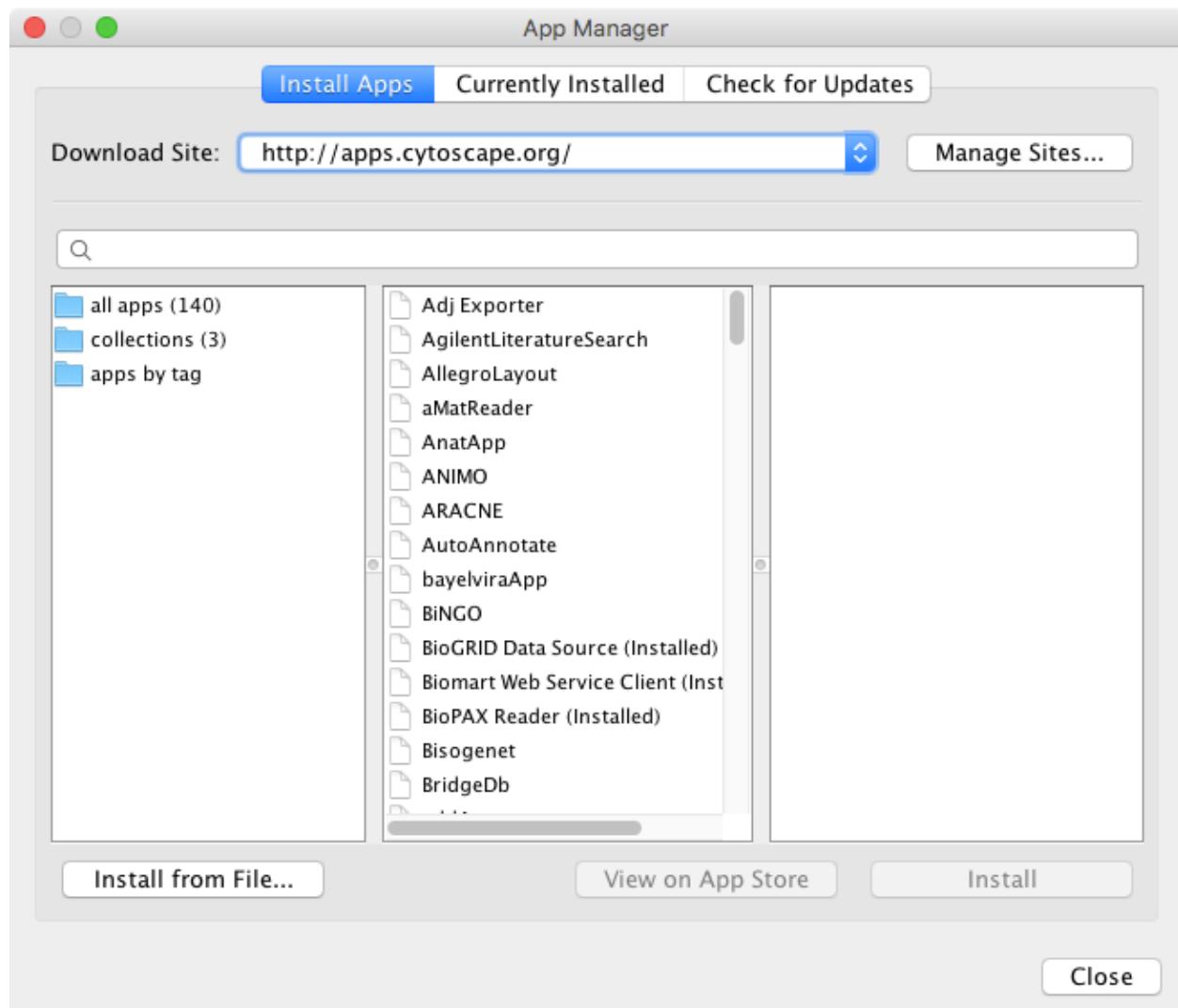
Cytoscape's capabilities are not fixed. They can be expanded with **apps**. They can extend Cytoscape in a variety of ways. One app can have the ability to import data from an online database. Another app could provide a new method for analyzing networks. You can install apps after you have installed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

13.2 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 Apps** tab selected.



There are four 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're not sure what sort of app you need and want to see everything, click the **all apps** folder. In the middle pane, you will see a list of all the apps.
- If you want to install a collection of apps for a specific use case, click on the **collections** folder. This will display the available collections in the middle pane. A collection is simply an app that installs other apps for a specific use case.
- 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 pane.

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

13.3 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 Apps** tab or find it in the App Store site and reinstall it from 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**.

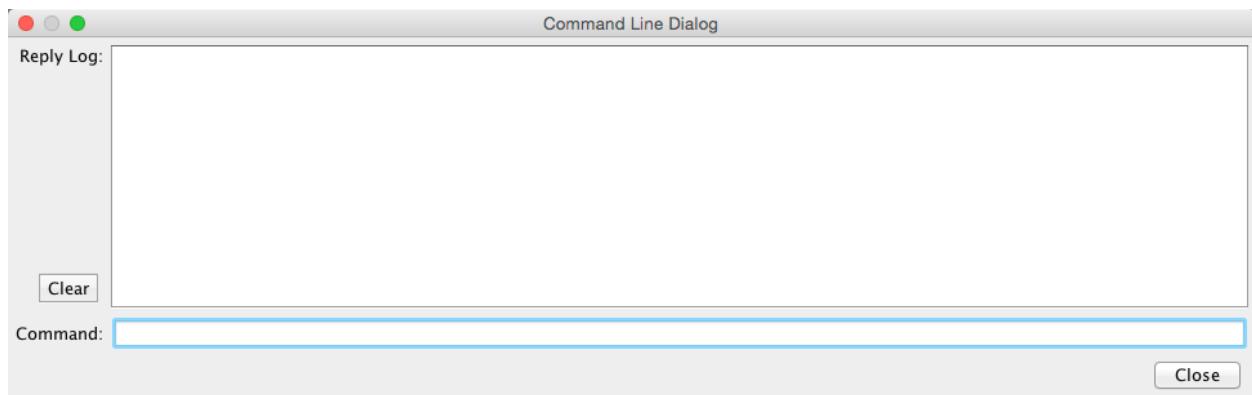
Note that uninstalling or disabling a collection will not uninstall or disable any apps installed by the collection.

CHAPTER 14

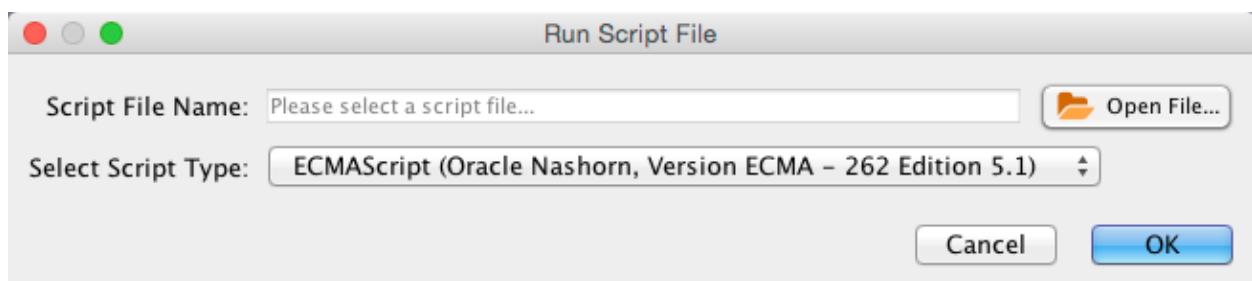
Command Tool

The **Command Line Tool** provides a simple command-line interface to Cytoscape using the Commands API. Any app that registers commands will be available through the Command Tool. Commands are part of the more general [Cytoscape Automation](#) feature, which includes multiple ways of scripting Cytoscape execution.

The Command Tool provides two main functions: first, a Command Line Dialog is available from **Tools → Command Line Dialog**, that allows the user to type commands into Cytoscape and see the results in a “Reply Log”.



Second, and arguably more useful, it will read script files and execute them. Each line in the script file is a command that is sent to a app. Script files may be entered on the Cytoscape command line using the “-S” flag to Cytoscape, through the **File → Run Script File...** menu item, or through **Tools → Execute Command File**.



Cytoscape commands consist of three parts: a command class, or namespace; a command within that namespace; and a series of arguments or options provided as a series of **name=value** pairs. For example, to import an XGMML format file from the **Command Line Dialog** or a command script, you would use:

```
network import file filePath="path-to-file"
```

where *network* is the namespace, *import file* is the command, and there is only one argument: *filePath="path-to-file"*. If there were more arguments they would appear on the same line separated by spaces.

The Command Tool also uses the Command API to provide help. “help” by itself will list all of the command classes (or namespaces) and “help “ followed by a namespace will list all of the commands supported by that namespace. Details of a specific command are available by typing “help “ followed by the namespace and command (e.g. “help layout force-directed”). The Command Tool registers the “command” namespace and supports a single command: run, which takes a file argument. Here is the help for the command run command from the command namespace:

```
help command run
    command run file=<File>
```

Similarly, the help for the “network import file” example from above is:

```
help network import file
network import file arguments:
dataTypeList=<String>: List of column data types ordered by column index
(e.g. "string,int,long,double,boolean,intlist" or just "s,i,l,d,b,il")
defaultInteraction=<String>: Default interaction type
delimiters=<ListMultipleSelection [,,;, ,\t]>: Text Delimiters
delimitersForDataList=<ListSingleSelection (\||\|/, )>:
    Text Delimiters for data list type
file=<File>: Data Table file
firstRowAsColumnNames=true|false: First row used for column names
indexColumnSourceInteraction=<int>: Column for source interaction
indexColumnTargetInteraction=<int>: Column for target interaction
indexColumnTypeInteraction=<int>: Column for interaction type
NetworkViewRendererList=<ListSingleSelection ()>: Network View Renderer
RootNetworkList=<ListSingleSelection ( -- Create new network collection
    --|Network)>: Network Collection
startLoadRow=<int>: Start Load Row
TargetColumnList=<ListSingleSelection ()>: Node Identifier Mapping Column
```

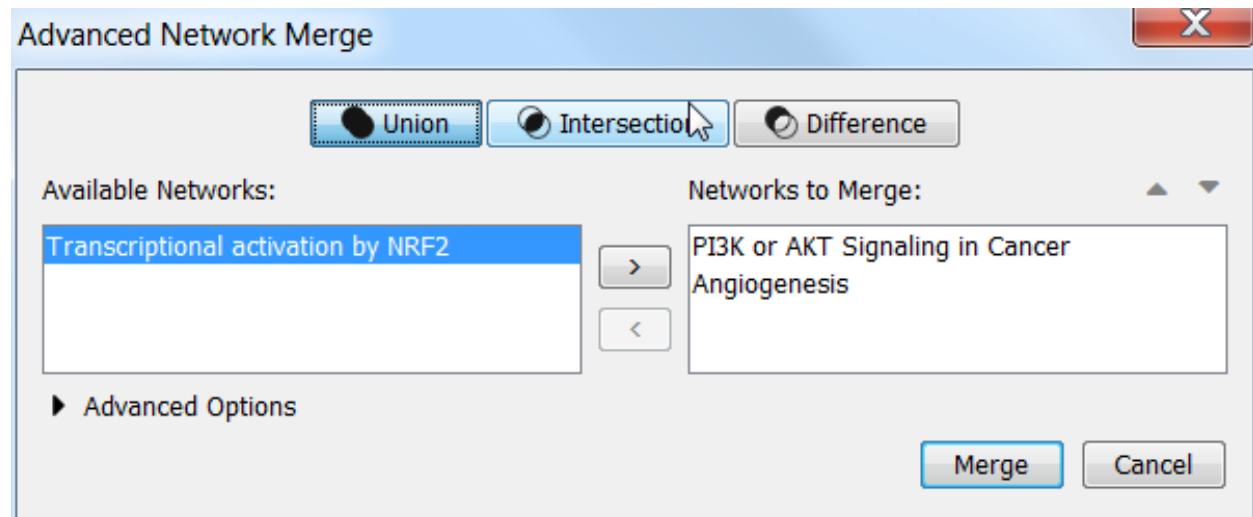
CHAPTER 15

Merge

Cytoscape allows for merging of both network and table data, through **Tools** → **Merge**.

15.1 Merge Networks

The **Advanced Network Merge** interface is available from **Tools** → **Merge** → **Networks...** and allows for merging of two or more networks.



15.1.1 Basic Operations

- With the buttons select either “union”, “intersection” or “difference”.
- Networks available for merge are listed under **Networks to merge**. Select a network from the list and click the right arrow to transfer the network to **Selected networks**. Click **Merge** to continue. The merged network will

be displayed as a separate network.

15.1.2 Advanced Options

The **Advanced Network Merge** interface includes an expandable **Advanced Network Merge** panel, where you can specify the details of how to merge the networks. The options available here are:

- **Matching columns:** This specifies the network columns that should be used for merging. Typically, the “name” column or some other column containing identifier information is used here.
- **How to merge columns?:** A table lets the user specify for each of the individual network columns, what the corresponding column in the merged network should be named and its data type.

▼ Advanced Options

Matching Columns (table columns to match nodes between networks):

Angiogenesis	PI3K or AKT Signaling in Cancer
name	name

How to merge columns:

Angiogenesis	PI3K or AKT Signali...	Merged Network	Column type
name	name	Matching.Attribute	List<String>
Datasource	Datasource	Datasource	String
GeneID	GeneID	GeneID	String
GraphID	GraphID	GraphID	String
WP.type	WP.type	WP.type	String
name	name	name	String
shared name	shared name	shared name	String

Node Edge

Enable merging nodes/edges in the same network

15.2 Merge Tables

CHAPTER 16

NetworkAnalyzer

NetworkAnalyzer computes a comprehensive set of topological parameters for undirected and directed networks, including:

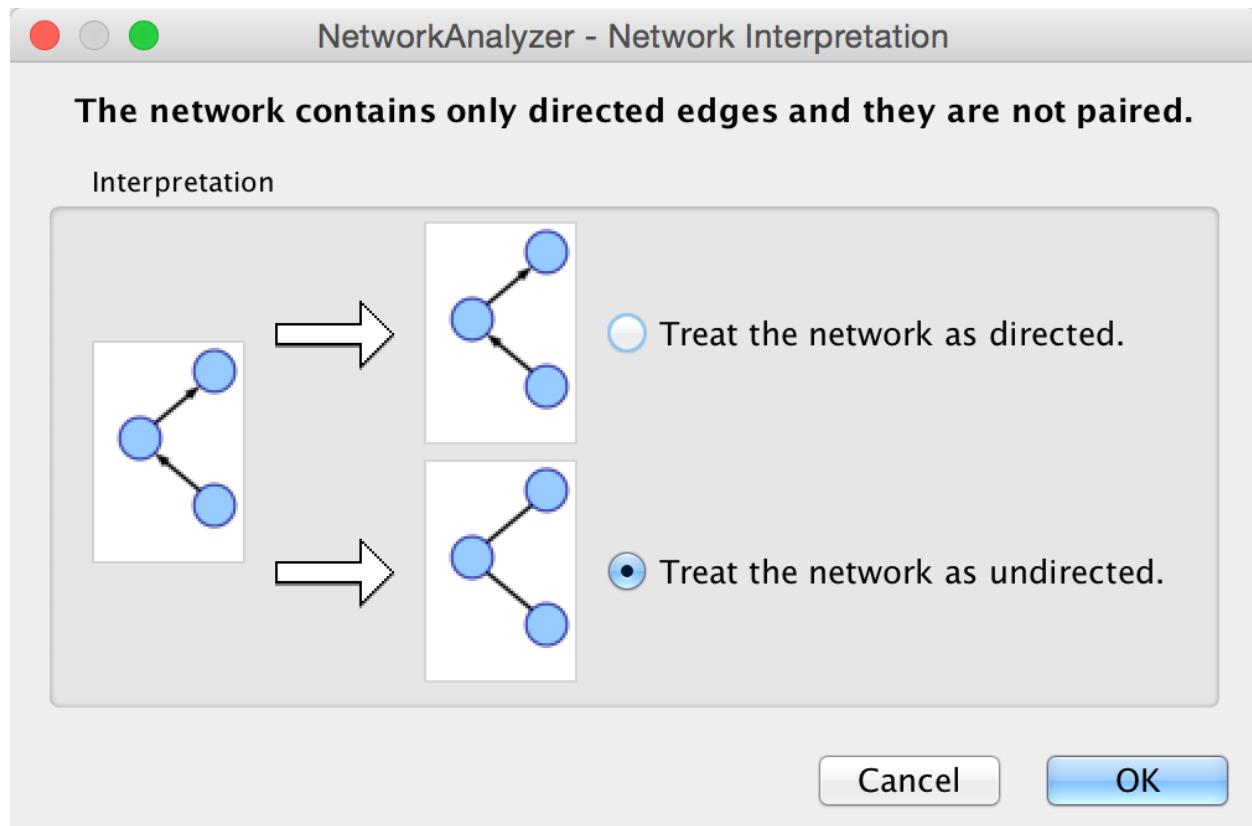
- Number of nodes, edges and connected components.
- Network diameter, radius and clustering coefficient, as well as the characteristic path length.
- Charts for topological coefficients, betweenness, and closeness.
- Distributions of degrees, neighborhood connectiveness, average clustering coefficients, shortest path lengths, number of shared neighbors and stress centrality.

NetworkAnalyzer also constructs the intersection, union and difference of two networks. It supports the extraction of connected components as separate networks and the removal of self-loops.

16.1 Network Analysis

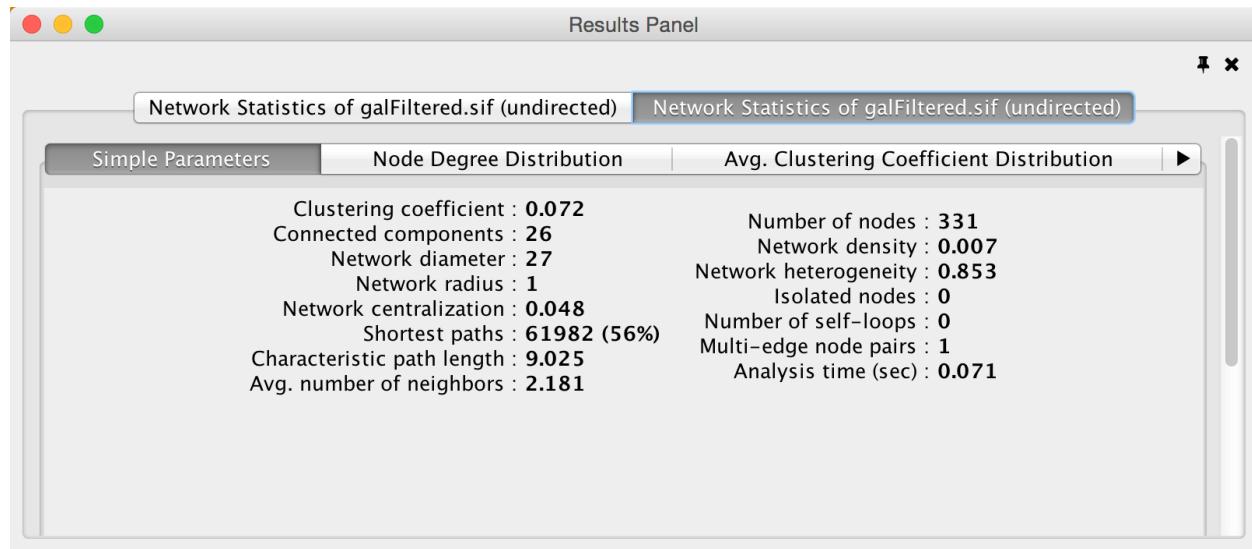
16.1.1 Analyze Network

To run NetworkAnalyzer, select **Tools** → **NetworkAnalyzer** → **Network Analysis** → **Analyze Network**.



The NetworkAnalyzer will determine whether your network contains directed or undirected edges. At this point, you can choose to ignore edge direction information.

When results are calculated, they will appear in the **Results Panel**.



The results have multiple tabs. Details on the network parameters can be found [here](#).

- **Simple Parameters**
- **Node Degree Distribution**
- **Avg. Clustering Coefficient Distribution**

- **Topological Coefficients**
- **Shortest Path Distribution**
- **Shared Neighbors Distribution**
- **Neighborhood Connectivity Distribution**
- **Betweenness Centrality**
- **Closeness Centrality**
- **Stress Centrality Distribution**

You can also save the statistics for later use by using the **Save Statistics** button.

16.1.2 Analyze Subset of Nodes

An exhaustive topological analysis of very large networks can be a time consuming task. The computation of local parameters for the nodes is significantly faster than the computation of global (path-related) parameters. Examples of local parameters are node degree, neighborhood connectivity, topological and clustering coefficients. Betweenness and closeness centralities, as well as stress, are global parameters.

[NetworkAnalyzer](#) provides the **Analyze Subset of Nodes** option for computing local parameters for a subset of nodes. If one or more nodes in the network are selected before starting an analysis, only the sub-network induced by the selected nodes is analyzed. Moreover, only local parameters are computed. Shared neighbors distribution and shortest path lengths distribution, among others, are not displayed in the results.

16.1.3 Batch Analysis

The **Batch Analysis** option is used to perform topological analysis on all networks stored in specific directory, using all possible interpretations for every network. Batch analysis consists of three simple steps:

- **Selecting directories:** The user selects the input and output directories. The input directory should contain network files that can be loaded into Cytoscape. Sub-directories of the input directory are not considered. The output directory is the one that will contain all analysis results after the batch analysis. In order to avoid file overwriting, NetworkAnalyzer requires that the output directory is empty (contains no files) before the batch analysis starts.
- **Analysis:** NetworkAnalyzer scans the input directory and loads all supported networks into Cytoscape, one at a time. Each loaded network is inspected and then it is analyzed considering all possible interpretations for it. The analysis step is complete after all networks are analyzed. Note that depending on the number of networks and their sizes, this might be a very time-consuming step.
- **Inspection of results:** After the analysis is complete, the button **Show Results** is enabled, and it displays the results dialog. The dialog contains a table of all topological analyses performed. Every row in the results table lists the loaded network, its interpretation and the resulting network statistics file that was saved in the output directory. By clicking on a network name and on statistics file name, the user can load the network and topology analysis results, respectively.

16.1.4 Load Network Statistics

Existing network statistics can be loaded from a file saved previously in [NetworkAnalyzer](#).

16.1.5 Plot Parameters

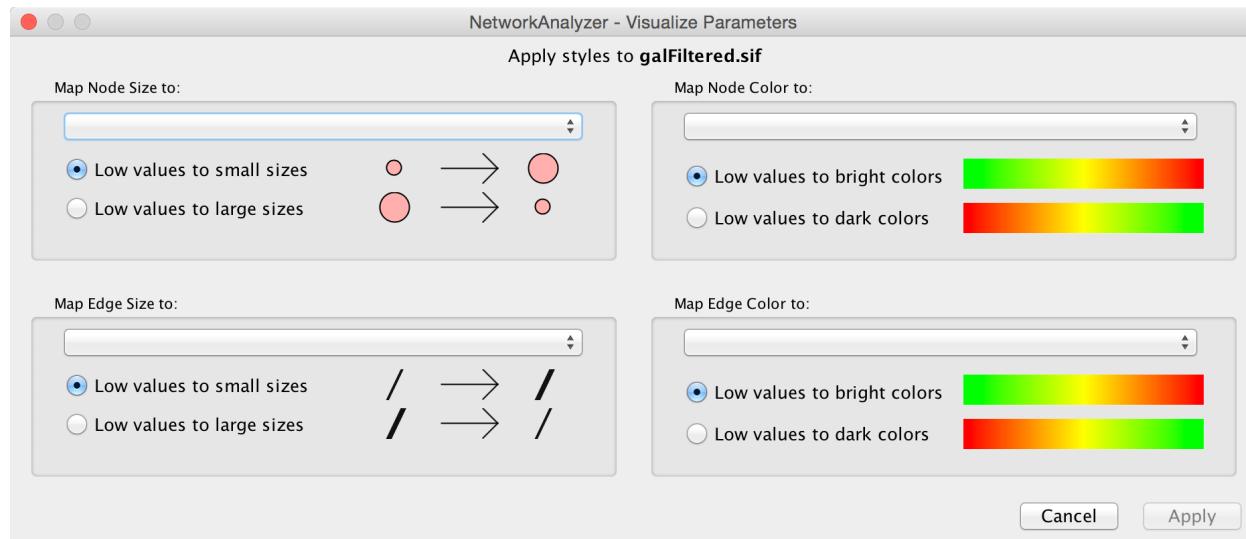
The **Plot Parameters** dialog offers a possibility to plot two parameters against each other. The parameters to be plotted can be chosen from two drop-down menus. The **Table Column 1** menu provides the values for the domain/category axis, and the **Table Column 2** menu specifies the values for the range/value axis. The plot is updated each time a different parameter is selected in one of the menus.

16.1.6 Generate Style from Statistics

NetworkAnalyzer computed parameters can be visualized as node/edge size and color, if the **Store node / edge parameters in node / edge table** option in **NetworkAnalyzer Settings** is enabled. Parameters loaded from a .netstats file cannot be visualized because the network itself is not stored in the network statistics file. If, after performing topological analysis, the network is modified by introducing or removing nodes or edges, it is recommended (and sometimes required) to run NetworkAnalyzer again before visualizing any parameters.

The visualization is initiated by the **Generate Style from Statistics...** menu option. There are two ways of mapping computed parameters.

- Map to node / edge size: The computed parameter is mapped to the size of the nodes or edges. Mapping can be straight or inverse, that is, low parameter values can be mapped to small sizes or to large sizes. The smallest node size is set to 10 and the largest one to 100. Regarding edges, size reflects the edge line width and varies between 1 and 8. Refer to the **Styles** section for details.
- Map to node / edge color: A computed parameter is mapped to the color of the nodes or edges. Two mapping styles are possible - mapping low parameter values to bright colors or to dark colors. By default, the brightest color is green and the darkest color is red. The mapping also uses a middle (intermediate) color, which allows for fine-grained perception of differing values through the color gradient. The default middle color is yellow.



16.1.7 NetworkAnalyzer Settings

The following settings can be configured by the user:

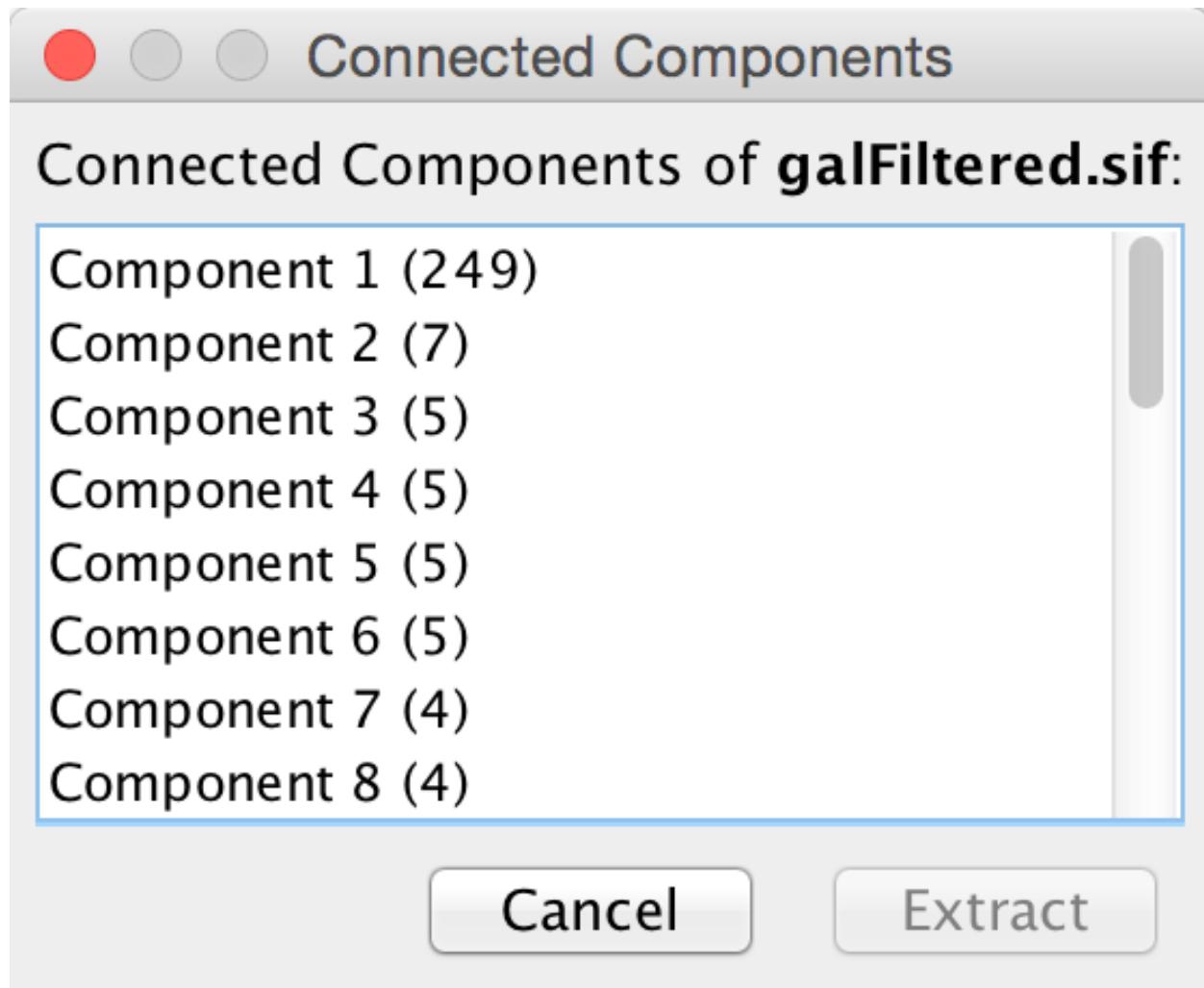
- **Store node / edge parameters in node / edge table:** For every node in a network, NetworkAnalyzer computes its degree (in- and out-degrees for directed networks), its clustering coefficient, the number of self-loops, and a variety of other parameters. NetworkAnalyzer also computes edge betweenness for each edge in the network.

If the respective options are enabled, NetworkAnalyzer can stores the computed values as columns of the corresponding nodes and edges. This enables the users to use the values in Styles or to select nodes or edges based on the values. A complete list of the computed node and edge columns is available [here](#).

- **Use expandable dialog interface for the display of network statistics:** If this option is enabled, analysis results are presented in a window in which all charts are placed below each other in expandable boxes. If this option is disabled, analysis results are presented in a window that contains tabs for the group of simple parameters and for every complex parameter (default). Users who wish to simultaneously view two or more complex parameters of one network, should enable this option.
- NetworkAnalyzer allows the user to change the default colors of parameter visualization.
 - **Background color for parameter visualization:** The color of the background in the network view. It is initially set to the default Cytoscape background color.
 - **Bright color to map parameters:** This color defines the brightest color that parameters can be mapped to. By default its value is green.
 - **Middle color:** This color defines the intermediate color, that parameters can be mapped to. By default its value is yellow.
 - **Dark color:** This color defines the darkest color that parameters can be mapped to. By default its value is red.
- **Location of the help documents:** URL of the original help web page for NetworkAnalyzer. This also enables the local download and storage of this help page.

16.2 Subnetwork Creation

NetworkAnalyzer allows for the creation of sub-networks of connected components. The user selects a number of connected components from a list and each selected component is visualized as a sub-network. To create sub-networks from connected components, select **Tools** → **NetworkAnalyzer** → **Subnetwork Creation** → **Extract Connected Components**.



16.3 NetworkAnalyzerDemo: Computation and Visualization of Topological Parameters and Centrality Measures for Biological Networks

Yassen Assenov¹, Nadezhda Doncheva¹, Thomas Lengauer¹, and Mario Albrecht¹

¹ Department of Computational Biology and Applied Algorithmics, Max Planck Institute for Informatics, Campus E1.4, 66123 Saarbrücken, Germany

NetworkAnalyzer is a versatile and highly customizable Cytoscape plugin that requires no expert knowledge in graph theory from the user. It computes and displays a comprehensive set of topological parameters and centrality measures for undirected and directed networks, which includes the number of nodes, edges, and connected components, the network diameter, radius, density, centralization, heterogeneity, clustering coefficient, and the characteristic path length. In addition, NetworkAnalyzer shows charts of the distribution of node degrees, neighborhood connectivities, average clustering coefficients, and shortest path lengths. NetworkAnalyzer also contains extra functionality, for instance, for constructing the intersection or union of two networks.

The NetworkAnalyzer plugin and a comprehensive online documentation with a tutorial are available at <http://med.bioinf.mpi-inf.mpg.de/networkanalyzer/>.

Data keywords: network, graph, topology

Cytoscape keywords: Network Analysis

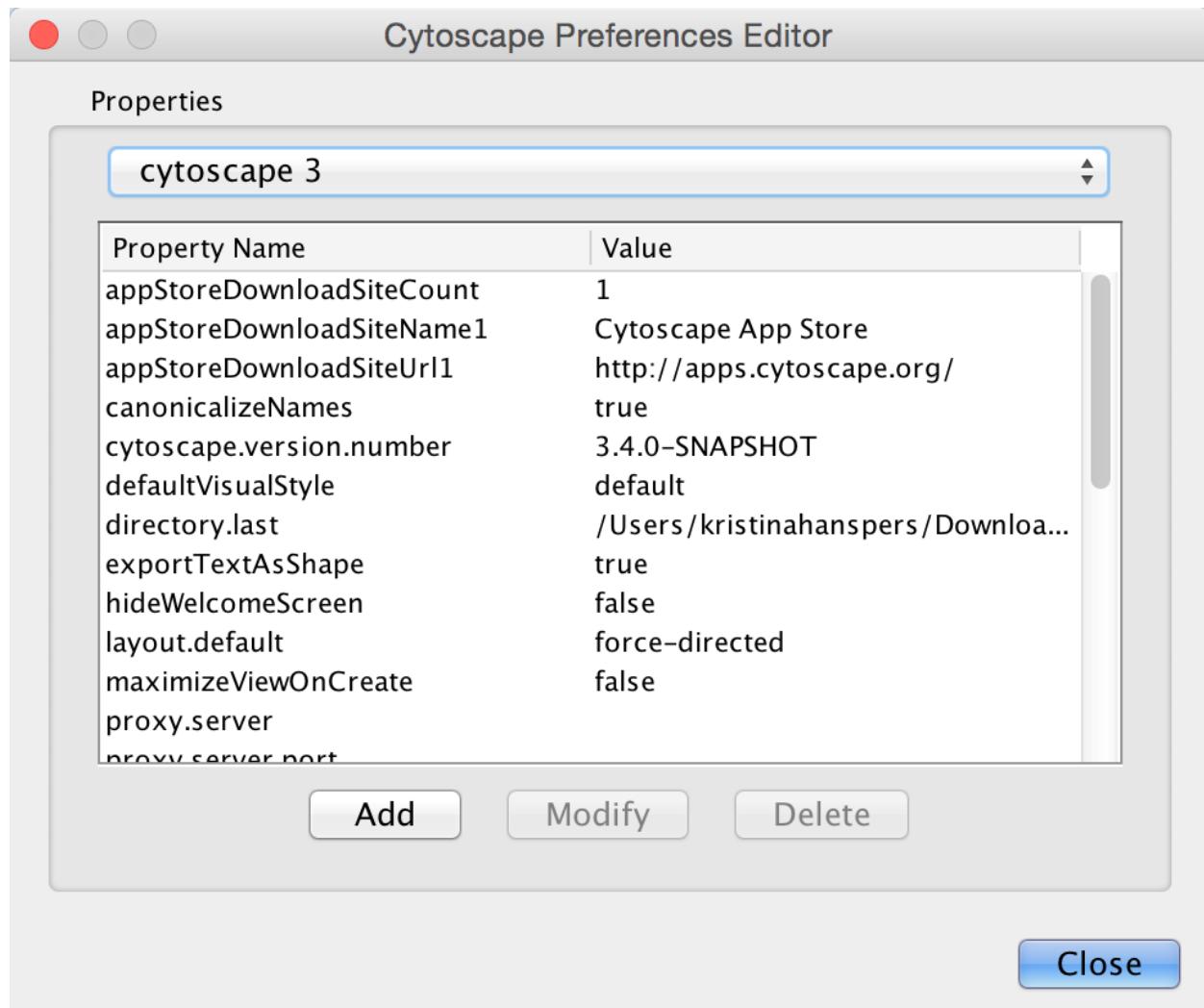
CHAPTER 17

Cytoscape Preferences

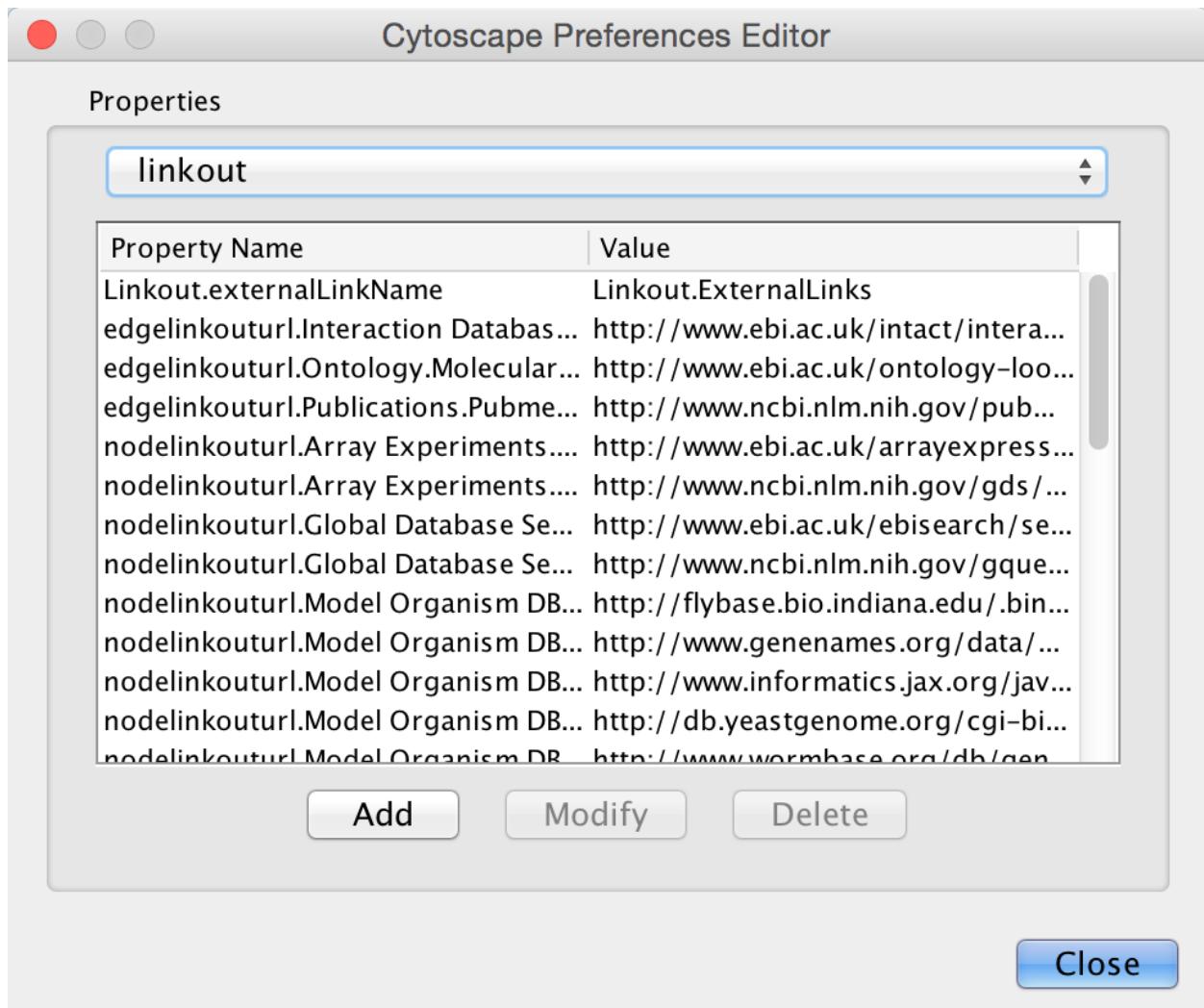
17.1 Managing Properties

The Cytoscape properties editor, accessed via **Edit → Preferences → Properties...**, is used to specify default properties. Any changes made to these properties will be saved in .props files under the `CytoscapeConfiguration` subdirectory of the user's home directory.

Cytoscape properties are configurable using the Add, Modify and Delete buttons as seen below.

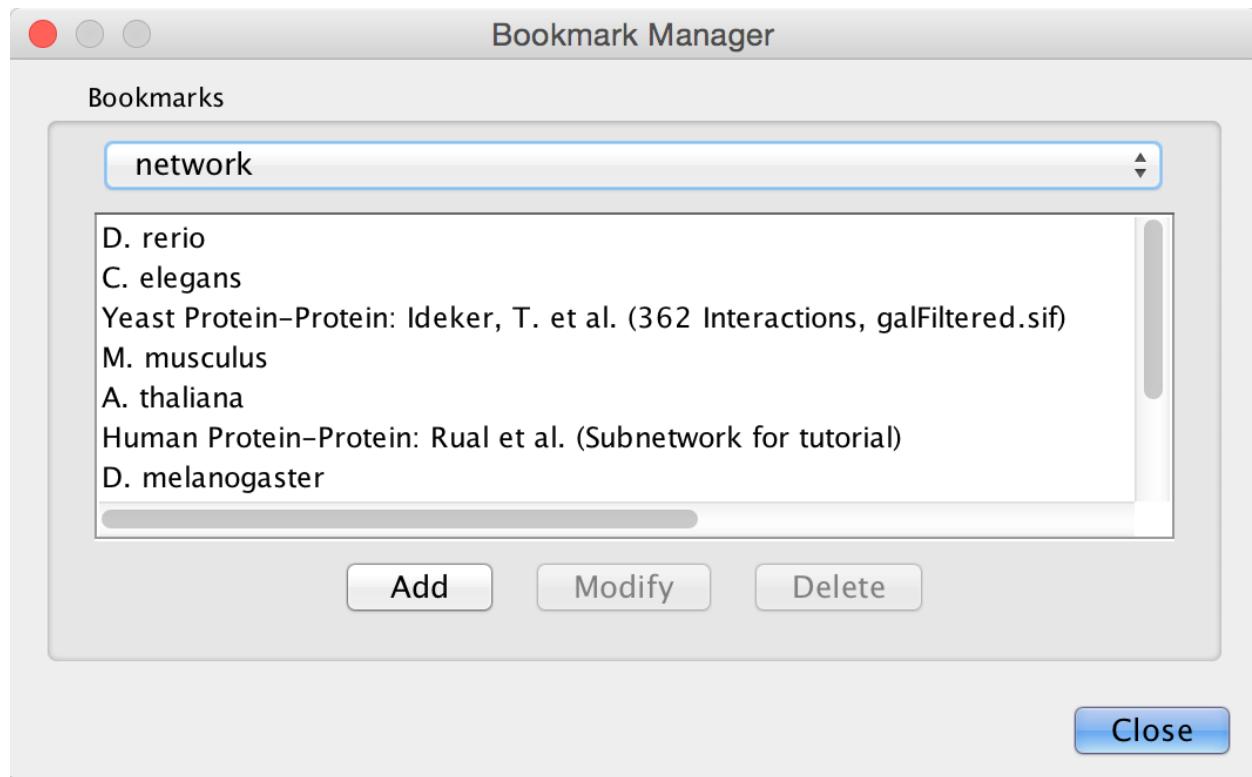


App properties may also be edited in the same way as editing Cytoscape properties. For example, to edit the properties of Linkout, select ‘linkout’ from the combobox of the Preferences Editor. Some apps may store properties inside session files in addition to (or instead of) storing them in the `CytoscapeConfiguration` directory.



17.2 Managing Bookmarks

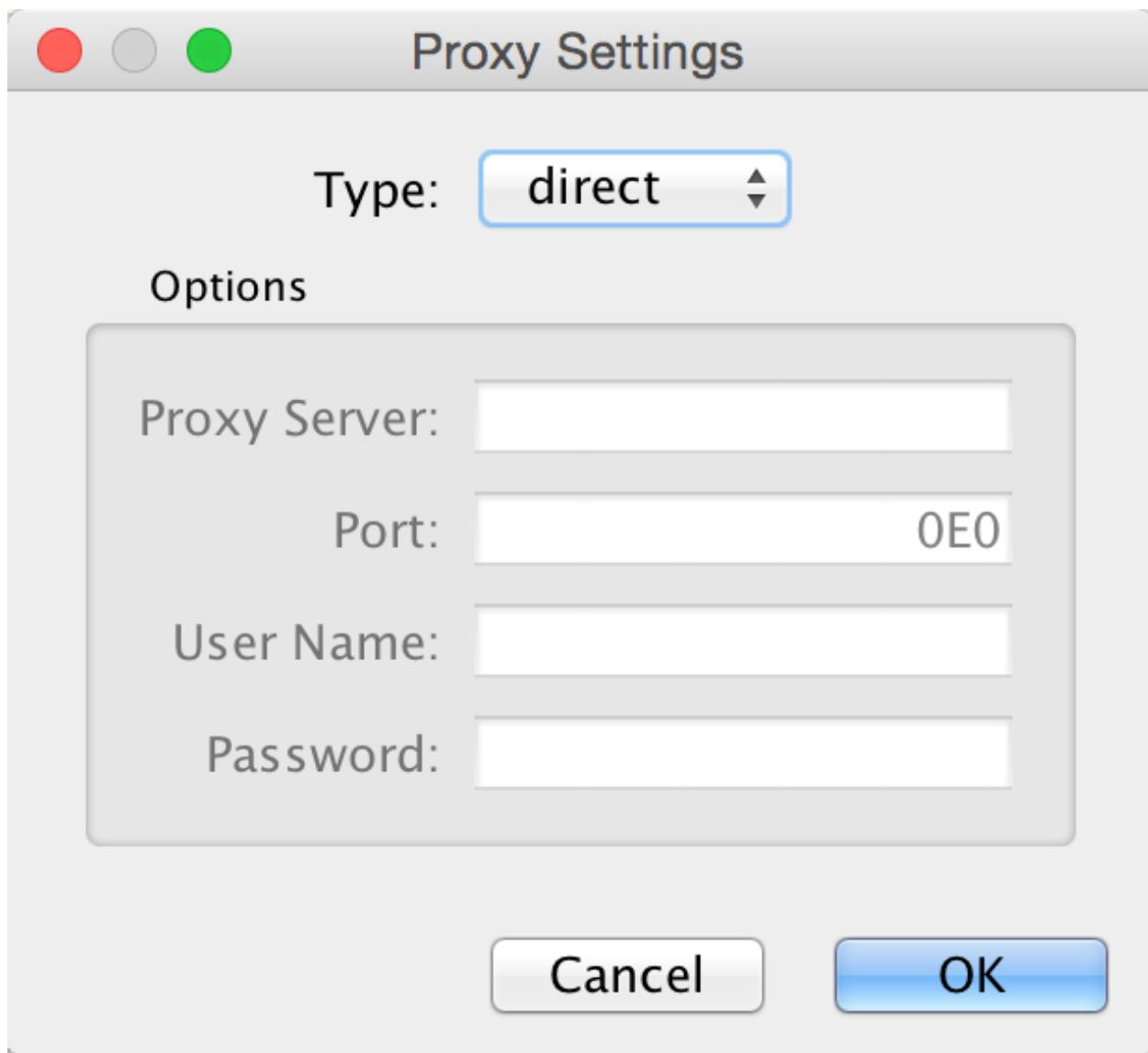
Cytoscape contains a pre-defined list of bookmarks, which point to sample network files located on the Cytoscape web server. Users may add, modify, and delete bookmarks through the Bookmark manager, accessed by going to **Edit** → **Preferences** → **Bookmarks**....



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.

17.3 Managing Proxy Servers

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

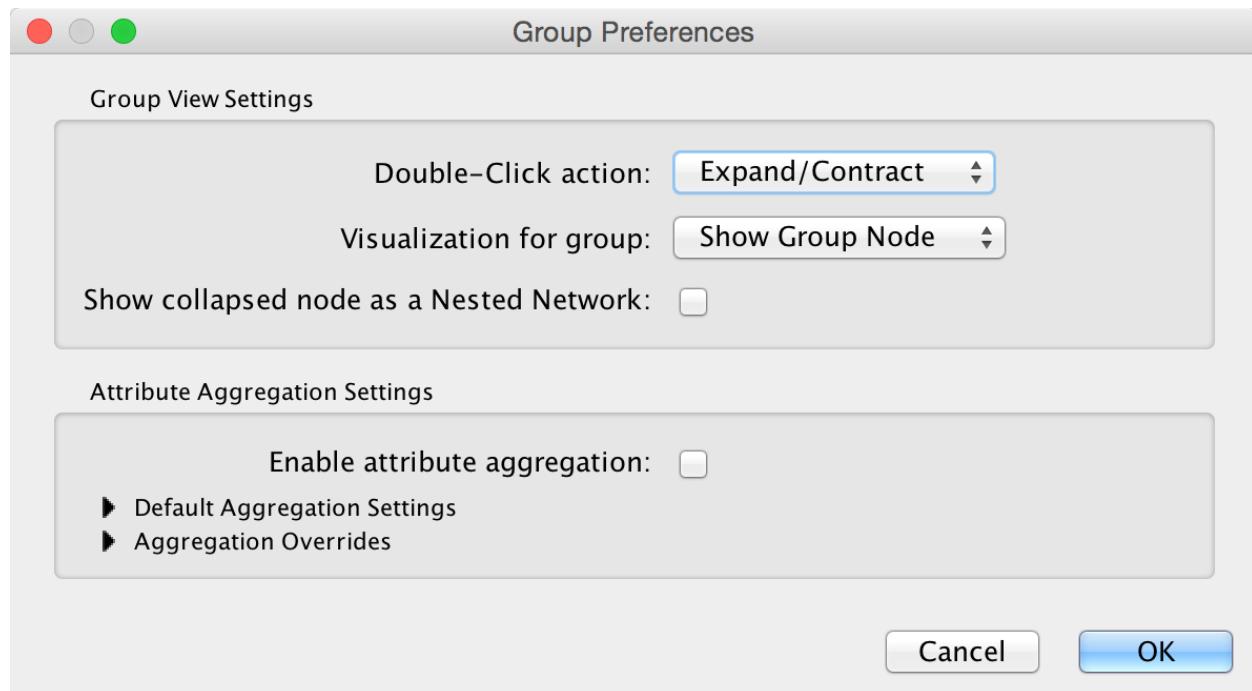


After the proxy server is set, all network traffic related to loading a network from URL will pass through the proxy server. Cytoscape apps use this capability as well. The proxy settings are saved in `cytoscape3.props`. Each time you click the OK button after making a change to the proxy settings, an 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.

17.4 Managing Group Settings

The configuration of Cytoscape group view may also be edited through **Edit → Preferences → Group Preferences...**.



Note that Group Preferences apply to [node groups](#) established after the group preferences are set. They do not apply to groups that already exist.

The **Group Preferences** dialog provides access to three **Group View Settings** and all of the **Attribute Aggregation Settings**. There are two interacting settings involved in the group view and group interaction:

- how an expanded group appears
- what happens when a node (group member or group node) is double-clicked
- what appears inside of a group node when it is collapsed

17.4.1 Group View Settings

The following node visualization options are available:

- None: No specific visualization – just do expand/contract, but don't treat the expanded group special in any way
- Compound Node: Show the group node as an area surrounding the member nodes, but position it behind the member nodes to allow direct selection of each of the members. If you move a node within this area, you may need to resize the area to encompass the newly positioned member node. Moving the group node will cause all member nodes to move with it. There are [visual styles](#) that allow you to change the color, shape and padding for the compound node.
- Show Group Node: When the group is expanded, show the group node as an additional node and add “member edges” between the group node and each of the member nodes. This is useful, for example, for complexes where it may be important to show a group node (the entire complex) at the same time as group member nodes (each individual protein).
- Single Node: Show the group node as an area surrounding the member nodes, but put it in front of the member nodes so that the member nodes can not be selected or moved individually.

If you set the Double-Click action to something other than **Expand/Contract**, the groups won't collapse on double-click, which may be appropriate for displaying group nodes that contain other nodes (i.e., the **Compound Node** visualization or **Single Node** visualization).

17.4.2 Attribute Aggregation Settings

Attribute aggregation provides an automated way for a group to aggregate all of the attributes of its children. Since the columns in a Cytoscape network must all be of the same type, a group can't simply create a list of integers to aggregate the integer columns of its members – the right approach to aggregating the attributes of group members will obviously depend on the application. Cytoscape allows you to set the default aggregation approach for each type of column: Integer, Long, Double, String, Boolean, String List, Integer List, Long List, or Double List in the **Default Aggregation Settings** section. In general, the user is provided with a list of some common aggregations. For example, Integer aggregations include Average, Minimum Value, Maximum Value, Median Value or Sum.

In addition, users can indicate that a specific column should use an aggregation approach different from the default by looking at **Aggregation Overrides**.

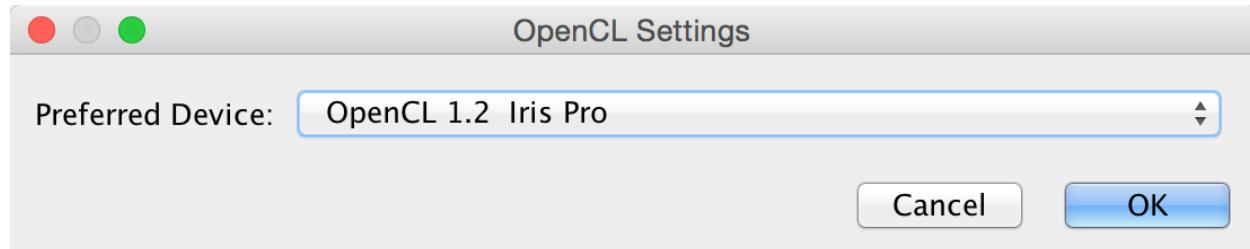
17.4.3 Grouping Tips

The **Show collapsed node as a Nested Network** checkbox determines the contents of a grouping node when its member nodes are hidden. If it is checked, the group node contains a graphic of the laid out member nodes. For the Compound Node visualization, this box should remain unchecked so the grouping node can appear as a background to member nodes.

Attribute aggregation only occurs when the group is collapsed or when the group visualization is changed (e.g., from **Single Node** to **Compound Node**).

17.5 Managing OpenCL Settings

You can choose between one or more OpenCL drivers installed on your system by going to **Edit → Preferences → OpenCL Settings....**



OpenCL is a library that enables Cytoscape to use your system's graphics processing unit (GPU) to accelerate certain layouts and other calculations. If no choices are presented, consult the support web page for your system's graphics card.

CHAPTER 18

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.

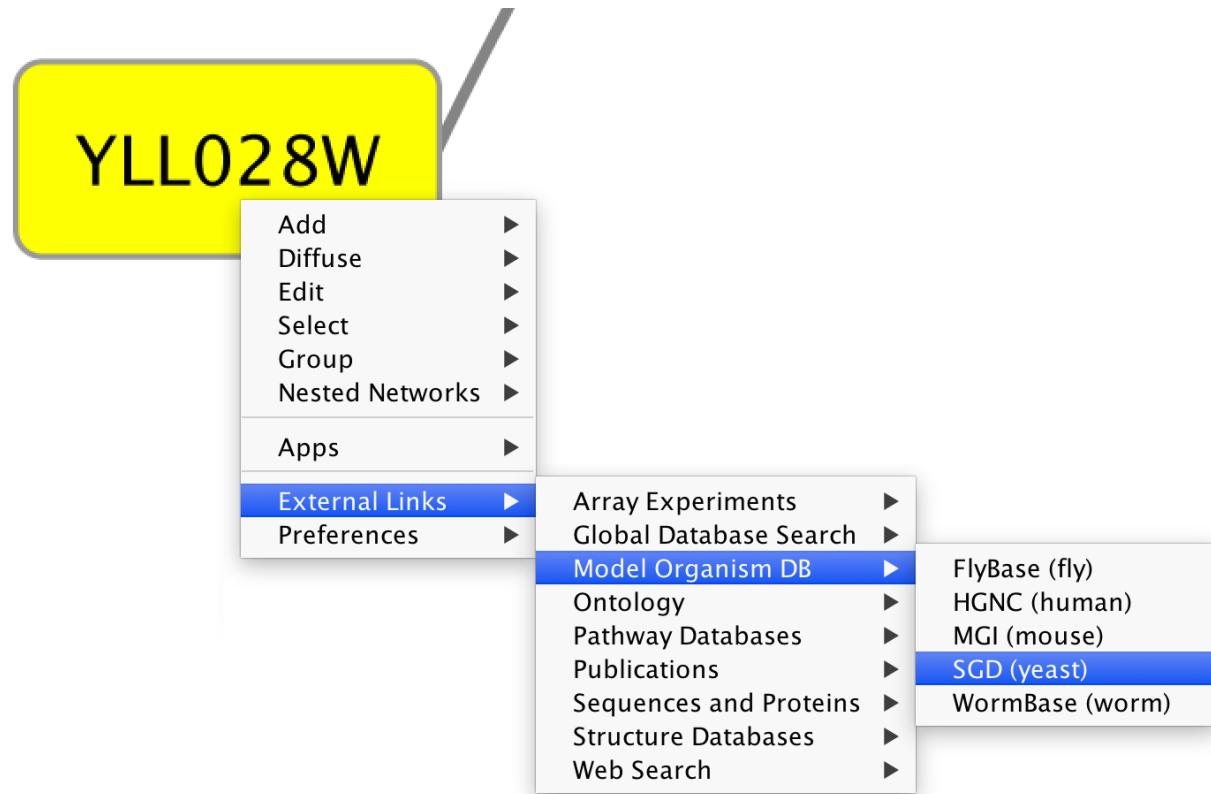
By default, Cytoscape includes a number of links such as Entrez, SGD and Google, as well as a number of species-specific links. In addition to the default links, users can customize the **External Links** menu and add (or remove) links by using the Cytoscape Preferences Editor (found under **Edit → Preferences → Properties...** in the *linkout* group).

External links are listed as '*key*'-'*value*' pairs in the editor where *Property Name* specifies the name of the link and *Value* is the search URL. Linkout menus are organized in a hierarchical structure that is specified as part of the key. Linkout key terms specific to 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/locus.  
↪fpl?locus=%ID%
```

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
```

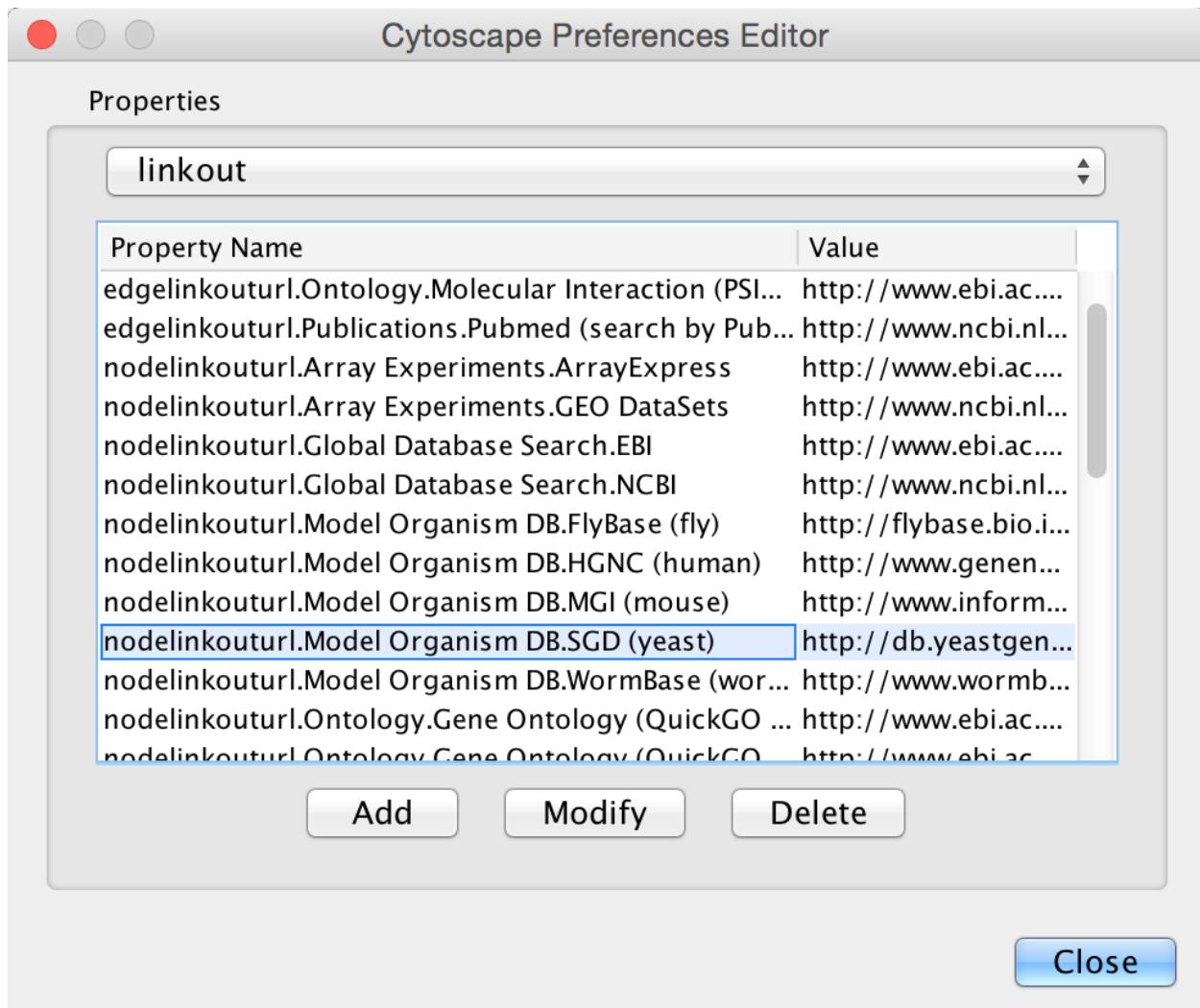
If you want to query based on a different column, you need to specify a different node label using Styles.

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.

18.1 Adding and Removing Links

The default links are defined in the `linkout.props` file contained in the `CytoscapeConfiguration` directory. These links are normal Java properties and can be edited directly via text editor or by using the Cytoscape Configuration Editor (**Edit → Preferences → Properties...**) and selecting `linkout` (shown below).

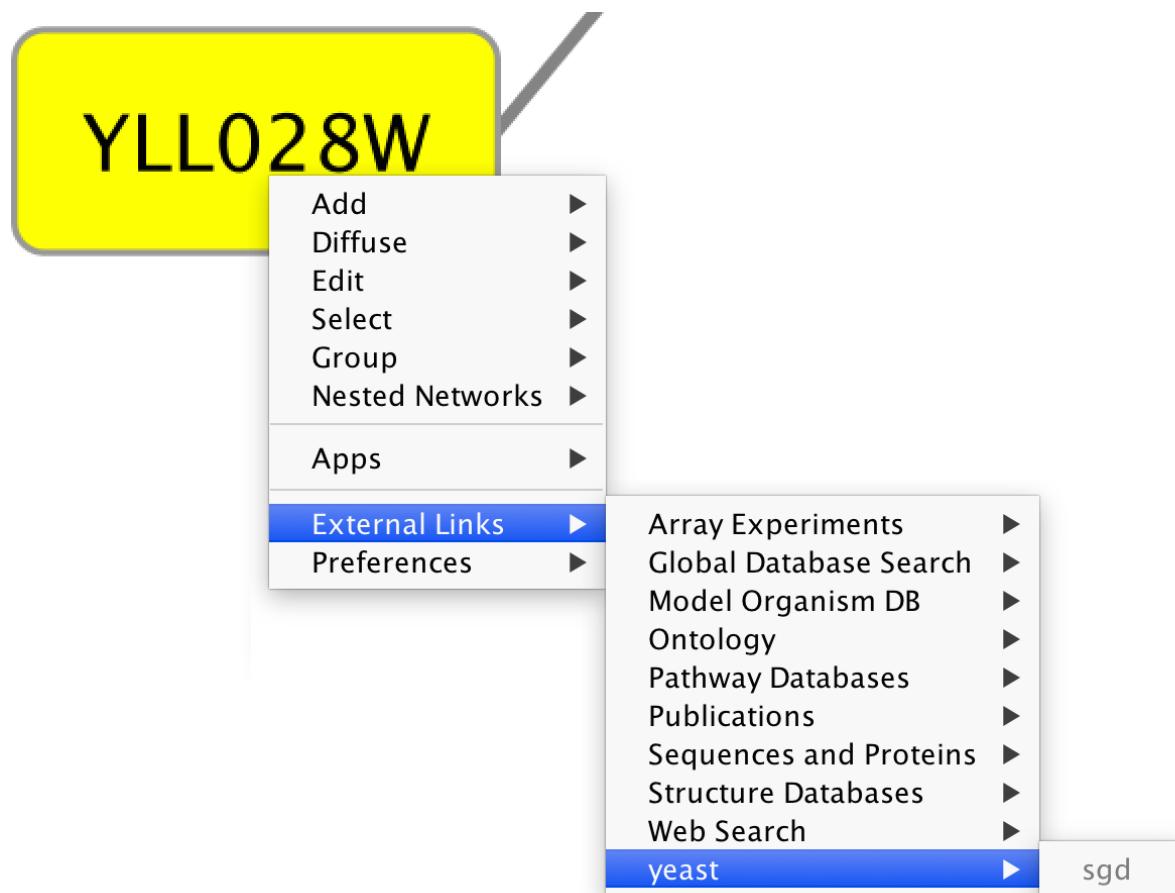


In addition, new links can be defined when starting Cytoscape from command line by specifying individual properties. The formatting of the command is `cytoscape.sh -P [context_menu_definition]=[link]`. `context_menu_definition` specifies the context menu for showing the linkout menu item. The structure of this definition is `nodelinkouturl` or `edgelinkouturl` followed by ":" and followed by the ":"-separated menu hierarchy.

For instance this command:

```
cytoscape.sh -P nodelinkouturl.yeast.SGD=http://db.yeastgenome.org/cgi-bin/locus.pl?
↪ locus\=%ID%
```

will add this menu item:

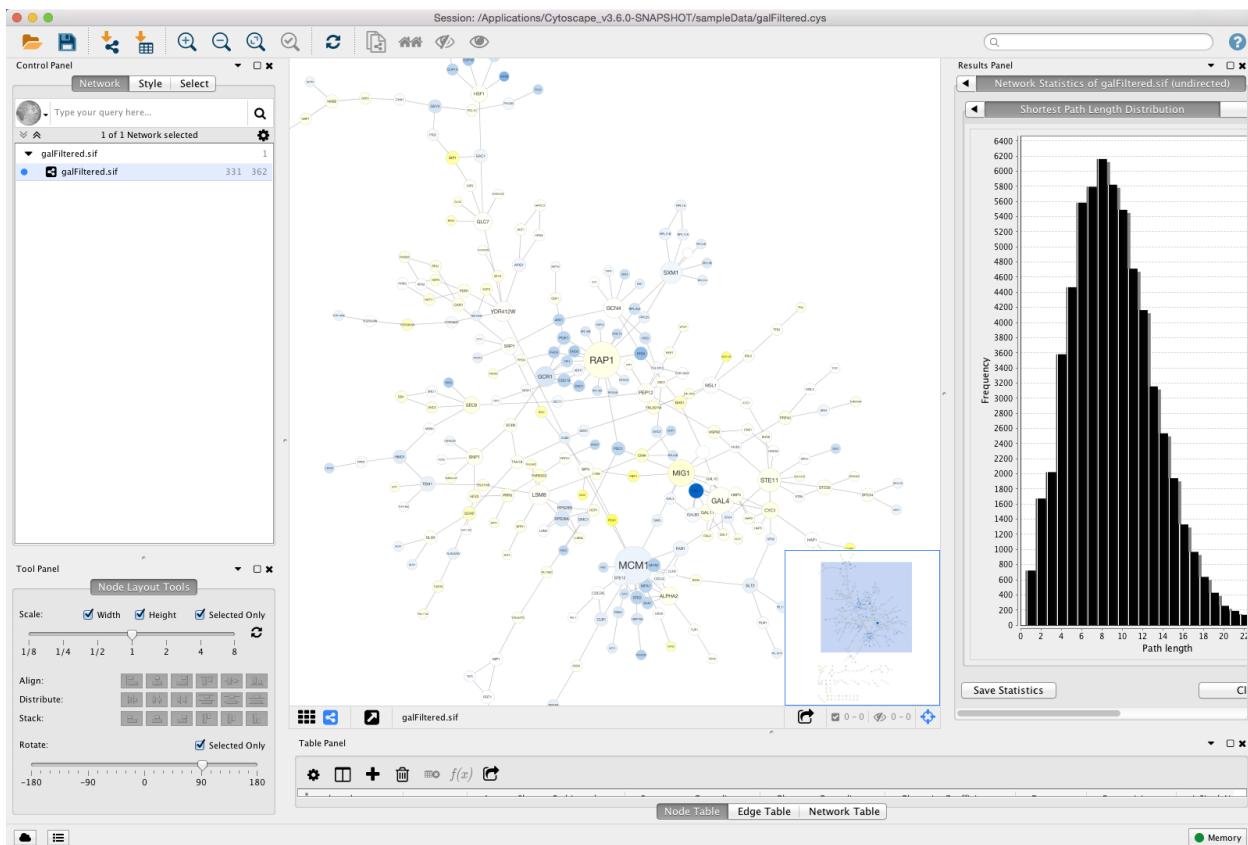


To remove a link from the menu, simply delete the property using **Edit → Preferences → Properties...** and selecting *commandline*. Linkouts added in the command line will be available for the running instance of Cytoscape.

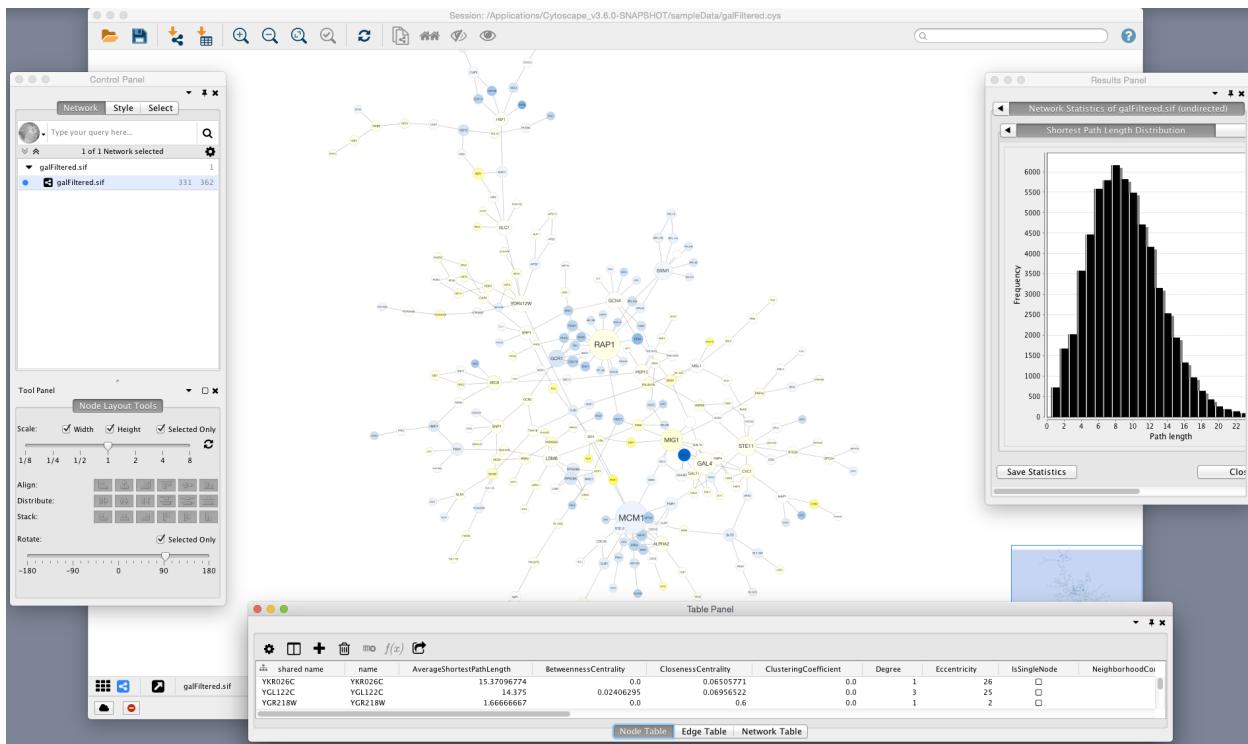
CHAPTER 19

Panels

Panels 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 **Results Panel**. The following screenshot shows the file `galFiltered.sif` loaded into Cytoscape, with a force-directed layout, **Rotate** tools showing in the **Tool Panel**, and with results from Network Analyzer (**Tools** → **Network Analysis** → **Analyze Network**). The **Control Panel** (at the left-hand side of the screen) contains the Network Manager, Network Overview, Styles and Select tabs. On the bottom of the panel, there is another panel called **Tool Panel**. In the **Table Panel**, the **Node Table** is shown. In addition, analysis results from Network Analyzer are shown in **Results Panel** (at the right-hand side).



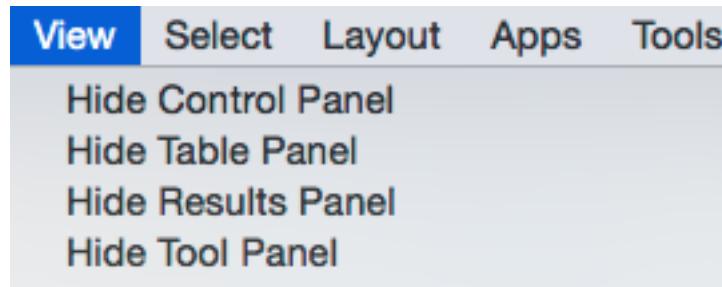
The user can then choose to resize, hide or float Panels. For example, in the screenshot below, the Control, Table and Results panels are floating:



19.1 Basic Usage

Cytoscape includes four Panels: the **Control Panel** on the left, **Tool Panel** on the bottom of the **Control Panel**, the **Table Panel** on the bottom right, and the **Results Panel** on the right. By default, only the **Control Panel** and the **Data Panel** will appear. The **Results Panel** may appear, depending on the mix of Cytoscape apps that you currently have installed. The **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, Panels can be floated or docked using icon buttons at the top right corner of each Panel. The **Float**

Window control will undock any panel which is useful when you want assign the network as much screen

space as possible. To dock the window again, click the **Dock Window** icon . Clicking the **Hide Panel** icon

will hide the panel; this can be shown again by choosing **View → Show** and selecting the relevant panel.

CHAPTER 20

Rendering Engine

20.1 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 anti-aliasing 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

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

20.1.1 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.

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.

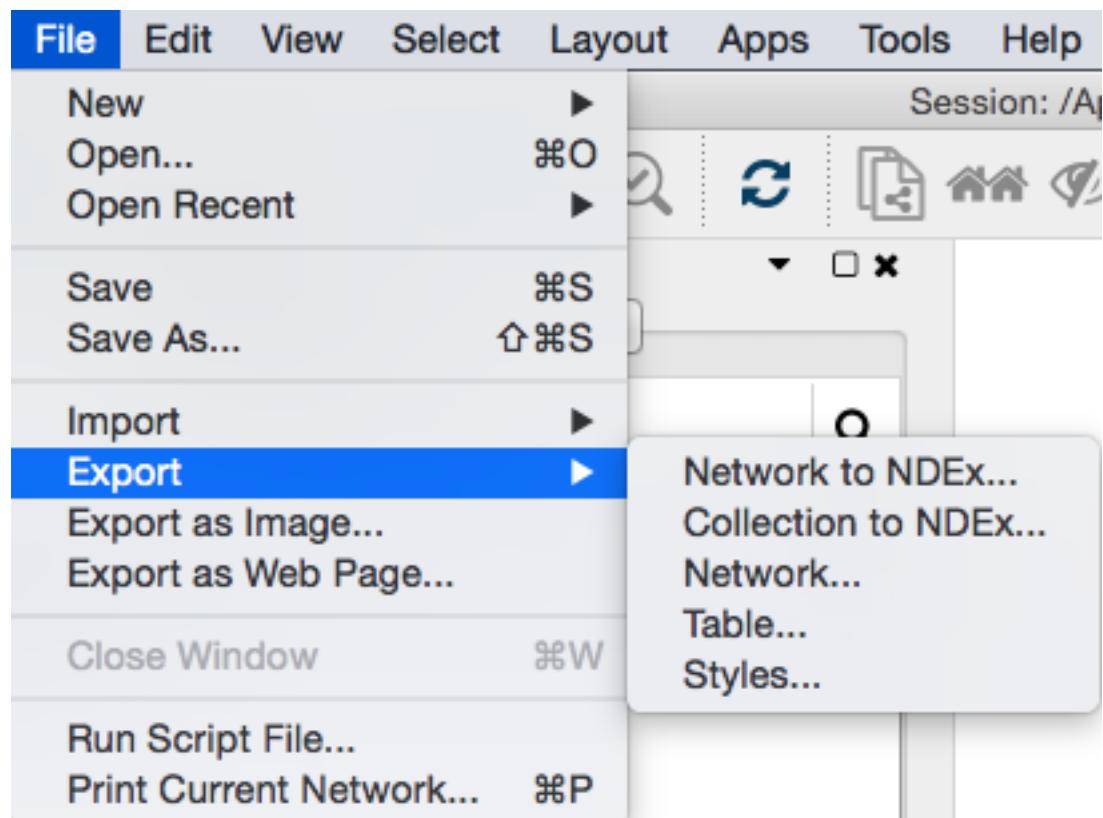
20.1.2 Force to Display Detail

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+D on Windows/Linux, Command+SHIFT+D 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**).

Export Your Data

21.1 Raw Data Export



To use data from Cytoscape networks in an external application, the data has to be exported. As a Cytoscape network consists of a number of different types of data, there are options in Cytoscape for each exportable data type. These options (Network, Table, Styles) can be found on the **File → Export** submenu (shown above).

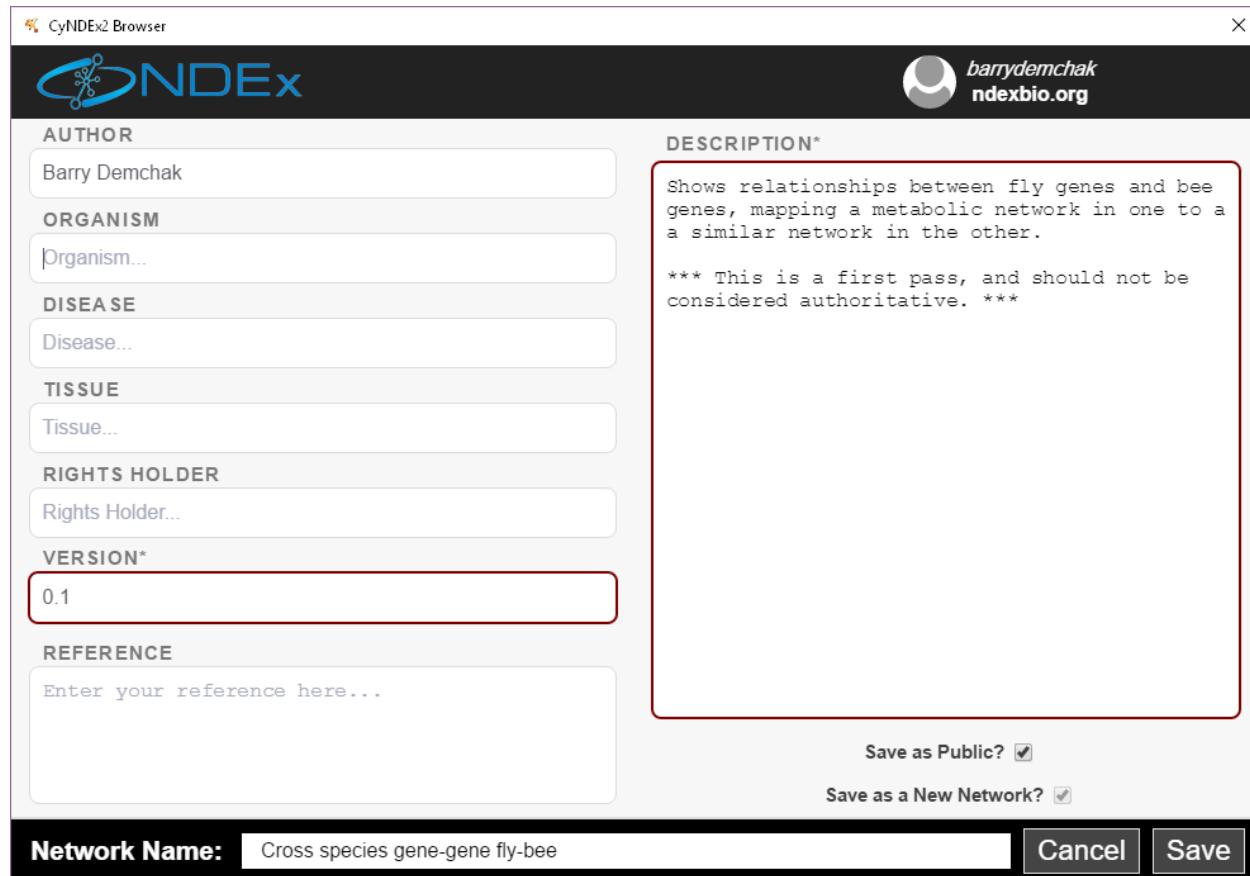
21.1.1 Network Data Exchange (NDEx) Public Database

The **Network to NDEx** and **Collection to NDEx** options are used to store a network on a public NDEx server. You can use an NDEx server as a cloud-based store for your networks, as a means of [sharing networks](#) with collaborators, or to [archive networks](#) for publication. To store a network, you must have an NDEx account, which you can create by visiting the [NDEx Public Server](#) website.

The **Network to NDEx** option saves only the currently selected network to your NDEx account. Such networks are suitable for processing by other applications and web services, as they contain only the nodes and edges in the selected network – other networks and the nodes and edges unique to them are not written. If you import this network (using the Cytoscape [Search bar](#)), only the single network will be loaded.

The **Collection to NDEx** option saves all networks in the collection containing the currently selected network. If you import this collection (using the Cytoscape [Search bar](#)), the entire collection will be loaded. This option is closest to saving an entire Cytoscape session, except that only the current collection is saved.

When you select either option, the CyNDEx-2 Browser dialog will appear and enable you to provide metadata that will be displayed during subsequent NDEx searches. To save a network, you must add an NDEx account profile to the CyNDEx-2 Browser (by filling out the profile form reachable by clicking on the *Anonymous* credential in the upper right), and then make it current.



You can set the name of your network by editing the **Network Name** attribute at the bottom. The **Save as a New Network** option enables you to create a new network instead of overwriting an existing NDEx network of the same name. If you choose to make your network public, you must fill in the **Version** and **Description** attributes. Choose the **Save as Public** option to enable collaborators to load your network.

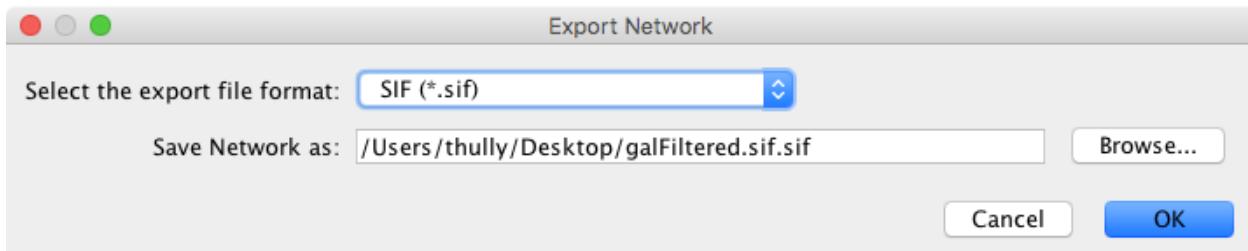
When the export to NDEx is complete, you will be given an opportunity to capture the network's UUID in your clipboard. The UUID is suitable for e-mailing to colleagues so they can load your network via their Cytoscape [Search](#)

bar (provided you selected the **Save as Public** option). For more sharing options, see the NDEx Public Server website.



Further information on NDEx is available at the [NDEx Informational/Doc](#) website.

21.1.2 Network

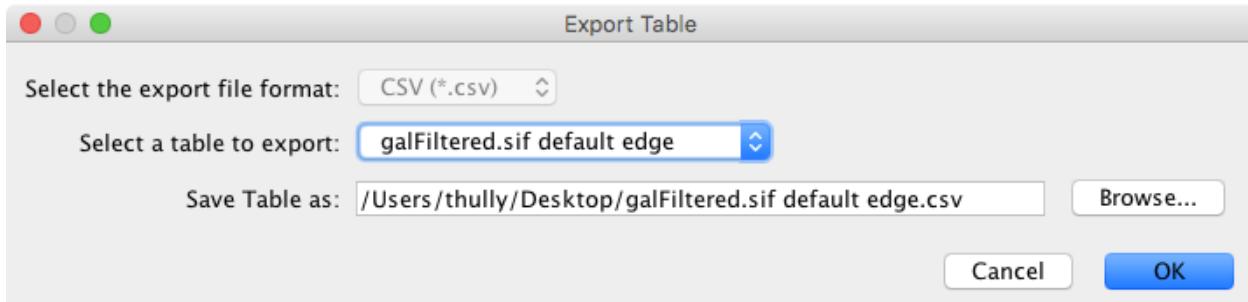


The **Network** option is used to export network data. Network data includes the nodes and edges in the network, and also will include data attributes and visual information if the chosen format supports it (like XGMML). Network data can be exported in a number of formats, including:

- SIF (Simple Interaction Format)
- NNF (Nested Network Format)
- XGMML
- GraphML
- PSI-MI Level 1 and 2.5
- Cytoscape.js JSON (can be used with Cytoscape.js, a tool described in future sections)
- CX JSON (for Cytoscape Cyberinfrastructure network exchange)

The default format is SIF, which is a simple tab-delimited network format that provides node names and edge interactions only. You can choose the format using the dialog drop-down box. A file name is automatically suggested based on the network name and selected format, with the file path defaulting to the current working directory. You can change the suggested file name/path using the text box or Browse button. Press OK to export with the selected options.

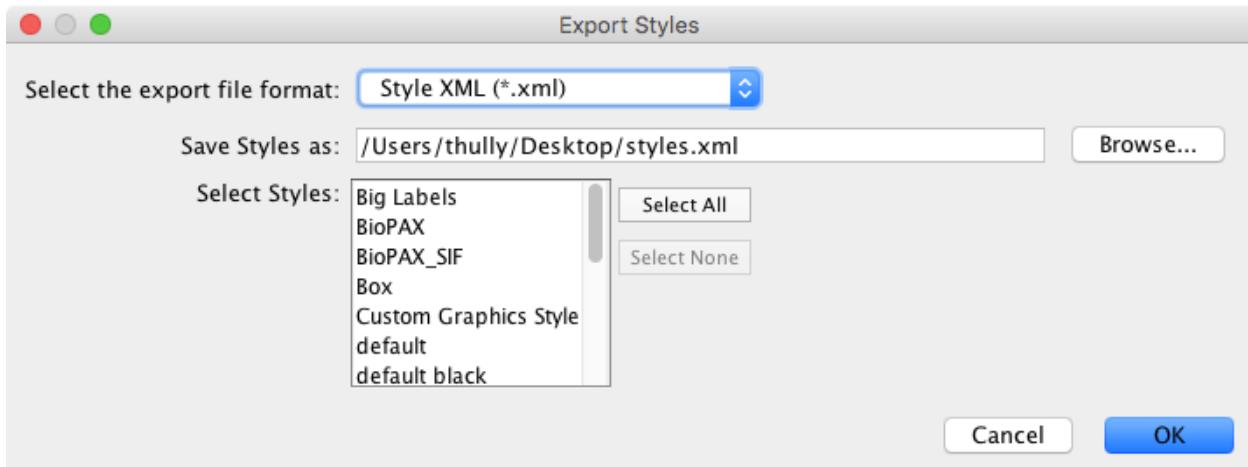
21.1.3 Table



The **Table** option is used to export any of the data tables that are available in the current Cytoscape session. This includes the node, edge, and network tables that can be seen in the Table Browser, as well as any unassigned tables. The only currently supported file format currently for export is CSV (comma-separated value).

The export dialog for Table has only two selectable options - the table to export and the file name (file type is always CSV). A file name is suggested based on the name of the selected table with the path defaulting to the current working directory. As in Network, the suggested file name can be changed using the Browse button or the text field. Press OK to export the selected table with the given file name.

21.1.4 Styles



The **Styles** option is used to export one or more visual styles available in the current Cytoscape session. Supported formats include Style XML (which can be imported by 3.x versions of Cytoscape) or Cytoscape.js JSON (which can be used with Cytoscape.js, described in future sections).

Style export has three options - the export format (as described above), the style(s) to export, and the file name (a suggestion will be made based on the current directory and file type). As in Network → Table, the suggested file name can be changed using the Browse button or the text field. The list of styles is a multiple-selection list - one or more can be selected using Ctrl/Cmd-click (to select multiple discontiguous elements) or Shift-click (to select a contiguous element range). Press OK to export all selected styles with the given file path.

21.2 Exporting for Publication

When you finish your data analysis and visualization, you can publish your data to share the results. Cytoscape has several options to do this, with most options suitable for Cytoscape users and other options suitable for programmers wanting to create unusual or complex network viewers. These are further explained below.

- An NDEx network
- A session file
- A static image
- An interactive web application
 - CyNetShare
 - Full web application
 - Simple network view (for web application developers)

21.2.1 As an NDEx Network

The easiest way to share your results with others is to save your network in the NDEx Public Database by using the [Network to NDEx](#) or [Collection to NDEx](#) options. When you save your network, you will receive a UUID (e.g., 50e3dff7-133e-11e6-a039-06603eb7f303) that you can e-mail to collaborators for use in their own Cytoscape [Search bar](#) (provided you exported using the **Save as Public** option – see the [Save as Public](#) option.)

NDEx also has options that enable you to:

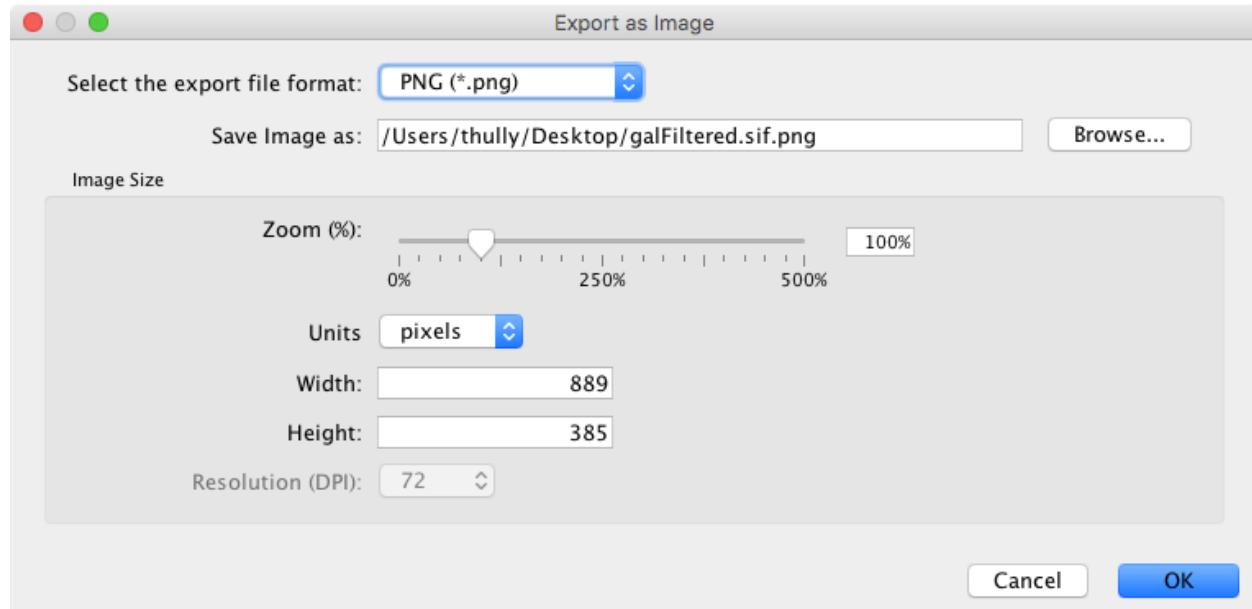
- generate a DOI suitable for use in publications
- create an anonymous sharable link (similar to a Google Doc shared link)
- create interest groups and specific sharing within NDEx

For more information on NDEx publishing and sharing features, see the [Publishing in NDEx](#) and [Sharing Networks in NDEx](#) tutorials.

21.2.2 As a Session File

Another easy way to share your results with others is simply saving everything as a session file (which is a zipped session archive). You can save your current session by clicking the **Save Session** icon. You can save to a thumb drive, a shared file system, or even a cloud drive directory such as Dropbox. If you save to a shared drive, beware not to have two people work on the same session file with Cytoscape at the same time, as unpredictable results may occur.

21.2.3 As a Static Image



Cytoscape can generate publication-quality images from network views. By selecting **File → Export as Image...**, you can export the current network view into the following formats:

- JPG
- PNG
- PS (Post Script)
- SVG
- PDF

The image export dialog has a variable number of options based on the file format selected, with file type and file name always being selectable. PNG and JPEG have options for scaling, while other formats only have a single option, Export Text as Font. The file name is suggested based on the name of the selected table with the path defaulting to the current working directory. As in raw data export, the suggested file name can be changed using the Browse button or the text field. Press OK to export an image with the given file name and parameters.

We recommend using **PDF** for publications because it is a standard vector graphics format, and it is easy to edit in other applications such as Adobe Illustrator.

Known Issues

For PDF export, there is an option to **Export text as font**. **This option does not work for two-byte characters such as Chinese or Japanese**. To avoid corrupted texts in the exported images, please uncheck this option when you publish networks including those non-English characters.

21.2.4 As an Interactive Web Application (New in 3.2.0)

The Web is an excellent platform for data sharing and collaboration, and Cytoscape provides a number of ways to publish your network on the web, with each choice representing tradeoffs between ease, simplicity, and customization options. All solutions leverage the cytoscape.js drawing library, and so enable not only viewing but also Cytoscape-style interactive browsing of networks and attributes.

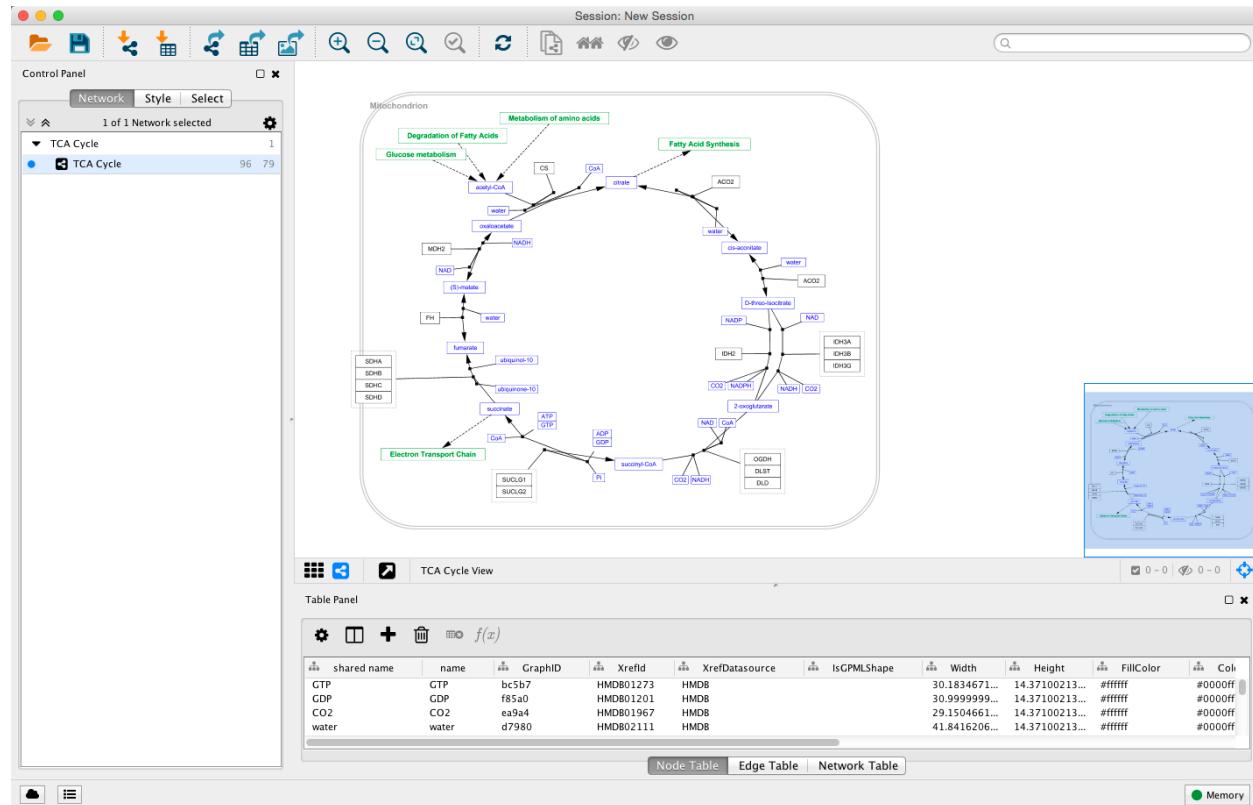
The simplest choice is [CyNetShare](#), where you save your network (and optionally a style) on a public file system, then interactively view the network in a browser. Like Google Maps, you can generate and publish a URL that allows collaborators to also view your network.

Alternatively, Cytoscape can generate an entire web site showing a single page containing the viewer with your network pre-loaded. You can load this directly onto your own web server to become part of your web site.

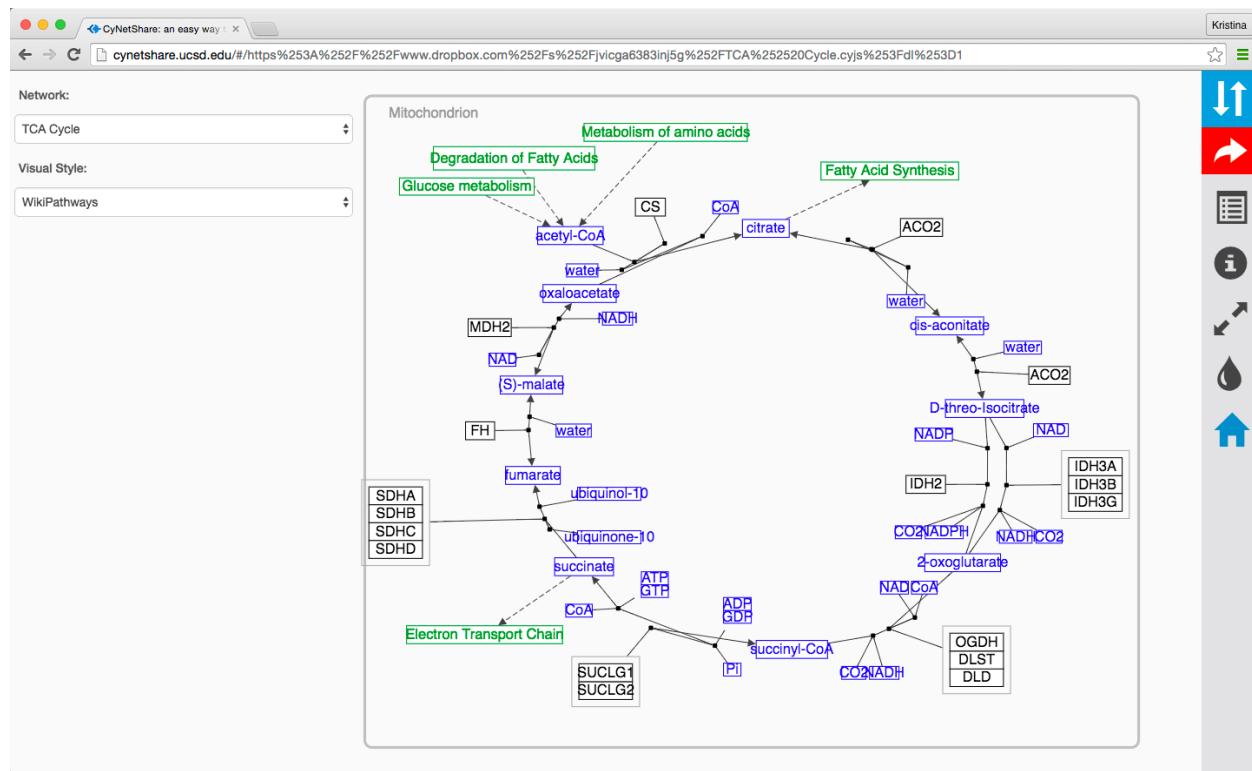
Finally, Cytoscape can generate the skeleton of a web site for further customization by [JavaScript](#) programmers.

These features are available as Export menu items under the File menu, and are described in sections below.

For example, here is a network in Cytoscape:



Here is the same network as an interactive web visualization:



Note that web browsers can render small networks (e.g., 1000 nodes) quickly and effectively, but attempting to render large ones (e.g., 5000 nodes) will take a very long time.

A word about exporting styles to interactive web applications: Our web applications are based on the `cytoscape.js` display library, which renders a subset of Cytoscape styles. For more information, see the [Export Styles to Cytoscape.js](#) section below.

Sharing via CyNetShare

[CyNetShare](#) is a browser-based web application that renders JSON-formatted networks and attributes saved in public directories. Optionally, you can specify visual styles that the web application will use to draw your network as it appears in Cytoscape. [CyNetShare](#) is similar to Google Maps in that once you have loaded your network and have tweaked its appearance to suit your needs, you can have [CyNetShare](#) generate a new URL that you can e-mail or post as a link on your own web site. That URL will bring up [CyNetShare](#) preloaded with your network for anyone to see.

To use [CyNetShare](#):

1. Select **File → Export → Network...** to export your network to a public directory. Choose the *Cytoscape.js JSON (*.cyjs)* export file format.
2. Optionally, select **File → Export → Style...** to export your style settings. Choose the *Style for cytoscape.js (*.json)* export file format.
3. Use your public directory system to determine direct URLs for the files you exported.
4. Start [CyNetShare](#)
5. Enter the network's URL as the Graph URL.
6. Optionally, enter the style's URL.
7. Click the **Visualize** button.

The [CyNetShare](#) User Guide is provided on the [CyNetShare](#) web page:

- CyNetShare

Note that if you specify a style URL, the style is added to the list of styles available in CyNetShare’s Visual Style dropdown, and you can apply the style by selecting it in the list. CyNetShare’s initial display uses the visual style named “default” – use the Visual Style dropdown to choose the style in effect when Cytoscape generated the .cyjs and .json files.

Note that the mechanics of generating a URL for a file in a public directory system is a fast moving topic. Until recently, systems like Dropbox (and others) allowed users to create a URL that resolved directly to a file – a “direct” URL would be appropriate for use with CyNetShare. As of this writing, some public directory systems (e.g., Dropbox) generate “shareable” URLs instead, which resolve to a web page that allows file download – a “shareable” URL doesn’t work with CyNetShare. Systems that offer “shareable” URLs may offer “direct” URLs as part of their premium (or Pro) package. To tell if your public directory system generates a “direct” URL, have it generate a URL for a file, then paste the URL into the address field of a browser and observe whether the browser displays the file itself (good!) or a download page for the file (bad!).

Hint: if Dropbox generates a “shareable” link that looks like <https://www.dropbox.com/s/w5e7towcchuvdeu/cynetworm.cyjs?dl=0>, you may be able to create a “direct” link by changing the `dl=0` to `dl=1`: <https://www.dropbox.com/s/w5e7towcchuvdeu/cynetworm.cyjs?dl=1>.

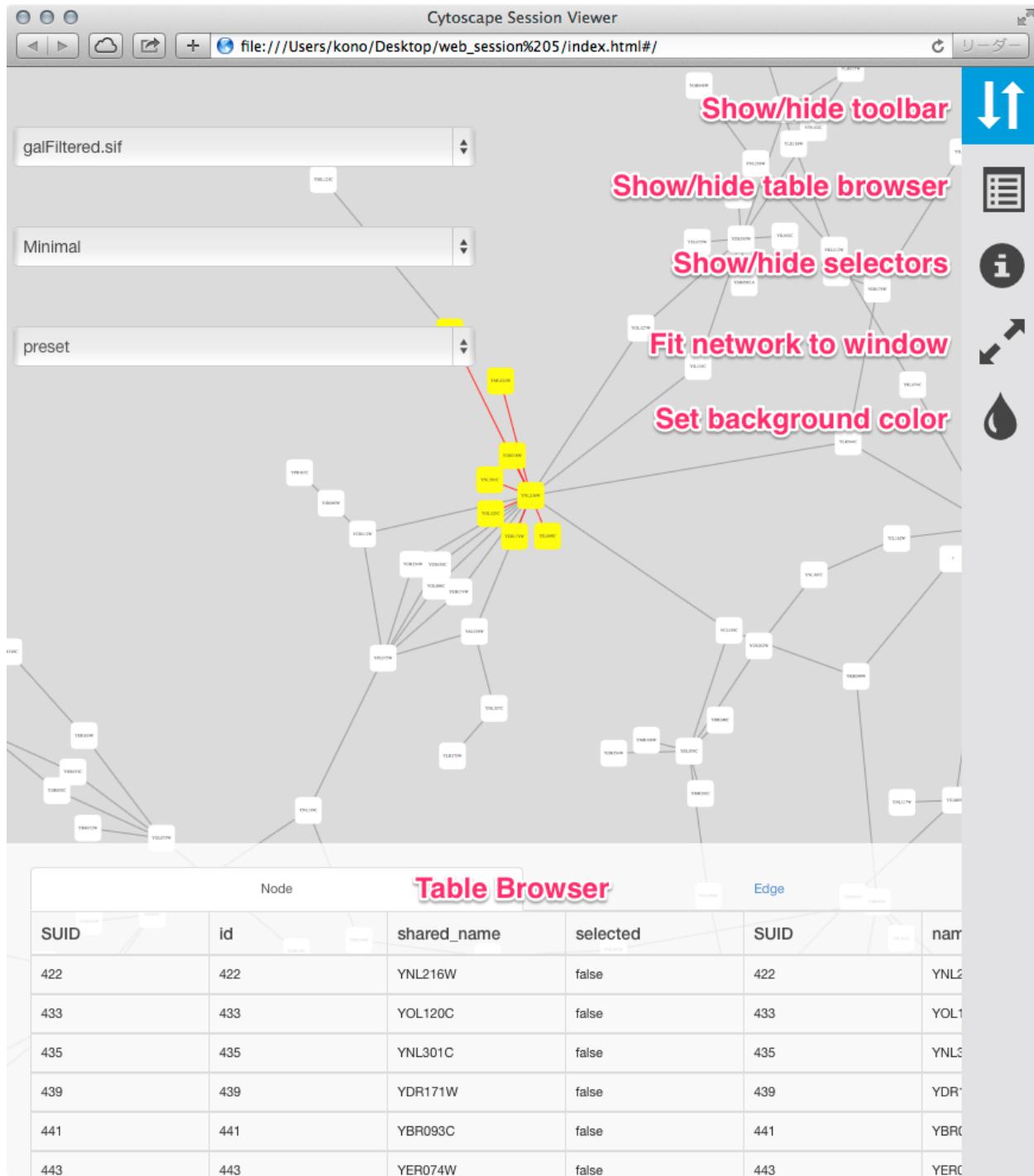
A simple strategy for always getting a “direct” URL is to store your file in a public directory served up by a web server, if you have access to one – a URL served by a web server might appear as: <http://myserver.com/~mypublicdir/netstyle.json>.

Alternately, you can use Gist to create a shareable document having a “direct” URL. To try this:

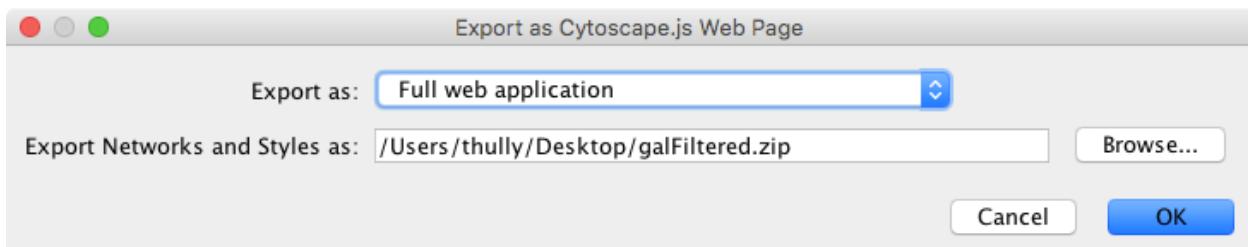
1. Use Cytoscape to generate your network as a .cyjs file on your local disk
2. Use an editor to open the file and copy its contents to the clipboard
3. Use a web browser to surf to Gist
4. Paste the contents into a Gist document
5. Select **Create public Gist**
6. Select **Raw** to place the “direct” URL into your browser’s address field
7. Use that URL with CyNetShare

Generating a Full Web Application

The full page export option is designed for users who want to publish their network as a complete single-page application. Cytoscape creates a zip archive containing a complete JavaScript-based web application that works as a basic viewer (like CyNetShare) for Cytoscape-generated network visualizations. You can unzip the archive onto a web server (or your PC) and view the network via a web browser on PCs and tablets.



To generate an entire web page for all networks in the session as a zip archive, select **File → Export as Web Page....**. The following dialog will appear:



This simple dialog has only two options - the file name and the type of web export (full, simple network view, or JSON). For full web application, we use the default. The file name is suggested based on the name of the selected table with the path defaulting to the current working directory. As in raw data export, the suggested file name can be changed using the Browse button or the text field. Press OK to export a web archive with the given file name.

To view the web page, unzip the archive into a folder on your PC or web server. The folder will contain an **index.html** file, the network data, and other files. You can open the **index.html** file in your browser (usually from your browser's **File → Open** menu item.)

Note: Depending on your browser's security settings, you may not be able to open the **index.html** file directly if it is stored on your PC. Attempting to do so may result in an empty browser window. To display the web page on your PC, you can install the CyWebrowser app, then view the **index.html** file inside of its browser window.

Alternately, you can start a local web server on your PC. An easy way to do this is by running the Python simple HTTP server. If you have Python installed on your machine, just go into the web archive folder and type:

```
python -m SimpleHTTPServer 8000
```

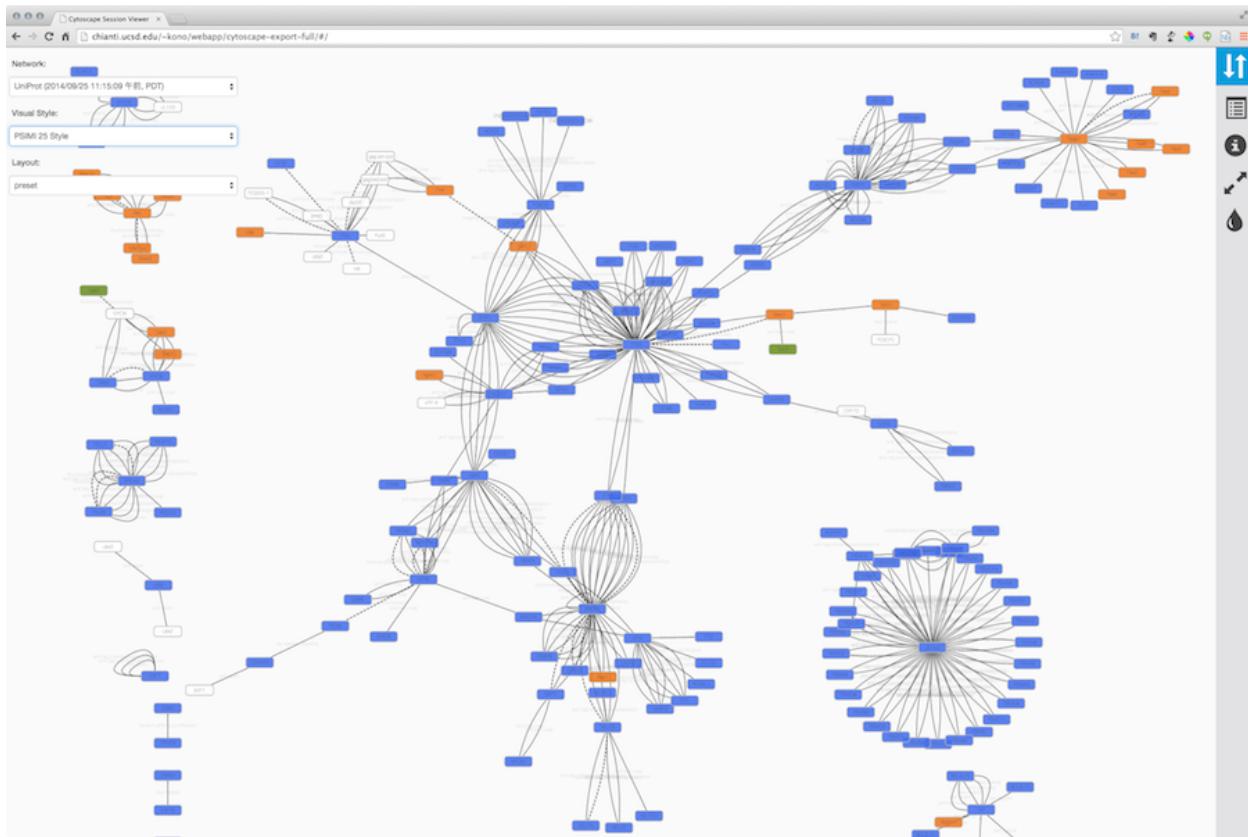
and use your browser to open:

```
http://localhost:8000/
```

Testing the archive on your PC will serve as an easy test of the web page, but to publish it to collaborators, you should unzip your archive onto a web server.

Here is an example exported file from Cytoscape:

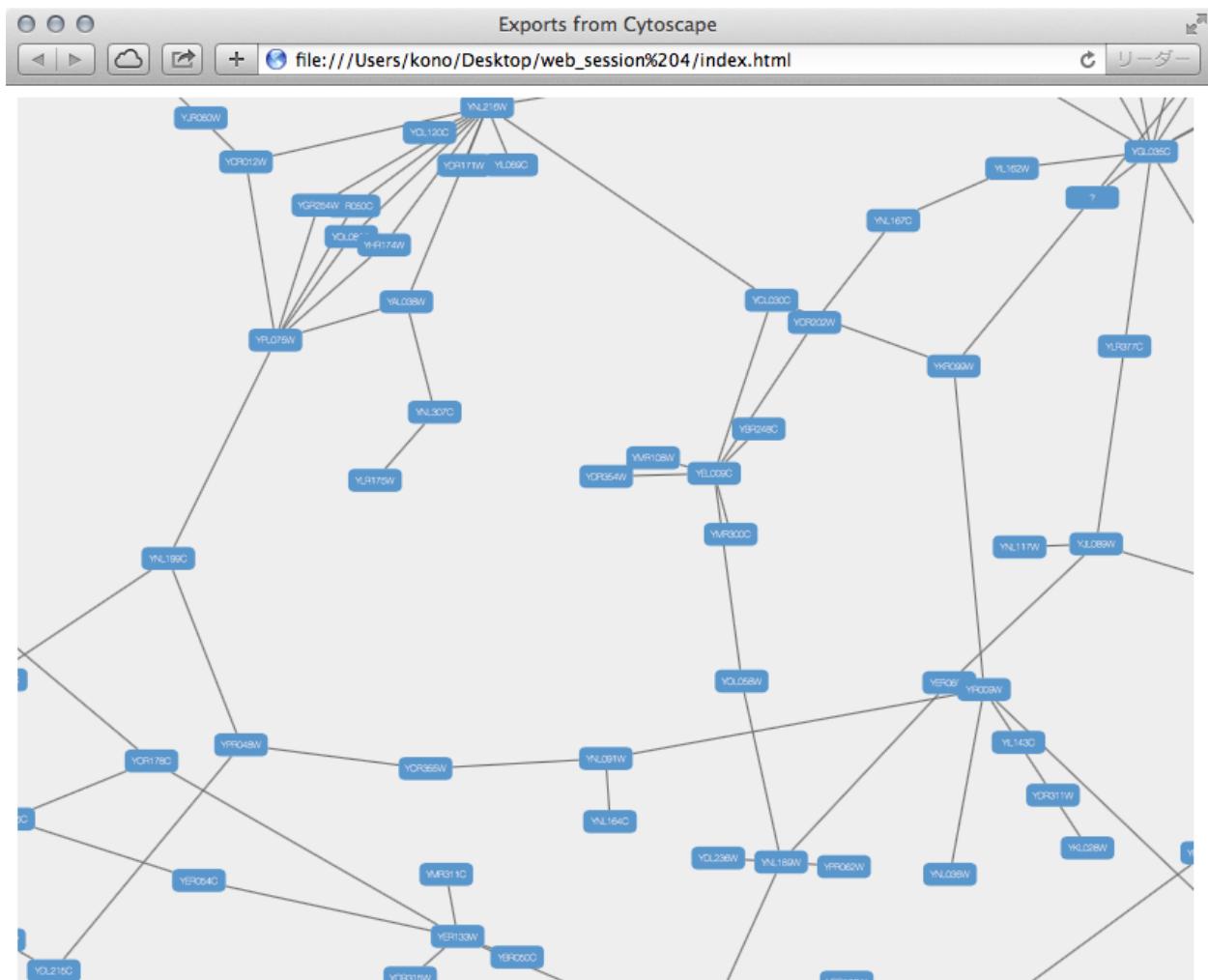
- Example full export deployed to web server
- Archive file



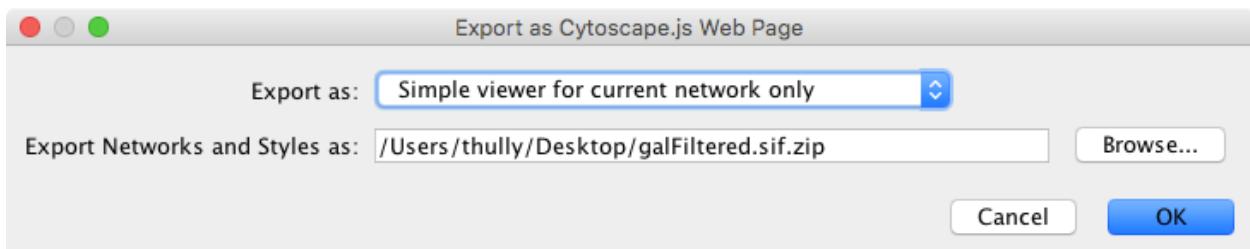
Note that because Cytoscape uses the latest HTML5-based web technologies, it cannot support older or non-conformant web browsers such as Internet Explorer. We strongly recommend using the latest version of modern web browsers such as Google Chrome, Mozilla Firefox, or Apple Safari.

Generating a Simple Network View (For Web Application Developers)

The Simple Network View export option is designed for users who want to publish their data as a complete [single-page application](#), but are interested in customizing the web viewer application themselves. Cytoscape creates a zip archive containing a partial [JavaScript](#)-based web application based on the `cytoscape.js` library and including simple “boilerplate” code and the current network view. The user can create a custom viewer by customizing this code.



To generate a web page for a single network view as a zip archive, select **File → Export as Web Page ...** as before, but this time choose the *Simple viewer for current network only* format as below:



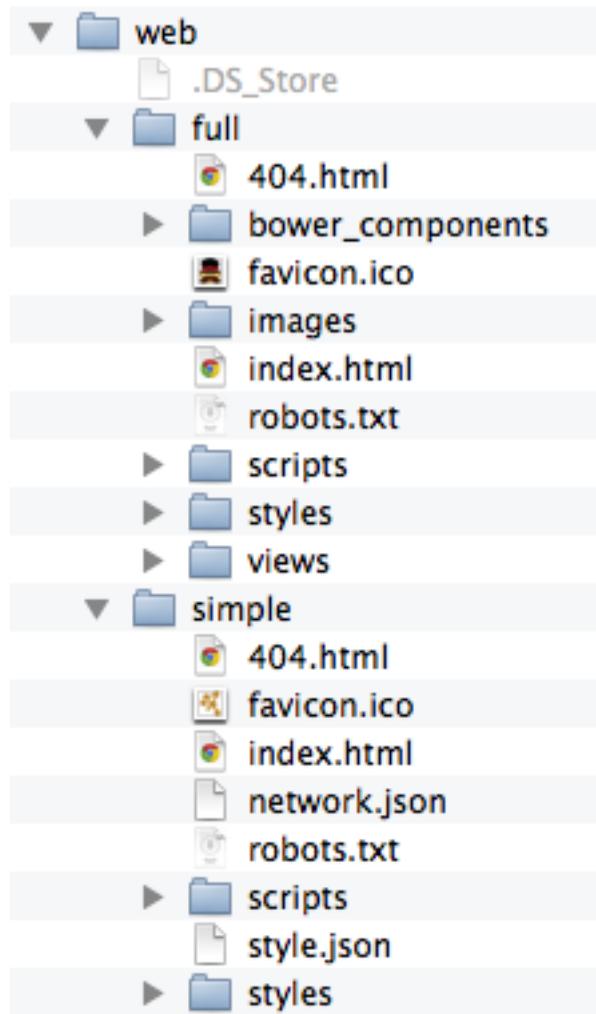
Everything else works as before, though this time the viewer will only include the current network. Press OK to proceed with the export.

For instructions on testing the customized web application, see *Generating a Full Web Application* above.

Customize Export Template (For Web Application Developers)

The code generated by Cytoscape for the Full Web Application and the Simple Network View web applications is minimalistic. While you can directly modify this code yourself to create your own page design or add new features, the modifications will apply to a single exported network. If you are a web application developer, you can change the

application code generated for **all** exports by editing HTML5 template code resource files in your `~/CytoscapeConfiguration/web` directory:



In this folder, you can find **full** and **simple** sub directories corresponding to Full Web Application and the Simple Network View described above.

Requirements

To build these project, you need the following tools installed:

- Node.js
- gulp
- grunt

Full Export Template

This is an AngularJS based web application built with grunt (at least for now – we have plans to migrate to gulp). Source code and more documentations are available here:

* <https://github.com/idekerlab/cyjs-full-export>

To build the project into *dist* directory, type:

```
grunt
```

Simple Export Template

This template is generated by a simple [gulp](#) project. The source code is available here:

- <https://github.com/idekerlab/cyjs-export-simple>

To build the project into *dist* directory, type:

```
gulp
```

Use Your Custom Templates for Export

Once you have your own builds, you can deploy your templates by replacing the contents of *full* and *simple* with your own builds.

CHAPTER 22

Cytoscape and OpenCL (Computing on the GPU)

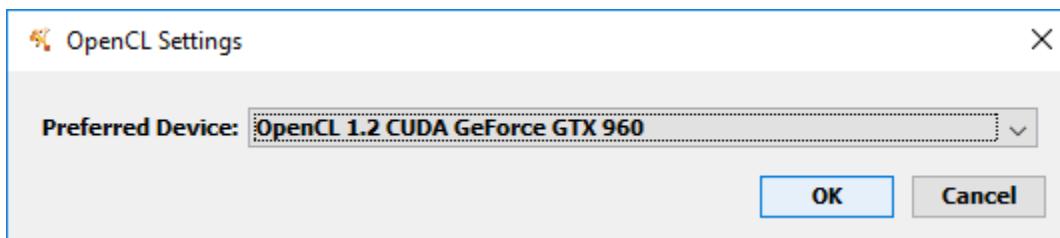
22.1 What it is for?

Cytoscape has basic support for offloading computationally intensive tasks for parallel processing on a GPU, multicore CPU or a multiprocessor card using [OpenCL](#). The OpenCL support in Cytoscape is mostly intended to provide a unifying OpenCL access for third-party Cytoscape Apps, but some core functionality (Prefuse layout) also has OpenCL implementation for increased performance.

As OpenCL can run on almost any device, including desktop computers without dedicated GPUs, you should be able to use OpenCL functionality on any platform where you can run Cytoscape, although you may need to install additional drivers for your device.

22.2 Setup & Configuration

The CyCL core app (should be already part of your Cytoscape distribution) lets you see the available OpenCL devices and configure your preferred device for OpenCL computation. To see the device list and configure your preferred device, select **Edit → Preferences → OpenCL Settings...**



In most cases, you want your preferred device to be your GPU (if you have one). If you do not see the device you intend to use (or any device at all) in the configuration window, you may need to install OpenCL drivers. For GPUs from major manufacturers, OpenCL drivers are usually installed with the graphics driver. If you have installed the latest graphics drivers and still do not see your device in the list, check the website of your GPU manufacturer. To use

OpenCL on a CPU (i.e., on any desktop computer) or a Xeon Phi, you need to install the drivers separately from the web page of the manufacturer of your CPU.

As of 2017-01-06, drivers for Intel CPUs can be downloaded at https://software.intel.com/en-us/articles/opencl-drivers#latest_CPU_runtime - look for “runtime-only”. AMD’s drivers can be found at <http://support.amd.com/en-us/kb-articles/Pages/OpenCL2-Driver.aspx>.

Warning: After you have just installed drivers for your OpenCL device, Cytoscape may require restarting to be able to “see” the device.

22.3 OpenCL/GPU Troubleshooting

If any OpenCL app is not working, here are some of the common causes:

22.3.1 CyCL core app is not installed or not initialized properly

Diagnosis: check if you can see the **Edit → Preferences → OpenCL Settings...** menu item. If not, this is a bad sign.

Treatment: Check the CytoscapeConfiguration directory, if it contains a file called `disable-opencl.dummy` remove it and start Cytoscape from the system console (using `cytoscape.bat` or `cytoscape.sh`). Check console output for additional information.

22.3.2 No OpenCL devices are available

Diagnosis and treatment: See *Setup and configuration* above.

22.3.3 The Cytoscape console says “Could not find OpenCL.dll”

Treatment: Make sure your Windows OS directory containing OpenCL.dll is listed in the value of the Java system property `java.library.path` (OpenCL.dll can usually be found in `%windir%\SysWOW64%` or `%windir%\system32`). You can modify `java.library.path` by adding a line of the form `-Djava.library.path=path1;path2` to the `Cytoscape.vmoptions` file. `Cytoscape.vmoptions` can be found in the Cytoscape installation directory.

22.3.4 The computer is unresponsive for a while and then resets the screen while running OpenCL on the GPU.

Treatment: This problem arises if you run computations on the GPU that also handles your main display and the computation occupies the GPU for too long, preventing the operating system (e.g., Windows) from updating the display. If the operating system detects this, it terminates the GPU computation. You can either try to run smaller chunks of the computation or configure the “Timeout Detection and Recovery” (TDR) to let the computation complete even if the system becomes unresponsive. Further info can be found at [this NVidia thread](#) or [Microsoft webpage](#). Microsoft also has [documentation for the registry keys that need to changed](#).

22.3.5 My problem is not listed here

Try to install drivers for your CPU and run on the CPU platform (see links above). CPU platforms offer lower performance, but are generally more reliable. If this does not help you, ask for help in the [Cytoscape helpdesk mailing list](#).

Cytoscape.js and Cytoscape

23.1 What is Cytoscape.js?

The screenshot shows the Cytoscape.js homepage on a web browser. The URL in the address bar is `cytoscape.github.io/cytoscape.js/`. The page features a large title "Cytoscape.js 2.0.3". To the right of the title is a brief description: "A JavaScript graph library for analysis and visualisation (compatible with CommonJS/Node.js, jQuery 1.4+, and plain JavaScript)". Below the title are links to GitHub, Downloads, Live code examples, Twitter, Previous questions, Ask a question, File an issue, and Future version milestones. On the left, there is a sidebar with a "Table of contents" and several sections: Introduction (About, Cytoscape.js & Cytoscape, Funding), Downloads (2.0), Roadmap, Notation (Architecture & API, Notation, Position, Elements JSON), Selectors (Notes & caveats, Group, class, & ID, Data, Compound nodes, State), and Style (Format, Properties, Mappers). At the bottom of the sidebar is a link to stackoverflow.com/questions/ask?tags=cytoscape.js. In the main content area, there is a section titled "This is how easy it is to get started with Cytoscape.js (this code creates the instance you see on the bottom-right)". It includes a snippet of JavaScript code:

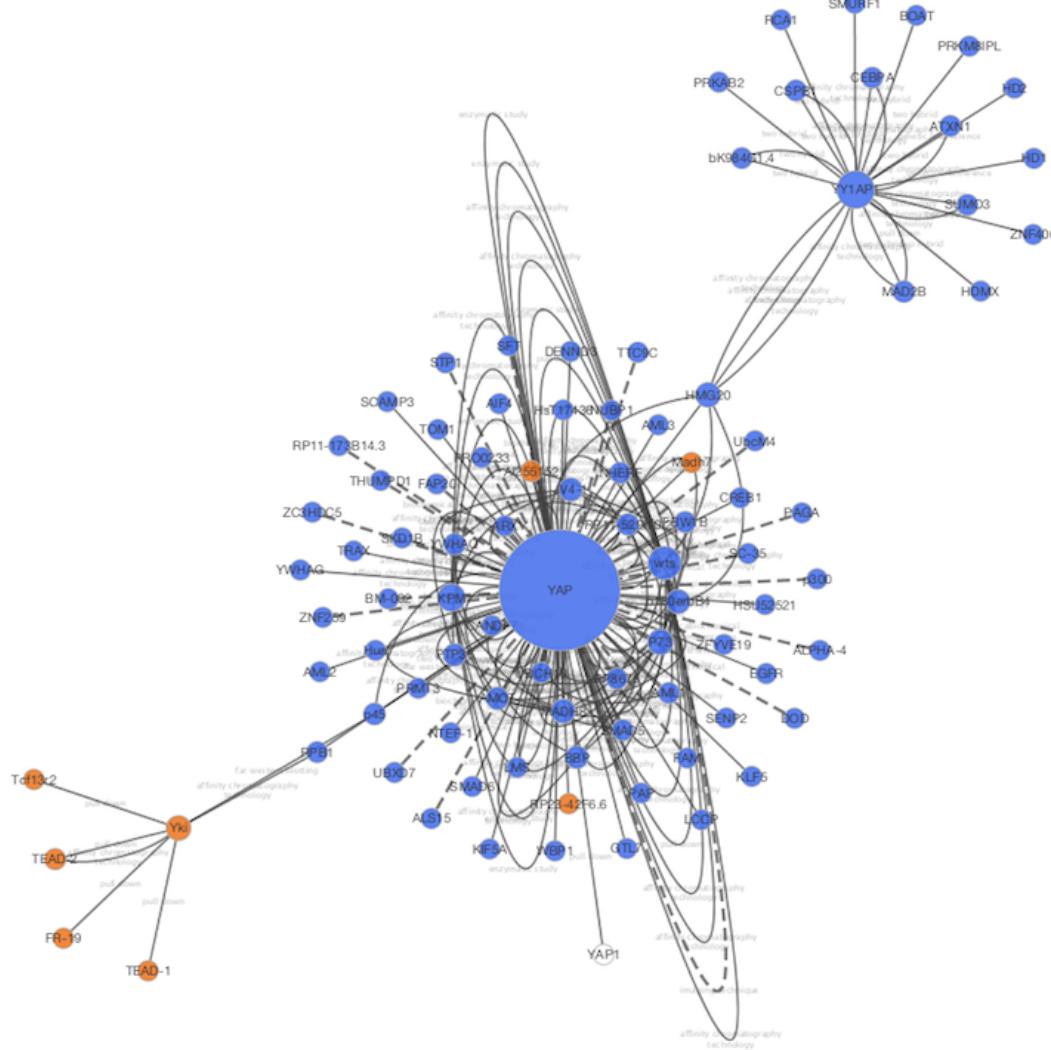
```
$loadCy = function(){
    options = {
        showOverlay: false,
        minZoom: 0.5,
        maxZoom: 2,
        style: cytoscape.stylesheet()
            .selector('node')
            .css({
                'content': 'data(name)',
                'font-family': 'helvetica',
                'font-size': 14,
                'text-outline-width': 3
            })
    };
}
```

Below the code is a button labeled "Show all the code...". To the right of the code is a network visualization showing four nodes: Jerry, Kramer, Elaine, and George. Jerry has bidirectional edges to Kramer and Elaine. Kramer has a bidirectional edge to Jerry and a unidirectional edge to George. Elaine has a bidirectional edge to Jerry and a unidirectional edge to George.

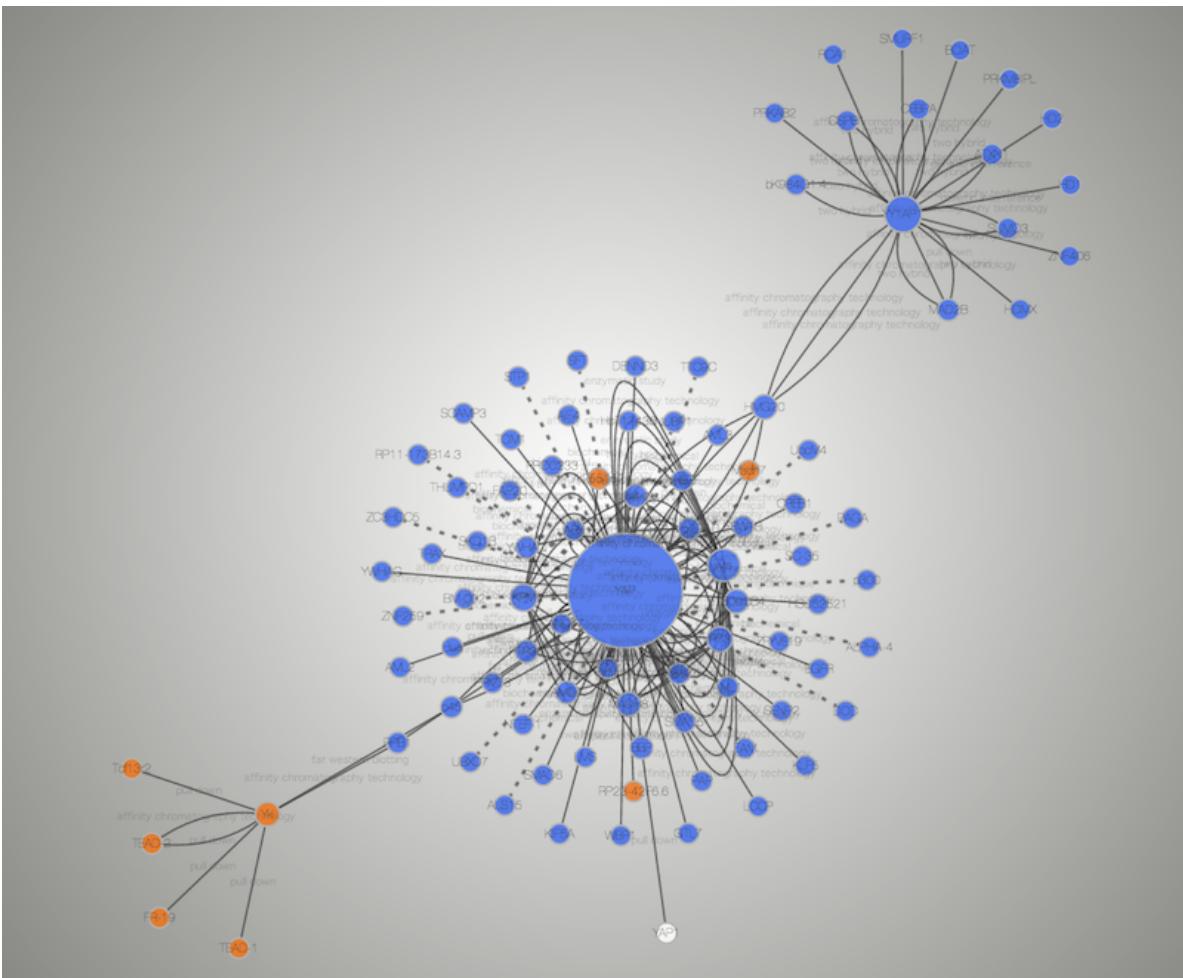
Cytoscape.js is a **JavaScript library** for interactive network visualization. It is a building block for web applications and is **NOT** a complete web application. If you have network data sets and want to share visualizations created with Cytoscape, you can build your own website using Cytoscape.js and this new **Export to Cytoscape.js** feature.

23.1.1 Examples

- A network visualized with Cytoscape 3.1.0



- Same network exported to Cytoscape.js (Rendered with Google Chrome and Cytoscape.js 2.0.3)



- Interactive example (galFiltered.sif rendered with Cytoscape.js 2.0.3)

23.1.2 Differences and Shared Concepts

Although Cytoscape and Cytoscape.js are two completely independent software packages, they are sharing higher level concepts. The following is the list of similarities and differences between the two:

Cytoscape

- **Desktop application** for network visualization written in [Java programming language](#)
- Needs desktop or laptop computers to run
- Users have to install Java runtime
- High performance application for large scale network analysis and visualization
- Expandable by [Apps](#)
- Use **Styles** to map data to network properties, such as node color, edge width, node shape, etc.

Cytoscape.js

- A [JavaScript library](#) for network visualization, NOT a complete web application nor mobile app
- Runs on most of modern web browsers, including tablets and smart phones
- No plugins are required to run. Modern web browser is the only requirement
- Need to write code to set up your web site or web application
- Expandable by [Extensions](#)
- Use **CSS-based Styles** to map data to network properties

In a long term, Cytoscape and Cytoscape.js will be more integrated, and as the first step Cytoscape now supports reading and writing Cytoscape.js network/table JSON files. In addition, Cytoscape can convert *Styles* to Cytoscape.js Style object.

23.2 Data Exchange between Cytoscape and Cytoscape.js

Since Cytoscape.js is a JavaScript library, its basic data exchange format is [JSON \(JavaScript Object Notation\)](#). All of these import/export functions are part of standard Cytoscape user interface, and you can read/write Cytoscape.js JSON files just like other standard files such as SIF.

23.2.1 Export Network and Table to Cytoscape.js

Cytoscape.js stores network data (graph) and its data table in the same object. Cytoscape writes such data complex as JSON, i.e., both network and data tables will be exported as a single JSON file. You can select a network and export it from **File | Export | Network**.

Cytoscape only supports one of the Cytoscape.js supported JSON formats, which is:

```
{  
  elements:{  
    nodes:[],  
    edges:  
  }  
}
```

SUID will be used as unique identifier for nodes and edges in the JSON. For more information about this format, please visit [Cytoscape.js web site](#).

Important Note about Data Compatibility

Cytoscape creates JSON file directly from data table and tries to extract as much data as possible from the original table. However, since table column names will be directly converted into [JavaScript](#) variable names, invalid characters will be replaced by underscore (_):

- Original Data Table Column Names:

Gene Name
KEGG.pathway

- The Names above will be replaced to:

Gene_Name
KEGG_pathway

You should be careful when you plan to use this feature for data roundtrip: from Cytoscape to Cytoscape.js back to Cytoscape. For such use cases, **we strongly recommend to use JavaScript-safe characters only in your table column names**. Naming your columns only with alphanumeric characters and underscore (_) is the best practice. (For actual data entries, all characters are allowed. This restriction is only for table column names.)

23.3 Export Styles to Cytoscape.js

Cytoscape and Cytoscape.js are sharing a concept called **Style**. This is a collection of mappings from data point to network property. oCytoscape can export its Styles into CSS-based Cytoscape.js JSON.

You can export *all Styles into one JSON file* from **File | Export | Styles** and select Cytoscape.js JSON as its format.

23.3.1 Limitations

Cytoscape.js does not support all of Cytoscape Network Properties. Those properties will be ignored or simplified when you export to JSON Style file.

Currently, the following Visual Properties will not be exported to Cytoscape.js JSON:

- Custom Graphics and its positions
- Edge Bends
- Tooltips
- Node Label Width
- Node Border Line Type
- Arrow Colors (they are always same as edge color)

23.3.2 Cytoscape.js and Cytoscape Compatibility

Cytoscape's network rendering system is designed for desktop use, while the browser-based renderer incorporates web technologies (e.g., cytoscape.js and Cascaded Style Sheets). As a result, most but not all networks will render the same in the browser as in Cytoscape. Cytoscape visual styles not supported in the web browser are ignored. A complete compatibility list is available [here](#).

23.3.3 Import Cytoscape.js data into Cytoscape

Cytoscape.js network JSON files can be loaded from standard Cytoscape file menu: **File | Import | Network ...**. Both File and URL are supported.

23.4 Build Your Own Web Application with Cytoscape.js

Although Cytoscape can export networks, tables, and Style as Cytoscape.js-compatible JSON, users have to write some **JavaScript** code to visualize the data files with Cytoscape.js. Details of web application development with Cytoscape.js is beyond the scope of this document. If you need examples and tutorials about web application development with Cytoscape.js, please visit the following web site:

- <https://github.com/cytoscape/cyjs-sample/wiki>

23.4.1 Questions?

If you have questions and comments about web application development with Cytoscape and Cytoscape.js, please send yours to our [mailing list](#).

CHAPTER 24

Cytoscape Automation

Cytoscape Automation is a collection of features that enable users to create workflows executed entirely within Cytoscape or by external tools (e.g., RStudio, Jupyter, GenomeSpace, etc), and whose results are reproducible. This enables Cytoscape to scale to large collections of datasets and to larger more complex workflows than is practical via keyboard and mouse.

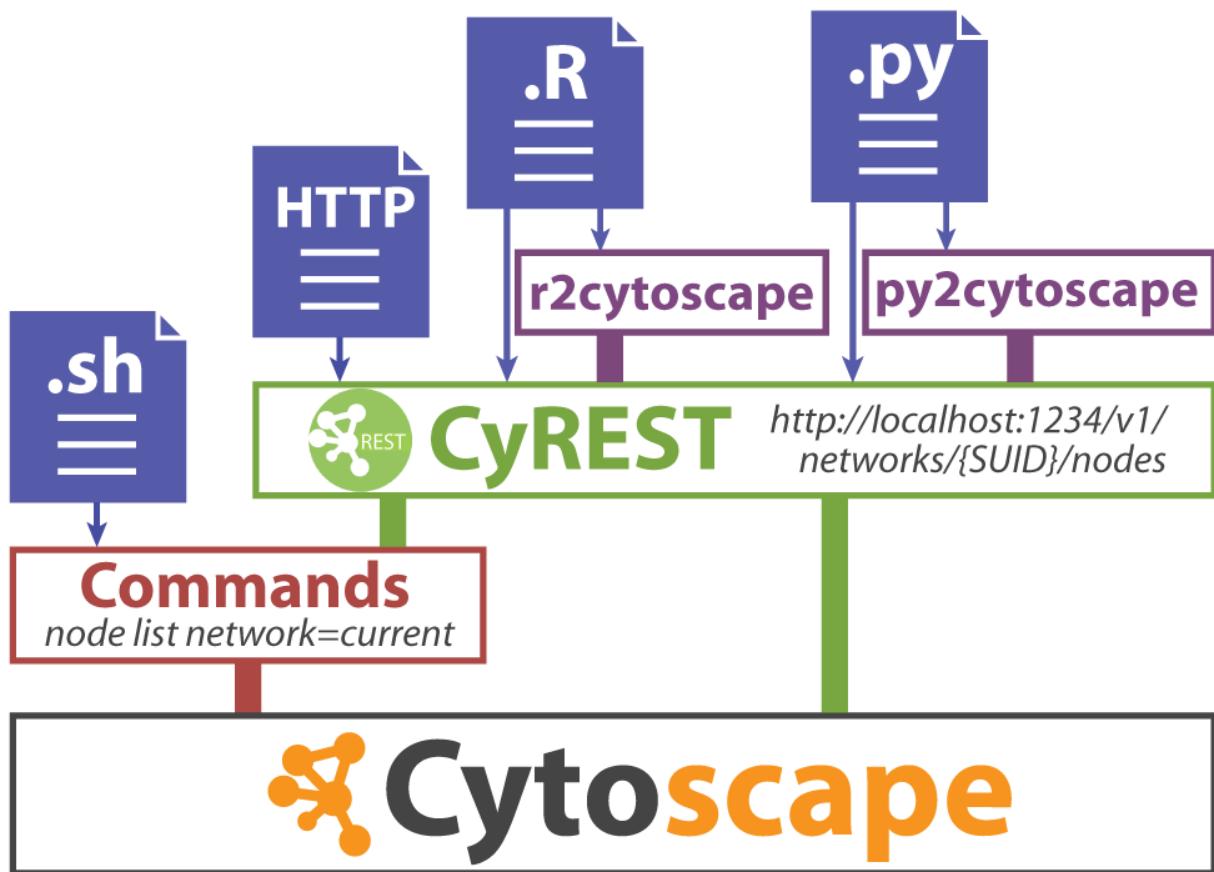
Cytoscape Automation exists in two skins – the Commands interface and the Functions interface. Both can accomplish similar results, but are focused on different usage styles. Commands reprise user-initiated interactions (e.g., open session, import data, export image), whereas Functions enable programmers to manipulate and operate on networks as internal Cytoscape data. Commands and Functions both call Cytoscape (and Cytoscape apps) via a REST interface known as CyREST.

This chapter describes how to produce custom workflows using CyREST natively and via Python and R interface libraries. More tutorials and examples are available at the [Cytoscape Automation web page](https://github.com/cytoscape/cytoscape-automation/wiki) (<https://github.com/cytoscape/cytoscape-automation/wiki>).

Note that for a Cytoscape app to be callable via CyREST, the app must be installed into Cytoscape and the author must have specifically added automation functionality to it. If there is an app you would like to call, but which doesn't offer automation, please contact the app author and request that app functionality be added. Instructions for adding automation to an app are available on the [Cytoscape Automation web page](https://github.com/cytoscape/cytoscape-automation/wiki) (<https://github.com/cytoscape/cytoscape-automation/wiki>).

24.1 Background

Cytoscape's intuitive graphical user interface is useful for *interactive* network data integration, analysis, and visualization. It provides great features for exploratory data analysis, but what happens if you have hundreds of data files or need to ask someone to execute your data analysis workflows? It is virtually impossible to apply the same operations to hundreds of networks manually using a GUI. More importantly, although you can save your **results** as session files, you cannot save your **workflows** if you perform your data analysis with point-and-click GUI operations. Cytoscape has several options that support automating your workflows, all under the umbrella of Cytoscape Automation:

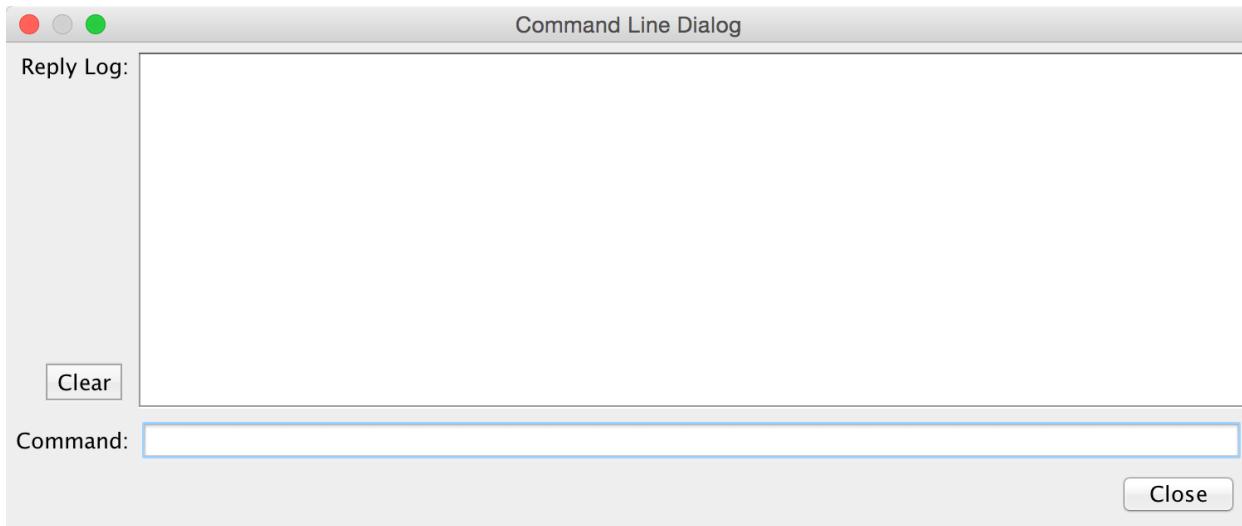


The **Commands** feature allows you to script a sequence of Cytoscape commands and menu items, where commands can have parameter values that would normally be provided by a user via a Cytoscape dialog box. For example, `session open file="C:\myfile.cys"` loads a session from a file similarly to the **File | Open** menu item. Commands may resolve to Cytoscape core functions or automation-enabled apps installed in Cytoscape. You can create a command script file that Cytoscape can execute via the **Tools | Execute Command File** menu item or on the Cytoscape command line at startup.

The **CyREST API** feature allows you to access Cytoscape from a separate application, thereby orchestrating Cytoscape operations via HTTP-based REST calls. Your workflows can execute either Commands or Functions in this way.

Automation applications may be written in a general programming language that keeps its own data structures, performs complex flow control, or directly manipulates Cytoscape nodes, edges, attributes, and visual styles. For R and Python, we provide language-specific interface libraries (e.g., r2cytoscape and py2cytoscape) that present Cytoscape Automation in language-friendly terms, and call Cytoscape and apps via the CyREST interface.

24.2 Commands

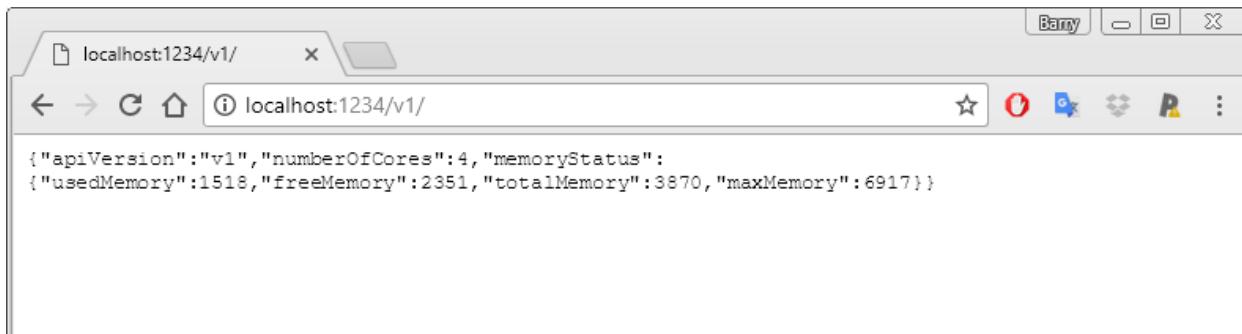


Commands is the built-in Cytoscape feature to automate your workflow as simple script. You can learn more about Commands in the [Command Tool](#) section.

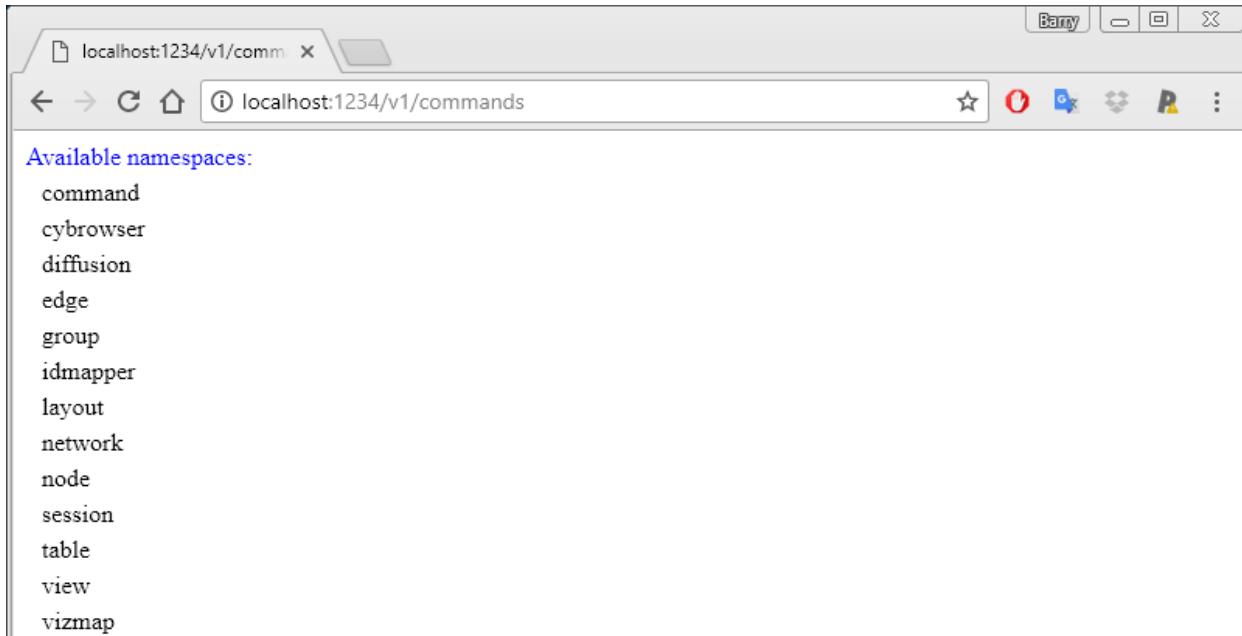
24.3 CyREST Interface Layer

In some cases, you may need to leverage the complex control and data structures available in a fully featured programming language, such as [R](#), [Python](#), [Ruby](#) or [JavaScript](#). Such languages enable complex Cytoscape-centric workflows or the integration of Cytoscape into larger workflows. Cytoscape enables programmatic access to both its Commands and Functions via its CyREST interface.

By default, the CyREST interface is enabled and available on TCP/IP port 1234. To verify this, start a web browser on your Cytoscape workstation and surf to either <http://localhost:1234/v1/> or <http://localhost:1234/v1/commands>. The first form is a Function that returns basic Cytoscape information as a JSON object:



The second form is a Command that returns a list of available Command namespaces:



Note that the list of namespaces will vary depending on the apps you install – some apps provide Commands in namespaces of their own.

If your workstation has port 1234 already in use, you can specify a different CyREST port in two ways:

- Use the **Edit → Preferences** menu to alter the *rest.port* property to select a different port (e.g., 8888)
- Specify the port on the Cytoscape command line via the *-R* parameter:

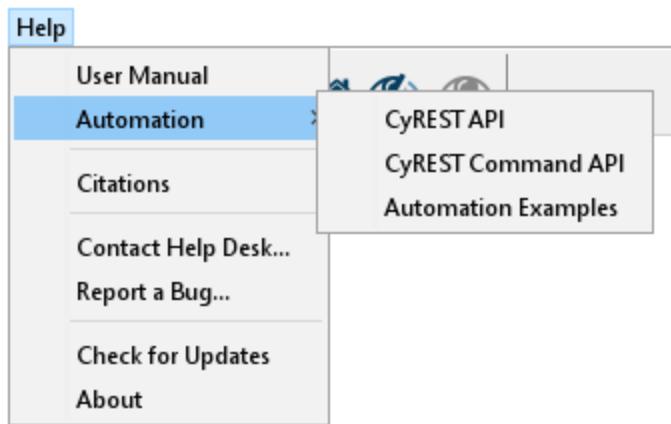
```
cytoscape.bat -R 8888 (for Windows)  
./cytoscape.sh -R 8888 (for Mac or Linux)
```

You can test the new port by using your browser to surf to <http://localhost:8888/v1/>

Note that if you expect to run more than one instance of Cytoscape on a single workstation, the CyREST port must be unique for each Cytoscape instance. You must use either the property or command line parameter technique to execute each instance with a different CyREST port.

24.3.1 Exploring CyREST Commands and Functions

Cytoscape makes a list of available Commands and Functions available via the **Help → Automation** submenus. The **CyREST API** submenu shows available Functions, and **CyREST Command API** shows available Commands. **Automation Examples** leads to a web portal containing the bulk of documentation, samples and tutorials for Cytoscape Automation in general.



Notably, **Automation Examples** contains a tutorial that explains how to explore CyREST Commands and Functions: <https://github.com/cytoscape/cytoscape-automation/wiki/Trying-Automation>.

The **CyREST API** and **CyREST Command API** submenus document CyREST in a Swagger web application, which allows you to explore CyREST entrypoints by reading about them and invoking them directly (using a *Try it out!*) button. A sample page matching the `http://localhost:1234/v1/` Function is:

```

Cytoscape System Show/Hide | List Operations | Expand Operations
GET /v1 Cytoscape RESTful API server status

Implementation Notes
500 If REST API Module is not working

Response Class (Status 200)
successful operation

Model Example Value
ServerStatus {
  apiVersion (string, optional),
  numberOfCores {integer, optional},
  memoryStatus {MemoryStatus, optional}
}
MemoryStatus {
  usedMemory (integer, optional),
  freeMemory (integer, optional),
  totalMemory (integer, optional),
  maxMemory (integer, optional)
}

Response Content Type application/json;charset=utf-8 ▾
Try it out!

```

Note that the page contains a description, input parameters, output values, result code and the *Try it out!* button.

Pressing the button executes the Function and shows the actual CyREST call and its results. **Making good use of Swagger as a prototyping tool can greatly reduce code writing and debug time!**

The screenshot shows the Swagger UI interface for a Cytoscape Automation API. The browser title bar says "Trying Automation · cyto" and "Swagger UI". The address bar shows "localhost:1234/v1/swaggerUI/swagger-ui/index.html?url=http%3A%2F...".

Response Content Type: application/json;charset=utf-8 ▾

Try it out! [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' 'http://localhost:1234/v1'
```

Request URL:

```
http://localhost:1234/v1
```

Response Body:

```
{
  "apiVersion": "v1",
  "numberOfCores": 4,
  "memoryStatus": {
    "usedMemory": 926,
    "freeMemory": 2912,
    "totalMemory": 3839,
    "maxMemory": 6917
  }
}
```

Response Code:

```
200
```

Response Headers:

```
{
  "access-control-allow-origin": "*",
  "access-control-allow-credentials": "true",
  "server": "Jetty(8.1.15.v20140411)",
  "access-control-allow-headers": "origin, content-type, accept, authorization",
  "content-length": "125",
  "access-control-allow-methods": "GET, POST, PUT, DELETE, OPTIONS, HEAD",
  "content-type": "application/json;charset=utf-8"
}
```

To find out more about how to use CyREST, visit the Cytoscape Automation tutorial pages (<https://github.com/cytoscape/cytoscape-automation/wiki/Trying-Automation>).

Note that Swagger pages reflect functionality in Cytoscape at the time it executes, including Commands and Functions contributed by installed apps. To discover CyREST functionality in uninstalled apps, you must first install them, and then use **Help → Automation** submenus to generate the Swagger pages (or simply reload your Swagger browser page).

24.4 cyREST and R/Python

CyREST enables access to Cytoscape Automation features from workflows written in high level languages (e.g., [R](#) and [Python](#)) and executing in separate environments (e.g., [RStudio](#) and [Jupyter](#)) on the Cytoscape workstation. A workflow can call Commands and Functions either directly via CyREST or via language-specific interface libraries.

To directly call CyREST Commands and Functions from a workflow, see the [mini-tutorial](#).

To understand and get started using language-specific libraries, visit the [Cytoscape Automation web page](#) (<https://github.com/cytoscape/cytoscape-automation/wiki>). The [Workflow Index page](#) contains useful code samples, and the [Sample Scripts page](#) contains working demonstration projects.

Below is a sample Jupyter-based Python script and the resulting sample output.

nbviewer
FAQ
IPython
Jupyter

```

# You can set default values as key-value pairs.

'NODE_FILL_COLOR': '#6AACB8',
'NODE_SIZE': 55,
'NODE_BORDER_WIDTH': 0,
'NODE_LABEL_COLOR': '#555555',

'EDGE_WIDTH': 2,
'EDGE_TRANSPARENCY': 100,
'EDGE_STROKE_UNSELECTED_PAINT': '#333333',

'NETWORK_BACKGROUND_PAINT': '#FFFFEA'
}

my_yeast_style.update_defaults(basic_settings)

# Create some mappings
my_yeast_style.create_passthrough_mapping(column='label', vp='NODE_LABEL', col_type='String')

degrees = yeast_net.get_node_column('Degree')
color_gradient = StyleUtil.create_2_color_gradient(min=degrees.min(), max=degrees.max(), colors=('white', '#6AACB8'))
degree_to_size = StyleUtil.create_slope(min=degrees.min(), max=degrees.max(), values=(10, 100))
my_yeast_style.create_continuous_mapping(column='Degree', vp='NODE_FILL_COLOR', col_type='Integer', points=color_gradient)
my_yeast_style.create_continuous_mapping(column='Degree', vp='NODE_SIZE', col_type='Integer', points=degree_to_size)
my_yeast_style.create_continuous_mapping(column='Degree', vp='NODE_LABEL_FONT_SIZE', col_type='Integer', points=degree_to_size)

cy.style.apply(my_yeast_style, yeast_net)

# Step 7: (Optional) Embed as interactive Cytoscape.js widget
yeast_net_view = yeast_net.get_first_view()
style_for_widget = cy.style.get(my_yeast_style.get_name(), data_format='cytoscapejs')
renderer.render(yeast_net_view, style=style_for_widget['style'], background='radial-gradient(#FFFFFF 15%, #DDDDDD 105%)')

```

(Sample Jupyter Notebook written with cyREST and py2cytoscape)

CHAPTER 25

Cytoscape Privacy Policy

We respect the privacy of all Cytoscape users, and we do not collect any information on Cytoscape users except in the situations listed below. In no case do we attempt to tie any of this information back to a user, nor do we give, share, sell, or transfer this information to any third party unless required by law. We use this information only in the aggregate to generate statistics to assist in securing continued funding for Cytoscape.

- On the Cytoscape download web page, we log the date, time, browser signature, and IP address to which we deliver Cytoscape.
- When Cytoscape starts, it fetches news and version information from a Cytoscape server. We log the date, time, and IP address of the workstation running Cytoscape.
- During Cytoscape execution, we log usage statistics tagged with the date, time, and IP address of the workstation running Cytoscape. This logging can be turned off during Cytoscape installation or by setting the installations.shareStatistics property to false using the Cytoscape Preferences Editor.

This policy may change from time to time, and if it does, we will notify you via our normal mass notification media. We will also update this section of the user manual.

Note that some internal Cytoscape Apps and Apps available through the Cytoscape App Store connect with third party services via the Internet. Once an App links to such a service, you are subject to the privacy policy of that service.

CHAPTER 26

Basic Expression Analysis Tutorial

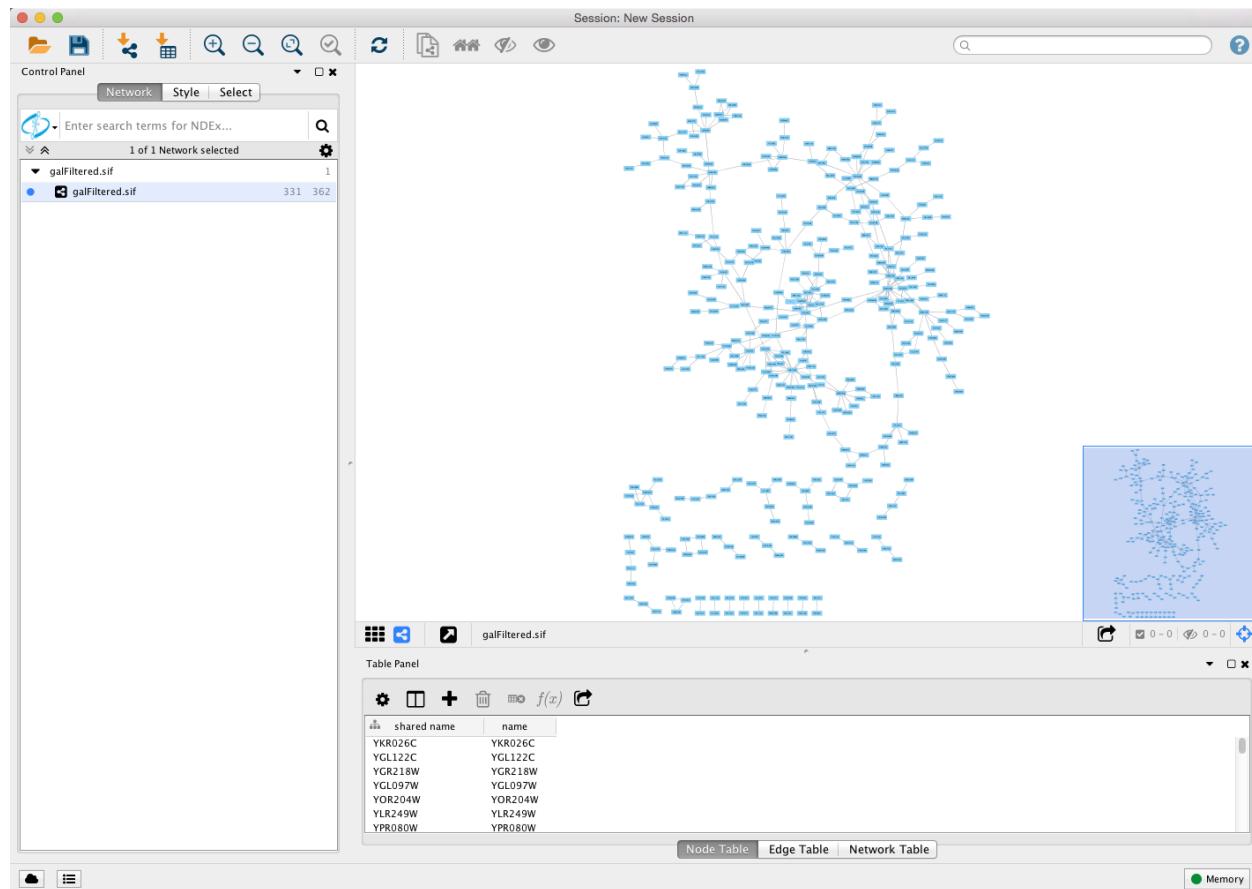
Cytoscape is an open source software platform for integrating, visualizing, and analyzing measurement data in the context of networks.

This tutorial presents one scenario of how expression data can be combined with network data to tell a biological story and includes the following concepts:

- Visualizing networks using expression data.
- Filtering networks based on expression data.
- Assessing expression data in the context of a biological network.

26.1 Loading Network

- Start Cytoscape and load the network galFiltered.sif using **File → Import → Network->File....**
- By default, the “Prefuse Force Directed Layout” is applied to organize the layout of the nodes. The default layout may be changed in the **Layout → Settings** menu.
- Cytoscape should now look similar to this:



26.2 Loading Expression Data

To begin with, here is some background on your data. You are working with yeast, and the genes Gal1, Gal4, and Gal80 are all yeast transcription factors. Your expression experiments all involve some perturbation of these transcription factor genes. Gal1, Gal4, and Gal80 are also represented in your interaction network, where they are labeled according to yeast locus tags: Gal1 corresponds to YBR020W, Gal4 to YPL248C, and Gal80 to YML051W.

Using your favorite text editor, open the file galExpData.csv. The first few lines of the file are as follows:

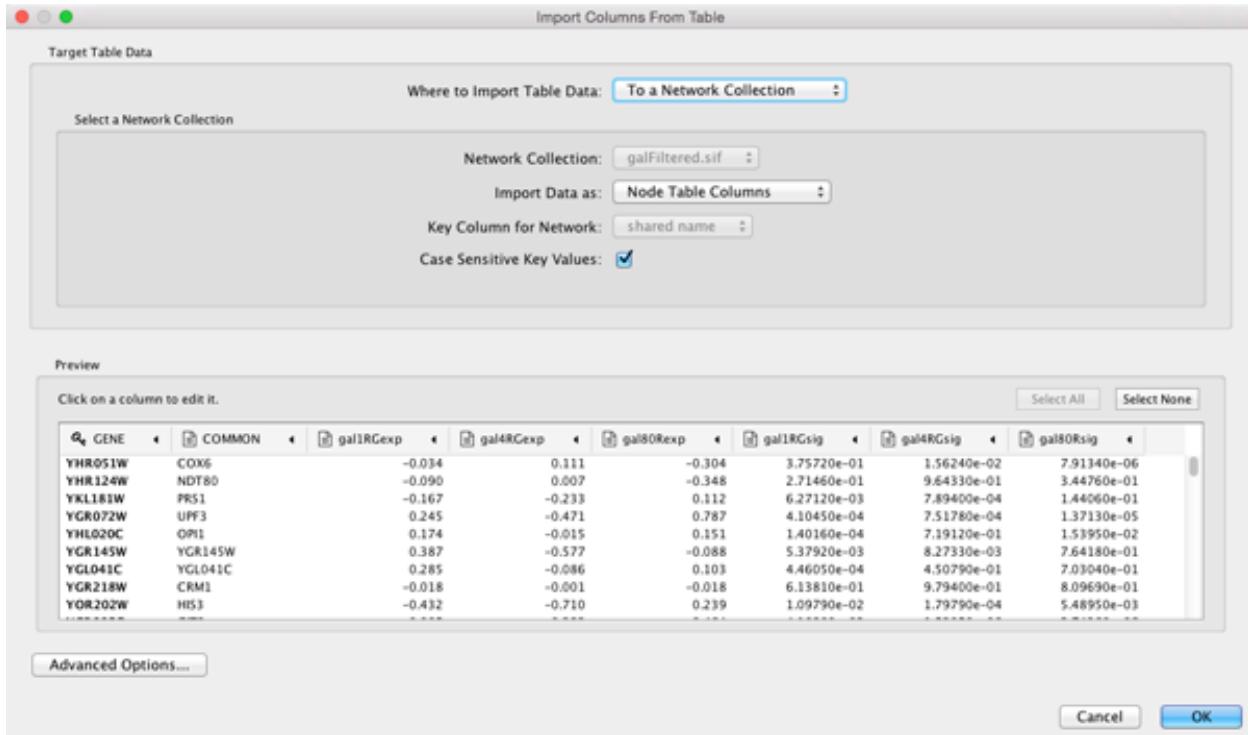
shared name	name
YKR026C	YKR026C
YGL122C	YGL122C
YGR218W	YGR218W
YGL097W	YGL097W
YOR204W	YOR204W
YLR249W	YLR249W
YPR080W	YPR080W

You should note the following information about the file:

1. The first line consists of labels.
2. All columns are separated by a single comma character.
3. The first column contains node names, and must match the names of the nodes in your network exactly!
4. The second column contains common locus names. This column is optional, and the data is not currently used by Cytoscape, but including this column makes the format consistent with the output of many microarray analysis packages, and makes the file easier to read.

5. The remaining columns contain experimental data, two columns per experiment (one column represents the expression measurement and the second represents the significance value for that measurement), and one line per node. In this case, there are three expression results per node.

- Under the **File** menu, select **Import** → **Table** → **File....**
- Select the file galExpData.csv.
- The default settings are usually correct, although you may need to change the Key column to indicate which column will be used to match with the network key column.
- Click the **OK** button to import the data.



Now we will use the **Node Table** to browse through the expression data, as follows:

- Select a node on the Cytoscape canvas by clicking on it.
- In the **Node Table**, You can limit the columns shown by going to the **Show Columns** button and select the attributes gal1RGexp, gal4RGexp, and gal80Rexp by clicking on the checkbox.
- Under the **Node Table**, you should see your node listed with their expression values, as shown.



26.3 Visualizing Expression Data

Probably the most common use of expression data in Cytoscape is to set the visual attributes of the nodes in a network according to expression data. This creates a powerful visualization, portraying functional relation and experimental response at the same time. Here, we will walk through the steps for doing this.

26.3.1 Label the Nodes

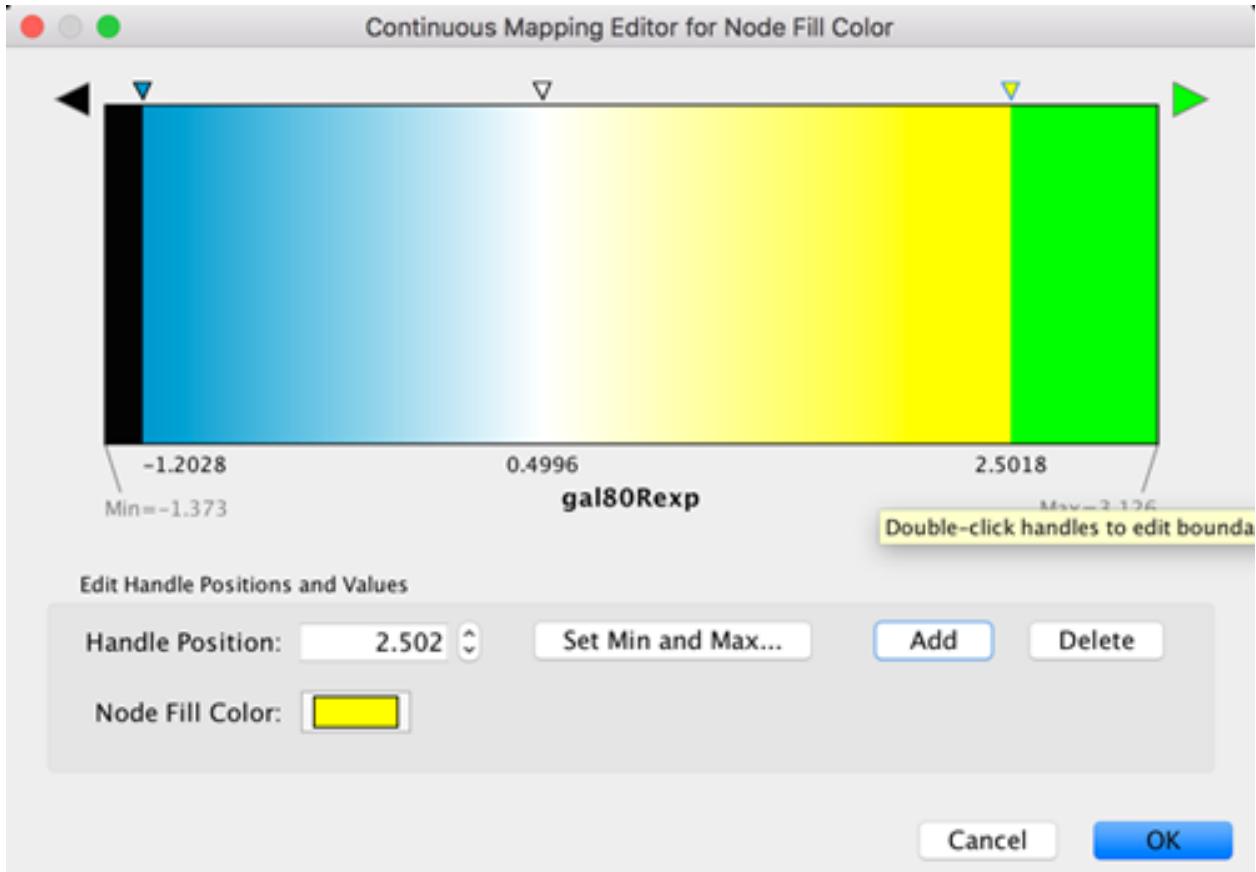
- Open the **Styles** by selecting its tab in the “Control Panel” (the leftmost panel).
- Use the “Common” name attribute to give the nodes useful names.
- Zoom in on the network so that node labels are visible.
- Click the second column of the “Label” row in the **Styles** panel. This should produce a drop-down panel with **Column** and **Mapping Type**.
- Change the “Column” to **COMMON** by clicking on the field to the right of the **Column** label. This should bring up a list of columns. Select **COMMON**.
- Verify that the node labels on the network have changed to their common names.
- By default, the **Mapping Type** is **Passthrough Mapping**, which is what we want to use. Other options are **Discrete Mapping** and **Continuous Mapping**.

26.3.2 Color the nodes

Define the node color of this visual style:

- Click on the middle square (**Map.**) next to the **Fill Color** row in the **Styles** panel.
- Click the – **select value** – cell in the **Column** section.
- This will produce a drop-down menu of available column names. Select “gal80Rexp”.
- Click the – **select value** – cell in the **Mapping Type** section.
- This will produce a drop-down menu of available mapping types. Select **Continuous Mapping**.
- This action will produce a basic black to white color gradient.
- Click on the color gradient to change the colors. This will pop-up a gradient editing dialog.
- We’re going to build a basic blue-white-yellow gradient for our expression values.
- Drag the left-most, black inverted triangle handle along the top of the gradient. Drag it to an value of approx. -1.2. Double-click on the handle and set a color in the blue range.
- Drag the white inverted triangle handle to approx 0.5. You can type the value in the **Handle Position** section to be more precise.
- Add a new handle by clicking **Add**, and drag that handle to 2.5. Set the color to yellow.
- Finally, set the **Maximum Color** by double-clicking on the white, left-pointing triangle. Set it to green.
- You can also change the color of each handle by double-clicking or using the **Node Fill Color** selector button in the **Handle Settings** section.
- This should produce a Blue-White-Yellow Color gradient like the image below, with min and max extremes colored black and green, respectively.

- Click **OK** to save the gradient adjustment dialog and verify that the nodes in the network reflect the new coloring scheme.



26.3.3 Set the Default Node Color

Note that the default node color of pale blue falls within this spectrum. A useful trick is to choose a color outside this spectrum to distinguish nodes with no defined expression value and those with slight repression.

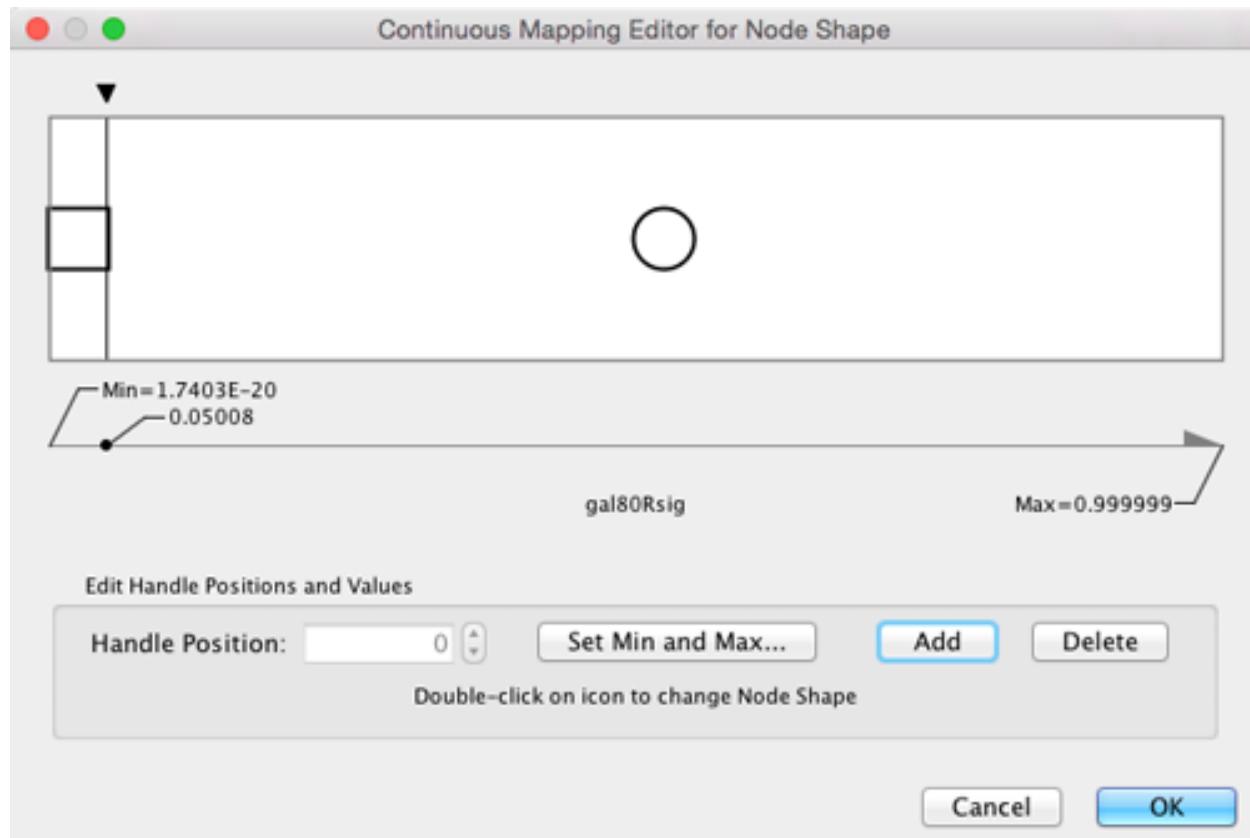
- Click the **Def.** (leftmost) square next to **Fill Color** and choose a dark gray color.
- Zoom out on the network view to verify that a few nodes have been colored gray.

26.3.4 Set the Node Shape

We imported both expression measurement values and significance values for those measurements. We can use the significance values to change the shape of the nodes so that measurements we have confidence in appear as squares while potentially bad measurements appear as circles.

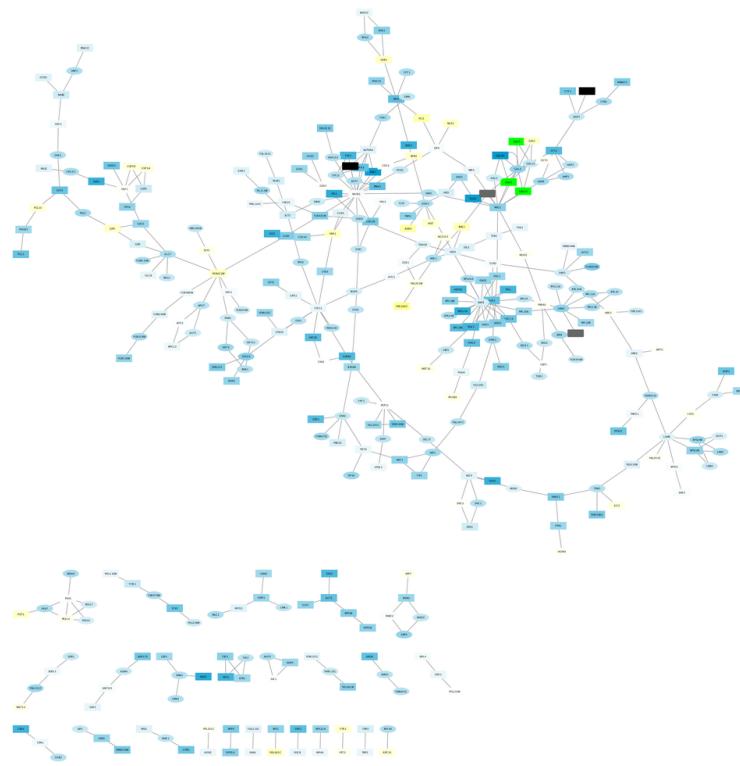
- Click the **Map.** cell next to the **Shape** row in the **Style** panel.
- Click the **- select value -** cell next to **Column**.
- This will produce a drop-down menu of available column names. Select “gal80Rsig”.
- Click the **- select value -** cell next to **Mapping Type**.
- This will produce a drop-down menu of available mapping types. Select **Continuous Mapping**.

- This will create an empty icon in the **Current Mapping** row of the **Shape** section. Click on this icon.
- This action will pop-up a continuous shape selection dialog.



- Click the **Add** button.
- This action will split the range of values with a slider down the middle with a node shape icon to either side of the slider.
- Double-click on the left node icon (a circle).
- This will pop-up a node shape selection dialog.
- Choose the **Rectangle** shape and click the **Apply** button.
- The continuous shape selection dialog should now show both a square and a circle node shape icon.
- Click on the black triangle and move the slider to the left, to slightly lower than 0.05, our threshold for significance.
- Close the continuous shape selection dialog and verify that some nodes have a square shape and some nodes have a circular shape.

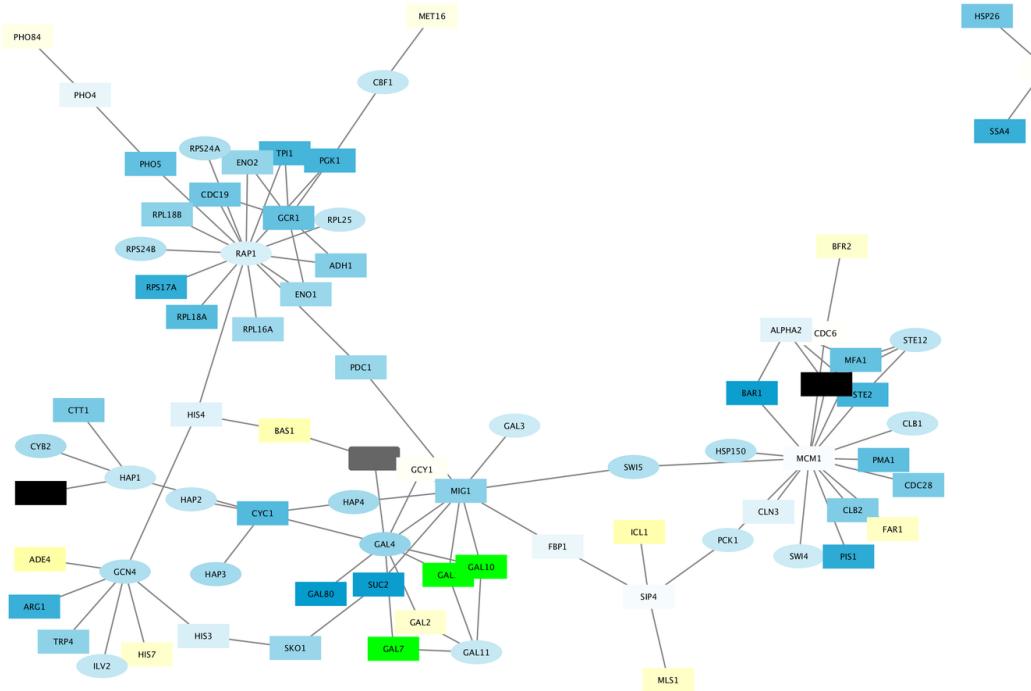
The network should now look like this:



26.3.5 Filter Interactions

Your network contains a combination of protein-protein (pp) and protein-DNA (pd) interactions. Here, we shall filter out the protein-protein interactions to focus on the protein-DNA interactions.

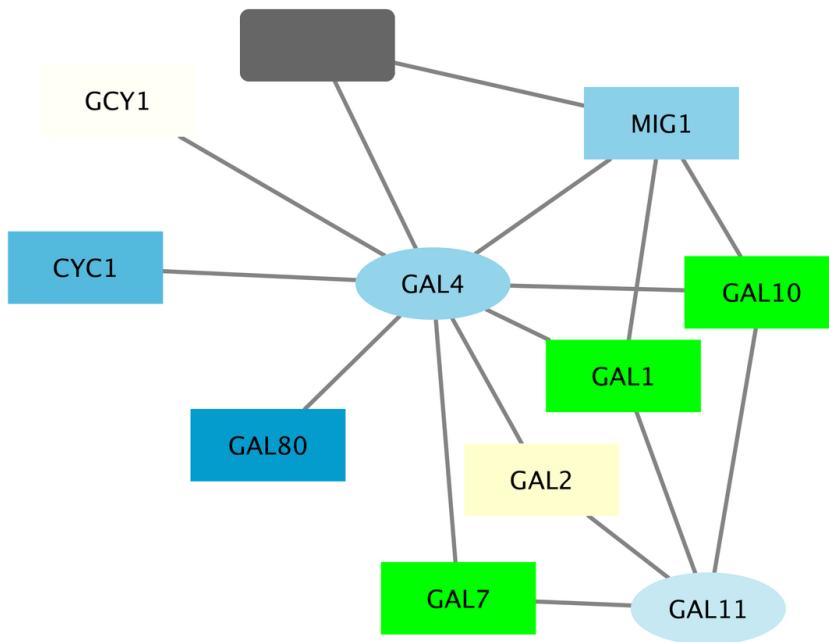
- Click the **Select** tab in the **Control** panel.
- Click the + icon Plus icon.png and select **Column Filter**.
- Select the drop-down list and choose **Edge: interaction**.
- This action will create a text search box entry in the filter.
- Type the letters “pp” into the text search box. This indicates that we’re searching for all edge interaction attributes that match the string “pp”.
- For reasonably sized networks, the filter will automatically be applied. For larger networks, you may need to click the **Apply** button at the bottom of the **Select** panel.
- You should now see many edges in the network selected (i.e., colored red).
- Since we’re only interested in the protein-DNA edges, we can delete the protein-protein edges we’ve just selected.
- Select the menu **Edit → Delete Selected Nodes and Edge**. You should now see many unconnected nodes in the network.
- Select the menu **Layout → Cytoscape Layouts → Prefuse Force-Directed Layout** to clean up the network visualization.
- The largest component of the final filtered and cleaned up network should look like this:



26.3.6 Observe the Network

Notice that three bright green (highly induced) nodes are in the same region of the graph. Zoom into the graph to see more details.

- Notice that there are two nodes that interact with all three green nodes: GAL4 (YPL248C) and GAL11 (YOL051W).
- Select these two nodes and their immediate neighbors by selecting the menu **Select → Nodes → First Neighbors of Selected Nodes → Undirected**.
- It is sometimes useful to create a new network from selected nodes. Do this by selecting the menu **File → New → Network → From Selected Nodes, All Edges**.
- With some layout and zooming, this new network should appear similar to the one shown:



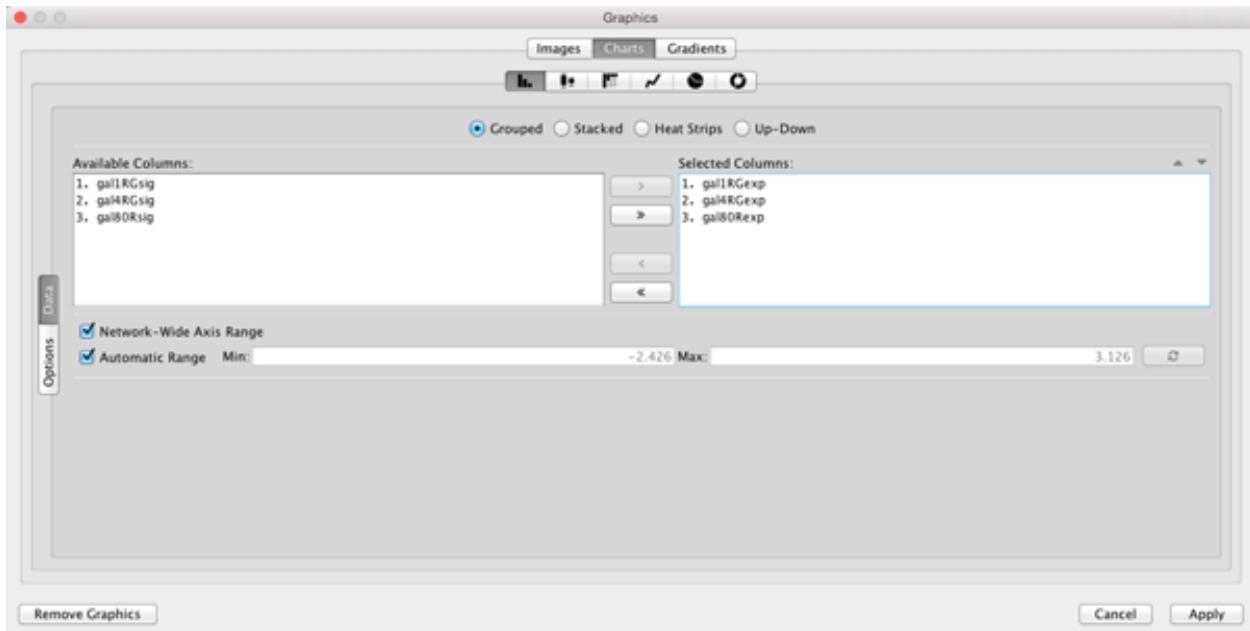
26.3.7 Exploring Nodes

- Right click on the node GAL4.
- Select the menu **External Links → Sequences and Proteins → Entrez Gene**.
- This action will pop-up a browser window and search the Entrez Gene database for the term “YPL248C”, the id of the node.
- In the results in the browser the first entry should be labeled GAL4. Click on this entry.
- The description of GAL4 tells us that it is repressed by GAL80.
- Our data show precisely this:
 - Both nodes (GAL4 and GAL11) show fairly small changes in expression, and neither change is statistically significant: they are rendered as light-colored circles. These slight changes in expression suggest that the critical change affecting the black nodes might be somewhere else in the network, and not either of these nodes.
 - GAL4 interacts with GAL80 (YML051W), which shows a significant level of repression: it is depicted as a red square.
 - Note that while GAL80 shows evidence of significant repression, most nodes interacting with GAL4 show significant levels of induction: they are rendered as green squares.
 - GAL11 is a general transcription co-factor with many interactions.
- Putting all of this together, we see that the transcriptional activation activity of Gal4 is repressed by Gal80. So, repression of Gal80 increases the transcriptional activation activity of Gal4. Even though the expression of Gal4 itself did not change much, the Gal4 transcripts were much more likely to be active transcription factors when Gal80 was repressed. This explains why there is so much up-regulation in the vicinity of Gal4.

26.3.8 Fun with Charts

In addition to coloring the nodes, Cytoscape also provides the ability to draw charts and graphs on each node. For example, suppose we wanted to display a bar chart showing all of the expression values on each of our nodes?

- To reset things a little, remove the mapping for **Fill Color** by doing a Right-Click over the **Fill Color** row and selecting **Edit → Remove Mappings from Selected Visual Properties**.
- Now change the default value to a lighter shade of grey so we can see our chart.
- Near the top of the panel, select **Properties** and choose **Paint → Custom Paint 1 → Image/Chart 1**. This will add a new row in our list of **Node Visual Properties** called **Image/Chart 1**.
- Select the **Def.** (leftmost) cell in the **Image/Chart 1** row to bring up the **Graphics** dialog.
- Select the **Charts** tab.
- Move the three columns containing the expression data (gal1RGexp, gal4RGexp, gal80Rexp) from **Available Columns:** to **Selected Columns:** by selecting the rows and clicking the right arrow. This indicates that we're going to use the data from these three columns to create our chart.
- Now select **Heat Strips** for the type of bar chart.



- Click on **Options** if you want to add labels to the graphs, change the default coloring, etc.
- Click **Apply** to see the resulting charts.

