



SOFTWARE TOOL

setsApp: Set operations for Cytoscape Nodes and Edges [v1; ref status: indexed, <http://f1000r.es/3ml>]

John H. Morris¹, Allan Wu¹, Nadezhda T. Doncheva^{1,2}, Mario Albrecht³⁻⁵, Thomas E Ferrin¹

¹Resource for Biocomputing, Visualization, and Informatics, University of California, San Francisco, CA 94143, USA

²Max Planck Institute for Informatics, Saarbrücken, 66123, Germany

³University Medicine Greifswald, Greifswald, 17489, Germany

⁴Graz University of Technology, Graz, 8010, Austria

⁵BioTechMed-Graz, Graz, 8036, Austria

v1 First published: 01 Jul 2014, 3:149 (doi: [10.12688/f1000research.4392.1](https://doi.org/10.12688/f1000research.4392.1))
 Latest published: 01 Jul 2014, 3:149 (doi: [10.12688/f1000research.4392.1](https://doi.org/10.12688/f1000research.4392.1))

Abstract

setsApp (<http://apps.cytoscape.org/apps/setsapp>) is a relatively simple Cytoscape 3 app for users to handle groups of nodes and/or edges. It supports several important biological workflows and enables various set operations. setsApp provides basic tools to create sets of nodes or edges, import or export sets, and perform standard set operations (union, difference, intersection) on those sets. The sets functionality is also exposed to users and app developers in the form of a set of commands that can be used for scripting purposes or integrated in other Cytoscape apps.



This article is included in the **Cytoscape App Collection**

Open Peer Review

Referee Status:

	Invited Referees		
	1	2	3
version 1 published 01 Jul 2014	 report	 report	 report

- 1 **Jiguang Wang**, Columbia University USA
- 2 **Thomas Kelder**, EdgeLeap B.V. Netherlands
- 3 **Tamara Munzner**, University of British Columbia Canada

Discuss this article

Comments (0)

Corresponding author: John H. Morris (scooter@cgl.ucsf.edu)

How to cite this article: Morris JH, Wu A, Doncheva NT *et al.* **setsApp: Set operations for Cytoscape Nodes and Edges [v1; ref status: indexed, <http://f1000r.es/3ml>]** *F1000Research* 2014, **3**:149 (doi: [10.12688/f1000research.4392.1](https://doi.org/10.12688/f1000research.4392.1))

Copyright: © 2014 Morris JH *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

Grant information: JHM was funded by NIGM grants P41-GM103504 and P41-GM103311. AW and TF were funded by NIGMS grant P41-GM103311. NTD was partially funded by a Boehringer Ingelheim Fonds travel grant, and her research was conducted in the context of the DFG-funded Cluster of Excellence for Multimodal Computing and Interaction. MA was financially supported by the projects GANI_MED and BioTechMed-Graz.

Competing interests: No competing interests were disclosed.

First published: 01 Jul 2014, **3**:149 (doi: [10.12688/f1000research.4392.1](https://doi.org/10.12688/f1000research.4392.1))

First indexed: 14 Jul 2014, **3**:149 (doi: [10.12688/f1000research.4392.1](https://doi.org/10.12688/f1000research.4392.1))

Introduction

Cytoscape^{1,2} provides an environment for the visualization and analysis of networks and associated annotations. The primary audience for Cytoscape is the biological community and Cytoscape supports a number of standard use cases for analyzing and visualizing biological data. Many of these use cases involve the selection of a number of nodes or edges based on some analysis or annotation and either performing an action on that selection or comparing those nodes or edges to a different set of nodes or edges that resulted from alternative analyses or analyses based on alternative annotations. The core capabilities for Cytoscape provide some tools to facilitate these types of comparisons but they can be counterintuitive or complicated to use. *setsApp* is a Cytoscape 3 application that provides a general set of tools for users and developers to define and maintain sets of nodes or edges and compare those sets using the standard set operations of union, intersection, and difference.

In this paper, we present the implementation of *setsApp*, in particular, how *setsApp* integrates with the Cytoscape command system, and then present a sample biological workflow using *setsApp*.

Implementation

Cytoscape provides two approaches to implementing apps: a simple app and a bundle app. Simple apps are implemented using the same general approach as in Cytoscape 2.8, but at the cost of significant flexibility. Bundle apps utilize Open Service Gateway Initiative (OSGi)³ interfaces through APIs provided by Cytoscape to interact with the Cytoscape core functionality. *setsApp* is implemented as a bundle app and utilizes the Cytoscape 3.1.0 API. There are three main components to the *setsApp* implementation: the user interface, the command interface, and the underlying data model for maintaining sets of nodes and edges.

User interface

The *setsApp* user interface consists of menu items in the main Apps menu, node and edge context menus, and a panel added to the **Control Panel** (left or west) section of the Cytoscape user interface. The main feature of the **Sets** panel is the list of currently defined sets. Each set can be expanded to see all of the nodes or edges within that set, and context menus provide the ability to select, deselect, rename, or remove sets. When multiple sets are selected, the **Set Operations** buttons are enabled. This allows users to create new sets based on the union, intersection, or difference of other sets. Note that the results of a union or intersection are well-defined for multiple sets, but the difference operation is order dependent. If only two sets are selected, the order of selection is preserved. If more than two sets are selected, the order is the order of selection, so care must be taken when attempting to create a difference set of more than two sets. Sets can be created from the currently selected nodes or edges, or based on a particular node or edge attribute. When creating sets from attributes, the user will need to supply a prefix for the sets to be created and choose the attribute (currently only String attributes are supported) from a list. Sets can also be created by importing them from a simple text file. Each set can be individually exported to a text file.

setsApp provides a context menu for sets and set members in the control pane. In addition to having menu items to manage sets,

the user may select all set members in the network, or if the set is expanded, individual members. This functionality presents an easy way for users to visualize the results of set operations and to perform interactive exploratory analysis.

The menus provided through the top-level Apps menu offer the same functionality as the **Create set from** menu in the *setsApp* control panel with the addition of a menu to import a set from a file. Node and edge context menus provide the user with the ability to add or remove the corresponding node or edge from sets.

Command interface

In addition to the standard user interface described above, *setsApp* provides a number of “commands”. These commands may be used for scripting purposes or by other Cytoscape apps that wish to take advantage of the *setsApp* functionality. [Table 1](#) provides a list of commands and the arguments.

A command is made available to Cytoscape by creating a standard Cytoscape TaskFactory with two new properties in the org.cytoscape.work package: ServiceProperties.COMMAND_NAMESPACE, which is always set to “*setsApp*”; and ServiceProperties.COMMAND, which is the command name (e.g. “addTo”). The command arguments are implemented as Tunables within the Task called by the designated TaskFactory. Because there is no guarantee that the Task will be executed within the context of a GUI, care should be taken to make sure that the appropriate Tunable types are used. For example, the NodeList Tunable allows the command-line user to enter a list of nodes rather than assuming that the user will have selected nodes interactively.

For another Cytoscape app to use any of these commands, it would need to call one of the Cytoscape TaskManagers and provide it org.cytoscape.command.CommandExecutorTaskFactory’s createTaskIterator method with the appropriate argument map, command, and command namespace. The TaskObserver method may be used if the command returns any values. For example:

Listing 1. Example Command Usage.

```
SynchronousTaskManager tm =
    serviceRegistrar.getService
        (SynchronousTaskManager.class);
CommandExecutorTaskFactory cetf =
    serviceRegistrar.getService
        (CommandExecutorTaskFactory.class);
Map<String, Object> argMap = new
    HashMap<String, Object>();
argMap.put("name", "NewSet");
// selected is a special keyword for the
//   NodeList tunable
argMap.put("nodeList", "selected");
// the current network
argMap.put("network", "current");
// Assumes that this implements TaskObserver
tm.execute(cetf.createTaskIterator("setsApp",
    "createSet", argMap, this), this);
```

Table 1. setsApp Commands. Arguments with asterisks are required.

Command	Arguments	Description
<i>setsApp</i> addTo	edgeList =list of edges* nodeList =list of nodes* network =network to use name =name of set*	Adds the listed nodes or edges to the named set. An error will occur if the types (node or edge) do not match or if the set does not exist. The set name and one of edgeList or nodeList are required.
<i>setsApp</i> createSet	edgeList =list of edges* nodeList =list of nodes* network =network to use name =name of set*	Creates the set from the edge or node list. It throws an error if both edge and node lists are provided.
<i>setsApp</i> difference	name =name of new set* set1 =name of the first set* set2 =name of the second set*	Performs a difference of two sets and puts the result into a new set.
<i>setsApp</i> export	Column =column containing the id key name =name of set* setFile =path to the file to import*	Exports a set to the specified file using the designated column to identify the node or edge.
<i>setsApp</i> import	Column =column containing the id key Type =[Node Edge]* name =name of set* setFile =path to the file to import*	Exports a set to the specified file using the designated column to match the node or edge identifier from the file.
<i>setsApp</i> intersect	name =name of new set* set1 =name of the first set* set2 =name of the second set*	Performs an intersection of two sets and puts the result into a new set.
<i>setsApp</i> remove	name =name of set*	Removes (deletes) the set.
<i>setsApp</i> removeFrom	edgeList =list of edges* nodeList =list of nodes* network =network to use name =name of set*	Removes the listed nodes or edges from the named set. An error will occur if the types (node or edge) do not match or if the set does not exist. The set name and one of edgeList or nodeList are required.
<i>setsApp</i> rename	newName =new name for the set oldName =old (current) name for the set	Renames an existing set.
<i>setsApp</i> union	name =name of new set* set1 =name of the first set* set2 =name of the second set*	Performs a union of two sets and puts the result into a new set.

Data model

The main model object for a node or edge set is the *Set* object, which stores a map of all of the nodes or edges in this set. A *SetsManager* provides the methods to create and destroy sets. The *SetsManager* also serves the critical function of serializing the information about Sets to the default hidden table (see CyNetwork.HIDDEN_ATTRS) for nodes or edges (depending on the type of the Set). Each Set is created as a boolean column in the hidden table which is set to **true** if the corresponding node or edge is in that set. By storing values in the default hidden tables, the information about sets is automatically saved in Cytoscape sessions and restored when sessions are reloaded. *SetsManager* implements *SessionReloadedListener* and recreates the *Sets* from the information stored in the hidden table columns.

Results

A simple example use case might be the exploration of the data set from Ideker *et al.*, 2001⁴, which measured the change in expression for 331 genes after a systematic deletion of genes known to be involved in the *Saccharomyces cerevisiae* switch to galactose

metabolism. This data was combined with known protein-protein and protein-DNA interactions to explore the biological response to deletions in the presence (the G in the column names) or absence of galactose in the medium. This data set is now included as a sample with Cytoscape downloads (*galFiltered.cys*). In our workflow we use Cytoscape's **Select** panel to select all proteins that are underexpressed ($\text{gal1RGexp} < -0.5$ fold change) in the deletion of GAL1 (Figure 2). That selection is saved as a set named GAL1- (**Apps**→**SetsApp**→**Create node set**). We then select all of the proteins that are overexpressed ($\text{gal1RGexp} > 0.5$ fold change) in the deletion of GAL1 and save that selection as a set named GAL1+. Repeating this for GAL4 (column gal4RGexp) and GAL80 (column gal80Rexp) results in 6 sets altogether: GAL1+, GAL1-, GAL4+, GAL4-, GAL80+, GAL80- (Figure 3). Note that the data for GAL1 and GAL4 is in the presence of galactose, but the data for GAL80 is in the absence of galactose since GAL80 is a known repressor of GAL4.

Given those six sets of genes, we can explore the data sets by looking at combinations of the sets. For example, we could look at the

intersection of all of the underexpressed proteins by selecting each of GAL1-, GAL4-, and GAL80- in the **Sets** panel and pressing the **Intersection** button in the **Set Operations** box near the bottom of the panel. If we name the resulting set GAL- we see that it contains a single gene: ARG1. In this data set of 331 genes, only this one gene is repressed for all three of the deleted GAL genes. In the absence of galactose when GAL80 is deleted, ARG1 is underexpressed, and in the presence of galactose when either GAL1 or GAL4 are deleted the gene is also underexpressed. Looking at the expression significance values in the Node Table Panel of Cytoscape (gal1RGsig, gal4RGsig, and gal80Rsig) this is a highly significant result, although there is no direct correlation between the galactose switch and arginine biosynthesis regulation that we were able to find in the literature. On the other hand, ARG1 is regulated by the GCN4 activator which is known to repress protein synthesis during periods of stress or starvation⁵, which explains the significant down-regulation of ARG1. We can perform a similar analysis

to understand the consistent up-regulation of the five genes in the GAL+ set: GIP1, NCE103, YIG1, POT1, and ICL1. **Figure 1** shows the results of the intersection operations.

Conclusions

There are many Cytoscape workflows that could take advantage of the features of *setsApp*. Any workflow that might want to look for groups of nodes or edges that share multiple traits, or that explicitly do not share those traits. While it is possible to duplicate many of the final results enabled by *setsApp* by using Cytoscape 3.1's new **Select** panel, a user would need to know in advance exactly the combination of features that were biologically interesting. *setsApp* provides an alternative that allows users to explore various combinations of nodes and edges and to save such selections for later uses.

In the workflow we developed above, we combined the functionality of Cytoscape's **Select** panel with *setsApp* to explore combinations

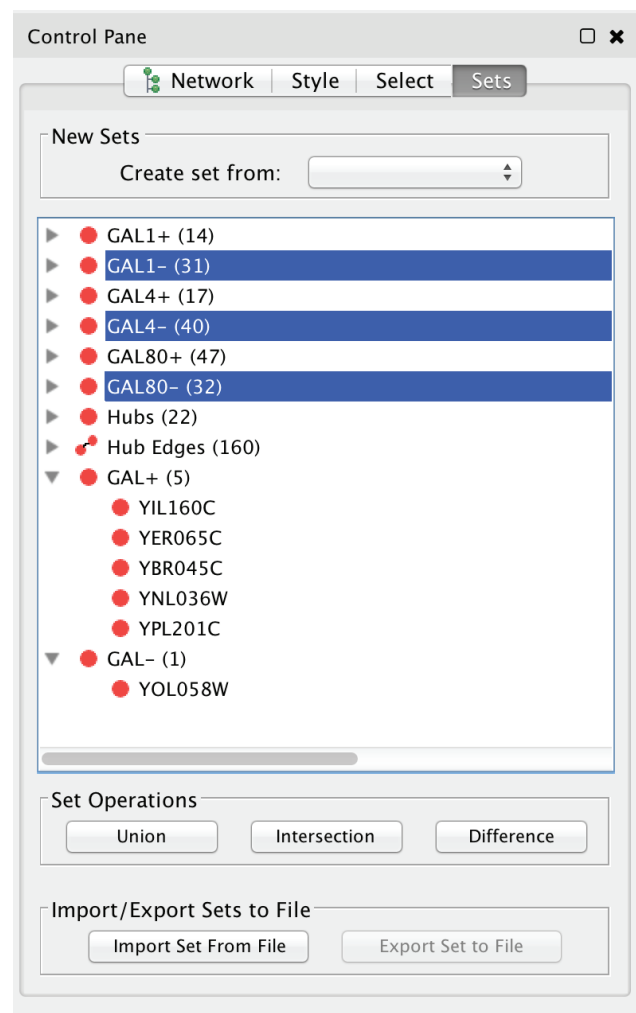


Figure 1. The *setsApp* Control Panel showing both node and edge sets. GAL+ and GAL- were created by performing the intersection of all of the + and - sets, respectively.

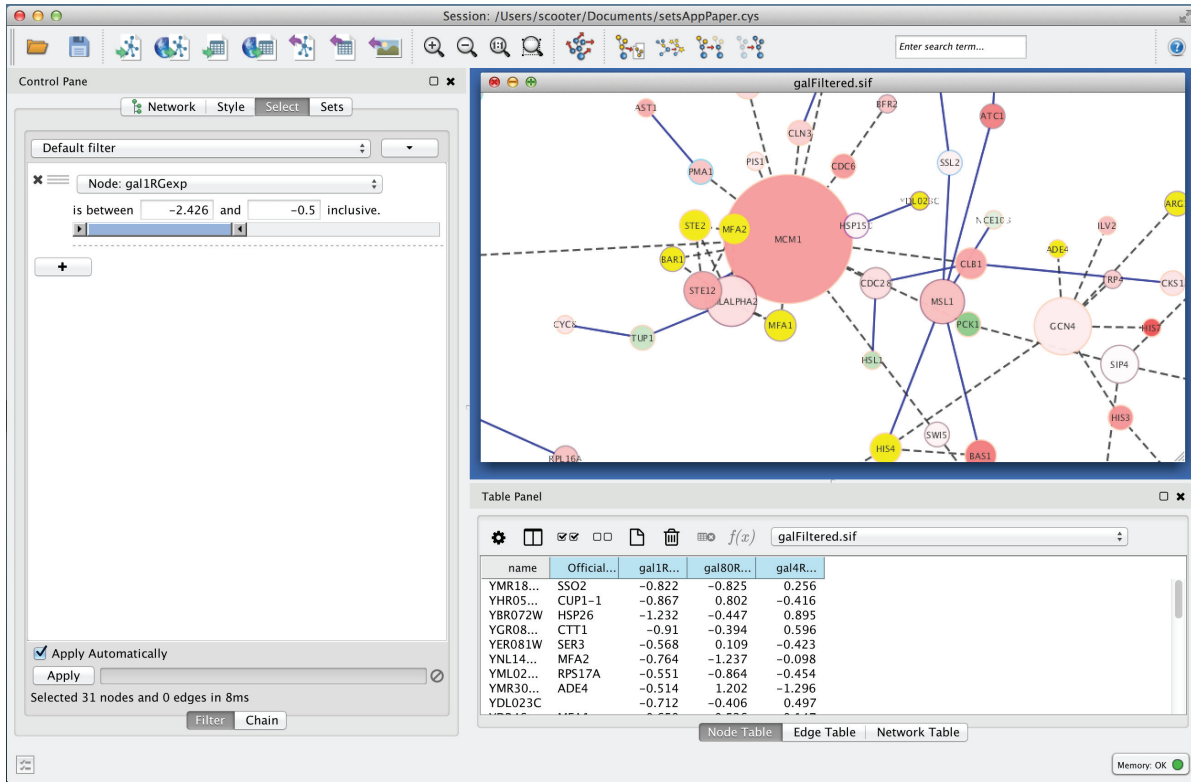


Figure 2. Screenshot of Cytoscape's Select panel with underexpressed genes in the gal1RGexp condition being selected.

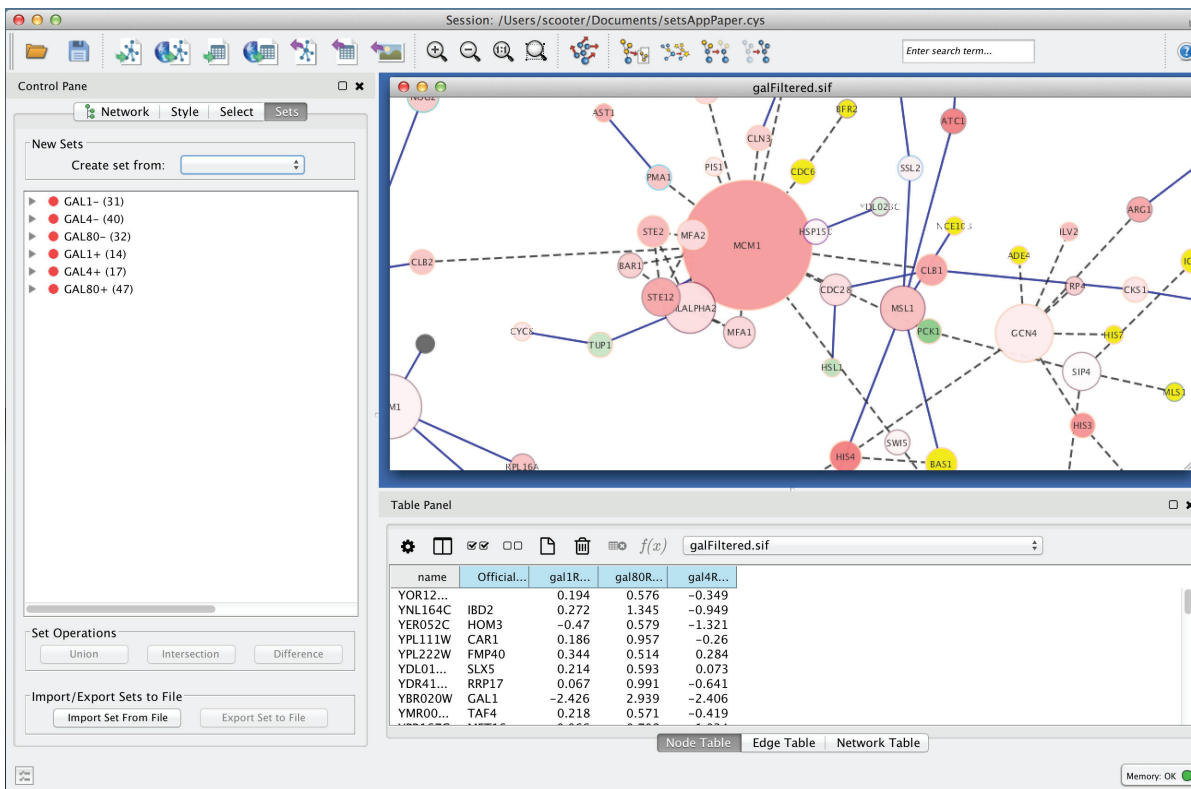


Figure 3. Screenshot of Cytoscape showing the Sets panel with all condition sets created.

of sets of genes based on shared properties. There are many more sophisticated apps available to Cytoscape users that could be used to do a more thorough analysis of this data set including jActiveModules⁶, clusterMaker⁷ and RINalyzer⁸, however, the workflow above demonstrates the utility of a simple set-oriented approach to exploring networks.

Software availability

Software available from: <http://apps.cytoscape.org/apps/setsApp>

Latest source code: <https://github.com/RBVI/setsApp>.

Source code as at the time of publication: <https://github.com/F1000Research/setsApp>

Archived code as at the time of publication: <http://www.dx.doi.org/10.5281/zenodo.104249>

License: Lesser GNU Public License 3.0: <https://www.gnu.org/licenses/lgpl.html>

Author contributions

JHM wrote the manuscript and enhanced the app. AW ported the initial version of the app from Cytoscape 2.8 to Cytoscape 3. NTD wrote the initial version of the app. MA and TF supervised app development and provided input on the manuscript.

Competing interests

No competing interests were disclosed.

Grant information

JHM was funded by NIGMS grants P41-GM103504 and P41-GM103311. AW and TF were funded by NIGMS grant P41-GM103311. NTD was partially funded by a Boehringer Ingelheim Fonds travel grant, and her research was conducted in the context of the DFG-funded Cluster of Excellence for Multimodal Computing and Interaction. MA was financially supported by the projects GANI_MED and BioTechMed-Graz.

References

- Shannon P, Markiel A, Ozier O, *et al.*: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res.* 2003; **13**(11): 2498–504.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Cline MS, Smoot M, Cerami E, *et al.*: **Integration of biological networks and gene expression data using Cytoscape.** *Nat Protoc.* 2007; **2**(10): 2366–82.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- OSGi Alliance. **OSGi service platform: release 3, March 2003.** IOS Press: Ohmsha, Amsterdam; Washington, DC, 2003.
[Reference Source](#)
- Ideker T, Thorsson V, Ranish JA, *et al.*: **Integrated genomic and proteomic analyses of a systematically perturbed metabolic network.** *Science (New York, N.Y.).* 2001; **292**(5518): 929–34.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Hinnebusch AG: **Translational regulation of yeast GCN4. A window on factors that control initiator-trna binding to the ribosome.** *J Biol Chem.* 1997; **272**(35): 21661–4.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Ideker T, Ozier O, Schwikowski B, *et al.*: **Discovering regulatory and signalling circuits in molecular interaction networks.** *Bioinformatics.* 2002; **18**(Suppl 1): S233–40.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Morris JH, Apeltin L, Newman AM, *et al.*: **clusterMaker: a multi-algorithm clustering plugin for Cytoscape.** *BMC Bioinformatics.* 2011; **12**: 436.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Doncheva NT, Assenov Y, Domingues FS, *et al.*: **Topological analysis and interactive visualization of biological networks and protein structures.** *Nat Protoc.* 2012; **7**(4): 670–85.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Morris JH, Wu A, Doncheva NT, *et al.*: **F1000Research/setsApp.** ZENODO. 2014.
[Data Source](#)

Open Peer Review

Current Referee Status:



Version 1

Referee Report 28 July 2014

doi:[10.5256/f1000research.4701.r5299](https://doi.org/10.5256/f1000research.4701.r5299)



Tamara Munzner

Department of Computer Science, University of British Columbia, Vancouver, BC, Canada

This short article clearly describes the setsApp plugin for Cytoscape and walks the reader through example analysis. It ends with a useful disclaimer that the goal of the app is to provide users with a simple workflow, rather than completely novel capabilities.

The paragraph covering TaskFactory details will probably require multiple passes for readers unfamiliar with Cytoscape internals, but the major point can still be gleaned from the write-up as it stands.

Minor issue:

It would be easier to read if the Figures were renumbered so that they match the order of discussion in the text; now Figure 1 comes after 2 and 3.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Referee Report 14 July 2014

doi:[10.5256/f1000research.4701.r5302](https://doi.org/10.5256/f1000research.4701.r5302)



Thomas Kelder

EdgeLeap B.V., Utrecht, Netherlands

The authors describe setsApp for Cytoscape, an app that provides a user and developer friendly way to define sets of nodes and edges and perform union, intersect and difference operations on these sets. The features of this app are rather basic (set definition and simple set operations), but nevertheless provide a very useful extension to core Cytoscape functionality. By designing the app so that it can be used through either the GUI or the commands interface makes it a very powerful utility component that can be used in different use cases, workflows, or as building block in other apps. The manuscript is well written and provides a clear and detailed description of the app data model, usage instructions and use case. I tested the app using Cytoscape 3.1.1 and everything worked as described.

Minor comments on the manuscript:

1. The paragraph following Table 1 is very technical. A code example to illustrate the use of Tunables would be useful. Also consider placing this paragraph after Listing 1, so the reader is first being shown how to call the setsApp commands, before diving into technical details.
2. Listing 1: The code example is incomplete, please define the “serviceRegistrar” variable, how would I get an instance as developer?

Minor comments and suggestions on the App:

1. It doesn't seem possible to compare sets across different networks, the App gives an error when I tried. What is the reasoning behind this? If the networks contain overlapping nodes, wouldn't it be valid to perform the operations across the different networks?
2. In case sets have been defined for multiple networks, it is hard to see in the Sets panel to which network each set belongs. The only way I could find was to click the set and choose “Select” so the nodes get selected in the corresponding network. It would be useful to group the sets by network or indicate the parent network otherwise (i.e. different colors of the red dots).
3. Small GUI tweak proposal: in the dialog where the user needs to specify the set name, it would be handy if the “Enter” key would map to the “Ok” button, so you don't have to switch to the mouse. It would be more intuitive and speed up the creation of several sets.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Referee Report 14 July 2014

doi:10.5256/f1000research.4701.r5300



Jiguang Wang

Department of Biomedical Informatics, Columbia University, New York, NY, USA

This manuscript described a new Cytoscape App, named *SetsApp*. This app provides very simple function about the operations on node or edge sets. It is easy to install, and easy to use. I have only minor suggestions for revision:

1. The operation "difference" on three sets is not well defined. This operation should be disabled when selecting three sets.

2. If a new set has the same name with the old one, there will be a warning. This is good, but when I close the warning, my operation is also closed. It will be more convenient if this app goes back to the window of "Enter a new name for the new set".
3. When importing sets from file: if the parameters are not properly set, the app will import nothing. It will be better if warning information tells users the parameters are not properly used.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.
