
Community Detection APplication and Service

Release 1.12.2-SNAPSHOT

Dec 21, 2022

Contents

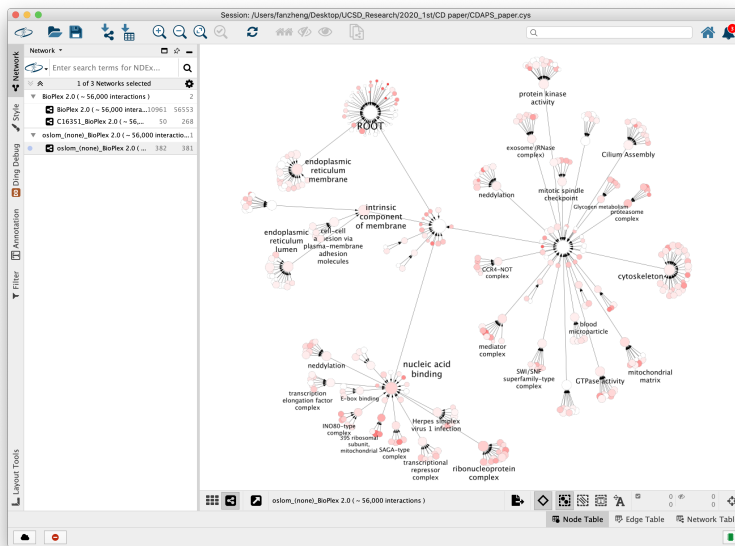
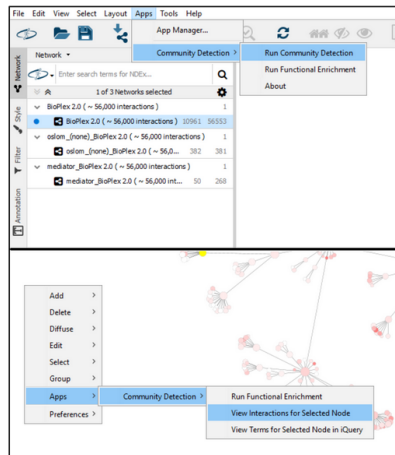
1	Feature Requests and Reporting Bugs	3
2	Cite CDAPS	5
2.1	Installation	5
2.2	What's New	6
2.3	Quick Tutorial	6
2.4	Tally Attributes on Hierarchy	14
2.5	Columns	16
2.6	Settings	18
2.7	Frequently Asked Questions	19

Community Detection Application and Service (CDAPS)

CDAPS performs multiscale community detection and functional enrichment for network analysis through a service-oriented architecture. These features are provided by integrating popular community detection algorithms and enrichment tools. All the algorithms and tools run remotely on a dedicated server.

Currently supported features:

- Community detection algorithms: [Louvain](#), [Infomap](#), [OSLOM](#), [CliXO](#), [HiDeF](#)
- Functional enrichment tools: [g:Profiler](#), [Enrichr](#), [iQuery](#)



Feature Requests and Reporting Bugs

The CDAPS GitHub issue tracker can be used to report a bug or request a feature.

To Report a bug:

- Go to <https://github.com/cytoscape/cy-community-detection/issues>
- Click on *New Issue*
- Write a short description of the issue. It is very helpful to provide a series of steps that can be taken to reproduce the issue.
- If possible attach a session file (.cys) or example input files.
- Enter App version, Cytoscape version and operating system.
- Click on *Submit new issue*

Singhal A, Cao S, Churas C, Pratt D, Fortunato S, Zheng F, et al. (2020) Multiscale community detection in Cytoscape. PLoS Comput Biol 16(10): e1008239. <https://doi.org/10.1371/journal.pcbi.1008239>

2.1 Installation

- Cytoscape minimum version 3.7 is required.

Install Cytoscape

Download and install the latest version of Cytoscape at
<https://cytoscape.org/download.html>

Install **CyCommunityDetection**

- Open Cytoscape
- In the main menu select *Apps* -> *App Manager*
- In the App Manager select **CyCommunityDetection** in the list of All Apps and click the Install button.

CyCommunityDetection can also be installed from the Cytoscape App Store at
<https://apps.cytoscape.org/apps/cycommunitydetection>

2.2 What's New

2.2.1 Version 1.12.0

- Added new menu option *Apps -> Community Detection -> Tally Attributes in Hierarchy* that annotates the hierarchy network by adding columns using values tallied from a set of attributes/columns in the parent network.
- Order of algorithms displayed in **Run Community Detection** and **Run Functional Enrichment** dialogs are now consistent with what is returned from CDAPS REST server. UD-1573

2.2.2 Version 1.11.0

- Added support for new edge list format COMMUNITYDETECTRESULTV2 allowing Community Detection algorithms to add additional custom node columns. UD-1091

Bug fixes

- Fixed bug where *View Interactions for Selected Node* no longer worked if a session was reloaded. UD-1087
- Fixed bug where any values set in the community detection or functional enrichment dialogs were lost when reopening those dialogs. [Issue #3](#)

2.2.3 Version 1.10.0

- Added new menu option *Apps -> Community Detection -> Settings* that lets caller easily change CDAPS REST server. UD-1066
- Added message letting user know using weight column in Community Detection dialog is an advanced parameter. UD-988
- In Community Detection dialog replaced **About** button with info icon next to algorithm selection dropdown. UD-987

Bug fixes

- Fixed bug where changes to properties under *Edit -> Properties -> CyCommunityDetection* were not being loaded. UD-986

2.2.4 Version 1.0

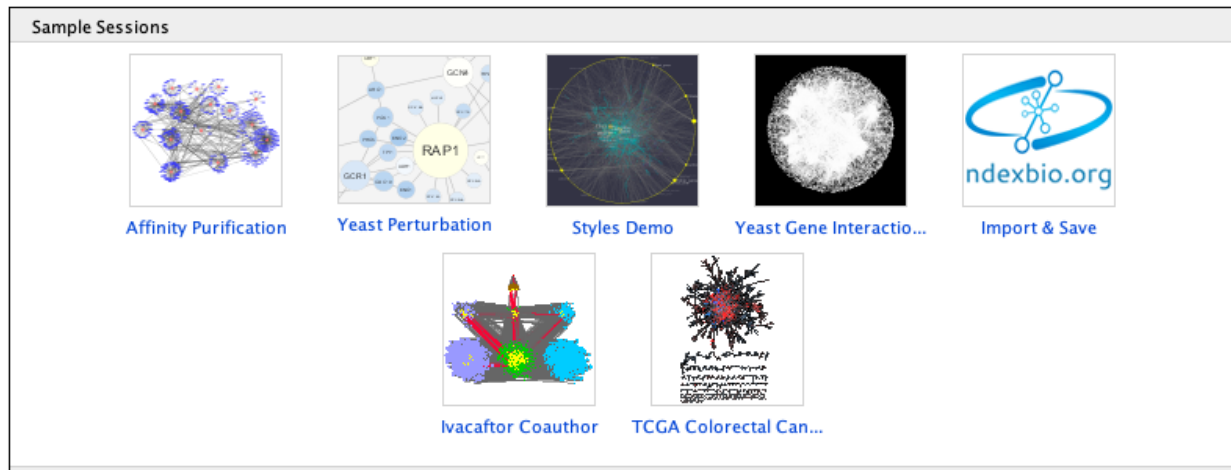
- First release

2.3 Quick Tutorial

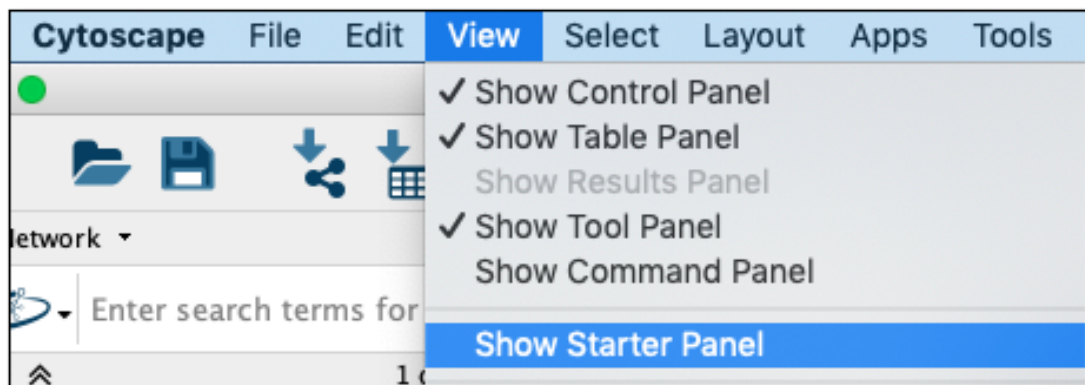
2.3.1 Open a network

To run Community Detection, a network must be loaded in Cytoscape.

From within Cytoscape click on **Affinity Purification** network on the starter panel:

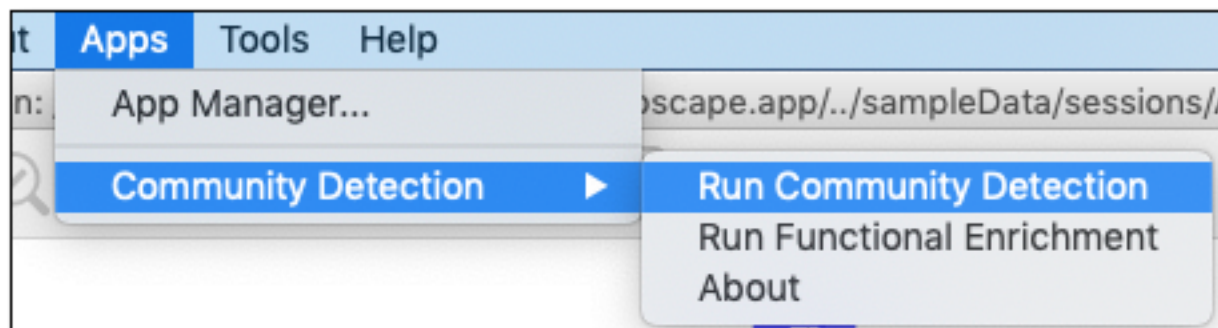


If **not** displayed, the **Starter Panel** can be displayed by invoking the menu option *View -> Show Starter Panel*.



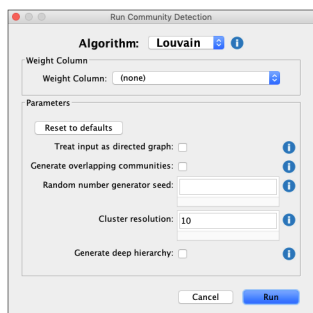
2.3.2 Run Community Detection

With the network loaded click on *Apps -> Community Detection -> Run Community Detection* menu option.

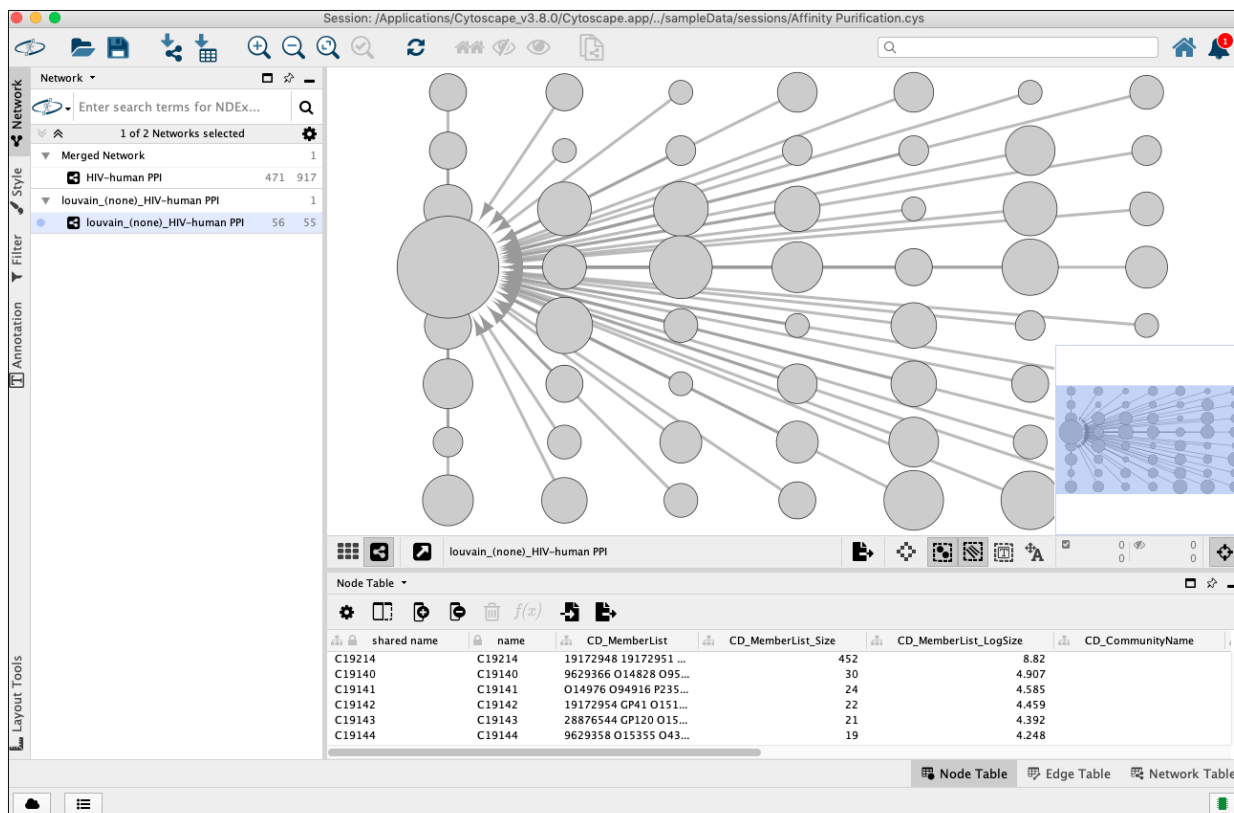


The above step will display a dialog seen below.

Select **Louvain** from algorithm dropdown and click **Run** button.



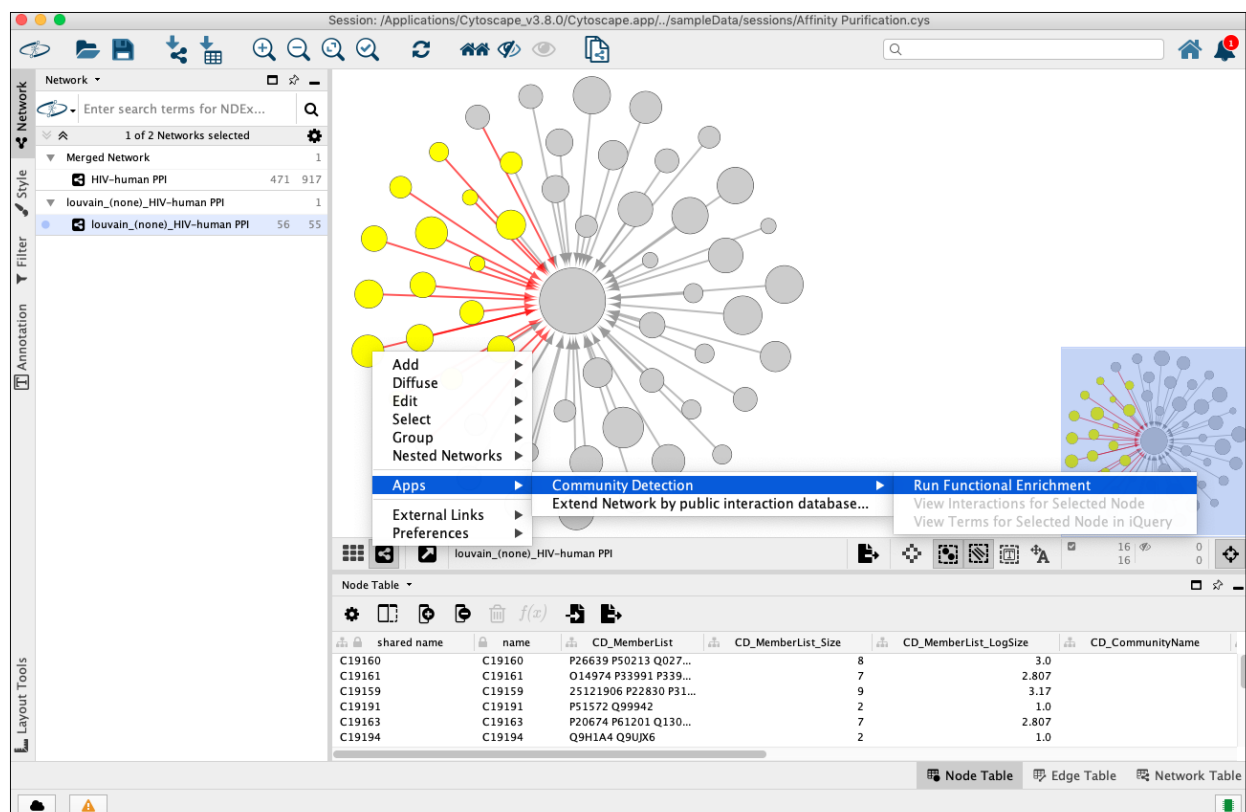
A new network/hierarchy should be generated as seen here (The current default layout will be used):



Note: Each node in the network/hierarchy above represents a cluster with the members of that cluster set in the **CD_MemberList** node column

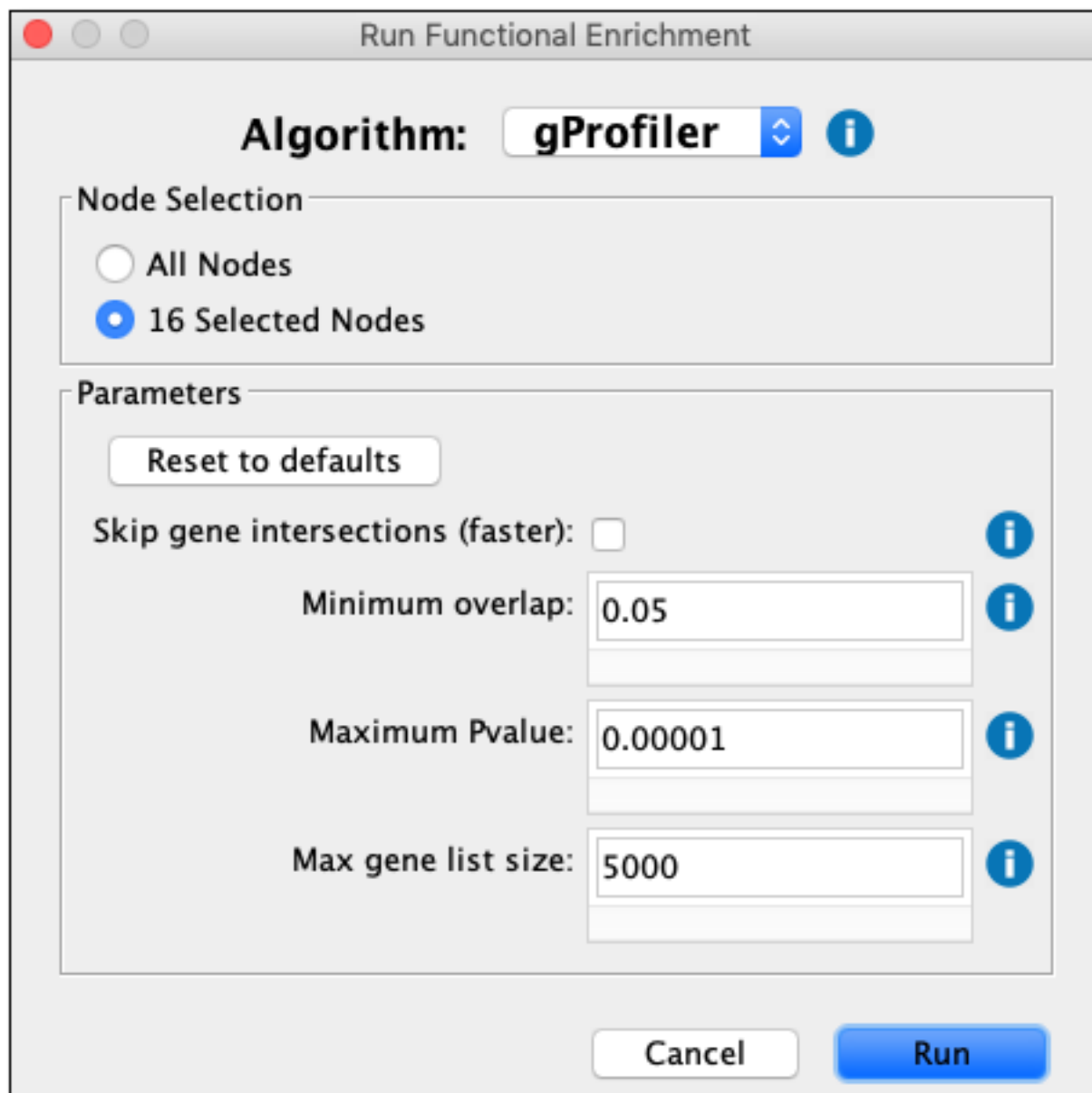
2.3.3 Perform Term Mapping

Using network/hierarchy generated above select a few nodes in the network view and then right click on a selected node to display the submenu and select *Apps -> Community Detection -> Run Functional Enrichment* as seen here:



The above step will display a dialog seen below:

Select **gProfiler** from algorithm dropdown and click **Run** button.



The image shows a 'Run Functional Enrichment' dialog box. At the top, the title bar says 'Run Functional Enrichment'. Below it, the 'Algorithm:' dropdown is set to 'gProfiler'. There are two sections: 'Node Selection' and 'Parameters'. In 'Node Selection', '16 Selected Nodes' is selected with a radio button. In 'Parameters', there is a 'Reset to defaults' button. Below it, 'Skip gene intersections (faster):' is an unchecked checkbox. 'Minimum overlap:' is a text input field with '0.05'. 'Maximum Pvalue:' is a text input field with '0.00001'. 'Max gene list size:' is a text input field with '5000'. Each of these three input fields has an information icon (i) to its right. At the bottom, there are 'Cancel' and 'Run' buttons.

Run Functional Enrichment

Algorithm: **gProfiler**

Node Selection

☐ All Nodes

☒ 16 Selected Nodes

Parameters

Reset to defaults

Skip gene intersections (faster): ☐

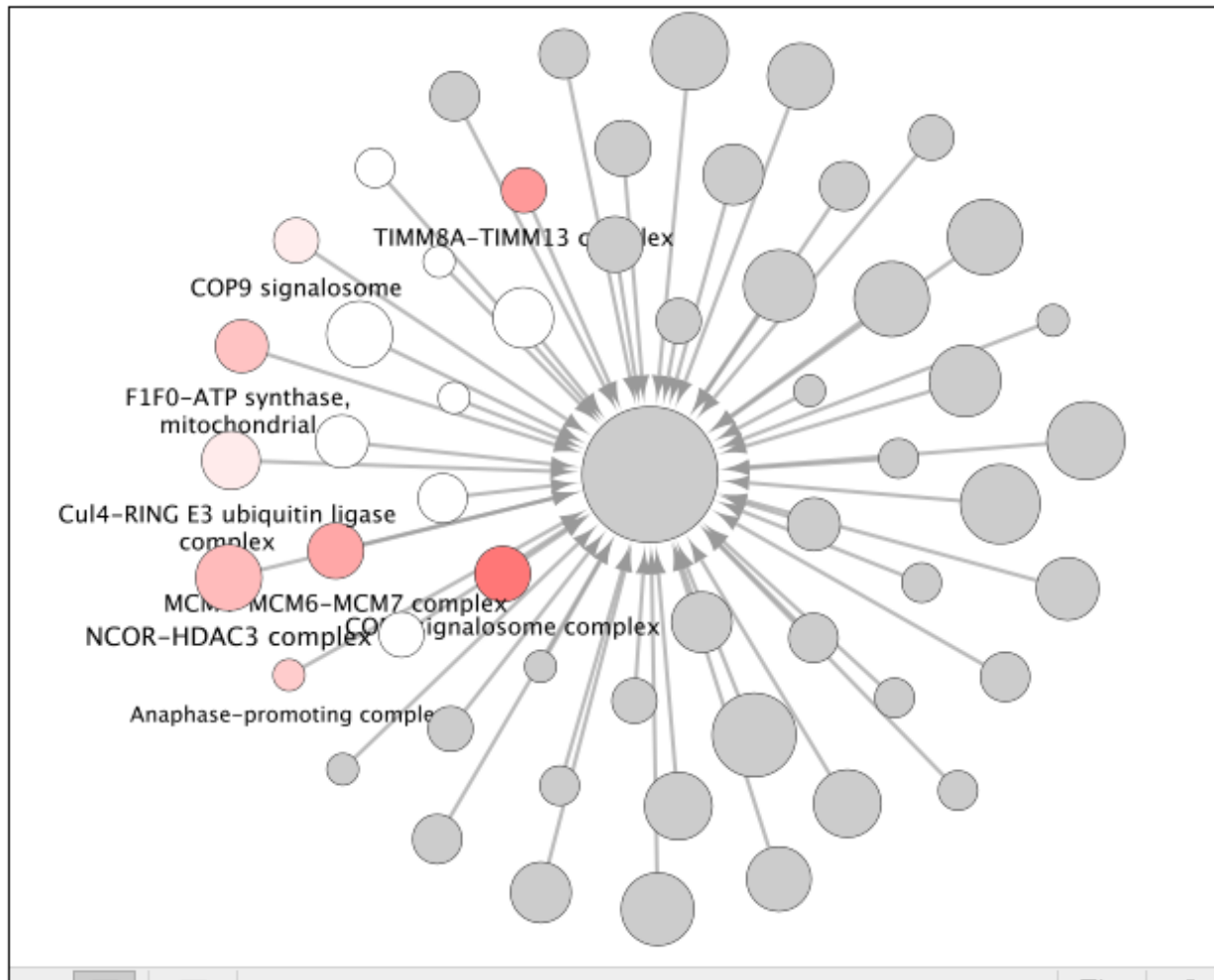
Minimum overlap: 0.05

Maximum Pvalue: 0.00001

Max gene list size: 5000

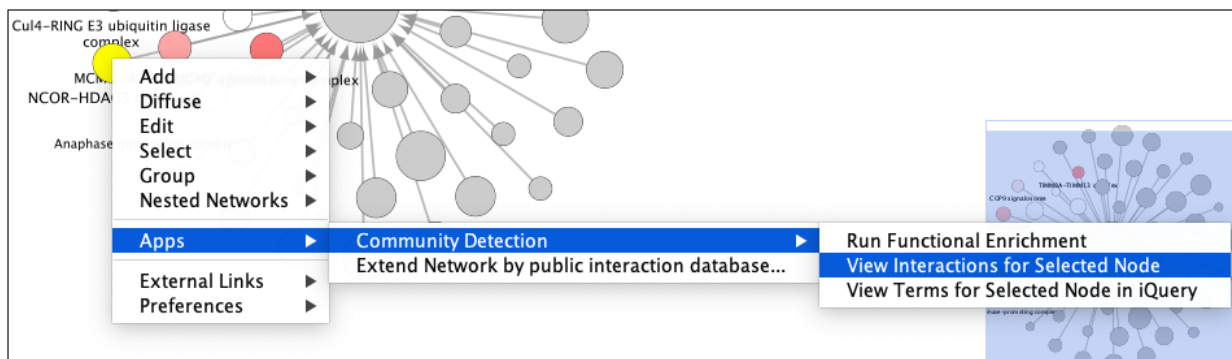
Cancel Run

gProfiler will be run and nodes will be named and colored according to overlap as seen here:

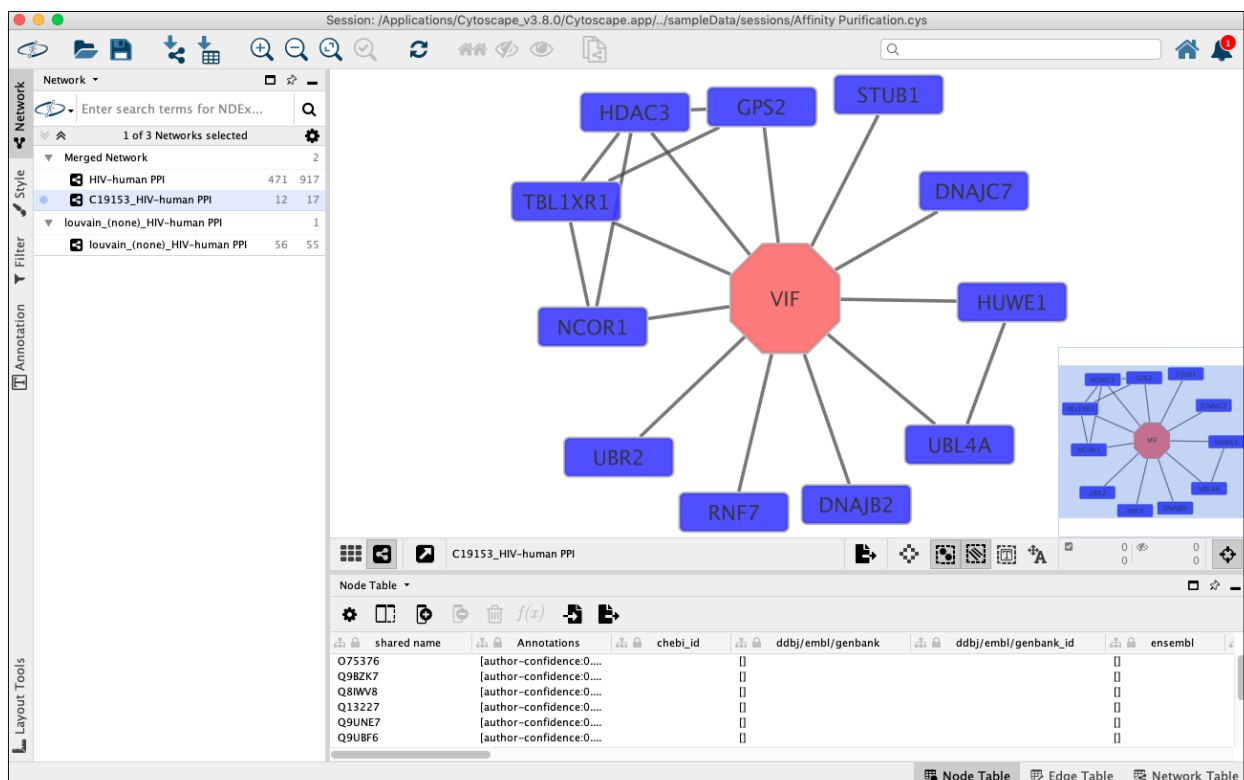


2.3.4 View Interactions

Using network/hierarchy generated above select a **single** node in the network view and then right click on the selected node to display the submenu and select *Apps -> Community Detection -> View Interactions for Selected Node*

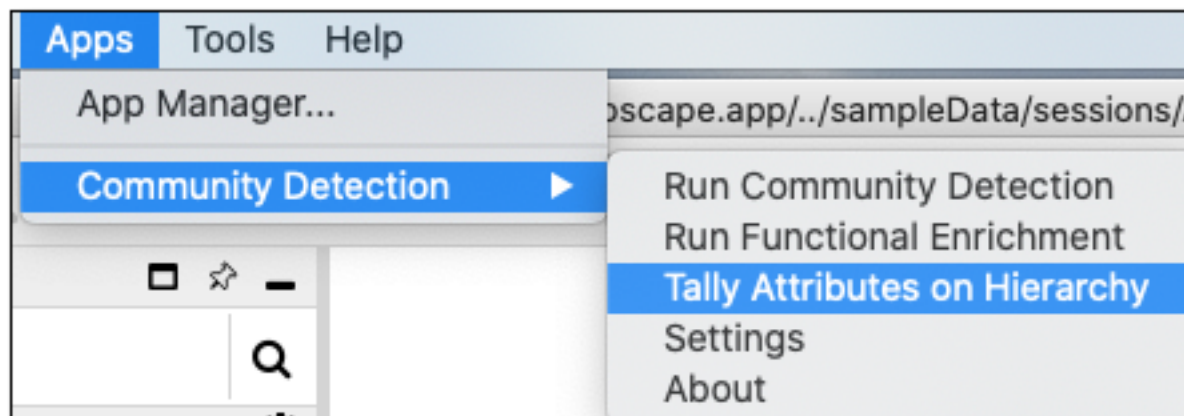


Invoking this menu option will show all the nodes pertaining to this cluster in the parent network like as seen here:



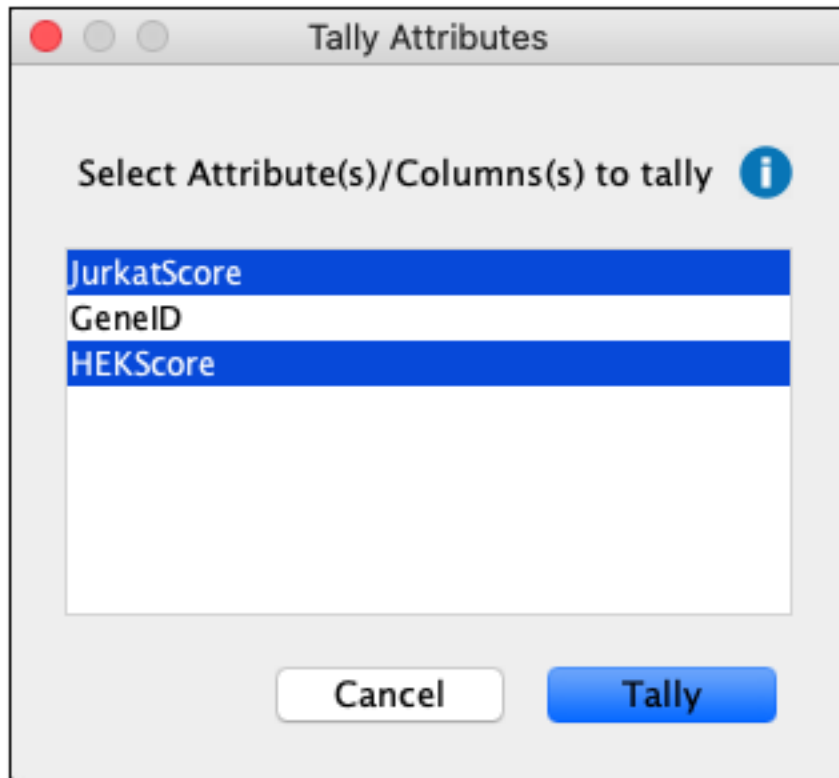
2.3.5 Tally Attributes on Hierarchy

With a CDAPS hierarchy network (as generated via *Run Community Detection* step above) click on *Apps -> Community Detection -> Tally Attributes on Hierarchy* menu option.



The above step will display a dialog seen below.

Select one or more columns from parent network to tally on hierarchy network and click **OK** button. (only columns of type Integer, Boolean, or Double will be listed)



New corresponding columns will be added to the hierarchy network node table as seen below.

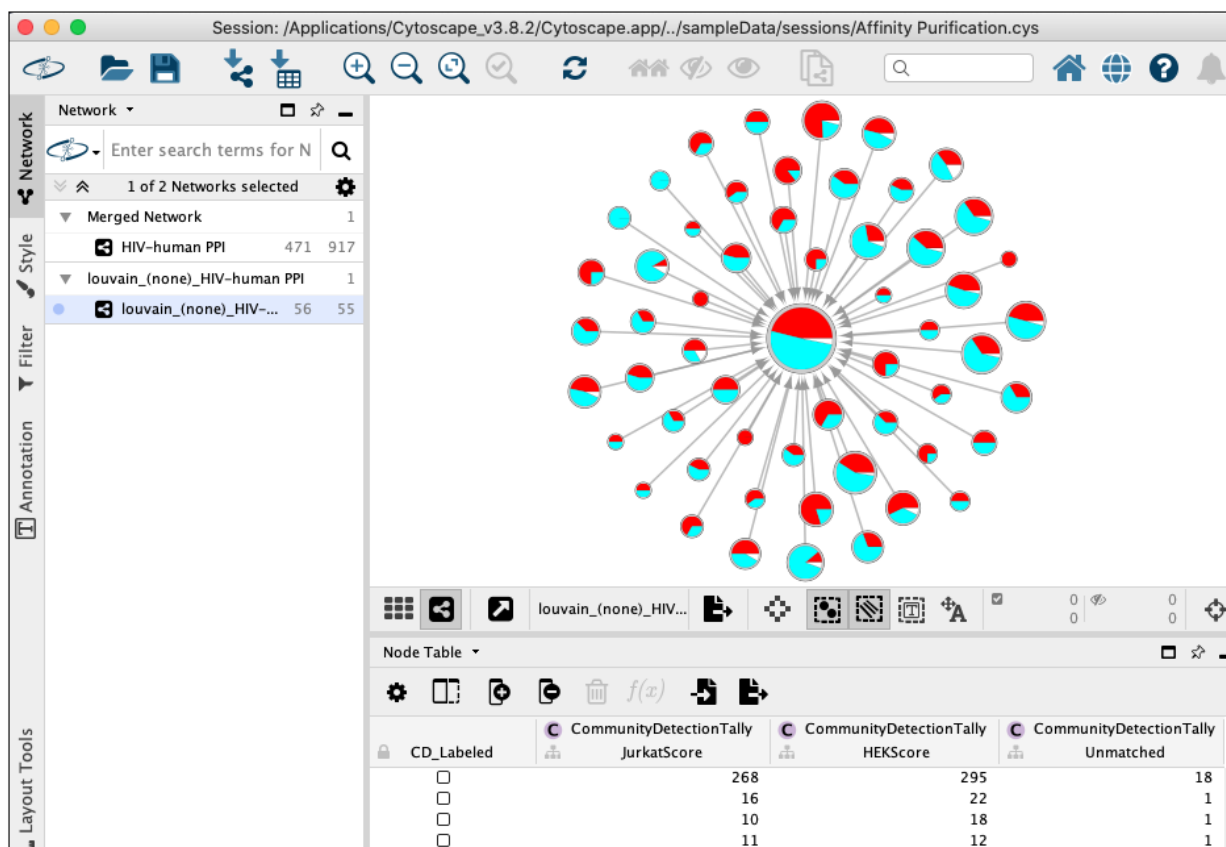
CD_Labeled	CommunityDetectionTally JurkatScore	CommunityDetectionTally HEKScore	CommunityDetectionTally Unmatched
<input type="checkbox"/>	268	295	18
<input type="checkbox"/>	16	22	1
<input type="checkbox"/>	10	18	1
<input type="checkbox"/>	11	12	1
<input type="checkbox"/>	19	5	1
<input type="checkbox"/>	10	15	1
<input type="checkbox"/>	8	14	1

These new columns match the names of the columns selected in the dialog, but with the namespace **CommunityDetectionTally** prefixed.

In addition, a **CommunityDetectionTally::Unmatched** column will also be added.

Note: These columns can be used to create charts on each node as described [here](#)

Example hierarchy network styled with [node charts](#):

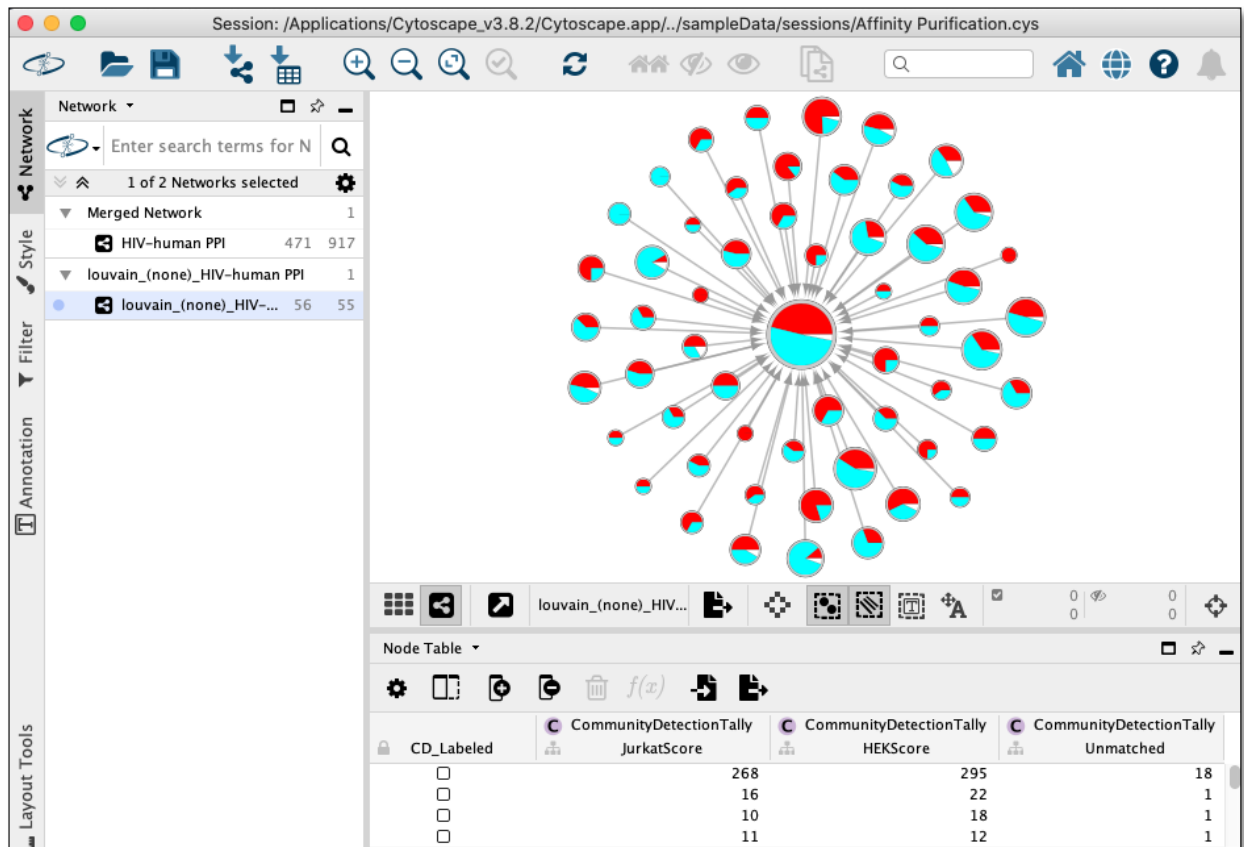


For more information click: [Tally Attributes on Hierarchy](#)

2.4 Tally Attributes on Hierarchy

The menu option *Apps -> Community Detection -> Tally Attributes in Hierarchy* provides a way to annotate the hierarchy network created when running a Community Detection algorithm from this App.

Example of *Apps -> Community Detection -> Tally Attributes in Hierarchy* styled with node charts



More specifically, *Apps -> Community Detection -> Tally Attributes in Hierarchy* menu option provides a way to count the number of members in each hierarchy network node that have a *true/positive* value for a specified set of attributes/columns in the parent network.

Note: Members of of a hierarchy node cluster are set in the node `CD_MemberList` column as a space delimited string of parent node names

These counts are stored as new columns/attributes on the hierarchy network with the same name as seen in the parent network, but prefixed with **CommunityDetectionTally** namespace.

In addition, any members in the hierarchy cluster that do **NOT** match any of the specified set of attributes/columns are counted in the **Unmatched** column/attribute.

Warning: For attribute(s)/column(s) of type **Double**, the value is rounded to nearest integer before checking to see if the value is positive

2.4.1 To Run

Select a hierarchy network created by CDAPS in Cytoscape and click on *Apps -> Community Detection -> Tally Attributes on Hierarchy* menu option.

Note: When run, any existing columns in **CommunityDetectionTally** namespace will be removed from the hierarchy

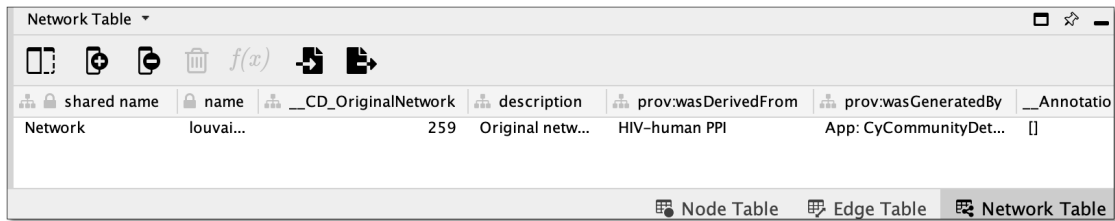
network.

2.5 Columns

This page describes the columns created by CDAPS in the network and node tables.

2.5.1 Network Columns

- Columns created in network table when invoking **Run Community Detection**



shared name	name	__CD_OriginalNetwork	description	prov:wasDerivedFrom	prov:wasGeneratedBy	__Annotation
Network	louvai...	259	Original netw...	HIV-human PPI	App: CyCommunityDet...	[]

- name - String in format:

Example:

- __CD_OriginalNetwork - SUID of parent network. This value can change upon session save/reload. If CDAPS is unable to find the parent network a dialog will be displayed to the user letting them select the correct parent network.

Note: Setting the value of __CD_OriginalNetwork network attribute to a negative value will display the Parent network dialog chooser the next time *View Interactions* is invoked

- description - Contains string in this format:

Example:

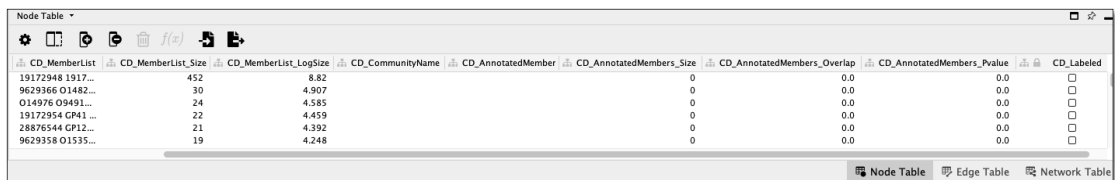
- prov:wasDerivedFrom - Name of parent network
- prov:wasGeneratedBy - Denotes version of CDAPS and algorithm used in this format:

Example:

2.5.2 Node Columns

In the node table columns created by CDAPS are prefixed with CD_

- Columns created in node table when invoking **Run Community Detection**



CD_MemberList	CD_MemberList_Size	CD_MemberList_LogSize	CD_CommunityName	CD_AnnotatedMember	CD_AnnotatedMembers_Size	CD_AnnotatedMembers_Overlap	CD_AnnotatedMembers_Pvalue	CD_Labeled
19172948 1917...	452	8.82			0	0.0	0.0	<input type="checkbox"/>
9629366 O1482...	30	4.907			0	0.0	0.0	<input type="checkbox"/>
O14976 O9491...	24	4.585			0	0.0	0.0	<input type="checkbox"/>
19172954 CP41 ...	22	4.459			0	0.0	0.0	<input type="checkbox"/>
28876544 CP12...	21	4.392			0	0.0	0.0	<input type="checkbox"/>
9629358 O1535...	19	4.248			0	0.0	0.0	<input type="checkbox"/>

- CD_MemberList - String of space delimited node names representing members of this cluster

- **CD_MemberList_Size** - Size of CD_MemberList
 - **CD_MemberList_LogSize** - Log of CD_MemberList_Size
 - **CD_CommunityName** - Name of community set by invocation of **Run Functional Enrichment**
 - **CD_AnnotatedMembers** - String of space delimited node names used to set value in CD_CommunityName
 - **CD_AnnotatedMembers_Size** - Size of CD_AnnotatedMembers
 - **CD_AnnotatedMembers_Overlap** - CD_AnnotatedMembers_Size divided by CD_MemberList_Size
 - **CD_AnnotatedMembers_Pvalue** - Pvalue obtained from term mapping algorithm invoked by **Run Functional Enrichment**
 - **CD_Labeled** - Boolean denoting if CD_CommunityName was set to a value other then blank or (*none*)
- Columns created in node table when invoking **Run Functional Enrichment**

CD_Labeled	CD_AnnotatedAlgorithm	CD_NonAnnotatedMembers	CD_AnnotatedMembers_SourceDB	CD_AnnotatedMembers_SourceTerm
<input type="checkbox"/>	Annotated by gProfiler [Docker...			
<input checked="" type="checkbox"/>	Annotated by gProfiler [Docker...	9629358 015355 043865 07560...	CORUM	CORUM:6084

- **CD_AnnotatedAlgorithm** - Algorithm used to set value in CD_CommunityName in format:
Example:
 - **CD_NonAnnotatedMembers** - String of space delimited node names **NOT** used by algorithm to set mapped term in CD_CommunityName
 - **CD_AnnotatedMembers_SourceDB** Source database used by algorithm to set mapped term in CD_CommunityName
 - **CD_AnnotatedMembers_SourceTerm** Id of mapped term set in CD_CommunityName
- Columns created in node table when invoking **Tally Attributes on Hierarchy**

CD_Labeled	CommunityDetectionTally_JurkatScore	CommunityDetectionTally_HEKScore	CommunityDetectionTally_Unmatched
<input type="checkbox"/>	268	295	18
<input type="checkbox"/>	16	22	1
<input type="checkbox"/>	10	18	1
<input type="checkbox"/>	11	12	1
<input type="checkbox"/>	19	5	1
<input type="checkbox"/>	10	15	1
<input type="checkbox"/>	8	14	1

- `CommunityDetectionTally::<TALLY SELECTED COLUMNS>` - Each column selected in the tally will have an entry and the value is the count of members that did have a positive or **true** value in the column on the parent network
- `CommunityDetectionTally::Unmatched` - Count of members in hierarchy cluster node that did **NOT** have a positive or **true** value in any of the tally columns

Note: `CommunityDetectionTally` is the namespace prefixed onto columns created when invoking **Tally Attributes on Hierarchy**

For more information click: [Tally Attributes on Hierarchy](#)

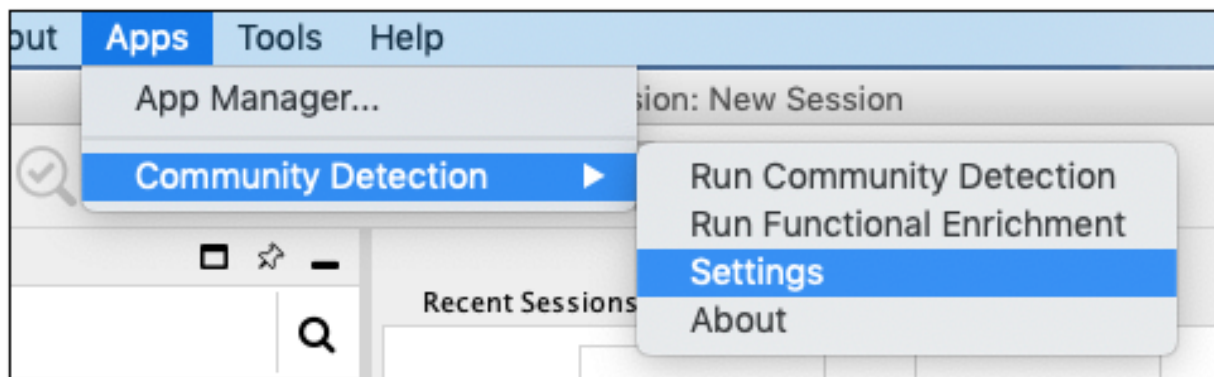
2.6 Settings

CDAPS allows user configuration via **two** methods.

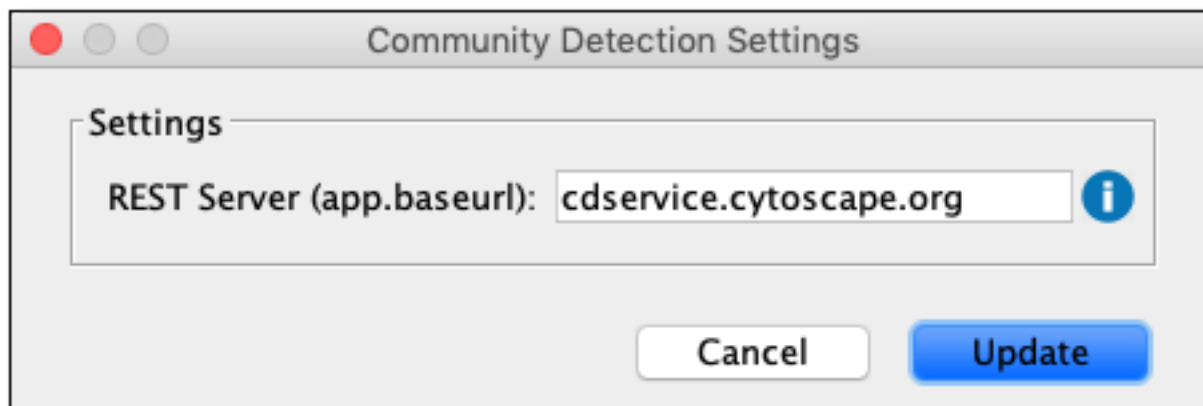
2.6.1 The first method

User configurable options can be changed via a dialog built into CDAPS. This can be found by clicking on the Cytoscape menu as follows:

Apps -> Community Detection -> Settings



Doing the above will display a dialog as seen here:



Currently the only setting that can be changed is to set an alternate CDAPS REST Server which is the server that runs the community detection and functional enrichment (term mapping) algorithms.

Note: To reset back to the default server just remove all text from this field and click **Update**

2.6.2 The second method

Warning: This is for advanced users only and may cause CDAPS to fail and/or to use excessive CPU and network bandwidth

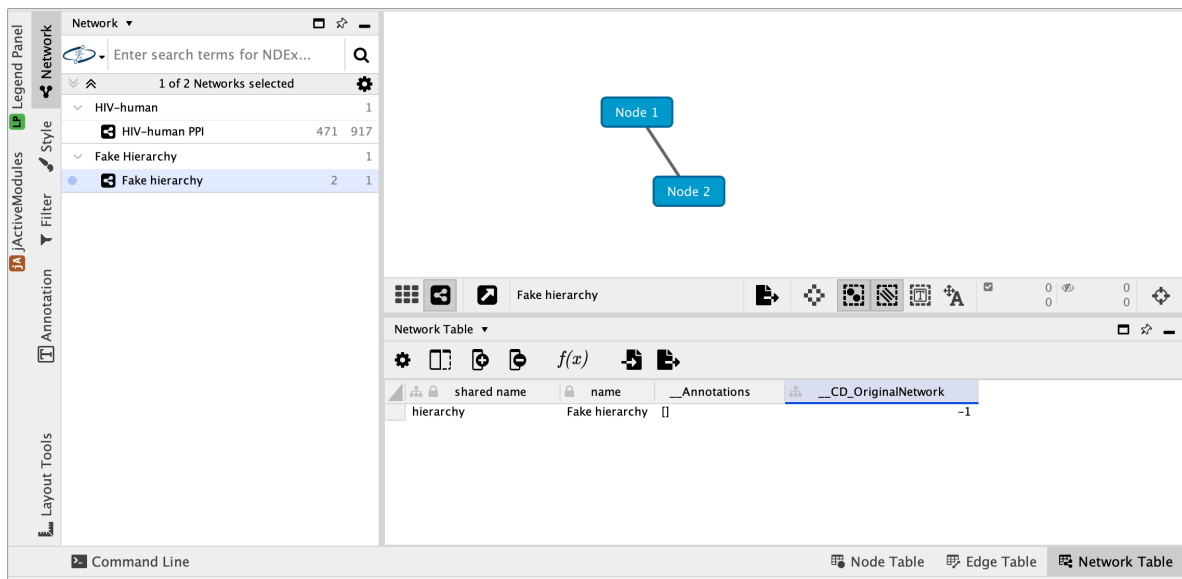
Edit -> Preferences -> Properties

In the dialog that appears, CDAPS configuration can be found by selecting `CyCommunityDetection` from drop-down menu.

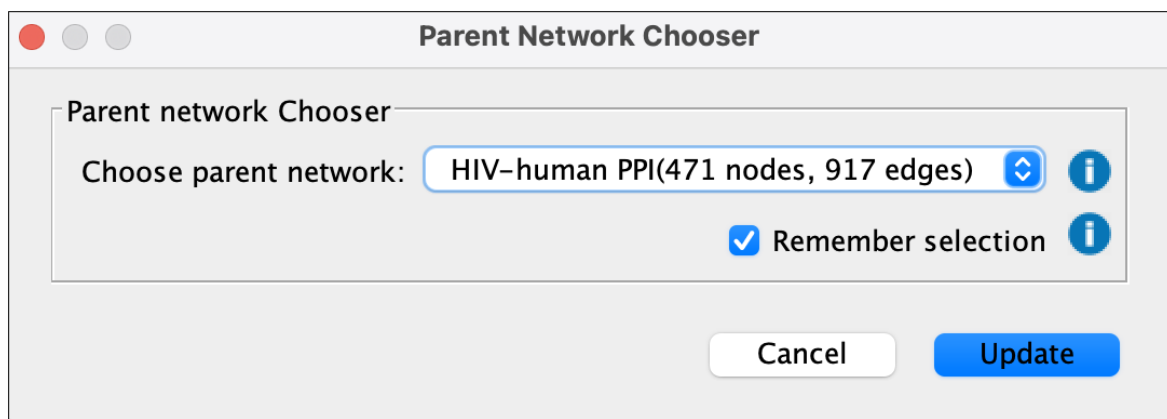
2.7 Frequently Asked Questions

2.7.1 How to set alternate parent network

1. To set an alternate parent network for a hierarchy, change the value of the network attribute `__CD_OriginalNetwork` to a negative number such as `-1` as seen here:



2. Load the alternate parent network into Cytoscape. Then select a node in the hierarchy and invoke [View Interactions](#) to, if needed, display the Parent network dialog as seen below:

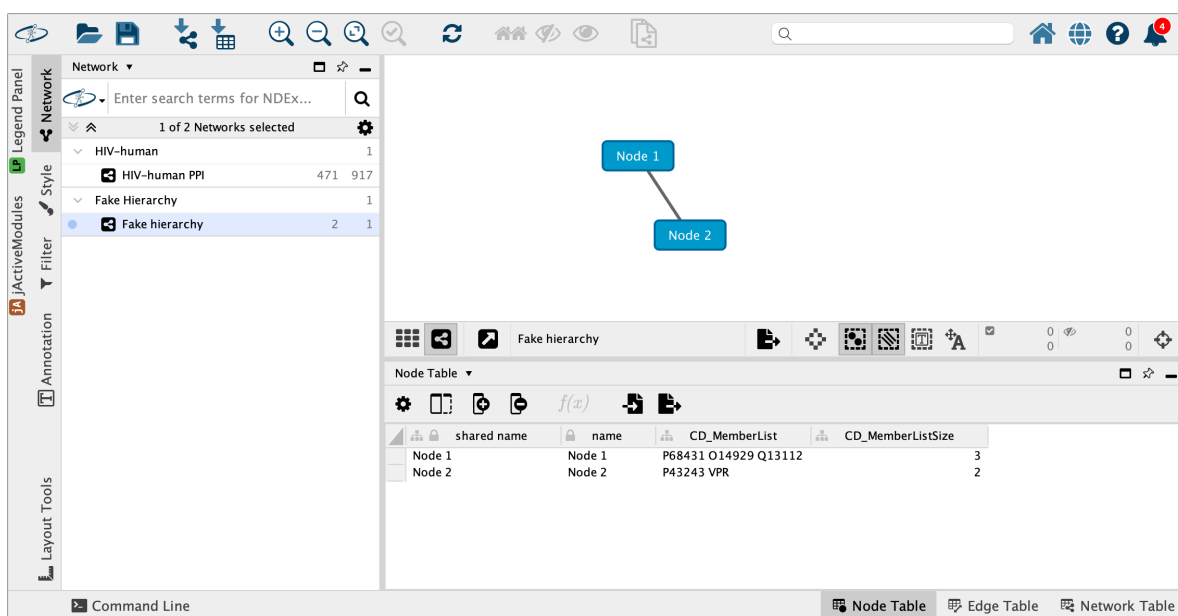


Note: Parent network dialog will only be displayed if there is more than one potential parent network

2.7.2 How to make a network look like a hierarchy network

To make hierarchy network from scratch, do the following:

1. Create/load a network and add nodes and edges to represent the hierarchy
2. Add `__CD_OriginalNetwork` network attribute column with type set to `Long Integer` and the value set to `-1`. This column tells this tool the network is a hierarchy.
3. Add `CD_MemberList` node column to each node in network with type `String` and with the value set to a list of space delimited node names that correspond to name column values from the parent network. This column is needed when doing [View Interactions](#)
4. Add `CD_MemberList_Size` node column to each node in hierarchy with type `Integer` and with the value set to the number of values in `CD_MemberList` node column. This column is needed when doing [Term Mapping](#)



5. If not already done, load the parent network into Cytoscape. In screenshot above, *HIV-human PPI* network has been loaded

Note: To get the same look and feel, copy the [style](#) from a hierarchy already run by this tool onto the hierarchy made from scratch
