Clust&See3.0 Plugin

## User Documentation

1. Starting the plugin

Once installed, the user can start/stop the plugin using the Start and Stop entries in the menu Apps->Clust&See.

Once started, the plugin opens a new tab named "Clust&See" in the left hand Cytoscape panel. This panel will be referred to as the "Clust&See Panel" below. The plugin also opens two new panels named "C&S Details" and "C&S Partition" in the bottom panel. Finally, when a partition(s) has been generated, the plugin will create new panels in the right-hand Cytoscape panel.

All those panels will be described in detail below.

2. Generating partitions

The Clust&See Panel describes the parameters required to generate a partition. These elements are described below:
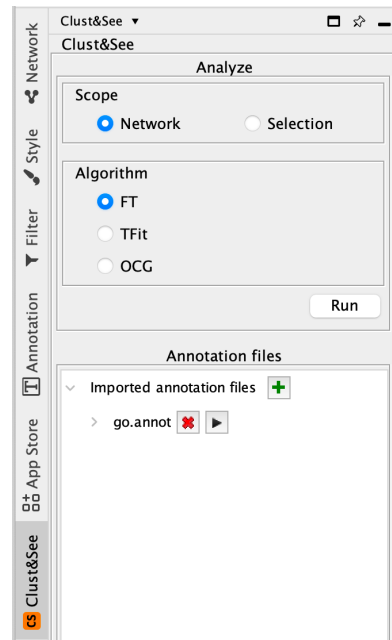
Scope

First, the user must select the scope on which to apply the plugin's actions. There are two choices: **Network** or **Selection**. The **Network** option will apply the partitioning method to the whole network (in the currently selected view) while the **Selection** option will apply it only to those nodes that are currently selected in the network. Note that this option considers only the selected nodes and will, therefore, retrieve only the edges between those nodes to create the network to be analyzed.
Clust&See should only be used for the analysis of undirected networks.

Algorithm
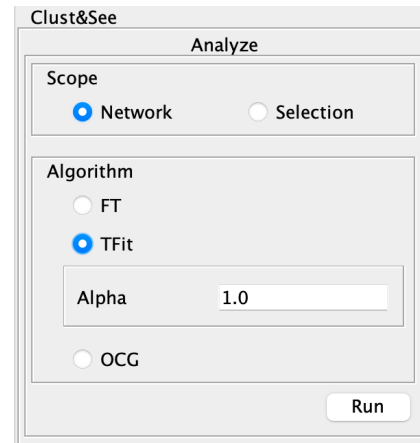Clust&See offers several algorithms to partition a network. All have been published and are succinctly described below:

* **FT [1]**: This algorithm produces strict partitions (no overlapping clusters) and has no optional parameters.

* **TFit [2]**: This algorithm produces strict partitions. The optional parameter 'alpha' is a variation of the modularity formula described in [2]:

mod = 2 * m * Aij - di * dj

With the alpha parameter:

mod = 2 * *alpha * m * Aij - di * dj / alpha

When alpha = 1, the modularity remains the one described in [2]. When 0 < alpha < 1, the partition results in smaller clusters since the optimal modularity is reached earlier than when alpha = 1. Note that alpha >1 and alpha < 0 are not allowed.

Clust&See

Analyze

Scope

○ Network   ○ Selection

Algorithm

○ FT

● TFit

Alpha   1.0

○ OCG

Run

* OCG [3]: This algorithm produces overlapping partitions. It has several optional parameters that require expert knowledge of the network partitioning domain, which is why default options are proposed to the user. However, the expert level options are available if needed.

These options are:

* Initial clusters: the type of initial cluster that will be used. Possibilities are "Centered Cliques", "Maximal Cliques" and "Edges". See [3] for more details.

* Class system: the method for detecting optimized clusters. Possibilities are "Maximize Modularity" and "Final Class".

* Cluster max. cardinality: sets an upper limit to the number of nodes a cluster may contain. When this option is set to 0, no limit is set.

* Cluster min. number: sets a lower limit to the number of final clusters the algorithm may return.

Analysis

When all the parameters have been chosen, the user may click on the "Run" button to launch the clustering.

Note: If the chosen network/selection contains more than 5000 nodes, a warning will be displayed. This warning informs the user that making partitions on a network of this size:
- may lead to a very large number of clusters that will be difficult to analyze.

- may result in very long computation times (especially for the OCG algorithm) or memory issues if the computer is not sufficiently powerful or if the memory settings on the JVM (Java Virtual Machine) have not been correctly customized.
To work around these issues, we suggest using the "Build Neighborhood Network" function (see below) to produce a smaller network centered on the nodes of interest. The user can also use alternative clustering methods, either in-house, or through the 'Linkcomm' R package [4] producing compatible output files that can be loaded in the Clust&See3.0 Cytoscape App.

3. The partition

Once a partition has been produced, Clust&See3.0 displays a summary of the partition and the global quotient graph (a graph showing the edges connecting the clusters).

3.1 Partition Summary

For each computed partition, Clust&See3.0 displays a new panel with the name of the partition in the right hand side panel.

The panel contains a table listing the discovered clusters and, for each of them:

** An image of the cluster network
** The number of nodes in the cluster.
** The number of Intra Cluster Edges: the number of edges between nodes of the cluster. This number can differ from the edge number if loop edges (self-interactions) exist in the cluster.
** The number of Extra Cluster Edges: the number of edges in the network between nodes in the cluster and nodes outside the cluster.
** The ratio of edges between nodes within the cluster over edges between nodes of the cluster and nodes in other clusters (Intra/Extra edge ratio).

   Note that the list of clusters can be classified by cluster size or by in/ext edges ratio.

 3. A "New Partition View" button that generates a new view of the complete partition (quotient graph).

 4. An "Export Partition" button that exports the partition to a file. See Export/Import below for more details

 5. A "Discard Partition" button that removes all the information related to this partition.
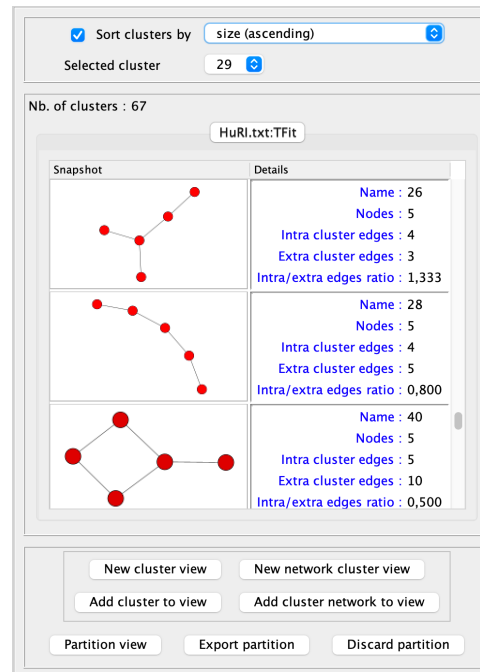
3.2 Quotient graph

At the end of the analysis, Clust&See3.0 creates a new view containing the discovered clusters represented as metanodes. Nodes representing a cluster (hereafter referred to as **metanodes**) are represented as blue square nodes with an image of the corresponding expanded subnetwork inside.
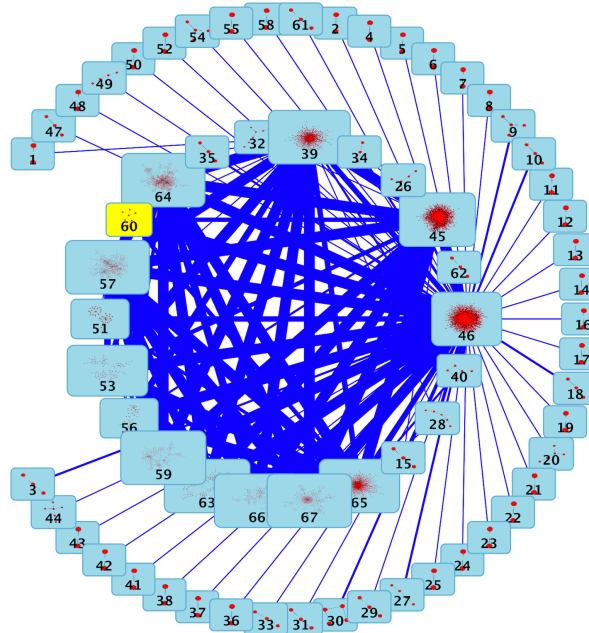
In the newly created view, the metanodes may be connected by two kinds of edges (hereafter referred to as **metaedges**):

* blue metaedges: these metaedges summarize the analyzed network edges between the nodes that are members of the two clusters. The width of the metaedge is proportional to the number of edges it summarizes. Blue metaedges can also be displayed between a metanode and a single node. In such cases, they summarize the edges existing between the single node and the nodes in the cluster.

* green metaedges: these metaedges indicate that the two connected clusters share one or several nodes. These green metaedges exist only if the clustering algorithm leads to overlapping clusters (e.g. OCG). The width of the metaedge is proportional to the number of shared nodes. Green metaedges can also be displayed between a metanode and a single vertex. In such cases, they simply indicate that the node is a multi-classed member of the cluster.

4. Analyzing partitions

Clust&See3.0 offers several functions to help in the analysis of the computed partition.

4.1 Detailed information on objects

Selecting any object shown in a view generated by Clust&See3.0 causes detailed information about this object to be shown in the C&S Details tab located in the Cytoscape bottom panel. Objects can be selected from a view (nodes, edges, metanodes, metaedges) or from the cluster list in the Results Panel (clusters). Note that selecting a metanode automatically selects the corresponding cluster in the cluster list. Similarly, selecting a cluster in the cluster list automatically selects the corresponding metanode in all open views. In addition, selecting a cluster or a metanode automatically selects all the nodes of the cluster in all the views (including the original network view).

4.1.1 Metanode details

When selecting a metanode, the C&S Details tab shows the following information:
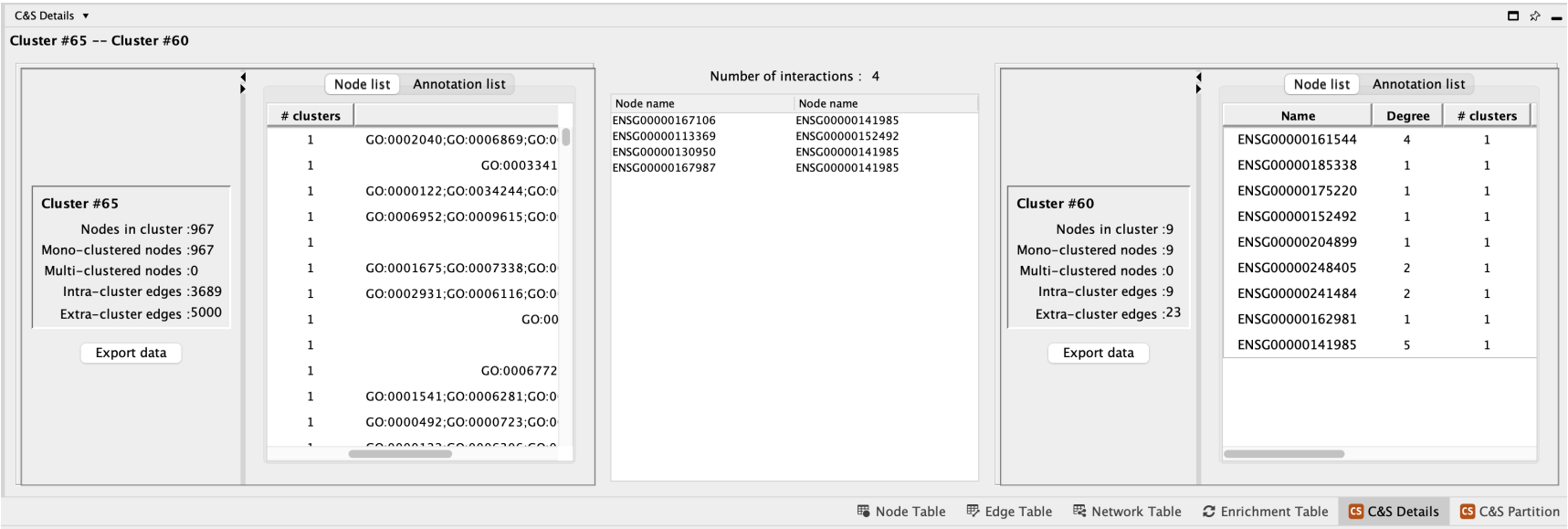

4.1.2 Blue metaedge details

When selecting a blue metaedge, i.e. a metaedge summarizing the connections between two clusters or between a cluster and a single node, the C&S Details tab shows the following information:

Cluster to cluster metaedge :

4.1.3 Details on green metaedge

When selecting a green metaedge, i.e. a metaedge indicating overlapping clusters/multiclassed nodes, the Partition Details show the following information:

Cluster to cluster metaedge :

Cluster #79 -- Cluster #76

Cluster #79
Nodes in cluster :30
Mono-clustered nodes :10
Multi-clustered nodes :20
Intra-cluster edges :55
Extra-cluster edges :1460

Export data

Node list | Annotation list

Name	Degree	# clusters	A
ENSG00000012822	3	1	GO:0007165;GO:0010628;GO:001
ENSG00000101473	1	1	GO:0006637;GO:0006699;GO:000
ENSG00000117461	4	4	GO:0001934;GO:0002042;GO:000
ENSG00000124074	6	28	GO:0000132;G
ENSG00000124788	5	10	GO:0000122;GO:0006396;GO:000
ENSG00000127191	7	36	GO:0002637;GO:0002726;GO:000
ENSG00000130559	3	1	GO:0000226;GO:0007010;G
ENSG00000137478	5	2	GO:0007274;GO:0015031;GO:003
ENSG00000138443	5	49	GO:0007010;GO:0007399;GO:000
ENSG00000145365	3	3	GO:0002753;GO:0007249;G
ENSG00000147996	1	1	G
ENSG00000149499	1	1	GO:0000226;GO:0007
ENSG00000155367	1	1	G

Multiclassed nodes : 4

Node name
ENSG00000138443
ENSG00000184033
ENSG00000197256
ENSG00000268651

Cluster #76
Nodes in cluster :199
Mono-clustered nodes :74
Multi-clustered nodes :125
Intra-cluster edges :1197
Extra-cluster edges :5248

Export data

Node list | Annotation list

Name	Degree	# clusters	A
ENSG00000011523	2	1	GO:0007098;G
ENSG00000036672	17	6	GO:0000122;GO:0007049;GO:000
ENSG00000042304	0	1	
ENSG00000051128	7	9	GO:0006605;GO:0007216;G
ENSG00000053371	1	1	GO:0005975;GO:0006081;GO:000
ENSG00000066117	32	7	GO:0006337;GO:0006338;GO:000
ENSG00000068024	10	4	GO:0000122;GO:0006338;GO:000
ENSG00000070031	2	1	GO:0002024;GO:0007165;GO:000
ENSG00000073584	17	2	GO:0006337;GO:0006338;GO:000
ENSG00000074219	3	3	GO:0001570;GO:0001843;GO:000
ENSG00000075240	2	1	GO:0006915;G
ENSG00000075426	2	2	GO:0000902;GO:0001649;GO:000
ENSG00000079102	14	5	GO:0006351;G

Command Line ··· Node Table   Edge Table   Network Table   Enrichment Table   C&S Details   C&S Partition

4.2 Customized views

When analyzing a partition, it is often necessary to focus on sub-groups of clusters. To facilitate this kind of analysis, Clust&See3.0 allows the creation of customized views of the different clusters.

Four buttons allow the user to produce a suitably customized view for analysis.

* **New Cluster View**: this button adds the cluster metanode to a new, empty view.
* **Add Cluster to View**: this button adds the cluster metanode to the currently selected view.
* **New Cluster Network View**: this button adds the entire cluster network to a new empty view.
* **Add Cluster Network to View**: this button adds the entire cluster network to the currently selected view.

Any view generated as described above can be further manipulated as follows:

* **Add all edges**: Right-clicking on a metanode or node opens a contextual menu with the entry Clust&See->Add all edges. This action will add all the edges/metaedges that connect the nodes/metanodes present in the view.

* **Expand cluster**: Right-clicking on a metanode opens a contextual menu with the entry Clust&See->Expand cluster. This action will replace the metanode by the cluster network it represents in the current view. All edges/metaedges connecting the cluster's nodes to the rest of the nodes/metanodes in the view will be added.

* **Compress cluster**: Right-clicking on a node opens a contextual menu with the entry Clust&See->Compress cluster. This action will replace the node by the cluster(s) it is a member of. All nodes belonging to the same cluster(s) will also be removed from the view.
Note that if the selected node is a member of several clusters, Clust&See3.0 will only display as metanodes those clusters that already have at least one member visible in the current view. If the selected node is the only member of a given cluster currently visible, then that cluster will not be shown as a metanode. The metaedges connecting the new metanode(s) to the rest of the nodes/metanodes in the view will be added.

4.3 Manual cluster annotation

Once a partition has been analyzed, it is often necessary to annotate clusters in order to save the conclusion of the analysis. Clust&See3.0 offer the option of adding one or several annotations to a cluster and to its nodes.

Right-clicking on a node or metanode opens a contextual menu with the entry Clust&See->Annotate Cluster. Choosing this option opens the Annotate Cluster frame where annotations can be entered:
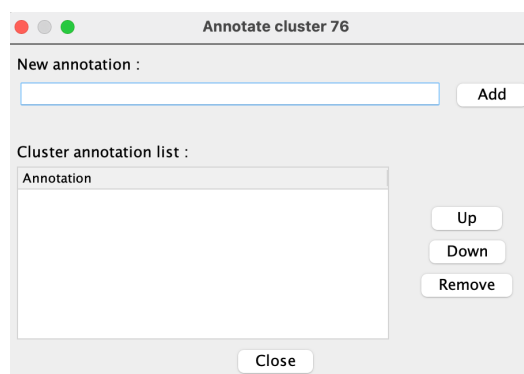
Note: this action is not available on multi-classed nodes since they are members of several clusters.

Annotations assigned to a cluster are stored at the cluster level (the "C&S Details" tab contains a specific tab to display these annotations). Annotations are also inherited by the cluster's member nodes.

4.4 Automatic Cluster Annotation

Clusters can be annotated automatically using custom annotation files.
The annotation file must be a text file and must consist in two columns, such as:

```
ENSG00000000003    GO:0039532;GO:0043123;GO:1901223
ENSG00000000005    GO:0001886;GO:0016525;GO:0001937
```

The first column contains the node identifier (one identifier per line), the second one, the list of terms associated with this identifier, separated by a comma, a semicolon or a tab. The two columns can also be separated by a comma, semicolon or tab, as long as the column separator is not the same as the term separator.
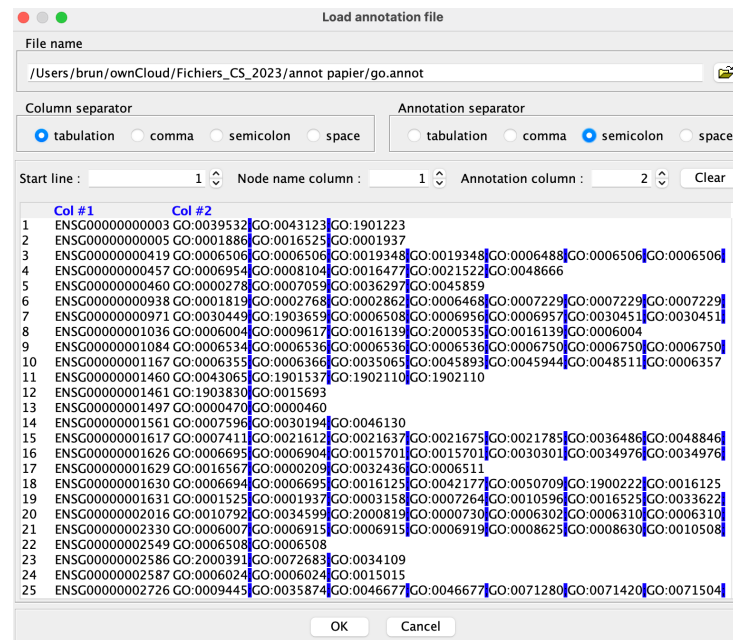
After importing the annotation file, its name is shown in the left panel.
Click the black arrow button to proceed with the annotation.
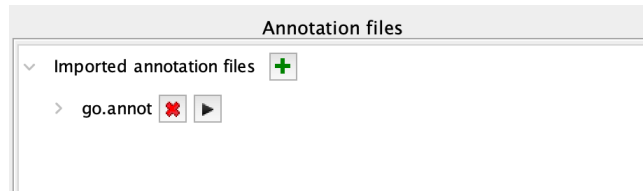
Clust&See3.0 then provides the number and percentage of annotated nodes in the network.
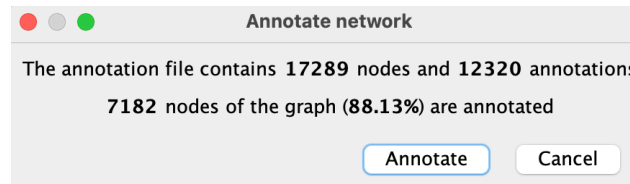
Then, the annotation of the clusters can be performed and the annotation rules applied.

4.4 Cluster & Annotation analyses

The "C&S Partition" table shows all clusters under the "Cluster list" tab.
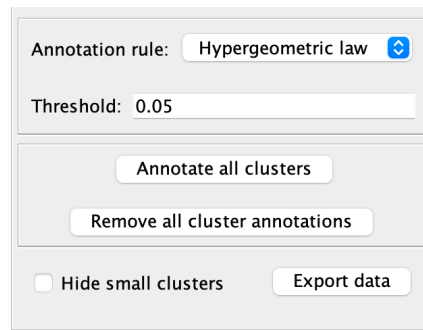
The button "Hypergeometric law" / "Majority rule" indicated the rule chosen for annotating the clusters, with their custom thresholds.
The button "Hide small clusters" hide the lines corresponding to clusters containing less than 5 nodes.

Under the "Cluster Analysis" and the "Annotation term analysis" tabs, the button allow choosing the cluster and the annotation term to analyze, respectively.

The "Cluster annotation matrix" provides all the values of the chosen computations for all clusters in a matrix format, for further analyses.

4.5 Exiting Clust&See3.0

To stop using Clust&See3.0 you can simply use the menu "Plugins -> Clust&See-> Stop". This will close all the windows and views related to your work with Clust&See3.0. All the results and customization you may have produced will be lost.
Note that saving the Cytoscape session will not save the Clust&See3.0 environment. In order to save your work, you have to save your partition by exporting it to file. You will then be able to re-use the exported partition in a later session.
To have details on how to export results, see the section below "Exporting/Importing partitions".

5. Exporting/importing partitions

5.1 Exporting partitions to file

Clust&See3.0 partitions, annotations and computation results can be exported to text files, by clicking on the "Export" buttons available at each step of the analysis. Partitions, annotations and computations are exported along with their parameters (network name, scope, chosen algorithm...). These details are easily readable as headers at the beginning of the file. Global partition statistics (number and cardinality of each cluster) and cluster annotations are also exported. Partition exported files have the ".cns" extension.

5.2 Importing partitions from file

Partitions can be imported into Clust&See3.0 if they are described in a suitable file format (see below). Files imported as Clust&See partitions into Cytoscape can come from the Clust&See3.0 export partition function (see above) but also from files built by external partition tools or scripts. Currently, the R package "Linkcomm" (Kalinka and Tomancak, 2011, http://cran.r-project.org/web/packages/linkcomm/index.html), in which the LinkCommunities (Ahn et al., 2010) and OCG (Becker et al., 2012) algorithms are implemented, provides output files of the produced partitions that can be imported into Clust&See3.0.
If you want to visualize the result of your own partitioning algorithm into Clust&See3.0 to take advantage of its visualization features, let your tool or script produces a file describing the partition in the format described below.

In order to import a partition, load the network used to produce the partition into Cytoscape using standard load features, then click the "Import Partition" action in the "Plugins->Clust&See" menu and select the appropriate partition file (.cns file). The imported partition will behave exactly as if it was produced by Clust&See3.0 algorithms.

5.3 Partition file format

In order to be imported into Clust&See3.0, a partition must be described in a file with the ".cns" extension, containing the information described below. First, we will describe the required information, then we will detail the optional information.

5.3.1 Required information

The partition file must start with the following line:

```
#ClustnSee analysis export
```

The following lines must describe the clusters composing the partition. The first line of a cluster description starts with the string ">ClusterID:", followed by the cluster ID followed by the characters "||" (the cluster ID must be an integer). Any annotations the cluster has follow on the same line and are separated by the "||" character. Finally, the names of the nodes that form the cluster are listed below, one node name per line. For instance, the cluster with ID "1", consisting of nodes "nodeA", "nodeB" and "nodeC", and annotated with "Function1" and "Function2", will be described as follows:

```
>ClusterID:1||Function1||Function2||
nodeA
nodeB
nodeC
```

Annotations are optional. If the cluster has no annotations, its description will be:

```
>ClusterID:1||
nodeA
nodeB
nodeC
```

A partition's clusters are given one after the other in no particular order. What follows is an example of a minimal partition description (two clusters with 3 nodes each and no annotation):

```
#ClustnSee analysis export

>ClusterID:1||
MYC_HUMAN
SMAD3_HUMAN
SNW1_HUMAN

>ClusterID:2||
B4GT3_HUMAN
SAT1_HUMAN
CA103_HUMAN
```

5.3.2 Optional information

Clust&See3.0 can also import meta-data associated with clusters. These meta-data include:
- o The algorithm name. This information can be provided by the header "`#Algorithm:`" (followed by the algorithm name)
- o The parameters of the algorithm. Each parameter can be provided by the header "`#Parameter:`" followed by the parameter name, the equals sign ("=") and the parameter value.
- o Scope of the partition can be provided by the header "`#Scope:`" and followed by any string describing the used scope. If a sub-network has been used to produce the partition, it is possible to provide the list of the used nodes and edges thanks to the headers "`#NodeInScope:`" and "`#EdgeInScope:`" (one node per line and one edge per line).

The above information will be imported in Clust&See3.0 and associated with the partition. It will be available in the "Information" tab next to the partition's "Result" tab. Moreover, it is possible to insert arbitrary information in the file using the comment sign "#". Those lines (if not Clust&See3.0 headers) will be ignored during the import.

Below you will find complete examples:

Export of partition produced by the 'TFit' algorithm with parameter 'alpha=1' on the whole "LCIN_clean.gr" network:

```
#ClustnSee analysis export
#Algorithm:TFit
#Network:LCIN_clean.gr
#Scope:Network
#Parameter:alpha=1.0
```

```
>ClusterID:1||
MYC_HUMAN
SMAD3_HUMAN
SNW1_HUMAN

>ClusterID:2||
B4GT3_HUMAN
SAT1_HUMAN
CA103_HUMAN
```

Export of partition produced by 'MyAlgo' algorithm with parameters 'MyParam1=1.0' and 'MyParam2=centered' on a sub-network of the "LCIN_clean.gr" network:

```
#ClustnSee analysis export
#Algorithm:MyAlgo
#Network:LCIN_clean.gr
#Scope:Selection
#Parameter:MyParm1=1.0
#Parameter:MyParm2=centered
#NodeInScope:MYC_HUMAN
#NodeInScope:SMAD3_HUMAN
#NodeInScope:SNW1_HUMAN
#NodeInScope:B4GT3_HUMAN
#NodeInScope:SAT1_HUMAN
#NodeInScope:CA103_HUMAN
#EdgeInScope:MYC_HUMAN SMAD3_HUMAN
#EdgeInScope:SMAD3_HUMAN SNW1_HUMAN
#EdgeInScope:MYC_HUMAN SNW1_HUMAN
#EdgeInScope:B4GT3_HUMAN SAT1_HUMAN
#EdgeInScope:B4GT3_HUMAN CA103_HUMAN
#EdgeInScope:SAT1_HUMAN CA103_HUMAN
#EdgeInScope:MYC_HUMAN B4GT3_HUMAN

>ClusterID:1||
MYC_HUMAN
SMAD3_HUMAN
SNW1_HUMAN

>ClusterID:2||
B4GT3_HUMAN
```

```
SAT1_HUMAN
CA103_HUMAN
```

6. Extra functions

Clust&See offers several extra functions that are described below.

6.1 Search for Node Clusters

This feature is available from the menu Plugins->Clust&See. The user can enter a node name and retrieve the list of the clusters the node belongs to. This list is not restricted to a specific partition. If several partitions have been produced on the same network, the list of clusters retrieved from a node will contain clusters from the various partitions.
Each entry in this cluster list can be selected, causing the corresponding cluster and metanode to be selected in the currently open views and in the Results Panel, making it easy to locate them.
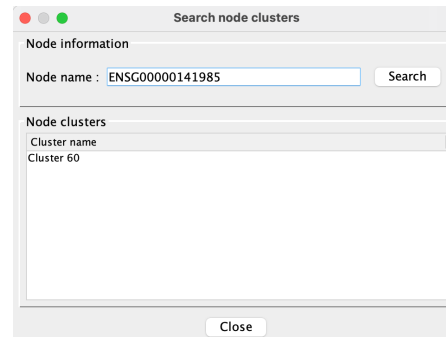

6.2 Compare Partitions

This function is available from the menu Plugins->Clust&See. It can compare two partitions computed on the same network by analyzing the dispersion of the nodes between the clusters of the two partitions. The Jaccard index between the partitions is computed and a contingency table is displayed. In this table, the columns are the clusters of the first partition and the rows are the clusters of the second. The number at the intersection of a column and a row corresponds to the number of nodes shared by the two corresponding clusters.

19

6.3 Build Neighborhood Network

As described below, Clust&See3.0 warns the user if the network/selection used for a partition analysis is too large. Launching an analysis on such a network/selection may lead, at best, to unmanageable results (too many clusters) and, at worst, to memory issues or very long computation times.

Clust&See3.0 offers the user the choice of extracting sub-networks of interest from the complete network. To do so, the user can use the "Build Neighborhood Network" feature available from the Plugins->Clust&See menu.
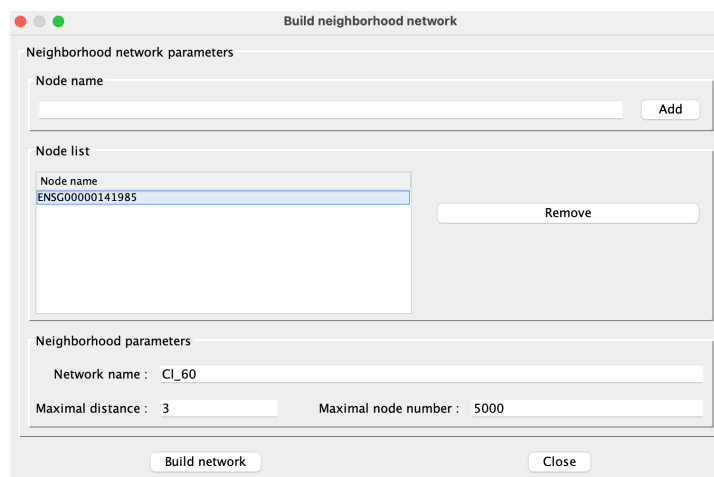
One or several nodes can be chosen as a starting point. The sub-network will then be extended to the nodes directly connected to the chosen nodes and, subsequently, to the nodes directly connected to those and so on until a user-defined distance limit is reached.

The user can specify:
 * The maximal distance between the nodes of the final sub-network and the starting nodes (i.e. the maximal degrees of separation allowed between a node of the sub-network and the starting nodes).
 * The maximal number of nodes the final sub-network may contain.
 * The name of the final sub-network.

The process stops as soon as one of these conditions is met (max number of nodes exceeded or maximal distance reached) and a view is created showing the resulting sub-network. The user can then keep this network, or change the parameters and try again.

Partition analyses can be carried out on the resulting sub-network in exactly the same way as on the parent network.

References

[1] Alain Guénoche (2011). Consensus of Partitions: a Constructive Approach, *Advances in Data Analysis and Classification* 5(3):215-229. DOI: 10.1007/s11634-011-0087-6

[2] Philippe Gambette & Alain Guénoche (2011). Bootstrap Clustering for Graph Partitioning, *RAIRO-Operations Research* 45(4):339-352. DOI: 10.1051/ro/2012001

[3] Emmanuelle Becker, Benoît Robisson, Charles E. Chapple, Alain Guénoche & Christine Brun (2012). Multifunctional Proteins Revealed by Overlapping Clustering in Protein Interaction Network, *Bioinformatics* 28(1):84-90. DOI: 10.1093/bioinformatics/btr621

[4] Alex T. Kalinka & Pavel Tomancak (2011). Linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type, *Bioinformatics* 27(14):2011-2012. DOI: 10.1093/bioinformatics/btr311