Module Manager

Manual v1

daniel.rovera@gmail.com

Team Computational Systems Biology of Cancer
Institut Curie 26 rue d'Ulm 75005 Paris

Contents

Module manager is an application of Cytoscape v3.x.
Module manager allows creating modular views of large networks, i.e. a more compact network, without loosing any detail of the initial network, the reference network. Identifier are the names which are used to match nodes between the reference network and the modules (and not SUID as in Cytoscape).
The basic function are completed by information functions about the link with the reference network and a function to group nodes according to a defined distance matrix.
Several functions useful for modularization are also available in Cytoscape as merge, self-loop...

⚠ In module manager, nodes are identified by their names and not by their ID as in Cytoscape 3 functions. Be careful to avoid synonyms.

1 Menu Group 1

1.1 Create Network of Modules

This function creates a new network from a list of networks. In Cytoscape, networks are selected in the network panel on the left (shift and control for multiple selections). A modular network is created without edges. Each node represents a module pointing to the nested network (right click on node, Nested network, Go to Nested Network).

⚠ It is better to name module/subnetwork differently than simple nodes.

1.2 Create Edges between Modules >All Edges From Reference

From the reference network, its all edges are copied in the modular network, their names are in previous_edges column. Edges included in module edges with become self loops. Self loop can be deleted by a function of edition menu. The link between in module nodes and reference nodes is made by the shared name, so same node names must be avoided. (see figure 1).

⚠ The existence of same nodes in several modules multiplies the number of edges. Two functions 2.1 1.5 below give the comparison between modular network and reference network.

1.3 Create Edges between Modules >Similar Edges Compacted

This function creates edges linking modules from all the edges of the selected network. The links are simplified, if rigth and left are interaction name. No distinction is made between left and right (molecule flow). There is no duplication if two interactions are the same. So similar edges become one edge. (see figure 2).
Same warning than 1.2

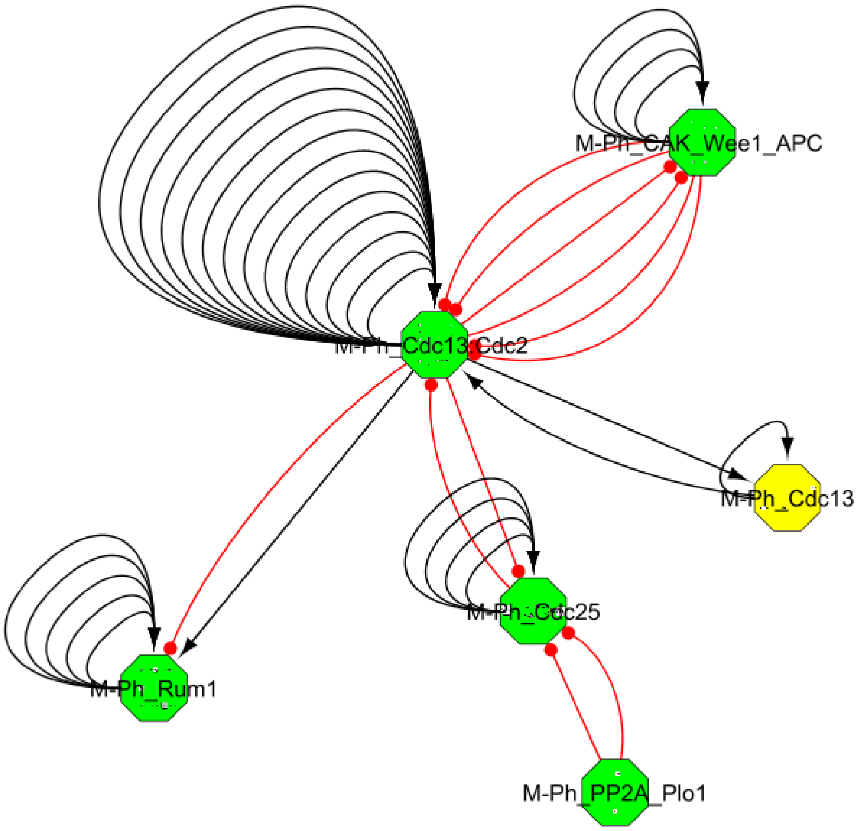


Figure 1: After creating a network of modules, modules are linked by edges from reference network. All edges are kept.

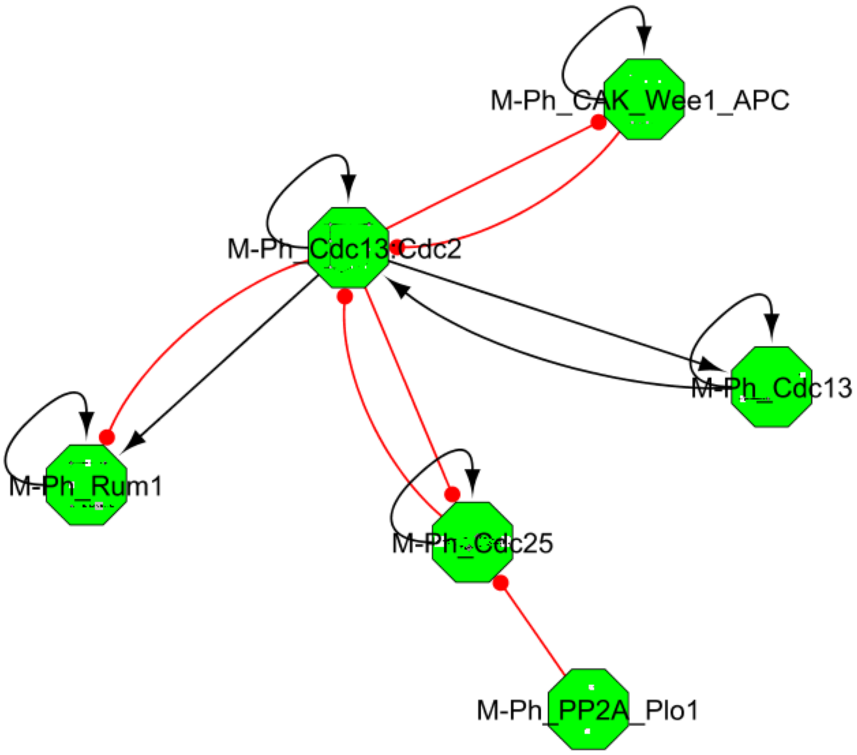


Figure 2: After creating a network of modules, modules are linked by edges from reference network. Only distinct edges are kept (rigth an left become molecule flow).

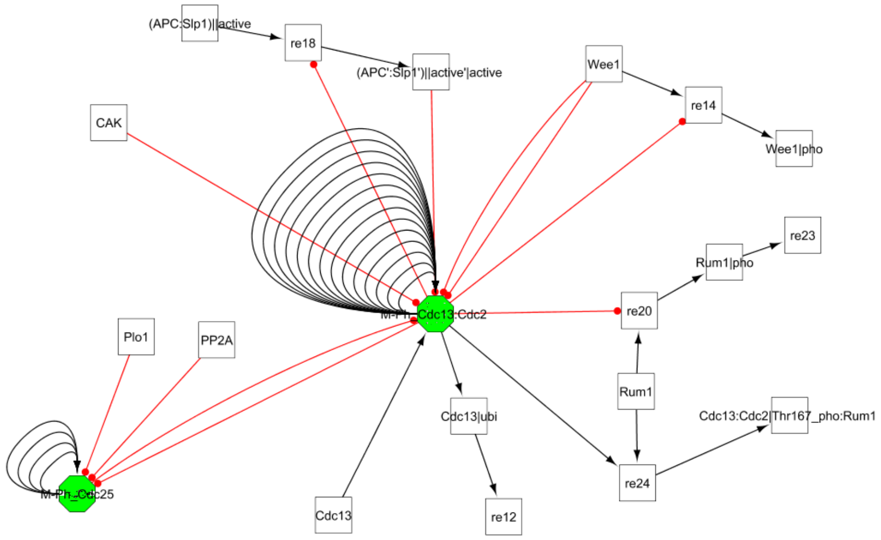


Figure 3: After cloning the network, nodes inside sub-network are packed in modules. Other nodes and all edges are kept.

1.4 Pack Network In Modules

This function packs nodes in modules. Modules are the networks selected in network panel and the network to pack is selected in a list box. It is better to pack a copy of the reference network to keep it intact. All edges are kept and duplicated nodes give the same effect than creating edges. Modules are placed in the barycenter of the nodes which are incuded in them. (see figure 3).
Same warning than 1.2

1.5 Display Warning From Reference

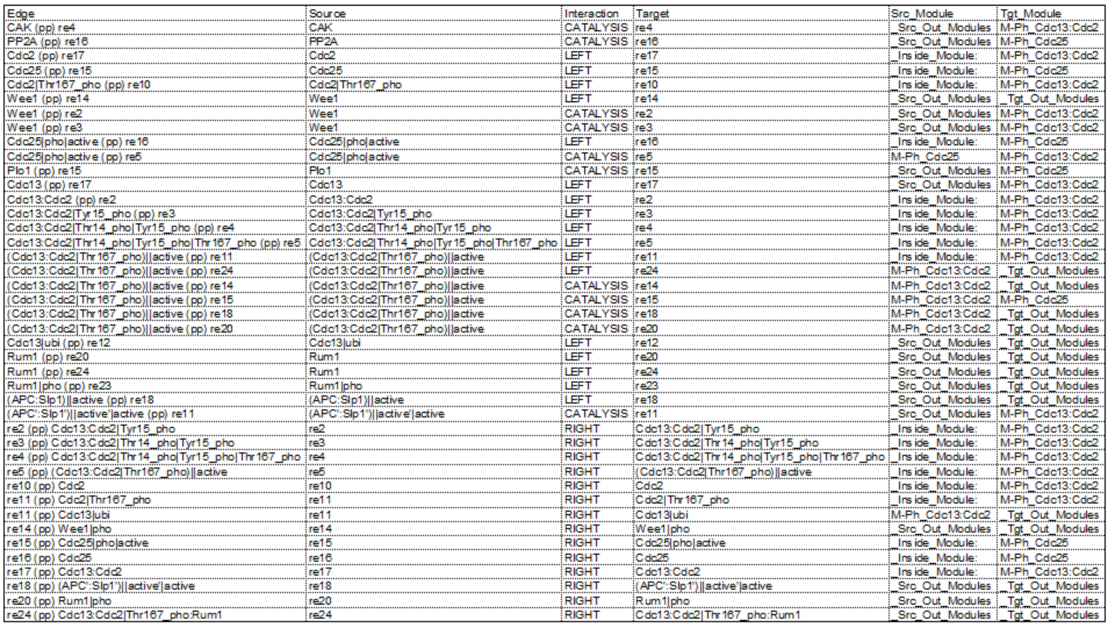
To check the modularization, messages including node or module name are displayed in these cases :

- A node belong to several modules
- A node is in a module but not in the reference network
- A node is in the reference network but not in modules

2 Menu Group 2

2.1 List Nodes & Within Nodes

This function lists the nodes of the global network and nodes included in modules. The results in the text box can be simply copied in a spreadsheet through the clipboard.



Edge	Source	Interaction	Target	Src_Module	Tgt_Module
CAK (pp) re4	CAK	CATALYSIS	re4	_Src_Out_Modules	M-Ph_Cdc13:Cdc2
PP2A (pp) re16	PP2A	CATALYSIS	re16	_Src_Out_Modules	M-Ph_Cdc25
Cdc2 (pp) re17	Cdc2	LEFT	re17	_Inside_Module:	M-Ph_Cdc13:Cdc2
Cdc25 (pp) re15	Cdc25	LEFT	re15	_Inside_Module:	M-Ph_Cdc25
Cdc2|Thr167_pho (pp) re10	Cdc2|Thr167_pho	LEFT	re10	_Inside_Module:	M-Ph_Cdc13:Cdc2
Wee1 (pp) re14	Wee1	LEFT	re14	_Src_Out_Modules	_Tgt_Out_Modules
Wee1 (pp) re2	Wee1	CATALYSIS	re2	_Src_Out_Modules	M-Ph_Cdc13:Cdc2
Wee1 (pp) re3	Wee1	CATALYSIS	re3	_Src_Out_Modules	M-Ph_Cdc13:Cdc2
Cdc25|pho|active (pp) re16	Cdc25|pho|active	LEFT	re16	_Inside_Module:	M-Ph_Cdc25
Cdc25|pho|active (pp) re5	Cdc25|pho|active	CATALYSIS	re5	M-Ph_Cdc25	M-Ph_Cdc13:Cdc2
Plo1 (pp) re15	Plo1	CATALYSIS	re15	_Src_Out_Modules	M-Ph_Cdc25
Cdc13 (pp) re17	Cdc13	LEFT	re17	_Src_Out_Modules	M-Ph_Cdc13:Cdc2
Cdc13:Cdc2 (pp) re2	Cdc13:Cdc2	LEFT	re2	_Inside_Module:	M-Ph_Cdc13:Cdc2
Cdc13:Cdc2|Tyr15_pho (pp) re3	Cdc13:Cdc2|Tyr15_pho	LEFT	re3	_Inside_Module:	M-Ph_Cdc13:Cdc2
Cdc13:Cdc2|Thr14_pho|Tyr15_pho (pp) re4	Cdc13:Cdc2|Thr14_pho|Tyr15_pho	LEFT	re4	_Inside_Module:	M-Ph_Cdc13:Cdc2
Cdc13:Cdc2|Thr14_pho|Tyr15_pho|Thr167_pho (pp) re5	Cdc13:Cdc2|Thr14_pho|Tyr15_pho|Thr167_pho	LEFT	re5	_Inside_Module:	M-Ph_Cdc13:Cdc2
(Cdc13:Cdc2|Thr167_pho)||active (pp) re11	(Cdc13:Cdc2|Thr167_pho)||active	LEFT	re11	_Inside_Module:	M-Ph_Cdc13:Cdc2
(Cdc13:Cdc2|Thr167_pho)||active (pp) re24	(Cdc13:Cdc2|Thr167_pho)||active	LEFT	re24	M-Ph_Cdc13:Cdc2	_Tgt_Out_Modules
(Cdc13:Cdc2|Thr167_pho)||active (pp) re14	(Cdc13:Cdc2|Thr167_pho)||active	CATALYSIS	re14	M-Ph_Cdc13:Cdc2	_Tgt_Out_Modules
(Cdc13:Cdc2|Thr167_pho)||active (pp) re15	(Cdc13:Cdc2|Thr167_pho)||active	CATALYSIS	re15	M-Ph_Cdc13:Cdc2	M-Ph_Cdc25
(Cdc13:Cdc2|Thr167_pho)||active (pp) re18	(Cdc13:Cdc2|Thr167_pho)||active	CATALYSIS	re18	M-Ph_Cdc13:Cdc2	_Tgt_Out_Modules
(Cdc13:Cdc2|Thr167_pho)||active (pp) re20	(Cdc13:Cdc2|Thr167_pho)||active	CATALYSIS	re20	M-Ph_Cdc13:Cdc2	_Tgt_Out_Modules
Cdc13|ubi (pp) re12	Cdc13|ubi	LEFT	re12	_Src_Out_Modules	_Tgt_Out_Modules
Rum1 (pp) re20	Rum1	LEFT	re20	_Src_Out_Modules	_Tgt_Out_Modules
Rum1 (pp) re24	Rum1	LEFT	re24	_Src_Out_Modules	_Tgt_Out_Modules
Rum1|pho (pp) re23	Rum1|pho	LEFT	re23	_Src_Out_Modules	_Tgt_Out_Modules
(APC:Slp1)||active (pp) re18	(APC:Slp1)||active	LEFT	re18	_Src_Out_Modules	_Tgt_Out_Modules
(APC':Slp1')||active'|active (pp) re11	(APC':Slp1')||active'|active	CATALYSIS	re11	_Src_Out_Modules	M-Ph_Cdc13:Cdc2
re2 (pp) Cdc13:Cdc2|Tyr15_pho	re2	RIGHT	Cdc13:Cdc2|Tyr15_pho	_Inside_Module:	M-Ph_Cdc13:Cdc2
re3 (pp) Cdc13:Cdc2|Thr14_pho|Tyr15_pho	re3	RIGHT	Cdc13:Cdc2|Thr14_pho|Tyr15_pho	_Inside_Module:	M-Ph_Cdc13:Cdc2
re4 (pp) Cdc13:Cdc2|Thr14_pho|Tyr15_pho|Thr167_pho	re4	RIGHT	Cdc13:Cdc2|Thr14_pho|Tyr15_pho|Thr167_pho	_Inside_Module:	M-Ph_Cdc13:Cdc2
re5 (pp) (Cdc13:Cdc2|Thr167_pho)||active	re5	RIGHT	(Cdc13:Cdc2|Thr167_pho)||active	_Inside_Module:	M-Ph_Cdc13:Cdc2
re10 (pp) Cdc2	re10	RIGHT	Cdc2	_Inside_Module:	M-Ph_Cdc13:Cdc2
re11 (pp) Cdc2|Thr167_pho	re11	RIGHT	Cdc2|Thr167_pho	_Inside_Module:	M-Ph_Cdc13:Cdc2
re11 (pp) Cdc13|ubi	re11	RIGHT	Cdc13|ubi	M-Ph_Cdc13:Cdc2	_Tgt_Out_Modules
re14 (pp) Wee1|pho	re14	RIGHT	Wee1|pho	_Src_Out_Modules	_Tgt_Out_Modules
re15 (pp) Cdc25|pho|active	re15	RIGHT	Cdc25|pho|active	_Inside_Module:	M-Ph_Cdc25
re16 (pp) Cdc25	re16	RIGHT	Cdc25	_Inside_Module:	M-Ph_Cdc25
re17 (pp) Cdc13:Cdc2	re17	RIGHT	Cdc13:Cdc2	_Inside_Module:	M-Ph_Cdc13:Cdc2
re18 (pp) (APC':Slp1')||active'|active	re18	RIGHT	(APC':Slp1')||active'|active	_Src_Out_Modules	_Tgt_Out_Modules
re20 (pp) Rum1|pho	re20	RIGHT	Rum1|pho	_Src_Out_Modules	_Tgt_Out_Modules
re24 (pp) Cdc13:Cdc2|Thr167_pho:Rum1	re24	RIGHT	Cdc13:Cdc2|Thr167_pho:Rum1	_Src_Out_Modules	_Tgt_Out_Modules

Figure 4: List of edges, their main features and what they link

2.2 List Edges Linking Modules

This function displays all edges linking modules or nodes, their interaction and sources and targets whether it is a module or a node in an array copyable in a spreadsheet. The column title are Edge, Source, Interaction, Target, Src_Module, Tgt_Module. According to the extremities of edges, these messages are displayed in column:

- _Src_Out_Modules in Src_Module column, if source is not inside a module
- _Tgt_Out_Modules in Tgt_Module column, if target is not inside a module
- _Inside_Module: followed by the name of a module where edge is inside

(see table 4).

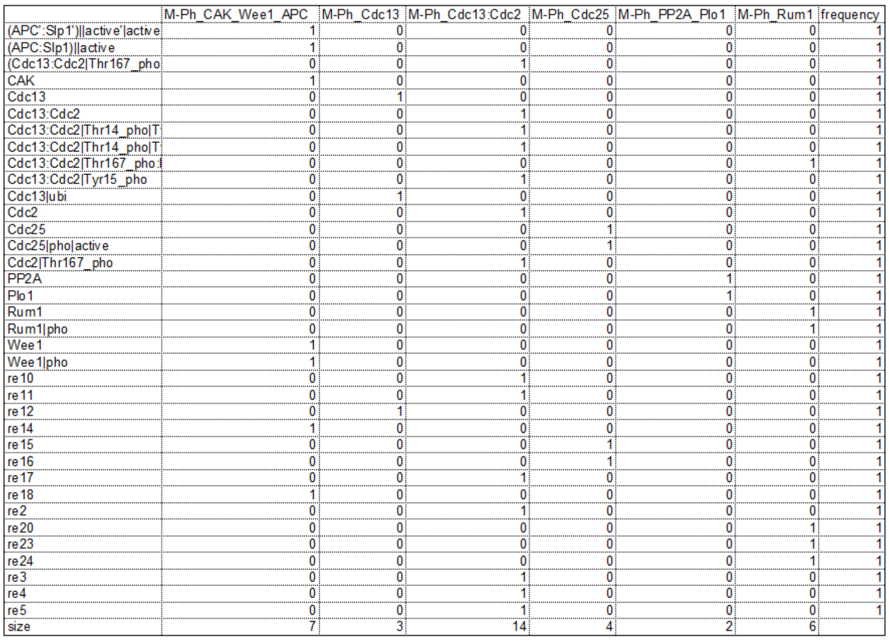
2.3 Find Common Nodes In Modules

Display in a text box the matrix of nodes (modules in columns, nodes in rows, size of modules in last row, frequency in modules in last column). The result is more easily usable after copying it in a spreadsheet. (see table 5).

2.4 Select Nodes by a Name List

When viewing a network, open a text editor where list of node names can be pasted or typed, one name by line. Selection of nodes of this network by clicking select.

⚠ Insure that is the good network in title of editor.



	M-Ph_CAK_Wee1_APC	M-Ph_Cdc13	M-Ph_Cdc13:Cdc2	M-Ph_Cdc25	M-Ph_PP2A_Plo1	M-Ph_Rum1	frequency
(APC':Slp1')||active'|active	1	0	0	0	0	0	1
(APC:Slp1)||active	1	0	0	0	0	0	1
(Cdc13:Cdc2|Thr167_pho	0	0	1	0	0	0	1
CAK	1	0	0	0	0	0	1
Cdc13	0	1	0	0	0	0	1
Cdc13:Cdc2	0	0	1	0	0	0	1
Cdc13:Cdc2|Thr14_pho|T	0	0	1	0	0	0	1
Cdc13:Cdc2|Thr14_pho|T	0	0	1	0	0	0	1
Cdc13:Cdc2|Thr167_pho:	0	0	0	0	0	1	1
Cdc13:Cdc2|Tyr15_pho	0	0	1	0	0	0	1
Cdc13|ubi	0	1	0	0	0	0	1
Cdc2	0	0	1	0	0	0	1
Cdc25	0	0	0	1	0	0	1
Cdc25|pho|active	0	0	0	1	0	0	1
Cdc2|Thr167_pho	0	0	1	0	0	0	1
PP2A	0	0	0	0	1	0	1
Plo1	0	0	0	0	1	0	1
Rum1	0	0	0	0	0	1	1
Rum1|pho	0	0	0	0	0	1	1
Wee1	1	0	0	0	0	0	1
Wee1|pho	1	0	0	0	0	0	1
re10	0	0	1	0	0	0	1
re11	0	0	1	0	0	0	1
re12	0	1	0	0	0	0	1
re14	1	0	0	0	0	0	1
re15	0	0	0	1	0	0	1
re16	0	0	0	1	0	0	1
re17	0	0	1	0	0	0	1
re18	1	0	0	0	0	0	1
re2	0	0	1	0	0	0	1
re20	0	0	0	0	0	1	1
re23	0	0	0	0	0	1	1
re24	0	0	0	0	0	1	1
re3	0	0	1	0	0	0	1
re4	0	0	1	0	0	0	1
re5	0	0	1	0	0	0	1
size	7	3	14	4	2	6	

Figure 5: Matrix of nodes: modules in columns, nodes in rows, size of modules in last row, frequency in modules in last column.

2.5 Assign Module Names to Node Attribute

Function to create a node attribute (named as the modular network), containing module names. This attribute may be used to visualize modules in the reference network.

3 Menu Group 3

3.1 Neighborhood Around Selected Nodes

In first, a table of oriented nearnesses (reverse of distances) must be imported, sources as column title and targets as rows in the column shared name. All shared names of nodes in the network must be present in rows.
The column title is built as ON_nodeSharedName as ON_Wee1 (ON as oriented nearness). The nearness from source to target is a real number (double).
The nearness is defined by the aim of modularization: correlation, influence weight, reverse of path length...
The pivot nodes are selected as sources and the function put in a column nearByN the nearest target nodes (N is a number to distinguish several result). If nearness=0, the result is _far_ .

⚠ If a source node is absent in nearness table, it cannot be used as pivot node.

3.2 Transfer Network Coordinates

Transfer node coordinates from a network to another network using names to match nodes.

3.3 List Components of Species in Network and Modules

Function to list the different components of a given species only when their names must respect BiNoM syntax 5.

3.4 Create Module Style

Create a style useful to view a modular network which can be adapted according to the attributes.

4 Example of modularization

The file to modularize is M_Phase in the Cytoscape session ModuleManagerExample. The different steps are:

1. Identify the stoichiometric cycle Cdc25 and CdC13:Cdc2 (by strong connected component on an only right-left edge network)
2. Clone the network M_Phase and pack it into these 2 modules to see what is not inside modules
3. Identify nodes to include in different new modules
4. Select them in reference network and create sub-networks, here: PP2A_Plo1, Cdc13, CAK_Wee1_APC and Rum1
5. Create a network of modules from all sub-networks
6. Link modules with two options: all edges and distinct edges compacted
7. Self loops can be removed by Edit menu

5 BiNoM Naming Service

When importing pathway information, BiNoM tries to generate meaningful, unique and short names for index entities. This function of the plugin is performed via BiNoM Naming Service. For proteins and other entities, the shortest available synonym is used. For genes, a g symbol is added at the beginning of the name, and for RNAs, a r symbol is added in order to avoid mixing genes and mRNAs with their products. If this leads to an ambiguity, it is resolved by adding a suffix specifying a unique id of the entity.

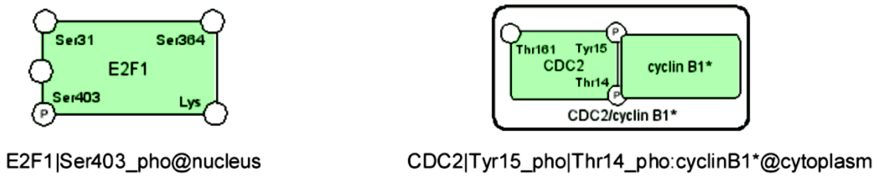


Figure 6: 2 examples of naming chemical species shown in Systems Biology Graphical Notation standard.

A chemical species in BiNoM is defined as a physical entity (such as protein) with some cellular localization and some (post-translational) modification (possibly none). The general template of the species label is the following:

Entity1_name|Modification1|Modification2|:Entity2_name|Modifications...[_active|_hmN]@compartment

Here, the colon symbol : delimitates the different components of a complex if the species has several components. Optional suffixes active or hm describe active state of the chemical species or N-homodimer state, respectively.

This figure 6 shows what chemical species named in Systems Biology Graphical Notation standard become in BiNoM syntax.