BiNoM: Biological Network Manager
Version 1.0 Manual

Andrei Zinovyev, Laurence Calzone, Eric Bonnet, Daniel Rovera, Gautier Stoll

Bioinformatics Unit of Institut Curie, Paris

December 20, 2011

Contents

1 Introduction **2**

2 BiNoM I/O **2**

3 BiNoM Analysis **3**
3.1 Get connected components . . . 3
3.2 Get strongly connected components . . . 3
3.3 Prune Graph . . . 3
3.4 Get Material Components . . . 5
3.5 Get Cycle Decomposition . . . 5
3.6 Path Analysis . . . 5
3.7 Extract subnetwork . . . 7
3.8 Calc centrality, Inbetweenness undirected, Inbetweenness directed . . . 7
3.9 Generate Modular View . . . 7
3.10 Cluster Networks . . . 8
3.11 Mono-molecular react.to edges . . . 8
3.12 Linearize network . . . 12
3.13 Exclude intermediate nodes . . . 12
3.14 Extract Reaction Network . . . 12
3.15 Path consistency analysis . . . 12
3.16 OCSANA analysis . . . 12
3.17 Create neighborhood sets file . . . 12

4 BiNoM Module Manager **13**
4.1 Create network of modules . . . 13
4.2 Create connections between modules . . . 13
4.3 Create modules from networks . . . 13
4.4 Agglomerate the nearest nodes in modules . . . 13
4.5 List nodes of modules and network . . . 13
4.6 Find common nodes in modules . . . 13
4.7 Assign module names to node attribute . . . 14
4.8 List components of species in network and modules . . . 14
4.9 Create network from union of selected modules . . . 14
4.10 Create network from intersection of 2 selected modules . . . 14
4.11 Recreate lost connections inside modules . . . 14
4.12 Destroy networks unused as module . . . 14

5 BiNoM BioPAX3 Utils **15**

6 BiNoM BioPAX3 Query **16**

7 BiNoM Utilities **17**

1 Introduction

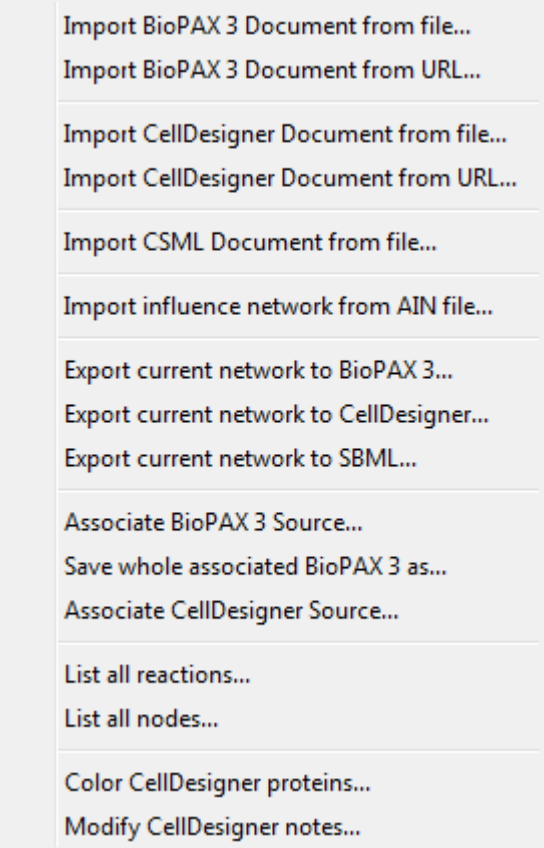
BiNoM (BIological NetwOrk Manager) is a Cytoscape plugin, developed to facilitate the manipulation of biological networks represented in standard systems biology formats and to carry out studies on the network structure. BiNoM provides the user with a complete interface for the analysis of biological networks in Cytoscape environment.

In an effort to exchange and curate pathway database knowledge, several standard formats have been developed (SBML, BioPAX [5] and others). Many softwares, which are centered on the description and representation of biological pathways, adopted these standards. CellDesigner[3] and Cytoscape[4], for instance, allow the visualization and manipulation of networks but meet some limitations. BiNoM was designed to facilitate the use of systems biology standards, the extraction and organization of information from pathway databases through BioPAX interface.

BiNoM concentrates on the following aspects: the import and export of BioPAX and (CellDesigner) SBML files and the conversion between them; the structural analysis of biological networks including decomposition of networks into modules, path analysis, etc.; the BioPAX query engine which provides the extraction of information from huge BioPAX files such as whole pathway databases; and various operations on graphs not offered by Cytoscape such as clipboard operations and comparison of networks

BiNoM plugin with documentation, API and source code is available for download at: http://bioinfo.curie.fr/projects/bi

2 BiNoM I/O



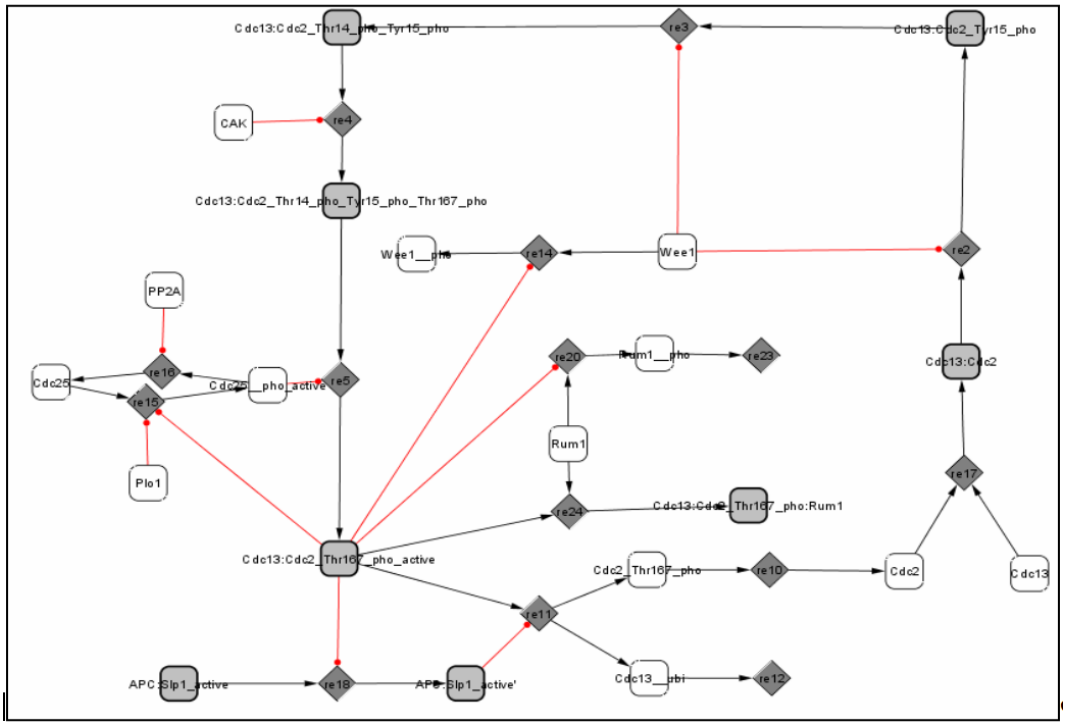


Figure 1: Cytoscape view of the M-Phase network

3 BiNoM Analysis

We illustrate, here, the different functions of BiNoM related to the structural analysis, using the modified version of the Novak et al. model, M-Phase.xml as an example (figure 1).
From the menu Plugins⇒ BiNoM analysis, we review all the functions one by one:

3.1 Get connected components

Plugins⇒BiNoM analysis⇒Get connected components
This command dissociates the unconnected subparts of the network. In our case, since the network is already completely connected, the one obtained when choosing this function is the same as the initial one (called M-Phase.xml_cc1).

3.2 Get strongly connected components

Plugins⇒BiNoM analysis⇒Get strongly connected components

Based on Tarjans algorithm[6], the strongly connected components are isolated. In simple words, the obtained network, M-Phase.xml_scc1(figure 2), insures that there exists a path from one node to another and deletes the components which do not respond to this requirement.

3.3 Prune Graph

Plugins⇒BiNoM analysis⇒Prune graph
Pruning the graph is equivalent to separating the network into three parts(figure 3: what comes in (M-Phase.xml_in), what goes out (M-Phase.xml_out) and the central cyclic part (M-Phase.xml_scc).
This decomposition corresponds to the idea of the bow-tie structure developed by Broder and colleagues[1]. In our example, the central cyclic part is the same as figure 2, the strongly connected component. In other cases, it can be composed from several strongly connected components, connected or disconnected.
The Prune graph operation decomposes the current network into three parts: IN, OUT and SCC (the later can contain several strongly connected components).

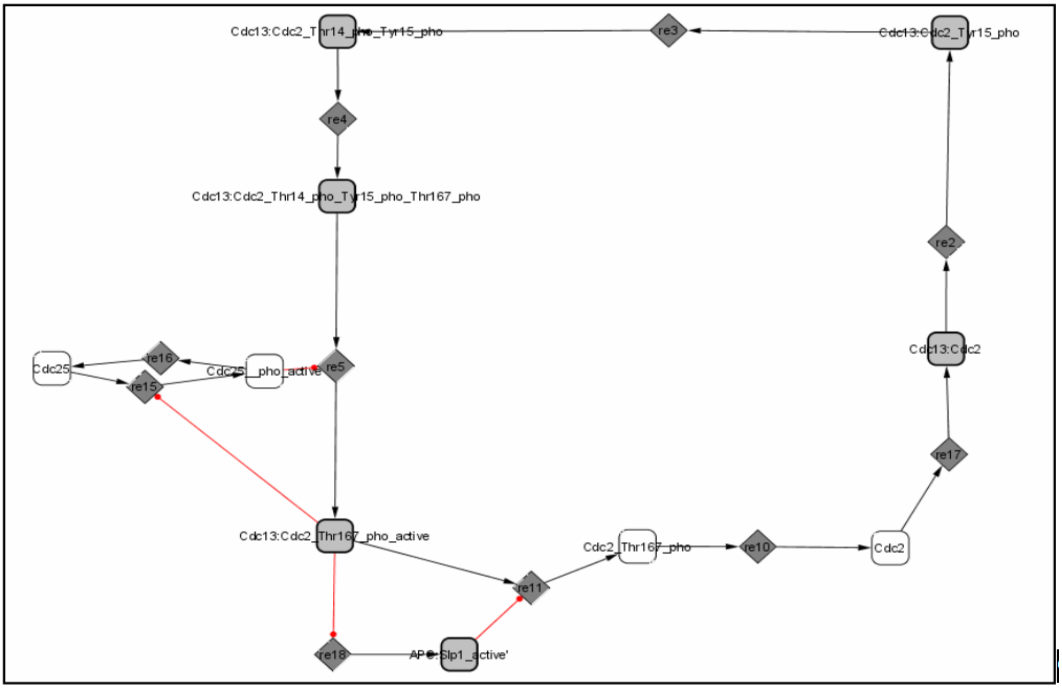


Figure 2: Strongly Connected Component of M-Phase network

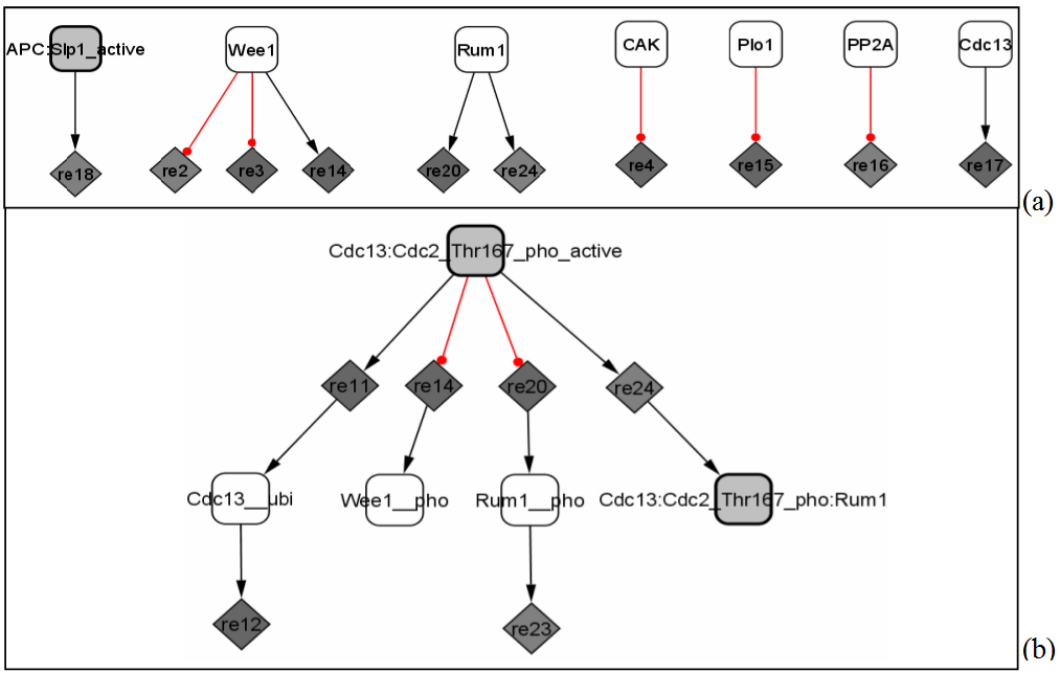


Figure 3: Prune the graph. (a) Incoming flux: molecules involved in the IN part of the network, and (b) Outgoing flux: molecules involved in the OUT part of the network.

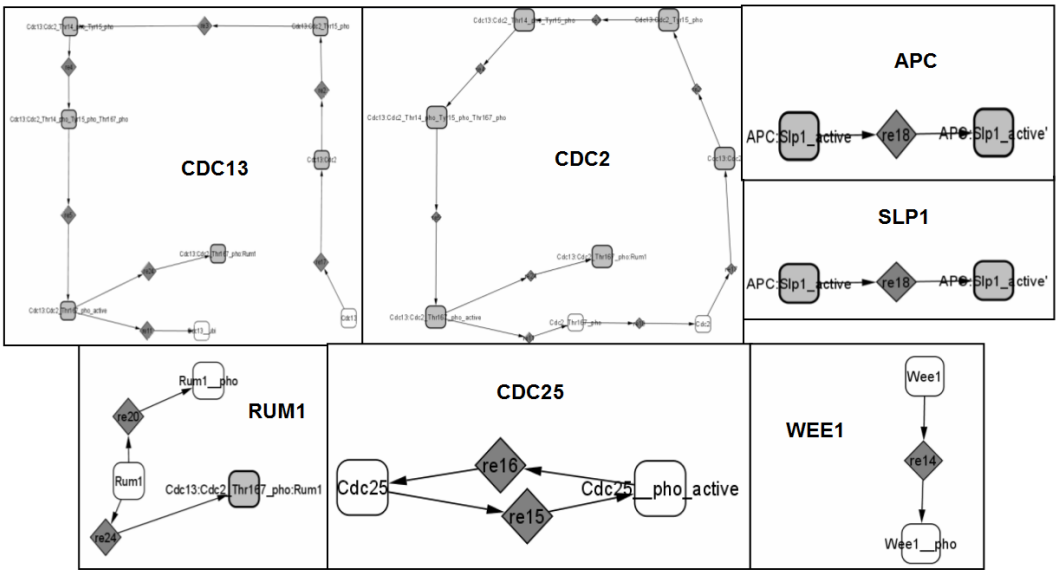


Figure 4: Material Components

3.4 Get Material Components

Plugins⇒BiNoM analysis⇒Get material components
This function uses node name semantics to isolate sub-networks in which each protein takes part. In our example(figure 4), seven sub-networks are created: M-Phase.xml_Cdc13, M-Phase.xml_Cdc2, M-Phase.xml_Rum1, M-Phase.xml_APC, M-Phase.xml_Slp1, M-Phase.xml_Cdc25 and M-Phase.xml_Wee1. Some major overlaps between sub-networks are expected, as it is the case for Cdc2 and Cdc13 which form a complex.

3.5 Get Cycle Decomposition

Plugins⇒BiNoM analysis⇒Get cycle decomposition
This command decomposes the network into relevant directed cycles[2], using a modification of the Vismaras algorithm[7]. Often, this feature gives information about the life cycle of a protein or a complex, about the feedbacks of the studied network, etc(figure 5). Note that the union of all the cycles corresponds to the strongly connected component figure 2.
⚠This operation can produce enormous number of cycles! Therefore it is rather suitable for analysis of small to moderate size networks. For a big network, one can start to understand the cyclic network structure by eliminating first the network hubs, which are contained in many network cycles. After that, the local, relatively short, cycles can be represented as meta-nodes (modules) and the analysis for cycles can be repeated.

3.6 Path Analysis

Plugins⇒BiNoM analysis⇒Path analysis
In a network, it can become handy to find out if there exists a path (or paths) from one species to another, or to verify that a protein or a protein complex is reachable from a starting molecule(figure 7). Provided (an) initial source and target protein(s) that are selected first on the graph then in the dialog window, the command Path analysis can find: the shortest paths, the optimal and suboptimal shortest paths, or all the non-intersecting paths (does not include inner loops), using a finite number of intermediary nodes (use finite breadth search radius), for either directed or undirected paths (figure 6).
⚠In big networks the number of paths can be exponential! It is recommended to find the shortest path first, take its length and increment gradually the breadth search radius starting from this value to find the second shortest, third shortest, etc., paths.

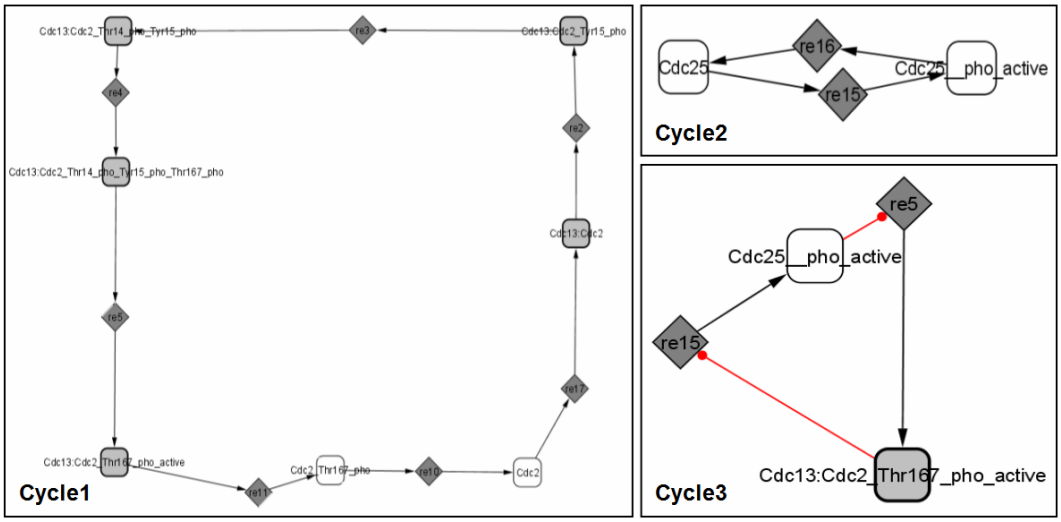


Figure 5: Minimal cycle decomposition of the M-Phase network. Cycle 1 includes CDC2 and CDC13 proteins, Cycle 2 CDC25 and Cycle 3 shows the feedback existing between CDC13/CDC2 and CDC25.

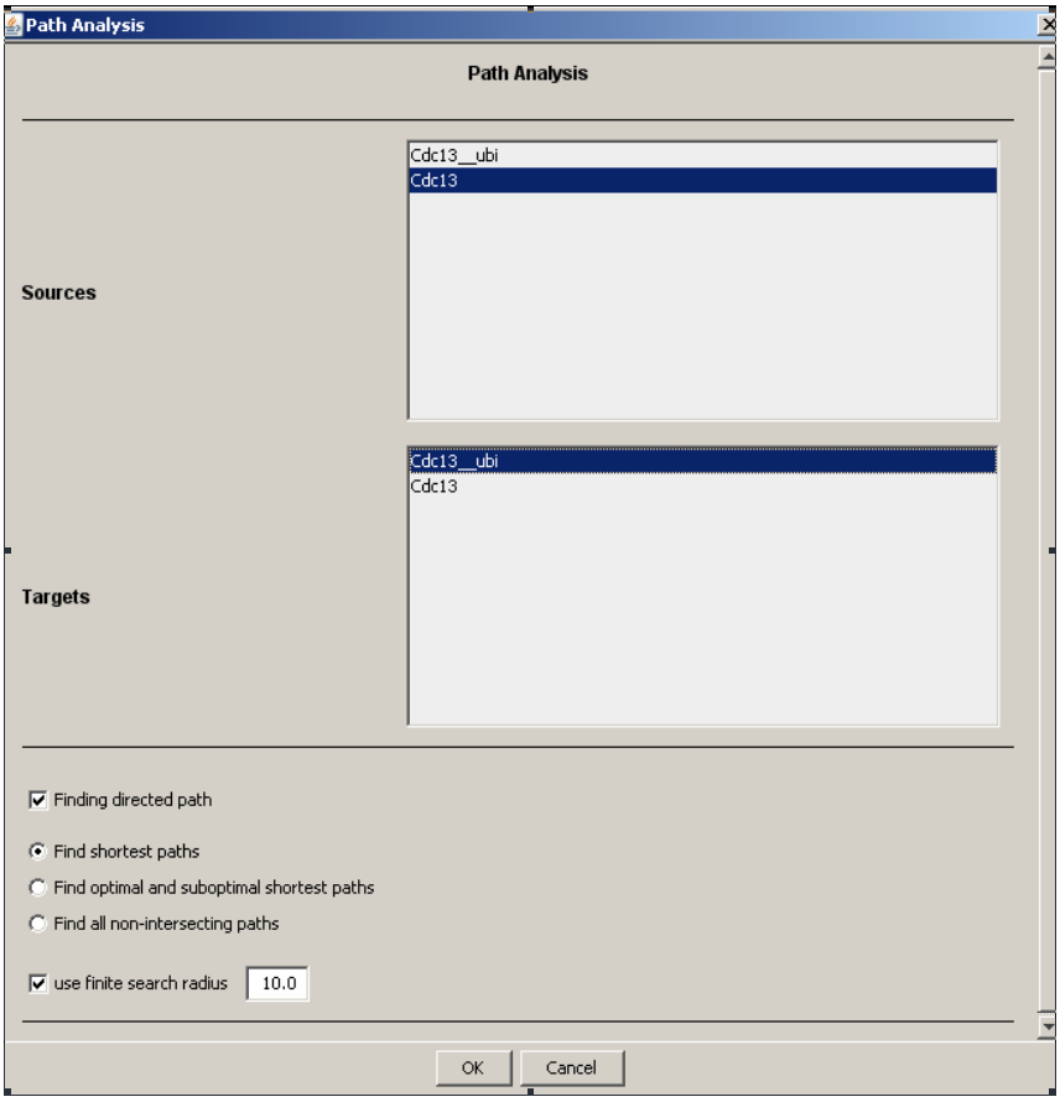


Figure 6: BiNoM Path Analysis: Pop-up window in which the source(s) and the target(s) need to be specified along with the type of paths (shortest, optimal shortest or all paths).

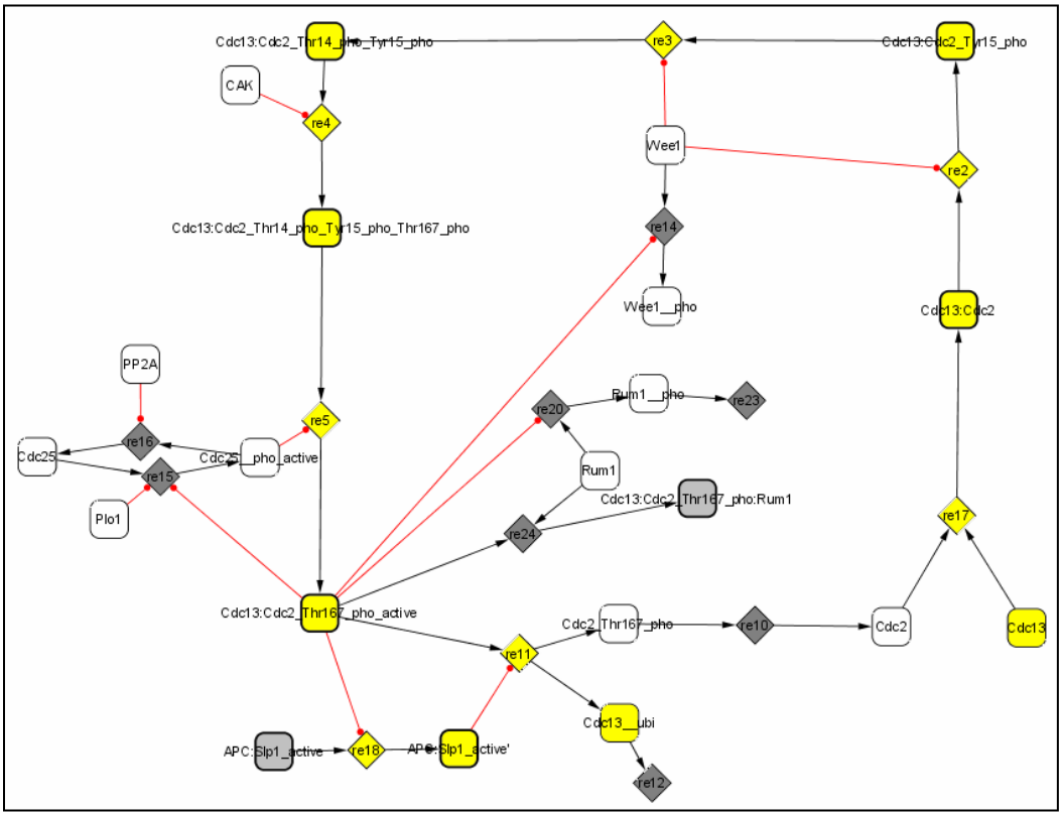


Figure 7: Path Analysis: All the paths leading from one molecular species (Cdc13) to another (Cdc13_ubi, ubiquitinated form of Cdc13) are highlighted in yellow.

3.7 Extract subnetwork

3.8 Calc centrality, Inbetweenness undirected, Inbetweenness directed

3.9 Generate Modular View

Plugins⇒BiNoM⇒analysis⇒Generate modular view
Given the initial diagram and some modules (which could be sub-networks of the initial network), it is possible to reconstruct a modular view of the network. For our example, we choose the initial network to be M-Phase.xml and the subparts or modules, the seven sub-networks corresponding to the material components described in (4). From these seven sub-networks only six are selected since two of them, Slp1 and APC, are exactly the same.
The sub-networks or modules need to be specified in the creating modular view window (figure 8).

There are different types of modular views. The modules are connected by: (1) the number of shared inter-

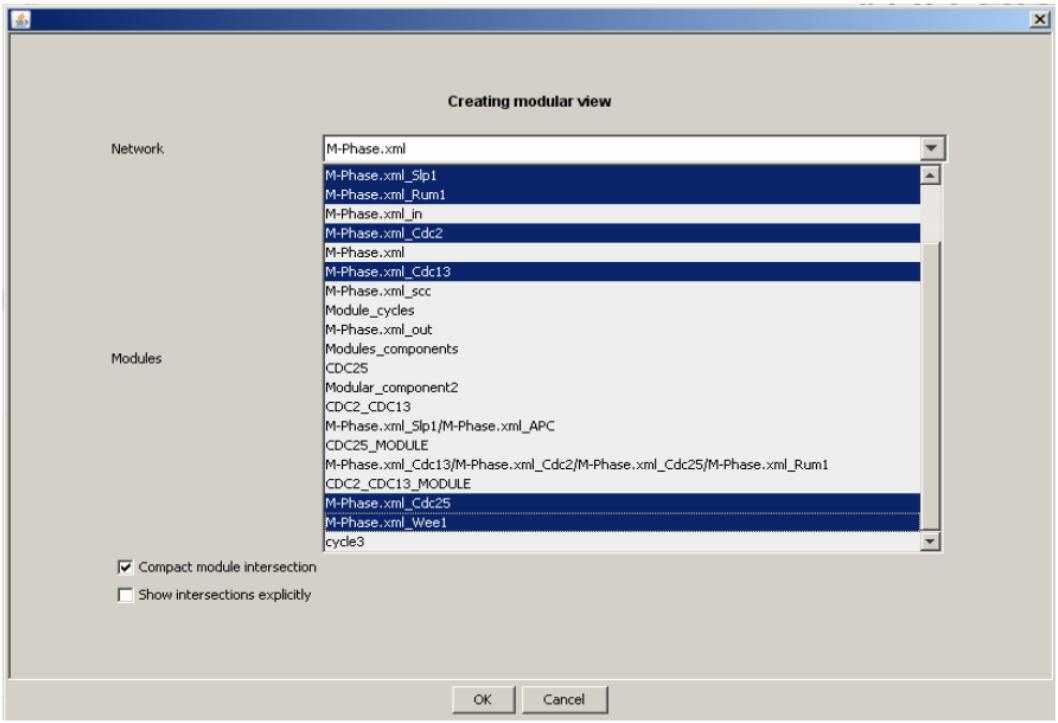


Figure 8: BiNoM modular view of the newtork: Pop-up window in which the initial graph and the modules are specified.

actions (figure 9, upper panel); (2) the number of shared nodes (reactions + species) for which case the box Compact module intersection must be checked (figure 9, middle panel); and (3) the shared nodes and reactions showed explicitly (figure 9, lower panel).

3.10 Cluster Networks

Plugins⇒BiNoM analysis⇒Cluster networks
This command lumps together the modules that share a certain proportion of nodes. At a first glance, it can easily be concluded from Figure **??** (middle panel) that, for example, the modules M-Phase.xml_Cdc13 and M-Phase.xml_Cdc2 share a lot of proteins or protein complexes. Therefore, we can assume that these two modules will collapse into one big module. To determine the clusters, the intersection threshold can be set (from 0 to 100% intersecting components). For a 30% intersection threshold, Figure 10 is obtained. Four clusters of modules were proposed and linked.
An alternative modular view has been obtained using the cycle decomposition instead of the material decomposition. The cycles are presented in Figure 5. They are obtained by clustering the three cycles into two (cycle 1 + cycle2/cycle3) and organized into a modular view (Figure 11).

3.11 Mono-molecular react.to edges

Plugins⇒BiNoM analysis⇒Mono-molecular react. to edges
This command transforms monomolecular (with one reactant and one product) reaction nodes into influence edges. Thus, monomolecular (linear) reactions are represented as edges and the reaction graph is not bi-partite anymore. When the reaction nodes have the type of influence specified (through the EFFECT attribute), the graph is transformed automatically into an influence graph (see Figure 12: upper panel: BioPAX network, lower panel, the equivalent influence network). Non-linear non-monomolecular reactions (such as complex assemblies) are not transformed and remain to be represented as network nodes.

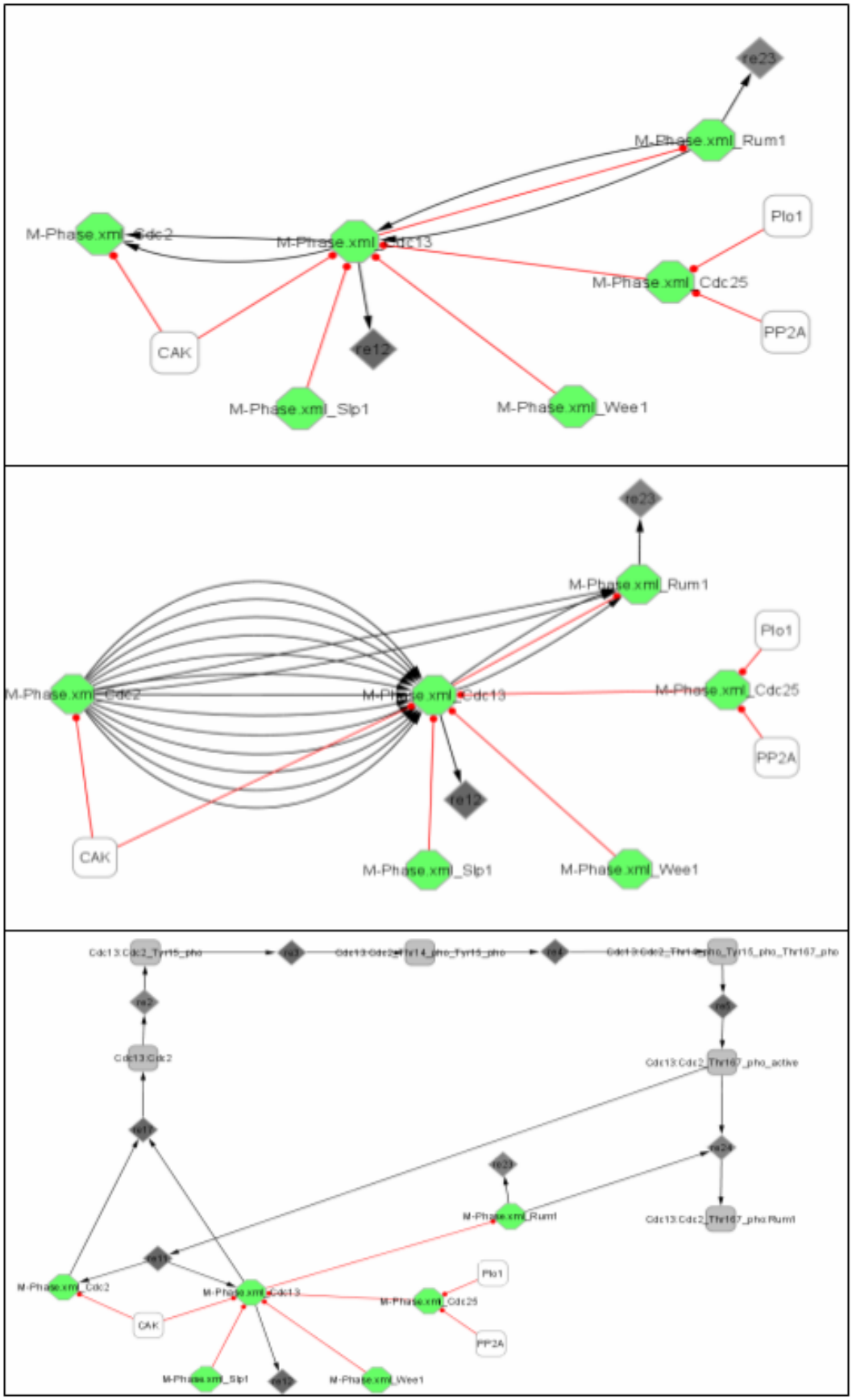


Figure 9: BiNoM modular view of the newtork: The resulting modular network (upper panel) with compact module intersections (middle panel) and with explicit intersections (lower panel).

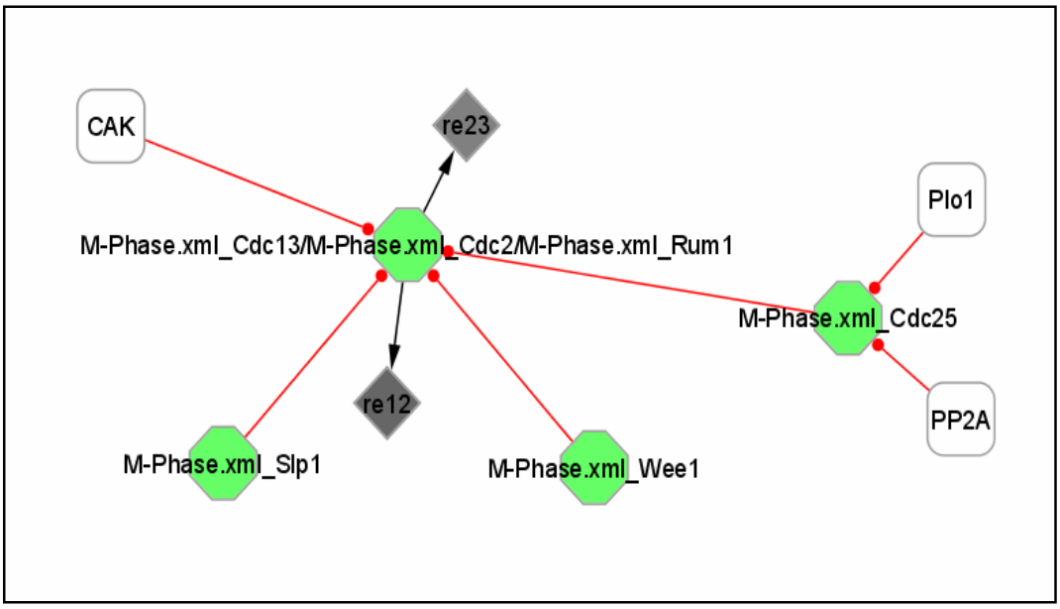


Figure 10: Clusters of modules. The obtained diagram is a compact modular view of the M-Phase network using the material decomposition and material components clustering

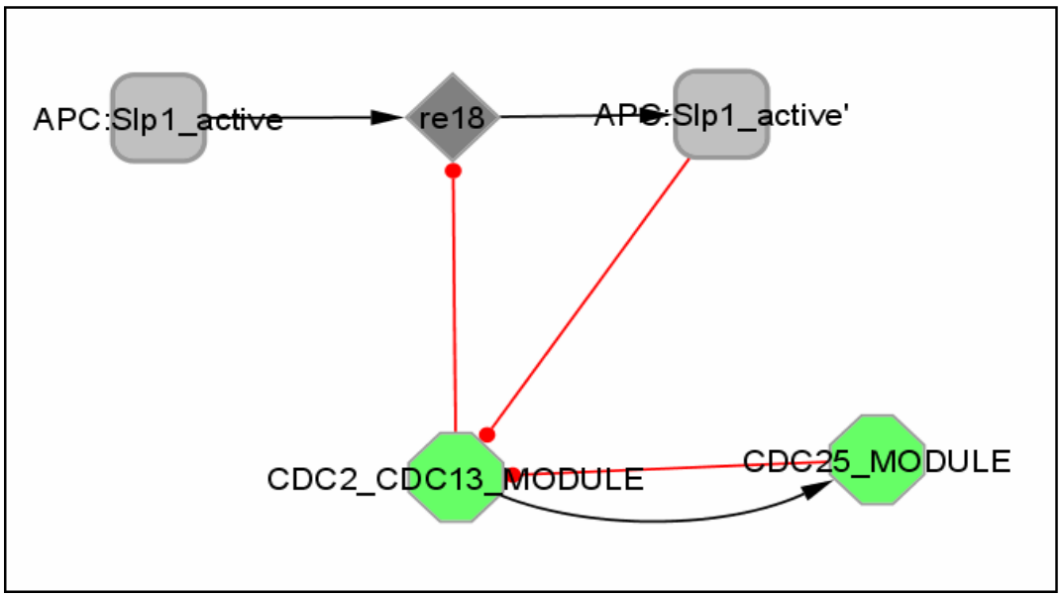


Figure 11: Clusters of modules. The obtained diagram is a compact modular view of the M-Phase network using the relevant cycle decomposition and cycle clustering

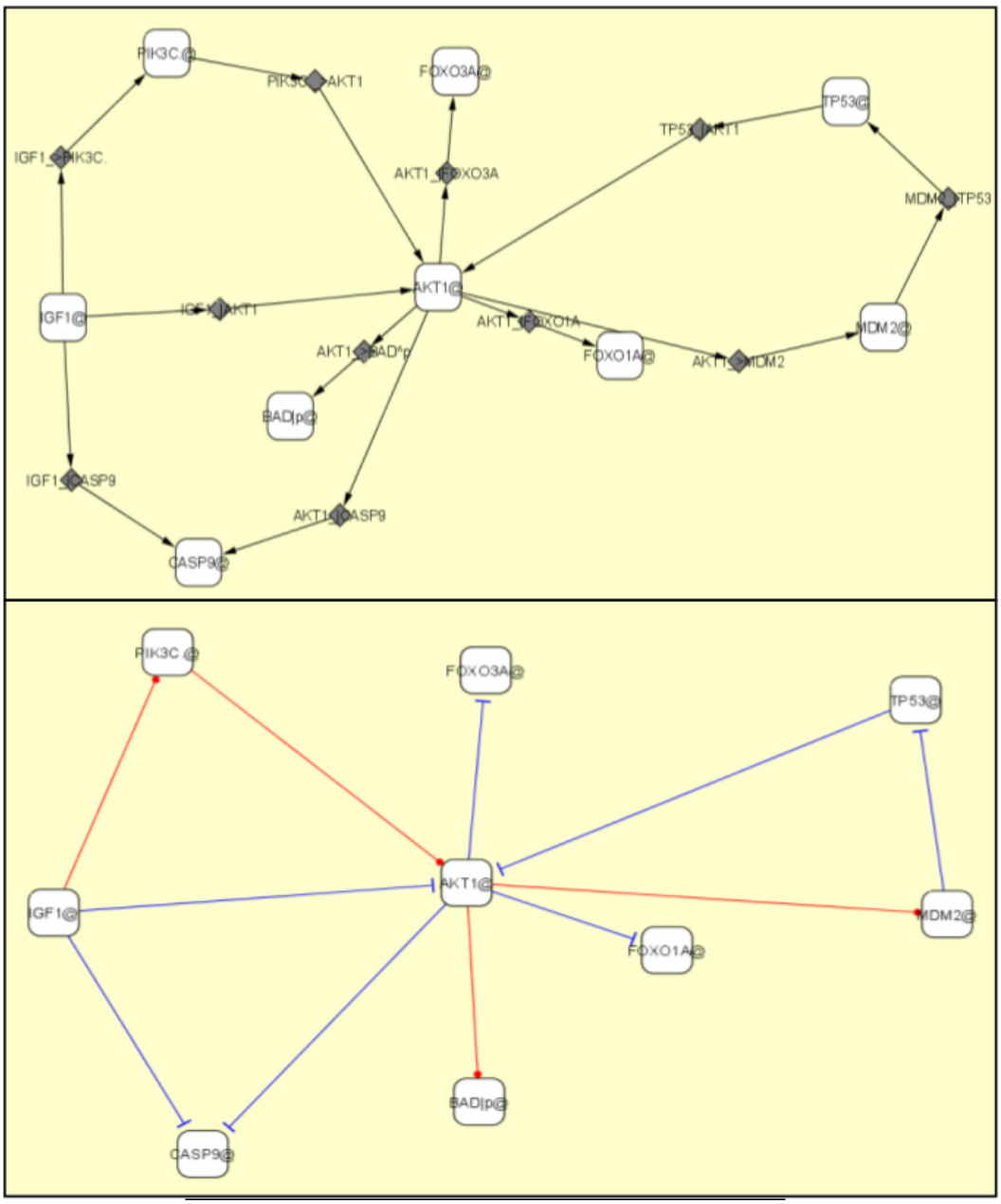


Figure 12: From a BioPAX network (upper panel) to an influence graph (lower panel).

3.12 Linearize network

3.13 Exclude intermediate nodes

3.14 Extract Reaction Network

Plugins⇒BiNoM analysis⇒Extract reaction networks
This function cleans up the diagram to only keep the reaction network. Only nodes with XXXX_REACTION and XXXX_SPECIES attributes (where XXXX stands for any word) are kept as a result of this operation. For example, it helps to clean the reaction network interface from the result of querying BioPAX index (which contains many other node types such as entities and publications.

3.15 Path consistency analysis

3.16 OCSANA analysis

3.17 Create neighborhood sets file

4 BiNoM Module Manager

Module manager is useful for creating modular view of large networks without loosing details of modules (using nest, object of Cytoscape v7 and after).

4.1 Create network of modules

Create a new network from a list of sub-networks (sub-networks are selected in the network list).
Nodes=modules, no edge. Visual style created in VizMapper for module network
⚠ Module names and node names must be different, all network names too.

To go from module to sub-network: select node >Right click >Nested Network >Go to Nested Network.

4.2 Create connections between modules

Create edges linking modules from all edges of the selected network.
Links are simplified, no distinction between left and right (molecule flow), no duplication if same interaction.
Warning message if duplicated or absent nodes (may disturb links).

4.3 Create modules from networks

Create modules in the active network from a list of sub-networks (sub-networks are selected in the network list)
All edges are kept. See edge attribute PREVIOUS_ID for their origin.
The attribute BIOPAX_NODE_TYPE is set to pathway (see visual style BiNoM BioPAX).
⚠ All nodes of sub-networks must be found once in the active network (no intersection between sub-networks).

4.4 Agglomerate the nearest nodes in modules

Create modules and a modular view by agglomerating the nearest nodes in the active network.

Input 2 parameters to get not too big sub-networks containing not too far nodes:

- Maximal distance between nodes or modules in number of edges,
- Maximal number of nodes in modules.

Confirm creation if agree with displayed result (distance, number and size of modules).

4.5 List nodes of modules and network

List nodes of network and nodes included in modules.
Result in text box can be simply copied in a spreadsheet through clipboard.

4.6 Find common nodes in modules

Display in text box the belonging matrix of nodes (modules in columns, nodes in rows, size of modules in last row, frequency in modules in last column); result more easily usable after copying in a spreadsheet.
Create intersection edges with number of common nodes as attribute (COMMON_NODES).
Create node attribute containing the node numbers of modules (NODE_NUMBER).

Module Visual StyleCan be adapted to the wished visual aspect by hands in VizMapper, for example:

- To visualize NODE_NUMBER: double click Node Size, select NODE_NUMBER, continuous mapping, adjust width by graphical view.
- To visualize COMMON_NODES double click Edge Line Width, select COMMON_NODES, continuous mapping, adjust width by graphical view.

4.7 Assign module names to node attribute

Create a node attribute (named as the modular network), containing module names

4.8 List components of species in network and modules

List components of species (their names must respect BiNoM syntax).

4.9 Create network from union of selected modules

Create a network from union of selected modules and its corresponding module in the current network.

4.10 Create network from intersection of 2 selected modules

Create a network from intersection of 2 selected modules and its corresponding module.
Confirm for deleting the common nodes in the selected modules.

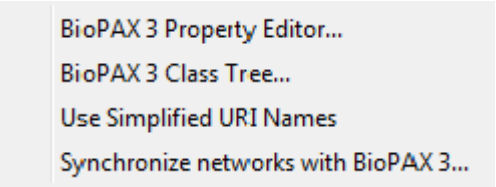
4.11 Recreate lost connections inside modules

Recreate connections inside modules which may have been lost by modularizing operations.

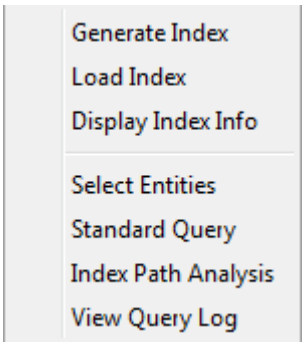
4.12 Destroy networks unused as module

Select networks to be deleted among a list of networks which are not used as modules in the current network (simplify cleaning session)

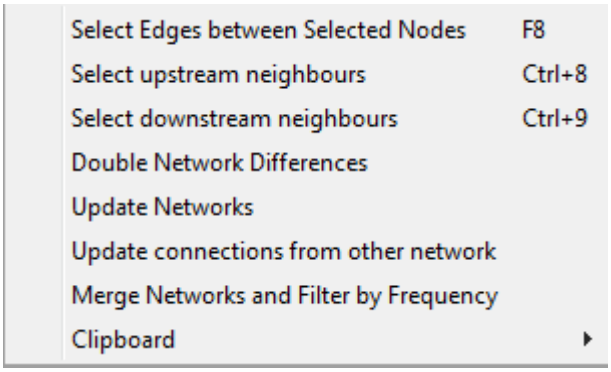
5 BiNoM BioPAX3 Utils



6 BiNoM BioPAX3 Query



7 BiNoM Utilities



References

[1] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer networks*, 33(1-6):309–320, 2000.

[2] P.M. Gleiss, P.F. Stadler, A. Wagner, and D.A. Fell. Relevant cycles in chemical reaction networks. *Advances in complex systems*, 4(2/3):207–226, 2001.

[3] Funahashi A. Matsuoka Y. Kitano, H. and K. Oda. Using process diagrams for the graphical representation of biological networks. *Nat Biotech*, (23):961–966, 2005.

[4] Markiel A. Ozier O. Baliga N. S. Wang J. T. Ramage D. Amin N. Schwikowski B. Shannon, P. and Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*, (13):2498–2504, 2003.

[5] L. Stromback and P. Lambrix. Representations of molecular pathways an evaluation of sbml. *PSI MI and BioPAX. Bioinformatics*, (21):4401–4407, 2005.

[6] R. Tarjan. Depth-first search and linear graph algorithms. pages 114–121, 1972.

[7] P. Vismara. Union of all the minimum cycle bases of a graph. *Electr. J. Comb*, 4(1):73–87, 1997.