BiNoM: Biological Network Manager
Version 2.0 Manual

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

Bioinformatics Unit of Institut Curie, Paris

January 30, 2012

Contents

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 [7] and others). Many softwares, which are centered on the description and representation of biological pathways, adopted these standards. CellDesigner[4] and Cytoscape[6], 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.
This manual describes only the functions of plugin-in BiNoM. Cytoscape proposes some functions close to those of BiNoM (import, export, set operations ...) which are explained in Cytoscape manual (http://www.cytoscape.org).

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

2 BiNoM I/O

2.1 Import BioPAX 3 document

BioPAX level 3 information is fully supported (reaction network, interaction network, pathway structure, annotations).

Plugins⇒BiNoM I/O⇒Import BioPAX 3 Document from file
The model M-Phase.owl[5] is uploaded. A dialog window proposes to create three different interfaces to the BioPAX file: reaction network (RN), pathway structure (PS) and protein interaction (PP).

- Reaction network: M-Phase RN is a representation of the reaction network (figure 1).
- Pathway structure: M-Phase PS represents the pathway hierarchical structure. For this example, we choose to show a more detailed and complete pathway, the apoptosis sub-network extracted from Reactome database (figure 2).
- Protein interaction: M-Phase PP shows which proteins interact with each other.

For more details on BioPAX, its interfaces, etc, go to section 8.4.
In the case of creating the pathway structure interface, several choices are offered:

- Make Root Pathway Node: adds an extra node to which all pathways are connected. This feature can be useful for organizing the graph and joining separate and disjoint pathways.
- Include Next Links: shows the order of the reactions. From a node, an arrow indicates which node is the next step. This feature provides a timeline of the events in a pathway and could emphasize, for example, the linearity of a cascade.
- Include Pathways: includes green nodes (figure 2) which correspond to the names of the different pathways of the network.
- Include interactions: shows explicitly the reactions involved in the pathway (lower grey nodes in figure 2).

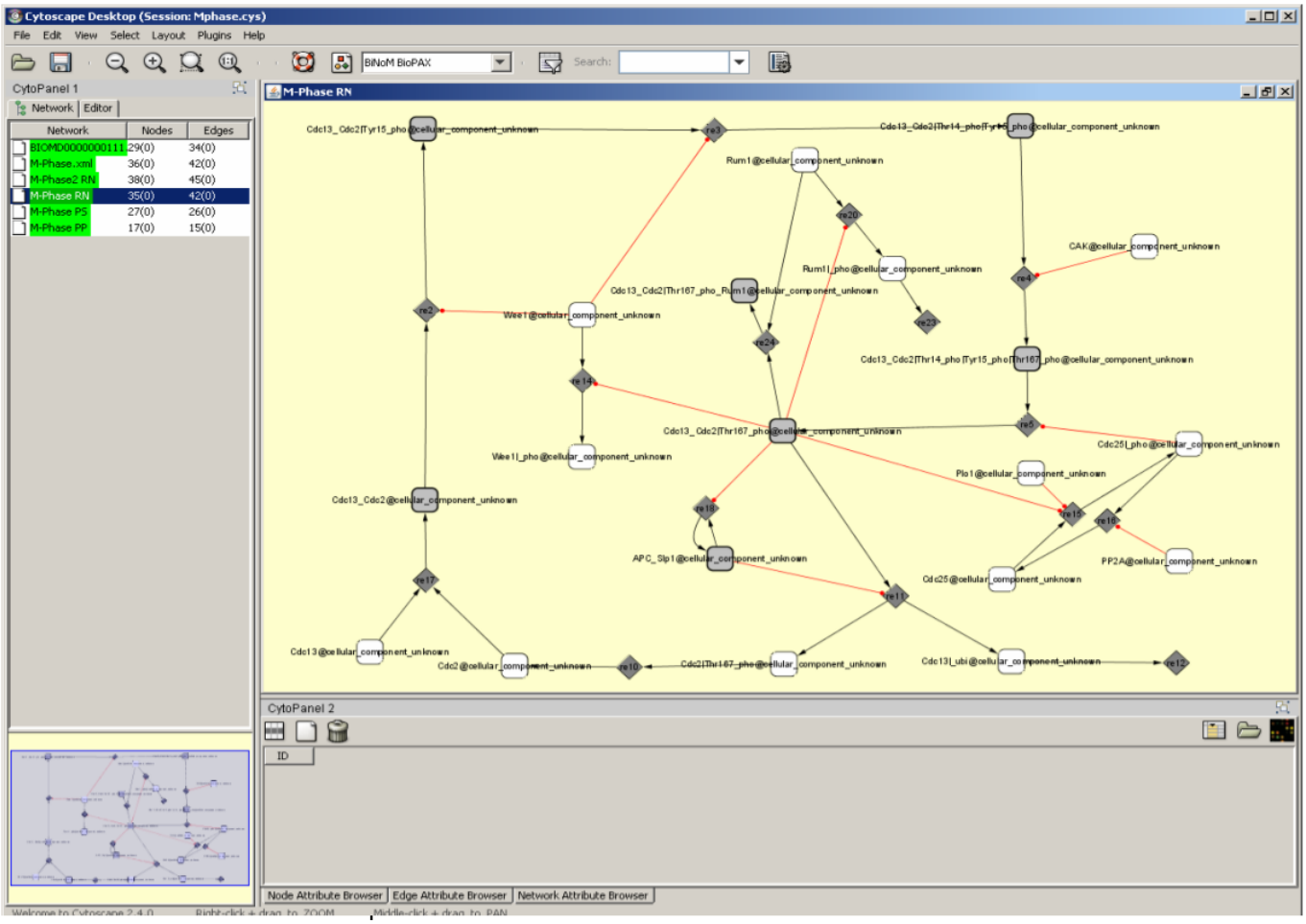


Figure 1: BioPAX view of Novak et al. model.

Plugins⇒BiNoM I/O⇒Import BioPAX 3 Document from URL
A BioPAX 3 document can also be imported directly from a URL. The web address must be typed in the dialog window.

2.2 Import CellDesigner document

Plugins⇒BiNoM I/O⇒Import CellDesigner Document from file
The model can be drawn or downloaded[5] in CellDesigner (figure 3) and saved as M-Phase.xml.
The Import CellDesigner Document from file function imports a model from CellDesigner to Cytoscape. A dialog window opens and M-Phase.xml needs to be selected and imported (figure 4).

Figure 3 and 4 show the same model viewed by CellDesigner and Cytoscape respectively. The layout information from CellDesigner is imported automatically into Cytoscape.

In specie notes in CellDesigner Attribute name:Value as HUGO:E2F1 (without blank) is converted in Cytoscape as the attribute HUGO with the value E2F1 for the specie.

Plugins⇒BiNoM I/O⇒Import CellDesigner Document from URL
A CellDesigner document can also be imported directly from a URL. The web address must be typed in the dialog window.

2.3 Import CSML document

Plugins⇒BiNoM I/O⇒Import CSML document
BiNoM imports a CSML (Cell System Markup Language, csml.org)

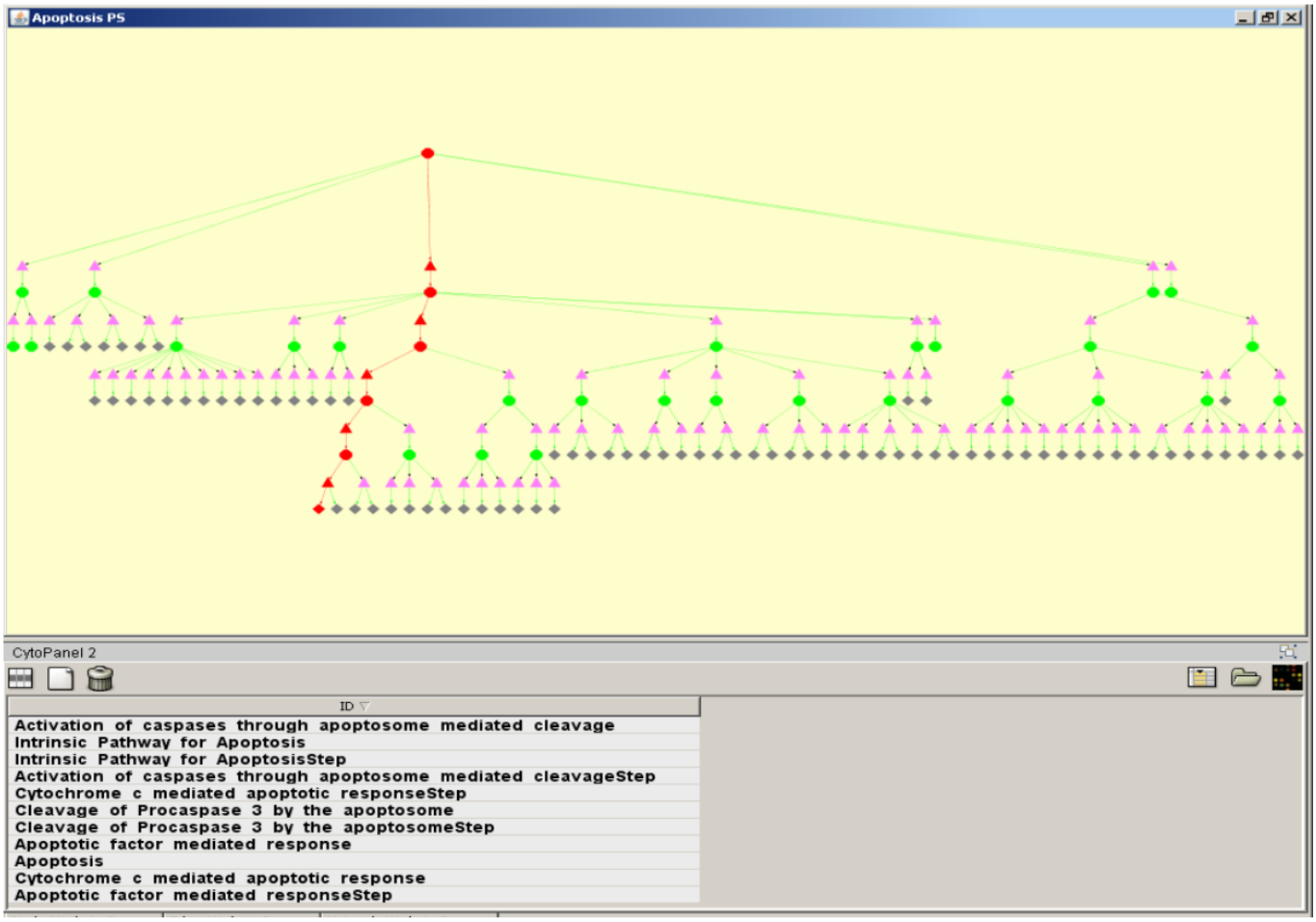


Figure 2: Apoptosis pathway hierarchical structure. Green nodes represent pathways, pink triangular nodes represent steps, and grey nodes represent reactions. From the apoptosis node (top node in red), the cell can choose through 5 different paths. The red-colored path shows one of them, the activation of apoptosis via the intrinsic pathway, leading to the cleavage of caspases 3.

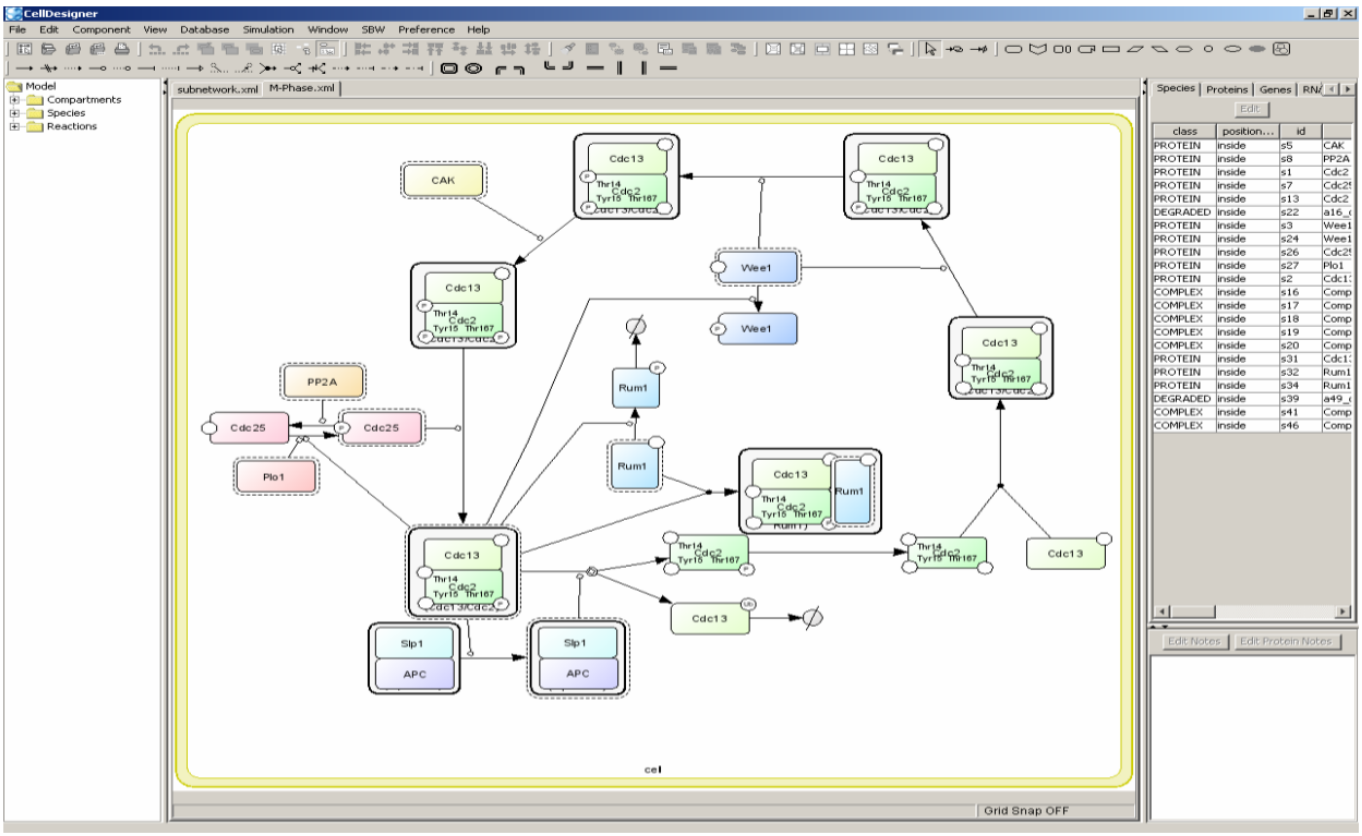


Figure 3: CellDesigner view of the cell division cycle model of fission yeast[5]

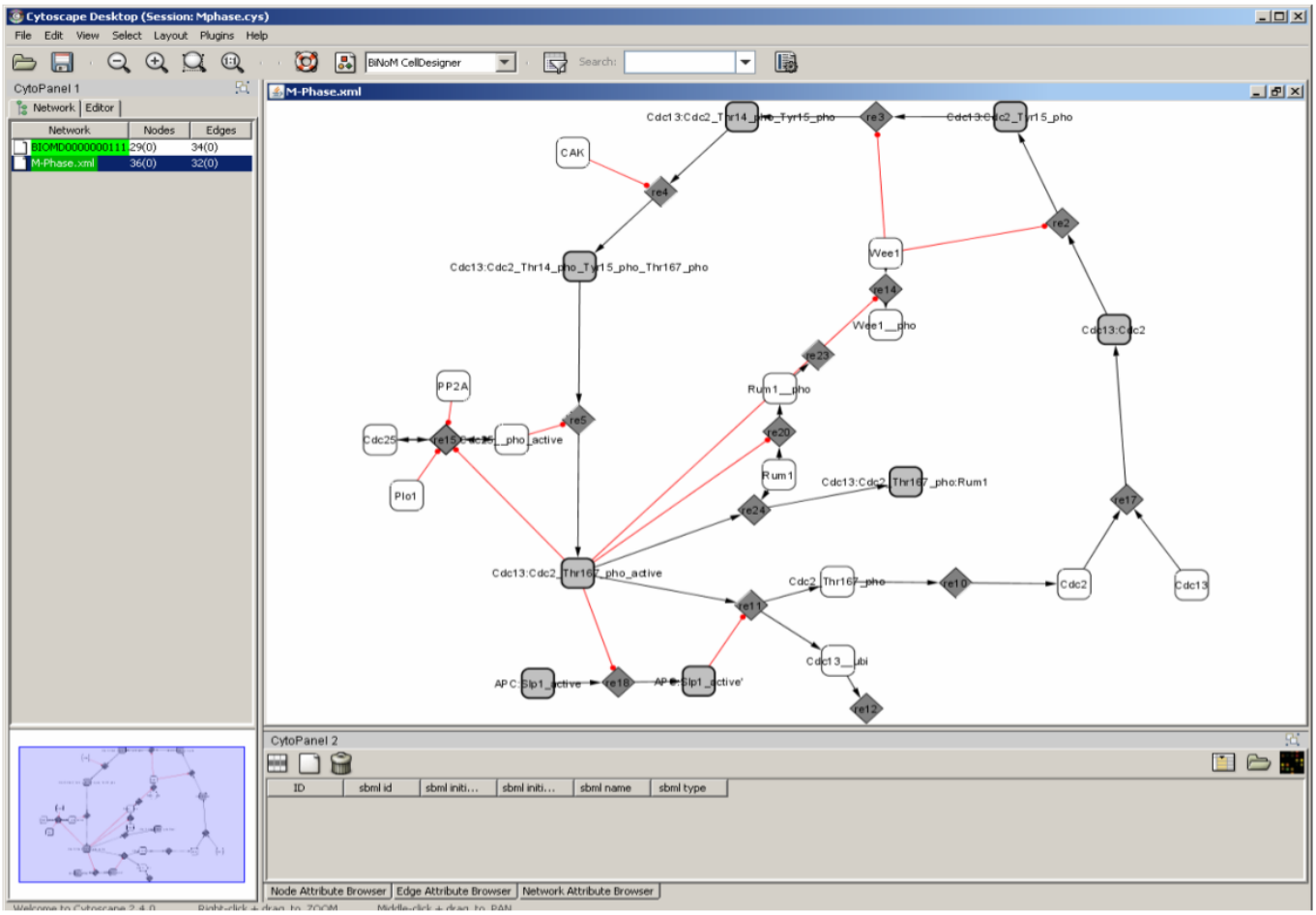


Figure 4: Cytoscape view of the cell division cycle model of fission yeast from a CellDesigner document

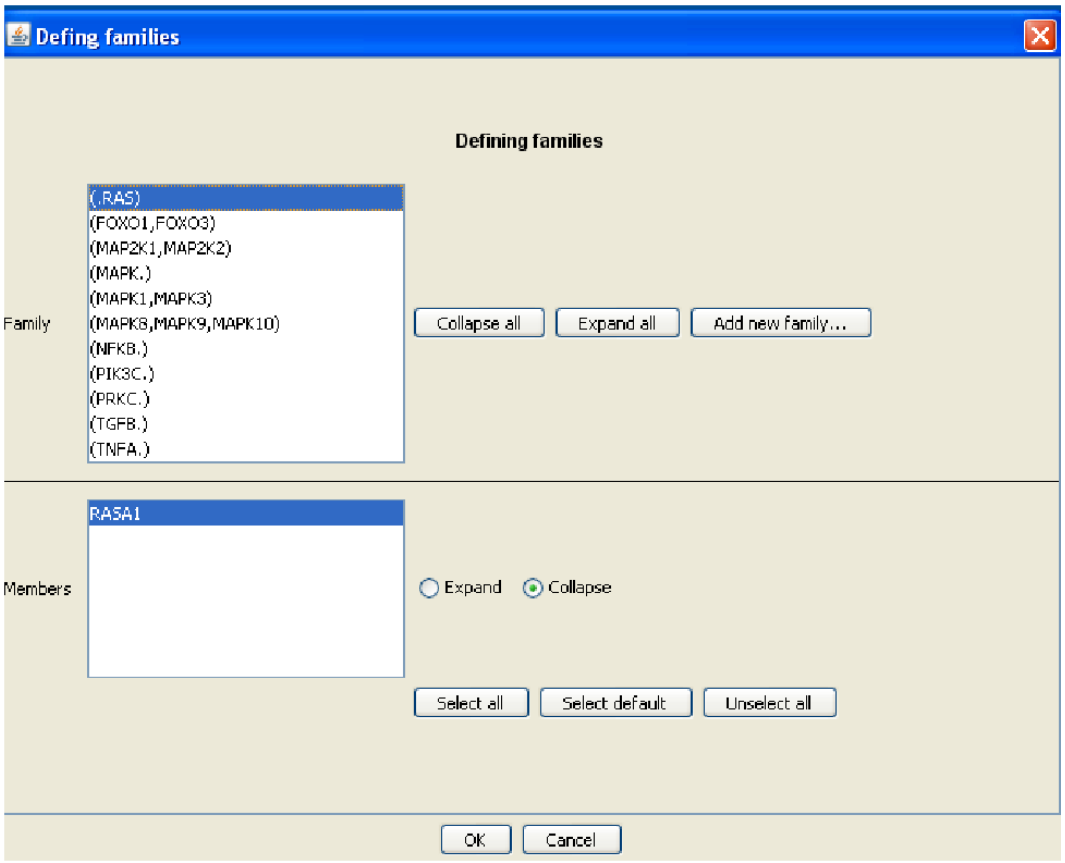


Figure 5: Dialog window for families management when importing apoptosis influence file. BiNoM recognizes AIN format for families and proposes to either collapse or expand them.

2.4 Import influence network from AIN file

The precise format of the influence file (AIN) is described in the appendices (section 8.5). It is an annotated list of links between genes, in a simple ASCII format.

Plugins⇒BiNoM I/O⇒Import AIN file
The AIN file of the apoptosis model ExamplApop.txt is imported. First, the user is asked to manage the families (groups of genes or proteins, see the appendices for a precise description): they can be expanded (replacing the family by all its members) or collapsed (replacing all family members by the name of the family). See figure 5.

Then a dialog window proposes to add constitutive reactions: influences that link proteins (or families) to their complexes and proteins (or families) to their phosphorylated state. See figure 6.

The imported network is synchronized with BioPAX format that includes the annotations of the AIN file. All this information can be accessed via BioPAX 3 property editor (see BioPAX Utils, section 5.1).

2.5 Export current network to BioPAX 3, Export current network to CellDesigner

The Cytoscape networks can be exported in BioPAX and CellDesigner by:

- **Plugins⇒BiNoM I/O⇒Export current network to BioPAX 3**
- **Plugins⇒BiNoM I/O⇒Export current network to CellDesigner**

⚠ provided that they are associated to existing CellDesigner or BioPAX by:

- Associate BioPAX Source, see section 2.7.
- Associate CellDesigner Source,see 2.9.

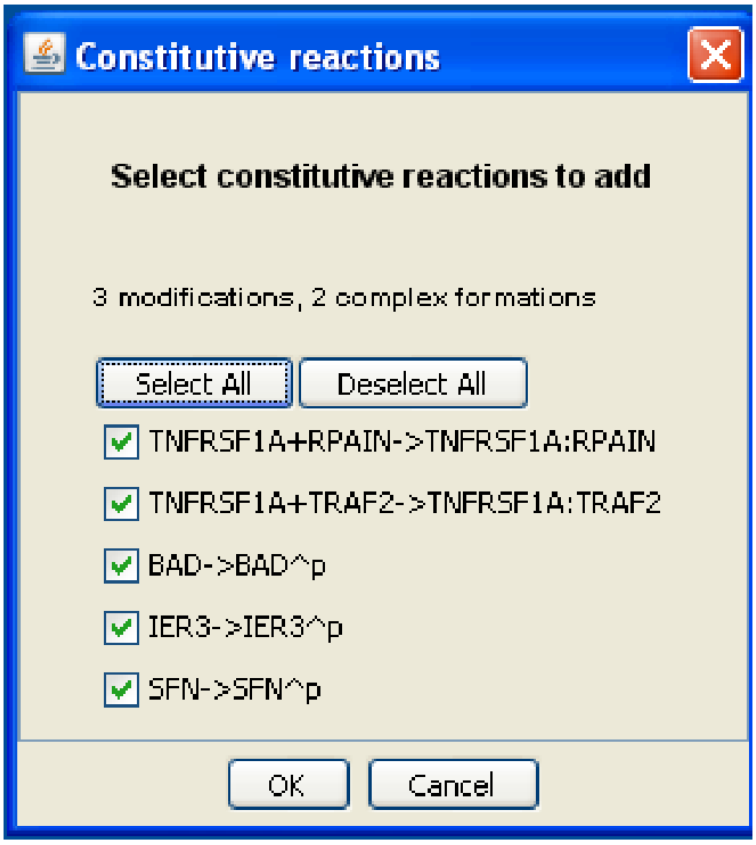


Figure 6: Dialog window for constitutive reactions when importing apoptosis influence file. BiNoM detects all possible constitutive reactions and proposes to add them.

More precisely, BiNoM is able to convert CellDesigner to BioPAX, and BioPAX reaction network interface to pure SBML. BiNoM is also able to export only a part of CellDesigner and BioPAX file, visible in the current Cytoscape network (interface). During the export operation, BiNoM is also able to merge a part of associated BioPAX file with already saved another part. BiNoM can modify the content of a BioPAX file.

⚠ BiNoM is NOT able to create CellDesigner file with all graphical notations from BioPAX or from scratch, and it is not able to modify the content of a CellDesigner file.

Here are the typical scenarios when BiNoM export operations can help.

1. User imports a big BioPAX file as reaction network and using Cytoscape creates a new subnetwork from the global reaction graph. After he can export this subnetwork into a separate self-containing BioPAX file.

2. User imports the pathway structure of a big BioPAX file and selects only a few pathway or pathwayStep nodes he is interested in. After he can export a part of the BioPAX file necessary to define these pathways.

3. User imports a BioPAX file as reaction network, selects a subnetwork and exports it as pure SBML to be used for creation of a computational model of this subpart later.

4. User imports CellDesigner file, selects a subnetwork and exports it as a CellDesigner file: it can be useful for creating a CellDesigner image of a network module of a big reaction network.

5. User imports CellDesigner file, selects a subnetwork and exports it as a BioPAX file (some SBML-specific information such as parameters values will be lost).

The networks created as a result of the import operation are already associated to the corresponding BioPAX or CellDesigner files. However, if the XGMML file is saved and used in another Cytoscape session, or if a new network is created from the initial network with Cytoscape New menu then this association is lost.

To perform export operation, the network should be Re-associated to the corresponding file (from which

it is originated) through Plugins⇒BiNoM I/O⇒Associate... operation. For huge BioPAX files the association might take some time for the first association, but once the file is loaded into memory cache, the following associations are almost instantaneous.

To understand better what BiNoM can do or can not, read the sections 8.1, 8.3 and 8.4 about the BiNoM data model.

2.6 Export current network to SBML

Plugins⇒BiNoM I/O⇒Export current network to SBML
Export the current network to pure SBML level 2.

2.7 Associate BioPAX 3 Source

Plugins⇒BiNoM I/O⇒Associate BioPAX 3 Source
Associate a BioPAX 3 Source to allow exportation in BioPAX 3 as explained in section 2.5

2.8 Save whole associate BioPAX 3 as

When the content of the BioPAX file is modified (through BioPAX property editor, see section 5.1), it can be to be saved as a whole (not only visible part) by
Plugins⇒BiNoM I/O⇒Save whole associated BioPAX 3 as
Otherwise, all modifications made in the different interfaces are lost. Changes are visible but only recorded permanently when the document is save.

2.9 Associate CellDesigner Source

Plugins⇒BiNoM I/O⇒Associate BioPAX 3 Source
Associate a CellDesigner Source to allow exportation in CellDesigner as explained in section 2.5

2.10 List all reactions

Plugins⇒BiNoM I/O⇒List all reactions
Display list of reactions, can be copied by control+A then control+C.

2.11 List all nodes

Plugins⇒BiNoM I/O⇒List all nodes
Display list of nodes, can be copied by control+A then control+C.

2.12 Color CellDesigner proteins

Plugins⇒BiNoM I/O⇒Color CellDesigner proteins
Cytoscape allows coloring nodes according to values of attributes (for example expression data) by the powerful possibilities of VizMapper. The export to CellDesigner keeps the colors. This process can be used to color species in CellDesigner. The function Color CellDesigner proteins allows to color proteins in CellDesigner which describe the components of complexes.

The gene expression file is based on Hugo names, data in columns (first line title and tabulation as column separator):
Hugo names<Tab>expression level 1<Tab>expression level 2...

Open dialog box Color CellDesigner proteins, input CellDesigner file name and gene expression file, click on ok. BiNoM generate a file *.conv where Hugo names are converted in protein names (links by annotation in CellDesigner, check if correct) and a CellDesigner file by column. When there are several Hugo name the highest is kept.

Figure 7 shows the aspect of colored proteins inside complexes.

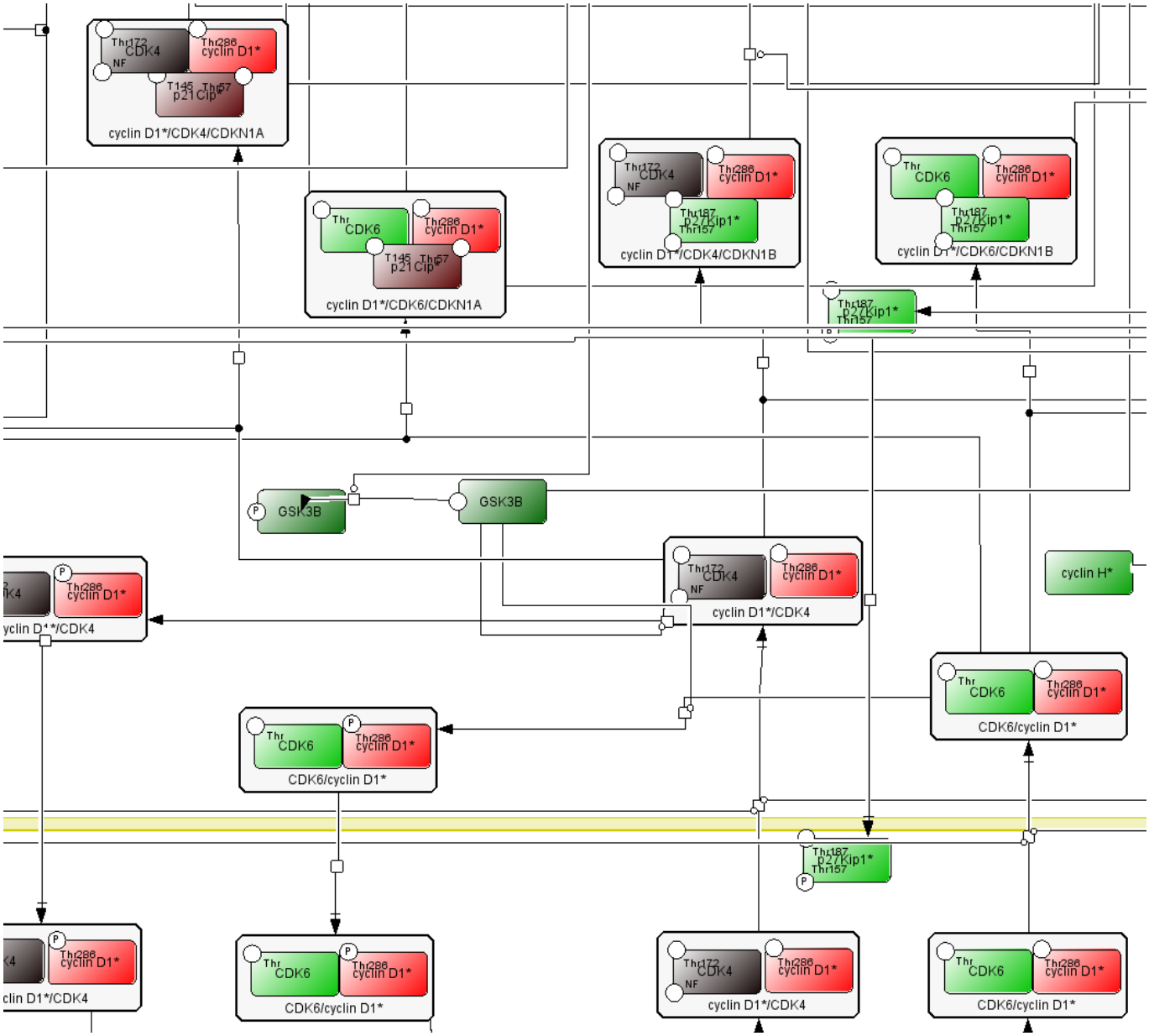


Figure 7: CellDesigner view of an extract from Rb-E2F[2] pathway colored by ficticious expression data

2.13 Modify CellDesigner notes

Plugins⇒BiNoM I/O⇒Color CellDesigner proteins
Modify in Cytoscape the notes of CellDesigner file when exporting.

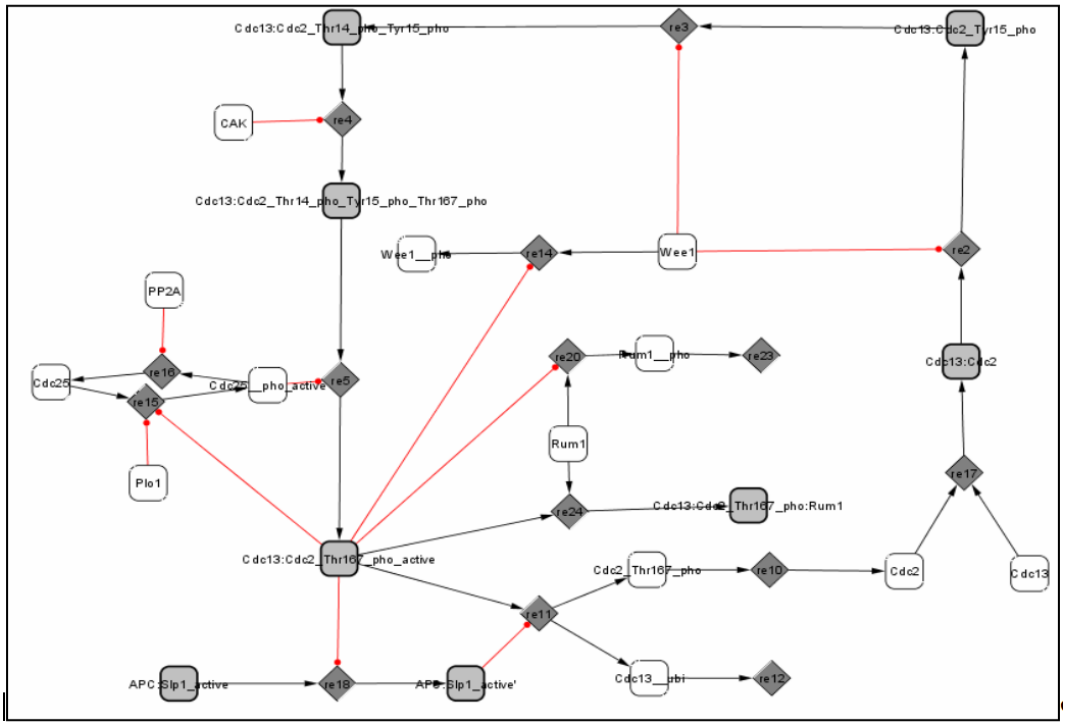


Figure 8: Cytoscape view of the M-Phase network

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 8).
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[8], the strongly connected components are isolated. In simple words, the obtained network, M-Phase.xml_scc1(figure 9), 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 10: 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 9, 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).

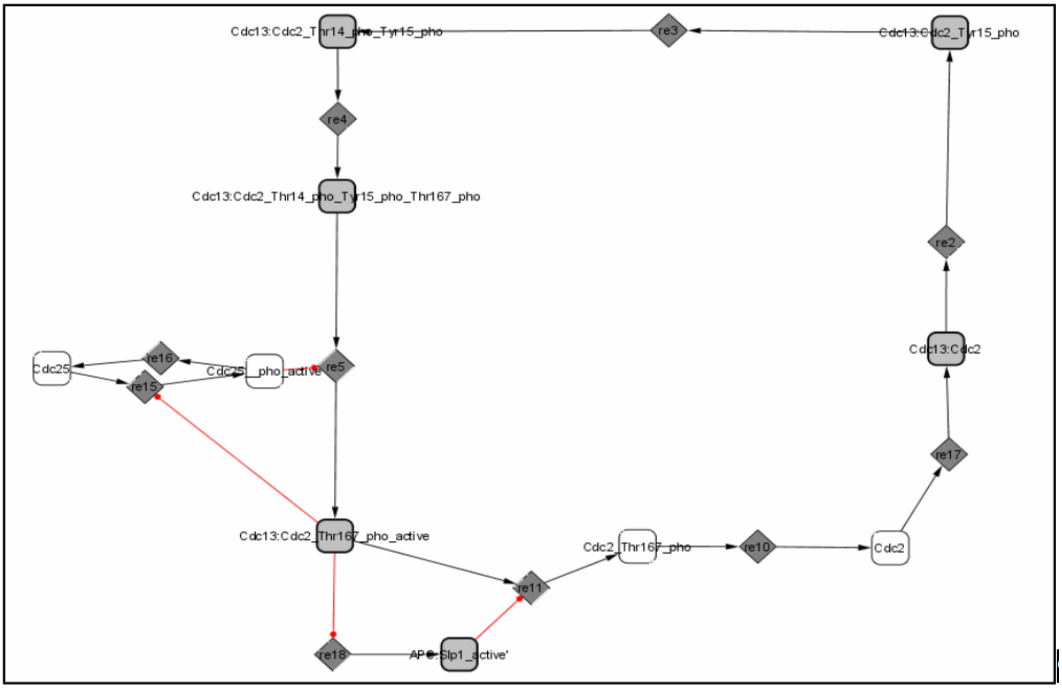


Figure 9: Strongly Connected Component of M-Phase network

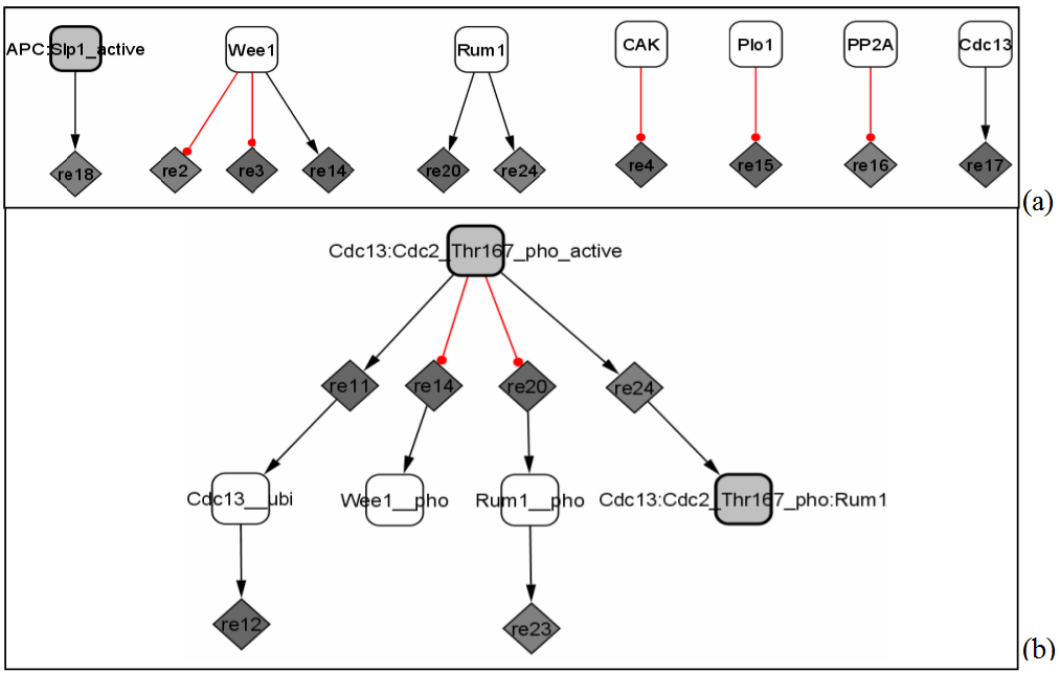


Figure 10: Prune the graph. (a) Incoming flux: molecules involved in the IN part of the network, and (b) Outgoing flux: molecules involved in the OUT part of the network.

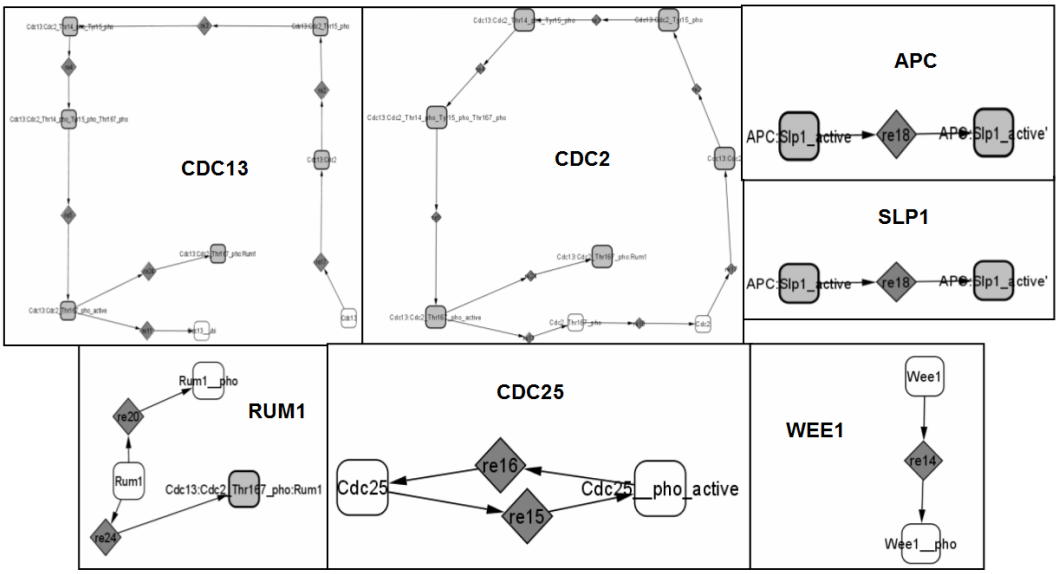


Figure 11: Material Components

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 11), 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[3], using a modification of the Vismaras algorithm[9]. Often, this feature gives information about the life cycle of a protein or a complex, about the feedbacks of the studied network, etc(figure 12). Note that the union of all the cycles corresponds to the strongly connected component figure 9.
⚠ 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 14). 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 13).
⚠ 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.

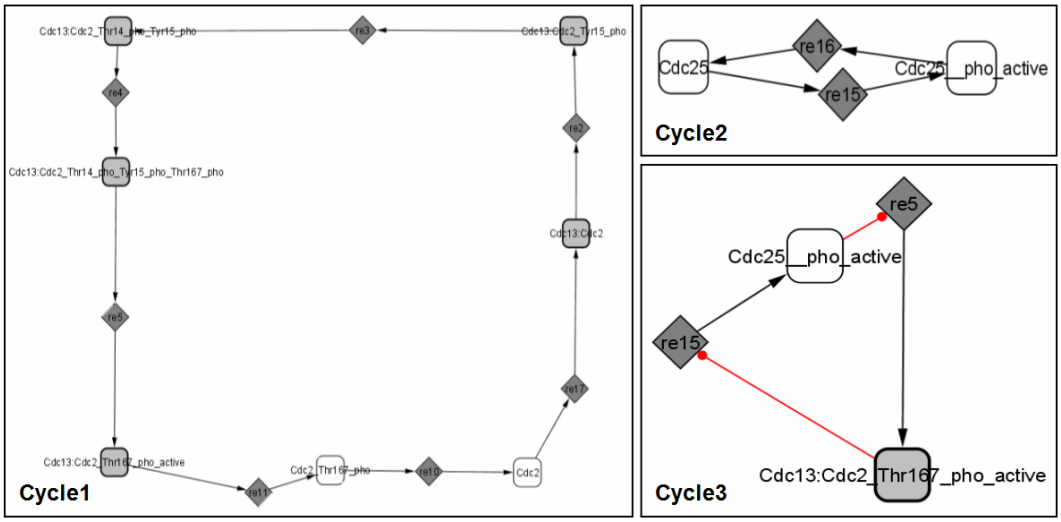


Figure 12: Minimal cycle decomposition of the M-Phase network. Cycle 1 includes CDC2 and CDC13 proteins, Cycle 2 CDC25 and Cycle 3 shows the feedback existing between CDC13/CDC2 and CDC25.

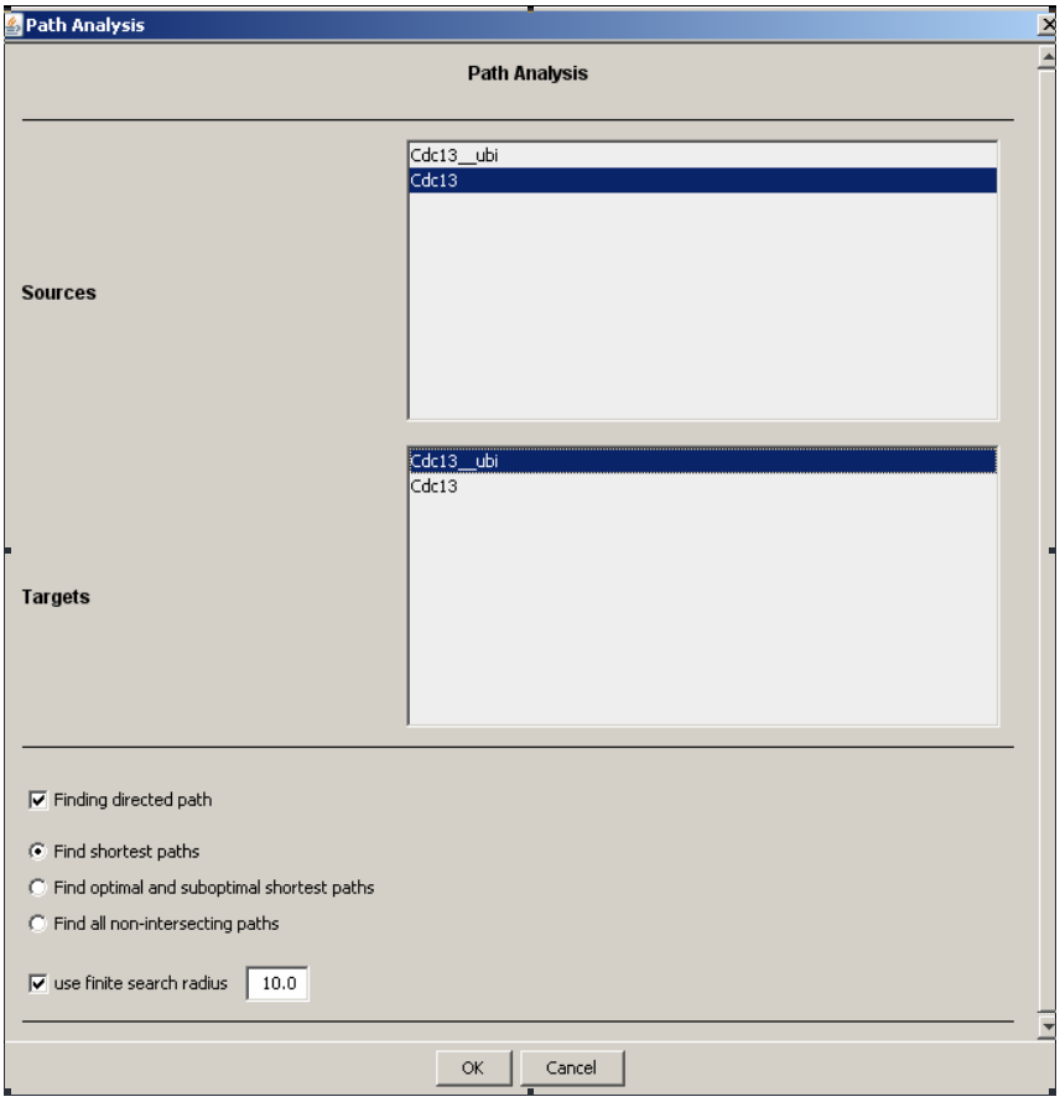


Figure 13: BiNoM Path Analysis: Pop-up window in which the source(s) and the target(s) need to be specified along with the type of paths (shortest, optimal shortest or all paths).

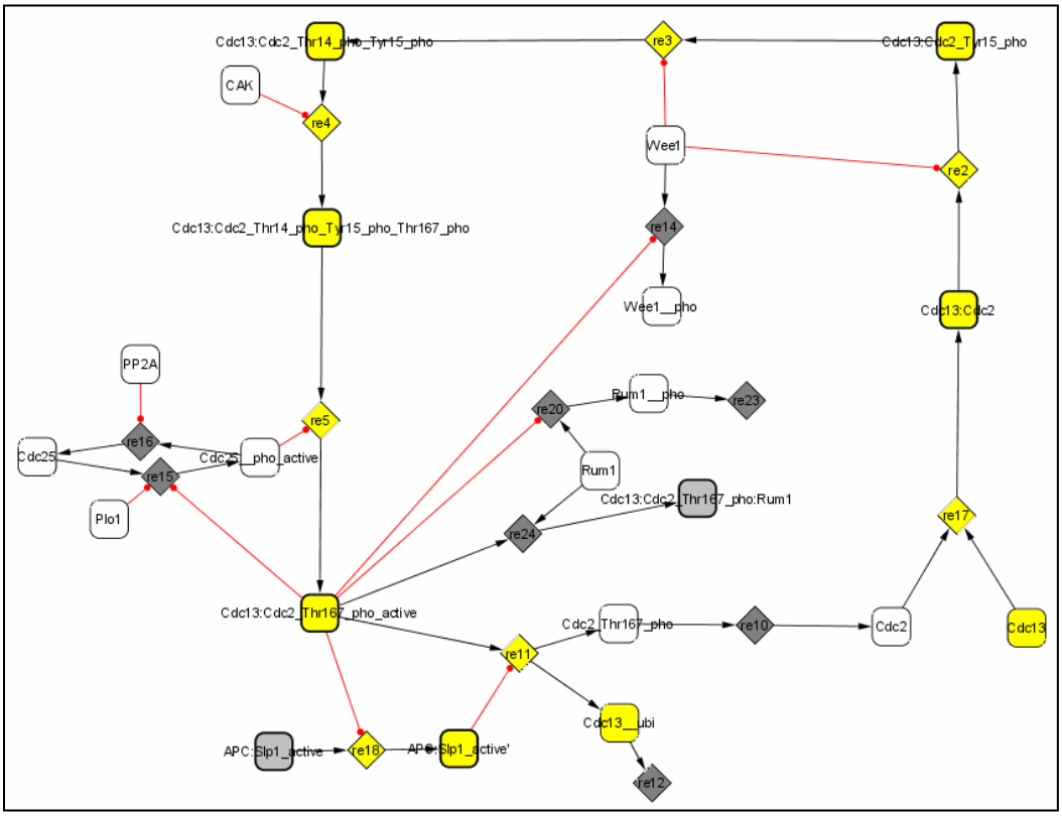


Figure 14: Path Analysis: All the paths leading from one molecular species (Cdc13) to another (Cdc13_ubi, ubiquitinated form of Cdc13) are highlighted in yellow.

3.7 Extract subnetwork

Plugins⇒BiNoM⇒analysis⇒Extract subnetwork
Extract a subnetwork from selected nodes of a network with various options (figure 15).

3.8 Calc centrality, Inbetweenness undirected, Inbetweenness directed

Plugins⇒BiNoM⇒analysis⇒Calc centrality⇒Inbetweenness undirected Plugins⇒BiNoM⇒analysis⇒Ca centrality⇒Inbetweenness directed Display centrality of nodes in cases undirected and 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 16).

There are different types of modular views. The modules are connected by: (1) the number of shared interactions (figure 17, upper panel); (2) the number of shared nodes (reactions + species) for which case the box Compact module intersection must be checked (figure 17, middle panel); and (3) the shared nodes and reactions showed explicitly (figure 17, 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

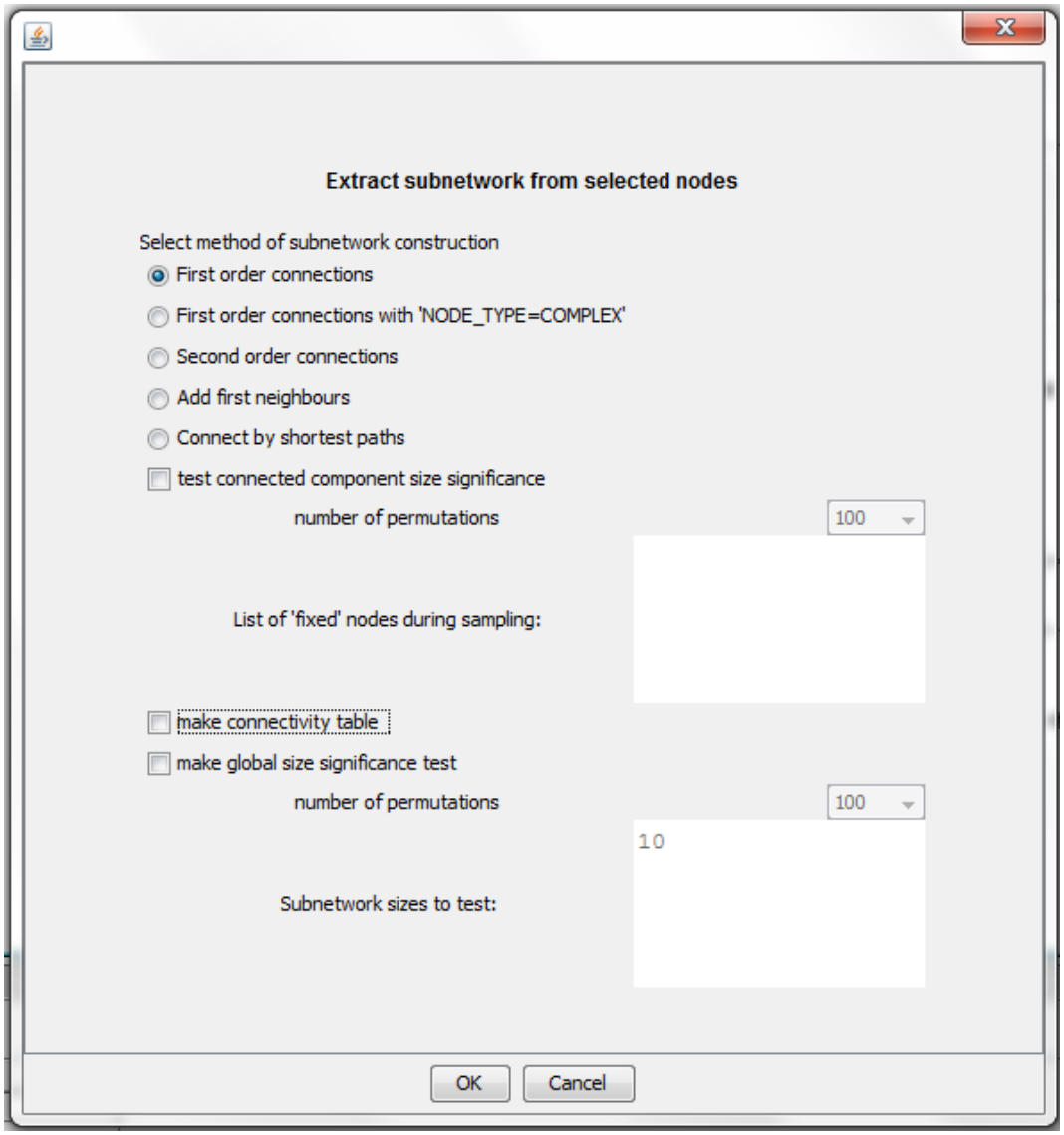


Figure 15: Dialog of extract subnetwork showing options.

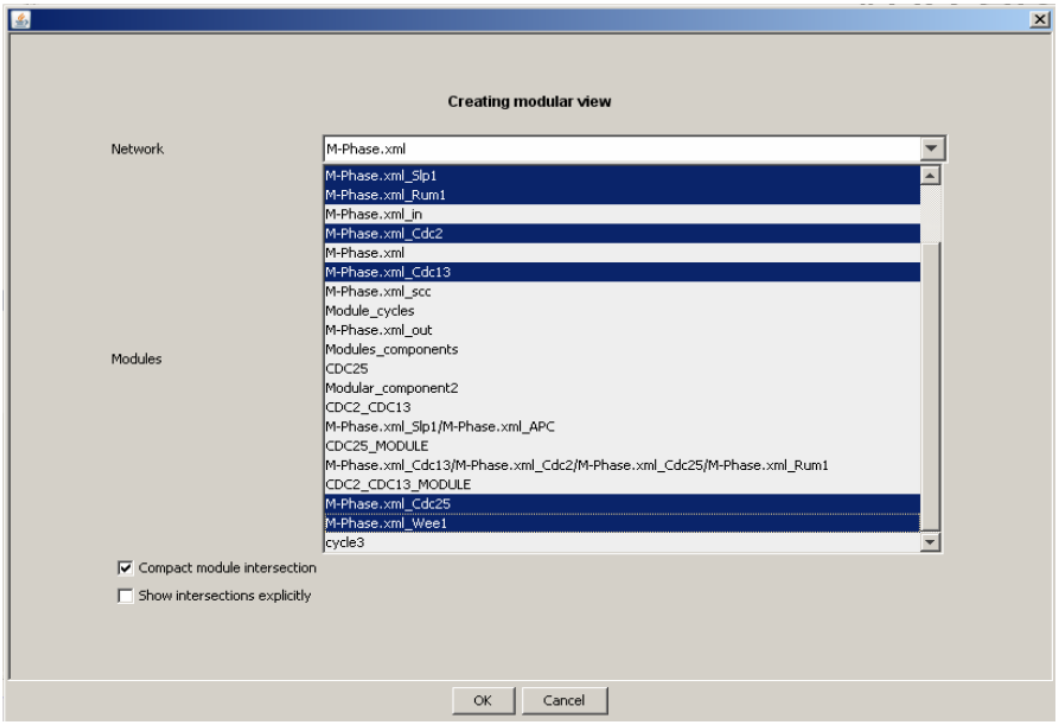


Figure 16: BiNoM modular view of the newtork: Pop-up window in which the initial graph and the modules are specified.

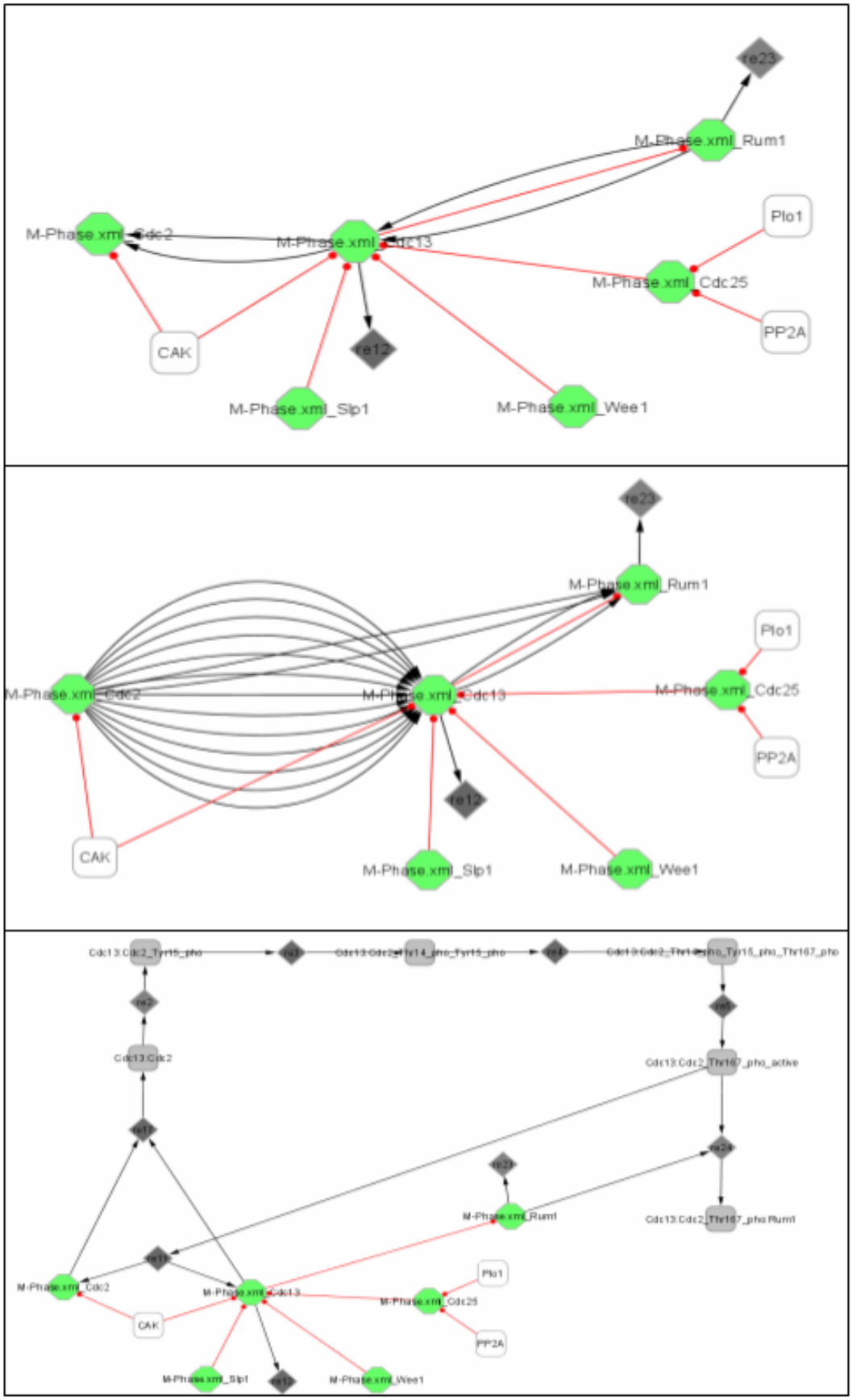


Figure 17: BiNoM modular view of the newtork: The resulting modular network (upper panel) with compact module intersections (middle panel) and with explicit intersections (lower panel).

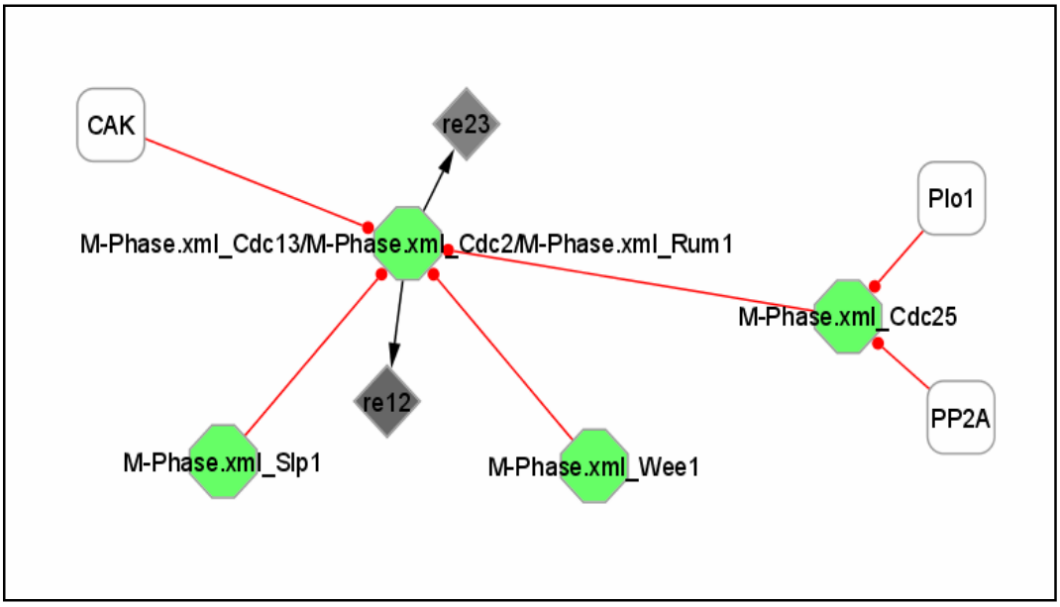


Figure 18: Clusters of modules. The obtained diagram is a compact modular view of the M-Phase network using the material decomposition and material components clustering

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 18 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 12. They are obtained by clustering the three cycles into two (cycle 1 + cycle2/cycle3) and organized into a modular view (Figure 19).

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 20: 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.

3.12 Linearize network

Plugins⇒BiNoM analysis⇒Linearize network
Remove reactions and reconnect edges according to a supposed influence (figure 21)
⚠ The got network is not an influence network in the biological sense. But, it can be used to build an influence network.

3.13 Exclude intermediate nodes

Plugins⇒BiNoM analysis⇒Exclude intermediate nodes
This function opens a dialog where nodes to be excuded can be selected (figure 22). It creates a network without the selected nodes and reconnects edges.

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.

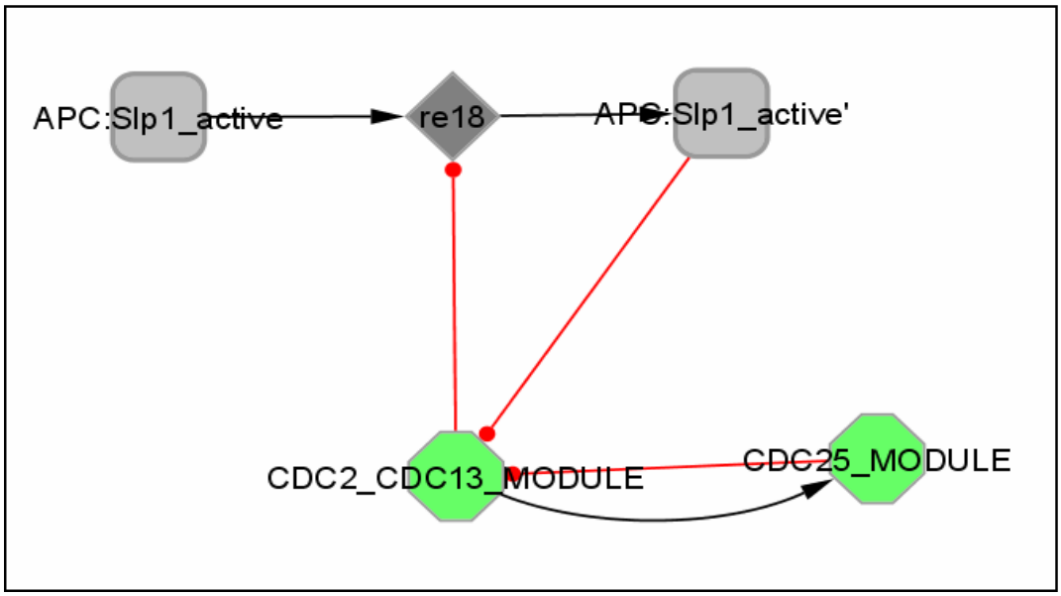


Figure 19: Clusters of modules. The obtained diagram is a compact modular view of the M-Phase network using the relevant cycle decomposition and cycle clustering

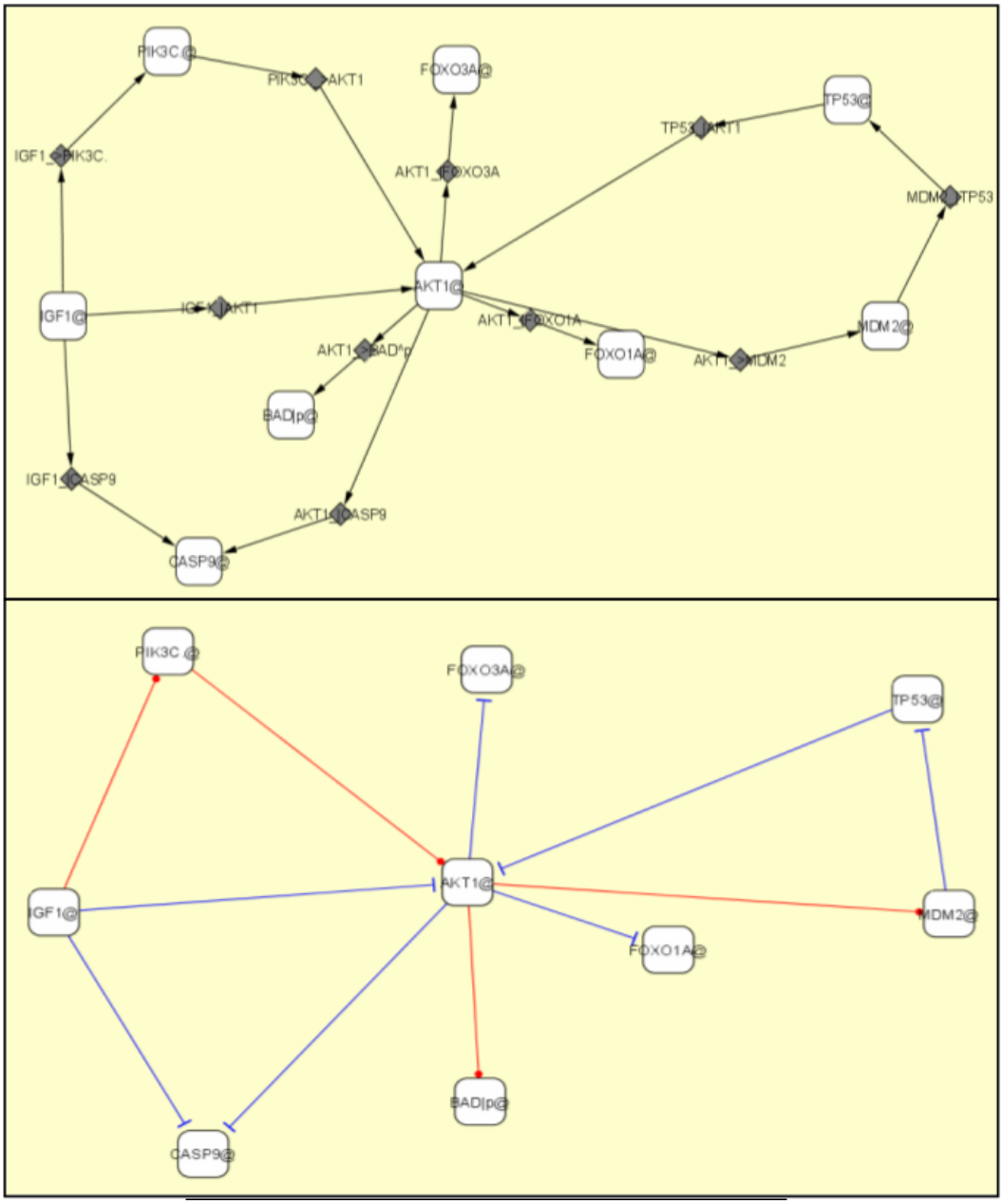


Figure 20: From a BioPAX network (upper panel) to an influence graph (lower panel).

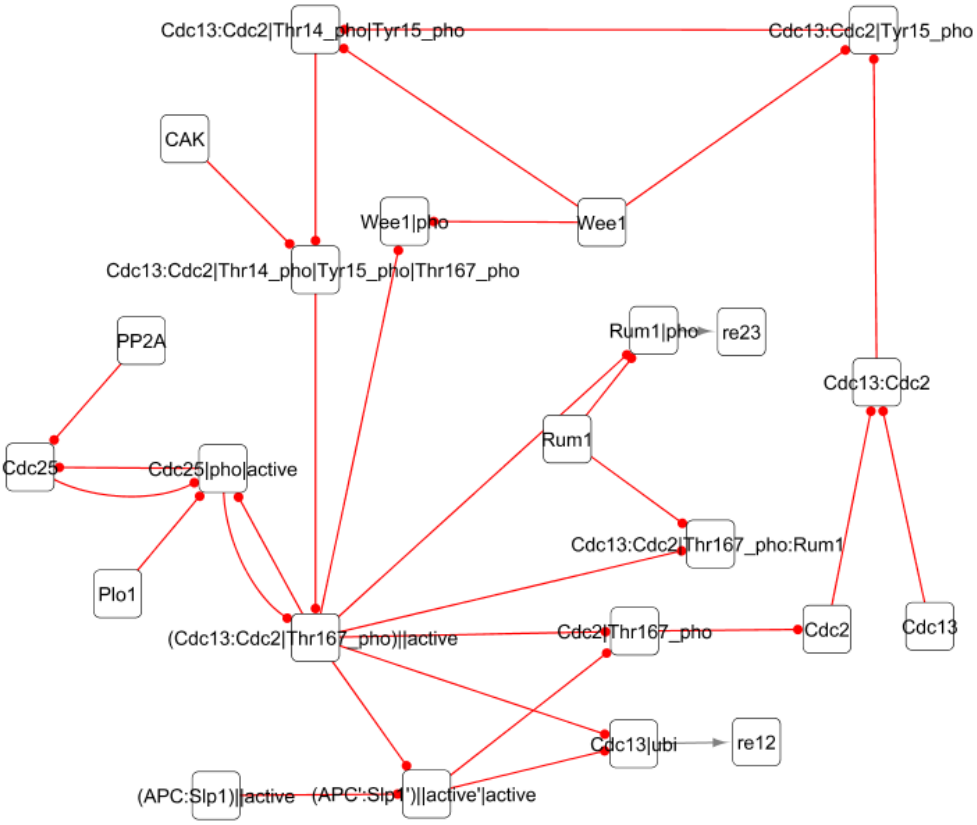


Figure 21: Result of applying "Linearize network" to M-Phase.

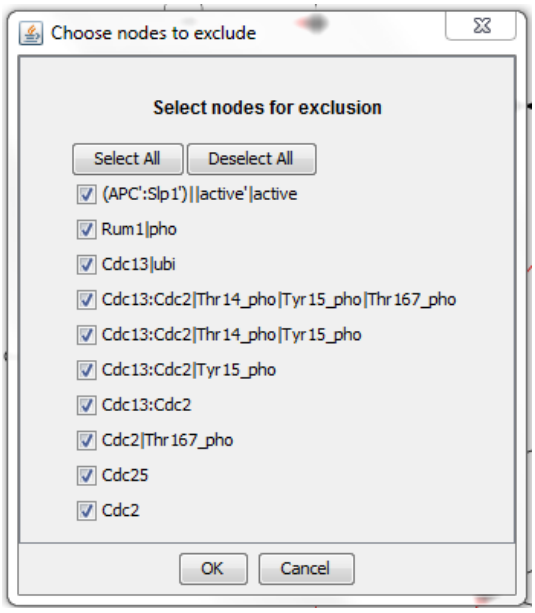


Figure 22: Dialog to select nodes to be excluded in the created network.

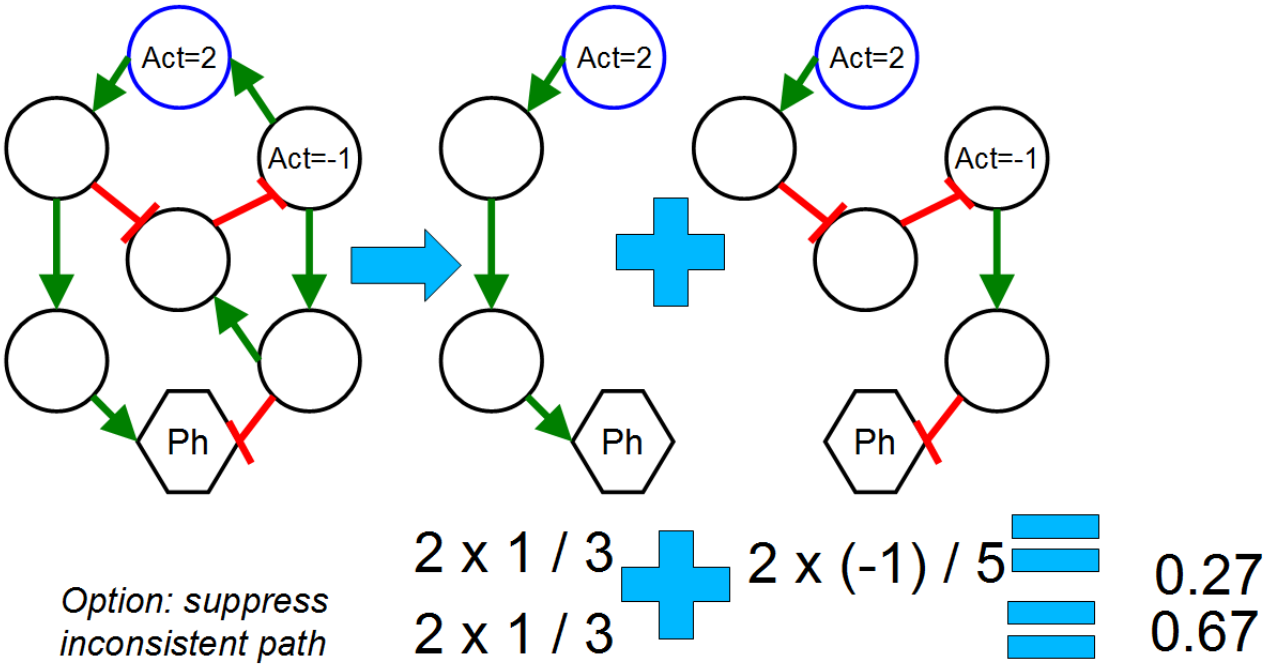


Figure 23: Computing influence in a simple example.

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

Plugins⇒BiNoM analysis⇒Path consistency analysis
This function is based on an algorithm of Path Influence Quantification (PIQuant). Shortly, for every annotated nodes, the influence is computed by summing (optimal and sub-optimal) paths contribution (1/(length of the path)), multiplied by the nodes annotation that represent biological data. For global influence computation, influences are summed on all over the annotated nodes. The figure 23 shows the computing in a simple example.

For global influences, a p-value can be computed, by shuffling the annotation. Among the possible options accessible in BiNoM, the possibility of removing inconsistent path (paths that have an intermediate node with inconsistent annotation) was sometimes used.

The dialog has 2 steps: input parameters and options, select path and display results.

Step 1 see figure 24:

- Select activity attribute name and update list, only active nodes are displayed in box.
- Targets are phenotypes, any node can be selected.
- Choose options of paths, see glossary in section 8.7.
- Click OK.

Step 2 see figure 25:

- Click on a target node, paths and features of path evaluated according to PIQuant are displayed.
- A text report about optimal cut set can be get by "path activities".
- Possibility to filter nodes and paths.
- A significance test by p-value allows to insure got activities are significant.

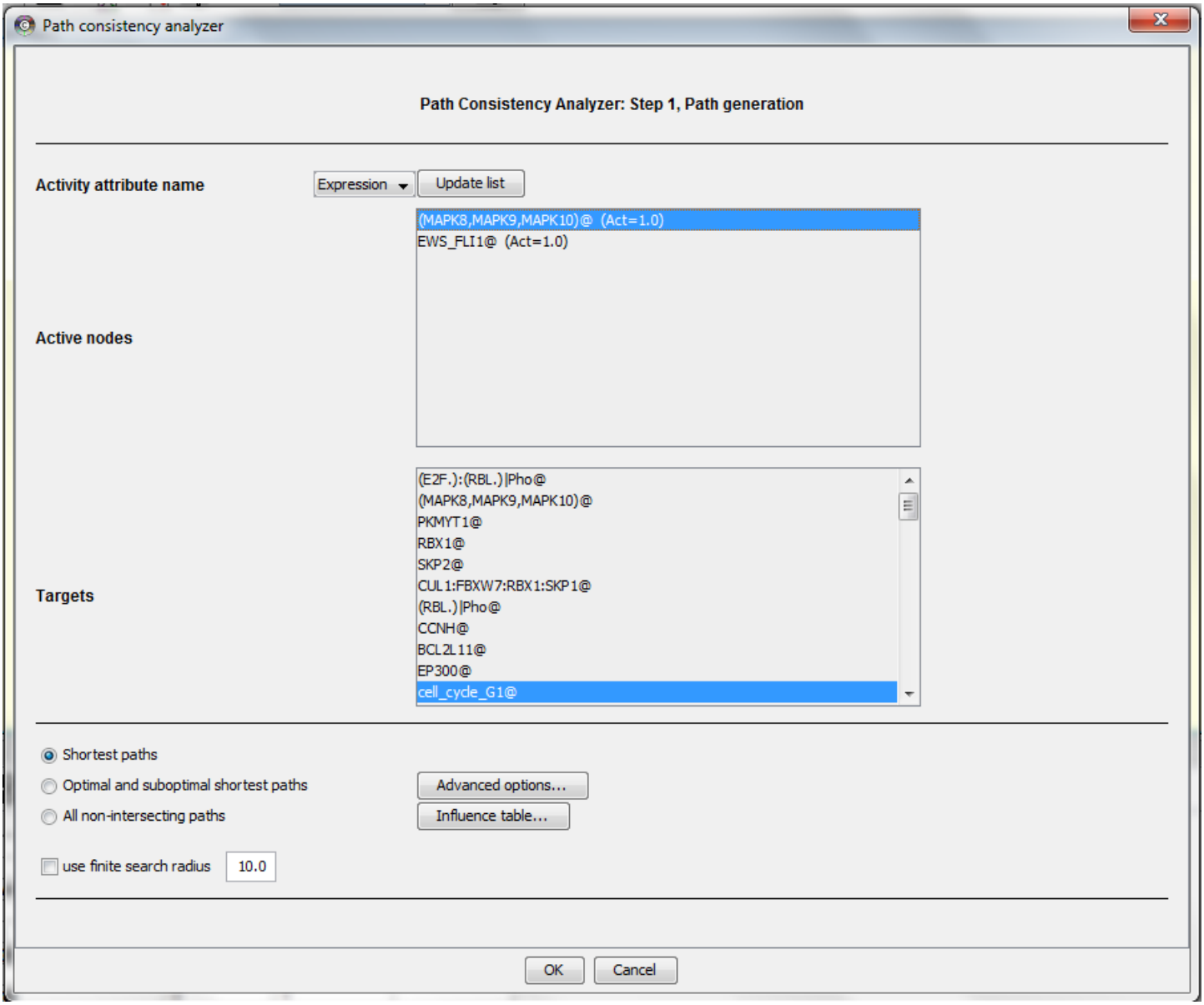


Figure 24: Path consistency analysis: dialog of step 1, select attribute, sources, targets and options.

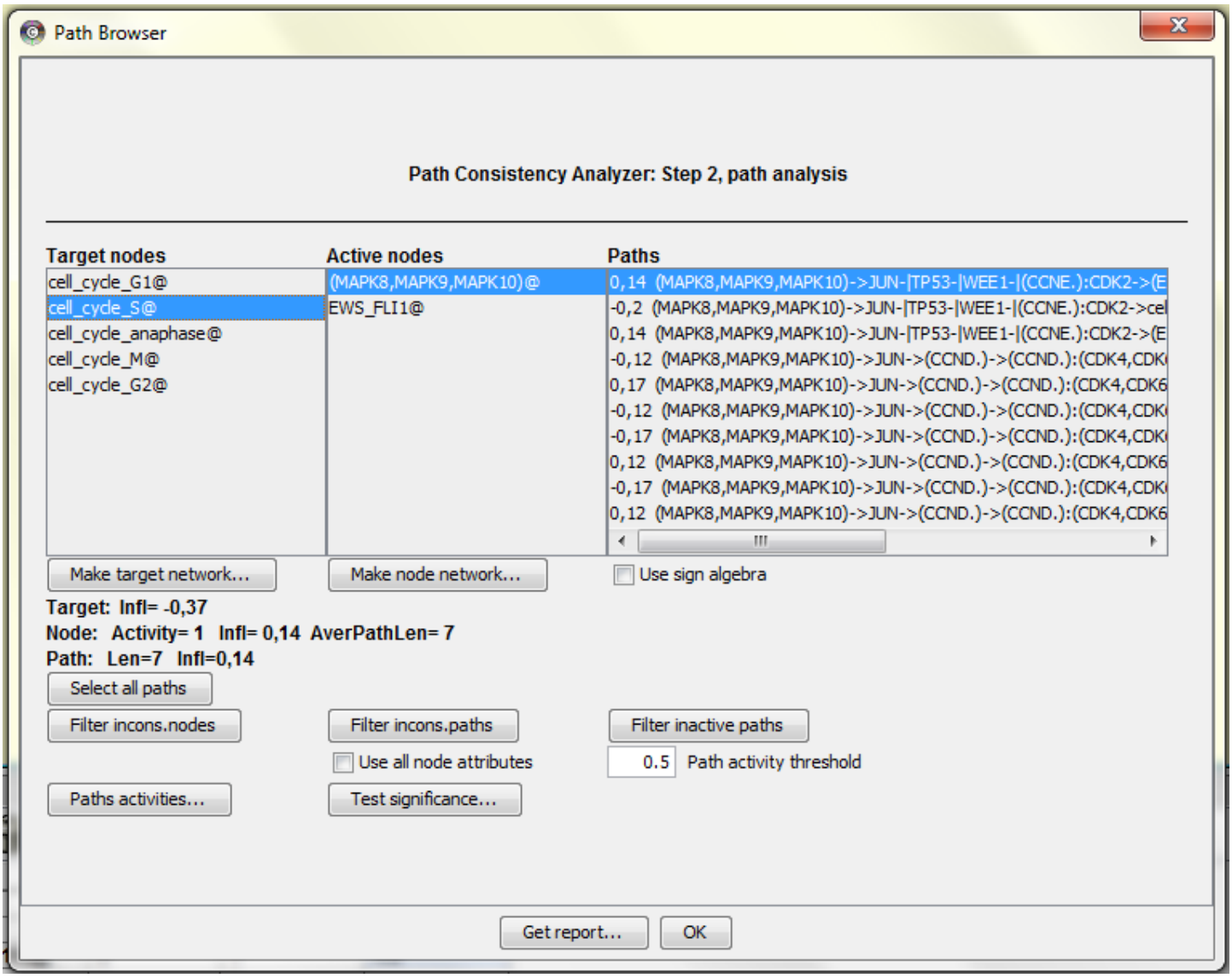


Figure 25: Path consistency analysis: dialog of step 2, display paths and activities, get all results

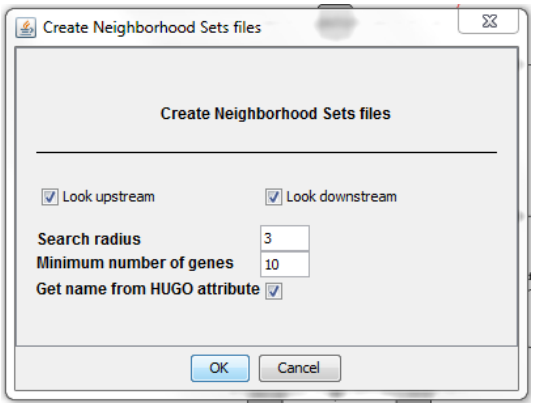


Figure 26: Diialog for options of creating a neighborhood sets file.

3.16 OCSANA analysis

Plugins⇒BiNoM analysis⇒OCSANA analysis

3.17 Create neighborhood sets file

Plugins⇒BiNoM analysis⇒Create neighborhood sets file
This function creates a file *.gmt (a text file where nodes are separated by <Tab>) containing the neighbors of selected nodes according to option of the dialog(figure 26)

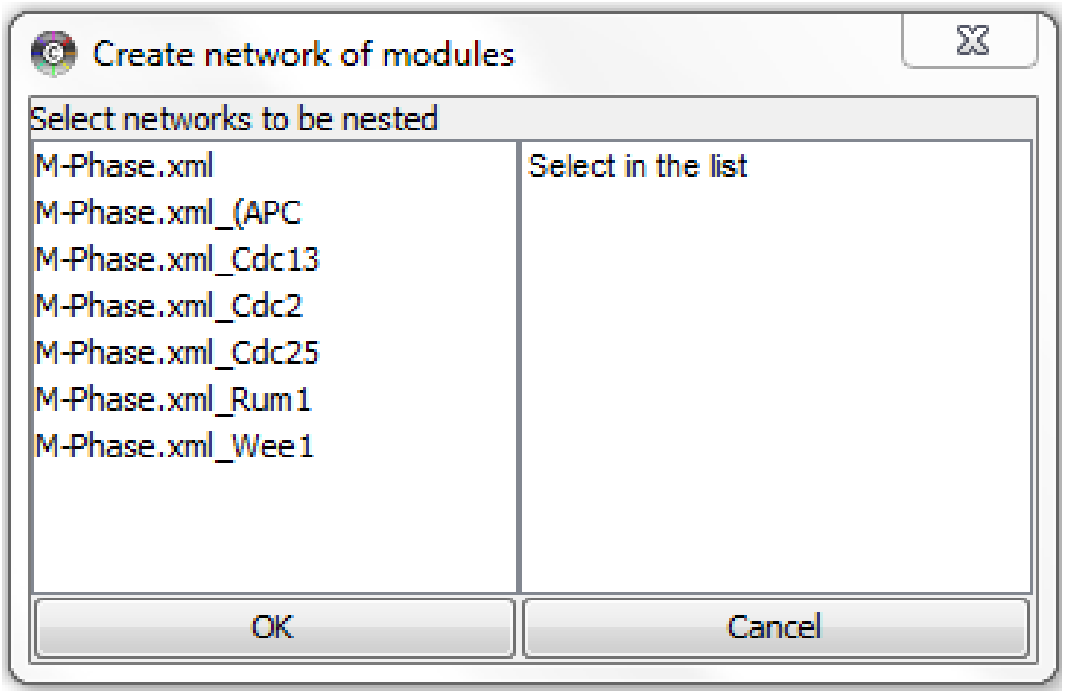


Figure 27: Dialog for select networks to become modules in modular0

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

Plugins⇒BiNoM module manager⇒Create network of modules
Create a new network from a list of sub-networks (sub-networks are selected in the network list see figure 27). Nodes=modules, no edge. Visual style created in VizMapper for module network . The got network is as 28 without edge and with nodes on grid.
⚠ Module names and node names must be different, all network names too.

To go from module to sub-network:
select node⇒CRight click⇒CNested Network⇒Go to Nested Network.

4.2 Create connections between modules

Plugins⇒BiNoM module manager⇒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

Plugins⇒BiNoM module manager⇒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

Plugins⇒BiNoM module manager⇒Agglomerate the nearest nodes in modules
Create modules and a modular view by agglomerating the nearest nodes in the active network (see algorithm in

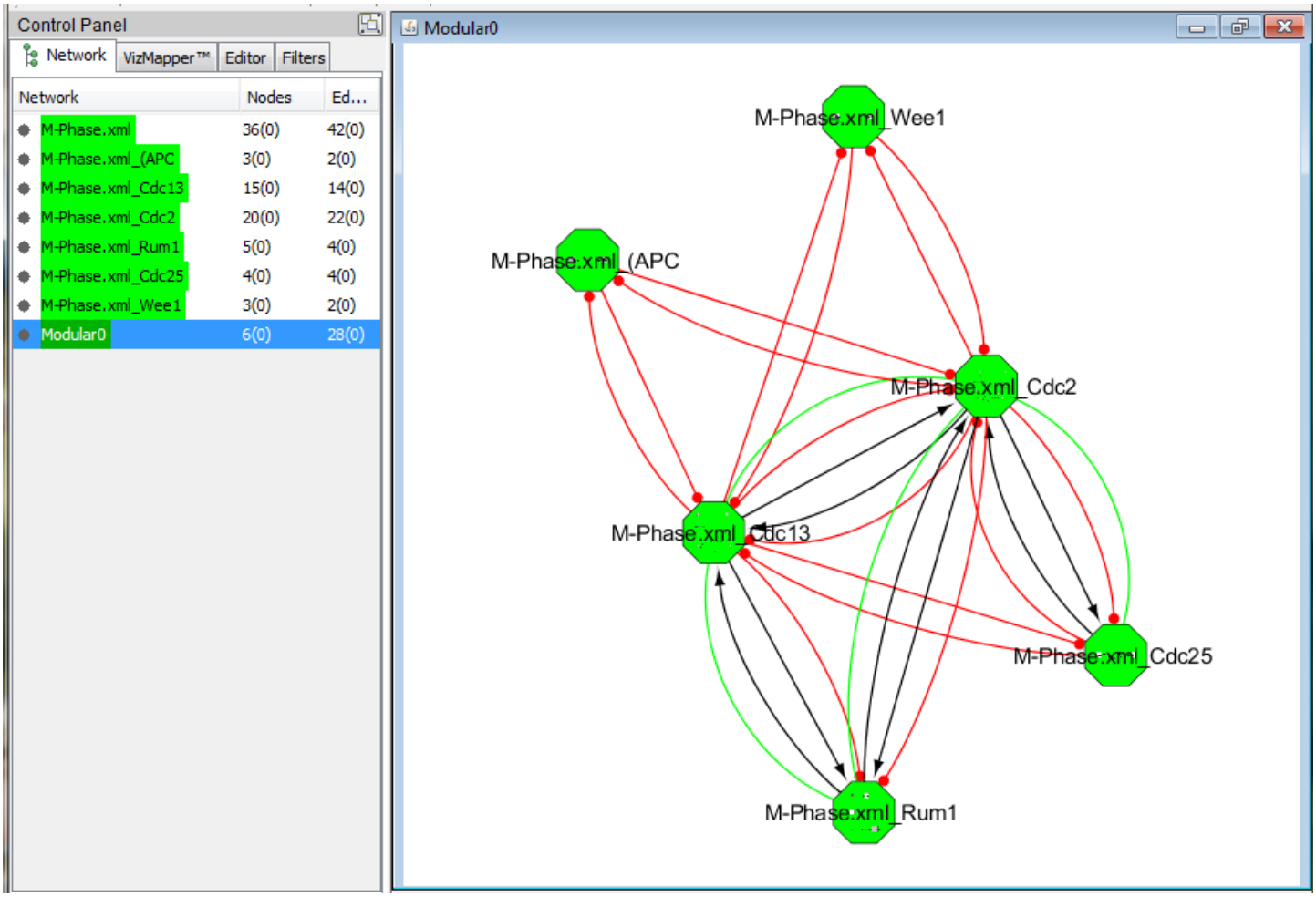


Figure 28: M-Phase is divided into modules by get material component. The modular view is got by creating network of modules with organic layout. The function "Create connections between modules" links modules according to the reference network. The function "Find common nodes in modules" creates intersection edges.

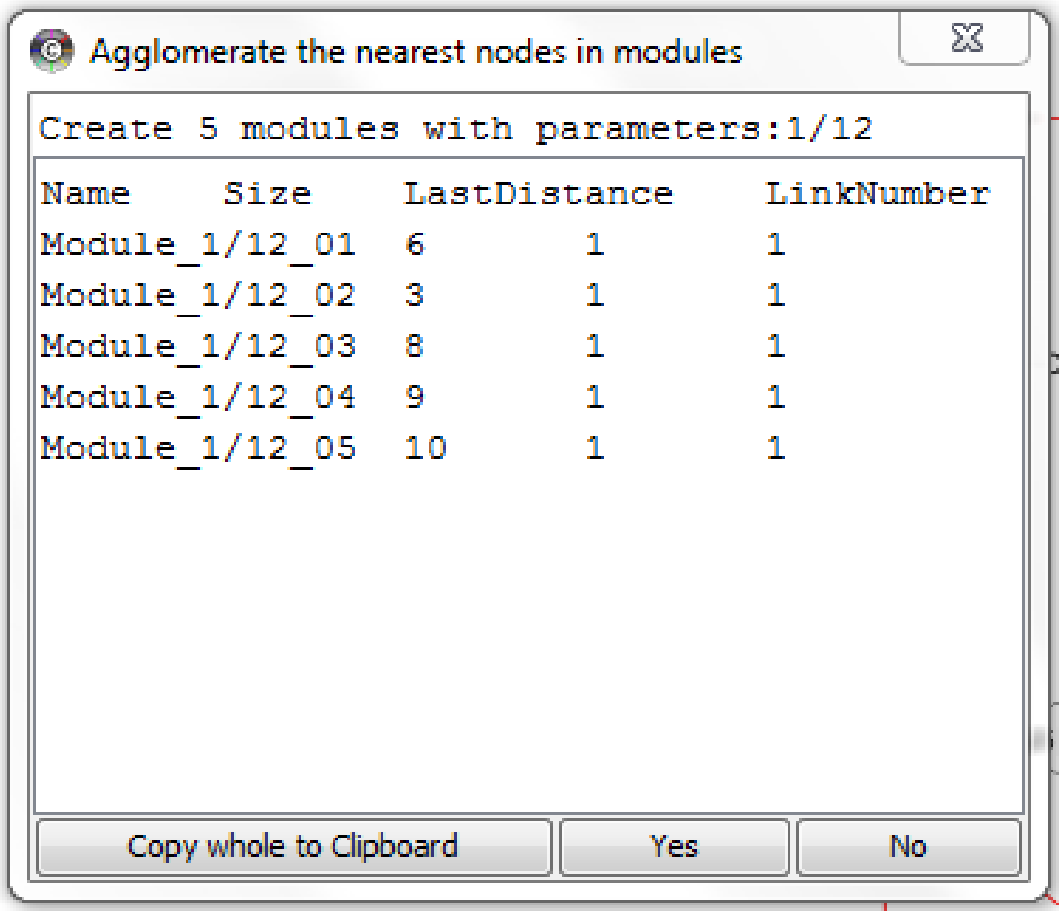


Figure 29: This window displays modules, number of node, last distance and number of links of agglomerating process. Yes lauches the process of agglomerating.

section 8.6.

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 (see dialog 29).

Sub-networks are created and gathered in a packed network as the function "Create modules from networks" (see figure 30).

4.5 List nodes of modules and network

Plugins⇒BiNoM module manager⇒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

Plugins⇒BiNoM module manager⇒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 (see 31.

Create intersection edges with number of common nodes as attribute (COMMON_NODES), green edges in figure 28.

Create node attribute containing the node numbers of modules (NODE_NUMBER).

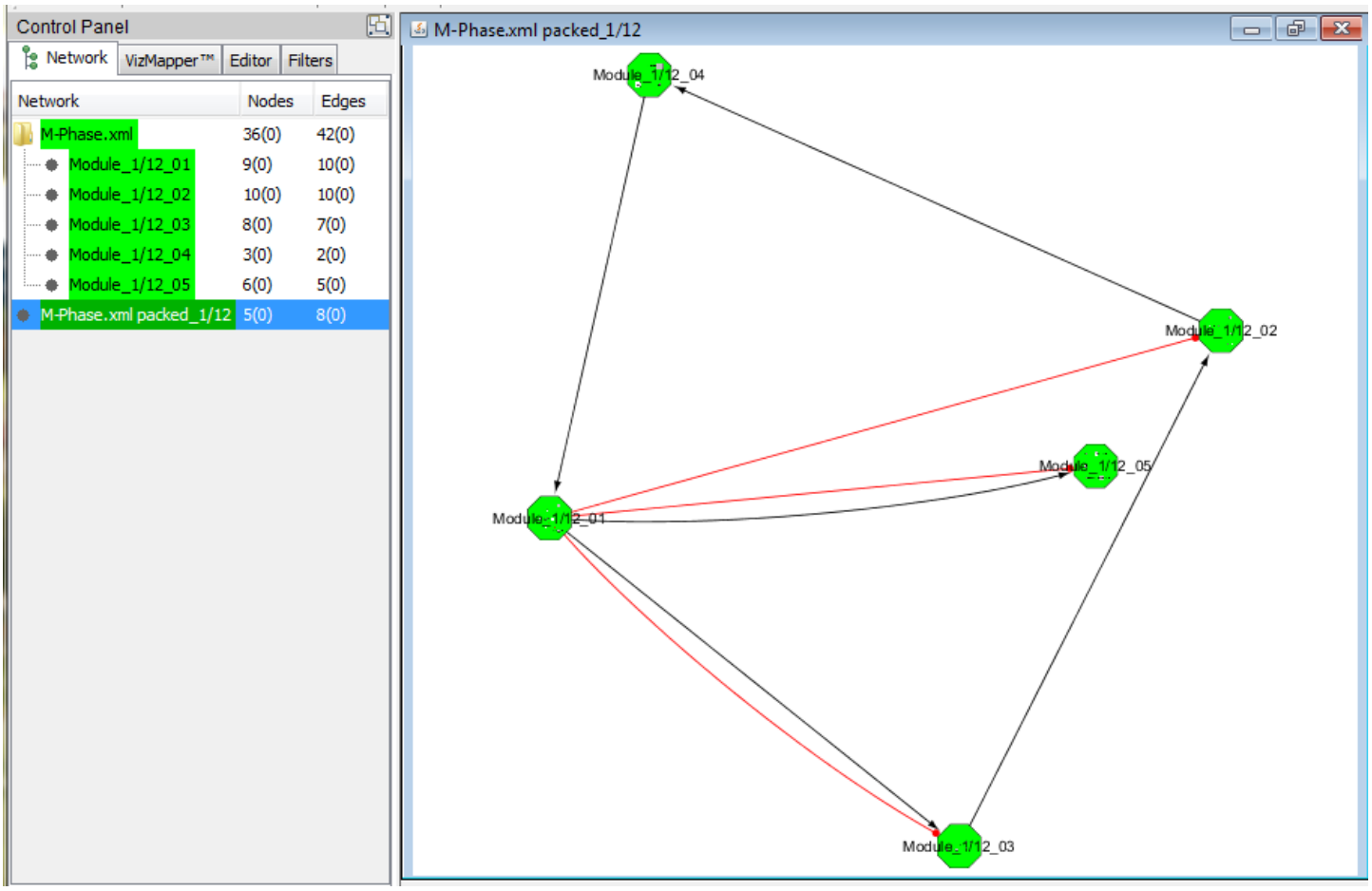


Figure 30: M-Phase is got by creating network from modules, modules created by agglomerating the nearest nodes (maximal distance=1, maximal size=12 nodes).

	M-Phase.xml_Wee1	M-Phase.xml_Rum1	...	M-Phase.xml_Cdc25	M-Phase.xml_Cdc2	M-Phase.xml_Cdc13	M-Phase.xml_(APC	**Frequency**
(APC:Slp1)||active	0	0	...	0	0	0	1	1
Cdc13:Cdc2|Thr14_pho|Tyr15_pho|Thr167_phc	0	0	...	0	1	1	0	2
Cdc13:Cdc2|Tyr15_pho	0	0	...	0	1	1	0	2
Wee1	1	0	...	0	0	0	0	1
Cdc13:Cdc2	0	0	...	0	1	1	0	2
Wee1|pho	1	0	...	0	0	0	0	1
Cdc13:Cdc2|Thr167_pho:Rum1	0	1	...	0	1	1	0	3
...	...	...	...	...	...	...	...	...
Cdc25|pho|active	0	0	...	1	1	0	0	2
re3	0	0	...	0	1	1	0	2
(APC':Slp1')||active'|active	0	0	...	0	0	0	1	1
Cdc13	0	0	...	0	0	1	0	1
re15	0	0	...	1	1	0	0	2
re2	0	0	...	0	1	1	0	2
re20	0	1	...	0	0	0	0	1
Size	3	5	...	4	20	15	3	

Figure 31: Matrix of nodes: modules in columns, nodes in rows, size of modules in last row, frequency in modules in last column.

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

Plugins⇒BiNoM module manager⇒Assign module names to node attribute
Create a node attribute (named as the modular network), containing module names. This attribute may be used to visualize modules in the reference network.

4.8 List components of species in network and modules

Plugins⇒BiNoM module manager⇒List components of species in network and modules
List components of species (their names must respect BiNoM syntax). Useful to name modules.

4.9 Create network from union of selected modules

Plugins⇒BiNoM module manager⇒Create network from union of selected modules
Create a network from union of selected modules and its corresponding module in the current network (named by module names separated by &).

4.10 Create network from intersection of 2 selected modules

Plugins⇒BiNoM module manager⇒Create network from intersection of 2 selected modules
Create a network from intersection of 2 selected modules and its corresponding module (named by module names separated by |.

Confirm for deleting the common nodes in the selected modules.

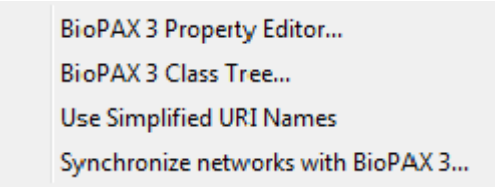
4.11 Recreate lost connections inside modules

Plugins⇒BiNoM module manager⇒Recreate lost connections inside modules
Recreate connections inside modules which may have been lost by modularizing operations.

4.12 Destroy networks unused as module

Plugins⇒BiNoM module manager⇒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



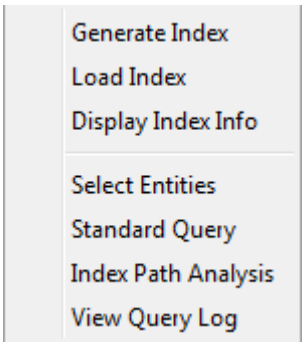
5.1 BioPAX 3 Property Editor

5.2 BioPAX 3 Class Tree

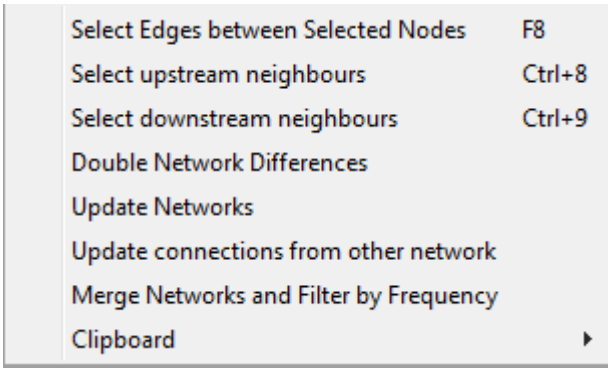
5.3 Using Naming Service Names / URI Names

5.4 Synchronize networks with BioPAX 3

6 BiNoM BioPAX3 Query



7 BiNoM Utilities



8 Appendices

8.1 Attributed graph model

BiNoM manipulates the information contained in the standard systems biology files by mapping it onto a labeled graph, called index. The index does not try to map the totality of all details; it rather serves as a connection map for the objects contained in other ontologies such as BioPAX. In other words, the index contains the minimum information needed to graphically represent objects and connections between them. Index elements (nodes and edges) are annotated by identifiers sufficient to find these objects in the original files and extract and edit the information related to them.

This approach has several advantages, in particular, with respect to synchronization issues. BiNoM index is a light-weight construction which can be easily regenerated, does not duplicate the information in existing files and serves only to facilitate the visualization and to access existing systems biology files.

Currently, BiNoM index is mostly developed to map BioPAX ontology files and CellDesigner object schema. In future versions, other mappings will be available, for instance, a mapping to SBML files annotated with Systems Biology Ontology (http://www.ebi.ac.uk/sbo/).

The table 32 lists all attributes used by the index.

8.2 BiNoM CellDesigner and BiNoM BioPAX visual mappers

BiNoM has two built-in visual mappers supporting the visualization of the whole index or of its parts. The legend for deciphering the different types of visualization is provided in figure 33.

8.3 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.

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.

Several examples of naming chemical species are presented:

- Naming chemical species shown in Systems Biology Graphical Notation standard figure 34
- A conversion from CellDesigner figure 35.
- A conversion from BioPAX figure 36.

8.4 Standard BioPAX interfaces

BiNoM index serves as a visual connector to the content of a network file. However, with all types of relations, the index is a highly connected graph and not very insightful when represented entirely. A subgraph of the index can be extracted according to a specific purpose and used to understand a specific aspect of the pathway information. We will call interface such a subgraph of the entire index.

When importing a BioPAX file, BiNoM proposes to generate three standard BioPAX interfaces referred to as

Node attributes		Edge attributes	
Attribute name	Meaning and possible values	Attribute name	Meaning and possible values
BioPAX attributes			
BIOPAX_NODE_TYPE	Type of entity represented by the node: protein, dna, rna, smallMolecule, transport, biochemicalReaction	BIOPAX_EDGE_TYPE	Type of connection between nodes: LEFT, RIGHT, CATALYSIS, NEXT, REFERENCE
BIOPAX_SPECIES	Species unique name	BIOPAX_EDGE_ID	Unique edge id
BIOPAX_REACTION	Reaction id	BIOPAX_URI	Full URIs of the objects associated with the edge
BIOPAX_NODE_SYNONYM	All entity synonyms		
BIOPAX_NODE_XREF	All entity accession numbers		
BIOPAX_URI	Full URIs of the objects associated with the node		
CellDesigner attributes			
CELLDESIGNER_NODE_TYPE	Type of entity represented by the node: PROTEIN, GENE, STATE_TRANSITION…	CELL_DESIGNER_EDGE_TYPE	Type of connection between nodes: such as LEFT, RIGHT, CATALYSIS
CELLDESIGNER_SPECIES	For species nodes – SBML species id		
CELLDESIGNER_REACTION	For reaction nodes – SBML reaction id		
CELLDESIGNER_ALIAS	CellDesigner node alias		
Common attributes			
EFFECT	For reactions and influence edges – effect of the influence. If something is known, such terms as 'activation', 'inhibition', 'catalysis' should be used		

Figure 32: All attributes of graph model used by the index

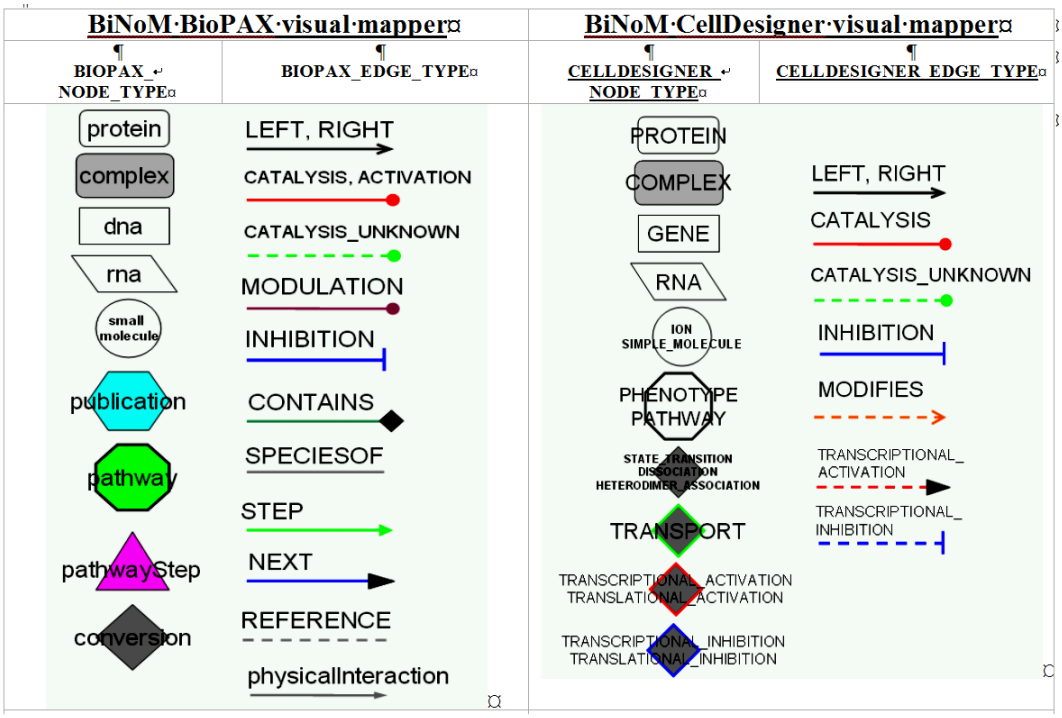
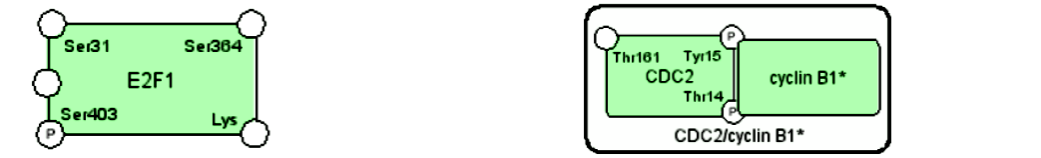


Figure 33: Types of visualization in BioPAX and CellDesigner



E2F1|Ser403_pho@nucleus

CDC2|Tyr15_pho|Thr14_pho:cyclinB1*@cytoplasm

Figure 34: 2 examples of naming chemical species shown in Systems Biology Graphical Notation standard.

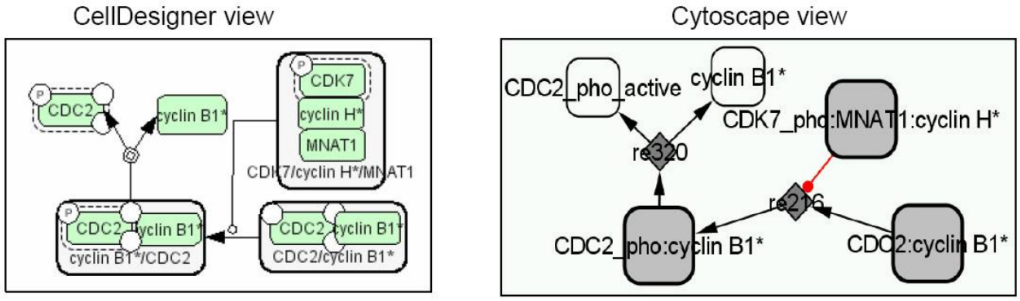


Figure 35: Conversion of a little network from CellDesigner Graphical Notation to BiNoM index representation

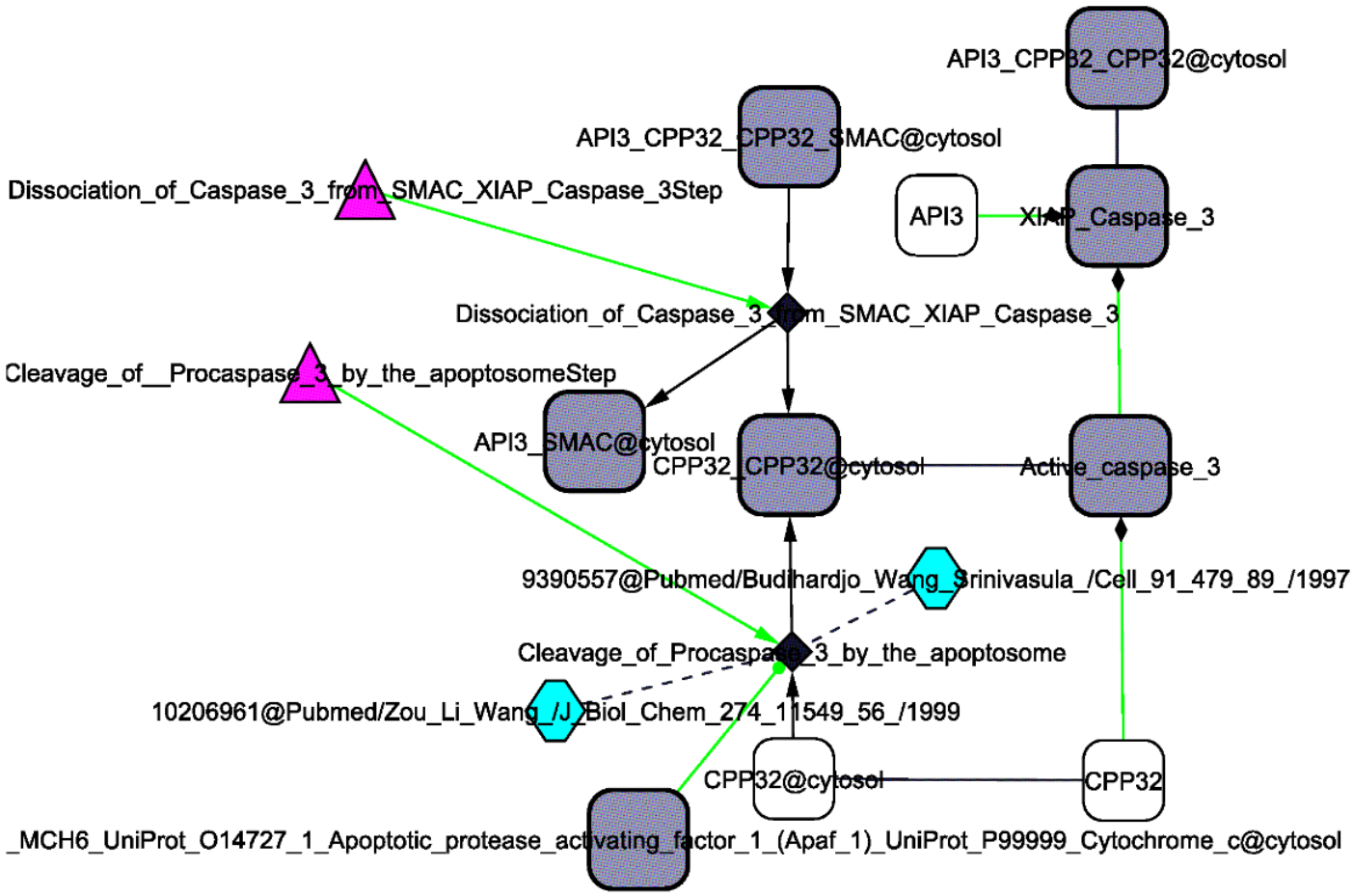


Figure 36: Small fragment of BioPAX index generated for Apoptosis pathway and extracted from Reactome database

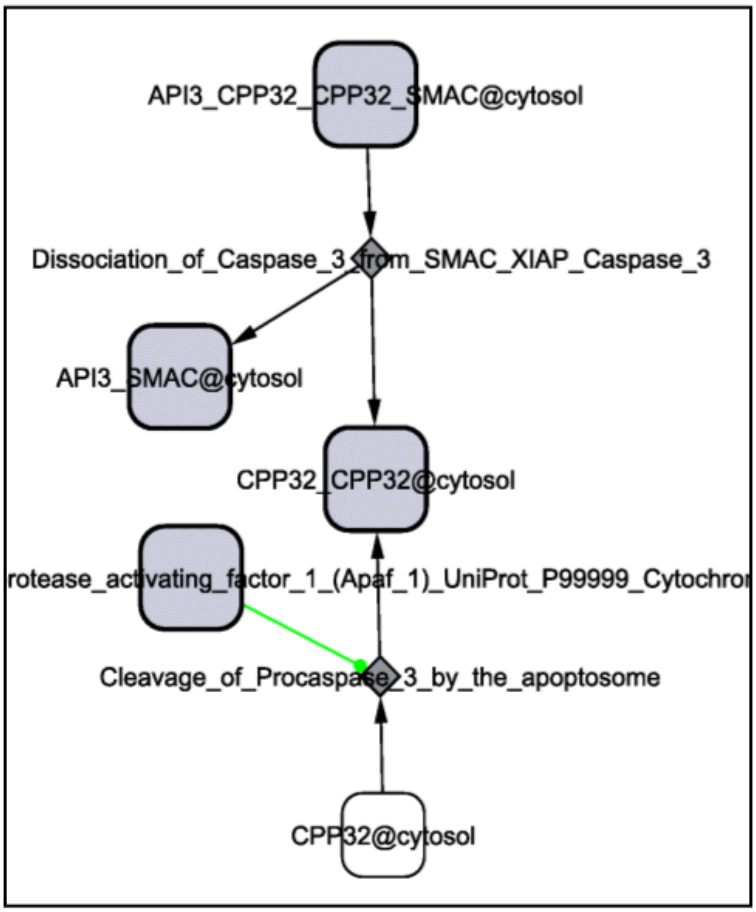


Figure 37: Fragment of Apoptosis from Reactome as Reaction Network.

- Reaction Network.
- Pathway Structure.
- Protein-Protein Interaction.

8.4.1 BioPAX interface as Reaction Network

The Reaction Network interface is a bipartite graph which contains nodes of only two types: species and reactions. Reactants are connected to reactions through edges of type LEFT, products are connected through edges of type RIGHT. Modifier species are connected through CATALYSIS, MODULATION and other edges. See figure 37.

Some BioPAX objects (catalysis, for example) are represented by edges with the corresponding BIOPAX_URI attribute. A chemical species node can correspond to several grouped physicalEntityParticipants, thus, it can have several BIOPAX_URI attributes. When calling BioPAX editor, all of them will be opened.

Standard Reaction Network interface can be exported to pure SBML format (level 2) and serve as a draft for further computational modeling.

8.4.2 BioPAX interface as Pathway Structure

Pathway Structure interface contains only nodes of pathway, pathwayStep and interaction types. The types of the edges connecting them are CONTAINS, STEP and NEXT. See figure 38.

8.4.3 BioPAX interface as Protein-Protein Interaction

Protein-protein Interaction interface contains only entities (not chemical species) with edges of CONTAINS and physicalInteraction type. This interface allows to visualize the composition of complexes like the Caspase3

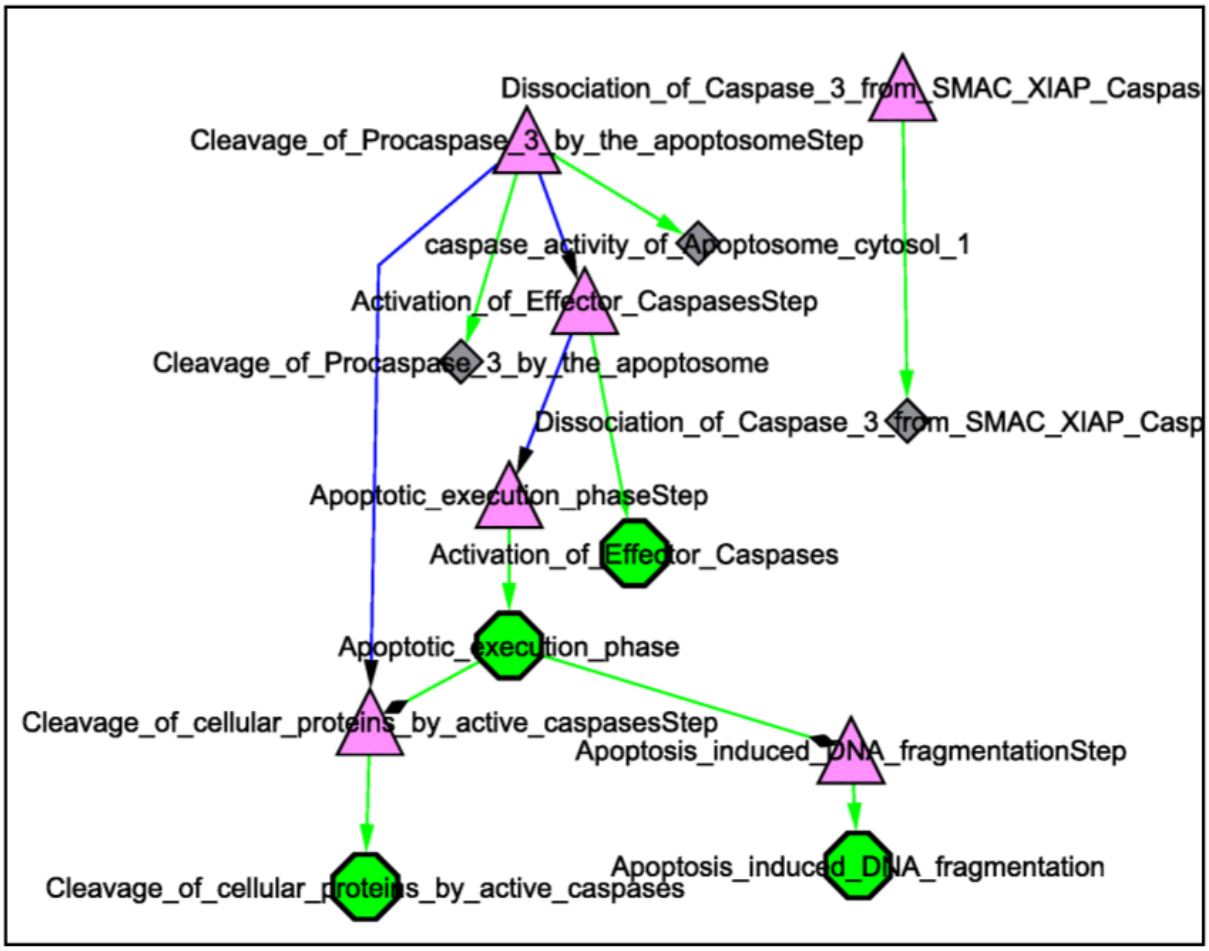


Figure 38: Fragment of Apoptosis from Reactome as Pathway Structure.

example of the Apoptosis pathway (left, first), or, explicit information about protein interaction with TGFB1 (left, second), as in the NetPath TGF-beta BioPAX file. See figure 39.

8.5 AIN file format

The AIN format describes a list of influences between genes, proteins, modified proteins or families. It is a table in ASCII, where the columns are separated by one tabulation (<Tab>) .

The first line must start with the name of each column as follows (the titles are fixed):
ReviewRef ExperimentRef Link ChemType Delay Confidence Tissue Comment
(each space corresponds to a <Tab>on your keyboard).

- For the references (ReviewRef and ExperimentRef), if one wants to include a PUBMED number, it should have the form PMID:123456.
- The Link column describes a connection (activation or inhibition) between two entities, like A->B or A-|B. The entities can be simply the name of a gene or a protein, but it can also be a complex ((C:D)), a phosphorylated protein ((C^p)) or a family. In the latter case, the family can be given explicitly by the list of all its members ((C1,C2,C3)) or implicitly, by un undefined name ((C.)), where the . can be replaced by any character..
- In the other columns, if the user wishes to add more than one word in each field, the sentences need to be inserted between .
- If a field cannot be filled, a simple dot should be inserted.
- A # in first column makes the line comment.

For an example of AIN format, one can open the file ExamplApop.txt in a simple text editor or in spreadsheet as EXCEL. All the information in this AIN file is translated in BioPAX format when the file is imported in Cytoscape via BiNoM.

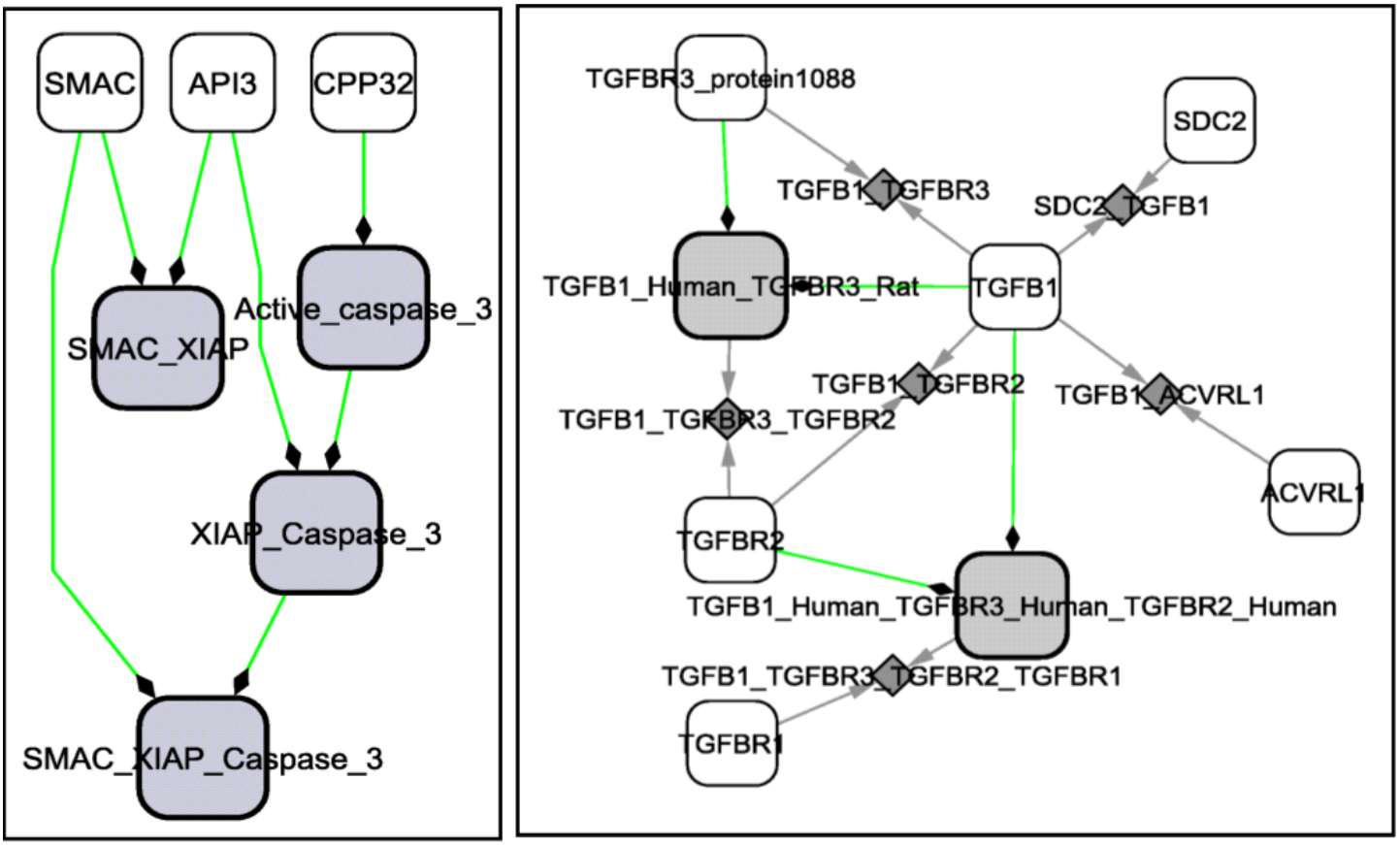


Figure 39: Fragment of Apoptosis from Reactome as Protein Protein Interaction.

8.6 Modularization by shortest path clustering

When only the structure of a network is known, the simplest method to agglomerate nodes in a network is to put the closest nodes together. And so modules may have the fewest links between them. This method can lead to an algorithm of modularization of an oriented network. The notion of closeness and the process of creating modules are to be clarified.

The distance between nodes is based on the length of the shortest paths and the number of occurrences if several paths are equal (the equality of the shortest path is frequent in a strongly connected network). The distance from node 1 to node 2 is generally different from distance from node 2 to node 1.

The used distance is the minimal linkage applied to the base distance, which is necessary to respect the triangular inequality. The distance between A and B is the minimum of distances from nodes in A to nodes in B and from nodes in B to nodes in A. And so, the agglomerative hierarchical clustering can be applied to build modules.

To avoid too speed increasing of clusters, they are ranked in a queue and the last created cluster is put at the end of the queue. For the same reason, nodes are sorted by in degree (sources in first). Despite of these precautions, the algorithm applied to strongly connected network gives unbalanced clusters (often a hudge cluster and several tiny clusters). So, a ceiling number of nodes in a cluster must be fixed.

The agglomerative clustering gives 1 cluster at the end, which has no interest. Thats why; these 2 stop conditions are added:

- The length of the shortest path between 2 clusters reaches the maximal length.
- The number of clusters to be compared in the queue is less than 2.

The first stop condition make that too far clusters are not merged. When the last cluster to be created contains more than the maximal number of nodes, the largest cluster is excluded from the queue. Only the clusters remaining in the queue are to be compared by distance and they must be 2 or more.

The next page shows 3 examples (network inspired by toynet). If the maximal length of the shortest paths is 1, nodes inside clusters are connected as a clique in a not oriented graph. But, if not, it may not be the case.

From a **practical point of view**, the input of ceiling number of nodes and maximal length of the shortest paths gives a set of not intersecting sub-networks. They are a partition of the network; their union is the whole network. This process is only useful for connected networks. Obviously, isolated nodes or sub-networks are not merged unless the maximal distance is infinity.

8.7 GLOSSARY

8.7.1 BioPAX

BioPAX is an OWL (Web Ontology Language) document designed to exchange biological pathways. BioPAX format provides separate layers of information: information about the reactions involved in the networks along with the participants, information about the structure of the pathway, and information about the protein-protein interactions.

8.7.2 CellDesigner

CellDesigner is a structured diagram editor for drawing gene-regulatory and biochemical networks. Networks are drawn based on the process diagram, with graphical notation system proposed by Kitano.

8.7.3 BiNoM Index

Directed labeled graph representing the objects in CellDesigner and BioPAX ontologies and their connections. Index maps only the information needed to display it and to identify the relevant information in the original CellDesigner or BioPAX files.

8.7.4 BiNoM interface

Part of the BiNoM index (subgraph) visually presented by Cytoscape network. There are standard interfaces (Reaction network, pathway structure, protein interaction) which can be combined to construct a user-defined interface.

8.7.5 Optimal / suboptimal shortest paths

Shortest paths in weighted directed graph paths in the graph between source and target nodes with minimal sum of weights of the edges making the path. Suboptimal path is constructed by removing all edges in all shortest paths one by one and one at a time and finding the shortest path.

8.7.6 Strongly Connected Components (SCC)

A subgraph in a directed graph, in which there is path from any node to any node

8.7.7 Relevant cycle

Any cyclic path in the graph which can not be decomposed further into simpler cycles

8.7.8 SBML

Systems Biology Markup Language (SBML) is a standard for representing models of biochemical and gene-regulatory networks.

.

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] L. Calzone, A. Gelay, A. Zinovyev, F. Radvanyl, and E. Barillot. A comprehensive modular map of molecular interactions in rb/e2f pathway. *Molecular systems biology*, 4(1), 2008.

[3] 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.

[4] 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.

[5] B. Novak, A. Csikasz-Nagy, B. Gyorffy, K. Nasmyth, and J.J. Tyson. Model scenarios for evolution of the eukaryotic cell cycle. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1378):2063, 1998.

[6] 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.

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

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

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