institutCurie

Biological Network Manager
Version 1.0

Andrei Zinovyev and Laurence Calzone

Institut Curie, Service de Bioinformatique

Figures (needs to be updated in text)

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 (Stromback and Lambrix, 2005) and others). Many softwares, which are centered on the description and representation of biological pathways, adopted these standards. CellDesigner (Kitano et al., 2005) and Cytoscape (Shannon et al., 2003), 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/binom/

1.1 List of files used in the manual

All files used in this manual can be found on the webpage at:
http://bioinfo.curie.fr/projects/binom.

- *BIOMD0000000111.xml*: Novak's model deposited on BioModels database
- *M-Phase.xml*: Modified CellDesigner version of Novak's model often referred to as the M-Phase network in this manual.
- *M-Phase.owl*: Modified BioPAX version of Novak's model
- *M-Phase.cys*: Cytoscape session of M-Phase example (section 2.3 of this manual)
- *Apoptosis.owl*: Apoptosis sub-diagram of Reactome database
- *Example.cys*: Cytoscape session with the simple example presented in section 4.4.
- *RBPathway.xml*: comprehensive network of RB/E2F pathway
- *RB_Manual.cys*: Cytoscape session of section 3.1.

2 Working with BiNoM

2.1 BiNoM I/O: importing and exporting documents

We will use a slightly modified version of the mathematical model published by Novak et al. (Novak et al., 1998) of the fission yeast cell division cycle as an illustration of the different functionalities developed in BiNoM. For some of them, a more appropriate model of the apoptosis pathway extracted from Reactome database will be used (no description of this model is necessary for our purpose).

> *Description of the Novak et al. model*
> The Novak et al. model is a description of the M phase dynamics in Schizosaccharomyces pombe. As soon as the cyclin CDC13 is synthesized, it binds to the cyclin-dependent kinase CDC2 to form a complex that controls M phase entry. There are different ways to control the complex activity: (1) through phosphorylation: the kinase WEE1, already present, phosphorylates CDC2 at Tyr15 and Thr14 and keeps the complex inactive. The CAK complex further phosphorylates CDC2 at Thr167 which allows the complex to activate as soon as the phosphatase CDC25 dephosphorylates CDC2 at Tyr 15 and Thr14. Once active the complex keeps CDC25 active and WEE1 inactive; (2) through inhibition: in the presence of the cyclin dependent kinase inhibitor Rum1, the complex is immediately sequestered and kept inactive in the newly-formed trimer. As the concentration of the cyclin slowly grows, it reaches a threshold that activates the complex enough to phosphorylate and shut off Rum1; (3) through degradation: CDC2/CDC13 initiates its own degradation pathway by activating Slp1 which participates in the ubiquitination of the cyclin (Figure 1, 2 and 3).

Import of SBML & CellDesigner files.

To import a model in Cytoscape, there exist several ways:

2.1..1 Import from a database

(This part is not specific to BiNoM).
File => Import => Network (multiple file types).

The model that will be studied in this manual can be found in the BioModels database under the following publication ID: 12779461. The SBML L2 V1 format of the model is saved as *BIOMD0000000111.xml* on a local file of the user's computer and opened in Cytoscape.

[Note that, once the model is imported, the layout can be chosen from the menu bar (**Layout => Organic**, for example)]

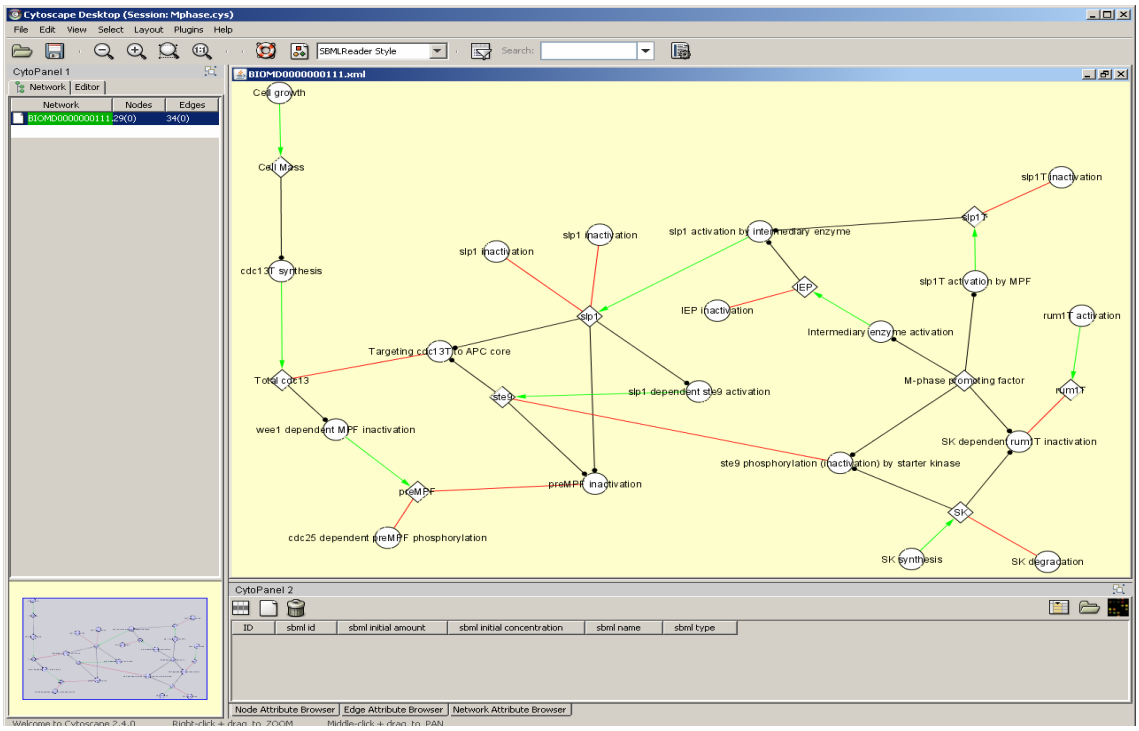


Figure 1. Import of SBML model from BioModels database (standard Cytoscape SBML Reader plugin)

2.1..2 Import from a CellDesigner or Cytoscape file

Plugins => BiNoM I/O => Import CellDesigner Document from file.

The model can be drawn – or downloaded – in CellDesigner (Figure 2, description of the model above) 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 3).

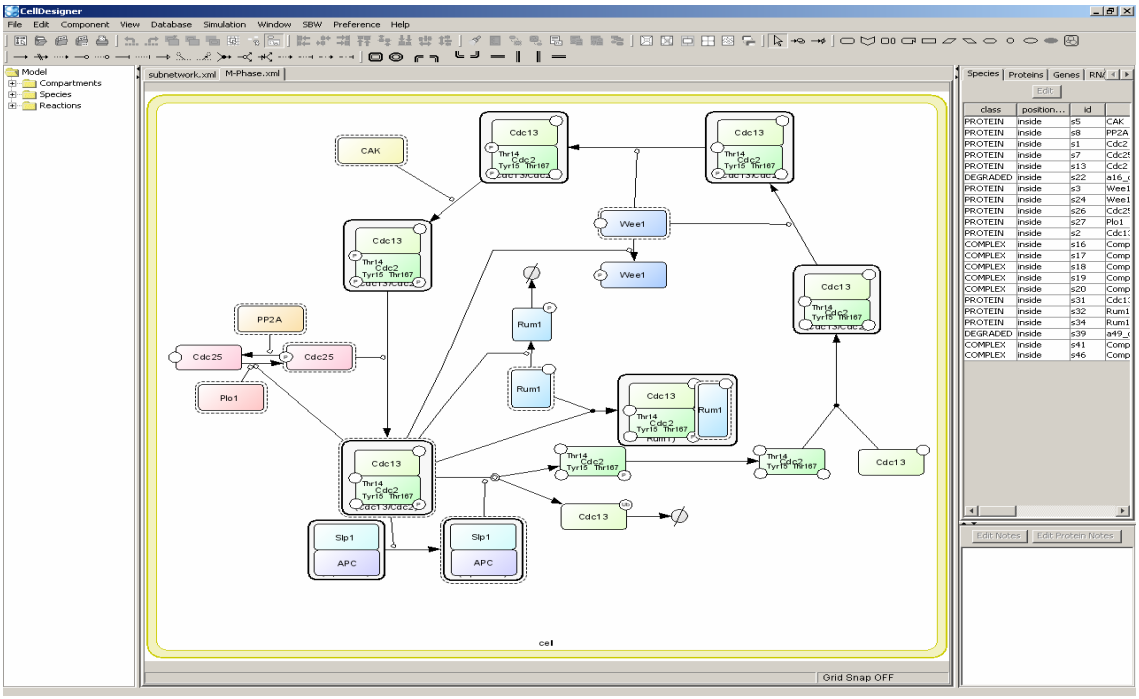


Figure 2. CellDesigner view of the cell division cycle model of fission yeast (Novak et al.)

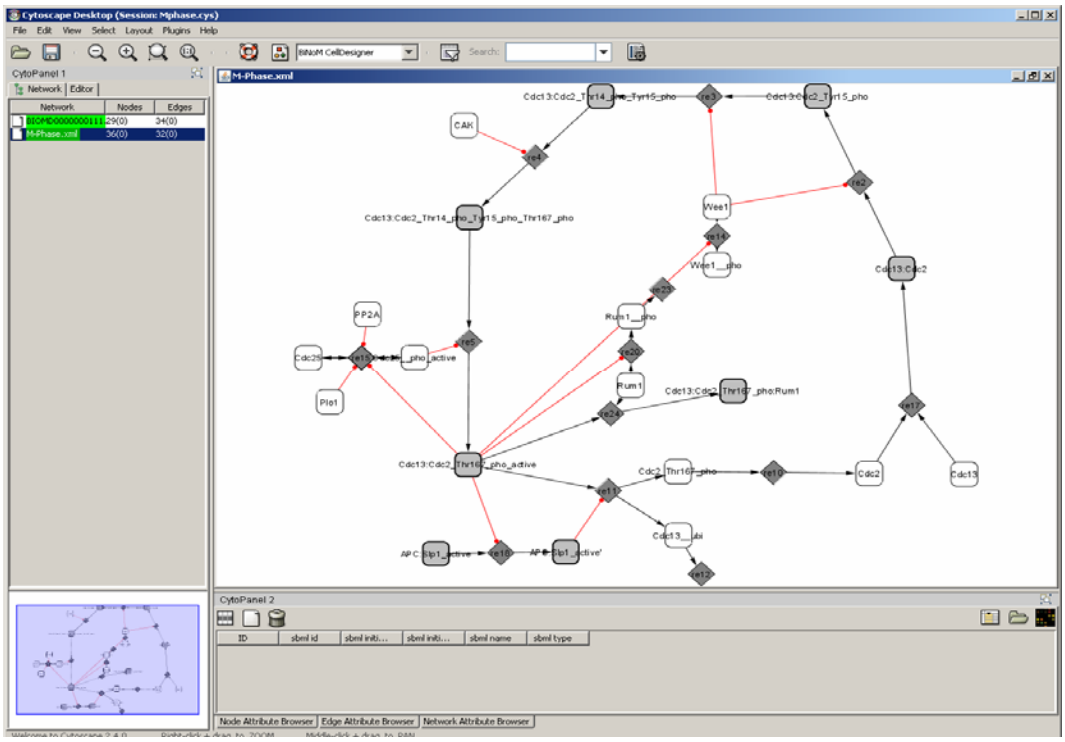


Figure 3. Cytoscape view of the cell division cycle model of fission yeast from a CellDesigner document

Figure 2 and 3 show the same model viewed by CellDesigner and Cytoscape respectively. The layout information from CellDesigner is imported automatically into Cytoscape.

2.1..3 Import from a URL

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.

Import of a BioPAX file

Plugins => BiNoM I/O => Import BioPAX Document from file.

The model *M-Phase.owl* 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 4).
- 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 5).
- Protein interaction: *M-Phase PP* shows which proteins interact with each other.

For more details on BioPAX, its interfaces, etc, go to section 5.3.

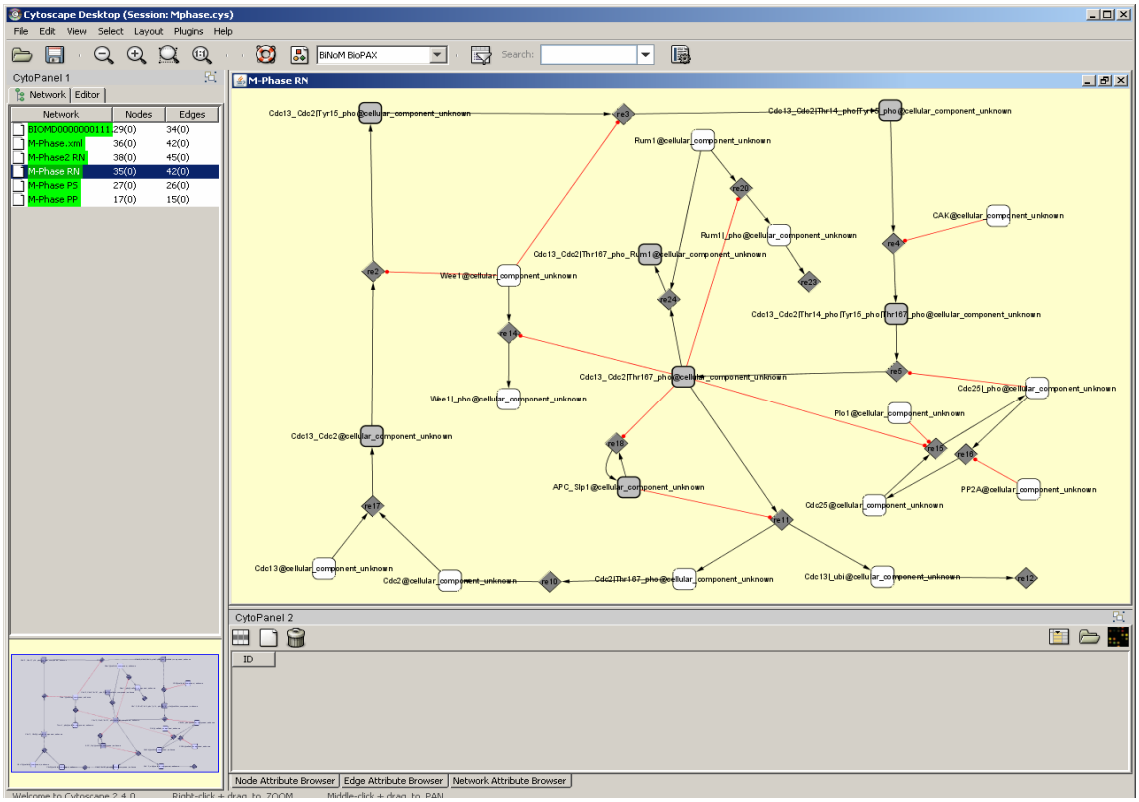


Figure 4. BioPAX view of Novak et al. model.

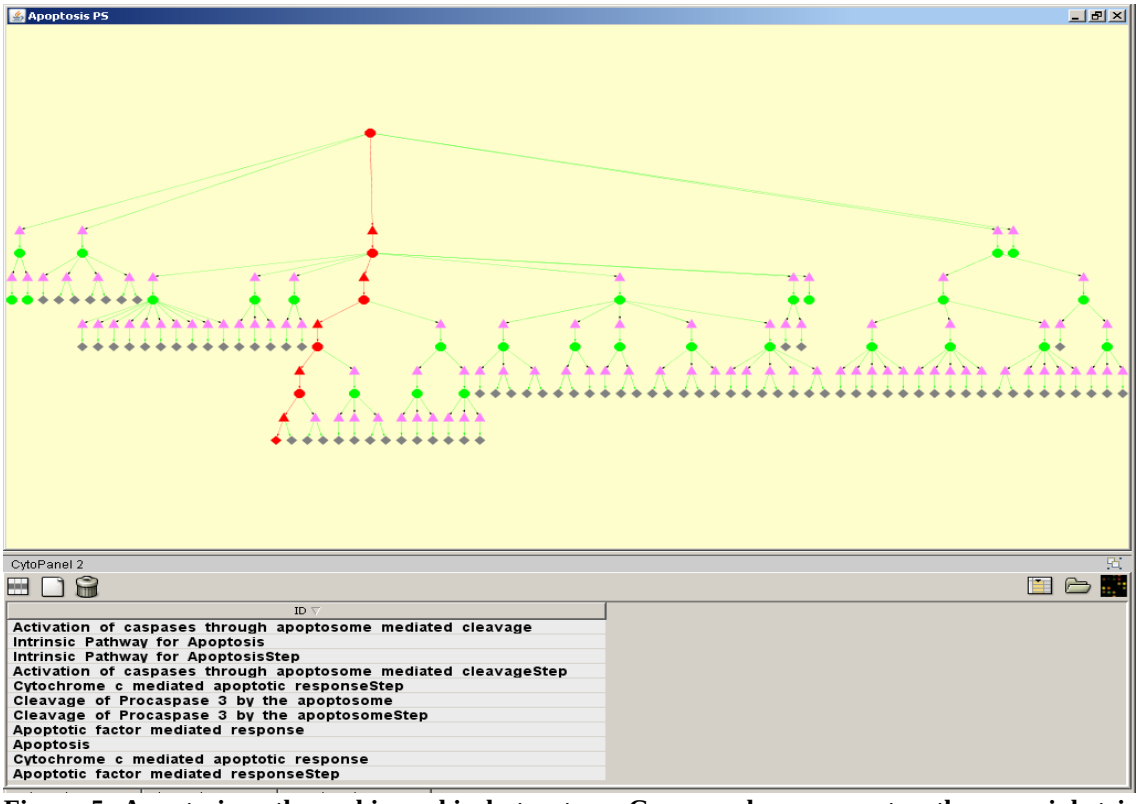


Figure 5. Apoptosis pathway hierarchical structure. Green nodes represent pathways, pink triangular nodes represent steps, and grey nodes represent reactions. From the apoptosis node (top node in red), the cell can choose through 5 different paths. The red-colored path shows one of them, the activation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

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 5) 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 5).

Export of networks

The Cytoscape networks can be exported in BioPAX, CellDesigner or SBML format…

Plugins => BiNoM I/O => Export current network to BioPAX
Plugins => BiNoM I/O => Export current network to CellDesigner
Plugins => BiNoM I/O => Export current network to SBML

… provided that they are associated to existing CellDesigner or BioPAX:

Plugins => BiNoM I/O => Associate BioPAX Source
Plugins => BiNoM I/O => Associate CellDesigner Source

More precisely, BiNoM is able to convert CellDesigner to BioPAX, and BioPAX reaction network interface to pure SBML level 2 (not to the CellDesigner dialect). 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 section 5 about the BiNoM data model.

When the content of the BioPAX file is modified (through BioPAX property editor), it can be to be saved as a whole (not only visible part):

Plugins => BiNoM I/O => Save whole associated BioPAX as

Otherwise, all modifications made in the different interfaces are lost (as it is the case in a WORD document, changes are visible but only recorded permanently when the document is saved).

2.2 BiNoM Analysis: structural analysis of reaction networks

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.

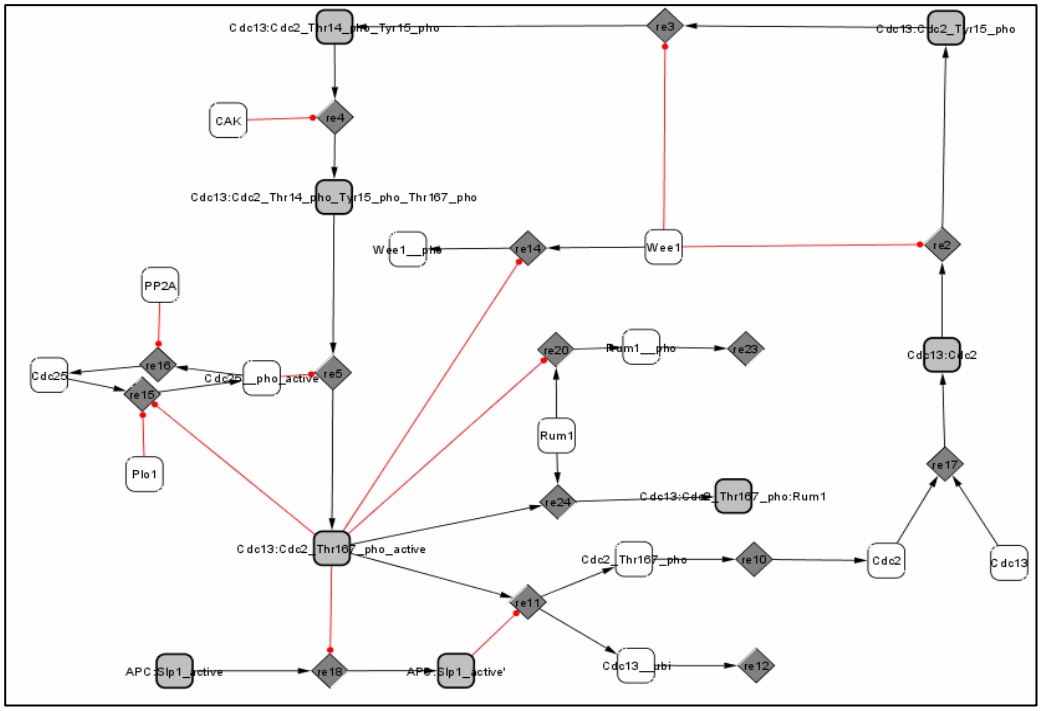


Figure 6. Cytoscape view of the M-Phase network.

From the menu **Plugins => BiNoM analysis**, we review all the functions one by one:

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*).

Get strongly connected components

Plugins => BiNoM analysis => Get strongly connected components

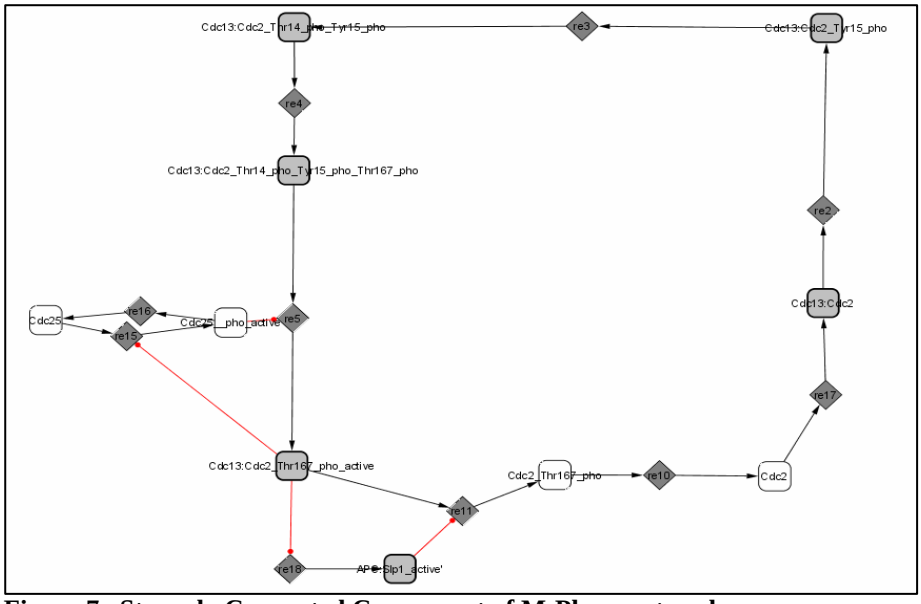


Figure 7. Strongly Connected Component of M-Phase network

Based on Tarjan's algorithm (Tarjan, 1972), the strongly connected components are isolated. In simple words, the obtained network, *M-Phase.xml_scc1*, insures that there exists a path from one node to another and deletes the components which do not respond to this requirement.

Prune graph

Plugins => BiNoM analysis => Prune graph

Pruning the graph is equivalent to separating the network into three parts: 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 (Broder et al., 2000). In our example, the central cyclic part is the same as Figure 7, 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).

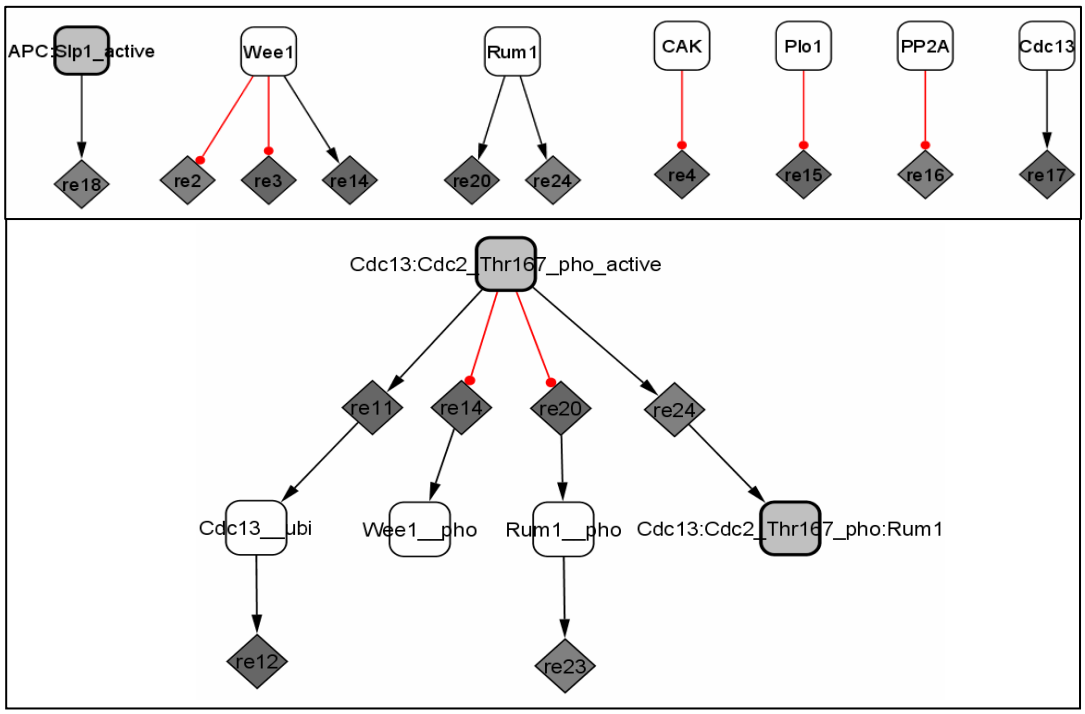


Figure 8. Prune the graph. (a) Incoming flux: molecules involved in the IN part of the network, and (b) Outgoing flux: molecules involved in the OUT part of the network.

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, 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.

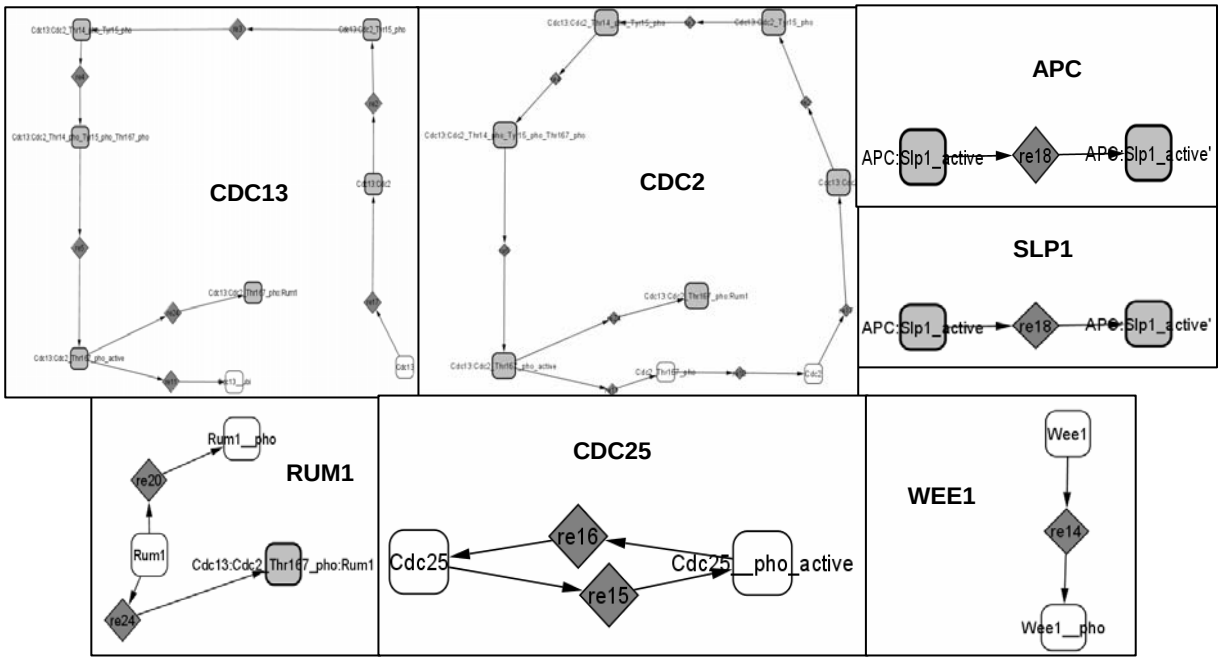


Figure 9. Material components.

Get Cycle Decomposition

Plugins => BiNoM analysis => Get cycle decomposition

This command decomposes the network into relevant directed cycles (Gleiss et al., 2001), using a modification of the Vismara's algorithm (Vismara, 1997). Often, this feature gives information about the life cycle of a protein or a complex, about the feedbacks of the studied network, etc. Note that the union of all the cycles corresponds to the strongly connected component figure. **Attention**: 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.

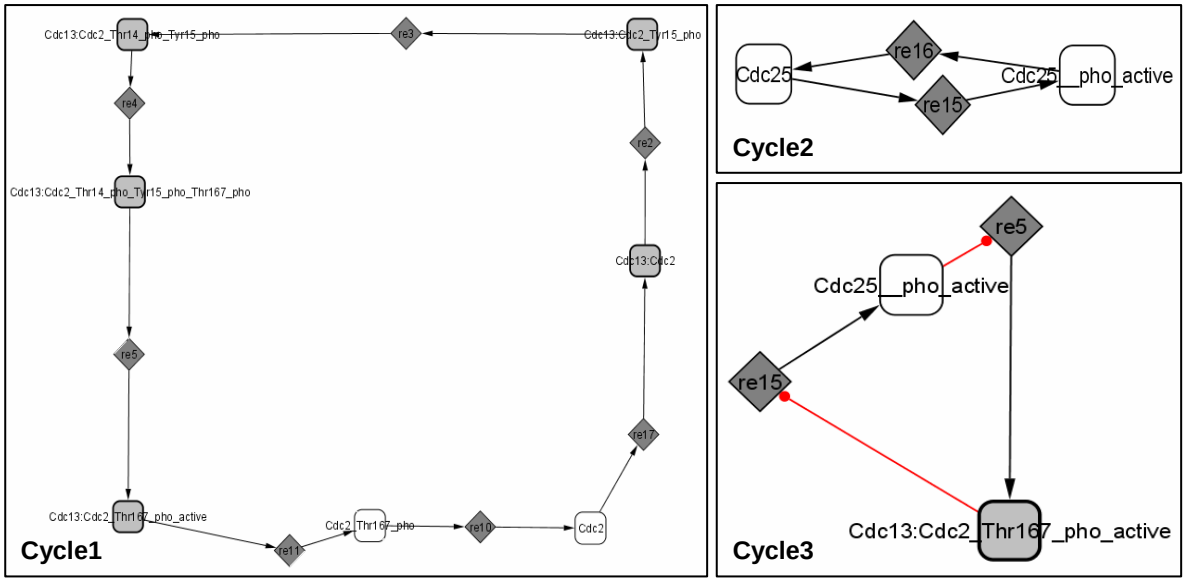
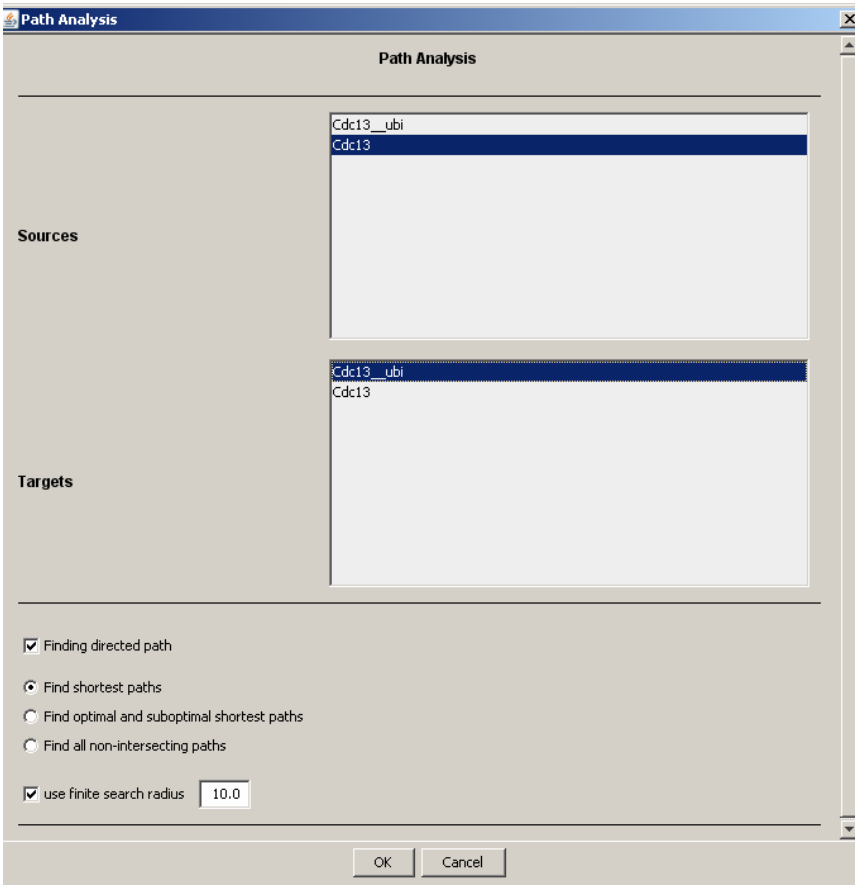


Figure 10. Minimal cycle decomposition of the M-Phase network. Cycle 1 includes CDC2 and CDC13 proteins, Cycle 2 CDC25 and Cycle 3 shows the feedback existing between CDC13/CDC2 and CDC25.

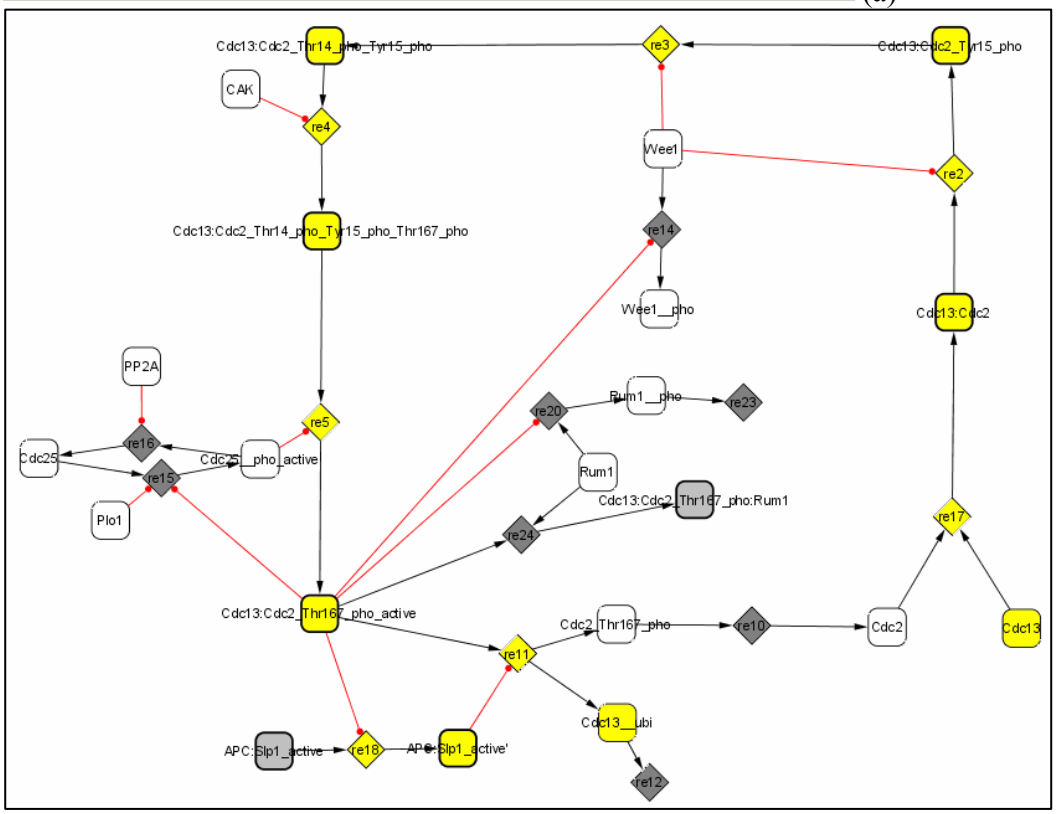
Path analysis

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. 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. **Attention**: 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.



(a)



(b)

Figure 11. BiNoM Path Analysis. (a) 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). (b) All the paths leading from one molecular species (Cdc13) to another (Cdc13_ubi, ubiquitinated form of Cdc13) are highlighted in yellow.

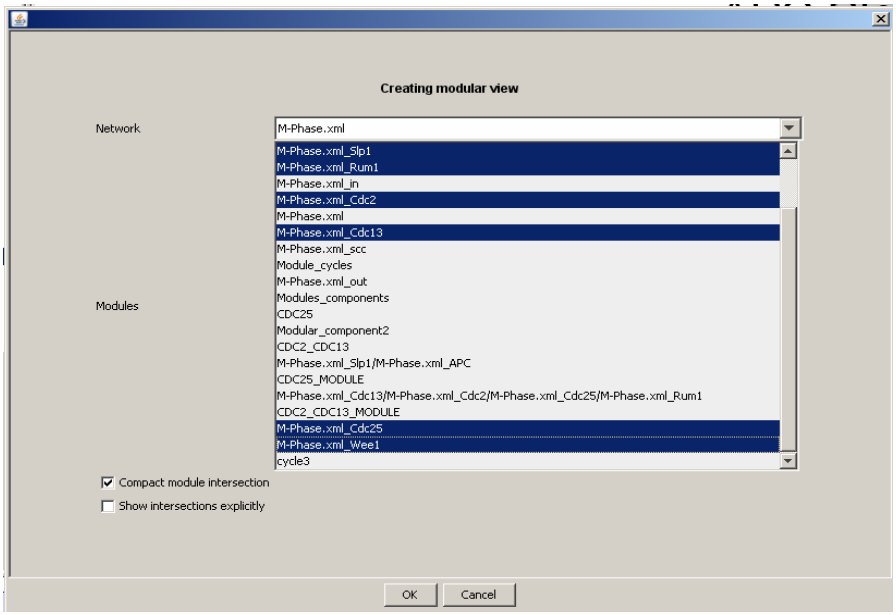
Generate modular view

Plugins => BiNoM analysis => Generate modular view

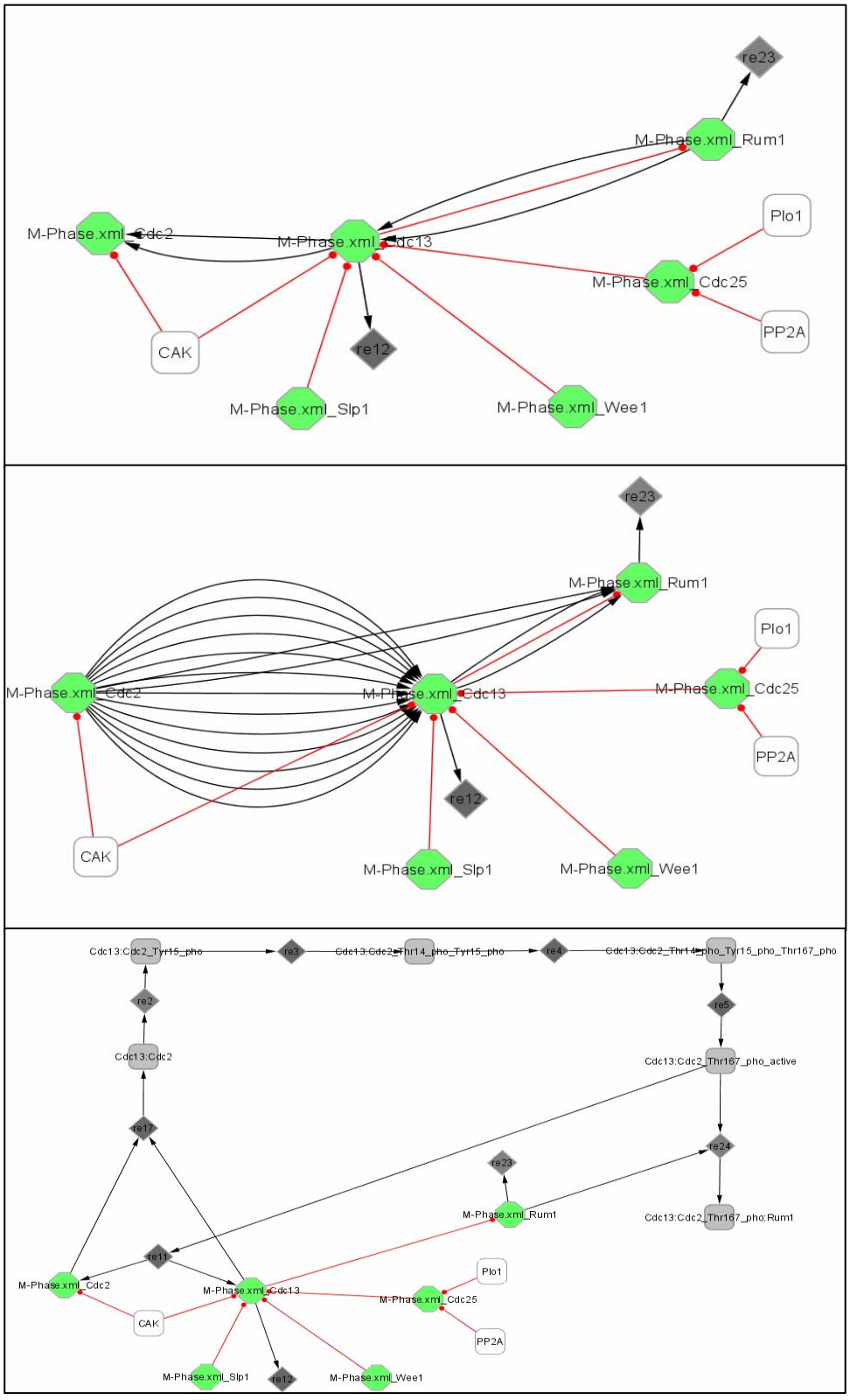
Given the initial diagram and some modules (which could be sub-networks of the initial network), it is possible to reconstruct a modular view of the network. For our example, we choose the initial network to be *M-Phase.xml* and the subparts or modules, the seven sub-networks corresponding to the material 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 12).

There are different types of modular views. The modules are connected by: (1) the number of shared interactions (Figure 12b, upper panel); (2) the number of shared nodes (reactions + species) for which case the box "Compact module intersection" must be checked (Figure 12b, middle panel); and (3) the shared nodes and reactions showed explicitly (Figure 12b, lower panel).



(a)



(b)

Figure 12. BiNoM modular view of the newtork. (a) Pop-up window in which the initial graph and the modules are specified. (b) The resulting modular network (upper panel) with compact module intersections (middle panel) and with explicit intersections (lower panel).

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 12b (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 13a 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 10. They are obtained by clustering the three cycles into two (cycle 1 + cycle2/cycle3) and organized into a modular view (Figure 13b).

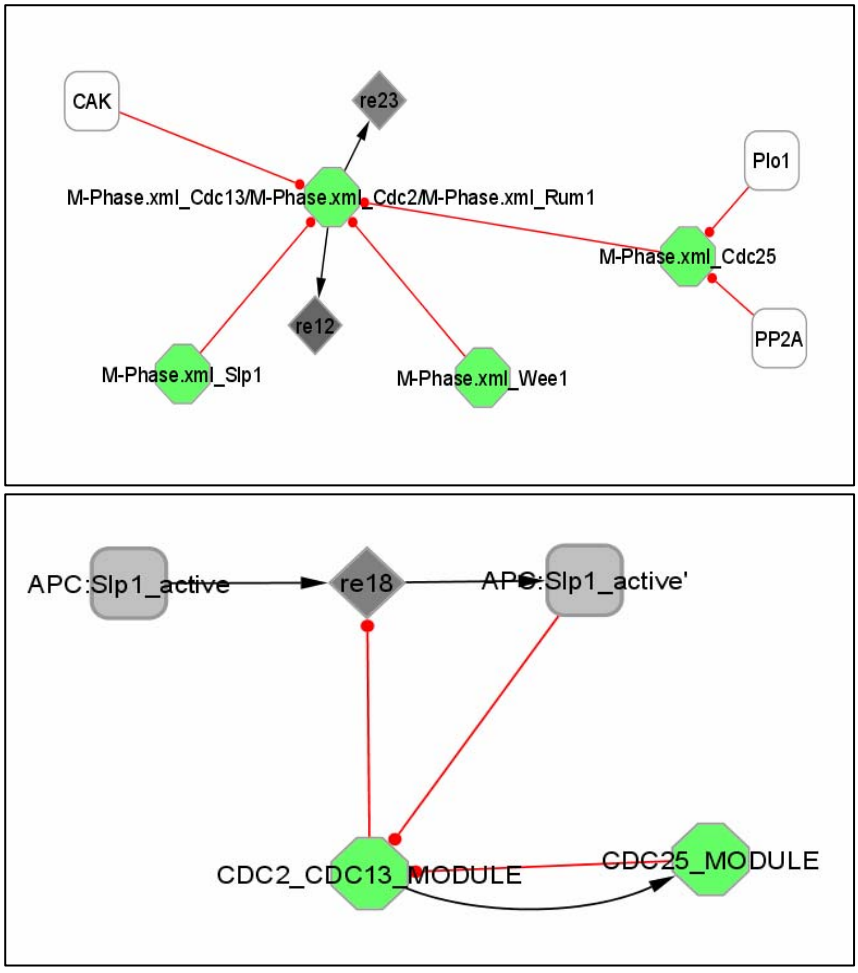


Figure 13. Clusters of modules. The obtained diagram is a compact modular view of the M-Phase network (a) using the material decomposition and material components clustering and (b) using the relevant cycle decomposition and cycle clustering.

Mono-molecular reactions 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 14: 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.

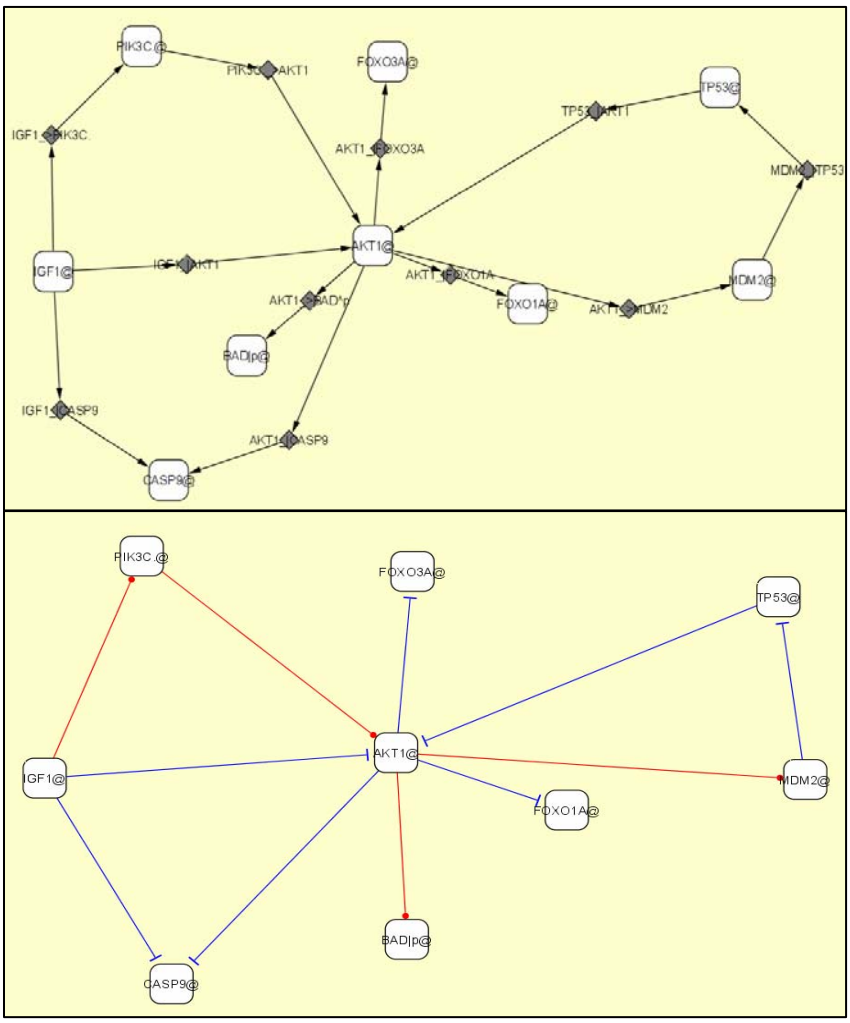


Figure 14. From a BioPAX network (upper panel) to an influence graph (lower panel).

Extract reaction networks

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).

2.3 BiNoM BioPAX Utils

BioPAX Property Editor

Plugins => BiNoM I/O => BioPAX Property Editor

All the information available in a BioPAX file can be easily retrieved using the **BioPAX Property Editor** function. A component on the diagram must be selected first (CDC2 in Figure 6) and a window appears with all available information concerning the molecule.

Figure 15. BioPAX Property Editor. (a) example of the properties concerning CDC2 component in M-Phase model and (b) example of "apoptosis pathway" node properties.

In the menu of the Property Editor, several options are offered:

- *Display valid attributes* / *Display all attributes* hides all the empty fields (for example, in Figure 15b: *Availability* or *Evidence* have *<empty object list>* and would be hidden) / shows all the available fields, even the empty ones.
- << and >> correspond to back or forward buttons and follow the historical exploration of the Property Editor (similar to 'Back' and 'Forward' buttons of a network browser).

- *Close current tab* or *Close all tabs* closes the current page of the property editor or all the open pages.
- *Display* / *Edit* shows a simple display of the page editor where no change can be made (Figure 15b) / allows changes in the fields by adding, removing or updating information (Figure 15a). For the latter, click first on the **Edit** tab on the upper menu, then on **update** situated near the field to modify. In Figure 15a, as an example, some comments were added manually: "CDC2 is a kinase that binds to CDC13 to form a dimer". In the *Apoptosis* example (Figure 15b), extensive information is already available concerning the pathway, references, etc.

For more details on BioPAX description standard, visit the webpage: http://www.biopax.org/

BioPAX Class Tree

Plugins => BiNoM I/O => BioPAX Class Tree

All the statistics concerning the pathway are listed: the number of reactions, associations or catalyses, the number of proteins or complexes, etc. More information can be accessed by selecting a specific object which, when clicked on, leads to the BioPAX Property Editor window (see 2.3.1).

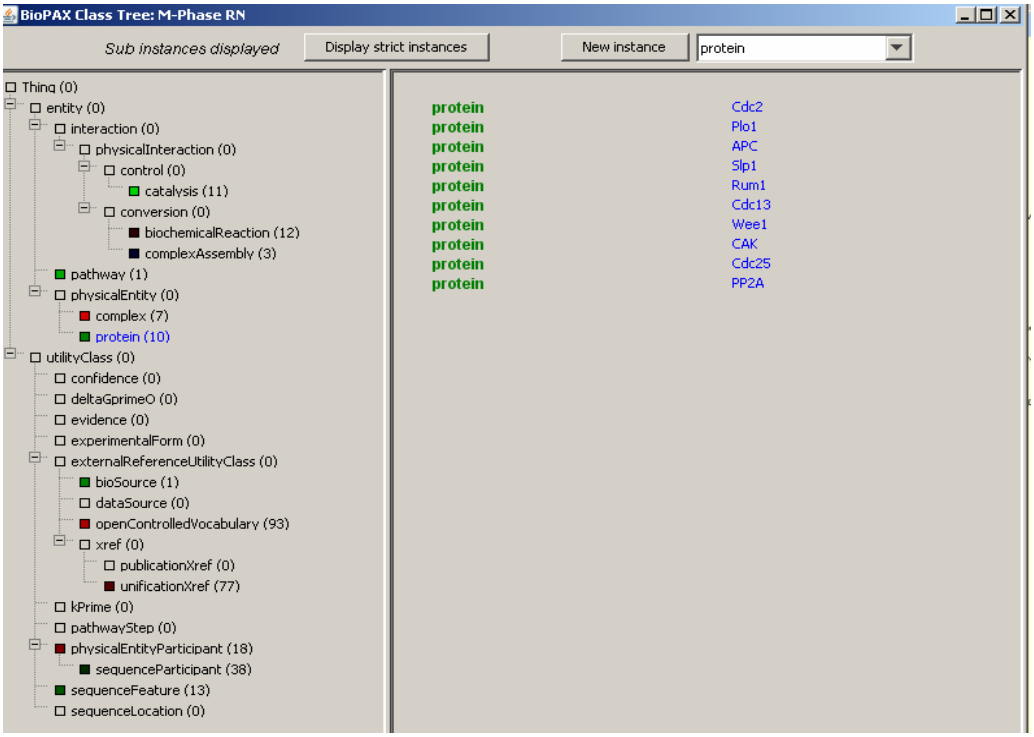


Figure 16. BioPAX Class tree. On the left frame, the model is described in terms of interactions, entities, etc. On the right frame, the proteins, selected in the left frame, are listed. The links are clickable and open a BioPAX Property Editor window.

To complete the network, the user can easily add new information or a new protein, protein complex, type of interaction, etc., by clicking on the *New Instance* tab.

Using Naming Service Names / URI Names

Plugins => BiNoM I/O => Using Naming Service Names or
Plugins => BiNoM I/O => Using URI Names

In the BioPAX Class Tree, protein names can have either URI names (Uniform Resource Identifier used to give a unique identification to proteins) or "BiNoM Naming Service" names. For example, for the apoptosis pathway, the protein BAD is referred to as "*UniProt_Q92934_Bcl2_antagonist_of_cell_death_BAD__Bcl_2_binding_component_6__Bcl__XL_Bcl_2_associated_death_promoter__Bcl_2_like_8_protein*" in the URI case and just "*BAD*" in the BiNoM Naming Service case. For the rules of how BiNoM generates names see section 5.3.

Synchronize networks with BioPAX

Plugins => BiNoM I/O => Synchronize networks with BioPAX

This command updates all the interfaces according to the changes made to individual BioPAX objects.

2.4 BiNoM BioPAX query: working with huge BioPAX files

Plugins => BiNoM BioPAX Query

The purpose of the functions related to the query language is to work with huge BioPAX files and extract from the BioPAX documents only the information that is of interest. For this part, we will use the apoptosis example initially extracted from Reactome database: *Apoptosis.owl*. This set of functions can be used with big pathway databases already exported to BioPAX: Reactome, BioCyc, NetPath (see http://www.biopax.org for the complete list).

Generate Index

Plugins => BiNoM BioPAX Query => Generate Index

Using this function BiNoM maps the content of BioPAX file onto a labeled graph (referred to as *index*). It creates an **.xgmml* file from an **.owl* one. For the definition of BioPAX index, see section 5.

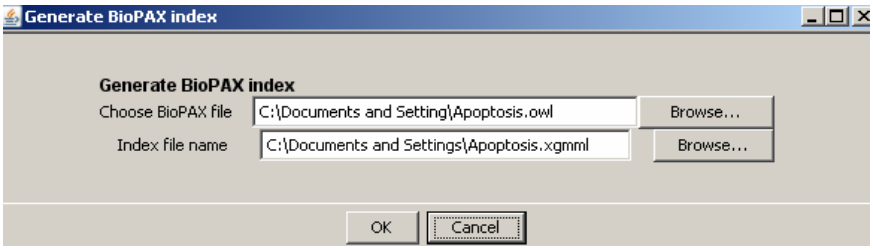


Figure 17. Generate BioPAX Index.

Load Index

Plugins => BiNoM BioPAX Query => Load Index

Once the *xgmml* is created, it can be loaded into memory. The index is global object, i.e. only one index can be used at a time.

Load Index loads the index file from *xgmml* format. Together with the index, you can also upload a tab-delimited "accession number file" which corresponds to a list of synonyms for the genes/proteins ids used in a network (see an example of the content of some accession number file at Fig.18). An entity in the index can be identified by its id, by any XREF attribute (see section 5), by node name, or by any synonym from the accession table (if it is provided).

cAMP-dependent protein kinase	ID001
cAPK	ID001
InterPro:IPR000719	ID001
PKA catalytic subunit	ID001
PKAc	ID001
PKAc-beta	ID001
Protein kinase	ID001
protein kinase A	ID001
protein kinase A catalytic subunit	ID001
protein kinase A catalytic subunit	ID001
Protein kinase C-terminal domain	ID001
Serine/threonine protein kinase	ID001
Serine/threonine protein kinase, active site	ID001
Tyrosine protein kinase	ID001
ADP-ribosylation factor	ID002
ADP-ribosylation factor family	ID002
ARF	ID002
Rab	ID002
Ran family	ID002
Ras family	ID002
Rho family	ID002
Rho/Rac family	ID002
small G-proteins	ID002
small GTPases	ID002
small guanyl-nucleotide binding proteins	ID002
activator of G-protein signaling	ID003
AGS	ID003
InterPro:IPR001806	ID003
InterPro:IPR001806	ID003
InterPro:IPR003577	ID003
InterPro:IPR005225	ID003
InterPro:IPR005225	ID003
p21ras	ID003
Rad	ID003
Ral	ID003
Rap	ID003
Ras	ID003
Ras family	ID003
Ras GTPase superfamily	ID003
Ras GTPase superfamily	ID003
Ras homolog enriched in brain	ID003
Ras small GTPase, Ras type	ID003

Figure 18. Example of accession Number file. First column is a synonym (which can have structure <database>:<standard_id>), the second column is the id used inside the BioPAX file.

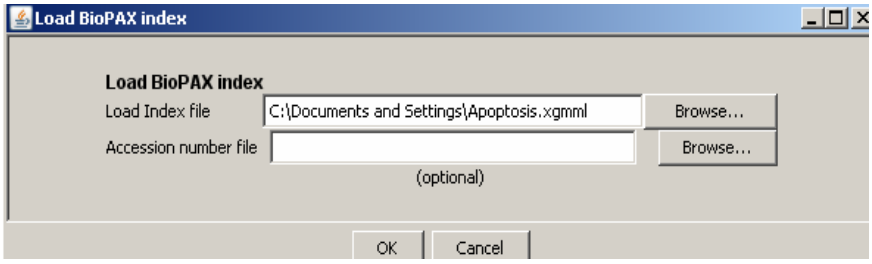


Figure 19. Load Index dialog.

Display Index info

Plugins => BiNoM BioPAX Query => Display Index info

This command opens a window indicating the name of the graph, the name of the file, the accession number file, when available, the number of records, and the various statistics of the index: number of publications, proteins, physicalEntities, complexes, biochemical reactions, pathways, pathwaySteps, catalyses, and modulations (necessary proteins for catalyses).

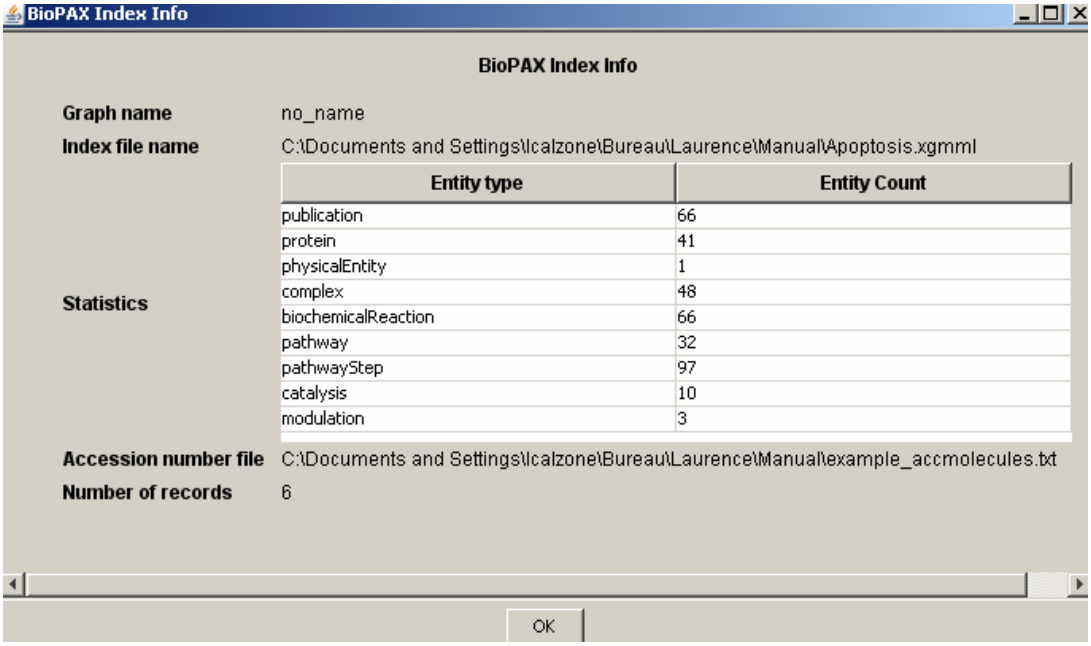


Figure 20. Display Index info.

Select entities from the index

Plugins => BiNoM BioPAX Query => Select entities from the index

The BioPAX document is often too big to find the protein or gene that needs to be studied. To access it easily and rapidly, it is possible to find the component directly with this command and build a specific network around that molecule.

For example, in *Apoptosis.xgmml*, we choose to find the caspases 8 and extend the network around it. When choosing **Plugins => BiNoM BioPAX Query => Select entities from the index**, a dialog window pops up and offers the possibility to find a protein or a gene by its name or id or XREF attribute or synonym, from the current network when a network is already opened, or from the list of identities associated with the BioPAX index (Figure 21).

For our example, we choose the second option. To increase the probability to find the protein in the list, we propose, in Figure 21, three different versions of the same name: CASP8, Caspase8 or caspases_8, all separated by space (the separator can be also comma and semi-colon or line-break symbol). One of them (CASP8) corresponds to the name from the BioPAX list and a new network is created with only one protein, CASP8 (= MCH5 in the index), at the center of it. The other ones were not found (see output in Figure 27). It is also possible to select more than one entity, in this case, the components all appear in the same window.

The output is chosen to appear in a new network (selection is made at the bottom of the dialog window in Figure 21) but it is also an option to view several genes or proteins in the same network by checking "output in the current network".

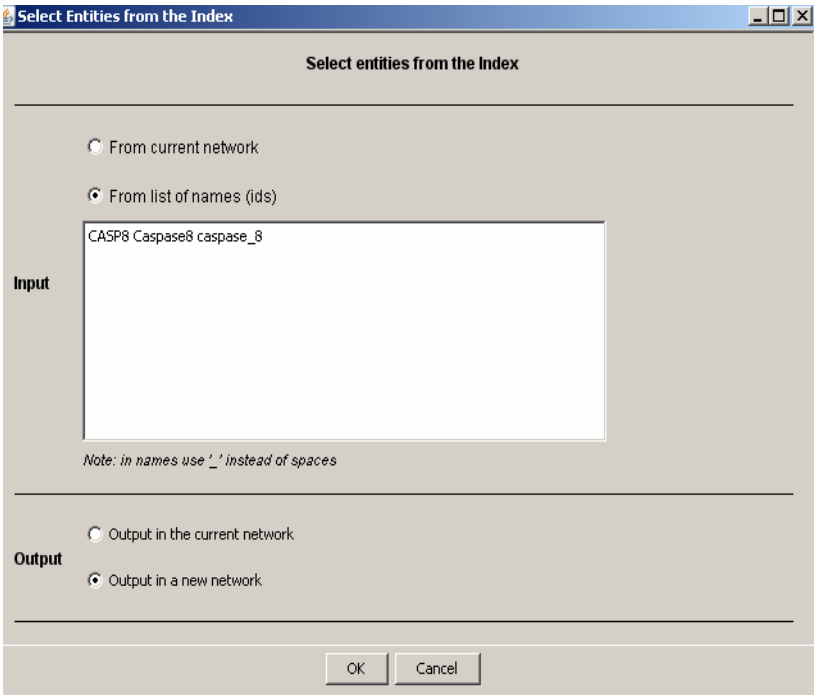


Figure 21. Select entities from the index.

A network is created with only one node, caspase 8, called MCH5 in the index. Note that for this part, it is advised to use the BiNoM BioPAX visual style to view the resulting network.

Standard Query

Plugins => BiNoM BioPAX Query => Standard Query

This command proceeds through a series of actions that will extend or make the studied network more specific to the user's needs.

Let's start with diverse queries from the network created for the Caspase 8 entity. A dialog window opens:

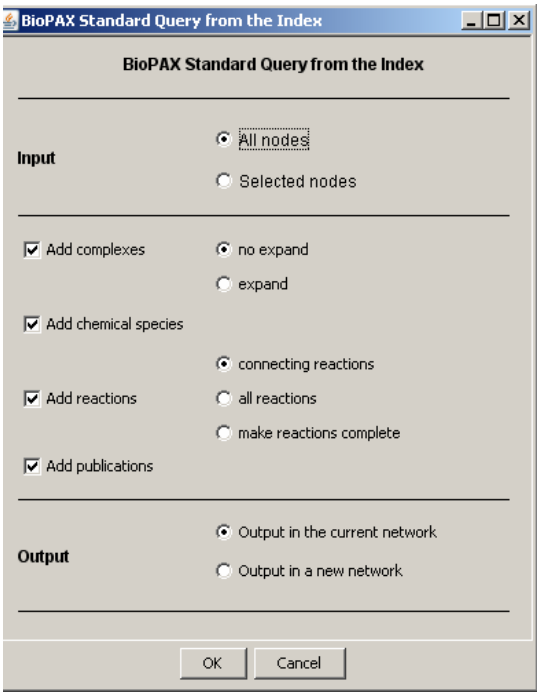


Figure 22. BioPAX Standard Query : dialog window.

All the options proposed in the dialog window (Figure 22) are listed here:

- In the input section

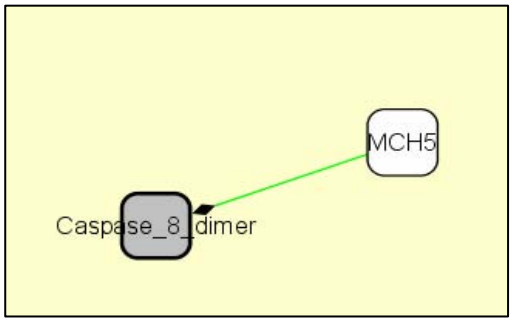
All nodes / Selected nodes: In the network, you can submit queries that concern all the nodes in the network or only the selected nodes (highlighted in yellow).

- Once you decide on which proteins you wish to work, you can:

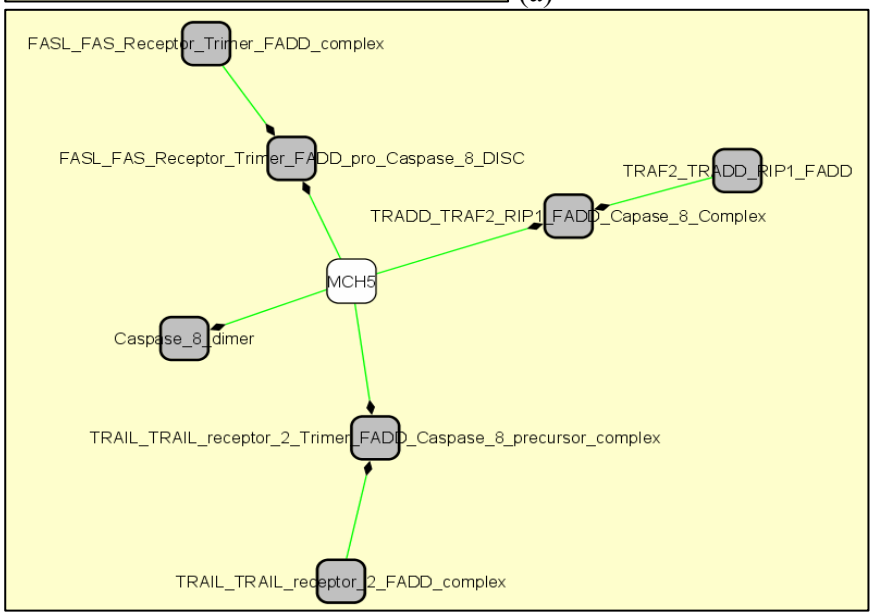
- Add complexes

"no expand" adds only the homodimers of the molecule (Figure 23a). If several proteins were queried, then all hetero-dimers in which all the proteins participate would appear.

"expand" adds all the complexes in which MCH5 is involved (Figure 23b). The green arrow with a diamond ending represents the inclusion of one protein in a complex form.



(a)

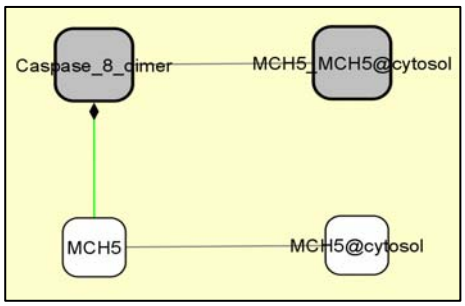


(b)

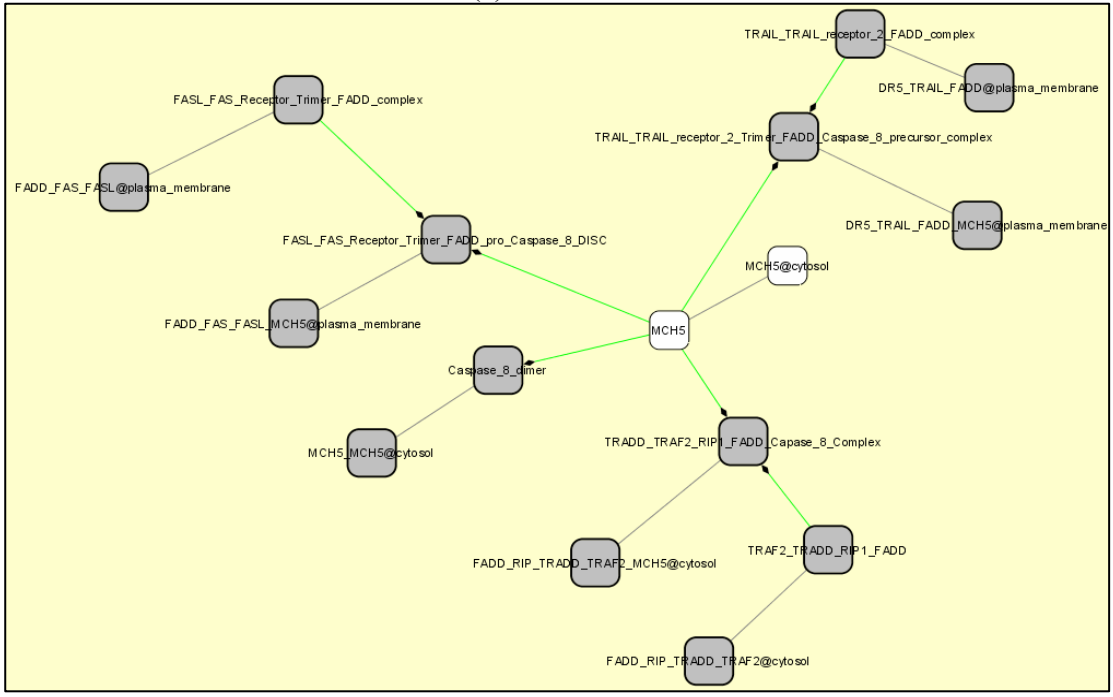
Figure 23. BioPAX Standard Query: Add complexes with (a) the « no expand" or (b) "expand" option.

- **Add chemical species**

This function adds, for each species, the cellular location and its specified modifications. It is linked to the protein with a grey edge.



(a)



(b)

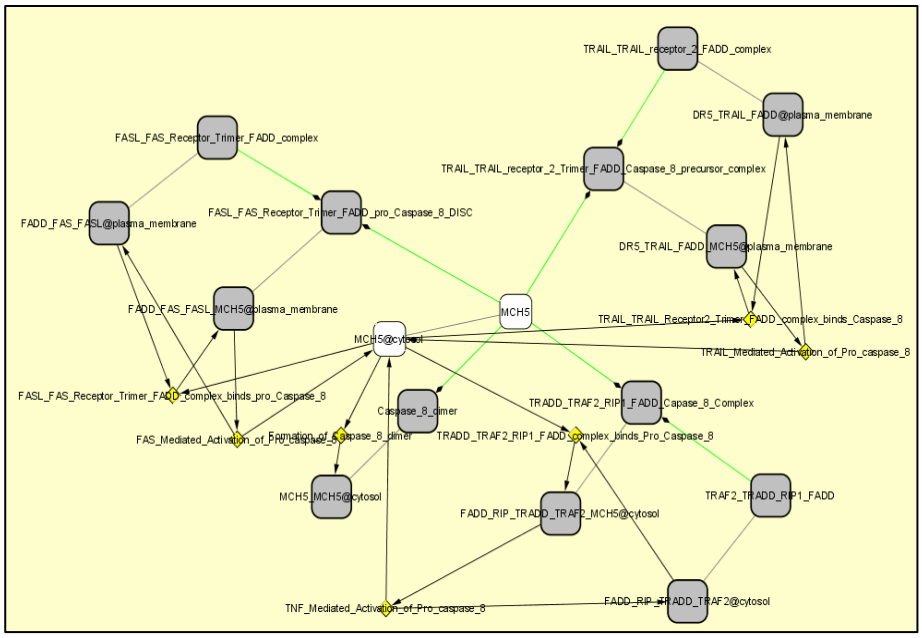
Figure 24. Chemical species. Cellular locations of all forms of MCH5.

- Add reactions

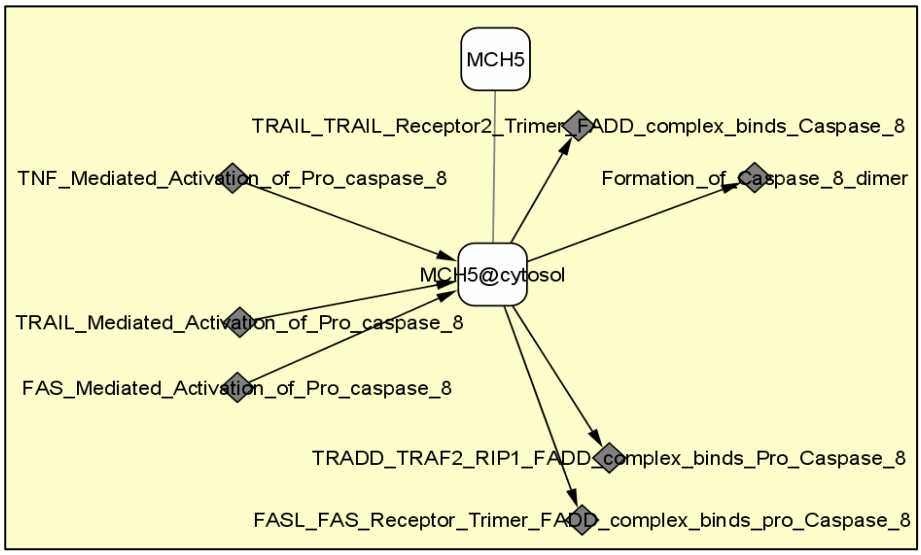
"connecting reactions" connects all present species that have common reactions (connected by a yellow node in Figure 25a).

"all reactions" includes all the reactions involving the chemical species (Figure 25b).

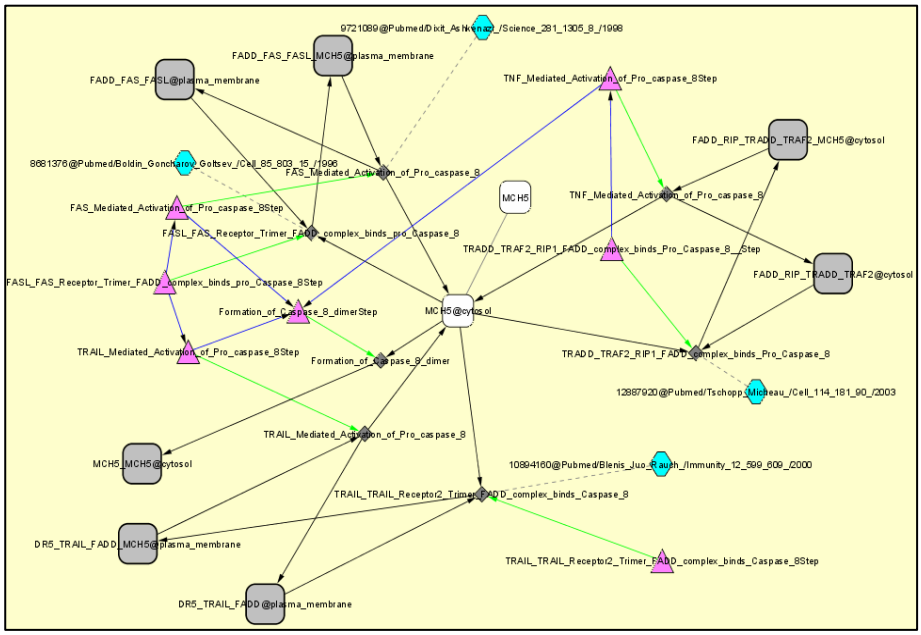
"make reactions complete" adds all the sources and targets of the reactions (Figure25c) listed in the BioPAX index, including, for example, the pathway nodes and publications links.



(a)



(b)



(c)

Figure 25. Adding reactions. Example when (a) all connecting reactions, (b) adding all reactions, and (c) making the reactions complete.

- Add publications

When available, this function adds all the references associated with a reaction.

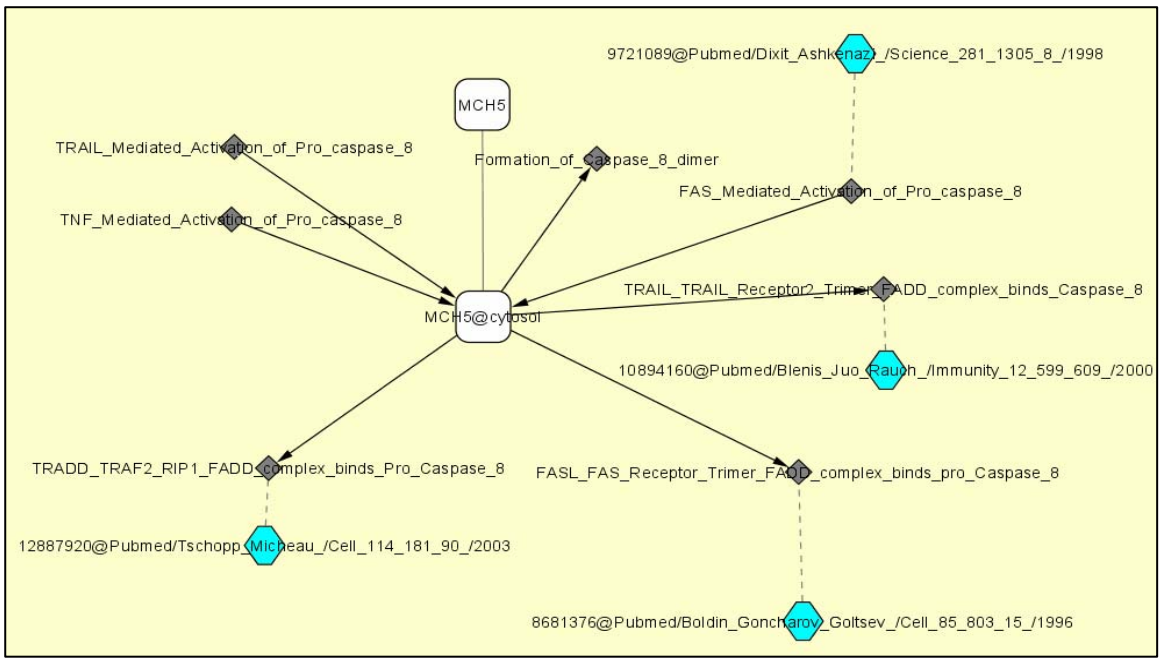


Figure 26. Adding publications

- Output

The result of the queries can be seen either in the current network or in a new network.

Index Path Analysis

Plugins => BiNoM BioPAX Query => Index path Analysis

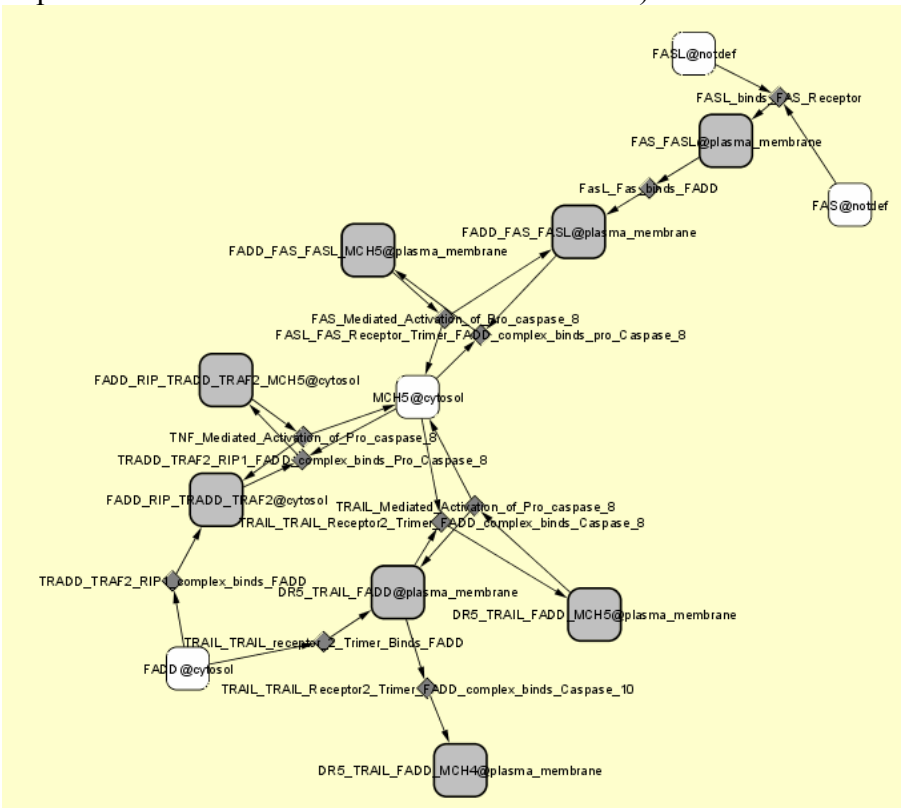
This command finds the directed or non-directed, shortest, optimal or suboptimal, non intersecting paths with a pre-defined number of intermediaries in an index file. Note that the species need to be selected on a graph before this query.

This part of the query engine uses the same algorithms and options as Path analysis dialog (see section 2.2.6), however, with the network (index) kept completely in memory, without explicit visualization. Moreover, the network is slightly modified before this type of query: in particular, all non-directed edges (of CONTAINS, SPECIESOF and some other types) are represented as bi-directional, some nodes (publications and, optionally, smallMolecules) are removed.

For example, the following steps

1) *Select Entities*: specify CASP8 and FAS proteins
2) Select the two nodes
3) *Index Path analysis*: *Find all non-intersecting paths*
4) *BiNoM Analysis: Extract Reaction Network*

produce the following network connecting CASP8 and FAS proteins (it also involves FADD because it complexes with FAS and CASP8 on the membrane):



View Query Log

Plugins => BiNoM BioPAX Query => View Query log

In this window, are recapitulated all the queries done during the session.

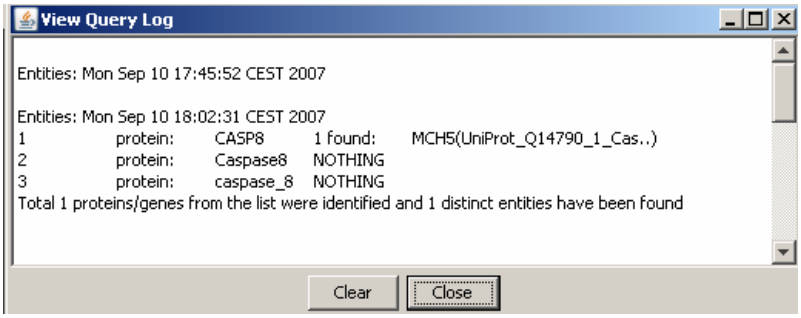


Figure 27. BioPAXViewQueryLogDialog.

2.5 BiNoM Utilities: simple functions that are missing in Cytoscape

There are various functions that facilitate the manipulation of the diagram, the copying and pasting of its subparts, the selection of some portions of it, etc.

Select edges between selected nodes

Plugins => BiNoM Utilities => Select edges between selected nodes

When some components are selected by their names (**Select => Nodes => By Name**) or simply with the mouse, the edges between the nodes are not selected. This function allows to remedy this problem.

This might be especially useful when the selection is copied and pasted in another network, although it is possible to paste the nodes with the edges connecting them, without selecting them, by choosing **File => New => Network => From selected nodes, all edges** when pasting them.

Double Network Differences

Plugins => BiNoM Utilities => Double Network Differences

A network is composed of nodes and edges. With this command, two networks *A* and *B* can be compared and the difference observed between the two is created in two new graphs: *A-B* for the differences of *A* compared to *B* (A-A∩B), and *B-A* for the differences of *B* compared to *A* (B-A∩B). Note that in the output networks, the layout of the first graph is conserved.

Update Networks

Plugins => BiNoM Utilities => Update Networks

When a network is modified, it is possible to update all the networks that are related to it, either because they are modules, sub-networks or older versions of it. That way, any changes, additions or deletions in a network can be propagated to the sub-parts that are derived from the initial version of that network. The user specifies which network is added, which one is deleted (when necessary) and which networks need to be updated in the proposed list.

Whatever is added or deleted in each sub-network is presented to the user in a separate window before any action is made. The user can agree or disagree by checking or unchecking the box in front of the network names. The previous version of the updated diagrams will not be deleted but saved under *network_name.xml_old*.

Clipboard

Plugins => BiNoM Utilities => Clipboard

The description of the commands is self-explanatory:

- Copy Selected Nodes and Edges to Clipboard

Plugins => BiNoM Utilities => Clipboard => Copy selected nodes and edges to clipboard

- Add Selected Nodes and Edges to Clipboard

Plugins => BiNoM Utilities => Clipboard => Add selected nodes and edges to clipboard

- Paste Nodes and Edges from Clipboard

Plugins => BiNoM Utilities => Clipboard => Paste nodes and edges from clipboard

- Show Clipboard Contents

Plugins => BiNoM Utilities => Clipboard => Show clipboard contents

3 Test cases – Tutorials

3.1 Disentangling retinoblastoma protein network

Based on the literature, we built a detailed network of the RB/E2F pathway. Details of the comprehensive map can be found at this address: http://bioinfo-out.curie.fr/projects/rbpathway/

The associated Cytoscape session for this example (*RB_manual.cys*) is available on BiNoM website: http://bioinfo-out.curie.fr/projects/binom/

We propose here to describe the steps leading to the semi-automatic decomposition of the comprehensive network (Chen et al., 2004). This is one way of getting modules using BiNoM but we do not pretend to present a strict method.

1. Import the model in Cytoscape

Plugins => BiNoM I/O => Import CellDesigner Document from file

Choose the file: *RBPathway.xml*

2. Prune the graph

To facilitate the study and the modular decomposition, we propose to work on the central cyclic part of the network. For that, we need to isolate it from the rest of the diagram:

Plugins => BiNoM Analysis => Prune the graph

The network is then divided into three graphs: the input (*RBPathway.xml_in*: proteins that are already present), the output (*RBPathway.xml_out*: proteins that are degraded or complexes that are not further modified), and a central cyclic part (*RBPathway.xml_scc*). We acknowledge that the model is not complete and that the input and output graphs would be minimized if the map were more detailed.

3. Get Material Components, or Get Cycle Decomposition

Depending on the studied diagram, it will be easier to decompose the graph either into components or into cycles. We show here how the material component decomposition isolates the different modules from the strongly connected component graph (*RBPathway.xml_scc*):

Plugins => BiNoM Analysis => Get Material Components

This function isolates subgraphs in which each component is involved. It provides some kind of life cycle of an individual protein, or more specifically, its modifications along the cycle.

29 subgraphs are obtained. Before analyzing the subgraphs, we propose to cluster the graphs with respect to a high percentage of similarities. For that, we select the 29 subgraphs and set the percentage to 30% of similarities among diagrams.

Plugins => BiNoM Analysis => Cluster networks

The 29 subgraphs are clustered into 7 sub-networks that we rename according to the elements composing the sub-networks: *CYCD1_CDK4_p27Kip_p21Cip*, *CYCD1_CDK6_p27Kip_p21Cip*, *p53_MDM2*, *CYCE1_CDK2_CYCA2*, *CDC2_CYCB1_WEE1_CDC25C*, *CDH1*, and one last one that we decompose further into three sub-networks: *E2F1_DP1*, *E2F4_DP2_p130_p107*, *RB* using, this time, the cycle decomposition (15 cycles obtained that we group into 4 clusters, 2 of them merged into one module).

4. Re-introduce input and ouput

Now that the core of the modules is given, we propose to re-introduce the components from the input and output files that were ignored so far. **Plugins => Merge networks**

Each module is merged with the two IN and OUT files and only the connected parts are kept. The rest is deleted. However, merging these IN and OUT parts back into each module requires manual curation: since the network is highly connected, a lot of components that are not disjoint will have to be selected carefully and deleted. For this purpose, the reorganization of the network with the organic layout becomes very useful as the highly connected components are often gathered as clusters.

5. Compare the union of the modules with the initial file

All the obtained modules are merged into one big file and compared to the initial diagram *RBPathway.xml***.** The comparison shows the elements that were not included in the modules and need to be replaced in the appropriate modules. This comparison checks at this point the omitted parts and isolates the modules that remain to be created. For example, in our example, three modules that were not in the cyclic part could be formed from what remained: *E2F6_DP1*, *CDK3_CYCC*, *CDK7_CYCH*.

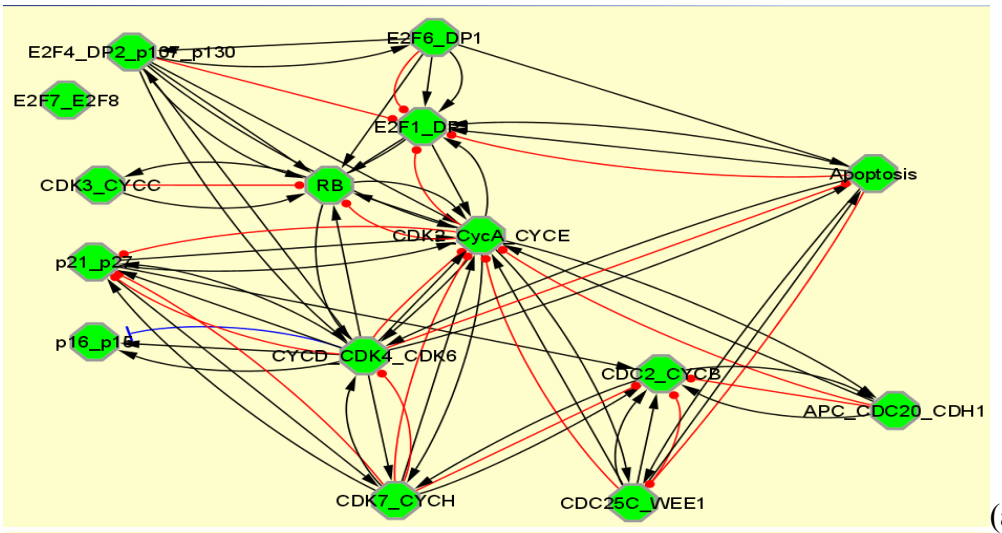
6. Refine the networks

The modules are separated or merged according to the choices made by the modeller. We proceed to a more refined decomposition of the existing individual modules using similar methods, either cycle or component decomposition. New modules were chosen. *CYCB1_CDC2_WEE1_CDC25C* was divided into two modules: *CYCB1_CDC2* and *WEE1_CDC25C* and *CYCD1_CDK6_p27Kip_p21Cip* and *CYCD1_CDK6_p27Kip_p21Cip* were merged and divided into three modules: *CYCD1_CDK4_CDK6*, *p16_p15* and *p27Kip_p21Cip*.

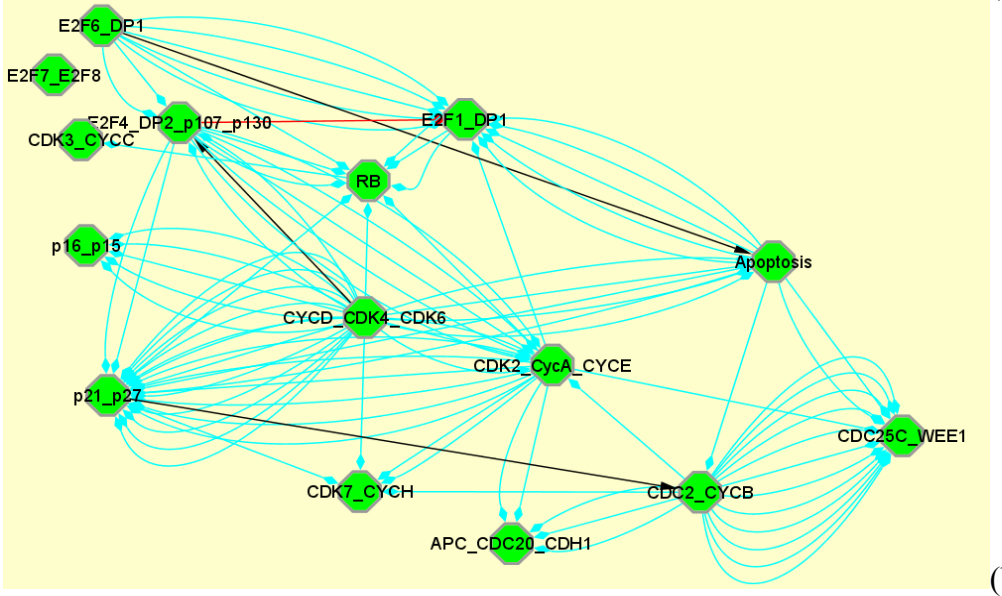
Method: To select specific nodes, for example WEE1 and CDC25C in *CYCB1_CDC2_WEE1_CDC25C*, we chose: **Select => Nodes => By Name** and in the dialog window, we wrote first *WEE1* (the * selects all the species in which WEE1 appears) then *CDC25C*. To select the reactions associated with the proteins: **Select => Nodes => First neighbors of selected nodes** and finally using BiNoM features: **Plugins => BiNoM Utilities => Select Edges between selected nodes**. The next step is simply the creation of a new network from the selected nodes and edges: **File => New => Network => From selected nodes, all edges**. Finally, the resulting network is renamed.

7. Create a modular view of the network.

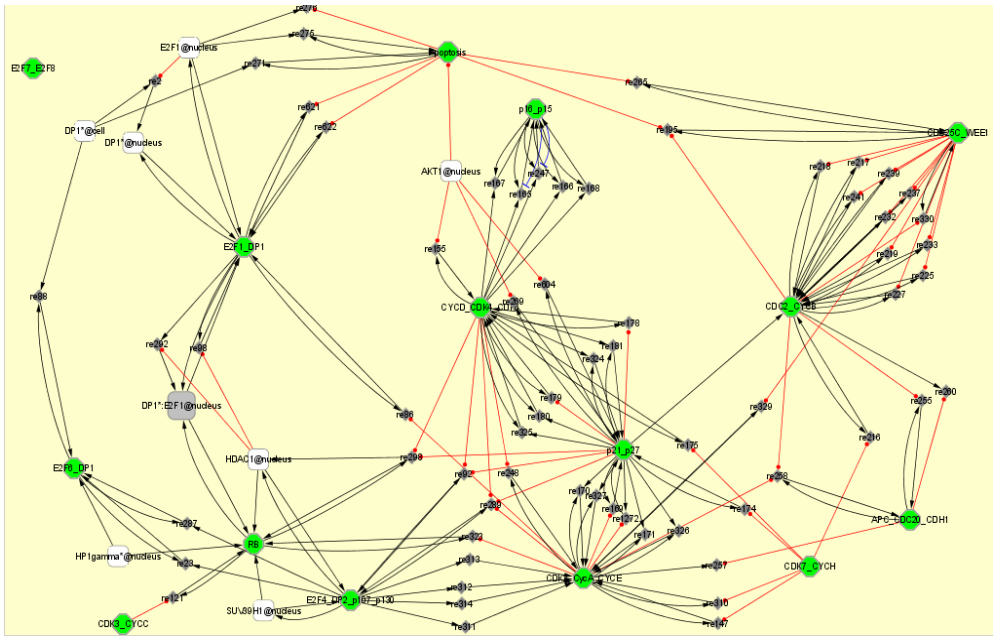
From the modules obtained with the described method, we can automatically generate the modular view of the initial network. We choose the BiNoM CellDesigner visual style and the resulting network can exhibit three different types of information (see section 2.2):



(a)



(b)



(c)

Figure 28. Modular views of RB/E2F network showing (a) simple molecular connections existing between modules, (b) modules that share molecules or reactions and (c) explicit connections between modules.

Note that the modules obtained here might slightly differ from the ones presented on RB/E2F website. If the resulting diagrams are very similar, the final outcome of BiNoM decomposition may not be unique since the refinement of the modules is left to the modeler.

3.2 Feedbacks in EGFR pathway diagram

Coming soon…

3.3 Bow-tie and the Toll-like receptor pathway

Coming soon…

4 Simple scenarios

4.1 Converting CellDesigner to BioPAX

1. **Import a CellDesigner document in Cytoscape:**
   Use the command: **Plugins => BiNoM I/O => Import CellDesigner Document from file.**
2. **Export the network as a BioPAX file:**
   Use the command: **Plugins => BiNoM I/O => Export current network to BioPAX.**
3. **Save the file:**
   Name the document to save. It will be saved as NAME.owl.
4. **Import the BioPAX file in Cytoscape:**
   The network will be available for import with all the features associated to BioPAX functions: BioPAX Class Tree and BioPAX Property Editor.

4.2 Analyzing BioPAX/CellDesigner network and extracting its part

1. **Select the BioPAX (or CellDesigner) network**:
   Select a subpart of the graph (Reaction, Pathway or Protein networks for BioPAX case) with the mouse or through successive queries by name (**Select => Nodes => By Name**).
2. **Save the selection**:
   The selected part needs to be isolated in a different network. For that, save the selection in the clipboard:
   Plugins => BiNoM Utilities => Clipboard => Copy Selected Nodes and Edges to Clipboard
   This step can be bypassed by creating directly the network
3. **Create the new graph**:
   File => New => Network => From selected nodes, selected edges.
4. **Export the graph in BioPAX format (or CellDesigner)**
   File => BiNoM I/O => Export current network to BioPAX (CellDesigner)
   A dialog window warns you that the file must be associated with an existing BioPAX (or CellDesigner) document. The sub-part is associated with the initial file and saved with a different name.
5. **Merge of two sub-networks**:
   Two subparts can be merged. The same process can be repeated for another sub-network. The sub-network is exported to BioPAX (CellDesigner) and associated with the initial file. However, when saving, you need to save the second sub- network with the same name as the first sub-network with which you want to merge this one. A dialog window will ask you if you want to overwrite or merge the networks.

We propose to show with a specific example the relationship between the two Cytoscape and CellDesigner softwares and how the model can be converted from one format to the other. We show how the changes made in one of the softwares can be propagated to the other.

Modifying the Cytoscape file
If the model is modified in Cytoscape or if the user wants to isolate a subpart of the network for a specific analysis, it is possible to view in CellDesigner the changes of the model made in Cytoscape. However, the export from Cytoscape to CellDesigner requires the association of an existing CellDesigner file to the Cytoscape model.

To illustrate this case, we propose a scenario describing the selection of a subpart of the network in Cytoscape that will be visualized in CellDesigner.

Let us assume that we want to isolate a sub-network of the interactions between CDC2/CDC13 and Slp1 in our M-Phase example in Cytoscape and that we want to view the resulting network in CellDesigner. We will proceed as follows:

- Select the graph in Cytoscape: **Select => Nodes => By Name**
In *M-Phase.xml* network, the part of the graph that we want to isolate is selected with the mouse or through successive protein selections through the menu.
- Save the selection: **Plugins => BiNoM Utilities => Clipboard => Copy Selected Nodes and Edges to Clipboard**
- Create the new graph: **File => New => Network => From selected nodes, selected edges**. A new graph is created as a "child" of the "parent" network (Figure 29).

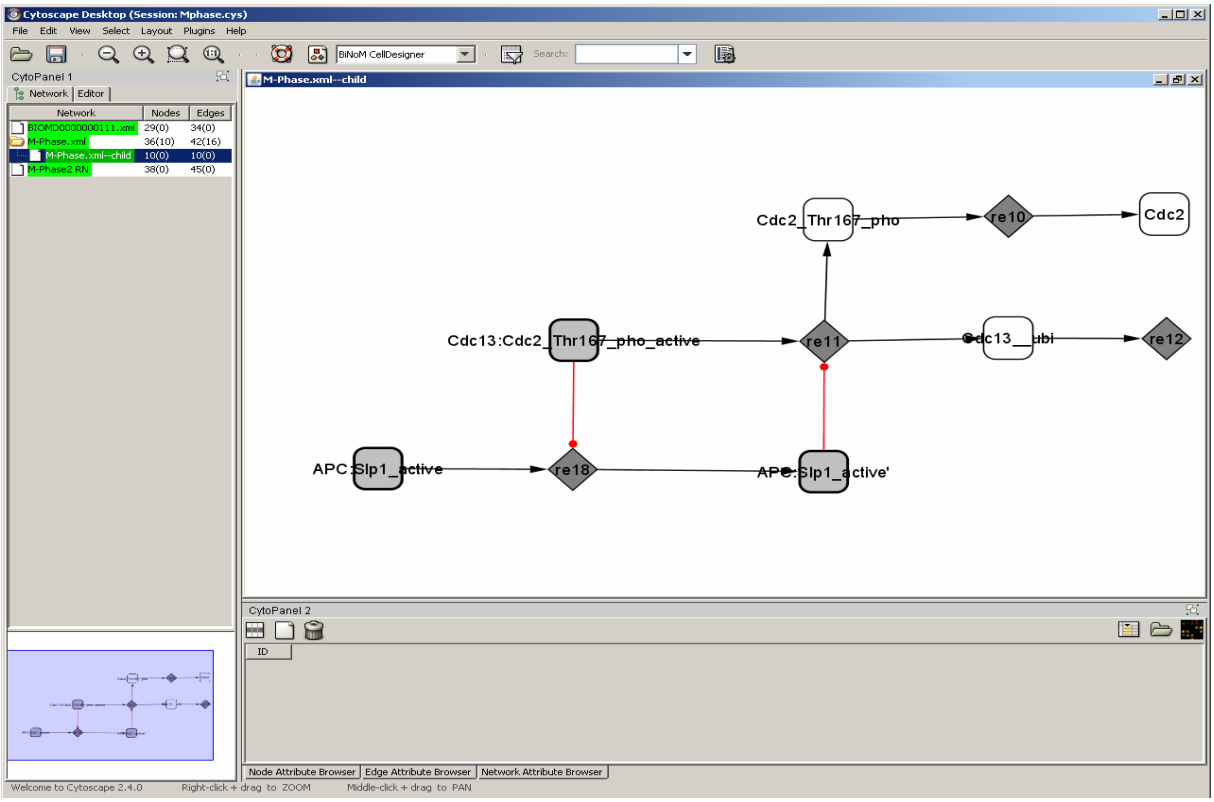


Figure 29. Cytoscape view of the sub-network involving Cdc13 and Slp1 interactions.

- View the subgraph in CellDesigner: **Plugins => BiNoM I/O => Export current network to CellDesigner**.
A window appears, warning the user that the network has to be associated to an existing CellDesigner document. One file from the list must be chosen (here, the initial CellDesigner file, *M-Phase.xml*) and then the sub-network needs to be saved (Figure 30).

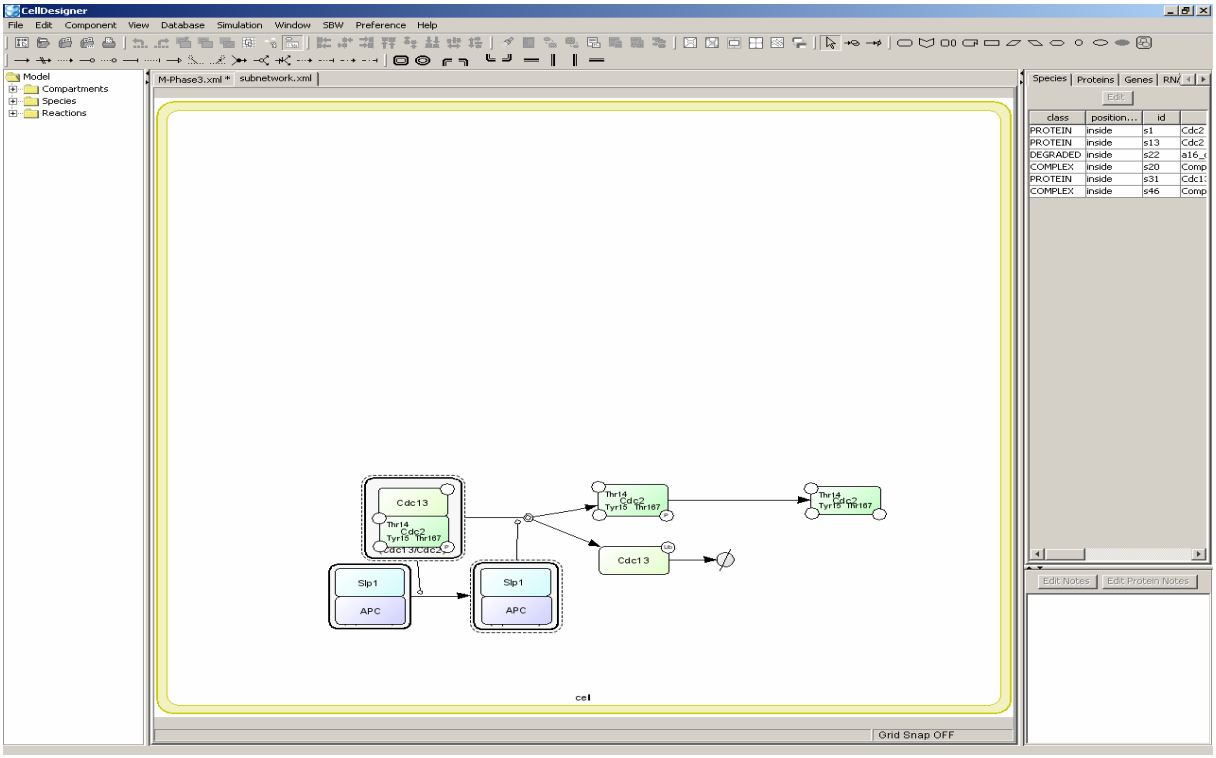


Figure 30. CellDesigner view of the sub-network involving Cdc13 and Slp1 interactions.

Note that several subgraphs can be isolated with this method and merged later in the same CellDesigner document. When exporting the Cytoscape network to CellDesigner, a pop-up window asks the user whether or not the graph needs to be merged with or over-write the existing file.

4.3 Accessing and modifying BioPAX content and saving it

1. **Import a BioPAX document**:
   Select from the menu, **Plugins => BiNoM I/O => Import a BioPAX Document from file.**
2. **Open the appropriate networks**:
   Check the boxes accordingly (Reaction, Pathway, or Protein).
3. **Modify the layout for better viewing**:
   For example, for an organic layout, choose: **Layout => yFiles => Organic**
4. **Access BioPAX content**:
   Open the Class Tree corresponding to the network:
   Plugins => BiNoM I/O => BioPAX Class Tree.
   A window opens (Figure 16). To get information on a specific protein, in the tree, unfold Thing, Entity, physicalEntity, protein. The list of proteins appears on the right frame.
   Click on one protein. The corresponding BioPAX Property Editor opens with all the available information: Availability, Comment, Data_dash_source, Name, Organism, Sequence, Short_dash_name, Synonyms, Xref.
5. **Modify BioPAX content**:
   In the BioPAX Property Editor, click on the *Edit* tab. Some options in front of each field appear: Remove, Update, or Add.
   In the BioPAX Class Tree, you can also add an entity: protein, interaction, protein complexes, etc. by clicking on the *New Instance* tab at the top of the window.

6. **Save BioPAX content**:
   To save all the changes made in the BioPAX Property Editor or in the BioPAX Class Tree, choose **Plugins => BiNoM I/O => Save whole associated BioPAX as.**

4.4 Creating a file from scratch

(Not related to BiNoM features but useful to know)

1. **Open a new empty network:**
   Create a new empty document: **File => New => Network => Empty network**.
   We suggest you already add a node on the empty network in order to immediately determine the preferences before constructing the network.
2. **Set up the appropriate preferences:**
   Make sure that you set up the preferences to fit the properties you want your graph to have. For example, you need to specify the type of names you want to appear on the screen for each node.
   View => Open VizMapper
   Choose the Visual Style you want to use (For example, BiNoM BioPAX).
   Define the preferences: click on the *Define* tab on the right of the visual style. Choose the node and edge attributes.
   We recommend the selection of canonicalName (Map Attribute) for the Node label.
   Then click on *Apply to Network* to validate your choices. You can save the node and edge attributes into a file (NAME.props) by selecting **File => Export => Vizmap Property File**.
   As a consequence, in CytoPanel 2, make sure to display *canonicalName* (in the top left icon) corresponding to the node ID that you will have to modify (by clicking on it) and give a name to your molecule every time you add a new one.
   You can import the node attributes directly if you already determine the preferences in another network. For more information, read Cytoscape manual at the following address: http://www.cytoscape.org/manual/Cytoscape2_5Manual.html
3. **Create a network:**
   In CytoPanel1, click on the *Editor* tab. Two possibilities appear: *Directed Edge* or *Add a Node*.
   To add a node, click on *Add a Node*, maintain the hold on your mouse and release it where you want to position your node on the graph. The ID Node0 is attributed. To change it, modify *canonicalName* in CytoPanel2 (as mentioned above). The new name appears on the graph. In CytoPanel2, you can also choose to display NODE_TYPE and specify the type of nodes you added: protein, complex, reaction, etc.

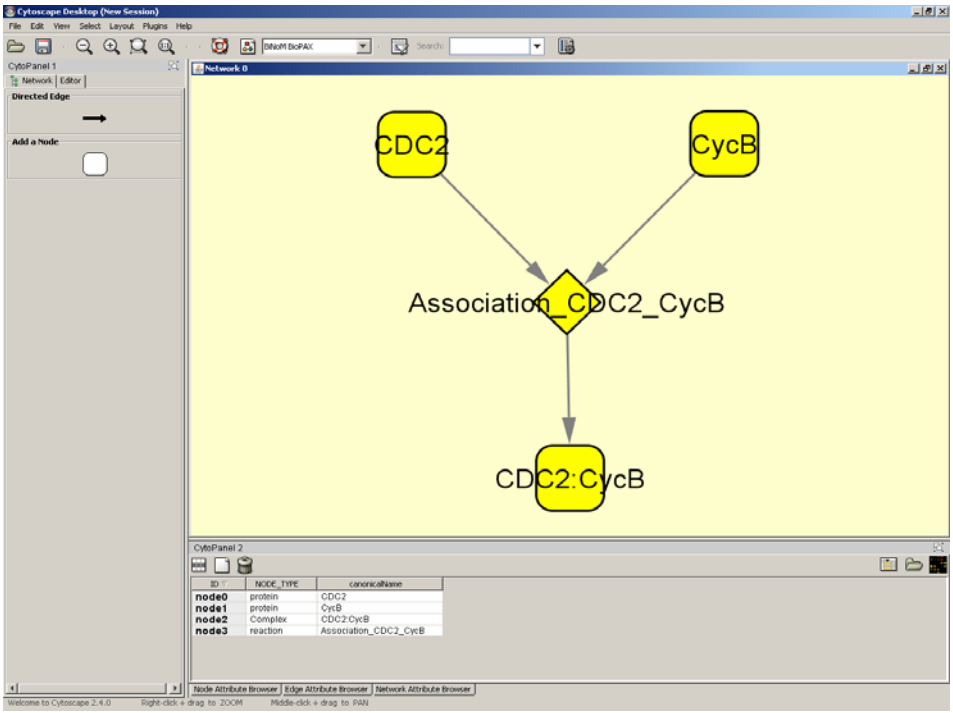


Figure 31. Create a network from scratch

When you right-click on a node, you get several possibilities:

- Visual Mapping Bypass: allows the modification of the node visualization. For example, we modified in Figure 31, the shape of "Association_CDC2_CycB" to represent the *reaction* attribute. Similarly (not shown here), we selected the grey filling for CDC2:CycB to represent the complex (**Visual Mapping Bypass => Fill**).
- LinkOut: sends you to different types of information available online: Entrez, Pubmed, IHOP, Reactome, etc.
- Delete selected nodes and edges.

4.5 Modularization of a network

5 BiNoM graph representation of data

5.1 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.

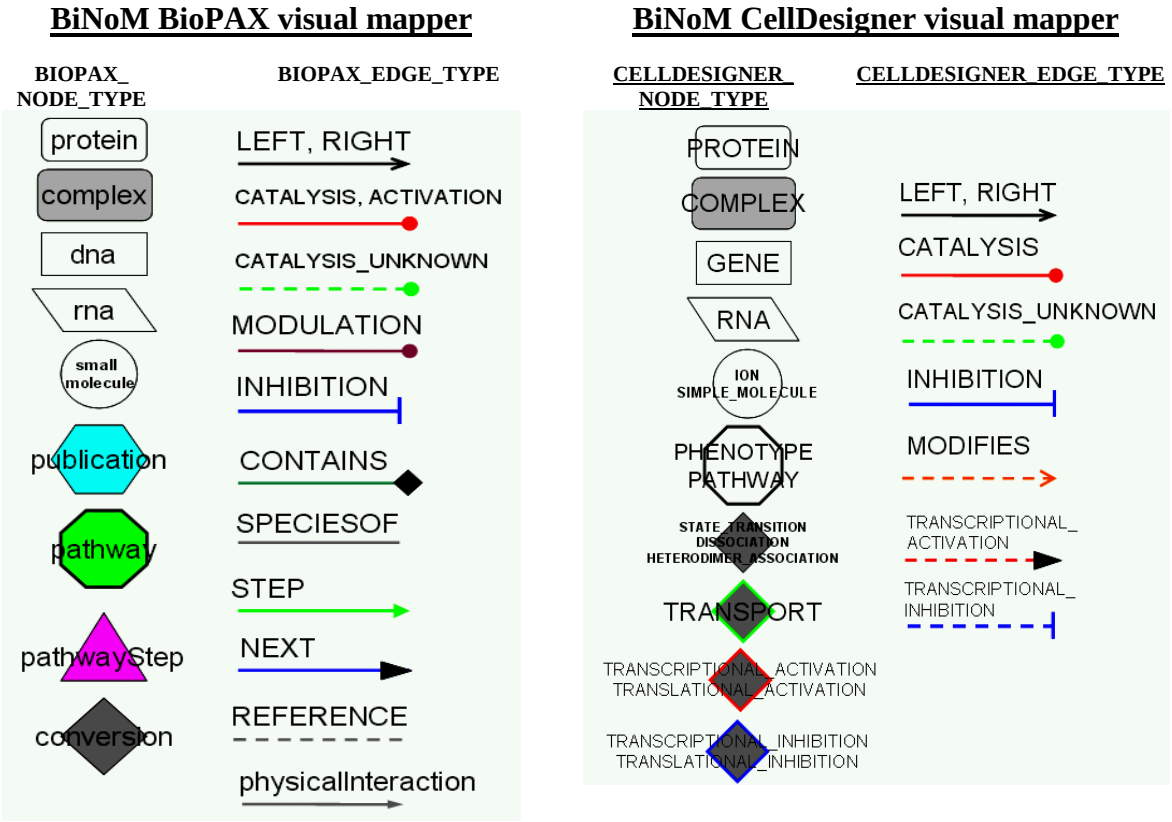
Currently, BiNoM index is mostly developed to map BioPAX ontology files and CellDesigner object schema. In future versions, other mappings will be available, for instance, a mapping to SBML files annotated with Systems Biology Ontology (http://www.ebi.ac.uk/sbo/).

In the table below all attributes used by the index are listed:

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		

5.2 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. Below, the legend for deciphering the different types of visualization is provided:



5.3 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 (shown in Systems Biology Graphical Notation standard) are presented in the next figure:

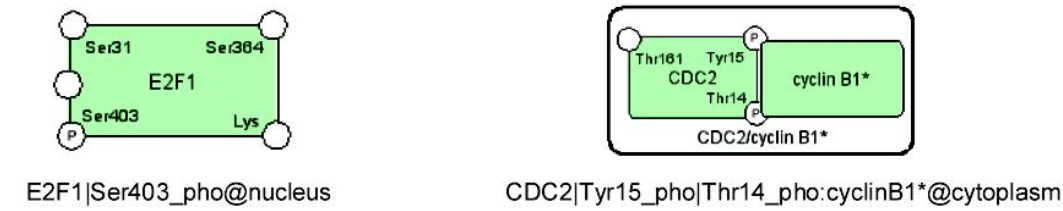


Figure 32. Names of chemical species represented in SBGN standard.

An example of a conversion of a little network from CellDesigner Graphical Notation to BiNoM index representation is given below:

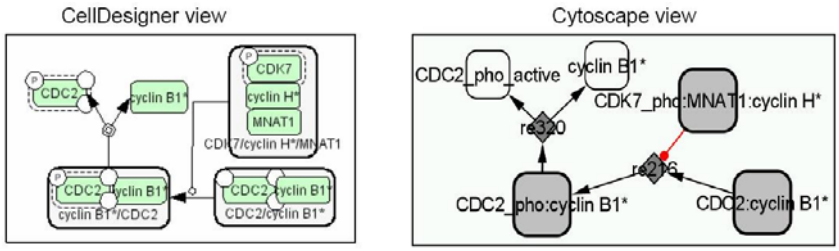
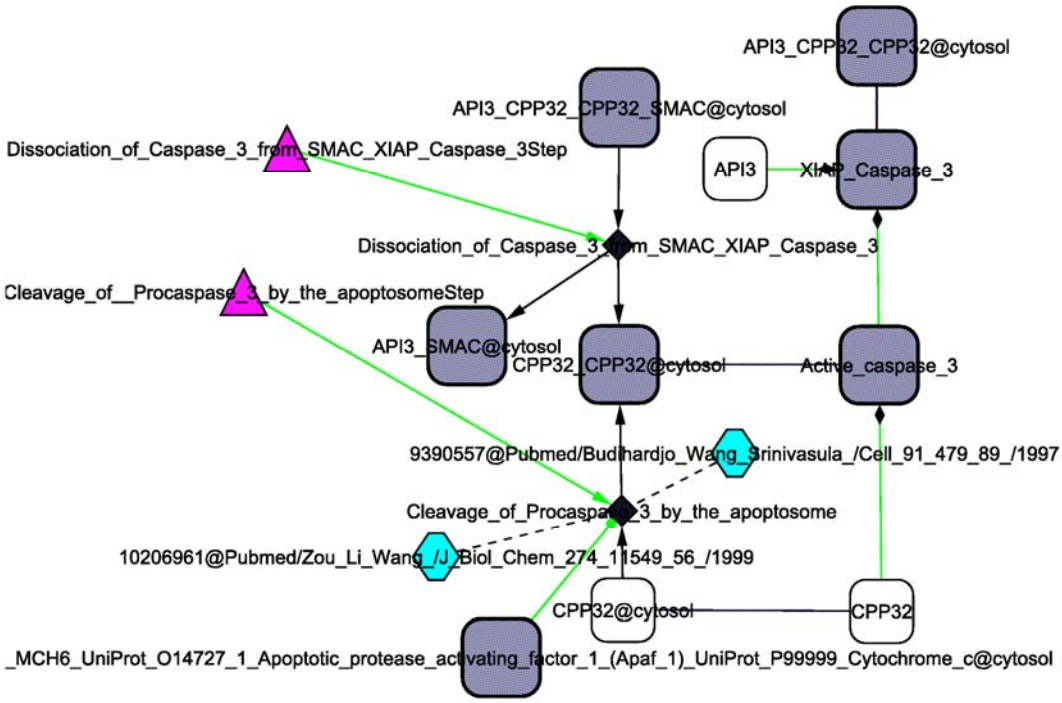


Figure 33. From CellDesigner to Cytoscape.

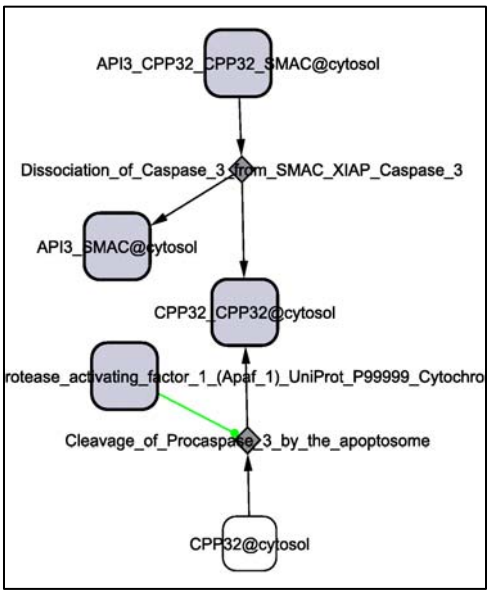
A small fragment of BioPAX index generated for Apoptosis pathway and extracted from Reactome database is presented below (see section 2.3 for details):



5.4 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 Reaction Network, Pathway Structure and Protein-Protein Interaction interfaces.

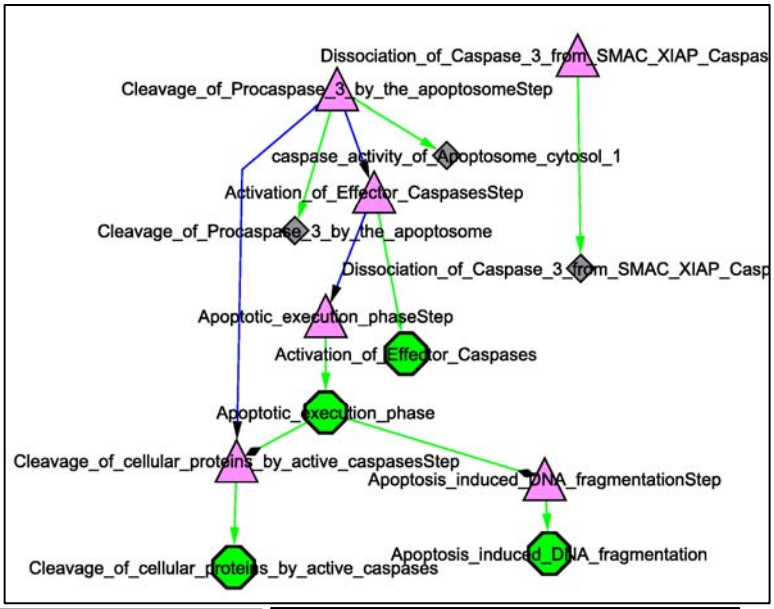


The Reaction Network interface is a bipartite graph which contains nodes of only two types: 'species' and 'reactions'. Reactants are connected to reactions through edges of type LEFT, products are connected through edges of type RIGHT. Modifier species are connected through CATALYSIS, MODULATION and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

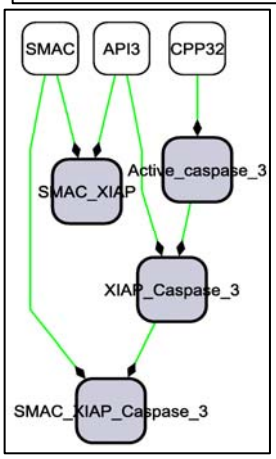
Some BioPAX objects (catalysis, for example) are represented by edges with the corresponding BIOPAX_URI attribute.

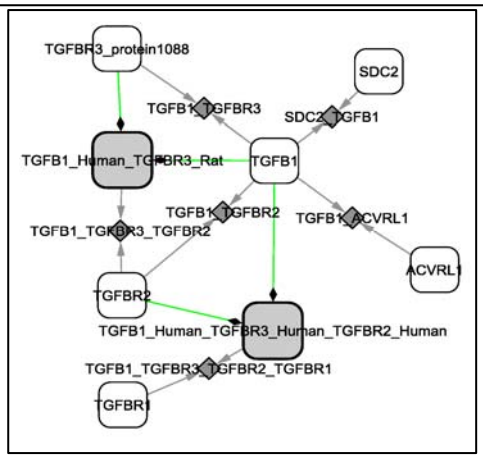
A chemical species node can correspond to several grouped physicalEntityParticipants, thus, it can have several BIOPAX_URI attributes. When calling BioPAX editor, all of them will be opened.

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



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





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 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.

6 Things to know and little bugs to avoid

We present here a list of things to know gathering some troubles we ran into when manipulating network using Cytoscape and BiNoM:

- There is an error message when saving a Cytoscape session: if two networks have the same name in the CytoPanel1, the session will not be able to be saved. Make sure you rename the networks that share the same name.

- Set operations are done on edges and nodes

APPENDICES

Installation and Distribution

BiNoM Cytoscape Plugin is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

BiNoM Cytoscape plugin is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details: http://www.gnu.org/licenses/lgpl.html .

BiNoM plugin together with its source code can be freely downloaded from the web-site http://bioinfo-out.curie.fr/projects/binom .

BiNoM can be installed and used as Cytoscape plug-in or a stand-alone library for performing manipulations on networks without Cytoscape, since its logic implementation is completely decoupled from Cytoscape libraries (see the web-site for details).

ALGORITHMS

For the description of the algorithms, see the references below. We will provide soon the detailed description of the simple modifications we made for the standard graph theory algorithms.

GLOSSARY

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.

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.

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.

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 (see section 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.

Strongly Connected Components (SCC):
A subgraph in a directed graph, in which there is path from any node to any node

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

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

References

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the Web Comput Networks *33*, 309-320.

Chen, K. C., Calzone, L., Csikasz-Nagy, A., Cross, F. R., Novak, B., and Tyson, J. J. (2004). Integrative analysis of cell cycle control in budding yeast. Mol Biol Cell *15*, 3841-3862.

Gleiss, P., Stadler, P., Wagner, A., and Fell, D. A. (2001). Relevant Cycles in Chemical Reaction Network. Adv Complex Systems *4*, 207-226.

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

Novak, B., Csikasz-Nagy, A., Gyorffy, B., Nasmyth, K., and Tyson, J. J. (1998). Model scenarios for evolution of the eukaryotic cell cycle. Philos Trans R Soc Lond B Biol Sci *353*, 2063-2076.

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

Stromback, L., and Lambrix, P. (2005). Representations of molecular pathways: an evaluation of SBML, PSI MI and BioPAX. Bioinformatics *21*, 4401-4407.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. SIAM Journal on Computing *1*, 146-160.

Vismara, P. (1997). Union of all the minimum cycle bases of a graph. ElectrJComb *4*, 73-87.