

OCSANA: Optimal Combinations of Interventions from Network Analysis

Paola Vera-Licona, Eric Bonnet, Emmanuel Barillot and Andrei Zinovyev

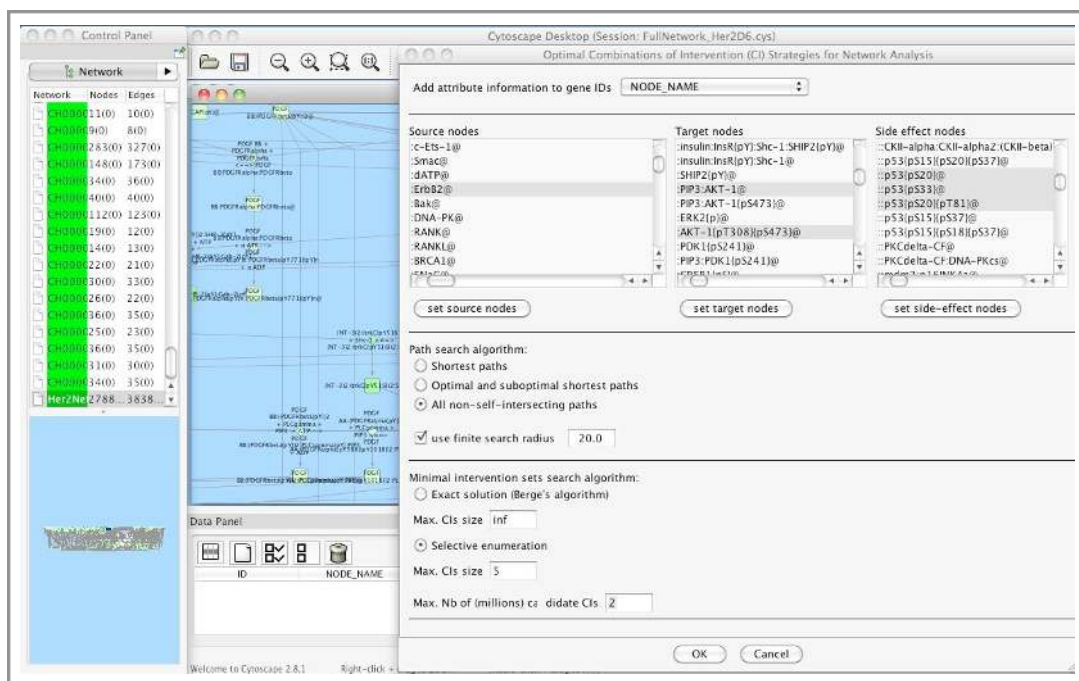


Table of Contents

Introduction	2
Download and Installation	3
Running OCSANA	4
A Walkthrough Example	7
Bibliography	15

Introduction

OCSANA (Optimal Combinations of Interventions from Network Analysis) is a software designed to identify and prioritize optimal minimal combinations of interventions (CIs) to disrupt the paths from source nodes to target nodes. When specified by the user, OCSANA seeks to additionally minimize the side-effects that CIs can cause on specified off-target nodes. To ensure the method's scalability, OCSANA includes an exact solution via an adaptation of Berge's algorithm [Berge, 1989] and [Haus *et al.*, 2008] and a novel selective enumeration approach based on a weighted-greedy algorithm. The exact solution computes all minimal CIs of all sizes and it is also adapted to compute all CIs up to a specified size. The selective enumeration approach computes optimal minimal CIs up to a specified size and it can be parametrized to identify by full enumeration all CIs up to a specified size.

OCSANA is implemented as a plugin to the open source network analysis and visualization software, Cytoscape [Shannon *et al.*, 2003]. It uses the Java library BiNoM [Zinovyev *et al.*, 2008], to facilitate the import, assembling and analysis of signaling networks.

We refer the user to the algorithm description included in the supplementary materials of OCSANA's paper, to retrieve the definitions of the different concepts used and for a full description of the algorithms and OCSANA's scoring.

Availability: Last version of OCSANA distributed under LGPL license together with tutorial and source code, are available at:

<http://bioinfo.curie.fr/projects/ocsana>.

Download and Installation

OCSANA is a Cytoscape plugin. Thus to install and run it, Cytoscape should be firstly installed. The user can download and install Cytoscape from: <http://www.cytoscape.org/>

Since **OCSANA** uses BiNoM Java library, we have integrated it within the plugin BiNoM (which can be installed from <http://binom.curie.fr/>). Using the version of OCSANA within BiNoM, will allow users to simultaneously take advantage of the different tools and capabilities for network assembling and analysis within BiNoM. If you have already installed the latest version of BiNoM, after opening a Cytoscape session, OCSANA is included in the menu:

Plugins => BiNoM 2._ => BiNoM Analysis => OCSANA analysis.

Otherwise, OCSANA can be also downloaded in its standalone version from here¹:

Once the OCSANA.jar file is downloaded, it has to be placed into the Plugins Folder contained within the Cytoscape folder. Start the Cytoscape session and run OCSANA from the menu:

Plugins => OCSANA analysis.

¹ This installation procedure is valid for the current official 2.8._ version of Cytoscape and it might change in the future 3._ version. In any given instance, BiNoM plugin and its updated versions, containing at all times OCSANA, will be available through the Plugin Manager of Cytoscape.

Running OCSANA

Once a network is launched in Cytoscape and is ready to be analyzed:

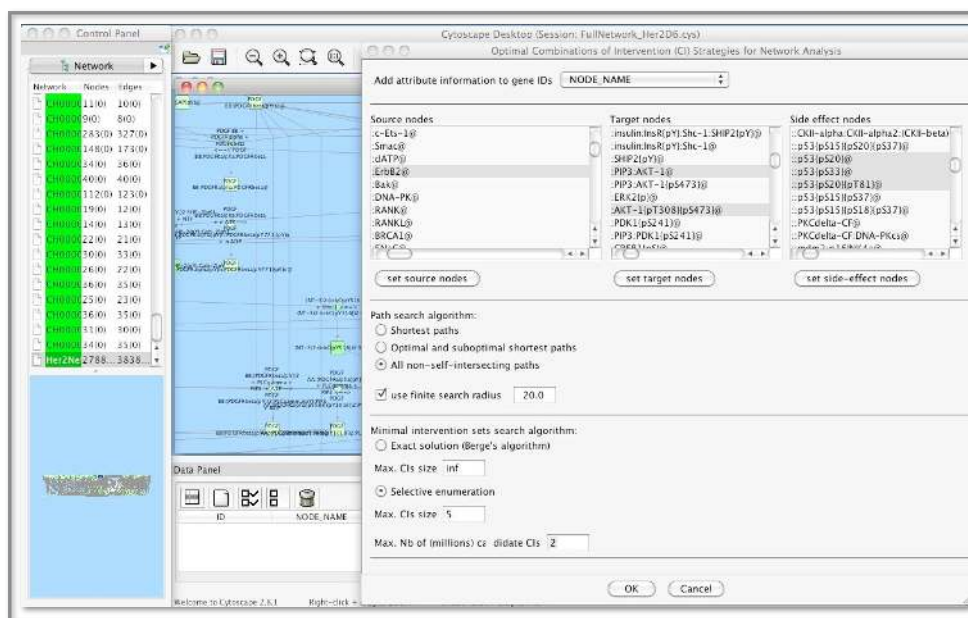
If BiNom has been installed, click on :

Plugins => BiNoM 2.2 => BiNoM Analysis => OCSANA analysis.

If the standalone version of OCSANA has been installed in Cytoscape, then click on:

Plugins => OCSANA Analysis.

Here below is a screenshot of the dialog window that appears in both cases:



Depending on the size of the screen of the computer in use and/or the platform in which the user is running OCSANA, it can occur that the interface window does not have the correct size to show fully all its written features. The user can resize this interface window manually by simple clicking on the low right corner of the window and dragging the computer mouse until the interface window has the desired size.

Now we present a summary of the different parameters to fill in the interface window:

Add Attributes to gene ID allows the user to select the node attribute to make nodes' query (None, Canonical Name, Count, Biopax_Node_Type, Node_Name, Biopax_Reaction, Biopax_Edge_Type, Interaction).

Input Nodes:

The **Source nodes** section allows the user to specify the source nodes (*i.e.* the nodes from which the signals' cascades will start). The nodes can be selected manually from the list of nodes appearing in this section or by copying and pasting the list of desired nodes in the dialog window after clicking on the **set source nodes** button.

The **Target nodes** section allows the user to select the target nodes (*i.e.* the nodes intended to be blocked). The nodes can be selected manually from the list of nodes appearing in this section or by copying and pasting a list of nodes in the dialog window after clicking on the **set of target nodes** button.

The **Side-effect nodes** section is an optional column for the user that allows to select off-target nodes, that is, nodes that are preferred to be avoided when constructing CIs. A CI that contains a node from such given list, will then receive a penalty. The nodes can be selected manually from the list of nodes appearing in this column or by copying and pasting the list of nodes in the dialog window after clicking on the **set side-effects nodes** button.

Specifying Signs of Edges:

The algorithm in OCSANA requires the type of interactions in the network (in the interaction attribute of edges as **activation or inhibition**). When the user does not specify the sign of a given edge in the network, OCSANA will automatically convert such unspecified sign to **activation**.

Specifying the Path Search Algorithm:

Because for large networks, the number of paths can be exponential, the computation of elementary paths and/or minimal hitting sets can be computationally prohibiting. To ameliorate this problem, we integrated from the Java library BiNoM [Zinovyev et al., 2008], three alternative path analyses: Shortest paths, Optimal and suboptimal shortest paths and, All the non-self-intersecting paths (which do not include inner loops).

Once the source, target and side-effects nodes have been specified, the user should select the type of path search algorithm to be used:

Shortest Paths searches for the paths with the shortest length among all the elementary paths connecting source nodes to target nodes and source nodes to side-effect nodes (via Dijkstra's algorithm).

Optimal and suboptimal shortest paths. A suboptimal path is constructed by removing all edges of all shortest paths one by one, and finding the new shortest path.

All non self-intersecting paths considers all the paths, between specified nodes, that do not contain loops (self-intersections).

Because the amount of non self-intersecting paths can as well increase exponentially for large networks, OCSANA allows to specify the maximum length of the non-self-intersecting paths to be found:

Use finite search radius, if selected, allows the user to specify an upper bound for the length of the non self-intersecting paths between nodes.

Specifying characteristics of optimal CIs sets:

Optimal CIs search algorithm: depending on the size of the network and the amount of nodes selected, the user has the option to select between Exact solution (Berge's algorithm) or selective enumeration approaches, to compute and prioritize CIs.

Exact Solution (Berge's algorithm) is based on the algorithm proposed in [Haus *et al.*, 2008], a special encoding of Berge's algorithm [Berge, 1989]. The Exact Solution computes all minimal CIs of all sizes and it is also adapted to compute all CIs up to a specified size .

The CIs are presented as a prioritized list according to their OCSANA scores. This approach is in general very efficient for small to medium size networks (approx. less than 300 nodes).

Selective Enumeration refers to the weighted-greedy algorithm to find the most optimal CIs up to a specified size. The selective enumeration approach can be as well parametrized to identify, by full enumeration, all minimal CIs of all sizes or up to a specified size.

Max. set size is a parameter used in both Exact Solution and Selective Enumeration, to specify the maximum size of optimal CIs to be identified. The default value for this parameter is 10.

Max Nb of (million) CIs is a parameter used when the selective enumeration algorithm is selected. OCSANA searches in tandem CIs of size 1, 2, 3, and so on; as the size of CIs increases, there is a combinatorial explosion of candidate sets to be considered. Thus **Max Nb of (million) hit sets** parameter allows the user to specify an upper bound for the number of candidate CIs to be tested at each given combination set size. The default value for this parameter is 50.

A Walkthrough Example

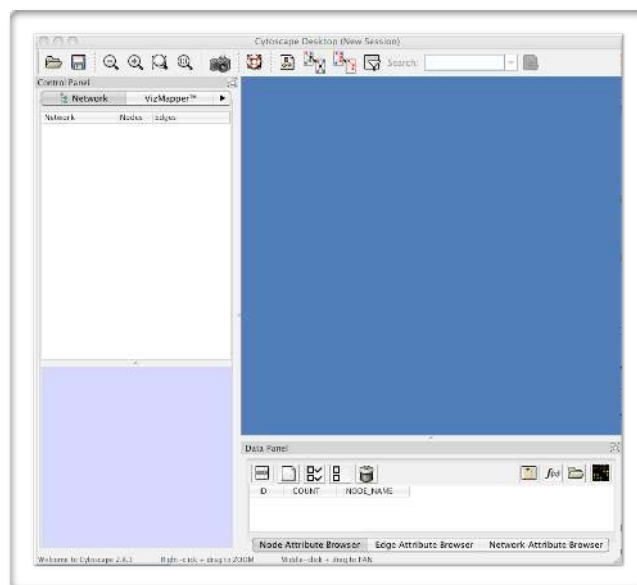
We have created a toy example to present a step by step analysis with OCSANA. You can download it from `ToyExample.cys` that we have included in our website.

In the “Download and Installation” section, we have explained where to obtain and how to install OCSANA’s software (either as standalone software or in its version included in BiNoM). For simplicity, we will run this example assuming OCSANA will be run from BiNoM.

Launch the Cytoscape application

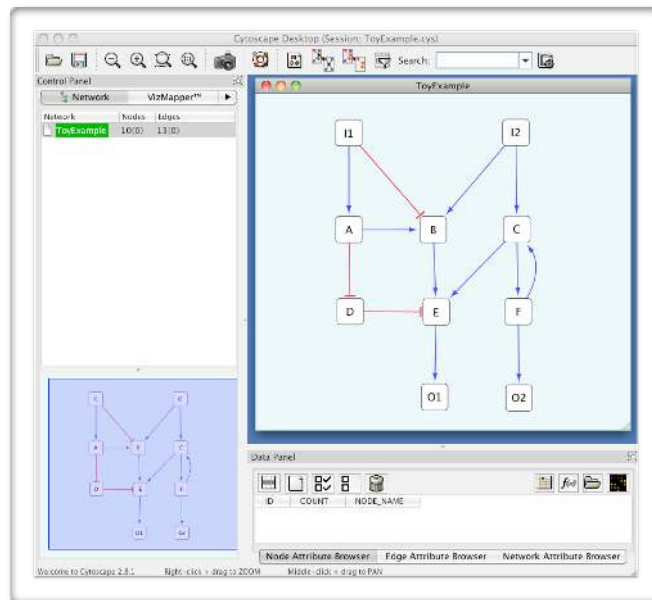
Double-click on the icon created by the installer or by running `cytoscape.sh` from the command line (Linux or Mac OS X) or double-clicking `cytoscape.bat` (Windows).

When you succeed in launching Cytoscape, a window will appear that looks like this (captured on Mac OS 10.4):



Launch the ToyExample.cys

Within the Cytoscape session, click on **File => Open**. This will open a dialog window to find where you have saved the `ToyExample.cys` file. Select this file. The `ToyExample` file should look like this:



This toy example network contains 10 nodes and 13 edges. The activation effects are represented as blue arrows whereas inhibition effects are in red blunt arrows.

Run OCSANA

With the ToyExample.cys open, we are ready to run OCSANA by clicking:

Plugins => BiNoM 2.2 => BiNoM Analysis => OCSANA analysis.

A dialog window to specify all the necessary parameters.

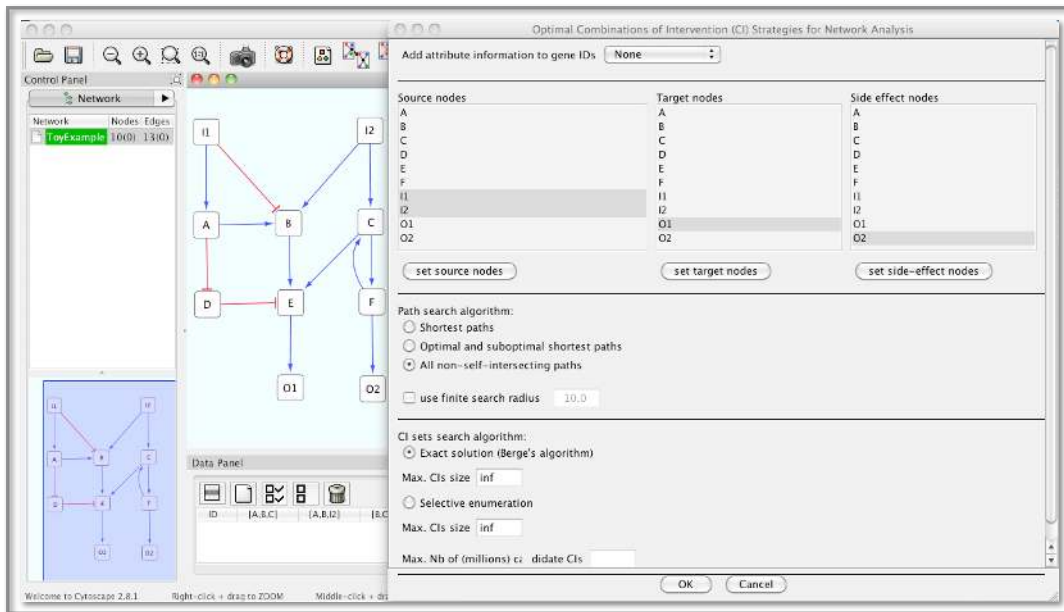
Let I_1 and I_2 be the two source nodes (where the signals start). Let O_1 be the selected target node (for which we want to block the signal from the specified source nodes). Let O_2 be the side-effect node².

Because the network is very small, we can select the for the Path Search Algorithm parameter the all non-self-intersecting paths. Also because the network is very small, we do not need to restrict the length of such paths, so we do not need to click further on “use finite search radius”.

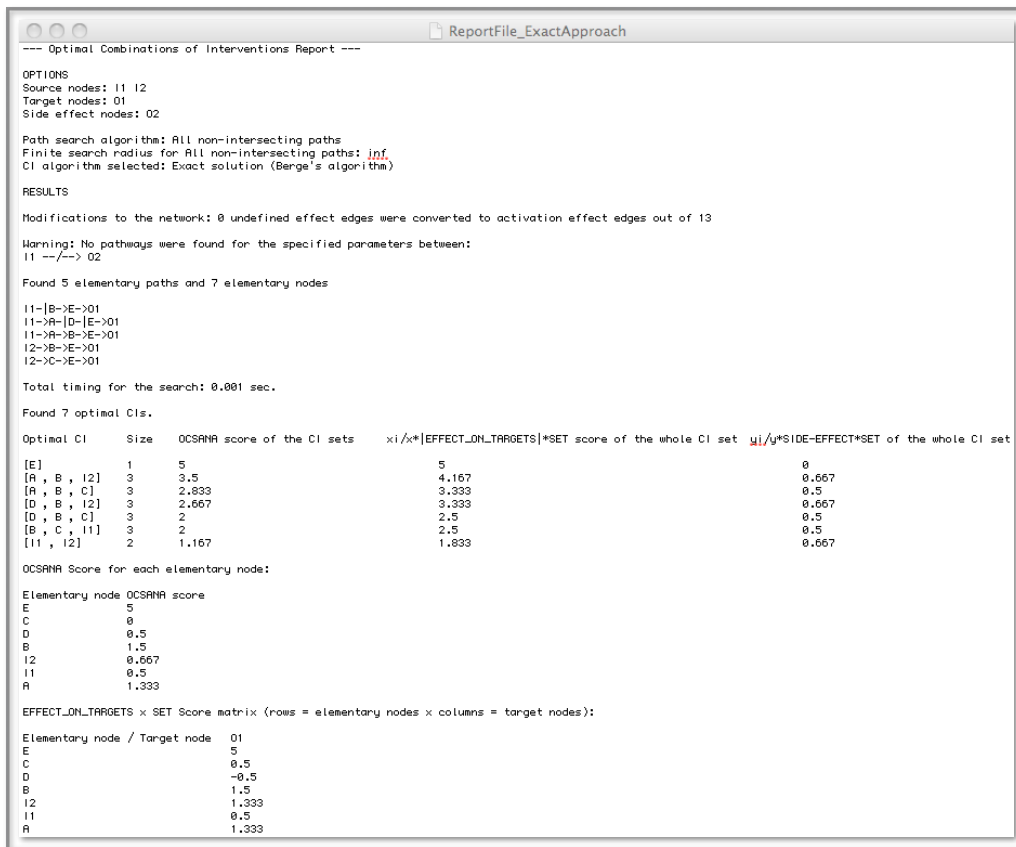
Exact Approach:

To run OCSANA under its **Exact Approach**, simply select the Exact solution (Berge’s algorithm) button from the CIs search algorithm option.

² Before launching the computations in OCSANA, if the user decides to deselect a node from any of the three input lists, it is enough to do control + click to the name of the node in the list, and thus de-select such node.



With all the aforementioned parameters, the dialog window should look like:



When the computations are finished, an output dialog window will display the results. To save this in a file, click on "Save report to file". Here are the displayed results:

At the top of the report file, the input nodes and parameters selected by the user, are displayed (source nodes, target nodes, side-effect nodes, path search algorithm, radius length if any was selected for All non-self intersecting paths and the type of approach used to compute CIs).

Next, the Results section is displayed.

Modifications to the network: Because OCSANA needs to have the information about the sign of the edges to compute nodes' scores, it will first verify if there were edges which sign was not defined. If so, then by default the undefined edge will be changed to activation (positive signed) edges and it will report how many of those undefined edge were changed.

Next there will be a report for the preprocessing part concerning the computation of the elementary paths (path between source and target nodes) and the paths between source nodes and side-effect nodes.

Warning: No paths found between selected nodes: First it will be reported if for the selected path analysis and its parameters, no paths were found between source and target nodes or between source and side-effect nodes. In this example, is shown that there were no paths between input node I₁ and the side-effect node O₂.

Elementary paths and elementary nodes found: The identified elementary paths between source and target nodes are displayed. It reports the number of elementary paths, the total number of elementary nodes involved in them, and the explicit list of such elementary paths. In this example, there were found 5 elementary paths: three connecting input node I₁ with O₁ and two elementary paths connecting I₂ with O₁.

Total timing for search: After this preprocessing step, it is reported the total time employed for computing optimal CIs. In this case the time was 0.001 seconds.

Number of found optimal CIs: The total number of identified CIs will be reported. In our example, 7 optimal CIs were identified.

Table displaying prioritized list of optimal CIs and their scores: A tab separated table will be displayed to the user, showing each one of the identified optimal CIs. These CIs will be displayed in a descending order according to their OCSANA's scores. For each one of these identified optimal CIs, their sizes and OCSANA's scores will be shown. Additionally, the OCSANA scores will be broken down in its two summands: $Sum_i \{x_i/x * |EFFECT_ON_TARGETS| * SET\ score\}$ and $Sum_i \{y_i/y * SIDE-EFFECT * SET\ score\}$, where i indexes the nodes in a given CI.

In our example, the highest ranked optimal CI is [E]. Because if formed by only one node, the size of the CI is 1. The OCSANA score of [E] is 5. Then $x_i/x * |EFFECT_ON_TARGETS| * SET\ score$ is 5. We see from the network's structure that E is not connected to the off-target node O₂, thus it is natural to see that the $y_i/y * SIDE-EFFECT * SET\ score$ is equal to 0.

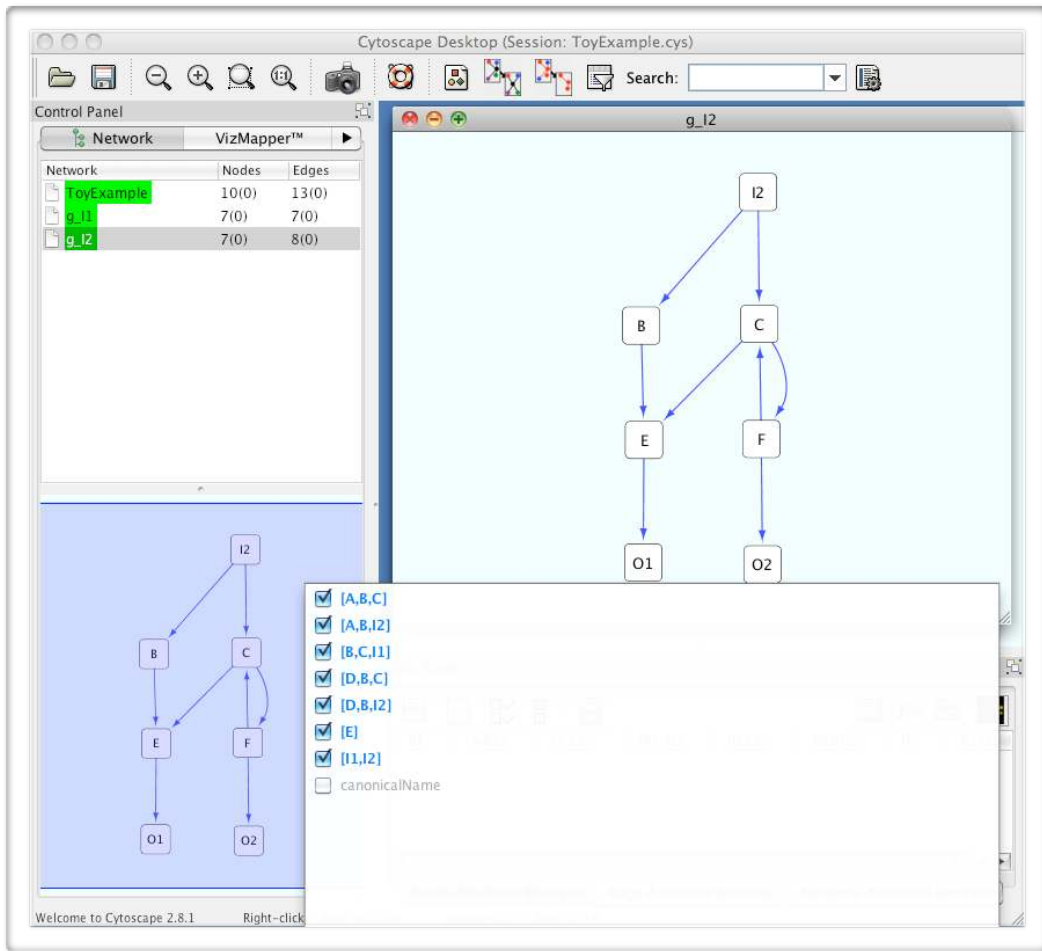
The user can copy-paste this tab separated table in order to analyze several aspects. For instance, the user might be interested to find the smallest possible size of CIs, even if they don't have the highest OCSANA scores (for example, when trying to identify synthetic lethal pairs, in which case, the CIs of size 2 will be selected and the user can re-rank them according to their OCSANA's scores). Also it is useful to observe in the last column corresponding to the summand involved for side-effects, whether the CIs has or not side-effect and how large is this in comparison with their effect on targets.

Table with OCSANA's score for each elementary node: this table displays each one of the elementary nodes with their corresponding OCSANA's score. This allows the user to appreciate the optimality of each individual elementary node.

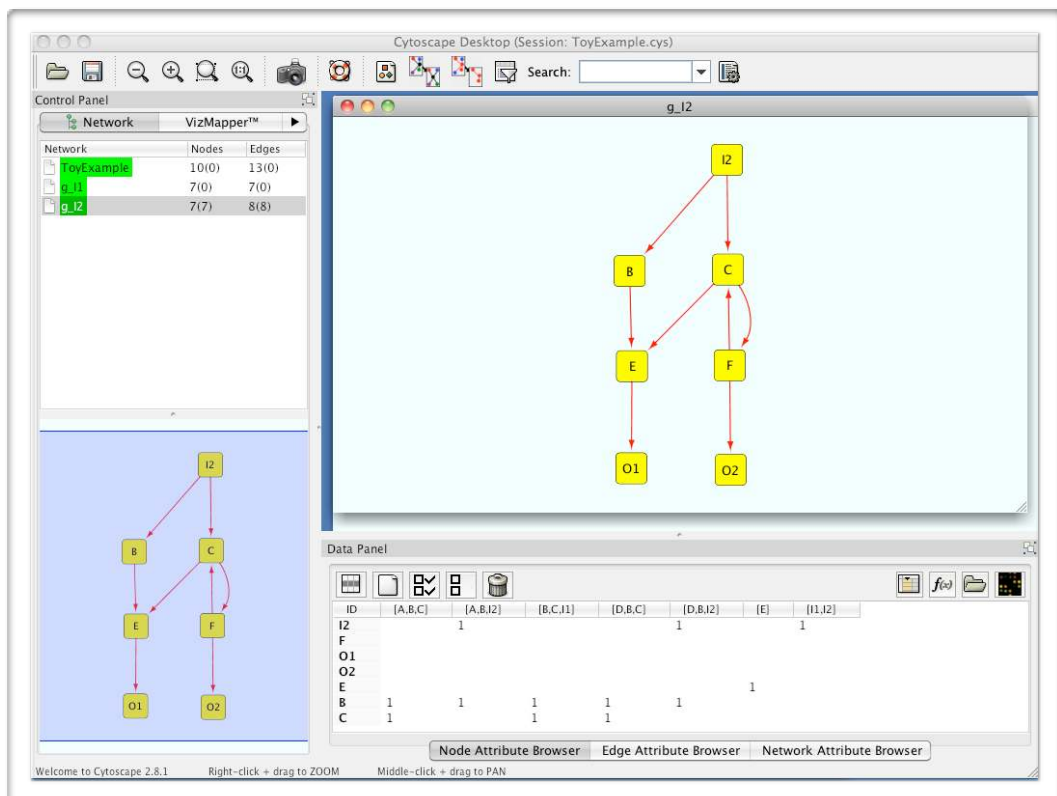
Table with elementary nodes and their corresponding EFFECT_ON_TARGET * SET score with respect to each one of the targets: the objective of this table is to provide an insight about the effect of a given elementary node on each one of the targets. This information aids the user to estimate the type of intervention to be applied in each node of a given CI. EFFECT ON TARGETS score provides an insight on the type of intervention to apply to vi. The type of intervention would depend on what is the desired action on the target node. For instance, E has a positive score in this table (and we see its effect on the target is pure activation), thus if the objective is to repress the target O_i , a knockout on E will be the appropriate. In contrast, if we would be to use the CIs involving node D, we observe that D has a negative score in this table. This indicates a inhibitory effect of D on the target (which can easily be verified on the graph), thus if the objective is to repress O_i then D should be knocked-in.

Visualize Results: This offers the option to generate the Cytoscape subnetworks for the top CIs identified (the amount of top CIs is a parameter the user can specify). For every CI identified, a subnetwork is generated with the involved source, target and off-target nodes and the elementary paths joining them (according to the selected path analysis chosen). The nodes in this subnetworks will now be assigned attributes to highlight the participation of the different nodes in the tops CIs.

For our current example, because we have two source nodes, two subnetworks will be generated, one for source nodes I_1 and all the downstream nodes to the targets and off-target nodes and, analogously, a second subnetwork for node I_2 . In the next picture, we see for instance, the subnetwork generated with the source node I_2 :



In this subnetwork we can easily observe the two elementary paths that connect the source node I₂ with the target node O₁: I₂->B->E->O₁ and I₂->C->E->O₁. We can also observe that I₂ has a path to the off-target (side-effect) node O₂.



Now, by clicking at the “Selection Attributes” button at the bottom of the Cytoscape window, we are able to see that the nodes in this subnetwork have new attributes: an attribute is per identified CI is created. Because we might have several CIs, the user can select the maximum amount of CIs attributes to be displayed (by default 100). From the picture above we see that 7 node attributes that were created. After highlighting each one of the nodes in the networks (appearing in yellow), we can see a “1” in a given node attribute if the elementary node in question belongs to the corresponding CI assign in the attribute. For example, the elementary node B, shows 1’s for the attributes for the first 5 node attributes because B belongs to each one of the assigned CIs, that is, [A,B,C], [A,B,I₂], [B,C,I₁], [D,B,C] and [D,B,I₂].

Selective Enumeration Approach:

Now, let’s assume we want to run OCSANA under the **Selective Enumeration**. In that case the additional parameters, approach, are the *Max. CI Size* and *Max. Nb. (million) of CIs* sets. Because the network has 10 nodes, 2 of which are targets, we have at most 8 nodes with which to ensemble CIs, thus being the network very small, we can select *Max. CI Size* to be 8. Now, for a given $1 \leq k \leq 8$ the largest possible number of candidate sets that can be constructed is $C_n^v = \frac{n!}{(n-v)!v!} = \frac{n(n-1)(n-2) \dots (n-v+1)}{v!}$, where $v = \lceil \frac{k}{2} \rceil$. Thus the largest amount of CIs that can be formed of each given size is at most $C_8^4 = 70$. Therefore selecting *Max. Nb. (million) of CIs* equal to 1, will allow to test at most 1,000,000 \gg 70 candidate CIs of each given size. What this means is that the selective enumeration approach, for these given parameters will test the entire set of candidate CIs (thus doing a full enumeration). Therefore the solution will be exactly the same as the exact approach and the output dialog window will display the exact same results:

```

--- Optimal Combinations of Interventions Report ---
OPTIONS
Source nodes: I1 I2
Target nodes: O1
Side effect nodes: O2
Path search algorithm: All non-intersecting paths
Finite search radius for All non-intersecting paths: inf.
CI algorithm selected: Approximation solution
Max. Nb of million of combinations to test: 1

RESULTS
Modifications to the network: 0 undefined effect edges were converted to activation effect edges out of 13
Warning: No pathways were found for the specified parameters between:
I1 --/--> O2

Found 5 elementary paths and 7 elementary nodes
I1->B->E->O1
I1->A->D->E->O1
I1->A->B->E->O1
I2->B->E->O1
I2->C->E->O1

Search for CI size 3
Total nb. of possible combinations: 20
Tested nb. of combinations: 20
found 5 sets size 3
Search for CI size 4
Total nb. of possible combinations: 15
Tested nb. of combinations: 15
Search for CI size 5
Total nb. of possible combinations: 6
Tested nb. of combinations: 6
Search for CI size 6
Total nb. of possible combinations: 1
Tested nb. of combinations: 1

Total timing for the search: 0 sec.
Found 7 optimal CIs.

Optimal CI      Size      OCSRNA score of the CI sets      xi/x*[EFFECT_ON_TARGETS]*SET score of the whole CI set      yi/y*SIDE-EFFECT*SET of the whole CI set
[E]             1             5             5             0
[A , B , I2]   3             3.5           4.167           0.667
[A , B , C]    3             2.833         3.333           0.5
[D , B , I2]   3             2.667         3.333           0.667
[D , B , C]    3             2             2.5             0.5
[B , C I1]     3             2             2.5             0.5
[I1 , I2]      2             1.167         1.833           0.667

OCSRNA Score for each elementary node:
Elementary node OCSRNA score
D      0.5
B      1.5
I2     0.667
C      0
A      1.333
E      5
I1     0.5

EFFECT_ON_TARGETS x SET Score matrix (rows = elementary nodes x columns = target nodes):
Elementary node / Target node  O1
D      -0.5
B      1.5
I2     1.333

```

BIBLIOGRAPHY

Berge, C. Hypergraphs. Combinatorics of Finite Sets. (1989) North-Holland, Amsterdam.

Haus, U.U., Klamt, S., Stephen, T. Computing knock-out strategies in metabolic networks. *J. Comput Biol.* (2008) 15, 259-268.

Zinovyev, A., Viara, E., Calzone, L., Barillot, E. BiNoM: a Cytoscape plugin for manipulating and analyzing biological networks. *Bioinformatics.* (2008) 24: 876-877.