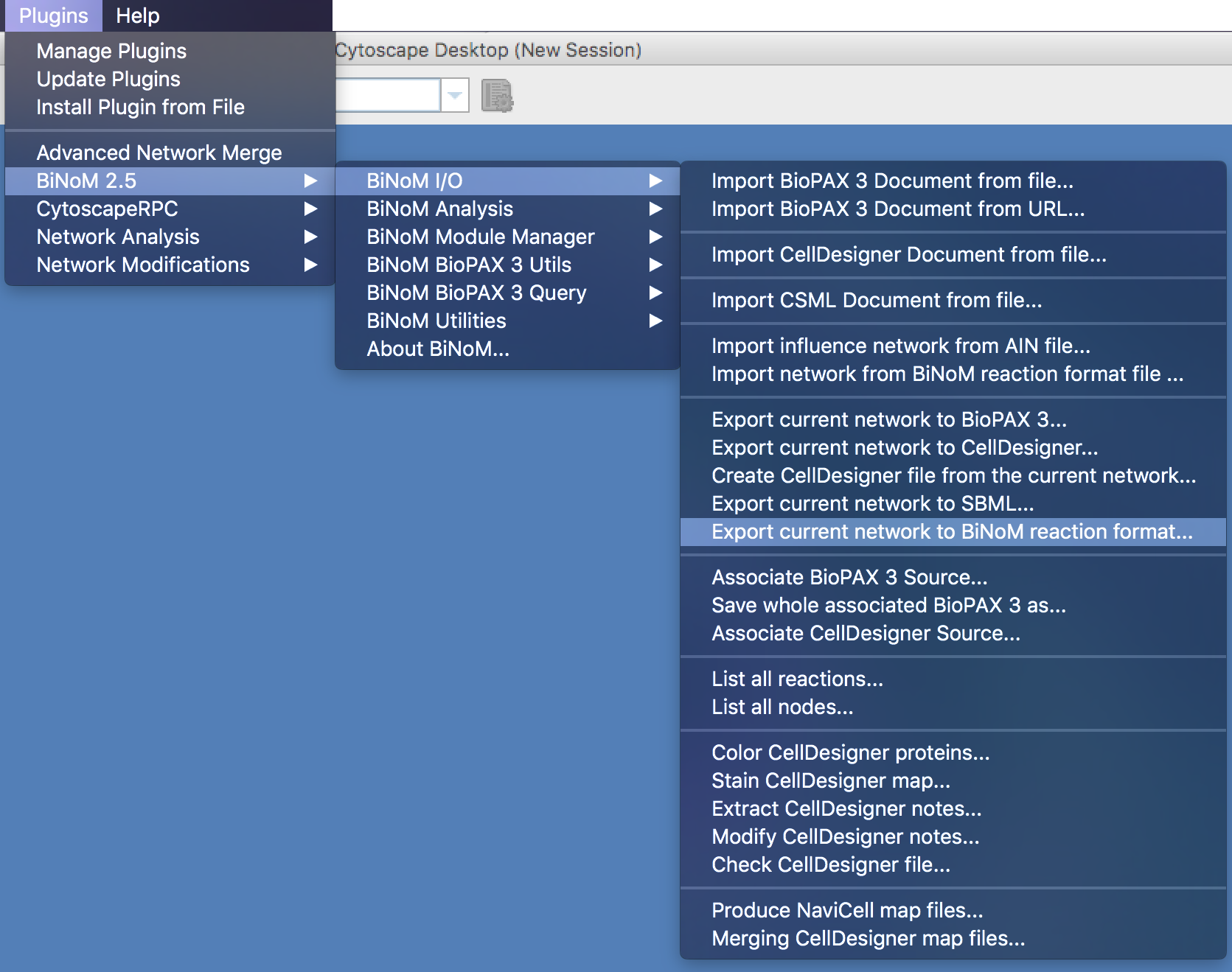
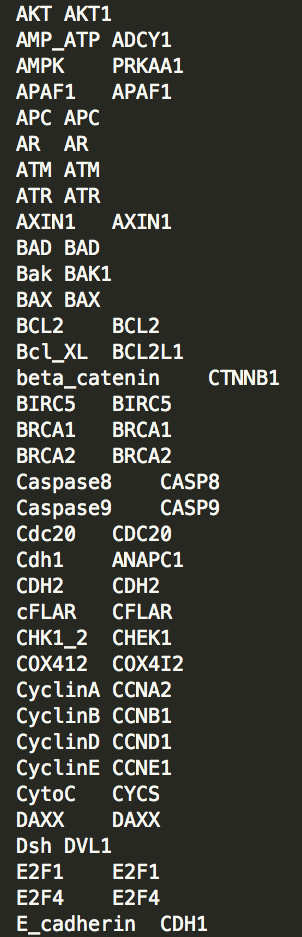
Tutorial to convert a Cytoscape network into a Navicell map

By Nicolas Sompairac, U900, Institut Curie

1. Create a working folder containing BiNoM\_all.jar and NaviCell Factory (<https://github.com/sysbio-curie/NaviCell>).
2. Create a folder in “*NaviCell-master/factory/maps*” called "*name\_src*"
3. From Cytoscape (2.8) export "*.brf*" file of the network using the BiNoM plugin and export network view as "name\_master.png" (Plugins -> BiNoM 2.5 -> BiNoM I/O -> Export current network to BiNoM reaction format):
   1. One can export network to “*.xgmml*” format in new version of Cytoscape, open it in Cytoscape 2.8 and export to “*.brf*” with BiNoM



1. Create smaller “.PNGs” for zoom-outs by using any image manipulation application or with the command:
   1. "sips -Z 320 name\_master-n.png --out name\_master-(n-1).png" (320 is width of new image, it should always be halved from the original. The largest image is the one with the biggest “n” and the smallest has to be “*name\_master-0.png*”)
2. Create “*.xml*” file by taking coordinates of topleft (x1,y1) and bottomright (x2,y2) nodes (from image with maximum size, this will be the maximum zoom), and run
   1. java -jar \*/BiNoM\_all.jar fr.curie.BiNoM.pathways.BiNoMReactionFormatToCytoscapeConverter -brf \*/name.brf -points 'topleftnode(x1,y1):bottomrightnode(x2,y2)'
3. Using NaviCell, it is possible to add links for species to specific databases such as Gene Cards or Gene Names. To do this, you have to generate a table containing two columns: The first column containing names of nodes in the network and the second column containing the correct ID of the wanted database. You can then integrate these IDs in the map file by using the python script “Modify\_annotations.py” (a help is provided by launching python Modify\_annotations.py --help). An example of the usage of this this script is:
   1. python Modify\_annotations.py --corresp IDtable\_file.txt --origin-file name.xml --result-file name.xml --link-type HUGO



1. Rename "*name.xml*" and "*name-n.png*" files to "*name\_master.xml*" and "*name\_master-n.png*" and move them to the folder “*NaviCell-master/factory/maps/name\_src*”
2. Create a “*config*” file in “*NaviCell-master/factory/maps/name\_src*” folder with the following 4 lines:
   1. base: name\_
   2. title: name
   3. name: name
   4. showDefaultCompartmentName: true
3. Make “*config*” file executable with the command line:
   1. chmod +x NaviCell-master/factory/maps/name\_src/config
4. Specify "*build\_name.sh*" file, by changing "configdir" and "destdir"
5. Generate the NaviCell file by running:
   1. sh build\_name.sh
6. Open in Chrome with (close Chrome first):
   1. open -a "Google Chrome" --args --allow-file-access-from-files 'file:///\*/*NaviCell-master*/site/docroot/navicell/maps/\*/master/index.html'
   2. (\* name of your folder)
7. If Markers are not on top of corresponding entities, the “.PNGs” have to be adapted to the correct resolution (***By Hand***). This procedure can be eased by taking as reference the image of the map obtained from CellDesigner.
8. If the map is not visible, be sure that you have opened Chrome with the previous command. If the problem persists, you may use the following command line to repair the Google Maps API conflict:
   1. find NaviCell-master/site/docroot/navicell/maps/***map\_name***/master/ -name "index\*" -exec sed -i 's/js?sensor=false/js?sensor=false\&v=3.30/g' {} +