Manual

EPIC: Elution Profile-based Inference of Protein Complex Membership

Welcome to the EPIC installation manual, in this document we will have a step-by-step guide on installing and running the EPIC pipeline.

1 Installing dependencies

In order to run EPIC locally on your machine, docker needs to be installed, and for easy installation of the EPIC docker image we recommend installing kitematic, as well as creating a docker hub login. Additionally to the docker suit we also recommend installing the current version of Cytoscape.

Docker:

For mac OS X follow instructions from (https://docs.docker.com/docker-for-mac/)

Kitematic:

The current version of Kitematic can be downloaded and installed from (https://kitematic.com/).

Cytoscape:

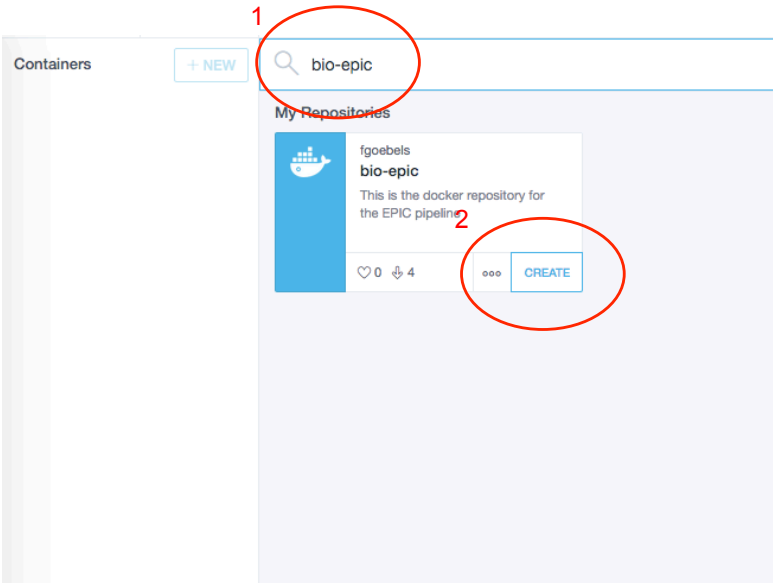
Cytoscape is available from (http://www.cytoscape.org/).

2 Building the docker image

The docker EPIC image is online available and can be easily installed by running the following command line in the Terminal/shell (this is recommanded):

docker pull baderlab/bio-epic

Alternatively the docker can also be installed using the Kitematic user interface as follows:



1. First starting Kitematic and entering bio-epic into the search mask

2. Clicking on the CREATE button on the Repository selection

3 Input files

EPIC accepts three types of input files:

3.1 Elution profile files

This is a tab separated file contains the elution profile for all the proteins in one distinct co-fractionation experiment, thus in case of multiple experiments each experiment has its own elution profile file. The first line of the elution profile file is the header line where the first column is ignored, and each subsequent column contains the name of the fraction. Each following line of the elution profile file contains the elution profile for one protein, where the first column contains the protein ID and the following columns each contain the MS value (either spectral counts or intensity) of the protein in each distinct fraction. Examples of the elution profile file can be found in EPIC GitHub (https://github.com/BaderLab/EPIC/tree/master/test_data/elution_profiles/).

3.2 Reference protein complexes (optional)

This file can be optionally supplied as protein complex reference set, instead of having them automatically generated from CORUM, IntAct, and GO. This file does not have a header line, and each row of the file describes one protein complex by concatenating all its members with tab characters. An example of this file can be found in EPIC GitHub (https://github.com/BaderLab/EPIC/tree/master/test_data/Worm_reference_complexes.txt).

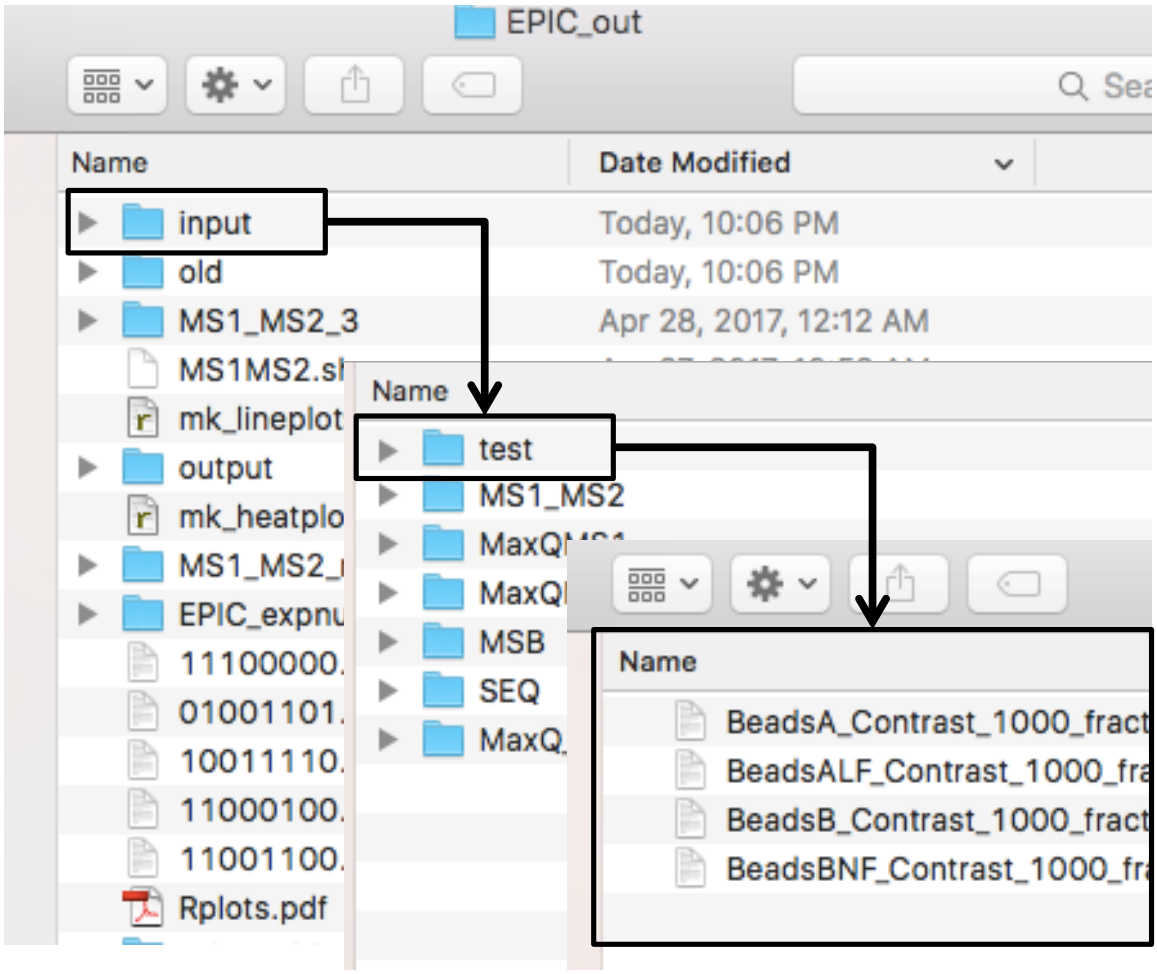
3.3 Functional annotation network (optional)

This file can be supplied as an alternative source for functional annotation data that should be used by the EPIC pipeline. This file is also tab separated and has a header line that contains the name of each supplied functional annotation score. Each row contains the scores for each protein pair, where the first two columns contain the protein IDs, and the following the annotation scores for the protein interaction. An example of functional annotation network file can be found in EPIC GitHub (https://github.com/BaderLab/EPIC/blob/master/test_data/WormNetV3_noZeros_no_physical_interactions.txt).

4 Input folder structure

For EPIC to properly run the user needs to create a folder that will be linked with the docker image, from which the docker can read and write data from the user. This folder should contain a sub-folder for each elution file the user wants to process, and this folder stores all the elution profile files for the given project. In turn EPIC will read in the data and once done will create an output folder in the same directory that contains all the output files and has the suffix "inputfoder_out". It is noted that before running, this folder should have only contain elution profile files. This file folder shouldn't have any other unnecessary files. If you want to run EPIC again, make sure remove the output files/folders first.

For example, here we created a folder named input, which contains a folder named test, which in turn contains 4 elution profile files.



5 Starting EPIC

We recommend setting 6.0 GiB for Memory and 3.0 GiB for Swap. Make sure you re-start Docker after update from our Docker repository.

For mac OS X, Jupyter EPIC can be started by opening a terminal window and first using cd to change to the input folder (see 4) and then using the following command:

```
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
```

Then type the following command:

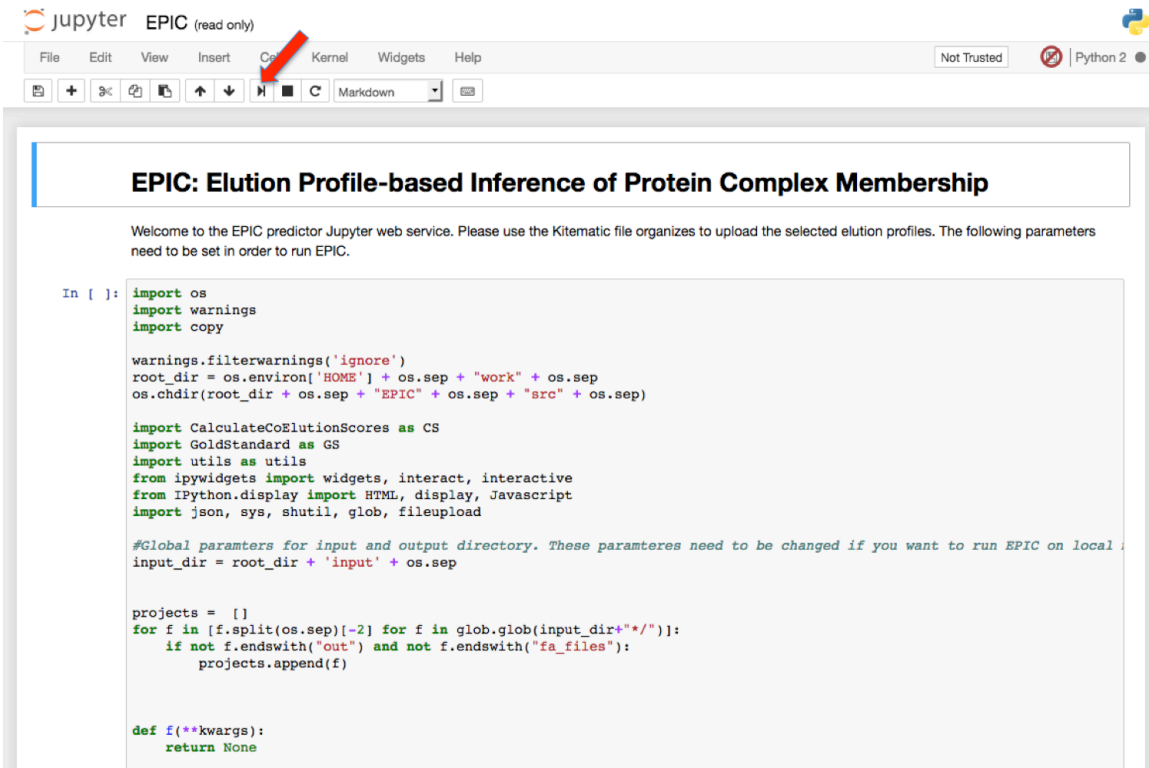
docker run --add-host="localhost:$(ifconfig en0 | grep inet | grep -v inet6 | awk '{print $2}')" -it --rm -p 8888:8888 -v "$DIR:/home/jovyan/work/input" baderlab/bio-epic start-notebook.sh --NotebookApp.iopub_data_rate_limit='100000000' --NotebookApp.token=''

After starting the EPIC container several messages will appear and one of the messages will be the Jupyter token that will allow you to connect to the Jupyter notebook.

```
[I 14:32:32.024 NotebookApp] JupyterLab alpha preview extension loaded from /opt/conda/lib/python3.5/site-packa
rlab
[I 14:32:32.034 NotebookApp] Serving notebooks from local directory: /home/jovyan/work
[I 14:32:32.035 NotebookApp] 0 active kernels
[I 14:32:32.035 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8888/
[I 14:32:32.035 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirm
[I 14:32:45.875 NotebookApp] 302 GET / (172.17.0.1) 0.90ms
[I 14:32:49.576 NotebookApp] Writing notebook-signing key to /home/jovyan/.local/share/jupyter/notebook_secret
```

The highlighted part can be modified to “http://localhost:8888/”, after typing this into a web browser, you will see this:

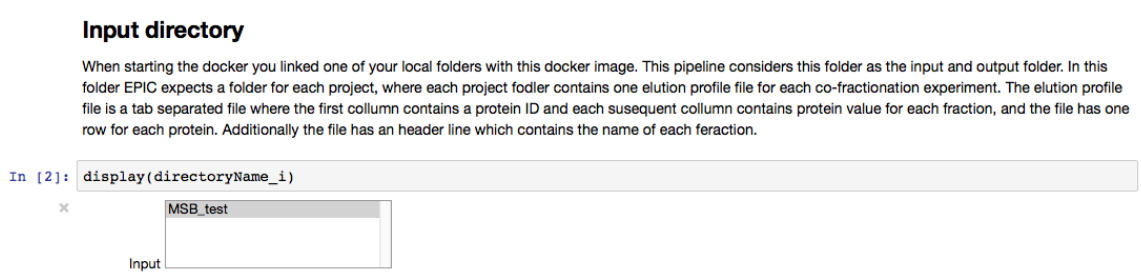
Form here, after clicking EPIC.ipynb the EPIC notebook will be started and you will see:



You can click the run button (red arrow) to process each cell. Jupyter is easy to use, as there are plenty of online tutorials available. If you are interested into this, you should definitely check them.

There are a couple of steps require user input parameters, which we highlighted as below:

Make sure you go back to re-click to run the cell after you typing your own input!



In the input box shown above, you should see the file folder contains all your elution profile files, and you should click it and then click run to process to the next cell/step.

Feature slection:

Please select which co-elution features should be used to generate the co-eluton network. We recommend using MI, Bayes, PCCN, and Apex.

```
In [4]: display(features_i)
```

- PCCN ☑
- Jaccard ☑
- Apex ☑
- MI ☑
- Euclidean ☑
- WCC ☑
- Bayes ☑
- PCC ☑

In the input box shown above, you should click the correlation scores you want to use. It is not necessary the more the better. You should make your own decision. Ideally, you can benchmark all possible combinations using super computing platform.

Num cores

Number of cores that can be used to calculate co-elution scores. Increasing this number reduces run time if the docher and the machine has multiple cores.

```
In [7]: display(num_cores_i)
```

num_cores 4

In the input box shown above, you should give the number of cores you want to use for running EPIC. This is totally dependent on the ability of your own computer, the more cores used, the faster the EPIC will be.

Classifier:

Here you can select the classifier used to generate the co-elution profiles. EPIC supports both SVM and random forest. We recommend to use random forest.

```
In [8]: display(clf_i)
```

Classifier ◉ Random forest

○ SVM

In the input box shown above, you see the two machine learning algorithms used in EPIC. You can select the one you want to use.

Reference data:

Here you can either supply an taxid for automatic generation of a reference, or upload a selected set of reference complexes. Please note automatic reference data gereneration is only supported by Uniprot IDs, and the format for the supplied reference complexes is one complex per line, in which the line contains all the cluster's member IDs separated by tabs.

```
In [10]: display(target_species_i)
         display(ref_file)
```

target_species 6239

Browse

In the input box shown above, you should type the taxonomy id for the species you want to study. EPIC helps you automatically downloaded all the data from public databases: CORUM, GO and IntAct. The reference protein complexes file was generated after merging all the redundant protein complexes.

```
In [9]: display(target_species_i)
        display(ref_file)
```

target_species taxid i.e. 6239 (Worm)

Browse

```
Uploaded to `/tmp/All.complexes.txt` (16.00 kB)
```

Or you can supply a reference file you want to use from you own collection. An example of this reference file can be found at GitHub (https://github.com/BaderLab/EPIC/blob/master/test_data/Worm_reference_complexes.txt). In this case, you should not touch your "target_species" input box. And browse the reference complexes file. When you see the "Uploaded to …" message, this tells you the local functional evidence data is uploaded successfully.

Mode:

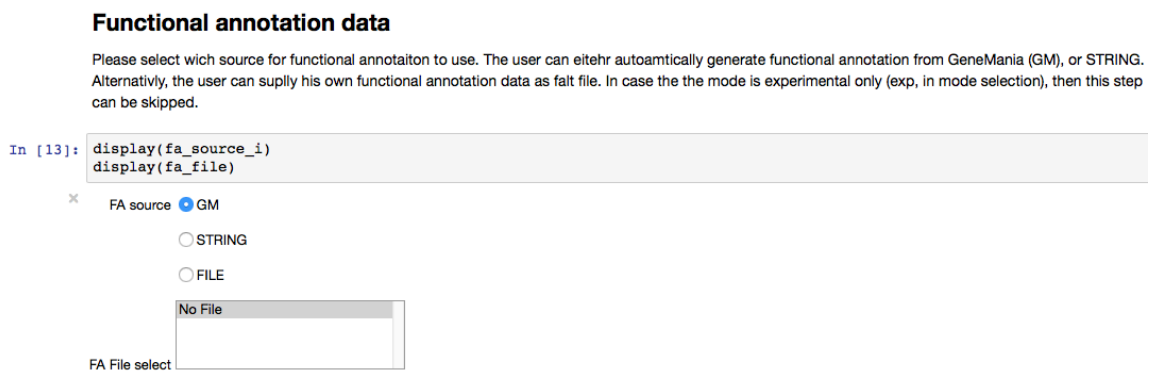
The mode which EPIC schould be run with. The supported modes are: experiment only (exp), t
experiment only when wanting to run EPIC without introducing functional annotation bias into

```
In [12]: display(mode_i)
```

Mode ◉ exp

○ comb

In the input box shown above, you can select, if you want to incorporate functional evidence or not. “exp” means purely based on functional evidence, and “comb” means the data is based on both experimental data and functional evidence.



In the input box shown above, you can select the resource of functional evidence you want to incorporate. If you select “exp” in the Mode section, this step can be skipped and left as default.

You can also supply your own functional evidence file. An example of function evidence file can be found in GitHub as mentioned before (https://github.com/BaderLab/EPIC/blob/master/test_data/WormNetV3_noZeros_no_physical_interactions.txt). If you want to use your own functional evidence data, you should create a file folder named “fa_files” in your input directory and put your functional evidence data into this file folder (as below). In the input directory, there should be two file folders, one has elution files (MSB_test) and one has functional evidence data (fa_files). It is noted that the elution file folder name can be anything you want, but the functional evidence data file folder is fixed.

```
In [17]: display(fa_source_i)
         display(fa_file)
```

FA source ○ GM

○ STRING

◉ FILE

FA File select: No File / WormNetV3_noZeros_no_physical

Then to use functional evidence data, you will select the "FILE" and the selected functional evidence data file as shown below.

Calculating co-elution scores

This is the most time intensive step of EPIC and on average take 20 min per co-elution score per experiment.

```
In [*]: # Caculating scores
        scoreCalc = CS.CalculateCoElutionScores(this_scores, elution_datas, output_dir + ".scores.txt", num_cores=num_cores)
        scoreCalc.calculate_coelutionDatas(all_gs)
        # For debug readin precalculated co-elution scores
        #scoreCalc.readTable(output_dir + ".scores.txt", all_gs)

        all_gs.positive = set(all_gs.positive & set(scoreCalc.ppiToIndex.keys()))
        all_gs.negative = set(all_gs.negative & set(scoreCalc.ppiToIndex.keys()))
        all_gs.rebalance()

        Filtering: all_6mg.txt
        Before filtering 8679861 PPIs
        filtering
        After filtering 2883171 PPIs
        Num of PPIs across all data sets after filtering 2883171
        100000
        200000
        300000
        400000
        500000
```

This is the most time-consuming step in EPIC. It might take hours, which totally dependent on the size of your input data. The star pointed by red arrow means this cell/step is currently running.

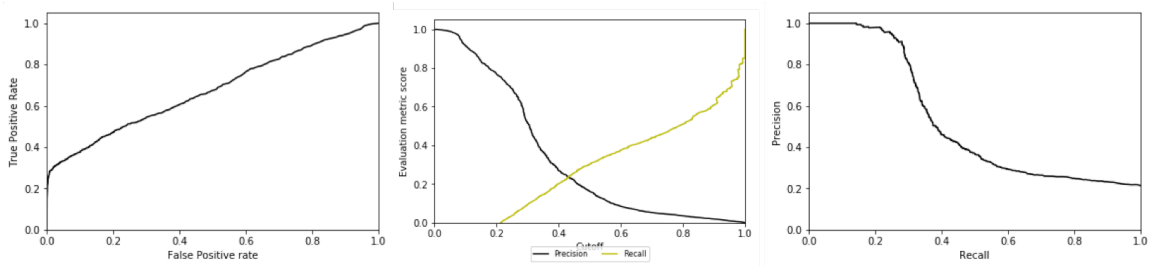
6 EPIC output

After following the prompts of the Jupyter notebook several output files and plots will be generated as well as Cytoscape network will be generated in a running Cytoscape instance.

7 EPIC output

After following the prompts of the Jupyter notebook several output files and plots will be generated as well as a Cytoscape network will be generated in a running Cytoscape instance.

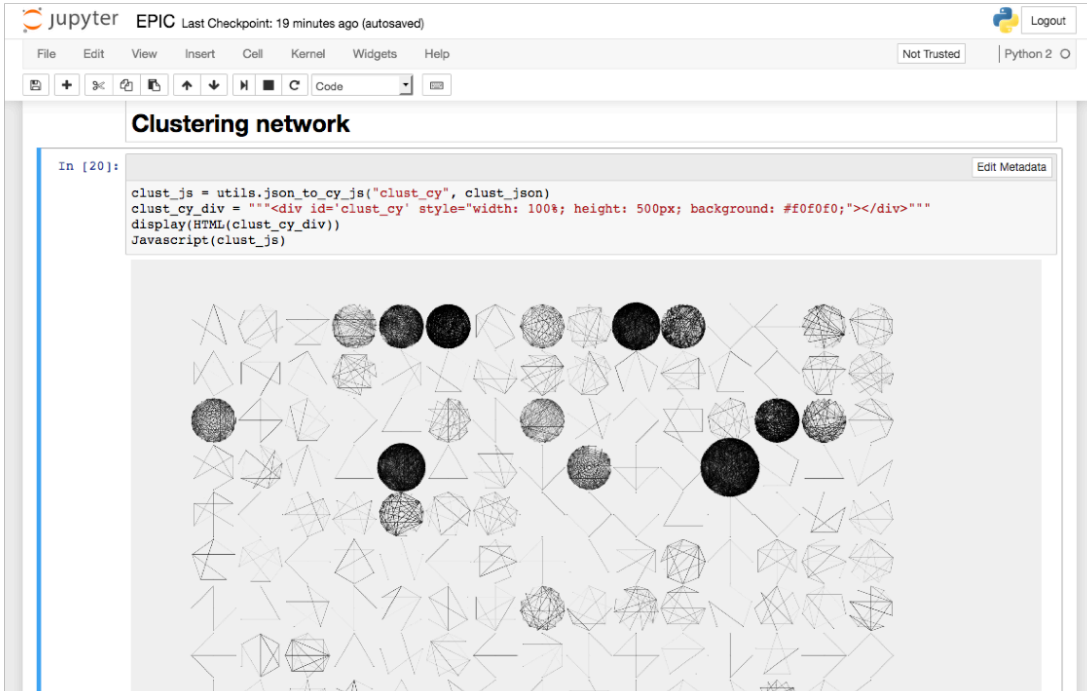
7.1 Classifier performance plots



From left to right: ROC curve, Precision/Recall vs confidence score, and PR curve.

The first three plots that are generated show the classifiers performance in predicting co-complex membership from the reference data. The first plot show both precision and recall for various classifier confidence values, and this plot will give the user insight to evaluate how confident an predicted interaction with a certain score is. The following two plots show the ROC and PR curve for the classifier.

7.2 Cytoscape visualization within the browser

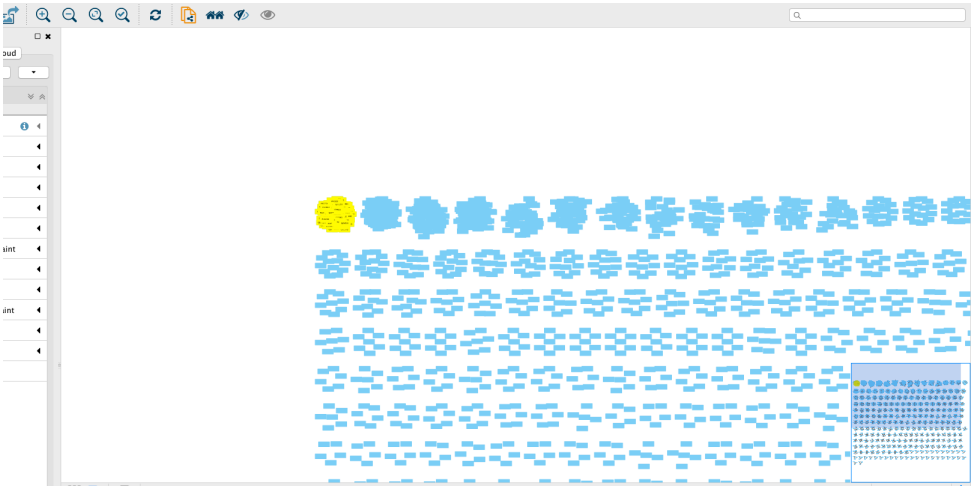


The next figure shows the interactive cytoscape.js widget that visualizes the generated protein complexes, and the user can zoom in and out as well move clusters in side this widget. This allows users to have an initial glance at their created protein complexes, and can always be generated even if the user does not have Cytoscape locally installed.

7.3 Cytoscape

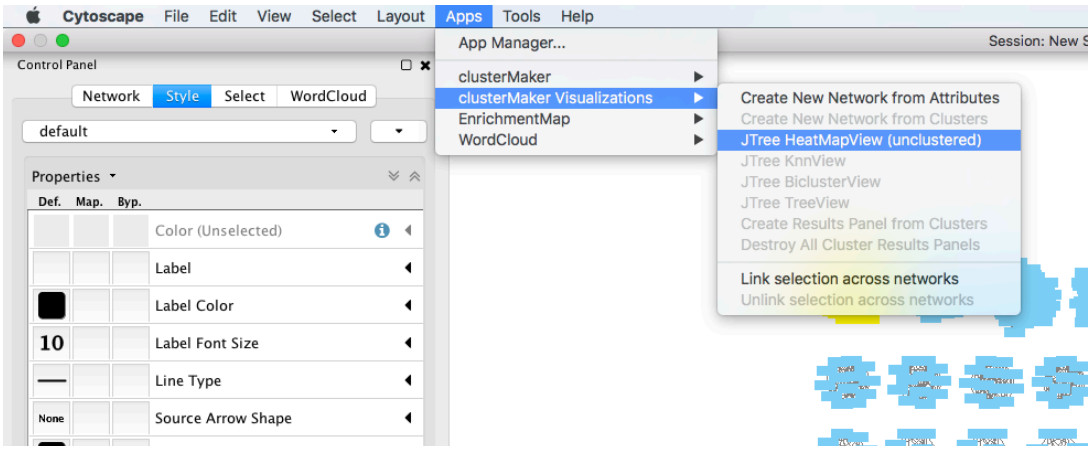
(Make sure your Cytoscape is running before running the Jupyter Notebook, if you want to process this step.)

Running the last cell in the EPIC Jupyter script will send the created clusters to a running Cytoscape instance and generate a new network named “EPIC clusters”, and once selecting the running Cytoscape session the user is prompted to create the view. Once the view has been created the user will see a network that contains a connected component for each predicted cluster. (Make sure to change the layout to see the map!)

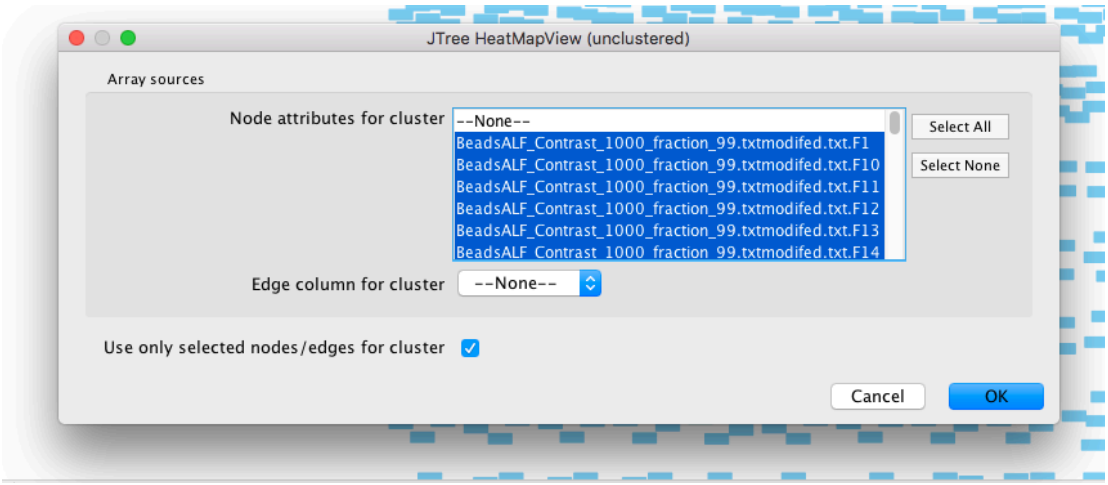


If the user has installed the clustermaker app from the Cytoscape app store he can visualize the elution profile for any selected protein as follows:

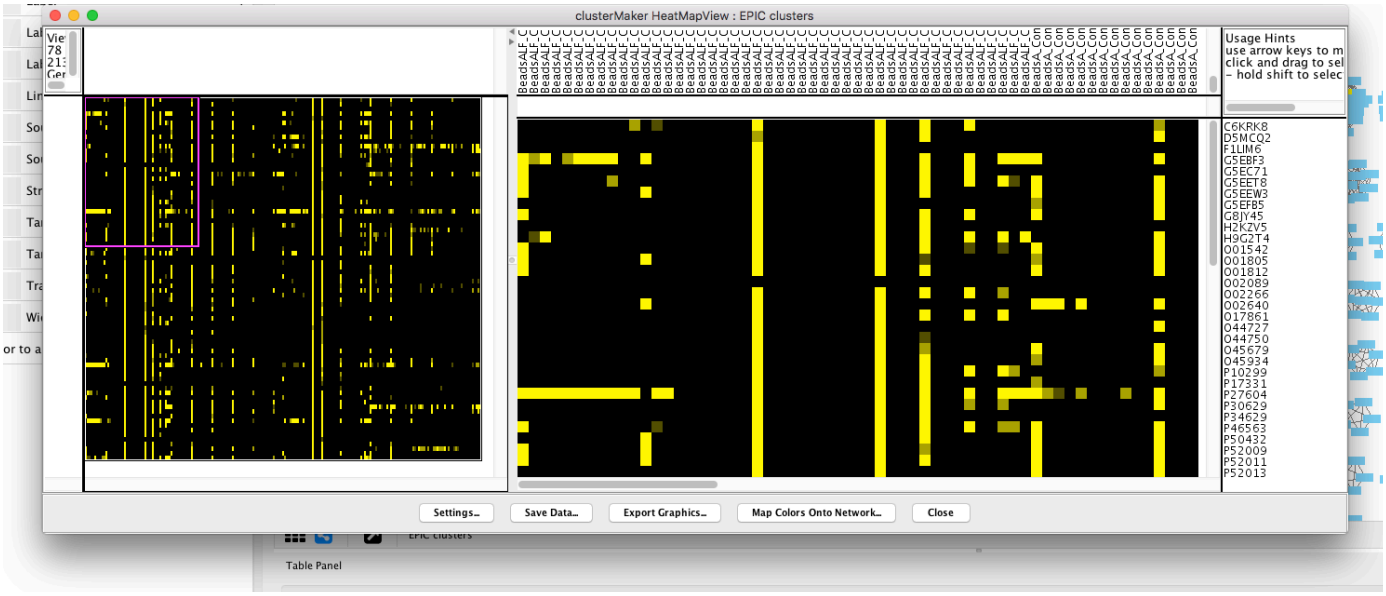
First select the JTree heatMapView from Apps:



This will open the following window, in which the user should select all elution score Node attributes, and check the lower box that says “Use only selected nodes/edges for clustering”:



This will result in opening another window in which the user can freely browse the elution profiles of the selected proteins:



8 Collecting the output

EPIC will create an output directory in the shared folder that contains the project folder and stores several output files in it. Each file has a prefix consisting of the used classifier method and the mode in which EPIC was run with (modes are EXP, or COMB). For example if EPIC was run with a random forest classifier and experiment only each file will have the prefix “Out.rf.exp”. The most important files are as follows:

***.scores.txt**
contains the raw co-elution score.

***.pred.txt**
This tab-separated file contains the generated binary protein interaction network, where the first two columns store the protein ids and the third column is the classifier confidence score.

***.comp.txt**
This is the predicted protein complex file, where each line describes one protein complex.