Chapter 25

Automated Computational Inference of Multi-protein Assemblies from Biochemical Co-purification Data

Florian Goebels, Lucas Hu, Gary Bader, and Andrew Emili

Abstract

Biology has amassed a wealth of information about the function of a multitude of protein-coding genes across species. The challenge now is to understand how all these proteins work together to form a living organism, and a crucial step for gaining this knowledge is a complete description of the molecular "wiring circuits" that underlie cellular processes. In this chapter, we describe a general computational framework for predicting multi-protein assemblies from biochemical co-fractionation data.

Key words Protein-protein interaction, Bioinformatics, Machine learning, Systems biology, Protein interaction prediction, Protein complex prediction, Python, Docker, Cytoscape

1 Introduction

Previously, in Chapter 12, we discussed in detail how to plan and execute the co-fractionation (e.g., non-denaturing chromatography) part of the biochemical purification/mass spectrometry (BP/MS) experimental pipeline, while in this chapter we provide an in-depth description of the computational part required for proteomics data processing, analysis, and interpretation. Specifically, we describe EPIC (**E**lution **P**rofile-based **I**nference of Protein **C**omplex Membership), a software toolkit which can automatically generate confidence binary protein interactions and predict the memberships of corresponding stable multi-protein assemblies from raw co-elution proteomics data [1]. The EPIC is accessible to the public via a GitHub (https://github.com/BaderLab/EPIC) or Docker Hub (https://hub.docker.com/r/baderlab/bio-epic/) repository.

Since it does not rely of achieving purity, co-fractionation is a practical but imperfect experimental approach to characterize multi-protein complexes. Our computational workflows have been optimized to minimize the number of spurious protein pairs that are predicted to interact because they simply happen to co-elute at the same time (due to similar biophysical behavior during

Joseph A. Marsh (ed.), *Protein Complex Assembly: Methods and Protocols*, Methods in Molecular Biology, vol. 1764, https://doi.org/10.1007/978-1-4939-7759-8_25,

chromatography) but which are actually functionally unrelated (we refer to such events as the "chance co-elution" problem). Toward this end goal, we apply basic statistical criteria to measure protein similarity based on their respective biochemical fractionation profiles, followed by more sophisticated machine learning to exploit publicly available supporting functional association evidence to guide the selective filtering of biologically irrelevant correlations.

We demonstrated the practical utility and real-world performance of this co-fractionation data analysis pipeline, which was first used to predict 13,993 high-confidence physical interactions among 3006 stable protein complexes in human [2] and in a follow-up experiments that identified 981 conserved metazoan complexes [3]. Below, we outline implementation of the stand-alone EPIC software designed to facilitate such analyses by biologists lacking computational expertise.

2 Materials

As EPIC is a computational pipeline, the only physical equipment required is suitable computer infrastructure (e.g., Linux- or Mac OSX-enabled machine). However, we provide suggestions for implementation as well as minimal and recommended specs. Moreover, we list both required and optional software for running EPIC.

2.1 Equipment

1. Working computer (Mac OSX/Linux-based) (*see* **Note 1**).
   - Minimal: one core, 8 GB RAM.
   - Recommended: four cores, 8 GB RAM.
2. Internet connection (optional).
   - Required for automatic generation of reference data set and automatic download of STRING and GeneMANIA.
   - Alternatively the user can supply own reference clusters and functional annotation scores as flat file (*see* below for file formats).

2.2 Supplementary Software

1. Docker (mandatory).
2. Cytoscape [4] (optional but highly recommended) (*see* **Note 2**).
3. We highly recommend basic understanding for navigating a Jupyter script.

2.3 File Formats

There are three main types of input files used in EPIC: elution profile data, reference protein complexes, and functional annotation data. Example files for Worm (taxid 6239) can be found in the test_data directory inside the EPIC GitHub repository (https://github.com/BaderLab/EPIC/tree/master/test_data).

1. Elution Profile Data

This is a tab delimited file or data matrix containing the elution profiles for all the proteins detected by mass spectrometry in one distinct co-fractionation experiment. For example data, see https://github.com/BaderLab/EPIC/tree/master/test_data/elution_profiles. Multiple experiments will result in multiple co-elution profiles (i.e., one file for each experiment). The header is located on the first line and contains the names for each fraction, while each subsequent row contains the various protein IDs (accessions/descriptions) and the corresponding detection values (e.g., spectral counts) recorded in each fraction.

2. Reference Protein Complexes (Optional)

   The user may supply a custom set of reference protein complexes (e.g., CORUM [5], IntAct [6], GO [7]) for use in training the EPIC scoring algorithm (*see* **Note 3**). In this file, each complex is summarized in one line by concatenating all member protein IDs with tab-delimited characters. Example reference complexes for Worm can be found here https://github.com/BaderLab/EPIC/blob/master/test_data/Worm_reference_complexes.txt.

3. Functional Annotation Data (Optional)

   EPIC uses functional associations as additional features to minimize chance co-elution, and in this step the user can provide a predefined set of functional associations (*see* **Note 4**). The data in this file should be on protein interaction level and will be added as additional features to each candidate PPI without further modifying the added features. In this file each column represents a functional association score, and each row consists of protein pair followed by available functional association scores (columns are tab separated). This file has a header row, which contains each column respective functional annotation score name. **Note 4** contains some examples for species-specific functional annotation resources, and Subheading 3.3.5 lists the default sources used in EPIC (e.g., https://raw.githubusercontent.com/BaderLab/EPIC/master/test_data/Wormnet_funanno.txt).

3 Methods

The EPIC software mostly runs automatically, and thus the most labor-consuming part for establishing the computational scoring pipeline is setting up docker and starting EPIC. However, this step can be easily completed within an hour. EPIC runs automatically and has on average a runtime of 40 min per co-elution score per experiment, divided by the number of available computer cores. The most computationally heavy part is generating the co-elution scores for all pair-wise protein combinations.

3.1 Installing Required Software

To run EPIC, it is mandatory to install docker, which is a lightweight virtual machine, which will enable operation of the entire EPIC pipeline.

- Docker. For Macintosh instructions see https://docs.docker.com/docker-for-mac/. For Linux see https://docs.docker.com/engine/installation/.
- Cytoscape. Cytoscape is available from http://www.cytoscape.org/.

Once docker is installed, one needs to change the assigned memory to at least 6 GB. This is achieved by selecting docker, followed by preference, and then selecting advanced options.

3.2 Installing EPIC

Once docker is installed, EPIC can be installed. This step can take time, depending on the available Internet speed, since the EPIC image is roughly 8 GB in size.

1. Open a terminal.
2. Enter the following command:

```
$ docker pull baderlab/bio-epic
```

3. Create a folder on your machine named EPIC.
4. Within this folder, create another subfolder for data (e.g., My_EPIC_PROJ).
5. Move all project-relevant co-fractionation data files into this folder (e.g., copy chromatographic elution files into the MY_EPIC_PROJ folder).

3.3 Running EPIC

1. Open/select a terminal window.
2. Navigate to the previously generated EPIC folder (*see* **Note 5**).
3. Download the EPIC start script (https://github.com/BaderLab/EPIC/blob/master/src/start-EPIC), and put it in the EPIC folder, and double click the file.
4. Open a browser and enter http://localhost:8888/tree.
5. Once the web page is finished loading, click on the EPIC.ipynb symbol.
6. When running EPIC for the first time, it is recommended to go through the EPIC script in a step-by-step wise manner by repeatedly pressing the play button.
7. Press play until an input directory selection appears (*see* Fig. 1a), and select a folder from the list (e.g., MY_EPIC_PROJ). From now on, we no longer indicate if a user has to press play to reach the next input mask, rather we describe what to do at each input mask.

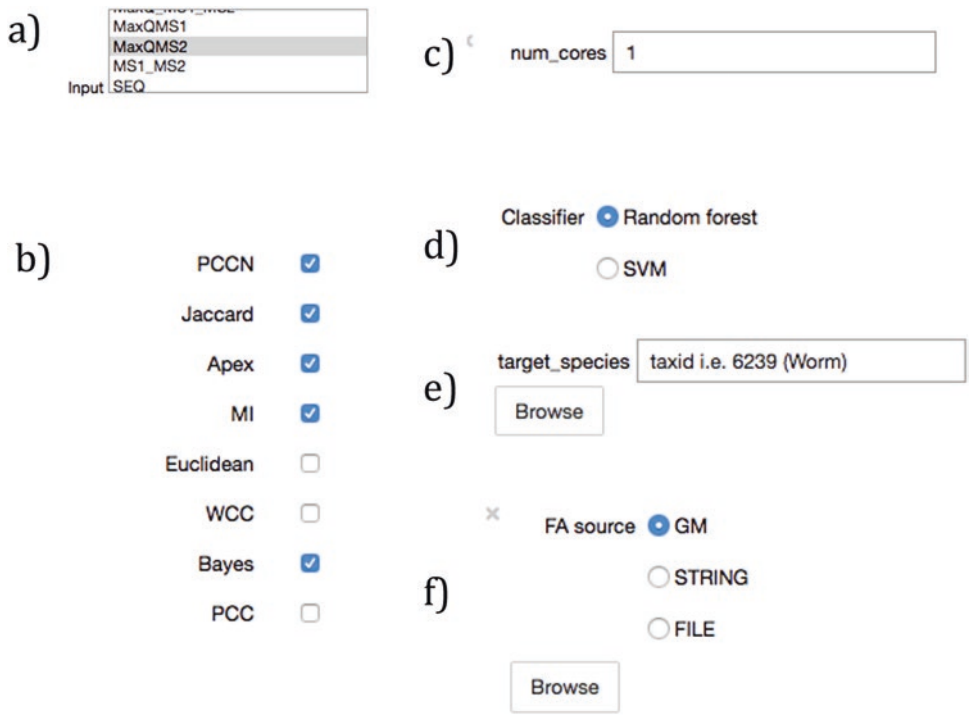

Fig. 1 Overview of different widgets for configuring the EPIC Jupyter notebook, they show options for selecting: (**a**) input data directory, (**b**) co-elution scores, (**c**) number of cores, (**d**) machine learning classifier, (**e**) reference data, and (**f**) functional annotation

3.3.1 Selecting Input Features

There are eight protein similarity score features available in total (Fig. 1b): Pearson with Poisson noise (PCCN), Jaccard, Apex, Mutual Information (MI), Euclidean, weighted cross correlation (WCC), Bayes correlation, and Pearson correlation coefficient. A short description for each feature follows below (citations provided as needed):

PCCN: To reduce the spurious correlations caused by fractions with low peptide counts, PCCN correlation is calculated by averaging multiple Pearson correlation values that are computed by taking the raw counts and adding a round of small value Poisson noise to them [2].

Jaccard: Determines co-elution based on the number of overlapping fractions two proteins are detected together in.
Apex: This score is one when the largest or peak signal (highest spectral count) of each of two proteins occurs in the same fraction together or else zero if otherwise [2].

Mutual Information: The mutual dependence between two variables (e.g., protein spectral counts) is used to identify statically significant protein pairs.

Euclidean: The Pythagorean theorem is used to calculate the Euclidean distance between two proteins by considering each fraction as an independent dimension.

WCC: A weighted correlation-based metric takes into account small possible shifts in the patterns of two proteins that co-fractionate together [8].

Bayes: Bayes correlation identifies statistical significant protein correlations [9].

PCC: Pearson correlation coefficient of two proteins calculated based on their respective co-elution profile patterns.

3.3.2 Number of Cores

Increasing the number of computer cores (if available) will greatly reduce the runtime of EPIC (Fig. 1c).

3.3.3 Machine Learning Classifier

Currently supported options are support vector machine [10] and random forest classifiers [11]; we recommend initially using the random forest classifier (Fig. 1d).

3.3.4 Reference Data

The user can either have reference complexes automatically generated from CORUM, GO, and IntAct by supplying a valid taxonomic (species) ID (taxid) (Fig. 1e) or supply a custom set of reference protein assemblies (*see* Subheading 2.3).

3.3.5 Functional Annotation Data

Analogous to the reference data, the user can either have it automatically obtained using EPIC (Fig. 1f) or by supplying custom data (*see* Subheading 2.3). For automatic generation the user can select either to use STRING (https://string-db.org/) [12] or GeneMANIA (http://genemania.org/) [13] as source. When using STRING we exclude "experimental," "database," and "combined_score" scores from the database to avoid circular reasoning in the training phase. We recommend using GeneMANIA if the target species is available in both databases, since we observed better performance for predicting Worm protein complexes when using GeneMANIA.

Once this step is completed, the user can either run the script cell by cell (pressing run cell and select next, i.e., play button) or run the entire EPIC script by selecting run cell and below. When running for the first time, we recommend to run cell by cell, so the user can check the output for each cell, and for repeated reruns the user can select "run cell and below" to run all cells automatically without human supervision.

3.4 EPIC Output

Once the Jupyter script is completed, it will generate an initial graphical overview of the generated protein clusters using Cytoscape.js in its second to last step. At the end, EPIC will generate an output folder with various result files in a specified input directory named My_EPIC_PROJ_out, including the following files:

1. Out.scores.txt: Raw co-elution scores for all candidate PPIs.
2. Out.roc.png – precision-recall curve for predicted PPIs [14].

3. Out.pr.png – receiver operating characteristic (ROC) curve for predicted PPIs [15].
4. Out.rf.cutoff.png – shows precision and recall values across all confidence cut-off values.
5. Out.pred.txt – predicted protein interactions with classifier confidence values.
6. Out.clust.txt – predicted multi-protein clusters.

The Out.scores.txt contains the features used for predicting the protein associations, while the Out.roc.png, Out.pr.png, and Out.rf.cutoff.png files give an overview of the classifiers performance (*see* **Note 6**). The Out.pred.txt and Out.clust.txt contain the main predicted outputs (PPI and clusters) generated by EPIC.

3.4.1 EPIC with Cytoscape

The last cell of the Jupyter script is used to visualize the generated protein clusters using Cytoscape. This step is optional but recommended.

1. Start the locally installed Cytoscape on your machine. This is done outside of the Jupyter script.
2. In Cytoscape, select Apps, and then select app manager.
3. In the search mask, enter clusterMaker2, and select clusterMaker2 from the selection, and press install. This step needs only to be performed once.
4. Switch back to the Jupyter script, and run the last cell, followed by switching back to Cytoscape.
5. In Cytoscape, select Layout, followed by yFiles Layout, and finally select organic. Now there should be one group of nodes (proteins) per cluster showing all associated interactions.
6. Use the mouse to select a cluster, and then select Apps and clusterMaker visualizations, and finally select JTree HeatMapView.
7. In the "Node attributes for cluster" field, select all the fractions that are displayed. Check the "use only selected nodes/edges for clusters" box, and press the OK button.

4 Notes

1. The central component for improving the runtime of EPIC is assigning it more cores if available. It is **important to assign the number of cores to the docker engine** so that EPIC can use those cores. For most normal use cases where you have 4–5 experiments (around 1000 fractions), EPIC can completely run between a night and an afternoon.
2. The main advantage of using Cytoscape with EPIC is visualizing both the network of protein complexes and PPIs that are generated, as well as the supporting co-fractionation data for

each putative protein member in heat-map format to confirm profile similarity. Each edge in the Cytoscape network provides the EPIC derived confidence score, and the user can adjust edge thickness (cutoff values) to define data consistency within a cluster. No prior knowledge of Cytoscape is required; however, it is encouraged to become familiar with network style and layout formats (see http://wiki.cytoscape.org/Cytoscape_User_Manual#Visual_Styles and http://wiki.cytoscape.org/Cytoscape_User_Manual/Navigation_Layout).

3. When supplying a custom set of reference complexes, there are certain aspects the user needs to be aware of. First, the automatically generated reference set is based on experimentally inferred complexes retrieved from the CORUM, IntAct, and GO curation databases, so if the user wants to use a custom set, it is recommended to use different sources. The most important thing to be aware of is to refrain using complexes derived from functional genomics, since using this resource will result in circular reasoning because EPIC uses functional-based features for boosting PPI scores. In case the user wants to use complexes derived from functional annotation, then it is recommended to run EPIC using only experimental evidences. Also, we liked to note when generating the reference set, the user should not use complexes derived using non-biochemically based experimental methods (e.g., yeast two hybrid assays) because these tend to overlap poorly with biochemical data (e.g., co-fractionation). In brief, we highly recommend users to generate their reference complexes using complexes that are manually curated and were verified by low-throughput experimental methods.
4. When deciding which functional associations to use for enhancing learning/scoring, we typically observed best performance using species-specific and tissue-specific data (when available). For example, when predicting complex membership by co-fractionation analysis of *H. sapiens*, *C. elegans*, or *M. musculus* protein extracts, we observed optimal performance using supporting functional associations from HumanNet (http://www.functionalnet.org/humannet/about.html) [16], WormNet (http://www.functionalnet.org/wormnet/) [17], and MouseNet (http://www.functionalnet.org/mousenet/) [18], respectively. If wanting to combine multiple resources to boost prediction confidence, the user needs to combine these data into one single functional annotation file; public data integration tools like GeneMANIA (http://genemania.org/) [13] facilitate this.
5. A user can select any folder as an input folder; however, it is highly recommended to create individual project folders within EPIC.

6. Precision-recall curves provide an overview of classifier performance indicating how many protein associations can be classified with a certain precision. Receiver operating characteristic (ROC) curves indicate how well the classifier can distinguish false-positive from false-negative interactions. The precision-recall plot for various classifier confidence values is intuitive, since it shows the precision (i.e., relative fraction of predicted interactions that are correctly classified) and recall (i.e., relative fraction of positive interactions that are correctly classified) across all possible classifier confidence cutoff values.

References

1. Lucas Hu Ming FG, Cuihong Wan, Gary Bader, Andrew Emili (2018) EPIC: elution profile-based inference of protein complex membership. Under revision.
2. Havugimana PC et al (2012) A census of human soluble protein complexes. Cell 150(5):1068–1081
3. Wan C et al (2015) Panorama of ancient metazoan macromolecular complexes. Nature 525(7569):339–344
4. Shannon P et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504
5. Ruepp A et al (2010) CORUM: the comprehensive resource of mammalian protein complexes—2009. Nucleic Acids Res 38(suppl 1):D497–D501
6. Kerrien S et al (2012) The IntAct molecular interaction database in 2012. Nucleic Acids Res 40(D1):D841–D846
7. Gene Ontology C (2015) Gene ontology consortium: going forward. Nucleic Acids Res 43(Database issue):D1049–D1056
8. Wehrens, R. and M.R. Wehrens, Package 'wccsom'. 2015
9. Sánchez-Taltavull D et al (2016) Bayesian correlation analysis for sequence count data. PLoS One 11(10):e0163595
10. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
11. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
12. Szklarczyk D et al (2017) The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic Acids Res 45(D1):D362–D368
13. Warde-Farley D et al (2010) The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function. Nucleic Acids Res 38(suppl_2):W214–W220
14. Davis J and Goadrich M 2006. The relationship between precision-recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning. ACM
15. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143(1):29–36
16. Lee I et al (2011) Prioritizing candidate disease genes by network-based boosting of genome-wide association data. Genome Res 21(7):1109–1121
17. Lee I et al (2010) Predicting genetic modifier loci using functional gene networks. Genome Res 20(8):1143–1153
18. Kim WK, Krumpelman C, Marcotte EM (2008) Inferring mouse gene functions from genomic-scale data using a combined functional network/classification strategy. Genome Biol 9(1):S5