PEPN-GRN : Documentation

Deepika Vatsa

April 2021

Author: Deepika Vatsa, vatsa.deepika.email@gmail.com

This is the documentation for the three variants of the PEPN-GRN method namely: PEPN-GRN_v1, PEPN-GRN_v2, and PEPN-GRN_v3.

The PEPN-GRN method is described in the following paper:

Vatsa D, Agarwal S (2021) PEPN-GRN: A Petri net-based approach for the inference of gene regulatory networks from noisy gene expression data. *PLOS ONE* 16(5): e0251666. `https://doi.org/10.1371/journal.pone.0251666`

1 Software

To be able to run PEPN-GRN code, you have to install MATLAB R2015a or above.

2 Data set

The PEPN-GRN method is designed to run on discretized (2bin or 3bin) time series expression data sets. 3-bin discretized DREAM4 time series data sets of five 10-gene and five 100-gene networks are provided in this package under “multi-bin-disc-dream4-data-repository” folder. For each network, discretized data is provided for three discretization methods namely: Equal Frequency Discretization (EFD), Equal Width Discretization (EWD), and Kmeans discretization method.

- **Data files:** Data matrices containing collated data of 5 time series of 10-gene networks and 10 time series of 100-gene networks are provided. In data matrix, rows represent genes and columns represent time points.

  Data files are named as:

  “all_data_*ngenes*_*netnum*_*disc-method*_*disc-level*.mat”

  where ***ngenes*** is number of genes, ***netnum*** is network number, ***disc-method*** is discretization method and ***disc-level*** is discretization level.

  For instance, “all_data_10_1_efd_3bin.mat” is the 10x105 data matrix containing collated data of five time series of 10 genes of the 1^{st} 10-gene DREAM4 network discretized using EFD 3bin method. The data file can be found at path:

  multi-bin-disc-dream4-data-repository > dream4_10_1 > efd >
  all_data_10_1_efd_3bin.mat

- **Ground truth:** Ground truth of the networks are provided.

  "all_pos_edge_*ngenes_netnum*.mat" contains indexes of positive edges (edges present) of the network. For instance, "all_pos_edge_10_1.mat" contains indexes of positive edges in the ground truth of 1^{st} 10-gene DREAM4 network.

  Similarly, **"all_neg_edge_*ngenes_netnum*.mat"** contains indexes of negative edges (edges absent) of the network.

  "groundtruth_edges_signed_*ngenes_netnum*.mat" contains ground truth edges with regulation sign in the form of *Regulator*, *Target*, *Sign*.

- **Time points:** Cell array containing the time points of the time series experiments of each network. Files are named as:

  "all_experiment_*ngenes_netnum*.mat"

  For 10-gene networks, cell array is of size 1x5 and for 100-gene networks, the size is 1x10.

  For instance, "all_experiment_10_1.mat" is 1x5 cell array containing time points of 5 time series of 1^{st} 10-gene DREAM4 network. Similarly, "all_experiment_100_1.mat" is 1x10 cell array containing time points of 10 time series of 1^{st} 100-gene DREAM4 network.

- **Gene names:** Cell array containing the names of the genes. Files are named as:

  genenames_*ngenes*gene.mat

  For instance, "genenames_10gene.mat" contains gene names of 10-gene networks.

3 Run PEPN-GRN

To run variants of the PEPN-GRN method on discretized DREAM4 time series data sets, run **"run_pepn-grn.sh"** on terminal. By default, this script runs the code for PEPN-GRN_v1 implementation. To run other variants, set the parameter "*variant*" to 2 for PEPN-GRN_v2 and 3 for PEPN-GRN_v3 respectively in the shell script.

Upon running the shell script, a **"PEPN-GRN_v1-results"** folder will be created containing results for each discretized network. Predicted edges are returned in **"predicted_edges_*ngenes_netnum*.txt"** file in the form: {*regulator*, *target*, *probability score*}. ROC and PR plots are also generated as **"rocplot_*ngenes_netnum*.eps"** and **"prplot_*ngenes_netnum*.eps"** highlighting the AUROC and AUPR values.

Source files for the three variants of the PEPN-GRN method are contained in *source_v1*, *source_v2*, and *source_v3* folders, respectively. *code_v1*, *code_v2*, and *code_v3* are the main source code files for each variant.

Common function files in *source_v1*, *source_v2*, and *source_v3* folders are:

- *prod_evidence.m*: contains code to identify edges from a state pair using logical rules for production evidence type.

- *decay_evidence.m*: contains code to identify edges from a state pair using logical rules for decay evidence type.

- *sus_prod_evidence.m*: contains code to identify edges from a state pair using logical rules for sustained production evidence type.

- *sus_decay_evidence.m*: contains code to identify edges from a state pair using logical rules for sustained decay evidence type.

- *rankwise_roc_pr_plot.m*: contains code to generate ROC and PR plots.

Score computing function file

In *source_v1*, “unwt_edge_prob.m” computes unweighted edge probability as score of each edge by taking the average of four evidence probabilities.

In *source_v2*, “weighted_edge_prob.m” computes weighted aggregation of edge probabilities as score of each edge.

In *source_v3*, score of an edge is probability value computed using Logistic regression using edge features (edge evidence probabilities).

Setting predefined candidate regulator genes

Indexes of candidate regulator genes can be specified in the variable “*regulators*” in *code_v1.m* (or *code_v2.m* or *code_v3.m*) file.

Restricting number of regulator genes

We can restrict the number of regulator genes of a target gene in the output by setting the variable *nreg* in *code_v1.m* (or *code_v2.m* or *code_v3.m*) file. So, setting it to 3, lets say, will produce results selecting top 3 regulator genes for each target gene. This is particularly helpful in case of small data sets where false positive rate is high.