TimeReg manual

Zhana Duren
durenzn@gmail.com
Apr 20, 2020

Contents

1. Getting started.

1.1 About TimeReg

Time course experiment is a widely used design in the study of cellular processes such as differentiation or response to stimuli. We propose TimeReg (Time Course Regulatory Analysis) as a method for the analysis of gene regulatory networks based on paired gene expression and chromatin accessibility data from the time course. TimeReg can be used to prioritize regulatory elements, to extract core regulatory modules at each time point, to identify key regulators driving changes of the cellular state, and to causally connect the modules across different time points.

TimeReg is a computational tool for gene regulatory analysis from time course paired gene expression and chromatin accessibility data. If you use TimeReg software, please cite following two papers:

Duren, Zhana, et al. "Modeling gene regulation from paired expression and chromatin accessibility data." Proceedings of the National Academy of Sciences 114.25 (2017): E4914-E4923.

Duren, Zhana, et al. "Time course regulatory analysis based on paired expression and chromatin accessibility data." Genome Research (2020): gr-257063.

1.2 Installation

TimeReg software source code (matlab based) can be downloaded from Github: https://github.com/SUwonglab/TimeReg.

To run TimeReg, you need to install and run PECA2 first.

Download and install PECA on Linux:

wget https://github.com/SUwonglab/PECA/archive/master.zip

unzip master.zip

cd PECA-master/

bash install.sh

Download and install TimeReg on Linux:

wget https://github.com/SUwonglab/TimeReg/archive/master.zip

unzip master.zip

cd TimeReg-master/

bash install.sh

2. Pre-processing data

To run TimeReg tool, you need to do following two steps: i) prepare input data and ii) do TimeReg analysis.

2.1 Prepare input data

To run TimeReg, you need to run PECA2 on each time point first. If you have multiple replicates, please merge them first to get one bam file and one expression profile on each time point. For chromatin accessibility data, you may merge the bam files of the replicates. For gene expression data, you may use the average expression (FPKM or TPM) of the replicates. After you have three files (sample.txt, sample.bam, sample.bam.bai), you can run PECA2 and get the TRS matrix (TFTG_score.txt).

3. Time Course Regulatory Analysis

3.1 Input of TimeReg

After the input data is prepared, do TimeReg analysis by following two steps: 1) edit config file, and 2) run TimeReg.

To prepare config file, please see following example and description:

```
PECA_Module_dir = './.';
Expfile='./exampleData/Exp.txt';
sample_trs_files='./exampleData/sample_trs_info1';
TFName_file='./exampleData/TFName.txt';
TGName_file='./exampleData/TGName.txt';
Outdir='./exampleData/out';
pre_process_required=0;
species = 'mouse';
lambda=0.2;
K=[1,3,3,4,4];

%input format (tab delimited), here T is the number of time points
    %1, Expfile: name (location) of the expression file. It is a T+1 columns
file, first column is the gene symbol, 2 to T+1 columns are expression value
(FPKM or TPM),
    %2, sample_trs_files: T rows 2 column file, first column is the sample
name (no space allowed), second column is the name of the TRS file from PECA2
(no space allowed).
    %example: mESC  ./PECA2/Results/mESC/TFTG_score.txt
    %3, TFName_file: file name of the TF Names from PECA2
    %4, TGName_file: file name of the TG Names from PECA2
    %5, out_folder: name of output folder
    %6, pre_process_required: 0 or 1. If you require preprocess, set this
variavble 1, othervise 0. pre-process will remove some genes with constant
expression.
    %7, species: 'mouse' or 'human'.
    %8, lambda: weight of correlation from your time course data (lambda)
verse public data (1-lambda), it is continious value between 0 and 1.
    %9, K: number of subpopulation on each time point, which is a 1*T vector.
```

After prepare the config files, you can run TimeReg by following scripts:

cat Your_config_file TCRA.m > run_TCRA.m
matlab -nodisplay -nosplash -nodesktop -r "run_TCRA; exit"

3.2 Output of TimeReg

On each time point, TimeReg will output following 6 types of files: 1) ${sample}_module${i}_Target.txt, 2) ${sample}_module${i}_TF.txt, 3) ${sample}_TF_Specific.png, 4) ${sample}_TF_TG_heatmap.png, 5) ${sample}_module${i}_DriverTF.txt, and 6) TimeCourse_ancestor-descendant_mapping.txt. Please see the description of these files one by one.

${sample}_module${i}_Target.txt: module specific target gene information file. By comparing the normalized TRS of in-module TFs and out-module TFs, we get a p-value by two-sample one tailed t-test. The third column FoldChange is mean(in-module TFs' TRS)/mean(out-module TFs' TRS). The forth column is the expression level, and the fifth column is gene specificity score which is defined as product of TF expression, FoldChange, and –log10(P-value).

${sample}_module${i}_TF.txt: module specific TF information file, which has similar format with ${sample}_module${i}_Target.txt.

${sample}_TF_Specific.png: list of top 30 module specific TFs in the figure format.

${sample}_TF_TG_heatmap.png: heatmap of normalized TF-TG TRS matrix.

${sample}_module${i}_DriverTF.txt. First column is the TF name and the second column is the driver TF p-value.

TimeCourse_ancestor-descendant_mapping.txt: ancestor-descendant mapping file. The first column represent ancestor and the second column represent descendant.

3.3 About the pre-processing

If you set the pre_process_required parameters as 1, it will calculate the gene specificity score on each time point and select the top 3000 high specificity score genes to do further analysis. The rank of high specificity score genes on each time point will be in file specific_list.txt. The ranking will provide rich information on cellular context characteristics. You may select top (i.e. 200 or 500) genes to do GO enrichment analysis.