Instructions for Computer Exercise 4

Systems Biology 2019

11/12/2019

Intended Learning Outcomes for this lab exercise

After this session, you will be able to use specific software tools to investigate the mRNA and protein dependence at the molecular level during a given time interval. You will be able to examine the gene-specific expression in a system using rate-ratios. You will know how to identify and differentiate between genes that are regulated at the transcript or at the protein level.

Introduction

The software PECAplus was developed by Christine Vogel and Hyungwon Choi (Ref 1). This software toolbox provides us with a robust statistical platform that can be used to analyze multi-omics data from transcriptomics and proteomics experiments. The software is comparing gene expression at different levels (mRNA and protein) and how they change in relation to each other over time.

This computer exercise has been designed to let you explore one dataset containing *paired* mRNA and protein measurements. The original set includes 2,131 genes with missing observations allowed at up to two-time points within each replicate. Today, we will focus on a subset consisting of only 1,500 genes, selected specifically for this lab exercise. This dataset is derived from an ER stress experiment performed by Cheng et al. (Ref 2). Briefly, this experiment investigates HeLa cells collected at eight different time points (0, 0.5, 1, 2, 8, 16, 24 and 30h) following treatment with DTT. *Note, DTT is a potent reducing agent and is often used to promote reductive stress.*

The mRNA and protein regulation and its correlation is still one area of great debate within the field of *integrative omics*. There is still no consensus regarding the relationship between mRNA and protein levels in human cells, tissues or organs. As preparatory work, you have been asked to read through the article "PECA: A Novel Statistical Tool for Deconvoluting Time-Dependent Gene Expression Regulation discussing concepts of RNA and protein expressed" (Ref 3), and this software package will be the foundation for this lab exercise. *Note. We will only use the PECAcore package and not the PECAr nor the PECAps.*

PECAcore performs statistical inference on the protein synthesis rate over the degradation rate for individual genes across time. The algorithm identifies significant change points at either the protein level or at the mRNA level without the need for absolute data. To perform this type of analysis, you need two paired datasets covering the same set of genes — one set must reflect mRNA levels (transcriptomics) and one set must be a measurement of the protein translated from the corresponding transcript (proteomics). This allow us to calculate rate-ratios that are independent on the actual expression level.

Briefly, as mentioned in the lecture series, transcriptomics is used to study an organism's transcriptome, which corresponds to the sum of all of its mRNA transcripts. Key concepts to consider when comparing mRNA and protein expression is, of course, if you're observing a static (steady-state) or perturbed system (dynamic). The basic part of the lab consists of two different stages that you have to complete to pass the lab course and be awarded the first set of bonus points. First, you will run the PECAcore alogrithm and, thereafter, evaluate the processed data files. To fulfill the extra lab exercise criteria, you are asked to process the datasets further and evaluate the result through a Gene Set Analysis (GSA). (Note. This part is highlighted as *Extra*.)

The PECA algorithm takes approximately 10 minutes to finish on a regular benchtop computer. In the meanwhile, while the program is running, you will be asked to answer a set of questions about the raw-dataset itself in order to assess the quality. We will examine the dataset both before and after the PECAcore algorithm is applied, and you will be asked to reflect on its effect. You will be encouraged to re-use and

modify code from this course published to Canvas from other lab exercises. *Note that you are free to use any program of your choice (python, R, matlab etc.).* Previous modules of this course have covered all the parts necessary for the analysis and visualization. You must be able to handle necessary visualization tools such as scatter plot, linear regression, pearson correlation, cluster dendrograms and heatmaps. You are also encouraged to use external sources in order to visualize and investigate the data if necessary.

PECAplus itself performs most of the analytical and statistical work, which means that the coding part of this lab exercise is mostly data visualization using different tools. The assessment part of this lab and its questions have been designed to make sure that you understand the key concepts of gene regulation and we anticipate that you can answer each question with sufficient detail.

Preparations

You will need access to the PECAplus software (Spelhallen / GitHub) and the lab related input-files (Canvas/Modules/Module 3/Lab 4/lab2019_4). You can either use PECAplus in Spelhallen or install it on your personal computer from GitHub.

Spelhallen

The PECAplus program is pre-installed in Spelhallen. You can find the program directory in the folder below.

```
/opt/pecaplus/2019-11-11/
```

Copy this folder to your home environment as you lack permission to write any files to the original directory (global folder).

GitHub

Follow the instructions on GitHub and download and compile the program on your computer. You can also find the PECAplus wiki useful, which you can find by following this link https://github.com/PECAplus/PECAplus_cmd_line/wiki.

Download the lab folder from Canvas

You must download the 'lab4_2019' folder from Canvas and place it into the PECAplus-master folder (GitHub installation) or in the 2019-11-11 folder (Spelhallen), which now should be located in your home environment. The lab4_2019 folder includes all the files necessary for you to complete this lab exercise.

1. Make sure that you have the peca_core_bin program-file located in the PECAplus-master/peca_core/ folder ((personal computer (GitHub Installation)) or in the 2019-11-11/peca_core/ folder (Spelhallen).

You must set the working directory to the folder where you have all the data (i.e. the 'input_params.txt'-file.), and you will have to call the program file from there. Therefore, make the 'lab4_2019/peca_rna-prot' folder to your working directory. *i.e. replace ~ to match your folder structure.*

```
cd ~/lab4_2019/peca_rna-prot/
```

Input Parameters

The initial part of this lab will focus on two main things. First, we will explore the reproducibility of data obtained from the transcriptomics and proteomics analysis, respectively. Second, we will combine and evaluate

these files using PECAcore, performing a multi-omics analysis and investigate the dynamics behind protein and mRNA regulation. At the same time, we will also compare the mRNA regulation itself based on the perturbed system. PECAplus implements the core functionality for analyzing a two-level time series data set (e.g., paired protein and mRNA concentration data or paired DNA and mRNA concentration data). It identifies significantly regulated genes at each time point with probability scores of significant change points.

1. Open the file input_params.txt that we will use for this exercise. It should look like this:

```
## using "peca_core" executables

## The two files are RNA and protein input files. Both files comtain 1500 genes.
## In order to compare the transcript specific regulation. Keep FILE_X empty and
## add the mrna_1500_lab.txt file as input to FILE_Y.
FILE_X = mrna_1500_lab.txt 0
FILE_Y = prot_1500_lab.txt 0

## Indicate there are two replicates in this data set
N_REP = 2

## We are purposefully giving an equally spaced time point for PECA modeling for a reason.
## When you have irregularly spaced time intervals, the rate ratios can be
## inaccurately estimated for larger intervals (30 min interval versus 8 hour interval)
TIME = 0 1 2 3 4 5 6 7

## MCMC parameters

N_BURN = 1000
N_THIN = 10
N_SAMPLE = 1000

## The Gaussian smoothing parameters.
## The first value indicates how smooth the curve should be;
## a small value will yield a smoother curve (smoother),
## a large value will yield a curve interpolating the data closely (less smooth).
## The second value will cause it to shrink towards the prior mean.
SMOOTHING= 2.0 1.0

## The file that gives the network information.
## A list of undirected edges.
MODULE= 9606.protein.actions.v10.5_ensg.txt
```

FILE_X = mRNA expression matrix, log2 normalized mRNA expression. Append 0 after a whitespace to switch off log-transformation

Note. RNA 'expression' data can be normalized read counts, e.g. FPKM, RPKM, or TPM, from an RNA seq experiment, or signal intensities from a single channel microarray experiment, or actual concentration (if known). In this lab, we are using microarray data.

FILE_Y = Protein expression matrix, log2 normalized protein expression. Append 0 after a whitespace to switch off log-transformation

Note. Proteomics is the large-scale study of proteins and proteomics gives a different level of understanding than transcriptomics. In contrast to mRNA that may be produced in abundance and often degraded rapidly or translated inefficiently, resulting in a small amount of protein. Many proteins experience post-translational modifications that profoundly affect their activities; for example, some proteins are not active until they become phosphorylated. Reproducibility. One major factor affecting reproducibility in proteomics experiments is the simultaneous elution of many more peptides than mass spectrometers can measure. This causes stochastic

differences between experiments due to data-dependent acquisition of tryptic peptides.

Gaussian Process / Variance Parameter

Determines the variation of values from the mean (default 2.0). A small value will result in the function values changing quickly. Scaling factor that determines the smoothness of the curve (default 1.0). A small value will result in a function that stays close to the mean value.

MCMC Parameters

The PECA model parameters are estimated using a sampling-based algorithm called MCMC (Markov chain Monte Carlo), which requires the parameters below. All values should be positive integers.

MCMC Burn-In defines the iterations to be thrown away at the beginning of MCMC run, i.e. the burn-in period (default: 1000). MCMC Thinning defines the interval in which iterations of MCMC are recorded (default: 10). MCMC Samples defines the total of number of post-burn-in samples to be recorded from MCMC (default: 1000). *Note. We will not focus on the MCMC module in today's exercise, but you should be aware that it is part of the PECAplus workflow.*

1. PECAcore: Proteins regulated at the transcript level

to the folder where we have the input file. Then, let's execute the first set of parameters using the input parameter. Please feel free to continue in the *Data evaluation and quality control* section below.

```
../../peca_core/peca_core_bin input_params.txt
```

Now, let's visualize the mRNA and protein regulation usinga python-script below. Execute the command below to create the output pdf.

Note. If you cannot execute this, please go to canvas and download the pdf-file from the modules section. This will not affect your computer lab or your ability to finish it, but you will need this pdf to successfully complete this lab exercise as questions will be asked on some examples below.

```
python3 peca_core.py
```

Hint. Detailed instructions for the figures can be found in Fig.1 in the 'npj Syst Biol Appl 2017 Teo.pdf' posted to Canvas.

2. PECAcore: mRNA-level regulation only

For a strict transcript-centric analysis, i.e. identifying transcripts that are turned on or off post DTT treatment. You can place the mRNA data as FILE_Y, and leave FILE_X empty. This has already been updated in the input_params_rna-only.txt file in the peca_rna folder. In this case, the expression series in FILE_Y is assumed to be DNA, which has concentration with one as values for all genes.

```
cd PECAplus_cmd_line-master/lab2019/peca_rna
../../peca_core/peca_core_bin input_params_rna-only.txt
```

```
python3 peca_core.py
```

Data evaluation and quality control

The dataset that you have been given is a subfraction of the original dataset and contains paired mRNA and protein data. The RNA has been determined by microarray technology and the intensities have been normalized and log2 transformed. The proteomics dataset originally consisted of a total of >3,200 proteins,

of which 2,130 mapped to the RNA data. To derive this high-confidence dataset, all genes with one or more missing data points were removed, resulting in 2130 genes for further processing. This dataset was normalized by the sum of all label-free quantification (LFQ) intensities.

This tutorial will be done in R, but you are free to use any tool to filter data and generate the plots necessary to complete this lab. Many problems can even be solved in Excel, which we strongly advise you not to use (of course).

Many questions can be answered by providing tables, and you are encouraged to group multiple questions together whenever possible. Just make sure that you clearly highlight this in the final pdf.

```
#install.packages('tidyverse')
#install.packages('ggplot2')
#install.packages('readxl')
#install.packages('ggdendro')
#install.packages('gplots')

library(tidyverse)
library(ggplot2)
library(readxl)
library(ggdendro)
library(gplots)
```

```
# make sure that your working directory is set to the lab2019/peca_rna-prot
# for more infomration, type ?setwd() or use the 'Session/Set Working Directory'-tab

# first, read the original dataset into the R environment
df1 <- as_tibble(read.table("RS_mrna_1500_lab.txt", header = T))
df2 <- as_tibble(read.table("RS_prot_1500_lab.txt", header = T))
```

The first column indicate the gene ID given in the *ENSG*-format. Each dataset (mRNA and protein) contain 8 timepoints with data collected at t = 0, 0.5, 1, 2, 8, 16 and 30 hours. The replicate is indicated by R1 or R3. The header in the proteomics experiment is labeled LFQ (Label free quantification) with labels (1-8) representing the same timpoints as above and each replicate is labeled with R1 or R3. The ENSG is given in the *Protein.IDs* column.

Example: The protein levels in the LFQ.intensity.8__30h__RS1 column are paired with mRNA levels in the R1t30 column.

1. What is the dynamic range (max/min) in each replicate for all transcripts (mRNA, given as intensity)?
2. Another alternative to use array technologies would be to use RNAseq, which is a very quantitative method for transcriptome analysis. Why don't we need any absolute values when running the PECA software?
3. What is the dynamic range (max/min) in each replicate for all proteins (given as LFQ/intensity)?
4. Which one of the two datasets is the most robust one? Justify your selection by plotting the biological replicates against against each other at t0.

Example, mRNA, y-axis = yR1t0, x-axis = R3t0) Example, proptein, y-axis = yR1t0, x-axis = R3t0)

```
df1 %>%
  ggplot(aes(x =R1t0, y = R3t0)) + geom_point()

df2 %>%
  ggplot(aes(x = LFQ.intensity.1_0h_RS1, y = LFQ.intensity.1_0h_RS3 )) + geom_point()
```

4. What is the median residual error for each method (Transcriptomics and Proteomics) respectively when comparing biological replicates? Provide answer as one table.

Hint. Compare replicates at each timepoint for the RNA and protein datasets respectively. Examples for t0 given below

```
summary(lm(df1$R1t0 ~df1$R3t0))
summary(lm(df2$LFQ.intensity.1_0h_RS1 ~df2$LFQ.intensity.1_0h_RS3))
```

5. From the previous section, what is the correlation (pearson's r, spearman rho) between biological replicates. Provide a table for each sample type and replicate. Why cannot we compare replicates for the proteomics dataset in its existing form?

Hint. Compare replicates at each timepoint for the RNA and protein datasets respectively. Examples for t0 given below

```
round(cor(df1$R1t0, df1$R3t0, method = "spearman"), 3)
round(cor(df1$R1t0, df1$R3t0, method = "pearson"), 3)
cor(df2$LFQ.intensity.1_0h_RS1, df2$LFQ.intensity.1_0h_RS3, method = "pearson")
cor(df2$LFQ.intensity.1_0h_RS1, df2$LFQ.intensity.1_0h_RS3, method = "spearman")
```

6. How many missing datapoints can you identify in each proteomics replicate, add this to the table above.

7.You can also exclude missing datapoints from the analysis. How many genes would be be able to use if we drop all genes with missing data?

```
df2_complete <- df2 %>%
  drop_na()
round(cor(df2_complete$LFQ.intensity.1_0h_RS1, df2_complete$LFQ.intensity.1_0h_RS3, method = "pearson")
round(cor(df2_complete$LFQ.intensity.1_0h_RS1, df2_complete$LFQ.intensity.1_0h_RS3, method = "spearman"
```

Great, the analysis performed by PECAcore should be done by now. Let us continue with the post-processing part of this lab exercise.

PECAcore: Post-processing

We will now look closer at what effects the PECAcore algorithm has on the dataset, both with regards to data imputation and smoothing.

General output

Let's have a look at the output file generated by the PECA algorithm. The most crucial result file is named *data_R_CPS.txt* and should be placed in the same folder as the input parameter file. You shall have two different result files, one for the RNA analysis only, and one for the combined mulit omics analysis studying protein regulation based on transcript levels.

*Note. The output file may need to be fixed; some operating systems cannot distinguish the header, which results in a frameshift. The first column should be the gene name, followed by the main/expression columns.

The other numeric columns contain RY, signedCPSX, FDRX, where X indexes time point (i.e. X=1 refers to the second time point) and Y indexes time point interval starting from 0 (i.e. Y=0 refers to the range between the first and second time points).

RY is the rate ratio for the time interval preceding the specified time point (e.g. if Y = 1, then the range is between time point indices 1 and 2). *signedCPSX* is the change point score with signs indicating up/downregulation. A positive sign describes upregulation; a negative sign down regulation. *FDRX* is the False Discovery Rate

Note. The rate-ratio itself does NOT inform on the significance or direction of the change. The DIFFERENCE between consecutive rate ratios (adjacent time intervals) describes the direction of change, and the FDR the significance.

Gene regulation

Define a cutoff level for the CPS with an FDR at 5%. Calculate how many genes that are significantly detected by peca_core and significantly changing between each time point. You will have to use

Please provide your answer to question 1 and 2 as a table.

1. How many proteins are significantly regulated at the transcript level for each time interval?

Hint. Filter FDR columns in the data_R_CPS.txt file using 0.05 (5%) as cutoff. Look at the signedCPSX column that indicate if they are up-regulated (positive value) or down-regulated (negative value).

```
# Copy and paste the output from PECAcore to your working environment.
# This should be the paired mRNA and protein dataset.
df3<- as_tibble(read.table("data_R_CPS.txt", header = T))

df3 %>%
  filter(FDR5 < 0.05) %>%
  summarize(pos = sum(signedCPS5 > 0), neg = sum(signedCPS5 < 0))
```

2. Is this different if you consider the result file for the transcript analysis (mRNA only)?

Hint. Make the same comparision with the 'mRNA-level regulation only'-output. Make sure that you rename the output file as it has the same name as the 'Proteins regulated at the transcript level'-output file.

4. Which one is the most important changepoint, i.e. when are most genes changing up or down repsectively and what time-point does this correspond to? Is it the same for both conditions (mRNA and protein vs mRNA only).

5. Identify one gene that is significantly up-regulated at the transcript level but not at the protein level. Copy and paste the graph from the output-pdf below.

Hint. This gene should be significantly regulated in the mRNA only dataset, but not in the mRNA/protein dataset

6. Identify one gene that is significantly up-regulated both at the transcript and protein level. Copy and paste the graph from the output pdf below.

Hint. This gene should be significantly regulated in the mRNA/protein dataset.

Missing data and reproducibility

Let's have a quick look at the reproducibility and how PECAplus has improved the data-quality.

1. How many missing data points do we have after applying the PECAcore algorithm?

2. How did the GP and MCMC affect the robustness between the biologial replicates? What is the correlation between biological replicates (pearson r) after smoothing? How much did it improve. Report the delta value (post-pre processing).

Re-use the code from question 4 above

3. What is the effect on the residual standard error between biologial replicates?

Re-use the code from question 4 above

Extra: Gene Set Analysis

This part is for extra bonus points only Let's continue exploring the dataset by a Gene Set Analysis (GSA). The goal with this type of experimental setup is of course to identify genes that are differentially expressed. One way of combining data is by the GSA. Following the PECAcore analysis, calculating rate-ratios, a GSA analysis can be done on the change points scores.

Move the output file called "data_R_CPS.txt" file into the peca_gsa_rna folder from the peca_rna and the peca_gsa_rna-prot in examples and execute the command below.

```
# change location to the gsa-folder for the mRNA data
cd ~/lab2019/peca_gsa_rna
../../peca_gsa/gsa_bin input_params.txt

# change location to the gsa-folder for the mRNA/protein data
cd ~/lab2019/peca_gsa_rna-prot
../../peca_gsa/gsa_bin input_params.txt
```

This will only take a couple of seconds and will generate one output file in each location named *Goterms.txt*. Load this file into your environment to construct one heatmap for each level of gene regulation (rna and rna/protein combined).

The *Goterms.txt* contains one table of p-values corresponding to the CPS in PECA analysis. Columns headers are:

MaxSig(Up): maximum of log10(- p-value) of all time points. MaxSig(Down): maximum of log10(- p-value) of all time points. Max(Both): maximum of log10(- p-value) of all time points.

GO_size: number of genes in the pathway GO_size_background: number of genes in the pathway that appears in the experimental data. Upx: p-value for enrichment based on the number of up-regulated genes Downx: p-value for enrichment based on the number of down-regulated genes Sigx: p-value for enrichment based on the number of up-regulated and down-regulated genes

Questions related to proteins regulated at the protein level using a FDR cutoff of 5%.

1. Between what two timepoints (minutes/hours after stimulation) are most transcripts regulated?
2. Between what two timepoints are most proteins upregulated based on transcriptional regulation?
3. In what time interval are most proteins significantly downregulated based on decreased mRNA expression?
4. Select one gene cluster from the GSA analysis that are significantly regulated at the transcript level, which you belive is relevant for this experimental setup (Remeber: DTT treatment and induced reductive stress). Investigate the protein expression profiles in relation to the corresponding transcript (visualized in the PDF output). Please copy one (or multiple) gene-plotsthat you think is relevant and provide a short analysis based on the regulation that we observe (200 words).

You shall address the following questions in your extended analysis. 1. What are the genes biological function? *Hint. Visit uniprot.org or proteinatlas.org* 2. What can be the explanation for the regulatory effect that we see? 3. Can you explain this based on the experimental condition? What can be the biological explanation for this observation?