Gene Expression Signatures in Blood that Predict Mortality in a West African Sepsis Cohort Define Host Response Phenotypes

AUTHOR LIST:

Josh G. Chenoweth[1qc], Carlo Colantuoni[2q], Deborah A. Striegel[1], Joost Brandsma[1], Rittal Mehta[1],Paul W. Blair[1,3], Subramaniam Krishnan[1], Pavol Genzor[1], Michael Considine[4], Leslie Cope[4],Audrey C. Knight[2], Anissa Elayadi[1], Anne Fox[5], Ronna Hertzano[6,7], Andrew G. Letizia[5], AlexOwusu-Ofori[8,9], Daniel Ansong[10,11], George Oduro[12], Kevin L. Schully[13], Danielle V. Clark[1]

[1] Austere environments Consortium for Enhanced Sepsis Outcomes (ACESO) at Henry M. Jackson Foundation, Bethesda

[2] Dept. of Neurology at Johns Hopkins School of Medicine, Baltimore

[3] Dept. of Pathology at Uniformed Services University, Bethesda

[q] Authors contributed equally to this work.

[c] Corresponding author for additional information and with questions.

Compiled by: **Pavol Genzor**

Date: **May 26th, 2023**

Summary

This document contains an example of the code used to analyze data associated with the manuscript. The "transfer learing" procedure consist of three main steps specific to each use where in general:

- 1. Expression data is subjected to negative matrix factorization using **CoGAPS** [**link**].
- 2. Public data of interest is acquired and prepared (*NOTE: To obtain public data please see original study or contact the corresponding authors.*)
  - Sepsis:
    - Reyes et.al., Nature Medicine, 2020
    - Tsalik et.al., Science Translational Medicine, 2016
    - Cazalis et.al., Intensive Care Medicine Experimental, 2014
  - COVID-19:
    - Reyes et.al., Science Translational Medicine, 2021
- 3. Decomposed data is projected onto a public data using **projectR** [**link**]
- 4. Results are plotted using preferred software.

Procedure details

Step1:

Install the **CoGAPS** package and load it into the environment. Next, load your normalized log2 transformed data into R environment (**DATi**). Make sure that data is configured in a matrix object where *rownames* correspond to genes names and *colnames* correspond to cells/samples. Use this data to generate *prcomp* object that will later be used for decomposition. Finally, load the data table will all relevant metadata information. Once data is loaded, configure the running parameters for the CoGAPS (see below exact parameters used in manuscript) and run the algorithm and perform data decomposition.

```
## Install and load required package

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("CoGAPS")

# Load the library

library("CoGAPS")

## Load the expression data

load("path/to/your/directory/your_expression_data_and_metadata.rda")

## Input should be matrix of log2 transformed data where:
## - genes as row names
## - cell/sample names as column names

DATi
cells <- colnames(DATi)
genes <- rownames(DATi)

## Table containing experiment-related metadata that will be used for plotting

sampORIG

## Run principal component analysis on the data set
## - PCA can be used in the projection in step 3

pca <- prcomp(t(DATi))
pca

## Set the analysis parameters

nThrd=1
params <- new("CogapsParams")
params <- setParam(params, "sparseOptimization", FALSE)
params <- setParam(params, "nIterations", 40000)
```

```
params <- setParam(params, "nPatterns", 30)
params <- setDistributedParams(params, nSets=1)

## Run the analysis using following parameters

xxCoGAPS <- CoGAPS(data=DATi,
                   params=params,
                   geneNames=genes,
                   sampleNames=cells,
                   messages=TRUE,
                   transposeData=FALSE,
                   nPatterns=30,
                   nIterations=40000,
                   nThreads=nThrd)

## NOTE: missing objects
#pca
#pcvars
#sampORIG
```

Step2:

Attain the public data set of interest making sure it contains all information of interest. You may need to contact authors if the information is incomplete. Format the data for use with the **projectR**. This may involve synchronizing the nomenclature, ensuring data is normalized, and most importatnly, organized into matrix with appropriate *rownames* and *colnames*.

- Data used in this manuscript:
  - Reyes et.al., Science Translational Medicine, 2021 - COVID-19 - [**link**]
  - Reyes et.al., Nature Medicine, 2020 - Sepsis - [**link**]
  - Tsalik et.al., Science Translational Medicine, 2016 - Bacterial vs Viral - [**link**]
  - Cazalis et.al., Intensive Care Medicine Experimental, 2014, - Sepsis shock - [**link**]

For each data set we used we generated following objects: **seq** - a *matrix* object of relevant expression data that has been log2 transformed; **samples** - a *vector* containing cell IDs or sample names depending on experiment; **genes** - a vector object containing gene/feature names.

```
## Load prepared public data
## - preparation entails obtaining the expression information and metadata
## - required components include the gene information, and cell/sample information
## - processed public data can be stored in R object for future use

load("BYOD/public_data_A.RData")

## Content

seq       #: expression matrix
samples   #: cell/sample -> column names
genes     #: genes -> rownames
```

```
## To coerce data.frame to matrix
## - "seq" object must be matrix NOT a data.frame

if(class(seq)=="data.frame"){
  seq=as.matrix(seq)}

## To log2 scale the raw data
## - add 1 and apply log2 normalization

if(range(seq)[2]>100){
  print("max<100 - log2'ing")
  seq=log2(seq+1)}
```

Step3:

Project your data onto the public data set using **projectR**. To get more information about configurations please reefer to the software vignette. HERE. Our sample configurations are shown below.

```
## Install an load required package

if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("projectR")

# Load the library

library(projectR)

## Extract A values from CoGAPS object
## - this was prepared in Step1 above

As = xxCoGAPS$featureLoadings


## Project first two principal components
## - "seq" corresponds to the public data set
## - "loadings" correspond to pca

PC.xproj=projectR(data=seq,
                  loadings=pca,
                  NP=1:2)

## PLOT the projections
## - can be accomplished via any method (baseR or ggplot graphics)
```

NOTES

```
## NOTE:
# -write notes here
```