Revision exercise2

Raffaele A Calogero

3/10/2019

The dataset

This dataset is made of 4 healthy donors (ctrl) and 4 neurological disease patients (neuro). From blood samples were extracted exosomal vesicles and Illumina TrueSeq whole transcriptome RNAseq was performed. The task is the identification of a singnature discriminating healthy donors from neurological patients.

Exercise 1: toy experiment with 10K reads for each sample

Please do count generation with wrapperSalmon function. Below a prototype of a wrapperSalmon function. You have to create a folder for each experiment and move the fastq files into the corresponding folder. You can do by hand of using a script like this:

```
## installing a new library to move files
install.packages("filesstrings")
library(filesstrings)

#setting the home folder as exercise1 folder
home <- getwd() #you must be in exercise1 folder

#samples folders
folders <- c("c14", "c15", "c17", "c18", "n2", "n4", "n5", "n11")
#creating folders
for(i in folders){
  dir.create(i)
}
#creating fastq names
fastqs <- paste(c("c14", "c15", "c17", "c18", "n2", "n4", "n5", "n11"), "R1.fastq.gz",sep="_")
#moving files in their folders
file.move(files=paste(home,fastqs, sep="/"), destinations =paste(home,folders, sep="/"))
```

To execute the mapping you need to use wrapperSalmon within a loop

```
library(docker4seq)
#exercise1 folder
home <- getwd()
#folders in exercise1
folders <- c("c14", "c15", "c17", "c18", "n2", "n4", "n5", "n11")

for(i in folders){
  setwd(i)
  #you need to set the transcriptome index folder and the scratch folder
  wrapperSalmon(group="docker", scratch.folder="",
        fastq.folder=getwd(), index.folder="",
        threads=12, seq.type="se", adapter5="AGATCGGAAGAGCACACGTCTGAACTCCAGTCA",
        adapter3="AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT", min.length=40, strandness="none")
  setwd(home)
}
```

Exercise2a: technical QC on real data generated using STAR/RSEM.

Please run multiQC analysis using multiQC. To run it the working folder must be exercise2. Please check the overall quality of the experiment and record the presence of outliers or “strange” behaviours.

```r
setwd("") #set exercise2 as working folder
library(docker4seq)
multiQC(group="docker", data.folder=getwd())
```

Exercise2b: aggreating counts in a counts table and evaluating experiment quality.

Aggregate samples using **sample2experiment** function Using **pca** evaluate if there are outliers.

```r
setwd("") #set exercise2 as working folder
library(docker4seq)
#folders MUST contain full path
folders <- paste(getwd(), c("c14", "c15", "c17", "c18", "n2", "n4", "n5", "n11"), sep="/")
#defining the two groups with healthy donors as Cov.1
my.covariates <- c(rep("Cov.1",4), (rep("Cov.2",4)))
sample2experiment(sample.folders=folders, covariates=my.covariates, batch = NULL,
  bio.type = c("protein_coding", "unitary_pseudogene",
  "unprocessed_pseudogene", "processed_pseudogene",
  "transcribed_unprocessed_pseudogene", "processed_transcript",
  "antisense", "transcribed_unitary_pseudogene", "polymorphic_pseudogene",
  "lincRNA", "sense_intronic", "transcribed_processed_pseudogene",
  "sense_overlapping", "IG_V_pseudogene", "pseudogene", "TR_V_gene",
  "3prime_overlapping_ncRNA", "IG_V_gene", "bidirectional_promoter_lncRNA",
  "snRNA", "miRNA", "misc_RNA", "snoRNA", "rRNA", "IG_C_gene", "IG_J_gene",
       "TR_J_gene", "TR_C_gene", "TR_V_pseudogene", "TR_J_pseudogene",
  "IG_D_gene", "ribozyme", "IG_C_pseudogene", "TR_D_gene", "TEC",
  "IG_J_pseudogene", "scRNA", "scaRNA", "vaultRNA", "sRNA", "macro_lncRNA",
  "non_coding", "IG_pseudogene"), output.prefix = ".")

pca(experiment.table = "./_log2TPM.txt", type="TPM", covariatesInNames=TRUE,
    samplesName=TRUE, principal.components = c(1, 2),
    legend.position="topright",
    pdf=TRUE, output.folder = getwd())

file.rename(from="pca.pdf", to="pca_w_c17.pdf")
```

Exercise2c: Execute differential expression with wrapperDeseq2 function.

Althouhg sample c17 is technically similar to the other samples it is very different from healty donors. We will remove it to run differential expression.

```r
library(docker4seq)

setwd("") #set exercise2 as working folder
library(docker4seq)
#folders MUST contain full path
folders <- paste(getwd(), c("c14", "c15", "c18", "n2", "n4", "n5", "n11"), sep="/")
#defining the two groups with healthy donors as Cov.1
```

```
my.covariates <- c(rep("Cov.1",3), (rep("Cov.2",4)))
sample2experiment(sample.folders=folders, covariates=my.covariates, batch = NULL,
  bio.type = c("protein_coding", "unitary_pseudogene",
  "unprocessed_pseudogene", "processed_pseudogene",
  "transcribed_unprocessed_pseudogene", "processed_transcript",
  "antisense", "transcribed_unitary_pseudogene", "polymorphic_pseudogene",
  "lincRNA", "sense_intronic", "transcribed_processed_pseudogene",
  "sense_overlapping", "IG_V_pseudogene", "pseudogene", "TR_V_gene",
  "3prime_overlapping_ncRNA", "IG_V_gene", "bidirectional_promoter_lncRNA",
  "snRNA", "miRNA", "misc_RNA", "snoRNA", "rRNA", "IG_C_gene", "IG_J_gene",
       "TR_J_gene", "TR_C_gene", "TR_V_pseudogene", "TR_J_pseudogene",
  "IG_D_gene", "ribozyme", "IG_C_pseudogene", "TR_D_gene", "TEC",
  "IG_J_pseudogene", "scRNA", "scaRNA", "vaultRNA", "sRNA", "macro_lncRNA",
  "non_coding", "IG_pseudogene"), output.prefix = ".")


pca(experiment.table = "./_log2TPM.txt", type="TPM", covariatesInNames=TRUE,
    samplesName=TRUE, principal.components = c(1, 2),
    legend.position="topright", pdf=TRUE, output.folder = getwd())

file.rename(from="pca.pdf", to="pca_wo_c17.pdf")

wrapperDeseq2(output.folder=getwd(), group="docker",
              experiment.table="_counts.txt", log2fc=1, fdr=0.1,
              ref.covar="Cov.1", type="gene", batch=FALSE)
```

Exercise2d: Use morpheus to make an heat map of the differetially expressed genes

The first step is the generation of tables containing only differentially expressed genes. This can be done using the function **filterCounts**. You need only to provide the folder in which the differentially expressed genes were generated.

```
setwd("") #set exercise2 as working folder
library(docker4seq)
filterCounts(data.folder=getwd(), type="gene")
```

The **DEfiltered___log2TPM.txt** can be used as input for Morpheus. In Morpheus do z-score normalization and hierarchical clustering and k-mean clustering

Use GOrilla to make a GO enrichment analysis of the dataset using all three Ontologies.

GOrilla results

In my opinion interesting elements are the GO:

- GO:0000398: mRNA splicing, via spliceosome from BP
- GO:0030627: pre-mRNA 5'-splice site binding MF
- GO:0097525:spliceosomal snRNP complex CC

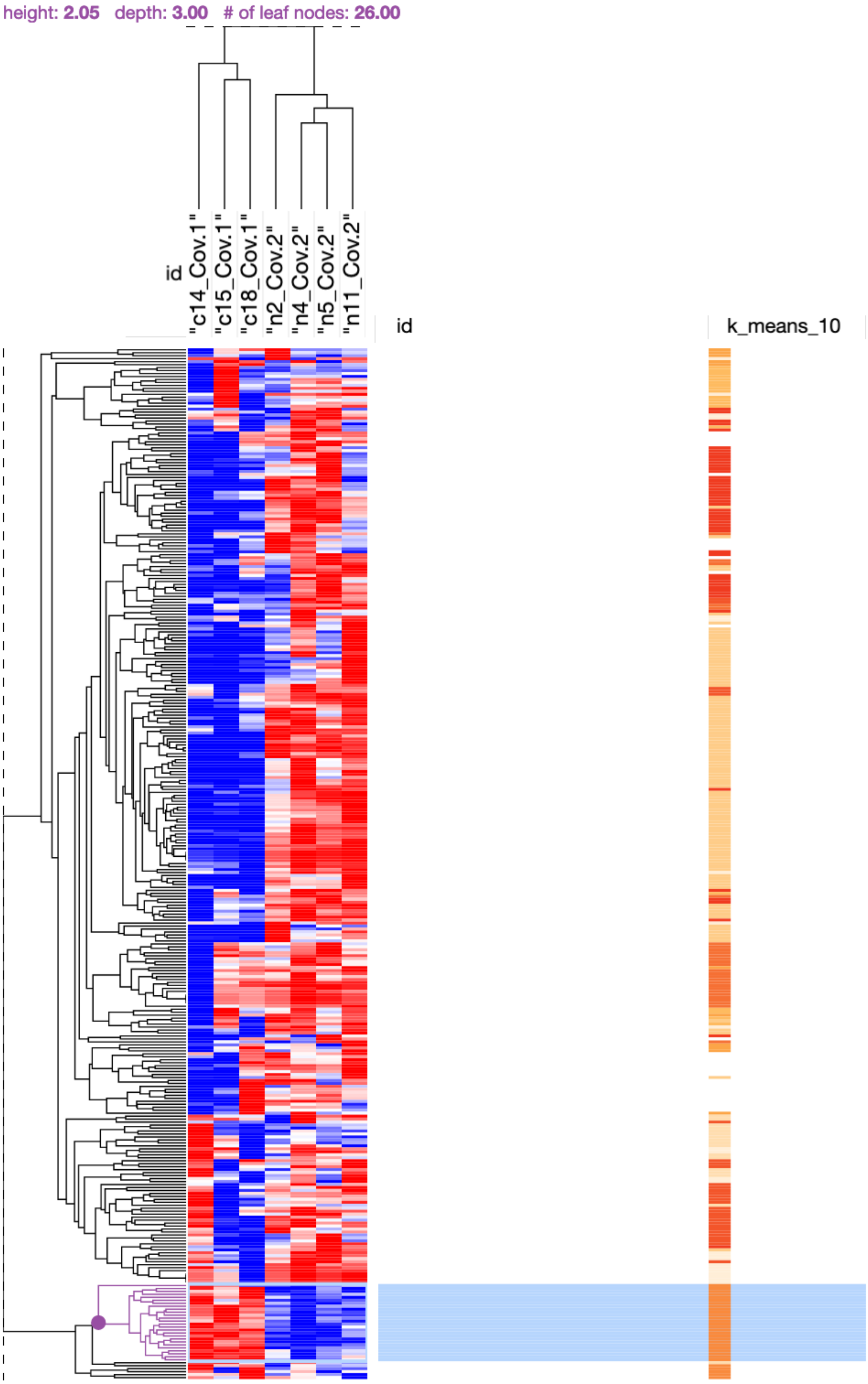

Figure 1: All data

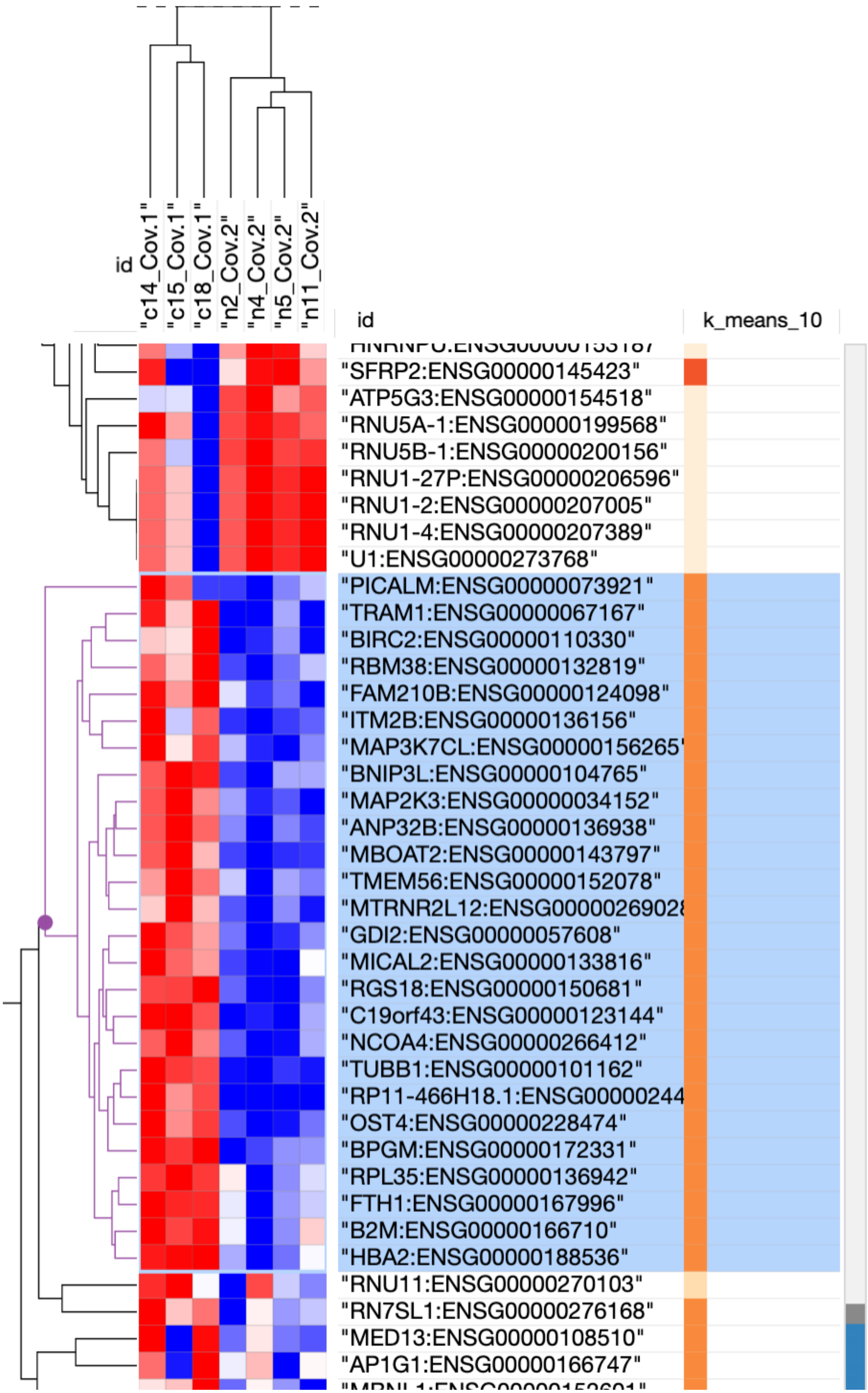

Figure 2: An example of a cluster where all healthy samples are characterized by high and homogeneous expression of these genes and neurological patients show down regulation,