DGE Analysis between IL17A+ and IL17A- T-cell spots on ST data

Christina Hillig

3/4/2021

Introduction

The following section describes the DGE analysis between IL17A positive and negative T-cell spots in the ST data set. The DGE analysis is performed on a spot-level and takes into account the cellular detection rate (cdr), variation between individuals and spots located in different tissue layers.

Initialisation

Load required R and Bioconductor packages:

```r
# R packages
rlibs <- c("dplyr", "gtools", "hash", "kableExtra", "knitr", "stringr", "tibble", "xlsx")
invisible(lapply(rlibs, require, character.only = TRUE))
# Bioconductor packages
bioclibs <- c("glmGamPoi", "pathview",  "org.Hs.eg.db")
invisible(lapply(bioclibs, require, character.only = TRUE))

getwd()
environment()

# Source R-scripts
source('../scripts/init.R')
source('../scripts/load_data.R')
source('../scripts/utils.R')
# R-script to prepare the count matrix and metaData
source('../scripts/data_preparation.R', local=environment())
# R-script to prepare and run the DGE Analysis
source('../scripts/run_DGEAnalysis.R', local=environment())
```

Define input directory.

```r
## Input
# Input directory
input.folder = "/Users/christina.hillig/R_studio/DGE_Analysis/Input/csv_files_DGE"
# Date
date = "2020-12-17"
# Data set
dataset.type = 'Tcells'
# Sequencing technique
seq.technique = "ST"
# Comparison
cytokine = 'IL17A'
```

```
# General input directory
input.dir = get_inputdir(input.dir=input.folder, date.file=date, dataset.type=dataset.type, seq.techniq

print(input.dir)

num.samples_per_donor = 2
```

Determine cut-parameters applied to identify differential expressed (DEx) genes.

```
## DGE Analysis parameters
# Used DGE Analysis library
dge.method = 'glmGamPoi'
# Design function
expr.design_function = ~ cdr + patient + annotation + condition
# Cut parameter
l2fc.factor = 1
fdr.value = 0.05
p.value = 0.05
# Multi-test method Benjamini-Hochberg (BH)
multitest.method =  "BH"
```

Create output directory.

```
## Output
# General output directory
output.dir <- get_outputdir(dataset.type = dataset.type, dge.method = dge.method,
                            seq.technique = seq.technique)
print(output.dir)
# Whether to save DGE Analysis results in a RData format
save.rdata = TRUE
```

Data Preparation

Load count matrix and metadata from input directory. MetaData contains information of barcodes, samples, batch; biospy type, disease, tissue layer as label; size factor, condition, and number of spots belonging to a condition.

```
# 1.1 get count matrix and MeatData as .csv or .xlsx files
file.names = list.files(path = input.dir, pattern = c("*.csv|*.xlsx"), recursive = TRUE)

# 1.2 load dataframe with counts and gene_ids
file.countmatrix = file.names[!grepl("metaData*", file.names)]
count.matrix <- load_countmatrix(path_file.name = file.path(input.dir, file.countmatrix))

# 1.3 load metaData like patients, number of spots involved in conditions and diseases
file.metaData = file.names[grepl("metaData*", file.names)]
cond.metadata <- load_metadata(metadata.path = file.path(input.dir, file.metaData))

table.counts <- kable(count.matrix[1:5, ], caption = 'T-cell Count matrix', format = 'html')
kable_styling(table.counts, bootstrap_options = "striped")
table.metadata <- kable(cond.metadata[1:5, ], caption = 'MetaData', format = 'html')
kable_styling(table.metadata, bootstrap_options = "striped")

# 1.4 Get name of comparison
name_comparison <- strsplit(file.countmatrix, "[.]")[[1]][1]
```

Prepare count matrix and metaData such that a pairwise comparison is possible. Further, the function ‘prepare_data’ was written such that multiple pairwise comparisons can be performed. Meaning, a for-loop can be inserted before the ‘Calculation of DEx genes’ to calculate multiple DGE Analysis between two conditions.

```
# Check if count matrix has at least 4 columns and that at least two conditions are in metaData.
# Here the conditions are IL17A+ vs. IL17A-
do.dgeanalysis = length(colnames(count.matrix)) > 4 &
  length(unique(cond.metadata[, 'condition'])) >= 2
if (do.dgeanalysis)
{
  # Split count matrix such that only pairwise comparisons are made
  counts_metaData_cellnames <- prepare_data(countmatrix = count.matrix,
                                            condmetadata = cond.metadata)

  sep_count_matrix <- counts_metaData_cellnames[[1]]
  sep_cond_metadata <- counts_metaData_cellnames[[2]]
} else {
  print("Conditions are not fullfilled -> Stop script!")
  quit(save="ask")
}
```

Calculation of DEx genes

The DEx genes are obtained between IL17A transcript positive and negative spots using the Bioconductor Package ‘glmGamPoi’. Each spot is treated as an individual sample. Further, the size factors which had been calculated on the whole ST data set is provided for the normalisation of the counts. Benjamini-Hochberg (BH) multi-test method is applied on the p-values to determine highly significant genes having a p-value and p-adjusted value below 0.05 and an absolute effect size above 1.

```
# Read out first pairwise comparison
# -> can be rewritten as for-loop for multiple pairwise comparisons
full.matrix <- sep_count_matrix[[1]]
full.metadata <- sep_cond_metadata[[1]]

#  create output path for pairwise comparison
str_comparison <- paste(unique(full.metadata['condition'])[[1]], collapse = "_vs_")
results.path <- file.path(output.dir, dataset.type,
                          tail(strsplit(input.dir, .Platform$file.sep)[[1]], 1),
                          str_comparison)
dir.create(results.path, recursive = TRUE, showWarnings = FALSE)

dge_res.design.metadata = run_dge.analysis(
  matrix = full.matrix, metadata = full.metadata,
  num_samples_donor = num.samples_per_donor, exp_design.function = expr.design_function,
  multitest.method = multitest.method, dge_method = dge.method, fdr.cut = fdr.value)

# Store output of DGE Analysis in different variables
dge_res.method = dge_res.design.metadata[[1]]
design.function = dge_res.design.metadata[[2]]
full.metadata = dge_res.design.metadata[[3]]

# Show DGE Analysis result
table.dgeres <- kable(dge_res.method, format = 'html')
table.dgeres <- kable_styling(table.dgeres, bootstrap_options = "striped")
table.dgeres
```

Add Information to DGE Analysis result and MetaData

Housekeeping genes

The housekeeping genes have been taken from https://www.genomics-online.com/resources/16/5049/housekeeping-genes/. The information whether a gene is a housekeeping gene or not is added. Additionally, this information can be later used to verify the DGE Analysis output. Housekeeping gene are expected to be not differentially expressed as they are necessary in each cells, basically they are needed to keep cells alive.

```
# Add label of housekeeping genes as 'y' for gene is a housekeeping gene and 'n' gene it not a housekee
dge_res.method <- add.hkg(dge.df.object = dge_res.method)
```

Gene names and Entrez gene IDs

For further analysis such as the Pathway enrichment analysis the gene names and Entrez gene IDs are mapped to the gene symbols. If possible, use directly the gene Ensembl and the same Genome as the reference Genome should be used due to possible discrepancies between the mapping.

```
# Add gene names and entrezid
symbols <- dge_res.method$gene_symbol
entrezid_genename <- map_genesymbol_entrezid_genename(gene.symbols = symbols)
dge_res.method <- do_add.column(df = dge_res.method, entrezid = entrezid_genename[[1]],
                                gene.name = entrezid_genename[[2]])

# Show DGE Analysis result
table.dgeres <- kable(dge_res.method, format = 'html')
table.dgeres <- kable_styling(table.dgeres, bootstrap_options = "striped")
table.dgeres
```

Design formula

Add design function to metaData.

```
num_rows <- length(full.metadata$condition)
character.designfunction <- paste(as.character(colnames(design.function)[-1]), collapse = " + ")
# add design matrix as column to metaData
full.metadata['design'] <- rep(character.designfunction, num_rows)

# Show metData
table.metadata <- kable(full.metadata, format = 'html')
table.metadata <- kable_styling(table.metadata, bootstrap_options = "striped")
table.metadata
```

Save result and MetaData

Save metaData with used design function in .csv file:

```
write.table(
  full.metadata,
  file.path(results.path, paste(dataset.type, str_comparison, "DGE_metaData.csv", sep = "_")),
            sep = ",", col.names = NA, append = FALSE)
```

Save genes with p-values, p.adjusted value and log2FC in .csv file:

```
write.table(
  dge_res.method,
  file.path(results.path,
            paste(dataset.type, str_comparison, dge.method, "DGE_all_genes.csv", sep = "_")),
  sep = ",", col.names = NA, append = FALSE)
```

Save multiple objects in RData format in your working directory:

```
if (save.rdata)
{
  save(
    full.metadata, design.function, dge_res.method,
    file = file.path(results.path, paste("DGE_object", dge_method, ".RData", sep = "_")))
}
```

References

Housekeeping genes: https://www.genomics-online.com/resources/16/5049/housekeeping-genes/ Cite used Bioconductor packages:

```
citation("glmGamPoi")
citation("org.Hs.eg.db")
citation("pathview")
```

Session Information

```
sessionInfo()
```