DESeq2 - Tutorial Pre-requisites and Homework

- R and Rstudio
- Bioconducter packages

Install R and RStudio

For Mac users

To Install R

- Open an internet browser and go to www.r-project.org.
- Click the "download R" link in the middle of the page under "Getting Started."
- Select a CRAN location (a mirror site) and click the corresponding link.
- Click on the "Download R for (Mac) OS X" link at the top of the page.
- Click on the file containing the latest version of R under "Files."
- Save the .pkg file, double-click it to open, and follow the installation instructions.

N* ow that R is installed, you need to download and install RStudio.

To Install RStudio

- Go to www.rstudio.com and click on the "Download RStudio" button.
- Click on "Download RStudio Desktop."
- Click on the version recommended for your system, or the latest Mac version, save the .dmg file on your computer, double-click it to open, and then drag and drop it to your applications folder.

For Windows users

To Install R

- Open an internet browser and go to www.r-project.org.
- Click the "download R" link in the middle of the page under "Getting Started."
- Select a CRAN location (a mirror site) and click the corresponding link.
- Click on the "Download R for Windows" link at the top of the page.
- Click on the "install R for the first time" link at the top of the page.
- Click "Download R for Windows" and save the executable file somewhere on your computer. Run the .exe file and follow the installation instructions.
- Now that R is installed, you need to download and install RStudio.

To Install RStudio

- Go to www.rstudio.com and click on the "Download RStudio" button.
- Click on "Download RStudio Desktop."
- Click on the version recommended for your system, or the latest Windows version, and save the executable file.
- Run the .exe file and follow the installation instructions.

After installing R and RStudio

- Start a new RStudio session.
- Copy the code snippet below.
- Paste the code to console and click enter.

```
### These will be installed from Bioconducter
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("pasilla")
BiocManager::install("DESeq2")

### These will be installed from CRAN
install.packages('ggplot2', repos='http://cran.us.r-project.org')
```

- Now, the required packages will be installed, this might need a coffee break.

Differential gene expression (DGE) with DESeq2

Self-Tutorial - Learning Objectives

- Importing package data into R
- Performing the differential expression analysis workflow with DESeq2

Review of the dataset

The data used in this workflow is stored in the *pasilla* package that summarizes an RNA-seq experiment. This package provides per-exon and per-gene read counts computed for selected genes from RNA-seq data that were presented in the article "Conservation of an RNA regulatory map between Drosophila and mammals" by Brooks AN, Yang L, Duff MO, Hansen KD, Park JW, Dudoit S, Brenner SE, Graveley BR, Genome Res. 2011 Feb;21(2):193-202, Epub 2010 Oct 4, PMID: 20921232. The experiment studied the effect of RNAi knockdown of Pasilla, the Drosophila melanogaster ortholog of mammalian NOVA1 and NOVA2, on the transcriptome. The package

vignette describes how the data provided here were derived from the RNA-Seq read sequence data that are provided by NCBI Gene Expression Omnibus under accession numbers GSM461176 to GSM461181.

- Please, allocate your 5 minutes to the given paper and review the dataset. The link to the paper (https://genome.cshlp.org/content/21/2/193)

Importing the data to R

- Firstly, import the `pasilla` data.

```
library("pasilla")

pasCts <- system.file("extdata",
                      "pasilla_gene_counts.tsv",
                      package="pasilla", mustWork=TRUE)

pasAnno <- system.file("extdata",
                       "pasilla_sample_annotation.csv",
                       package="pasilla", mustWork=TRUE)

cts <- as.matrix(read.csv(pasCts,sep="\t",row.names="gene_id"))

coldata <- read.csv(pasAnno, row.names=1)

coldata <- coldata[,c("condition","type")]

coldata$condition <- factor(coldata$condition)

coldata$type <- factor(coldata$type)

head(cts,2)

### You will be seeing the count matrix as below.


##             untreated1 untreated2 untreated3 untreated4 treated1 treated2
treated3
## FBgn0000003          0          0          0          0        0        0
1
## FBgn0000008         92        161         76         70      140       88
70
```

- We additionally need to chop off the "fb" of the row names of coldata, so the naming is consistent. Than, we should reorder the colnames of `cts` matrix according to the rownames of `coldata`

```
rownames(coldata) <- sub("fb", "", rownames(coldata))
cts <- cts[, rownames(coldata)]
```

- Now, let's check how the samples are assigned and grouped.

```
coldata

##              condition        type
## treated1     treated single-read
## treated2     treated  paired-end
## treated3     treated  paired-end
## untreated1 untreated single-read
## untreated2 untreated single-read
## untreated3 untreated  paired-end
## untreated4 untreated  paired-end
```

- We can use the function called DESeqDataSetFromMatrix() in order to convert a count matrix to a DESeqDataSet object. We load the DESeq2 package which the function belongs to.

```
library("DESeq2")
dds <- DESeqDataSetFromMatrix(countData = cts,
                              colData = coldata,
                              design = ~ condition)
```

- Now, lets see what the dds object stores for us. It is an object merging metadata and counts, later on statistical analysis results will be part of that object too.

```
dds

## class: DESeqDataSet
## dim: 14599 7
## metadata(1): version
## assays(1): counts
## rownames(14599): FBgn0000003 FBgn0000008 ... FBgn0261574 FBgn0261575
## rowData names(0):
## colnames(7): treated1 treated2 ... untreated3 untreated4
## colData names(2): condition type
```

- Now, lets do our first DGE test with DESeq2.

```
dds <- DESeq(dds)
results <- results(dds)
```

- Congratulations. You have succesfully completed your DGE analysis with DESeq2 with only 2 lines of code. In the tutorial, we'll be deeply covering these 2 lines of code. Now, lets check out the results.

```
View(data.frame(results)
```

- Please, try to understand what roles of the functions in this short tutorial are.
- In R, you can type ? before the function to see detailed usage. Such as ? DESeqDataSetFromMatrix

Please watch these videos before tutorial

1. On [FDR Values (https://www.youtube.com/watch?v=K8LQSvtjcEo&t=304s)](https://www.youtube.com/watch?v=K8LQSvtjcEo&t=304s)

2. On p-values (https://www.youtube.com/watch?v=vemZtEM63GY)

- If you have trouble through R, contact me at vogulcan@sabanciuniv.edu