**BMS353**

**Bioinformatics for Biomedical Science**

**Data analysis for   
Alveolar Macrophages after**

**Murine Lung Transplant – RNAseq**

**Registration No. 180166459**

**Word count: 1474**

Alveolar Macrophages after Murine Lung Transplant

180166459

2/8/2021

This is an R Markdown document. Markdown is a simple formatting syntax.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

**ABSTRACT**

**BACKGROUND:** Alveolar macrophages represent long-lived mononuclear phagocytes (immune cells) which are found in the alveolar sacs. Their role is ingesting particles that results from degradation and defense against respiratory pathogens, taking part of the innate immune system. Mus musculus organism

Data retrieved from GSE116583. The sequencing is done for expression in macrophages of mouse lungs in naïve (control group) compared with the 2h- and 24h- post-perfusion.

In this report, data analysis is done and biological differences that were shown in graphs and figures. Method implies working with R code in RStudio and generate using a markdown R file a report to document analysis, present data finding, results with plots and conclusions. This achieved using packages in R designed for this type of analysis.

**RESULTS AND INTERPRETATION:** The data obtained and the interpretation focus on biological differences between naïve and post-reperfusion. Differential expression was done and data visualization in plots with explanations

**Keywords: RNAseq, murine lung transplant, Galaxy, Differential expression, data visualisation**

**INTRODUCTION**

Next generation RNA sequencing replaces the microarray technology as it allows whole transcriptome sequencing. RNA sequencing involving the NGS – Next Generation Sequencing technologies require in general expertise from the user and understand the complex multi-step process. With packages available in R, Python and other languages, this type of analysis can bring some workflow automation and solutions. The field remains very complex and there is still a need to cover aspects and steps in the complex workflow of the RNA analysis.

For this project, R language was primarily used, with tools that facilitate differential expression. R works similarly with other high-level programming languages (such as Python) using libraries for data analysis. RNA seq differential expression is done to answer which genes are differentially expressed. This is shown in several plots, heatmaps, with statistical data.

The objective of the study was to examine transcriptional changes (in alveolar macrophagers after transplantation in mouse lung. The study presented results from 12 samples – 4 naive control, 4 transplanted, 4 after 24 hour transplantation. It is expected to see more variation in transplated and compare with naïve samples.

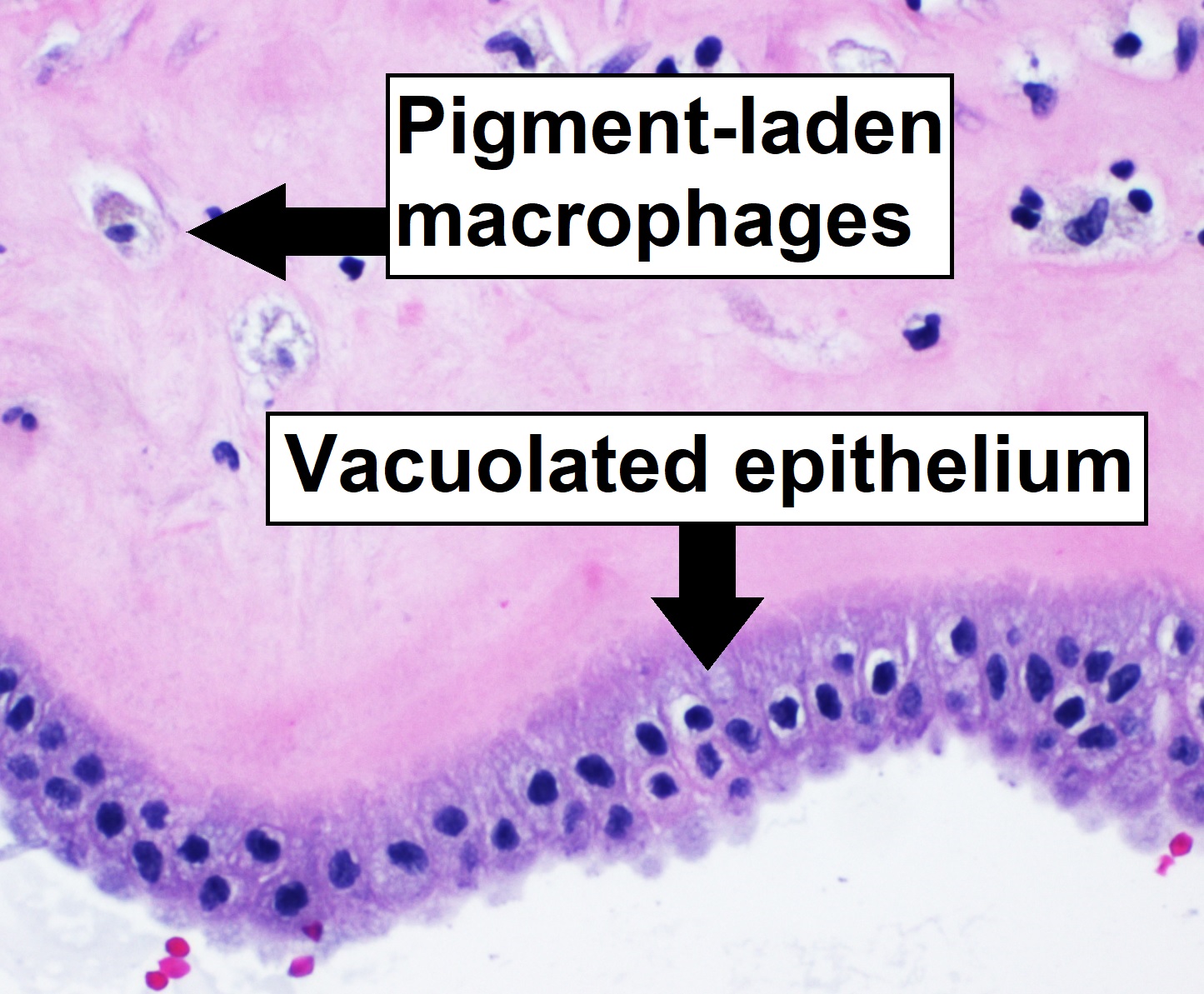


Figure 1 - Image that shows stained macrophage that is found in alveolar sac. Image is just informative but not necessarly relevant to this project

**METHODS**

R language used with RStudio IDE version 4.0.3. is strong in scientific research and analyzing data.

Naive represents the control group and 2h and 24h are 2 different stages after transplantation

Samples are in salmon quant output format file.

EXPERIMENTAL DESIGN

1. Identify at least 2 biological differences
2. Perform differential expression analysis
3. Data visualisation

Initially, all information was included in a R markdown file – code, comments, outputs, text, plots, images. Output of markdown chosen is docx, which contains all text, part of the chunk of codes and the outputs. Some of the document formatting and legends for images and plots were done in Word after generating the docx output from markdown file. For full code with comments, see markdown file BMS353 - 180166459 – document.Rmd. Version control system used is Git/Github. All files are saved in a Git repository. Public repository URL: <https://github.com/IoanaAndra/BMS353-Alveolar-Macrophages---RNA-seq-dataset>

|  |  |
| --- | --- |
| Sample number | Condition |
| 4 | naive |
| 4 | 2h post-perfusion |
| 4 | 24 h post-perfusion |

Data retrieved from: [GSE116583](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE116583)

In order to use the necessary functions and analyze the data, we use some packages available in R:

Installing all packages necessary for this project in R.

#SECTION FOR PACKAGES  
#installing all necessary packages  
# IF using same R version (4.0.3) and have all packages installed already, all lines of this section can remain commented   
   
# install.packages("readr")  
# install.packages("ggplot2") #\*

\*All code is visible in Rmd file.

**DATA RESULTS AND INTERPRETATION**

Steps of workflow

1. **Biological contrast, quality assessment, normalization**
2. **Perform differential expression analysis**
3. **Annotation, Data visualization**

1. Import raw data (.csv and quant.sf files)

library(readr)  
  
dataPath <- "tx2gene.csv" #asigning value which represent path of file  
  
file.exists(dataPath) #file.exists() return TRUE id file can be found or FALSE if it is not found

## [1] TRUE

tx2GeneFile <- read.csv(file = 'tx2gene.csv') #file with delimiter ","  
quantFile <- read\_tsv(file = 'salmon\_quant/SRR7457551/quant.sf') #file with delimiter "\t"  
head(tx2GeneFile, n = 2)

## TXNAME GENEID  
## 1 ENSMUST00000193812 ENSMUSG00000102693  
## 2 ENSMUST00000082908 ENSMUSG00000064842

head(quantFile, n = 3 )

## # A tibble: 3 x 5  
## Name Length EffectiveLength TPM NumReads  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 ENSMUST00000193812 1070 756. 0 0  
## 2 ENSMUST00000082908 110 4 0 0  
## 3 ENSMUST00000162897 4153 3719. 0 0

#view(tx2GeneFile)

Name represents in quant. sf the TXNAME found in tx2gene.csv. But tx2gene.csv contains all 12 samples (in order …51-…62)

Sample txt. presents the full experimental design that was followed. 12 columns for all 12 samples that are found in a salmon output format file that can be further read and accessed using read\_tsv

## geo\_accession name condition time run  
## 1 GSM3243460 N01\_AM\_Naive naive 0hr SRR7457557  
## 2 GSM3243461 N02\_AM\_Naive naive 0hr SRR7457558  
## 3 GSM3243462 N03\_AM\_Naive naive 0hr SRR7457559

## # A tibble: 60,226 x 5  
## Name Length EffectiveLength TPM NumReads  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 ENSMUST00000193812 1070 756. 0 0  
## 2 ENSMUST00000082908 110 4 0 0  
## 3 ENSMUST00000162897 4153 3719. 0 0  
## 4 ENSMUST00000070533 3634 3376. 0 0

After asigning to quants variable the files

## [1] TRUE

## [1] "CDSID" "CDSNAME" "EXONID" "EXONNAME" "GENEID" "TXID" "TXNAME"

## [1] "CDSCHROM" "CDSEND" "CDSID" "CDSNAME" "CDSPHASE"   
## [6] "CDSSTART" "CDSSTRAND" "EXONCHROM" "EXONEND" "EXONID"   
## [11] "EXONNAME" "EXONRANK" "EXONSTART" "EXONSTRAND" "GENEID"   
## [16] "TXCHROM" "TXEND" "TXID" "TXNAME" "TXSTART"   
## [21] "TXSTRAND" "TXTYPE"

## TXNAME GENEID  
## 1 ENSMUST00000193812 ENSMUSG00000102693  
## 2 ENSMUST00000082908 ENSMUSG00000064842  
## 3 ENSMUST00000192857 ENSMUSG00000102851

Summarising data

## # A tibble: 6 x 5  
## Name Length EffectiveLength TPM NumReads  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 ENSMUST00000193812 1070 756. 0 0  
## 2 ENSMUST00000082908 110 4 0 0  
## 3 ENSMUST00000162897 4153 3719. 0 0  
## 4 ENSMUST00000159265 2989 2604. 0.0174 2  
## 5 ENSMUST00000070533 3634 3376. 0 0  
## 6 ENSMUST00000192857 480 230 0 0

## # A tibble: 1 x 4  
## `min(quants$Length)` `max(quants$Length)` `min(quants$Effec~ `max(quants$Effe~  
## <dbl> <dbl> <dbl> <dbl>  
## 1 9 123179 2 119488.

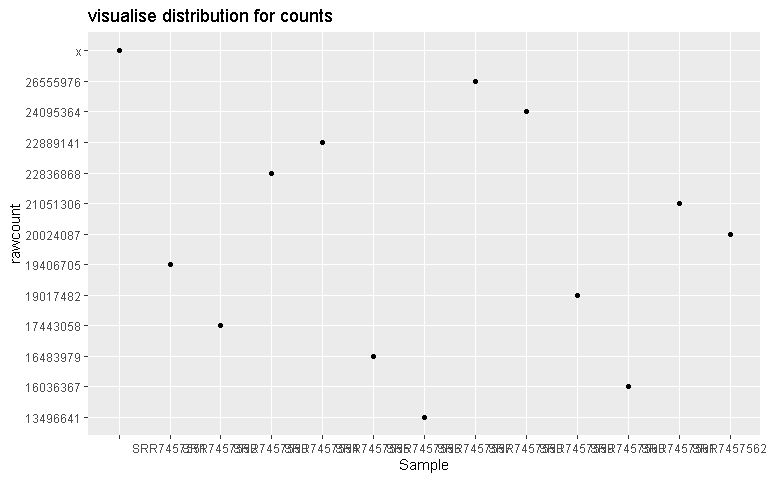
*2. Quality assesment*

Next, testing quality assesment for the raw reads of gene transcripts that were imported with tximport.

## DataFrame with 12 rows and 5 columns  
## geo\_accession name condition time  
## <character> <character> <factor> <character>  
## SRR7457557 GSM3243460 N01\_AM\_Naive naive 0hr  
## SRR7457558 GSM3243461 N02\_AM\_Naive naive 0hr  
## SRR7457559 GSM3243462 N03\_AM\_Naive naive 0hr

we can count the number of reads for each sample and print in the console. This can be written in a csv file.

This can be plotted into bar chart, scatterplot or boxplot.



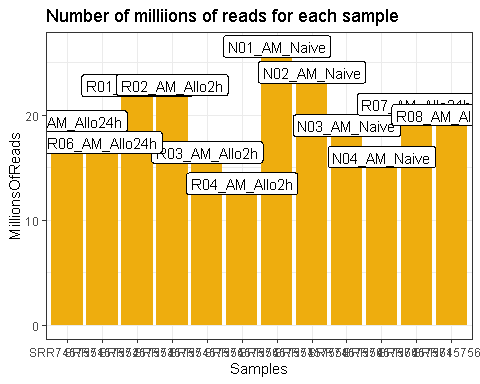
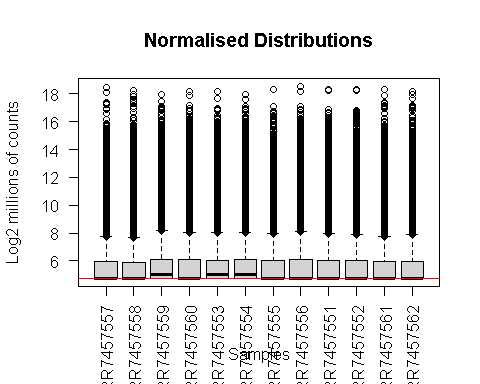
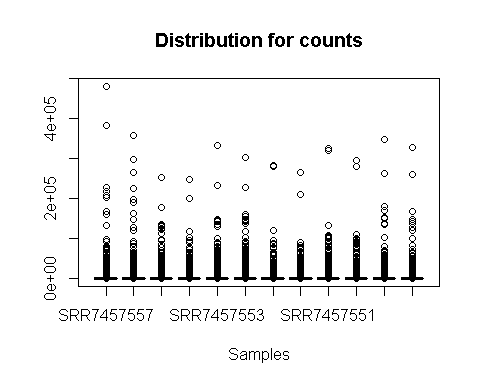
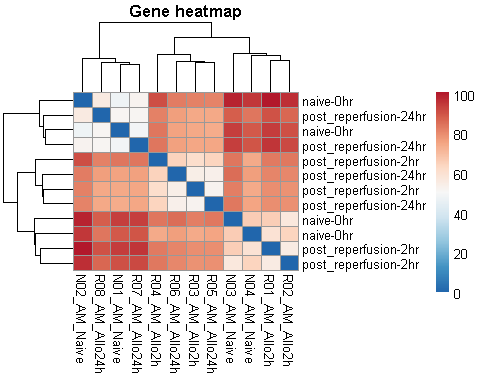


Figure - Bar charts that show number of millions of reads for each of the 12 samples

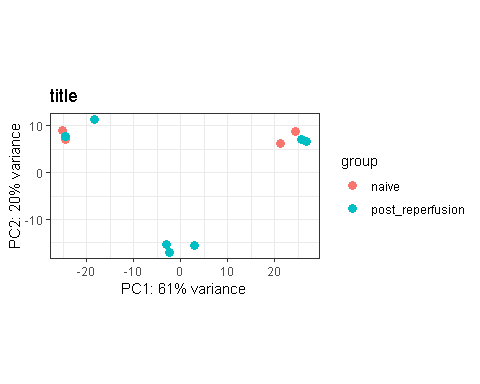
print('Gene from row 3 expressed in this number of samples:')  
sum(is\_expressed[3,])

After the imported files and counting from raw reads, normalization was done. PCAplot was done to show variance. It is expected to see more variance in 2h and 24h post-reperfusion. The initial sample info file presented 4 samples that were swapped.

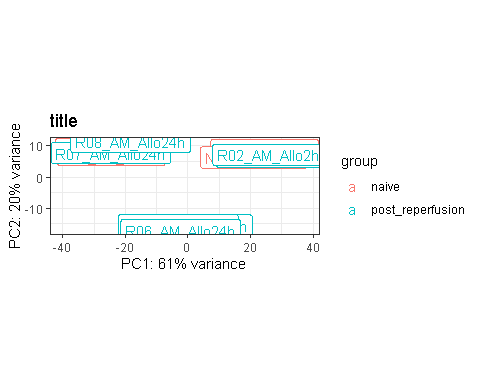




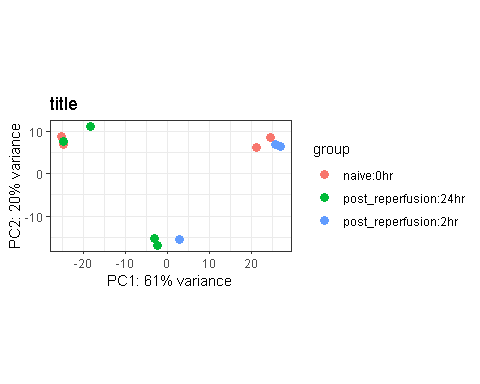
Heatmap of genes for all samples



PCA before correction, 2 samples of naive should be swapped with 2 samples of post-reperfusion



Change the PCA plot with one that shows name labels for each sample. The values entered were consistent, as the groups were split based on states (naive and post-reperfusion

 We need to swap N03 with R08 and N04 with R07 in order to get corrected sampleinfo txt file.

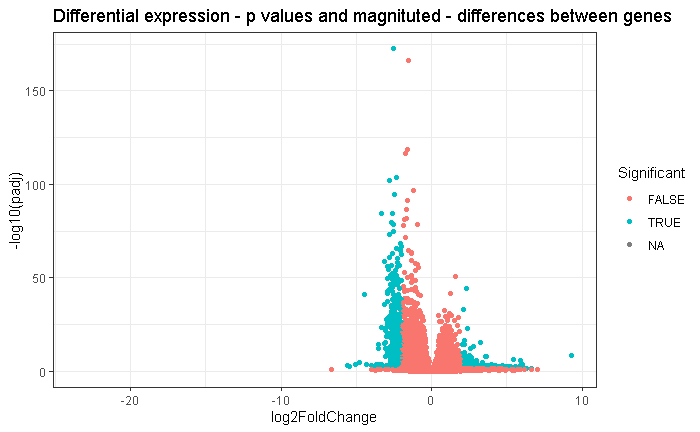
library(stringr)  
library(dplyr)  
sampleinfo %>%  
mutate(condition = str\_to\_lower(condition)) %>%  
mutate(time = str\_trim(time)) %>%  
mutate(condition = ifelse(name == "N03\_AM\_Naive","naive",condition)) %>%  
mutate(condition= ifelse(name == "R08\_AM\_Allo24h","post\_reperfusion",condition)) %>%  
   
write.table(file="meta\_data/sampleInfo\_corrected.txt",sep="\t",row.names = FALSE)  
  
 sampleinfo %>%  
 mutate(condition = str\_to\_lower(condition)) %>%  
 mutate(time = str\_trim(time)) %>%  
 mutate(condition = ifelse(name == "N04\_AM\_Naive","post\_reperfusion",condition)) %>%  
mutate(condition= ifelse(name == "R07\_AM\_Allo24h","naive",condition)) %>%  
  
 write.table(file="meta\_data/sampleInfo\_corrected.txt",sep="\t",row.names = FALSE)

library(readr)  
library(DESeq2)  
sampleinfo\_corrected <- read\_tsv("meta\_data/sampleInfo - Corrected.txt", show\_col\_types = FALSE)  
View(sampleinfo\_corrected)  
dds <- DESeqDataSetFromTximport(txi,   
 colData = sampleinfo\_corrected,  
 design <- ~condition)  
   
dds

vsd <- vst(dds)

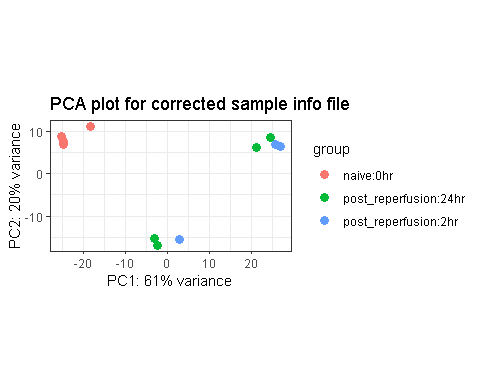
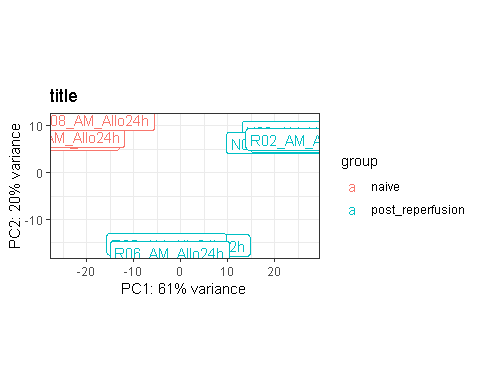
After quality assessment for counting and normalization of distribution, differential expression with DESeq2 was run.

1. **Annotation and data visualization**



**ACKNOWLEDGMENTS**

I would like to thank Dr Mark Dunning who is the module-coordinator for BMS353 for providing video materials and links to external resources via Blackboard and also to the rest of the team of the Sheffield Bioinformatics Core (based at the Sheffield Institute for Translational Neuroscience – SITraN) who offer online materials and training in bioinformatics.



*3. Differential Expression*