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:

**INTRODUCTION**

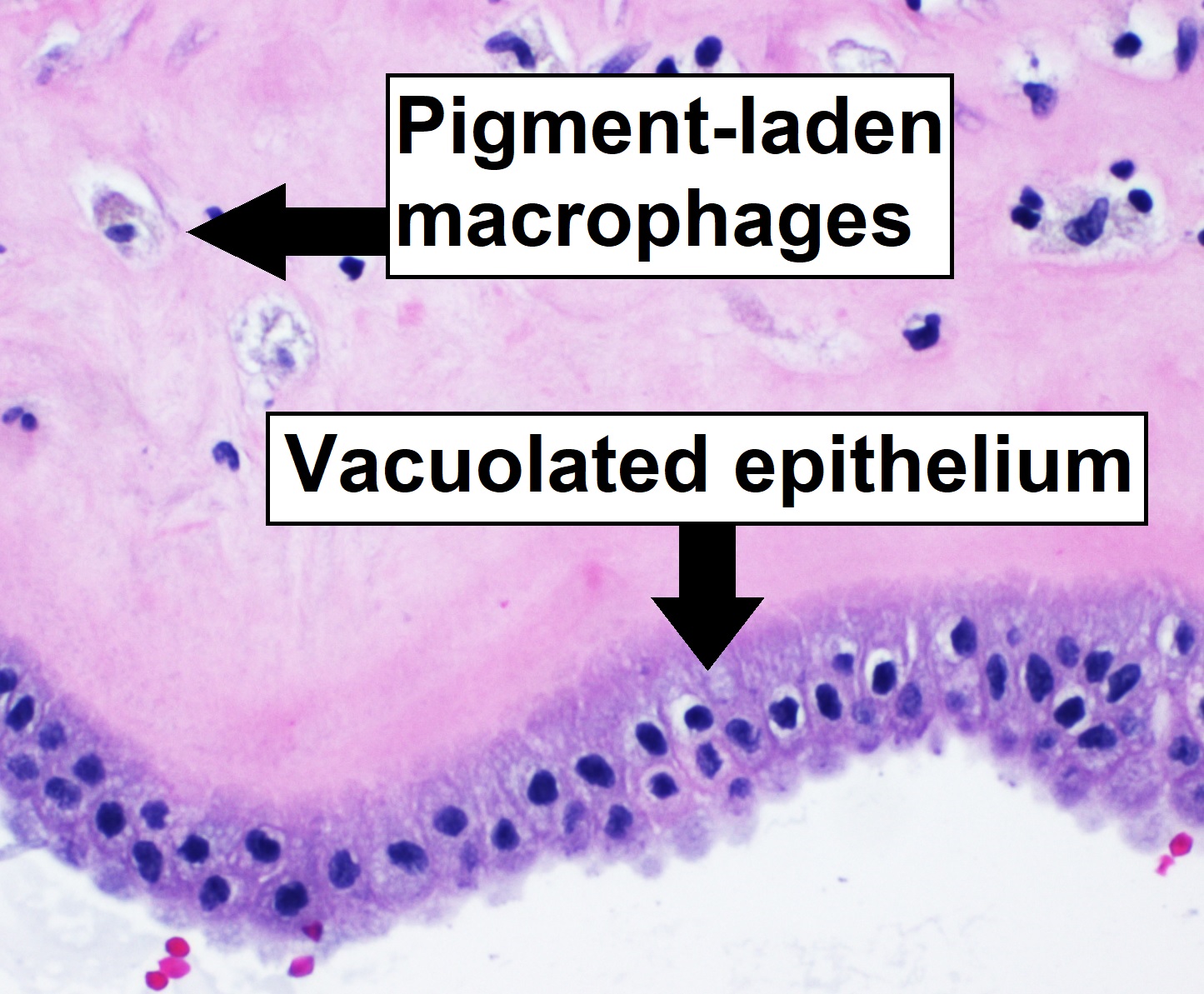


Figure 1 - Image that shows stained macrophage that is found in alveolar sac. Image is just informative but not necessarly relevant to this project

In this report we will analyze data for a research study that covers the following topic: Alveolar Macrophages after Murine Lung Transplant. For this we have the following resources provided:

*METHODS*

`````````````````````````````````````insert text from word

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

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

[Google drive](https://drive.google.com/drive/folders/1AqYi0Ps5t5xo6XYWXjOtO6eDOMpT1ADE) - Google drive folder with the following content:

* meta\_data
* salmon-quant.sf files
* tx2gene.csv

The project name BMS353-Alveolar-Macrophages—RNA-seq-dataset with all files and the code are available on github in a repository:

[Github](https://github.com/IoanaAndra/BMS353-Alveolar-Macrophages---RNA-seq-dataset)

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

* readr
* ggplot2
* dplyr

Every chunk of code can be name like this:

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.

*ANALYSIS*

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

CSV and TSV file formats can be stored in variables and display first row of them using head()

With head, we display first rows of data tx2gene.csv and a quant.sf

tx2gene.csv is the transcripts and genes relationship table file

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  
## 4 GSM3243463 N04\_AM\_Naive naive 0hr SRR7457560  
## 5 GSM3243464 R01\_AM\_Allo2h post\_reperfusion 2hr SRR7457553  
## 6 GSM3243465 R02\_AM\_Allo2h post\_reperfusion 2hr SRR7457554  
## 7 GSM3243466 R03\_AM\_Allo2h post\_reperfusion 2hr SRR7457555  
## 8 GSM3243467 R04\_AM\_Allo2h post\_reperfusion 2hr SRR7457556  
## 9 GSM3243468 R05\_AM\_Allo24h post\_reperfusion 24hr SRR7457551  
## 10 GSM3243469 R06\_AM\_Allo24h post\_reperfusion 24hr SRR7457552  
## 11 GSM3243470 R07\_AM\_Allo24h post\_reperfusion 24hr SRR7457561  
## 12 GSM3243471 R08\_AM\_Allo24h post\_reperfusion 24hr SRR7457562

## [1] "salmon\_quant//SRR7457551/quant.sf" "salmon\_quant//SRR7457552/quant.sf"  
## [3] "salmon\_quant//SRR7457553/quant.sf" "salmon\_quant//SRR7457554/quant.sf"  
## [5] "salmon\_quant//SRR7457555/quant.sf" "salmon\_quant//SRR7457556/quant.sf"  
## [7] "salmon\_quant//SRR7457557/quant.sf" "salmon\_quant//SRR7457558/quant.sf"  
## [9] "salmon\_quant//SRR7457559/quant.sf" "salmon\_quant//SRR7457560/quant.sf"  
## [11] "salmon\_quant//SRR7457561/quant.sf" "salmon\_quant//SRR7457562/quant.sf"

## # 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: 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  
## 5 ENSMUST00000192857 480 230 0 0  
## 6 ENSMUST00000195335 2819 250 0 0  
## 7 ENSMUST00000192336 2233 250 0 0  
## 8 ENSMUST00000194099 2309 250 0 0  
## 9 ENSMUST00000161581 250 20 0 0  
## 10 ENSMUST00000192973 2057 250 0 0  
## # ... with 60,216 more rows

## # A tibble: 3 x 5  
## Name Length EffectiveLength TPM NumReads  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 ENSMUST00000179077 887 762. 17.0 574.   
## 2 ENSMUST00000189352 548 295. 1.45 18.9  
## 3 ENSMUST00000178569 1083 250 0 0

## [1] 63095.94

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

## # 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.

## "1" "Name"  
## "2" "Length"  
## "3" "EffectiveLength"  
## "4" "TPM"  
## "5" "NumReads"

*2. Quality assesment*

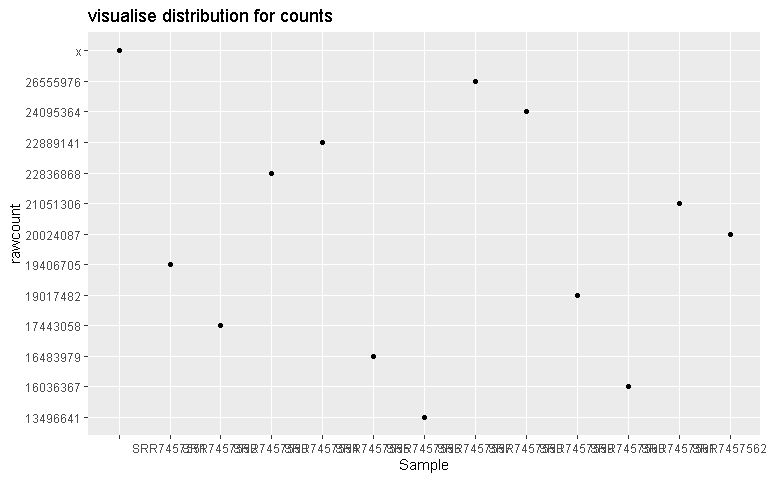
Next, testing quality assesment for the raw reads of gene transcripts that were imported with tximport.

## 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  
## SRR7457560 GSM3243463 N04\_AM\_Naive naive 0hr  
## SRR7457553 GSM3243464 R01\_AM\_Allo2h post\_reperfusion 2hr  
## ... ... ... ... ...  
## SRR7457556 GSM3243467 R04\_AM\_Allo2h post\_reperfusion 2hr  
## SRR7457551 GSM3243468 R05\_AM\_Allo24h post\_reperfusion 24hr  
## SRR7457552 GSM3243469 R06\_AM\_Allo24h post\_reperfusion 24hr  
## SRR7457561 GSM3243470 R07\_AM\_Allo24h post\_reperfusion 24hr  
## SRR7457562 GSM3243471 R08\_AM\_Allo24h post\_reperfusion 24hr  
## run  
## <character>  
## SRR7457557 SRR7457557  
## SRR7457558 SRR7457558  
## SRR7457559 SRR7457559  
## SRR7457560 SRR7457560  
## SRR7457553 SRR7457553  
## ... ...  
## SRR7457556 SRR7457556  
## SRR7457551 SRR7457551  
## SRR7457552 SRR7457552  
## SRR7457561 SRR7457561  
## SRR7457562 SRR7457562

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

## [1] "SRR7457557"  
## [1] 26555976  
##   
## [1] "SRR7457558"  
## [1] 24095364  
##   
## [1] "SRR7457559"  
## [1] 19017482  
##   
## [1] "SRR7457560"  
## [1] 16036367  
##   
## [1] "SRR7457553"  
## [1] 22836868  
##   
## [1] "SRR7457554"  
## [1] 22889141  
##   
## [1] "SRR7457555"  
## [1] 16483979  
##   
## [1] "SRR7457556"  
## [1] 13496641  
##   
## [1] "SRR7457551"  
## [1] 19406705  
##   
## [1] "SRR7457552"  
## [1] 17443058  
##   
## [1] "SRR7457561"  
## [1] 21051306  
##   
## [1] "SRR7457562"  
## [1] 20024087

This can be plotted into bar chart, scatterplot or boxplot



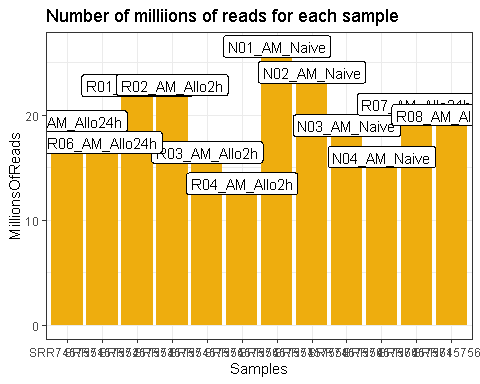
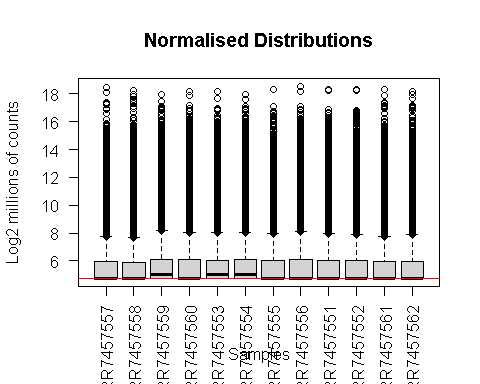
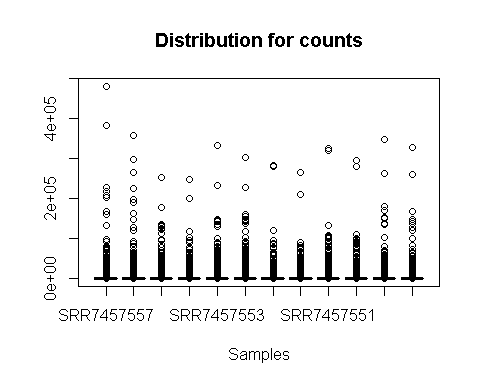
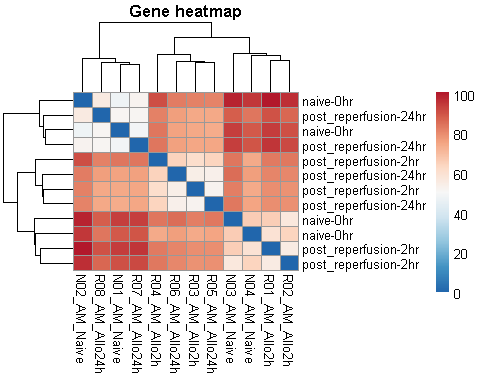


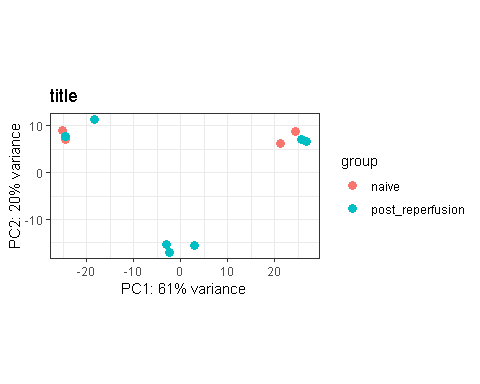
Figure - Bar charts that show number of millions of reads for each of the 12 samples

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

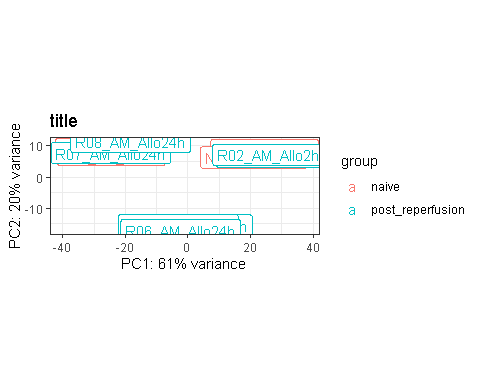




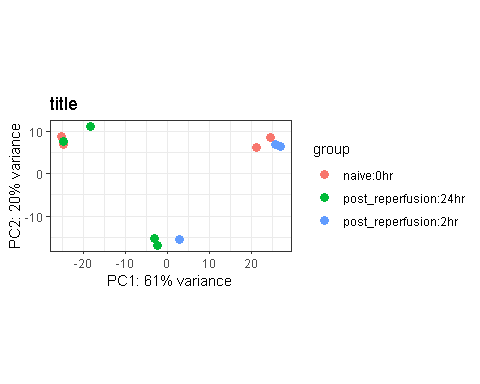
Heatmap of genes for all samples



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

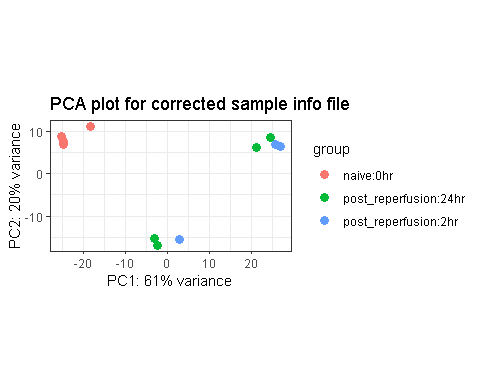
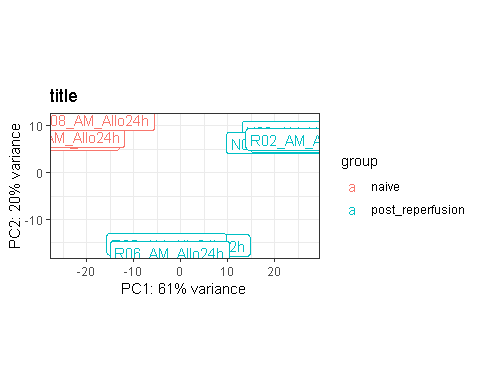
 We need to swap N03 with R08 and N04 with R07 in order to get corrected sampleinfo txt file.

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

## class: DESeqDataSet   
## dim: 52333 12   
## metadata(1): version  
## assays(2): counts avgTxLength  
## rownames(52333): ENSMUSG00000000001 ENSMUSG00000000003 ...  
## ENSMUSG00000115849 ENSMUSG00000115850  
## rowData names(0):  
## colnames: NULL  
## colData names(5): geo\_accession name condition time run

vsd <- vst(dds)



*3. Differential Expression*