Differential Gene Expression profiling of female with Lobular carcinoma and Infiltrating duct carcinoma in early stage of Breast Cancer

Yutian (Margery) Liu

Note: All data has been put in my Google Drive under the Folder 510_Final_Project

Introduction

Breast cancers that have spread into surrounding breast tissue are known as `invasive breast cancer` .Most breast cancers are invasive, but there are different types of invasive breast cancer.

The two most common are `invasive ductal carcinoma` and `invasive lobular carcinoma` .

This projected is aimed to find the differential expressed genes in early stage in these 2 types

- **Invasive (infiltrating) ductal carcinoma (IDC)**

This is the most common type of breast cancer. About 8 in 10 invasive breast cancers are invasive (or infiltrating) ductal carcinomas (IDC).

IDC starts in the cells that line a milk duct in the breast. From there, the cancer breaks through the wall of the duct, and grows into the nearby breast tissues. At this point, it may be able to spread (metastasize) to other parts of the body through the lymph system and bloodstream.

- **Invasive lobular carcinoma (ILC)**

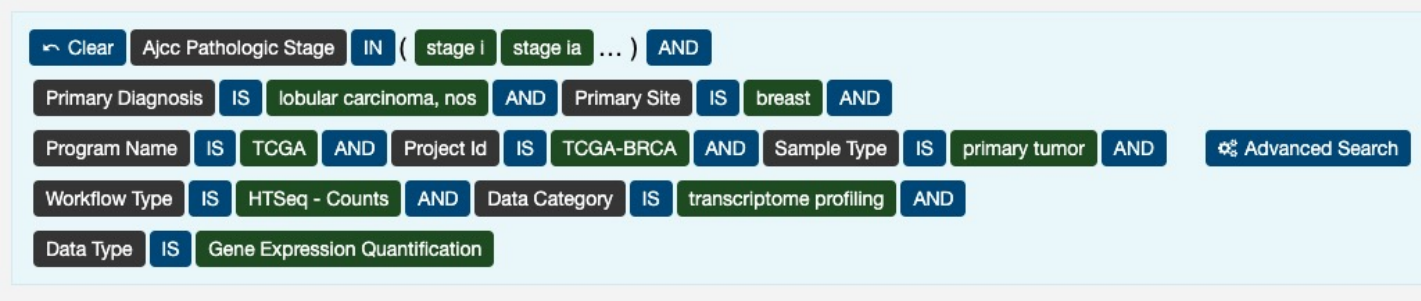
About 1 in 10 invasive breast cancers is an invasive lobular carcinoma (ILC).

ILC starts in the milk-producing glands (lobules). Like IDC, it can spread (metastasize) to other parts of the body. Invasive lobular carcinoma may be harder to detect on physical exam and imaging, like mammograms, than invasive ductal carcinoma. And compared to other kinds of invasive carcinoma, about 1 in 5 women with ILC might have cancer in both breasts.

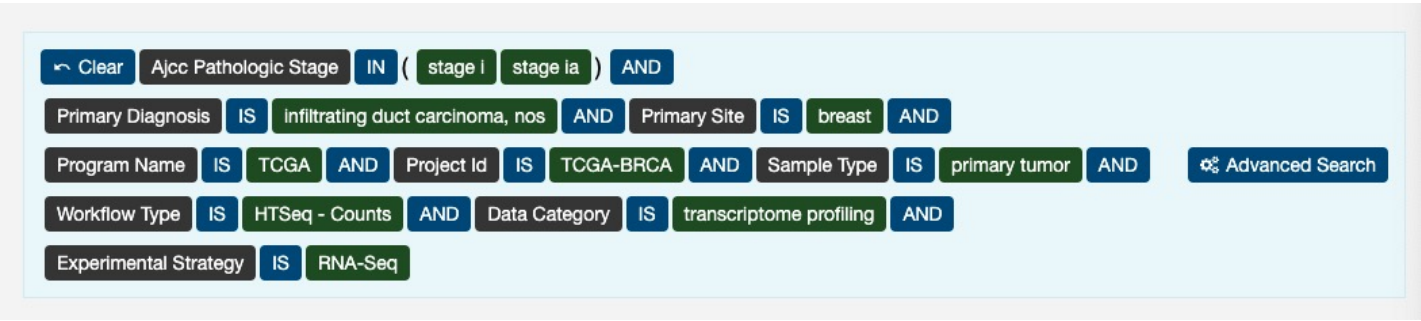
DATA Acquisition & Pre-processing

I. Filter:

- Filter the Files: Choose the conditions in following graph to generate two groups
  - 1. Group of **Loular Carcinoma** - Get 130 Files & 130 Cases (referred as `Logroup` in the following)



  - 2. Group of **Infiltrating Duct Carcinoma** -Get 135 Files & 135 Cases (referred as `Ductgroup` in the following)



Note: To make sure get similar number of 2 groups, in the `Logroup` , I choosed the stage I, stage IA, stage IB, stage II, stage IIA. stage IIB , and in `Ductgroup` , the stages are only stage I ,stage IA and stage IB.

II. DownLoad files and Pre-process Filenames

- Download (all done by clicking download bottons in the website)
  - 1. Download `Manifest` files of both groups
  - 2. Download `json` files of both groups
  - 3. Download `clinical` files of both groups

Rename files & Build a clinical data frame

Done by Changefilenames.Rmd

Do not need to change the name of uploaded files in GoogleDrive

- Step1 : Read Downloaded clinical csv in both groups
  - 1. Read files

```
coldata_lobular <- read.csv("~/Files/linicalLobular.tsv",sep = "\t")
coldata_Duct <- read.csv("~/Files/clinicalDuct.tsv",sep = "\t")
```

  - 2. Remove duplicated values

```
coldata_Duct <- coldata_Duct  %>%
  distinct(case_submitter_id,.keep_all = T)
coldata_lobular <- coldata_lobular  %>%
  distinct(case_submitter_id,.keep_all = T)
```

  - 3. Combine 2 groups

```
coldata <- rbind(coldata_Duct,coldata_lobular)
```

- Step2: Extract files (Do not need when Using data in GoogleDrive
  Extract HT-Seq counts in different folders and put them into a new folder in both groups
  The example script is provided by the following code:

```
setwd("*Directory*")
dir.create("*NewFolderName*")
for (dirname in dir('*Downloaded GGC files* ')){
     file <- list.files(paste0(getwd(),'/*Downloaded GGC files dirname*),pattern =
     file.copy(paste0(getwd(),'/*Downloaded GGC files* /',dirname,'/',file),'*NewF
```

- Step3 : Find the corresponding TCGA id by filename and change htseq counts filenames
  - 1. Map Filenames to TCGA id & Save it as `Duct/Lobular_filename_TCGAid.txt`

  The example script is provided by the following code:

```
metadata <- jsonlite::fromJSON("*Meta.json*")
naid_df <- data.frame()
for (i in 1:nrow(metadata)){
    naid_df[i,1] <- metadata$file_name[i]
    naid_df[i,2] <- metadata$associated_entities[i][[1]]$entity_submitter_id}
```

  - 2. only grab the TCGA's first 12 characters in clinical file

```
attach(naid_df)
naid_df$TCGA_id=substr(TCGA_id,regexpr("T",TCGA_id,),regexpr("T",TCGA_id)+11)
```

  - 3. save the file to change file names

```
write.table(naid_df,"Lobular/Duct_filename_TCGAid.txt", quote = FALSE, row.names = FALSE,col.names =
```

Used Files :

Lobular_filename_TCGAid.txt

Duct_filename_TCGAid.txt

- Step4: Change filename
  - 1. Use `change_name.sh` to change the filenames into TCGA-Format

```
#!/bin/bash

cat $1 |while read line
do
     arr=($line)
     filename=${arr[0]}
     submitterid=${arr[1]}
     mv ${filename} ${submitterid}.htseq.counts.gz
done
```

Useage :

```
bash change_name.sh ~/Files/Lobular_filename_TCGAid.txt

bash change_name.sh ~/Files/Duct_filename_TCGAid.txt
```

  - 2. I want the filename to be as "Ductgroup/Lobulargroup + Digital seria number"
  - 3. So, I Used Excel to connect this name to TCGA id

**Built Files **

Lobulargroup_id.xlsx

Ductgroup_id.xlsx

  - 4. Use `name_change.sh` Change the filename (TCGA-ID format) to "Ductgroup/Lobulargroup + Digital seria number" based on their correspondence (Built Files)

```
#!/bin/bash

cat $1 |while read line
do
     arr=($line)
     filename=${arr[1]}
     TCGAid=${arr[0]}
     mv ${filename}.gz ${TCGAid}.gz
done
```

Used files

Ductgroupchange.txt

Lobulargroupchange.txt

Useage :

```
bash name_change.sh ~/Files/Ductgroupchange.txt

bash name_change.sh ~/Files/Lobulargroupchange.txt
```

- 5. Paste these files into a new folder named `New_Lobular_Duct`
- 6. Set the directory to point at this file for further analysis in `Rstudio`

Analyzing RNA-seq data with DESeq2

R Script is saved as PDF , you can check them here .

Rscripts_Analyzing RNA-seq data with DESeq2.PDF

Rscripts_Analyzing RNA-seq data with DESeq2.Rmd

DESeq2 Tutorial Website

Differential expression analysis

- Step1: Set the Directory to point at the folder with changed names of both groups & library all the required packages

```
library("DESeq2")
library("apeglm")
library("ggplot2")
library("vsn")
library("pheatmap")
library("RColorBrewer")
directory <- "~/New_Lobular_Duct/"
```

Note: in the script, it is the Absolute path

- Step2 : Generate required input for building `DESeqDataset`
  - 1. Generate the `sampleFiles` : ***Use `grep` to select those files containing string `group`***
  - 2. Generate the `sampleCondition` : ***Use `sub` to chop up the sample filename to obtain the condition status***
  - 3. Generate the `sampleTable` : ***Use `data.frame`*** to build the dataframe by `sampleFiles` & `sampleCondition`

```
sampleFiles <- grep("group",list.files(directory),value=TRUE)
sampleCondition <- sub("(.*group).*","\\1",sampleFiles)
sampleTable <- data.frame(sampleName = sampleFiles,
                          fileName = sampleFiles,
                          condition = sampleCondition)
sampleTable$condition <- factor(sampleTable$condition)
```

- Step3 : Build the `DESeqDataset` ---Get 60483 elements

```
library("DESeq2")
dds <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable,
                                  directory = directory,
                                  design= ~ condition)
dds
```

```
class: DESeqDataSet
dim: 60483 265
metadata(1): version
assays(1): counts
rownames(60483): ENSG00000000003.13 ENSG00000000005.5 ... ENSGR0000280767.1
  ENSGR0000281849.1
rowData names(0):
colnames(265): Ductgroup-TCGA-3C-AALK-01A-11R-A41B-07.htseq.counts.gz
  Ductgroup-TCGA-A2-A04N-01A-11R-A115-07.htseq.counts.gz ...
  Lobulargroup-TCGA-WT-AB44-01A-11R-A41B-07.htseq.counts.gz
  Lobulargroup-TCGA-XX-A89A-01A-11R-A36F-07.htseq.counts.gz
colData names(1): condition
```

Note : Extract the conditional information directly on the basis of the name of files , which ensures the one-to-one correspondence betweem the expression matrix and the sample

- Step4 : Build sampleTable with **multiple factors** (condition: Ductgroup/Lobulargroup & Stage)
  - 1. remove the suffix of filename in `sampleTable`

```
library(tidyr)
sampleTable <- sampleTable %>%
     tidyr::separate(fileName,into = c("fileName"),sep = "\\.")
```

  - 2. Merge the name information and clinicla information

```
colnames(Col_Duct_Lobular)[2] <- "fileName"
sampleTable <- merge(sampleTable,Col_Duct_Lobular,by="fileName")
```

  - 3. Select the necessary columnns only

```
library(dplyr)
sampleTableselect <- sampleTable%>%
     dplyr::select(fileName,sampleName.x,condition,ajcc_pathologic_stage)
```

- 4. Add the `Stage` factor

```
sampleTableselect$condition <- factor(sampleTableselect$condition)
sampleTableselect$Stage <- factor(sampleTableselect$ajcc_pathologic_stage)
```

- 5. Build the `DESeqDataset` by Multiple factors

```
ddsMF <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTableselect,
                              directory = directory,
                              design= ~ condition + Stage)
ddsMF
```

```
class: DESeqDataSet
dim: 60483 264
metadata(1): version
assays(1): counts
rownames(60483): ENSG00000000003.13 ENSG00000000005.5 ... ENSGR0000280767.1
  ENSGR0000281849.1
rowData names(0):
colnames(264): Ductgroup1 Ductgroup10 ... Lobulargroup98 Lobulargroup99
colData names(3): condition ajcc_pathologic_stage Stage
```

- Step5: Pre-filtering & Specify the factor levels ---the number of elements has decreased from 60483 to 50860
  - 1. ***Remove the rows which are less than 10 reads***

```
keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep,]
```

    As a result, the number of elements has decreased from 60483 to 50860

  - 2. **Check Factors**

```
head(ddsMF$condition)
head(ddsMF$Stage)
```

```
[1] Ductgroup Ductgroup Ductgroup Ductgroup Ductgroup Ductgroup
Levels: Ductgroup Lobulargroup
[1] Stage I  Stage I  Stage I  Stage I  Stage I  Stage IA
Levels: Stage I Stage IA Stage II Stage IIA Stage IIB
```

log2 fold change and Wald test p value： last level / reference level
log2 fold change： log2 (Lobulargroupt / Ductgroup)

- Step5: Differential Expression Analysis

    - 1. Use function `results` to generate 6 columns including log2FC, P-value, corrected P-value .etc
    - 2. Save them as "Lobular_Duct_res.csv". You can check them in Folder "Results" Res_Lobular_Duct_All.csv

Define differential expressed genes and filter them

- Step1:
    - 1. Define GEG as `padj <= 0.05 & abs(log2FoldChange) >= 1.5`
    - 2. Check its dimension

```
diff_gene_deseq2 <-subset(res, padj <= 0.05 & abs(log2FoldChange) >= 1.5)
dim(diff_gene_deseq2)
head(diff_gene_deseq2)
```

    - 3. 463 DEG are saved the as `New_DEG_Lobular_Duct.csv"`
    - 4. You can check the result here New_DEG_Lobular_Duct.csv
- Step2: ID Transfer
    - 1. Transfer the Ensemble ID to geneID in DEG
    - 2. Extract Up-regulated genes
    - 3. Extract Down-regulated genes
    - 4. Annotate if this gene is UP/Down Regulated in the DEG

  Click to See Results

  All_diff_Reg.csv

  Down_diff.csv

  Up_diff.csv

Log fold change shrinkage for visualization and ranking

Pass the `dds` object to the function `lfcShrink` and use `apeglm` to shrink effect size.

```
resLFC <- lfcShrink(dds, coef="condition_Lobulargroup_vs_Ductgroup", type="apeglm")
resLFC
```

```
log2 fold change (MAP): condition Lobulargroup vs Ductgroup
Wald test p-value: condition Lobulargroup vs Ductgroup
DataFrame with 50860 rows and 5 columns
                    baseMean log2FoldChange     lfcSE      pvalue        padj
                   <numeric>      <numeric> <numeric>   <numeric>   <numeric>
ENSG00000000003.13 2994.2734      0.0398904 0.0911447 6.72717e-01 7.99464e-01
ENSG00000000005.5    53.8553      0.9556161 0.2381667 3.90512e-06 6.50184e-05
ENSG00000000419.11 2071.6910     -0.3288230 0.0629304 5.26143e-08 1.81304e-06
ENSG00000000457.12 2086.8536      0.1079616 0.0620041 7.09663e-02 1.69797e-01
ENSG00000000460.15  744.0237     -0.1095951 0.0785919 1.36537e-01 2.74298e-01
...                      ...            ...       ...         ...         ...
ENSG00000281909.1   0.854753      0.2339201 0.2373185 0.095029260 2.10845e-01
ENSG00000281910.1   0.422694     -0.0631372 0.2217272 0.926703856 9.60304e-01
ENSG00000281912.1  97.306569     -0.0273729 0.0890516 0.738561999 8.44083e-01
ENSG00000281918.1   2.374675      0.3136781 0.2636270 0.039290120 1.08932e-01
ENSG00000281920.1   5.935921      0.6980221 0.1739988 0.000004453 7.23534e-05
```

`resLFC` is more compacted compared to `res`
column `stat` is removed after shrinking

Information of results columns

```
mcols(res)$description
```

```
[1] "mean of normalized counts for all samples"
[2] "log2 fold change (MLE): condition Lobulargroup vs Ductgroup"
[3] "standard error: condition Lobulargroup vs Ductgroup"
[4] "Wald statistic: condition Lobulargroup vs Ductgroup"
[5] "Wald test p-value: condition Lobulargroup vs Ductgroup"
[6] "BH adjusted p-values"
```

Plot Vignette

- 1. MA- Plot

`plotMA` shows the log2 fold changes attributable to a given variable over the mean of normalized counts for all the samples in the DESeqDataSet

the plotMA function is used to plot the histogram of mean of Normalized Counts. If the adjusted P value is less than 0.1, the color is marked.Whatever exceeds it is marked as a triangle

X： Mean of Normalized Counts

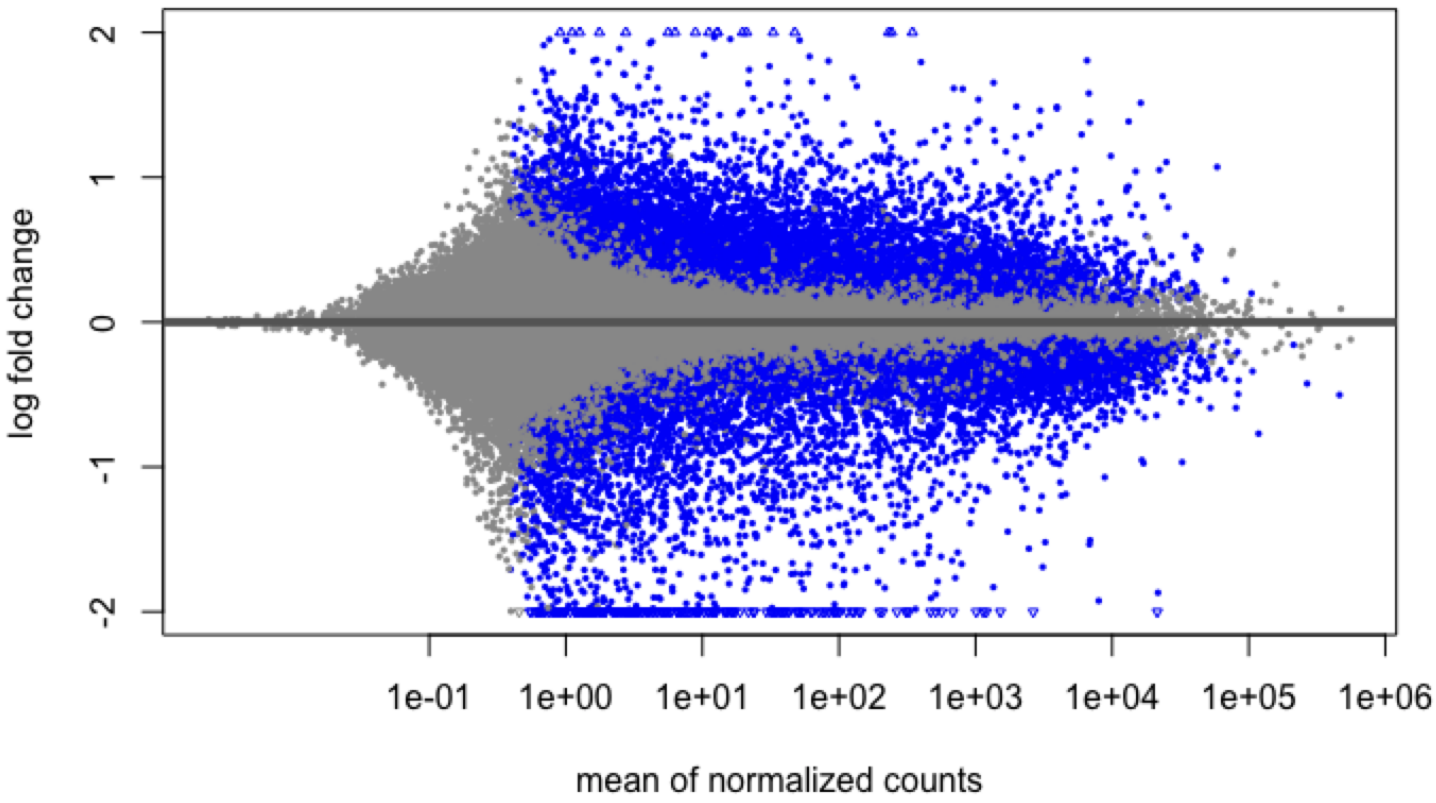
Y : Log Fold Change

Docs above the line mean Up-regulated

Docs below the line mean Down-regulated

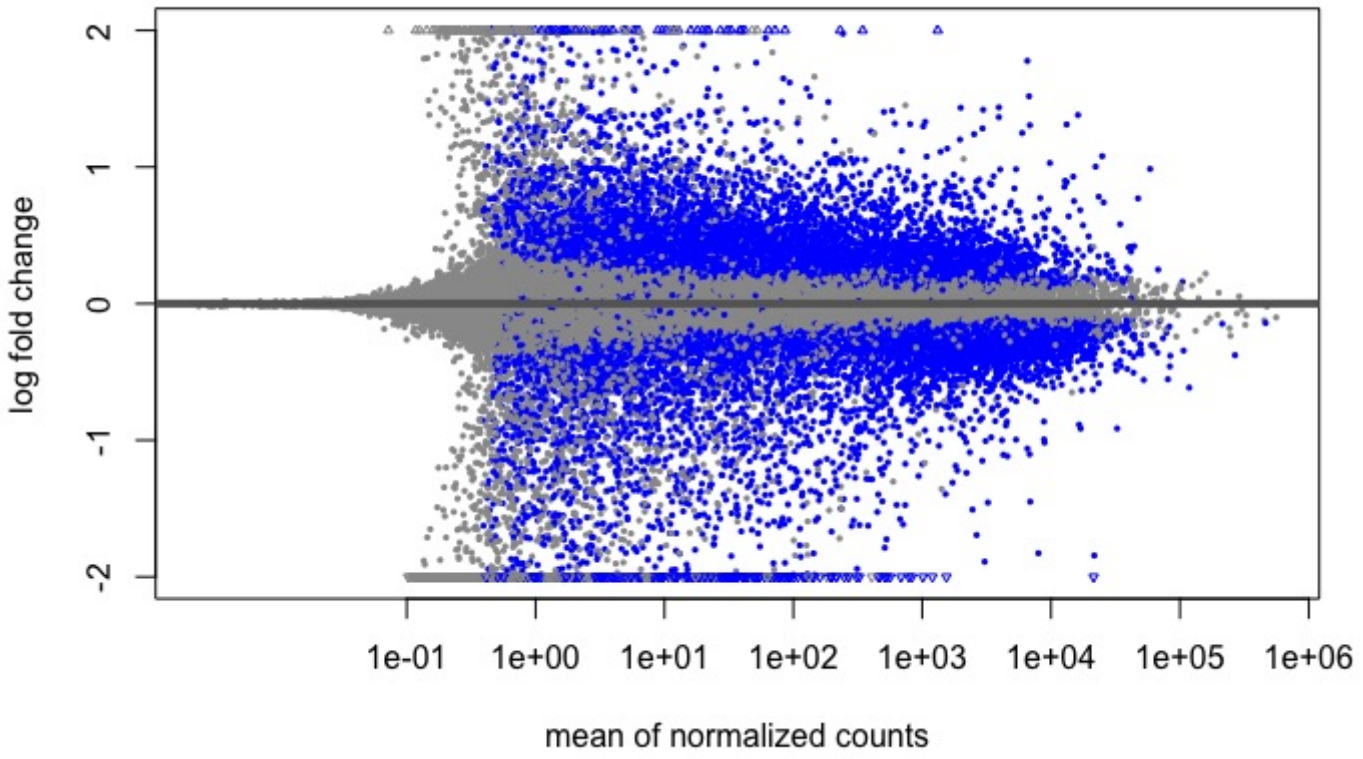
- res

```
plotMA(res, ylim=c(-2,2))
```



- resLFC

```
plotMA(resLFC, ylim=c(-2,2))
```



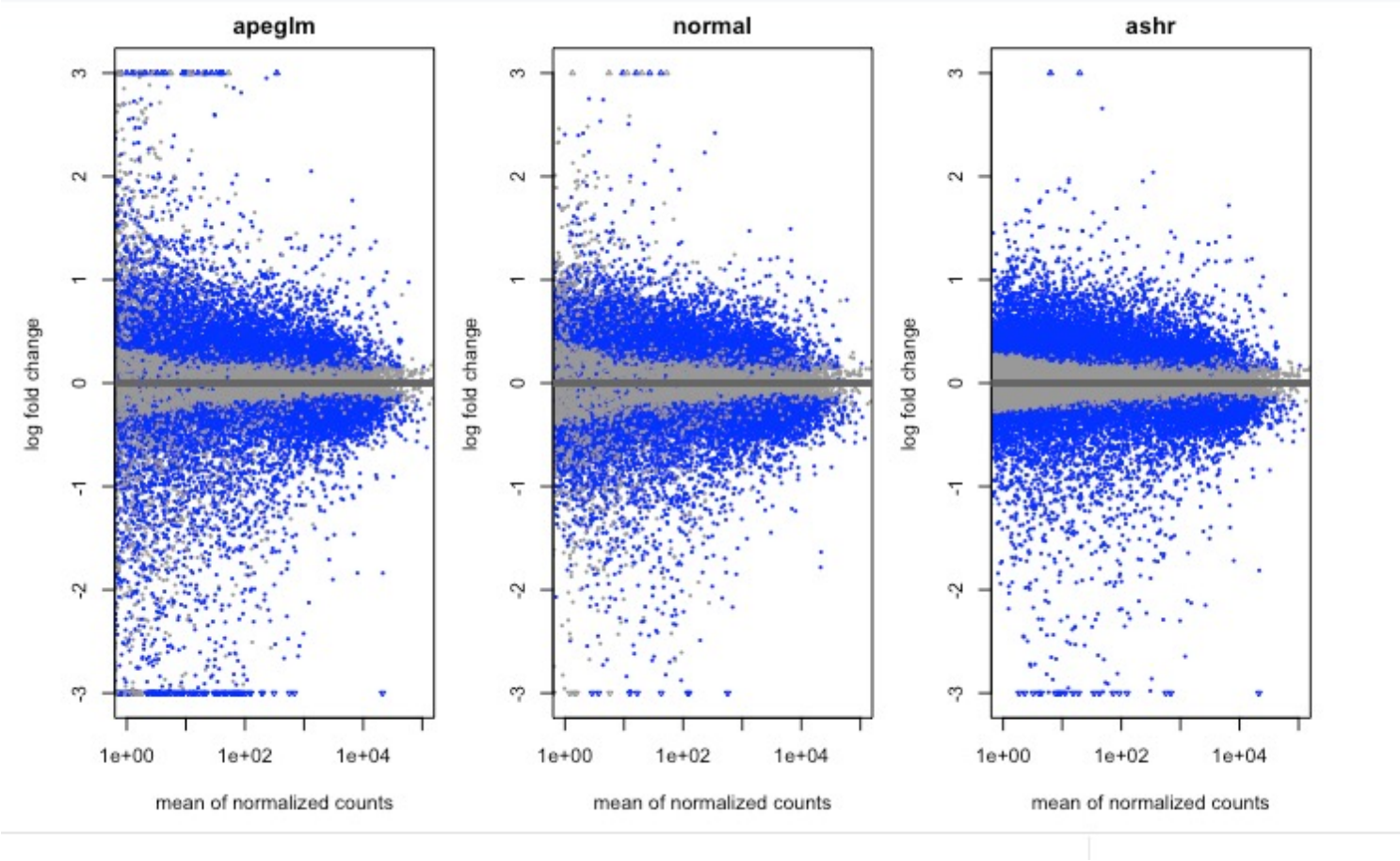
`resLFC` directly removes the noise associated with LFC from low-expressed genes without the need to manually set the threshold

- different Types

`apeglm` is the adaptive t prior shrinkage estimator from the apeglm package (Zhu, Ibrahim, and Love 2018). As of version 1.28.0, it is the default estimator.

`ashr` is the adaptive shrinkage estimator from the ashr package (Stephens 2016). Here DESeq2 uses the ashr option to fit a mixture of Normal distributions to form the prior, with method="shrinkage".

`normal` is the the original DESeq2 shrinkage estimator, an adaptive Normal distribution as prior. (Deleted in the Script for Running too slow)



`type='apeglm` and `type='ashr'` have shown to have less bias than `type='normal'`

`type='normal` has been removed in the script for running too slow

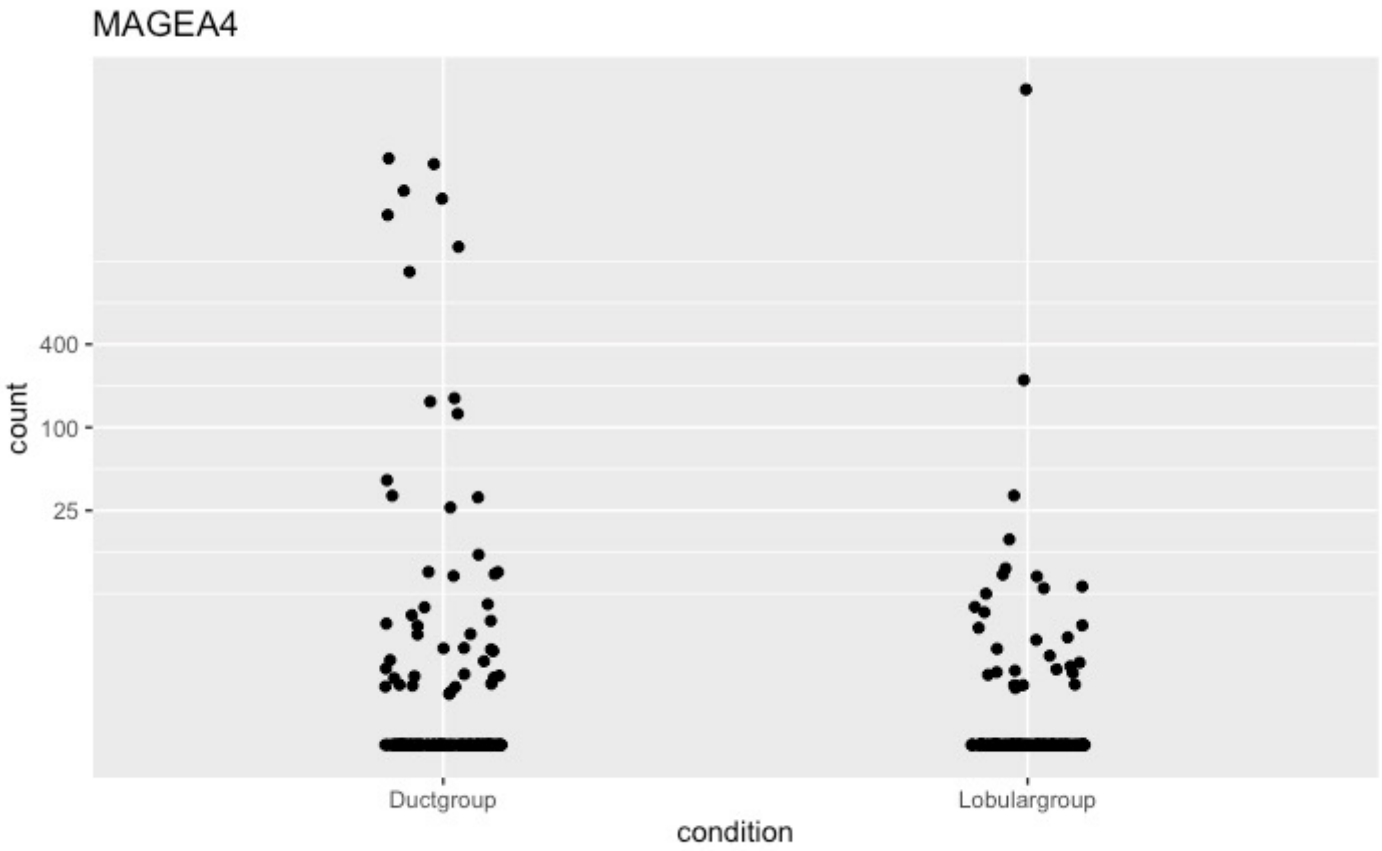
MA Plot fully demonstrated the relationship between gene abundance and expression changes. We can see that the lower to the left or the upper to the right, the more abundant and variable the genes are.

- 2. Plot Count

From the IPA result, gene `MAGEA4` which is Down-Regulataed compared to the reference (Duct group) is in the PATHway of disease "HER2 non-overexpressing breast carcinoma" (Category : Cancer,Organismal Injury and Abnormalities,Reproductive System Disease) , so I chose this gene to Plot Counts

- Plot Counts

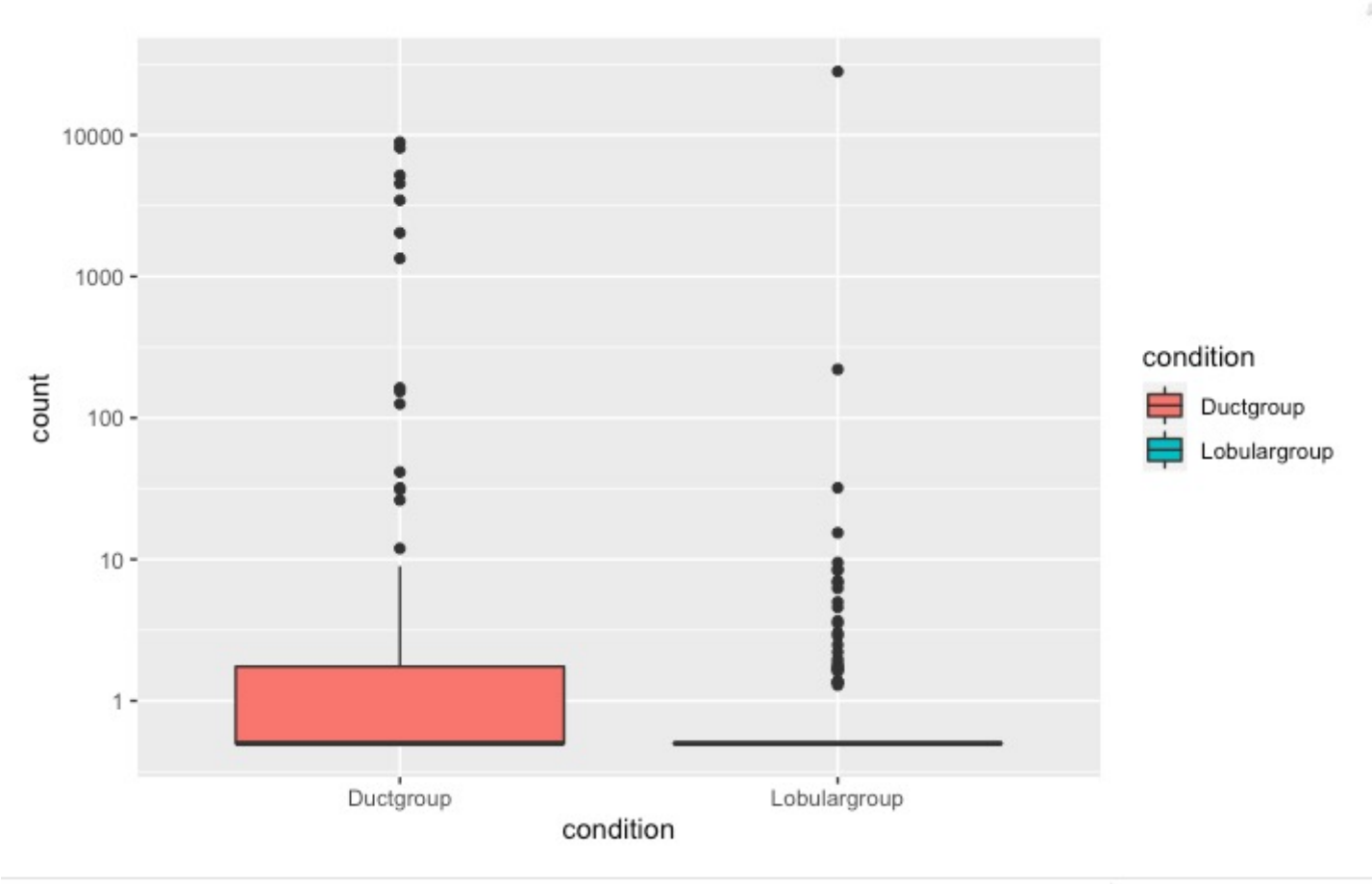
```
d <- plotCounts(dds, gene="ENSG00000147381.10", intgroup="condition",
                returnData=TRUE)
library("ggplot2")
ggplot(d, aes(x=condition, y=count)) + ggtitle("MAGEA4")+
  geom_point(position=position_jitter(w=0.1,h=0)) +
  scale_y_log10(breaks=c(25,100,400))
```



- Box plot

```
d1 <- plotCounts(dds,gene="ENSG00000147381.10", intgroup="condition",returnData = T)

ggplot(d1,aes(condition, count)) + geom_boxplot(aes(fill=condition)) + scale_y_log10()
```



From the boxplot, it is obvious that this gene is signicant expressed between 2 groups.

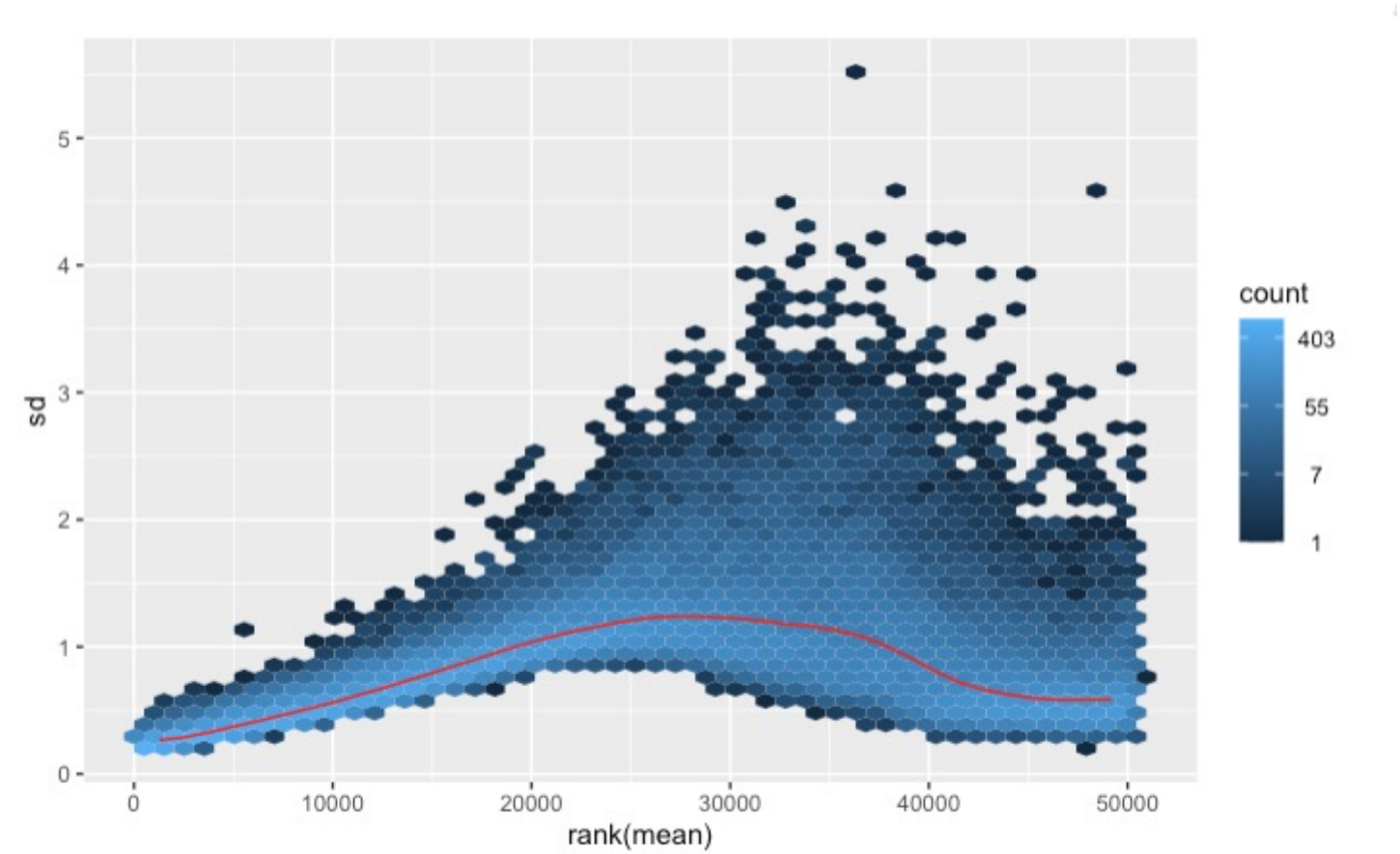
Data transformations and visualization

The mean and standard deviation of the converted data between samples were plotted by these Transformations

- dds

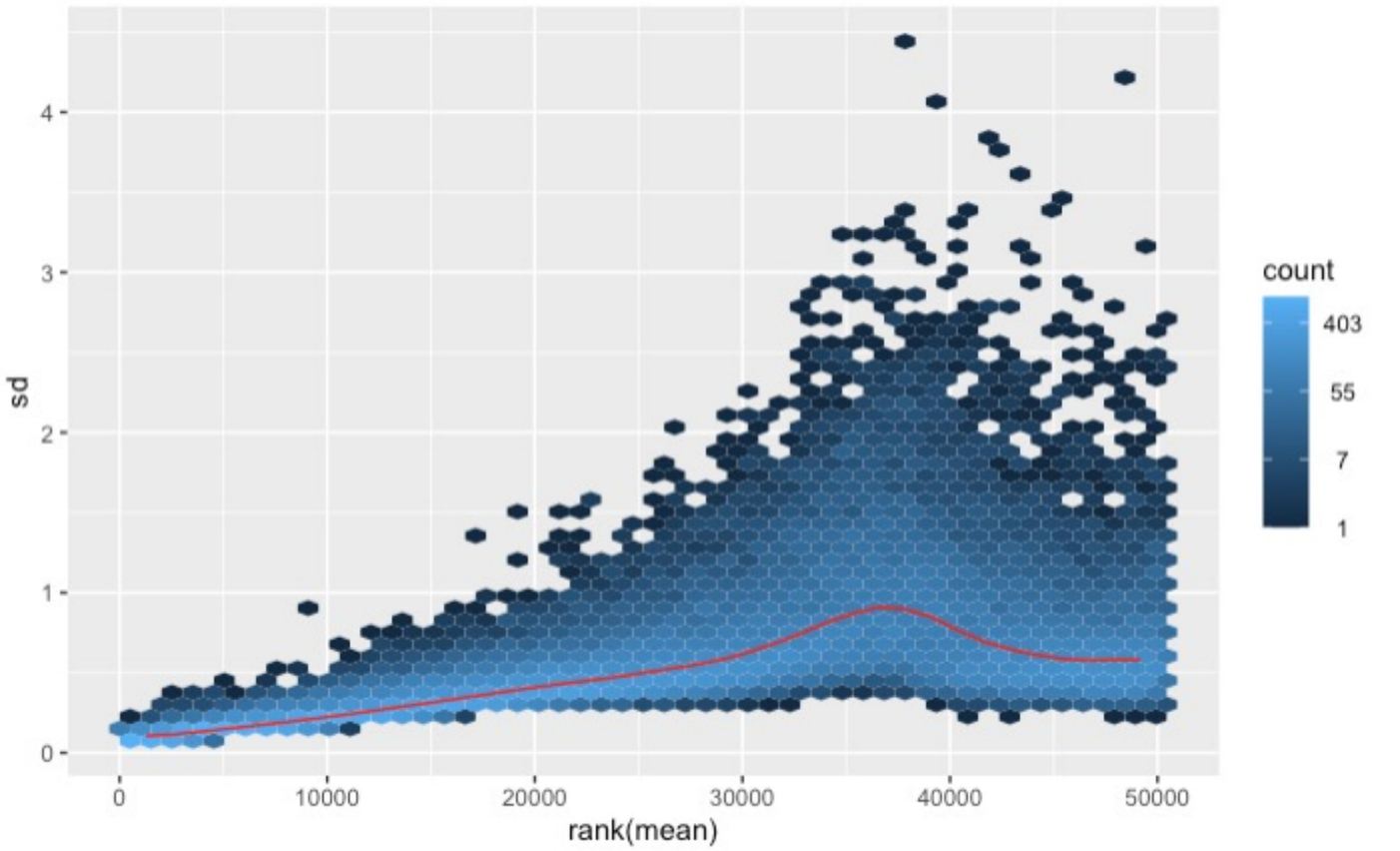
```
vsd <- vst(dds, blind=FALSE)
ntd <- normTransform(dds)
```

```
meanSdPlot(assay(ntd))
```



this gives log2(n + 1)

```
meanSdPlot(assay(vsd))
```



variance stabilizing transformation

The shifted logarithm has elevated standard deviation in the lower count range, and while for the variance stabilized data the standard deviation is roughly constant along the whole dynamic range.

- ddsMF

```
vsd <- vst(ddsMF, blind=FALSE)
ntd <- normTransform(ddsMF)
```

Data Quality Evaluation by sample clustering and visualization (Using Multiple Factors : condition and Stage)

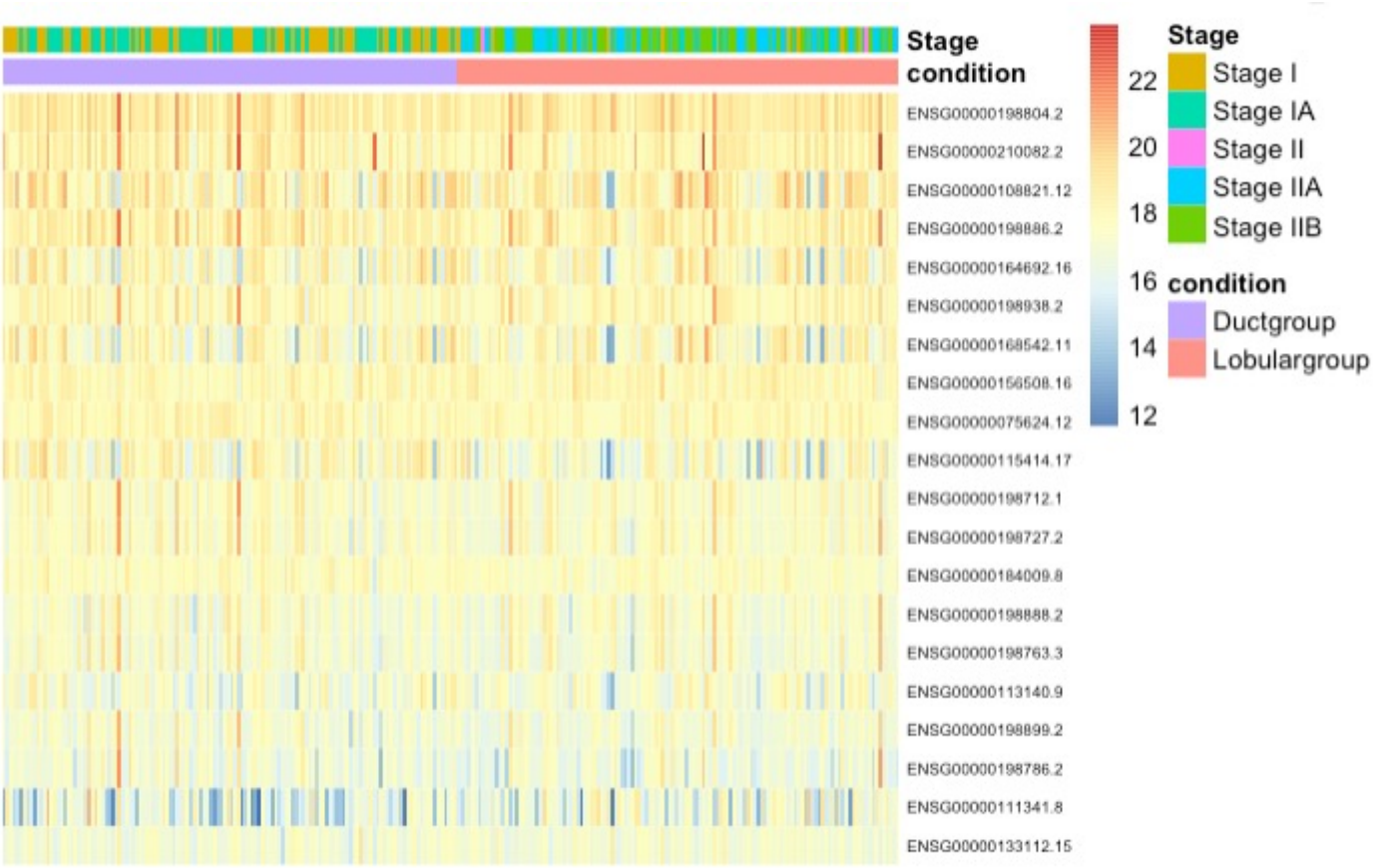
- Heatmap of the count matrix

Choose Top 20 gene to draw heatmap

```
library("pheatmap")
select <- order(rowMeans(counts(ddsMF,normalized=TRUE)),
                decreasing=TRUE)[1:20]
```

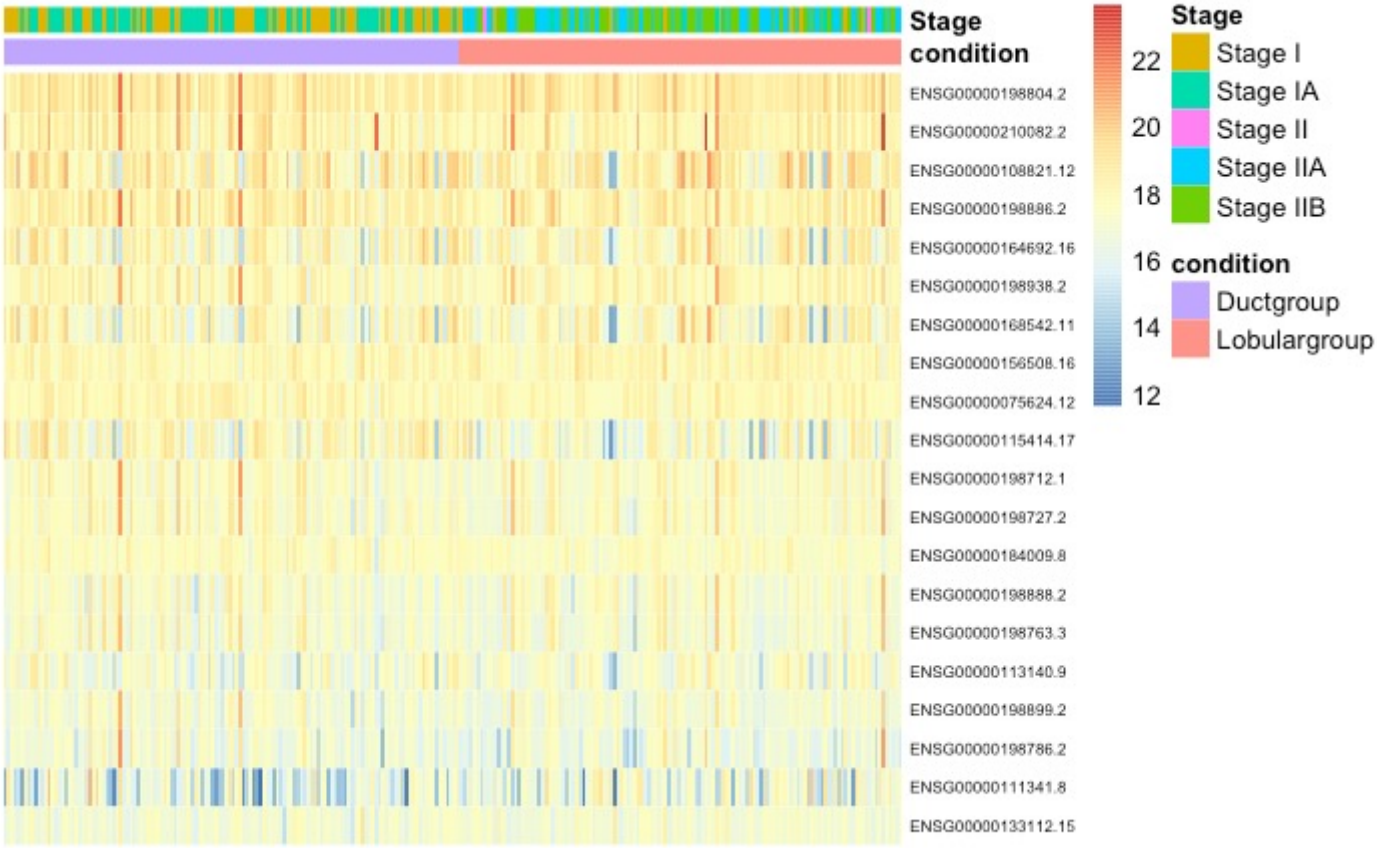
ntdMF

```
pheatmap(assay(ntdMF)[select,], cluster_rows=FALSE,show_colnames=FALSE,
    cluster_cols=FALSE,annotation = df,fontsize_row = 6)
```



vsdMF

```
pheatmap(assay(vsdMF)[select,], cluster_rows=FALSE,show_colnames=FALSE,
    cluster_cols=FALSE,annotation = df,fontsize_row = 6)
```



The difference is not clear

- Heatmap of the sample-to-sample distances

Sample Clustering

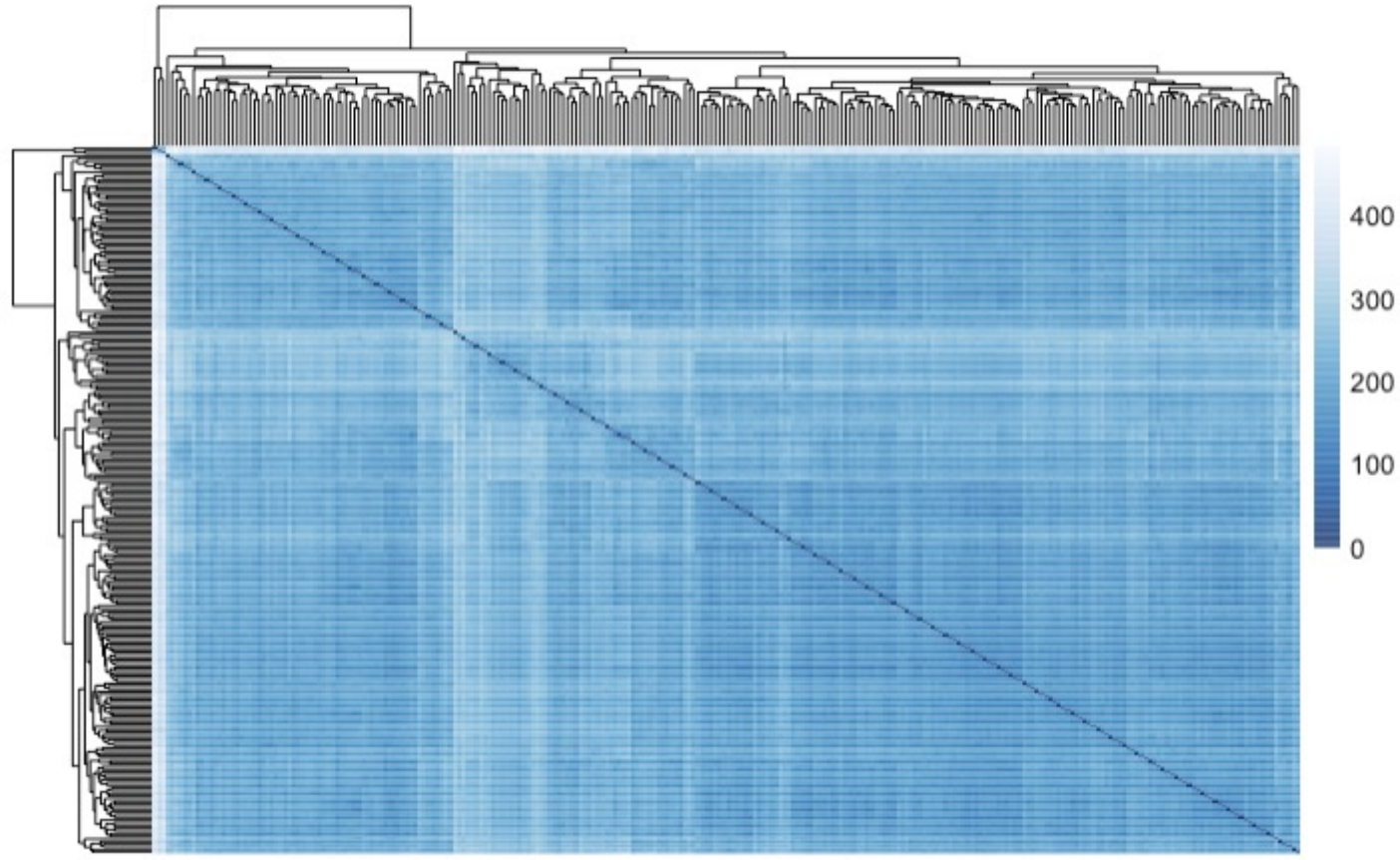
Here, I applyed the `dist` function to the transpose of the transformed count matrix to get sample-to-sample distances.

```
sampleDists <- dist(t(assay(vsdMF)))
```

Use `Euclidean distance`

```
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsdMF$condition, vsdMF$type,vsdMF$Stage, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)

pheatmap(sampleDistMatrix,
          clustering_distance_rows=sampleDists,
          clustering_distance_cols=sampleDists,
          col=colors,show_rownames = F)
```



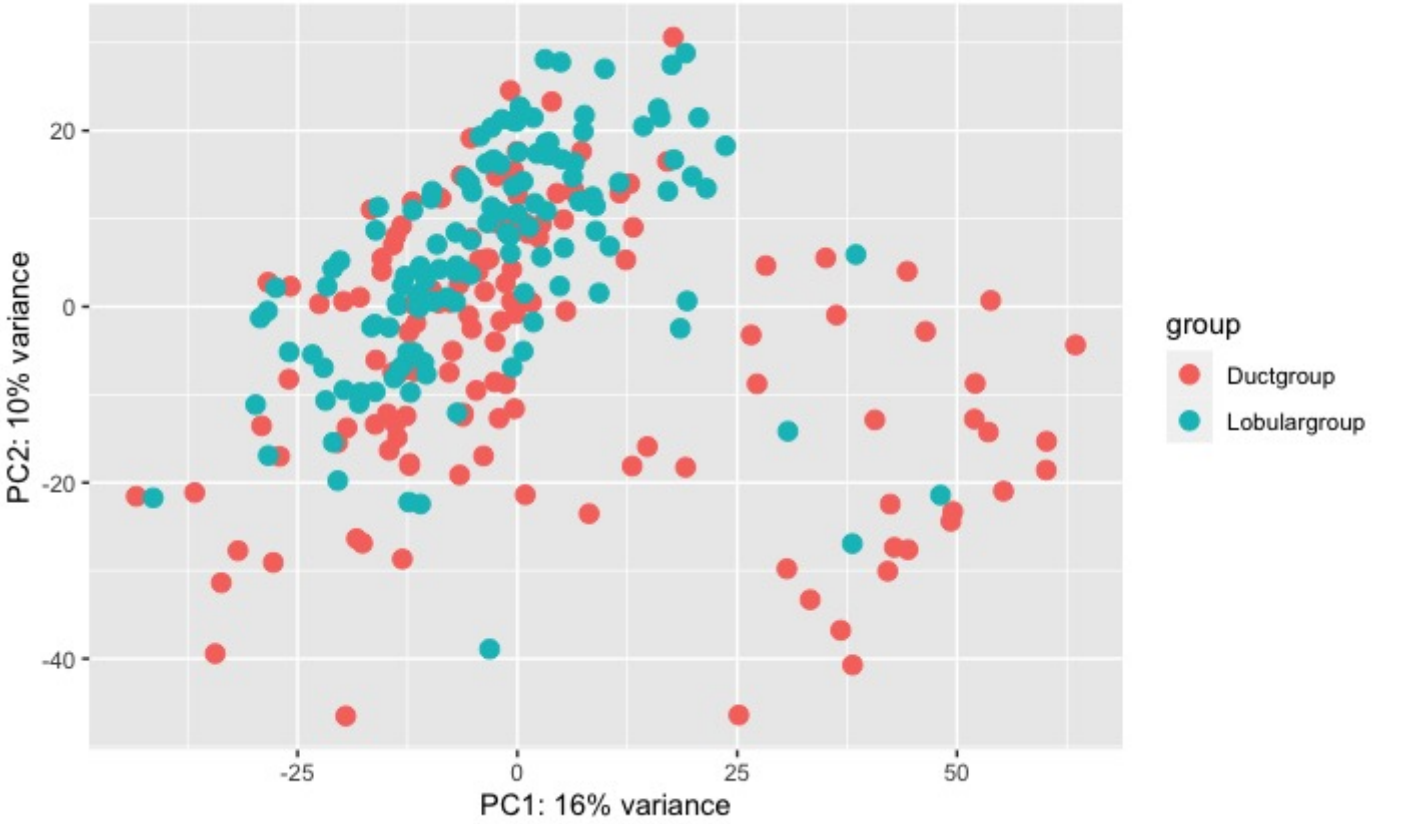
This heatmap shows the similarity between samples.

- PCA plot

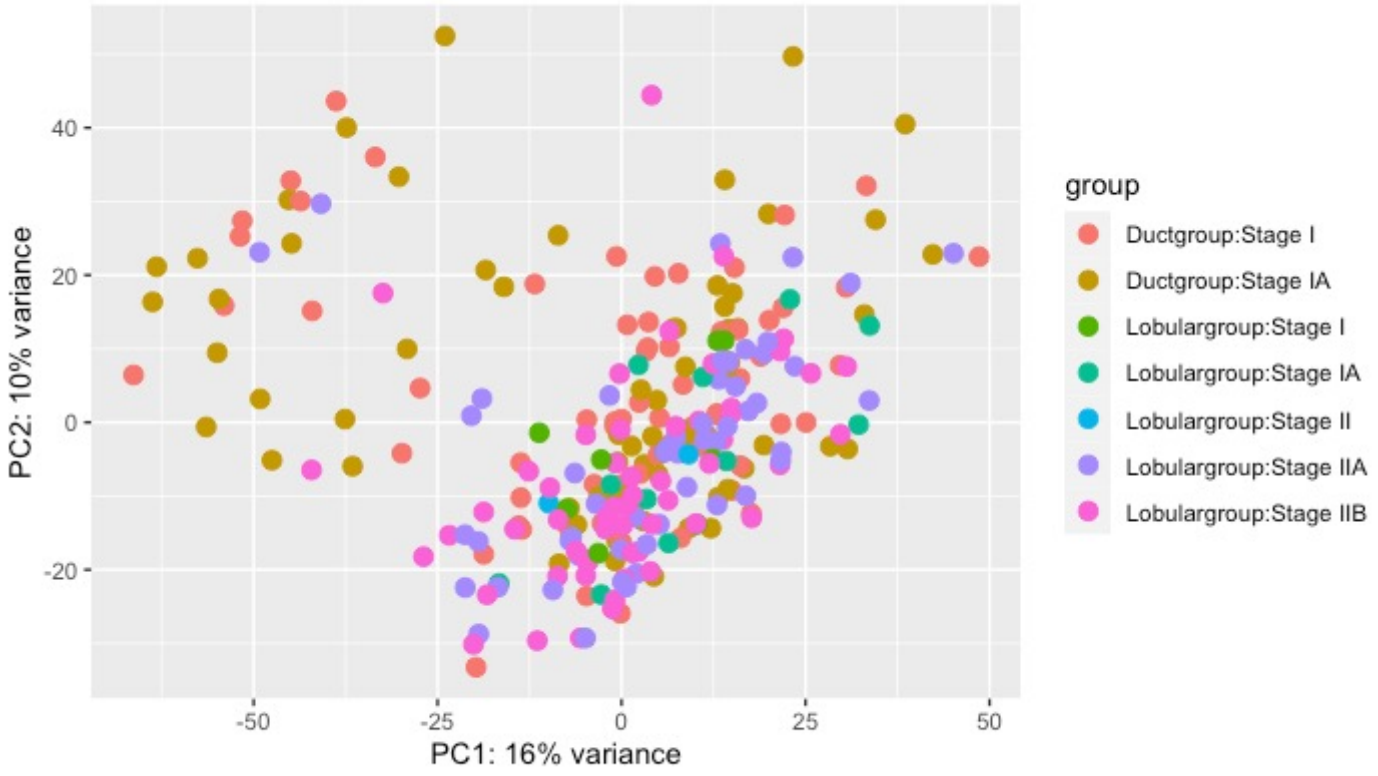
Principal component plot of the samples
Related to the distance matrix is the PCA plot, which shows the samples in the 2D plane spanned by their first two principal components. This type of plot is useful for visualizing the overall effect of experimental covariates and batch effects.

*Note: because PCA plot drawn only by factor conditoin did not show any pattern , so I add the Stage as New Factor to use both to draw PCA plots *

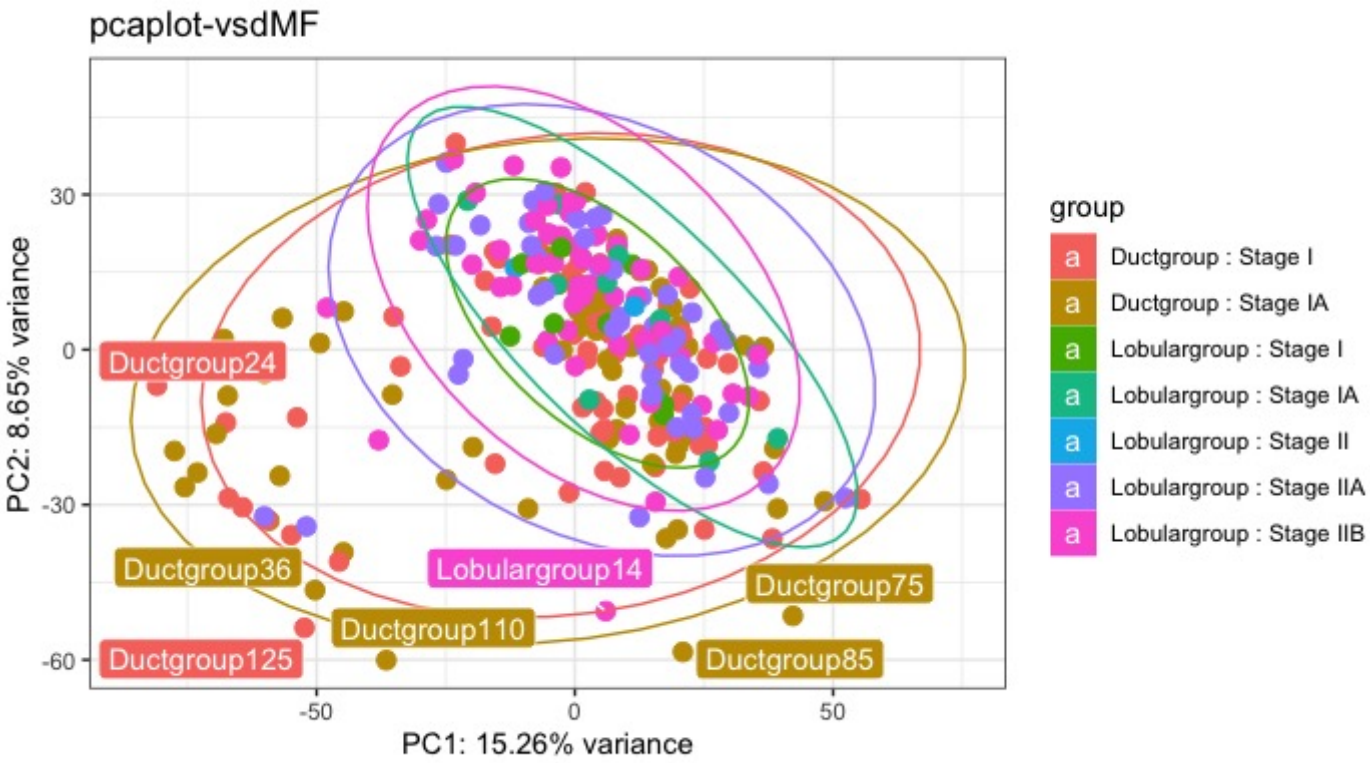
- PCA plot (with only condition as Factor)



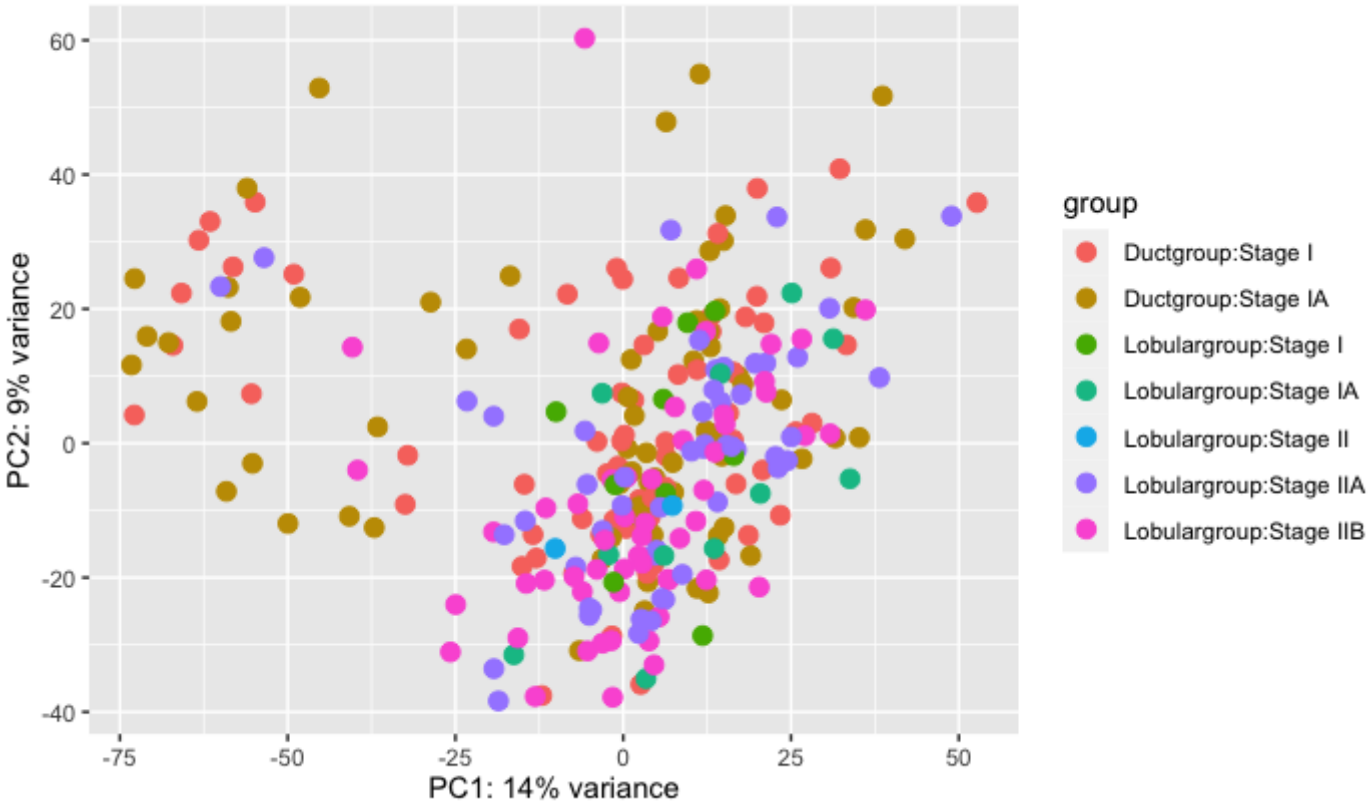
- `plotPCA` vsdMF



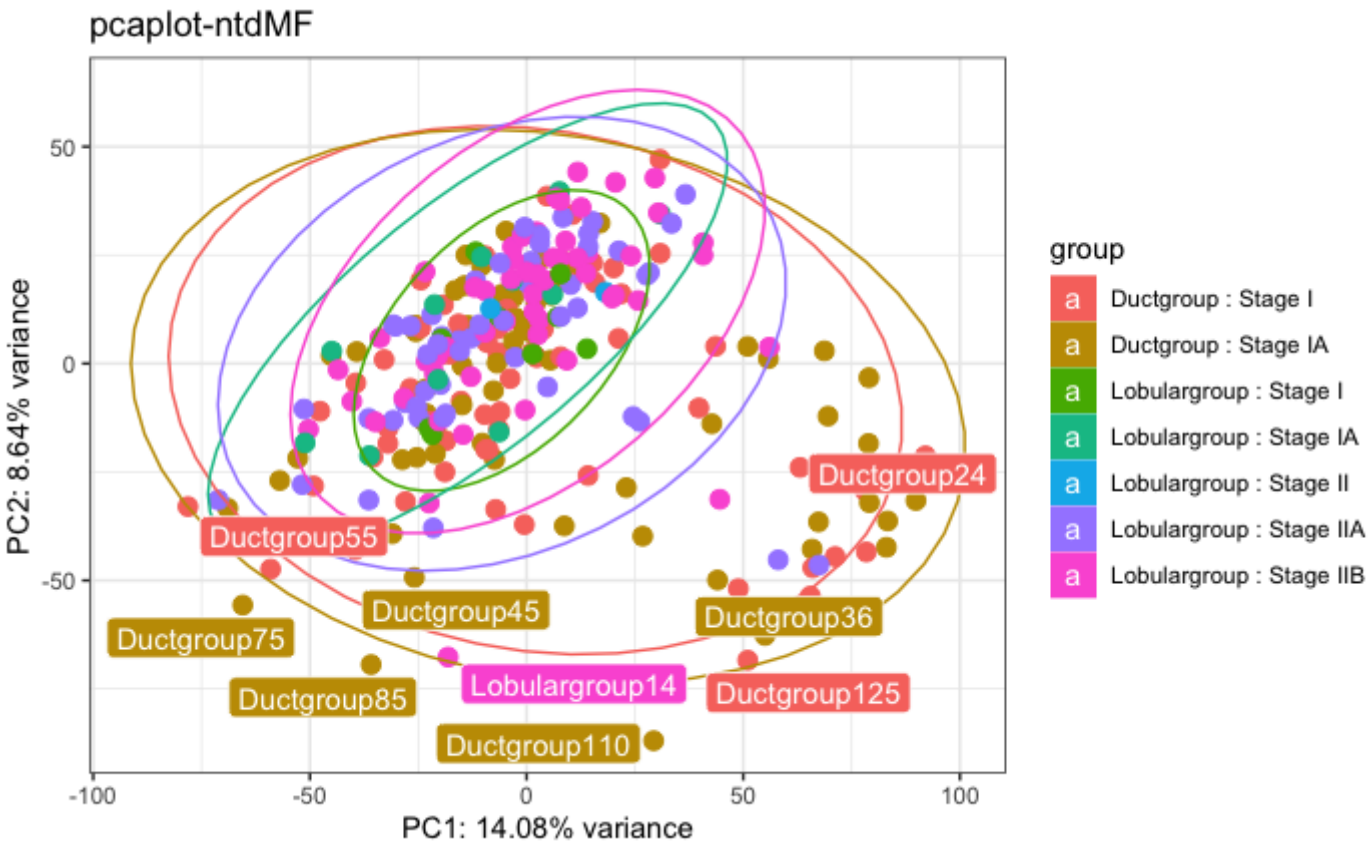
- `pcaplot` vsdMF



- `plotPCA` ntdMF



- `pcaplot ntdMF`



Although I have add this new factor , the pattern is still not clear, I will put it in my `Know Issues`

Vignettes are uploaded there

vignette

Transfer Names

Done by this script Name_Transfer(Genome & DEG).Rmd

Map EnsemblD to Gene ID in Genome

I transfered the Ensembl ID to gene ID and remove the version number in Ensembl ID in all the files

- 1. Make readble gtf file in R

```
if(!require("rtracklayer")) BiocManager::install("rtracklayer")
gtf1 <- rtracklayer::import("Homo_sapiens.GRCh38.104.chr.gtf")
gtf_df <- as.data.frame(gtf1)
```

- 2. Use AnnotationDbi package to do ID transfer

```
get_map = function(input) {
  if (is.character(input)) {
    if(!file.exists(input)) stop("Bad input file.")
    message("Treat input as file")
    input = data.table::fread(input, header = FALSE)
  } else {
    data.table::setDT(input)
  }

  input = input[input[[3]] == "gene", ]

  pattern_id = ".*gene_id \"([^;]+)\";.*"
  pattern_name = ".*gene_name \"([^;]+)\";.*"


  gene_id = sub(pattern_id, "\\1", input[[9]])
  gene_name = sub(pattern_name, "\\1", input[[9]])

  Ensembl_ID_TO_Genename <- data.frame(gene_id = gene_id,
                                       gene_name = gene_name,
                                       stringsAsFactors = FALSE)
  return(Ensembl_ID_TO_Genename)
}

Ensembl_ID_TO_Genename <- get_map("gencode.v29.annotation.gtf")
```

- 3. Remove the version number of EnsembleID

```
gtf_Ensembl_ID <- substr(Ensembl_ID_TO_Genename[,1],1,15)
Ensembl_ID_TO_Genename <- data.frame(gtf_Ensembl_ID,Ensembl_ID_TO_Genename[,2])

colnames(Ensembl_ID_TO_Genename) <- c("Ensembl_ID","gene_id")
```

- 4. Save the file as "Ensembl_ID_TO_Genename.csv".

Ensembl_ID_TO_Genename.csv

Transfer the name in DEG

- 1. Read DEG file

```
diff_gene_deseq2 <- read.csv("~/Results/New_DEG_Lobular_Duct.csv")
```

- 2. Change column name

```
colnames(diff_gene_deseq2)[1] <- "gene_id"
```

- 3. Remove the version number

```
library(tidyr)
diff_gene_deseq2 <- diff_gene_deseq2 %>%
     tidyr::separate(gene_id,into = c("gene_id"),sep = "\\.")
```

- 4. Get gene symbol ID in DEG

```
library(AnnotationDbi)
library(org.Hs.eg.db)
diff_gene_deseq2$symbol <- mapIds(org.Hs.eg.db,
                          keys=diff_gene_deseq2$gene_id,
                          column="SYMBOL",
                          keytype="ENSEMBL",
                          multiVals="first")
```

- 5. Remove duplicated genes

```
library(dplyr)
diff_gene_deseq2 <- diff_gene_deseq2 %>%
## Remove NA
filter(symbol!="NA") %>%
## Remove Duplicate
distinct(symbol,.keep_all = T)
```

Number has been reduced to 309 from 463

- 6. Save the correspondence between Gene id and Ensembl ID

```
DEG_Ensemble_Symbol <- diff_gene_deseq2[,-c(2:7)]
write_csv(DEG_Ensemble_Symbol, "DEG_Ensemble_Symbol.csv")
```

DEG_Ensemble_Symbol.csv

- 7. Remove the ensemble ID

```
diff_gene_deseq2$gene_id <- diff_gene_deseq2$symbol
diff_gene_deseq2 <- diff_gene_deseq2[,-8]
```

- 8. Export Results to csv file

```
write.csv(diff_gene_deseq2,"New_symbolID_refiltered.csv",row.names = F)
```

New_symbolID_refiltered.csv

IPA Analysis

- Step1: Data Preparation
  - 1. Put the Ensembl ID of UP-Regulated genes into a Excel file
  - 2. Put the Ensembl ID of Down-Regulated genes into a Excel file
- Step2 : IPA Analysis
  - 1. Selcet Human as species to do Core Analysis
  - 2. Save the PATHWay Results

Results

Click to see the Results

IPA_Down.csv

IPA Up.csv

In the `IPA_Up` files,Find 0 Pathways related to Breast

In the `IPA_Down` files, Find 8 Pathways related to Breast, Check the information here breast_concerned.csv

Related Molecules are FABP7, PI3, CDH1, ELF5, CA9, MAGEA4 and CDH1.

I want to focus on `MAGEA4` whose Category is `Cancer,Organismal Injury and Abnormalities,Reproductive System Disease` and Disease/Function Annotation is `HER2 non-overexpressing breast carcinoma`

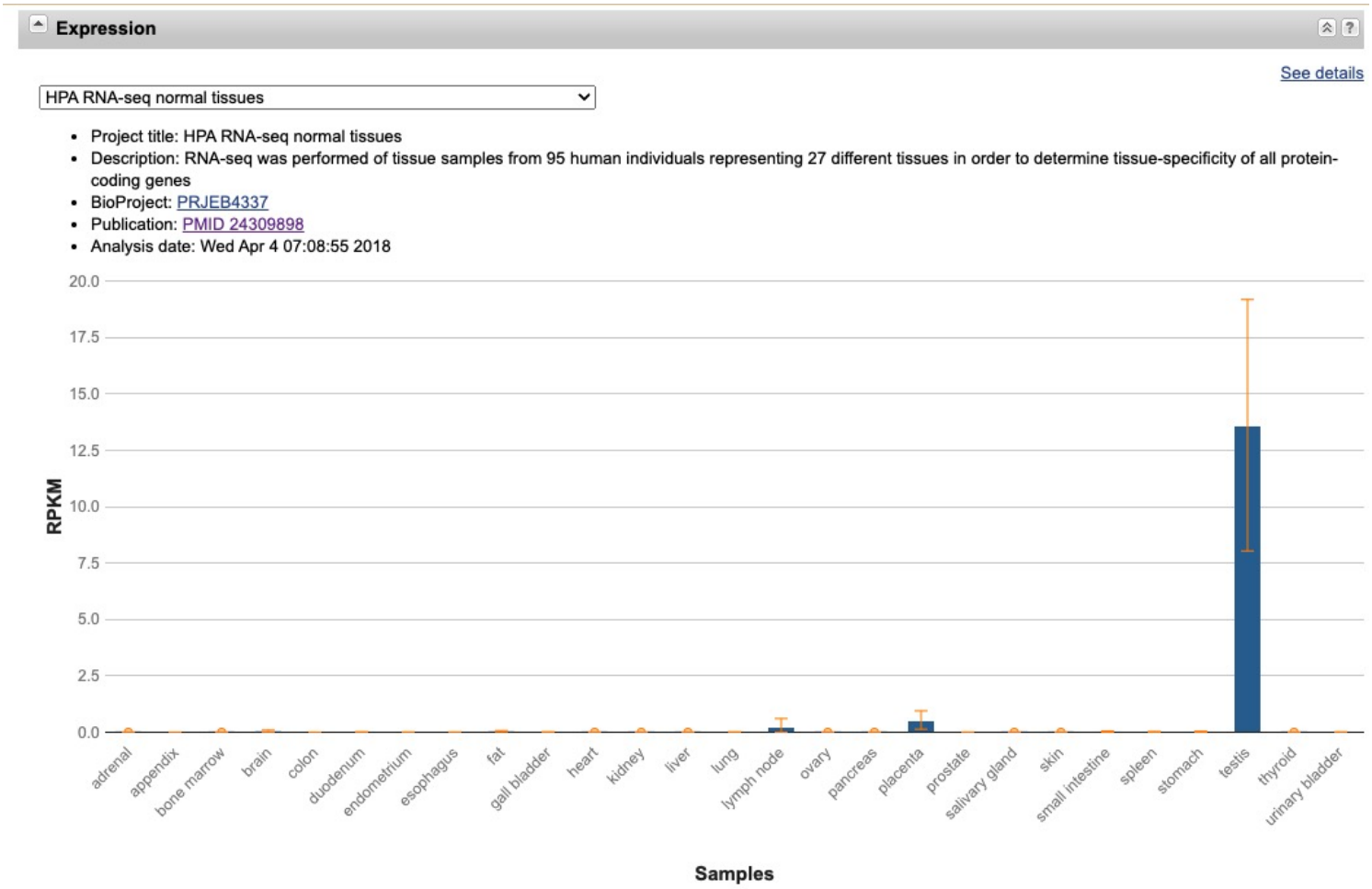
MAGEA4

- Basic Information:
  This gene is a member of the MAGEA gene family. The members of this family encode proteins with 50 to 80% sequence identity to each other. The promoters and first exons of the MAGEA genes show considerable variability, suggesting that the existence of this gene family enables the same

function to be expressed under different transcriptional controls. The MAGEA genes are clustered at chromosomal location Xq28. They have been implicated in some hereditary disorders, such as dyskeratosis congenita. Several variants encoding the same protein have been found for this gene. [provided by RefSeq, Aug 2020]

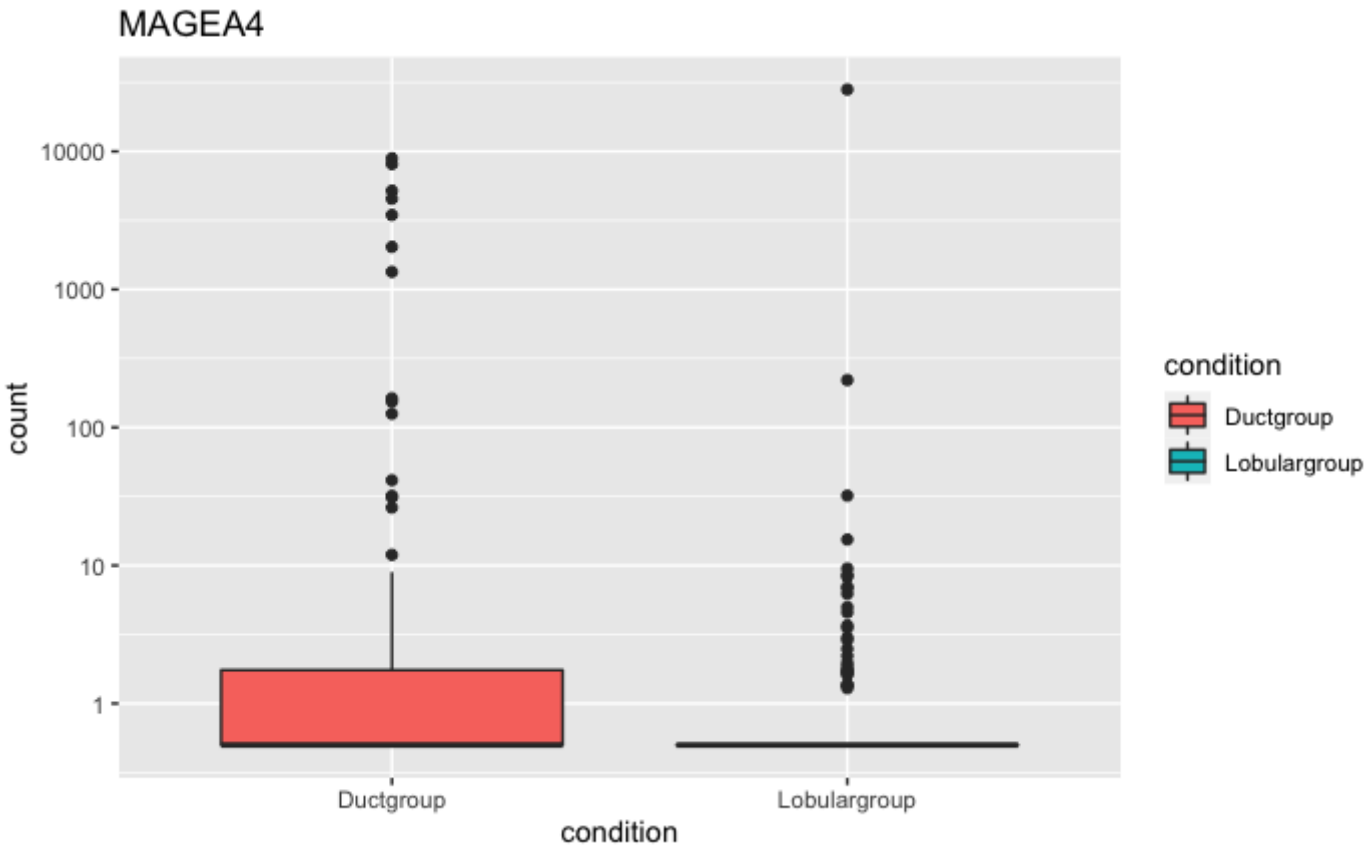
Ref: https://www.ncbi.nlm.nih.gov/gene/4103

`MAGEA4` is mostly expressed in testis in normal tissues



Cancer/testis antigens (CTAs) are expressed in a large variety of tumor types, whereas their expression in normal tissues is restricted to male germ cells, which are immune-privileged because of their lack of or low expression of human leukocyte antigen (HLA) molecules

- Compare 2 groups - BoxPlot:



Expressions between 2 groups are significantly different

Expressions in Ductgroup are signicantly more than them in Lobulargroup

- Previous Study :

Proteomic Profiling of Triple-negative Breast Carcinomas in Combination With a Three-tier Orthogonal Technology Approach Identifies Mage-A4 as Potential Therapeutic Target in Estrogen Receptor Negative Breast Cance

Summary

Input Summary :

TCGA-BRCA in TCGA_Portal

- Group Loular Carcinoma
  - Cancer: Breast cancer
  - Stage : stage I, stage IA, stage IB, stage II, stage IIA. stage IIB
  - Diagnosis : Lobular Caricinoma
  - Sample Type: Primary tumor
  - WorkFlow: HTSeq - Counts

    - Data Category : transcriptome profilling
    - Number:130 Files& 130Cases
- Group Infiltrating Duct Carcinoma
    - Cancer: Breast cancer
    - Stage : stage I, stage IA, stage IB
    - Diagnosis : Infiltrating Duct Carcinoma
    - Sample Type: Primary tumor
    - WorkFlow: HTSeq - Counts
    - Data Category : transcriptome profilling
    - Number:135 Files & 135 Cases

Differential Expressed Analysis by `DESeq2`

- Differential Expressed Genes
    - Define : padj <= 0.05, abs(log2FoldChange) >= 1.5
    - DEG : 309 (After removing Duplicated & NA)
    - Up-Regulated : 59
    - Down-Regulated: 250

IPA Analysis

- Find interesting gene `MAGEA4`
- Find previous study about its relationship with Breast Cancer

Known Issues

- 1. Have not found which Factor is driven the PCA plot, Factor `Stage` is not the most significant facor.
- 2. Pattern in Heatmap is not obvious
- 3. Have not delved into the Biological Significance of the results.