Differential gene expression in TCGA within stage 1 lung cancers occurring in lower and upper lobe using DeSEQ2

Bohan Zhang

11/23/2021

Milestone 1

1.Initial screening of data

1.1 Screening of Files

Data Category <- transcriptome profiling

Experimental Strategy <- RNA-Seq

Workflow Type <- HTSeq - Counts

Access <- open

1.2 Screening of Upper Lobe Lung Cancer Cases

Diagnoses Ajcc Pathologic Stage <- stage ia/stage ib/stage i

Diagnoses Tissue or Organ of Origin <- upper lobe, lung

Primary Site <- bronchus and lung

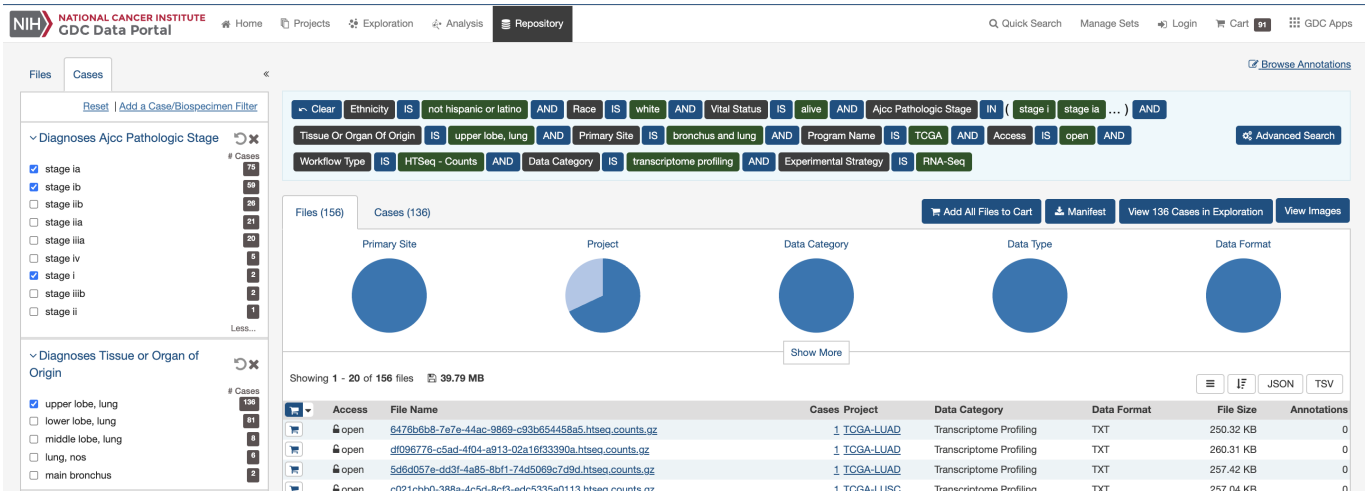
Program <- TCGA

Vital Status <- alive

Race <- white

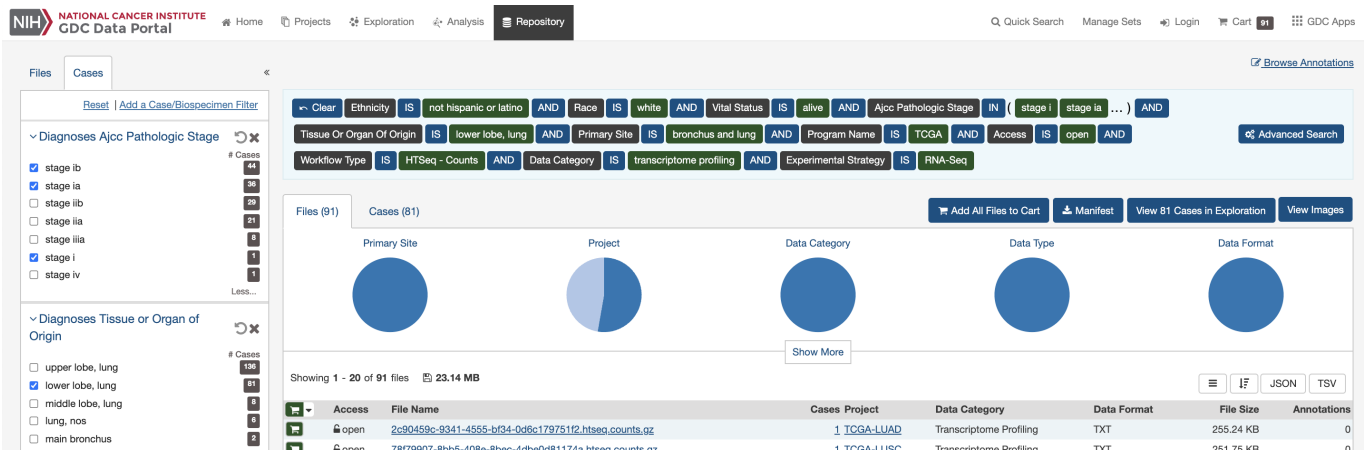
Ethnicity <- not hispanic or latino

Now, I have filtered out 156 files and 136 cases. The number of files and cases is different because some cases have duplicate files; however, since I will be downloading files, the filtering in this step is incomplete. The second filtering will be done in the later steps to remove the duplicate files.



1.2 Screening of Lower Lobe Lung Cancer Cases

The filtering method is similar to Upper lobe lung cancer, except that the Diagnoses Tissue or Organ of Origin is changed to lower lobe, lung. here, I filtered 91 files and 81 cases. again, in the next steps I will do a secondary filter to remove duplicate files.



2.Data download and collation

2.1 Installation and Configuration of gdc-client

Gdc-client is a tool used to download files from the GDC website. I went to the download page of gdc-client and selected GDC Data Transfer Tool Client's OSX version to download and install.

Add the path of the software installation to `.zshrc` by adding a line to the .zshrc file

```
export PATH="directory path:$PATH"
```

Type `gdc-client -version` in the terminal to check if the software is installed successfully.

2.1 Download of Count files

Here I click on the `Manifest` button to download a summary txt file with all the file names. Type the following command in the terminal to download all the files.

```
gdc-client download -m gdc_manifest.2021-11-11.txt
```

Here, I created two new directories `gdc_upper` and `gdc_lower` to download the files of upper lobe lung cancer and lower lobe lung cancer respectively.

2.2 Organizing count files

Here, I see that the downloaded files are not count.gz files but folders, so I use `R` to aggregate all the count.gz files into one folder. Taking upper lobe lung cancer as an example, I go to the `gdc_upper` directory in R, create a `gdc_upper_counts` directory, and run.

```
i <- list.dirs()

i
```

Now I can see all the folders in that directory and I find that the number of folders is greater than 156, this is because some of the downloaded folders contain subfolders. I'll ignore these subfolders to put all the files in my newly created `gdc_upper_counts` directory.

```
m = i[2:183]
for(n in m){
  x.path=paste(n,list.files(n),sep='/')
  file.copy(x.path,'./GDC_upper',recursive = T)}
```

Now, organize the files in the `gdc_upper_counts` directory and keep only 156 Counts compressed files. Then do the same for lower lobe lung cancer and now I get two directories `gdc_upper_counts` and `gdc_lower_counts` which contain all the count.gz files I need.

2.3 Metadata json file download

Add the files of upper lobe lung cancer and lower lobe lung cancer to the `cart` separately, and click the `Metadata` button to download the `json` file. The role of this file is to convert the count file name to the data number of TCGA, which will be used in the secondary screening later.

Metadata json file of lower

Metadata json file of upper

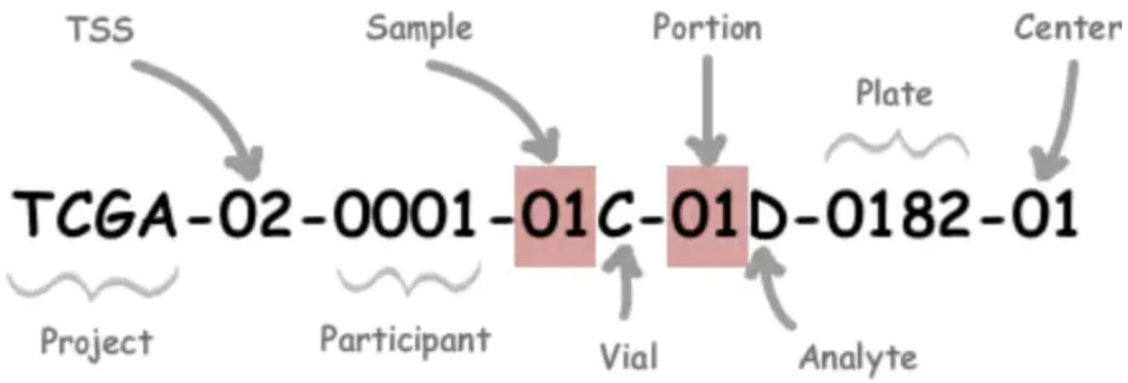
3. Secondary screening of data

Now, I have obtained the count files for upper lobe lung cancer and lower lobe lung cancer. However, the duplicate files mentioned before still exist in them. I need to find them out and delete them.

First, I need to know the meaning of TCGA data number. As shown in the figure, the TCGA data number consists of 7 parts. The third part represents the patient number, so I first need to delete the extra files of the same patient according to the patient number.

Second, if the number in the third part is greater than 10, it means the sample is a normal sample rather than a tumor sample; so I need to sun all samples with that number greater than 10.

Third, the letter in the third part indicates the sample quality. I choose to keep only the samples with quality A.



I wrote a bash script to replace the filenames of the previously downloaded counts files with the TCGA data numbers, and compared these files in Finder and recorded the numbers of the files to be deleted. The following is an example of the operation with lower lobe lung cancer.

3.1 Write a file name and TCGA data number mapping file with R

Lowerlobe mapping file building

```
meta <- jsonlite::fromJSON("metadata.cart.2021-11-09_lower.json")
View(meta)
```

```
ID = sapply(ids, function(x){x[,1]})
file2id_lower = data.frame(file_name = meta$file_name, ID= ID)
View(file2id)
write.table(file2id_lower, file = "sample2id_lower.txt", sep = "\t",
col.names = F, quote = F, row.names = F)
```

Here is the file of lower

Here is the file of upper

3.2 Use this mapping file to batch rename files via bash script

Batch rename lowerlobe files

```
#!/bin/bash

 cat $1 |while read line
 do
   arr=($line)
   filename=${arr[0]}
```

```
    submitterid=${arr[1]}
    gunzip -c ./${filename} > ./file_lower/${submitterid}.count
  done
```

Batch rename upperlobe files

```
#!/bin/bash

 cat $1 |while read line
 do
   arr=($line)
   filename=${arr[0]}
   submitterid=${arr[1]}
   gunzip -c ./${filename} > ./file_upper/${submitterid}.count
 done
```

Bash script usage

First, create a new `file2id_lower` directory in the `gdc_lower_counts` directory.

Terminal run `bash change_name.sh sample2id_lower.txt`

The files whose names are replaced are stored in the `file2id_lower` directory.

3.3 Deletion of unwanted files

The TCGA data numbers of the unwanted files were recorded and deleted according to the 3 selection principles described previously.

22 files were selected from Upper lobe lung cancer that needed to be removed. 10 were selected from Lower lobe lung cancer. So, the final number of Counts files for both is 134:81.

The file name of the deleted files

4.Load count files into the vignette

4.1 Adding prefix to the count files using a bash script

Because the files need to be loaded together when loading into the vignette. I use a bash script to add the prefix `upperlobe-` and `lowerlobe-` to the count files of upper lobe lung cancer and lower lobe lung cancer respectively. For example: `upperlobe-TCGA-NJ-A55R-01A-11R-A262-07.count`. This way I can put both sets of count files into the same directory `file_all` and use `regular expressions` to take out "upperlobe" and "lowerlobe" from the file names as conditions.

The script of lowerlobe

```
#!/bin/sh
for files in $(ls *.count)
```

```
    do mv $files "lowerlobe-"$files
done
```

The script of upperlobe

```
#!/bin/sh
for files in $(ls *.count)
    do mv $files "upperlobe-"$files
done
```

4.2 Loading all count files into vignette

I used R to do the loading of the files. First I created a value and saved the path to the file_all directory in it.

Import the filenames of the files in the directory into the `sampleFiles` value.

Use a regular expression to get the upperlobe or lowerlobe of the file name into the `sampleCondition` value.

Create a dataframe named `sampleTable` with three columns for `sampleName`, `fileName` and `condition`.

Using `DESeq2` package, enter all count files into vignette and create `dds`.

```
directory <- "~/myproject/file_all"
```

```
sampleFiles <- grep("lobe",list.files(directory),value=TRUE)
```

```
sampleCondition <- sub("(.*lobe).*","\\1",sampleFiles)
```

```
sampleTable <- data.frame(sampleName = sampleFiles,
                          fileName = sampleFiles,
                          condition = sampleCondition)
sampleTable$condition <- factor(sampleTable$condition)
```

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

Here is the sampleTable

Information of dds

```
R 4.1.1 · ~/
> dds
class: DESeqDataSet
dim: 60483 215
metadata(1): version
assays(1): counts
rownames(60483): ENSG00000000003.13 ENSG00000000005.5 ...
  ENSGR0000280767.1 ENSGR0000281849.1
rowData names(0):
colnames(215): lowerlobe-TCGA-18-3421-01A-01R-0980-07.count
  lowerlobe-TCGA-18-4721-01A-01R-1443-07.count ...
  upperlobe-TCGA-NJ-A4YQ-01A-11R-A262-07.count
  upperlobe-TCGA-NJ-A55R-01A-11R-A262-07.count
colData names(1): condition
```

Htseq_input.Rmd* × dds ×

Show Attributes

Name	Type	Value
dds	S4 [60483 x 215] (DESeq2::DI	S4 object of class DESeqDataSet
design	formula	~condition
dispersionFunction	function	function() { ... }
rowRanges	S4 (GenomicRanges::Compres	S4 object of class CompressedGRangesList
colData	S4 [215 x 1] (S4Vectors::DFra	S4 object of class DFrame
assays	S4 [60483 x 215] (Summarize	S4 object of class SimpleAssays
NAMES	NULL	Pairlist of length 0
elementMetadata	S4 [60483 x 0] (S4Vectors::DF	S4 object of class DFrame
metadata	list [1]	List of length 1

Milestone 2

5. Generation of differential expression results

5.1 Processing of dds data frames

Pre-filtering

The aim is to remove low-count genes. Here, I keep genes with at least 10 counts.

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

Note on factor levels

The aim is to set a factor for comparing differences in gene expression, i.e. upperlobe lung cancer and lowerlobe lung cancer.

```
dds$condition <- factor(dds$condition, levels =
c("upperlobe","lowerlobe"))
```

```
dds$condition <- droplevels(dds$condition)
```

5.2 Obtain data frames for differential gene expression results

Get Results

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

```
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
-- replacing outliers and refitting for 7234 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)
estimating dispersions
fitting model and testing
```

Here I use Independent hypothesis weighting (IHW) to filter the p-values. I want to filter the differentially expressed genes by p-value < 0.05, so, I add alpha=0.05.

```
library("IHW")
res <- results(dds, filterFun=ihw,
contrast=c("condition","upperlobe","lowerlobe"), alpha=0.05)
res
```

```
log2 fold change (MLE): condition upperlobe vs lowerlobe
Wald test p-value: condition upperlobe vs lowerlobe
DataFrame with 50475 rows and 7 columns
                    baseMean log2FoldChange     lfcSE       stat    pvalue      padj    weight
                   <numeric>      <numeric> <numeric>  <numeric> <numeric> <numeric> <numeric>
ENSG00000000003.13 3530.58158    -0.14349078 0.1152011 -1.2455675 0.2129232  1.000000  0.470066
ENSG00000000005.5     1.83533     0.92124505 0.3983635  2.3125742 0.0207461  0.304791  1.634073
ENSG00000000419.11 1988.56928    -0.02316500 0.0859819 -0.2694171 0.7876088  1.000000  0.470066
ENSG00000000457.12 1042.05870     0.00836846 0.0679910  0.1230819 0.9020422  1.000000  0.424619
ENSG00000000460.15  610.06415    -0.00423868 0.1112523 -0.0380997 0.9696082  1.000000  0.582609
...                       ...           ...       ...        ...       ...       ...       ...
ENSG00000281909.1    0.557662       0.323687  0.365305   0.886074 0.3755778  1.000000   0.00000
ENSG00000281910.1    0.291194      -0.186074  0.524813  -0.354553 0.7229245  1.000000   0.00000
ENSG00000281912.1   60.828380       0.226105  0.145742   1.551401 0.1208055  0.446624   3.94932
ENSG00000281918.1    2.456463      -0.573625  0.230559  -2.487977 0.0128472  0.239031   1.74462
ENSG00000281920.1    6.473478       0.222705  0.220144   1.011633 0.3117135  0.742966   2.29753
```

5.3 Change the Ensembl id in the result table to the gene name

remove the version number of the gene Ensembl number

```
enemble_id <- substr(row.names(res),1,15)
rownames(res) <- enemble_id
```

Add a colum to result table

```
RawCounts <- res
Ensembl_ID <- data.frame(Ensembl_ID = row.names(RawCounts))
rownames(Ensembl_ID) <- Ensembl_ID[,1]
RawCounts <- cbind(Ensembl_ID,RawCounts)
```

Download gencode.v38.basic.annotation.gtf

```
wget
http://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_38/gencod
e.v38.basic.annotation.gtf.gz
```

```
gunzip gencode.v38.basic.annotation.gtf.gz
```

Create a file to associate the ensembl id and gene id

```
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.v38.basic.annotation.gtf")

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")
write.csv(Ensembl_ID_TO_Genename,file = "Ensembl_ID_TO_Genename.csv")
```

Ensembl ID TO Genename.csv

Replace ensembl id with gene id

```
res_g <-merge(Ensembl_ID_TO_Genename,RawCounts,by="Ensembl_ID")
```

Remove unnecessary columns and duplicate gene ids

```
res_g <- res_g[order(res_g[,"gene_id"]),]
index <- duplicated(res_g$gene_id)
res_g <- res_g[!index,]
rownames(res_g) <- res_g[,"gene_id"]
res_g <- res_g[,-c(1:2)]
```

Check the new table

```
head(res_g)
```

Description: df [6 × 7]

	baseMean <dbl>	log2FoldChange <dbl>	lfcSE <dbl>	stat <dbl>	pvalue <dbl>	padj <dbl>	weight <dbl>
5_8S_rRNA	0.1801637	0.330600254	0.9476551	0.34886138	7.271934e-01	1.0000000000	0.0000000
5S_rRNA	0.9584783	-0.419697519	0.2380676	-1.76293457	7.791153e-02	0.7668729647	0.5104703
7SK	288.1952065	-0.009870201	0.1923845	-0.05130455	9.590828e-01	1.0000000000	1.9611284
A1BG	25.0049213	0.392721594	0.1452771	2.70325851	6.866332e-03	0.1141319312	3.6280103
A1BG-AS1	126.8942518	0.416677651	0.1325195	3.14427417	1.664994e-03	0.0649045696	2.2998804
A1CF	5.3990024	1.579033127	0.3120435	5.06029887	4.185998e-07	0.0001915898	2.2975332

6 rows

5.4 Optimization process

Log fold change (LFC) shrinkage

I use the `apeglm` method for LFC shrinkage which is useful for gene visualization and ranking.

```
resultsNames(dds)
```

```
[1] "Intercept"                          "condition_lowerlobe_vs_upperlobe"
```

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

```
using 'apeglm' for LFC shrinkage. If used in published research, please cite:
    Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
    sequence count data: removing the noise and preserving large differences.
    Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
log2 fold change (MAP): condition lowerlobe vs upperlobe
Wald test p-value: condition lowerlobe vs upperlobe
DataFrame with 50475 rows and 5 columns
                    baseMean log2FoldChange      lfcSE    pvalue      padj
                   <numeric>      <numeric>  <numeric> <numeric> <numeric>
ENSG00000000003.13 3530.58158    1.13928e-05 0.00144260 0.2129232  0.733175
ENSG00000000005.5     1.83533   -4.76141e-07 0.00144268 0.0207461  0.325044
ENSG00000000419.11 1988.56928   -2.27604e-04 0.00145166 0.7876088  0.963137
ENSG00000000457.12 1042.05870   -3.55648e-06 0.00144237 0.9020422  0.983322
ENSG00000000460.15  610.06415   -7.31588e-06 0.00144258 0.9696082  0.996177
...                       ...            ...        ...       ...       ...
ENSG00000281909.1     0.557662   -3.19932e-06 0.00144269 0.3755778        NA
ENSG00000281910.1     0.291194    3.22278e-07 0.00144269 0.7229245        NA
ENSG00000281912.1    60.828380   -9.78217e-06 0.00144264 0.1208055  0.633484
ENSG00000281918.1     2.456463    8.99681e-06 0.00144268 0.0128472  0.269497
ENSG00000281920.1     6.473478   -4.69361e-06 0.00144267 0.3117135  0.800859
```

Parallelization

Splitting the work into 4 cores to speed it up.

```
library("BiocParallel")
register(MulticoreParam(4))
```

5.5 View the number of differentially expressed genes based on p-value

Sort the data in the result table according to the p-value.

```
resOrdered <- res_g[order(res_g$pvalue),]
```

Check the number of differentially expressed genes at adjusted p-values less than 0.05.

```
sum(res_g$padj < 0.05, na.rm=TRUE)
```

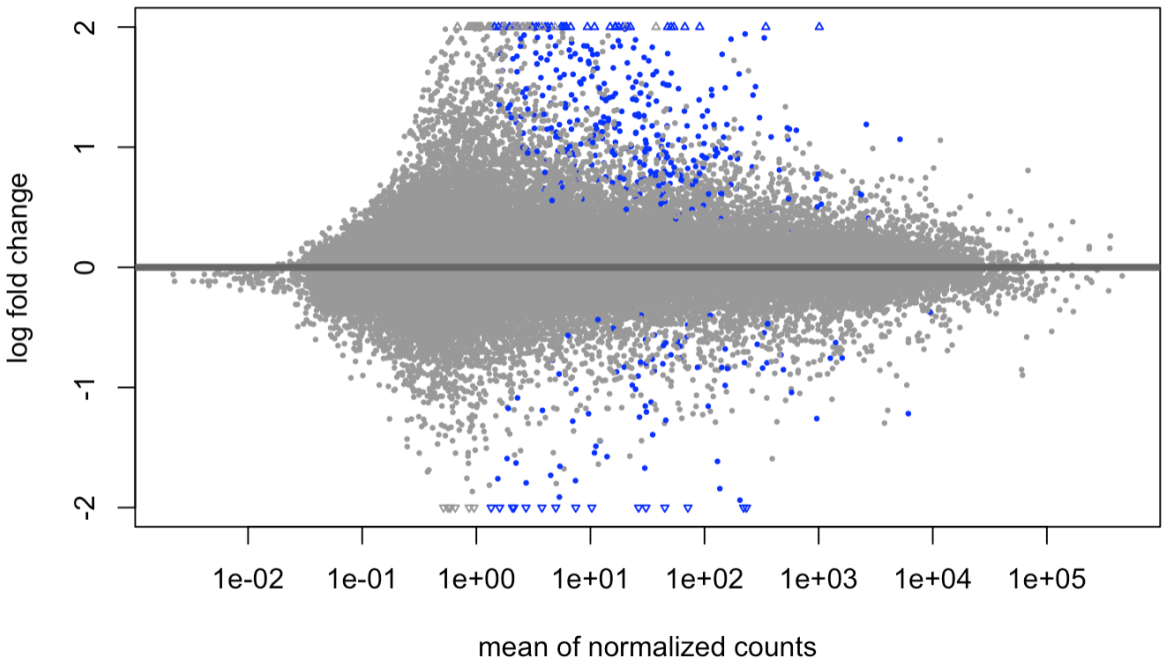
```
[1] 433
```

Therefore, I finally screened 433 genes differentially expressed in upperlobe lung cancer and lowerlobe lung cancer.

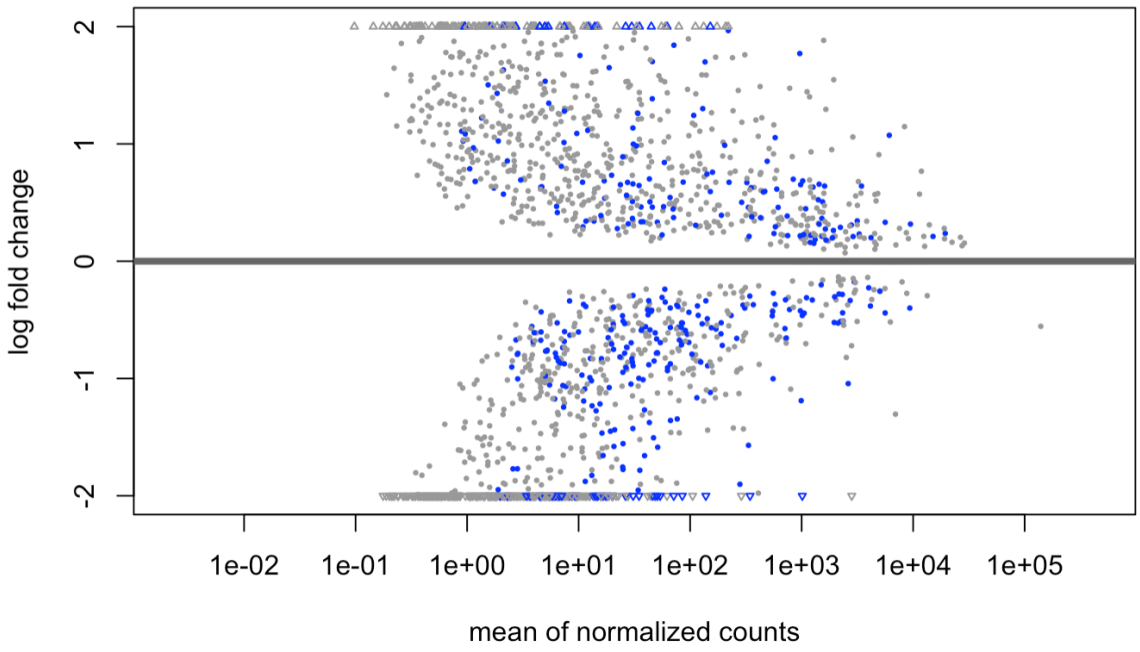
6 Exploring and exporting results

6.1 MA-plot

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



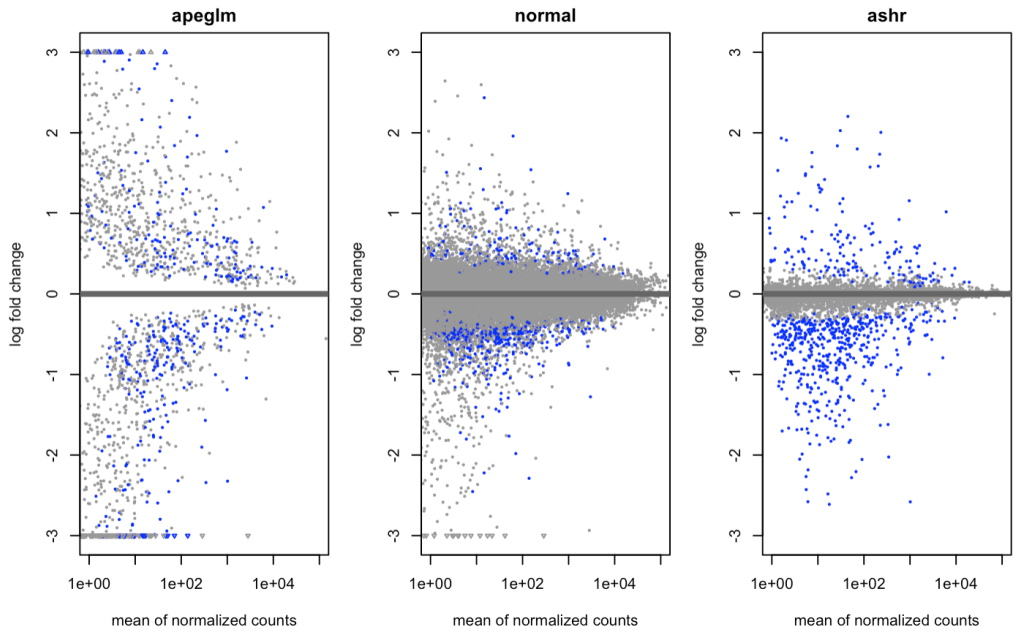

```
plotMA(resLFC, ylim=c(-2,2))
```




```
resNorm <- lfcShrink(dds, coef=2, type="normal")
resAsh <- lfcShrink(dds, coef=2, type="ashr")
```

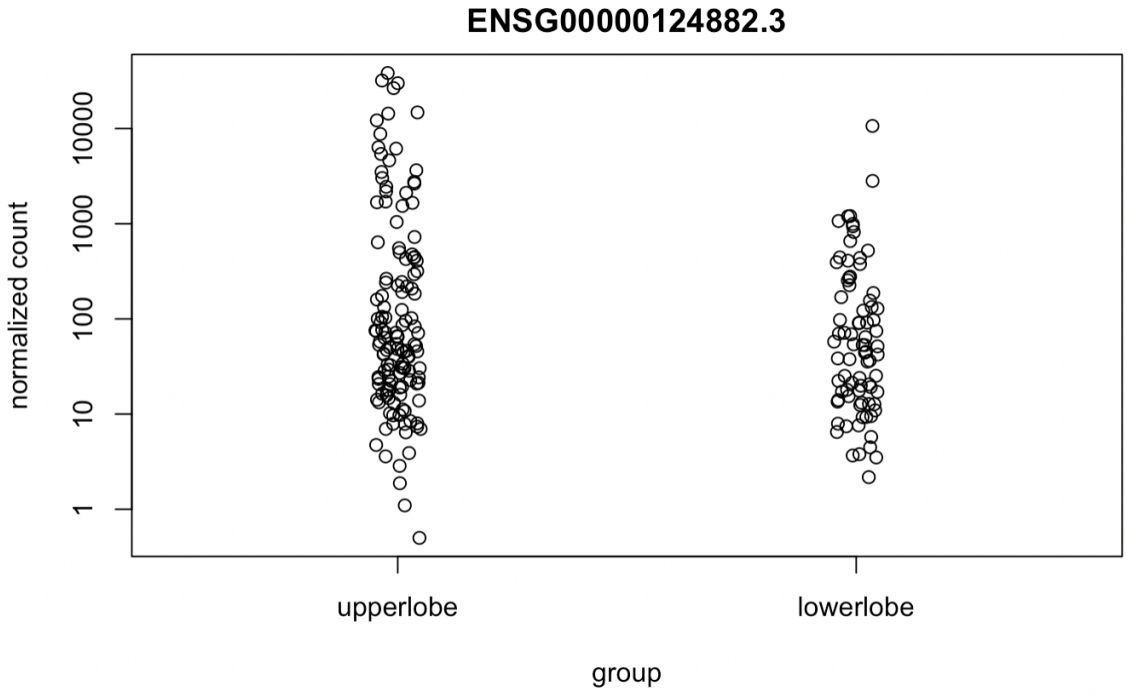
```
using 'normal' for LFC shrinkage, the Normal prior from Love et al (2014).

Note that type='apeglm' and type='ashr' have shown to have less bias than type='normal'.
See ?lfcShrink for more details on shrinkage type, and the DESeq2 vignette.
Reference: https://doi.org/10.1093/bioinformatics/bty895
using 'ashr' for LFC shrinkage. If used in published research, please cite:
    Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.
    https://doi.org/10.1093/biostatistics/kxw041
```

```
par(mfrow=c(1,3), mar=c(4,4,2,1))
xlim <- c(1,1e5); ylim <- c(-3,3)
plotMA(resLFC, xlim=xlim, ylim=ylim, main="apeglm")
plotMA(resNorm, xlim=xlim, ylim=ylim, main="normal")
plotMA(resAsh, xlim=xlim, ylim=ylim, main="ashr")
```

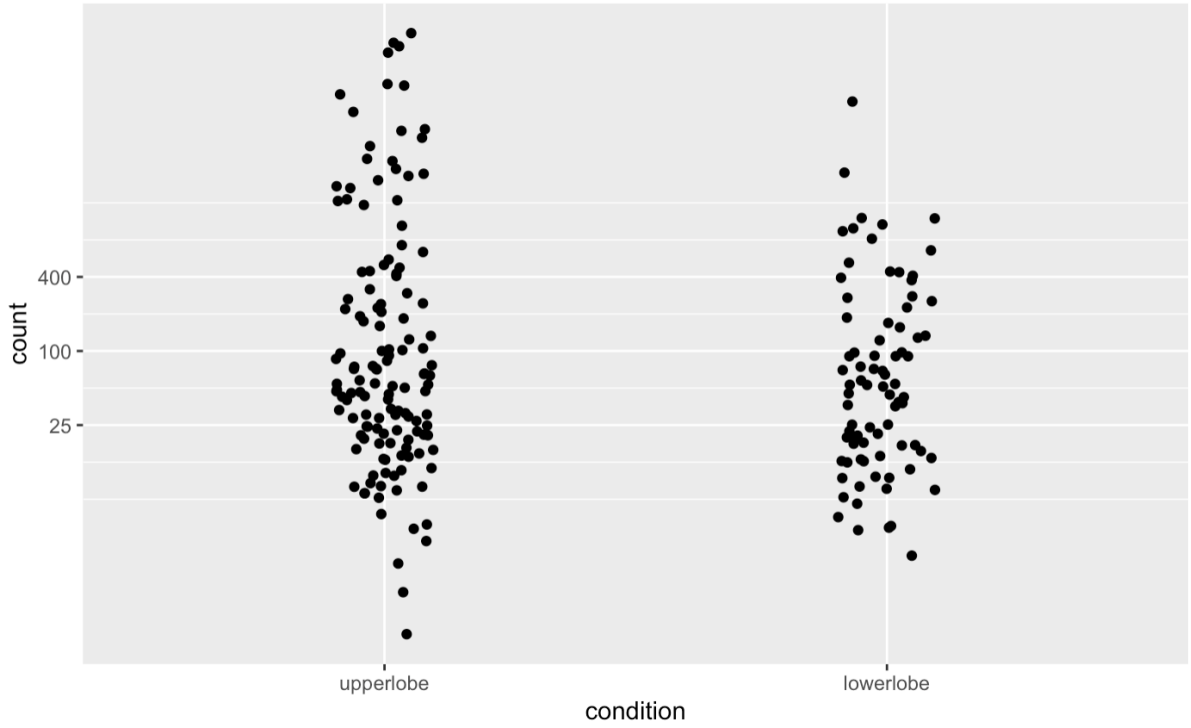


6.2 Plot counts

```
plotCounts(dds, gene=which.min(res$padj), intgroup="condition")
```




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



6.3 More information on results colums

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

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

6.4 Write csv files

Original table of all genes

```
getwd()
write.csv(as.data.frame(res_g),
          file="res_g.csv")
```

Here is the file res_g.csv

Differential expression genes

Genes with p-values <0.05 were screened and determined as differentially expressed genes. And a new column was created to record the up- or down-regulation of genes.

```
resSig_0.05 <- subset(resOrdered, padj < 0.05)
resSig_0.05[which(resSig_0.05$log2FoldChange > 0), "up_down"] <- "up"
resSig_0.05[which(resSig_0.05$log2FoldChange < 0), "up_down"] <- "down"
resSig_0.05
```

Description: df [433 × 8]

	baseMean <dbl>	log2FoldChange <dbl>	lfcSE <dbl>	stat <dbl>	pvalue <dbl>	padj <dbl>	weight <dbl>	up_down <chr>
EREG	1013.610643	2.9182803	0.36864016	7.916339	2.446067e-15	1.941336e-10	0.4700663	up
LEP	53.853720	2.5488065	0.33551523	7.596694	3.037920e-14	1.941336e-10	3.9493163	up
MIA	30.679116	-2.2230846	0.29413295	-7.558094	4.090172e-14	2.995262e-10	2.2975332	down
CALN1	16.530091	2.8720511	0.39868179	7.203868	5.852796e-13	3.250931e-09	2.2718097	up
LINC00973	17.701868	3.0732690	0.43149463	7.122381	1.060786e-12	4.656169e-09	2.2998804	up
PRAP1	50.859273	2.0169123	0.30346547	6.646266	3.006211e-11	6.970695e-08	3.6280103	up
SUN3	19.288962	2.0573725	0.32417352	6.346516	2.202455e-10	6.990583e-07	2.2718097	up
SOHLH2	34.819258	1.8320743	0.29407374	6.229983	4.664867e-10	8.112545e-07	3.6280103	up
LINC02055	6.114705	2.6003707	0.42224807	6.158396	7.348557e-10	1.950214e-06	2.0013131	up
LHFPL4	90.804845	2.4069284	0.39275024	6.128394	8.877032e-10	1.950214e-06	2.2975332	up

1–10 of 433 rows Previous 1 2 3 4 5 6 ... 44 Next

This table contains the 433 differentially expressed genes I screened and their up- or down-regulation information.

```
getwd()
write.csv(as.data.frame(resSig_p0.05),
          file="differential_expression.csv")
```

Here is the file differential_expression.csv

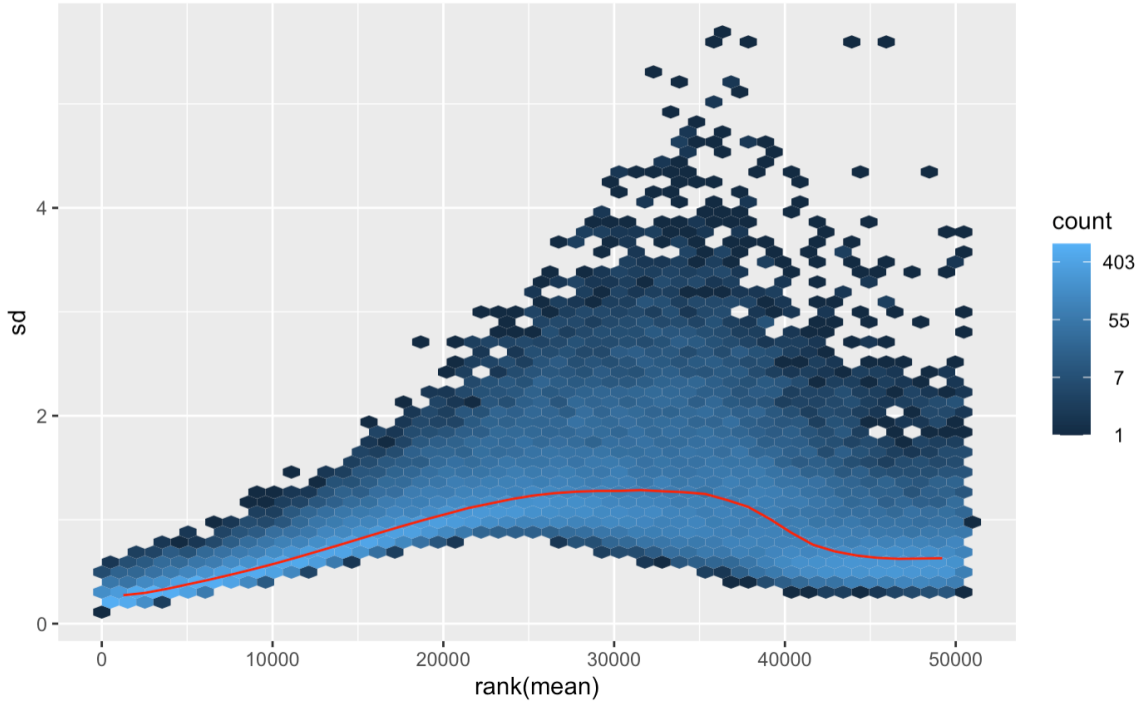
7 Data transformations and visualization

7.1 Extracting transformed values

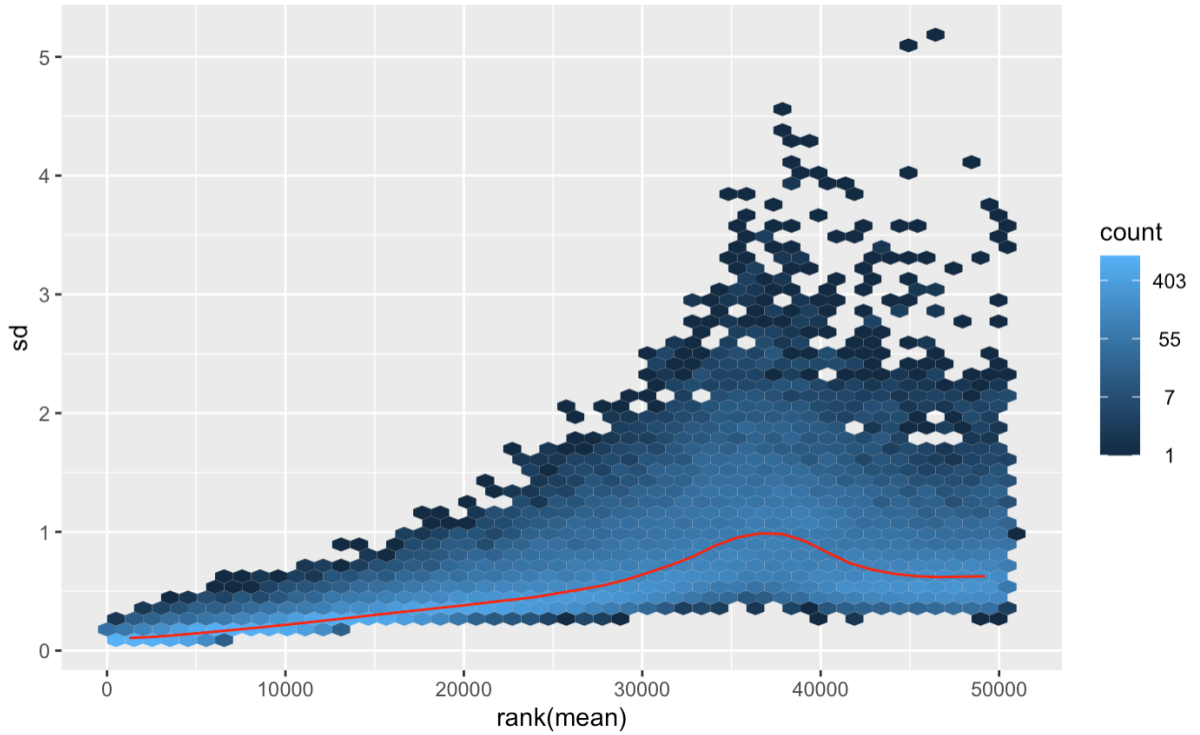
```
vsd <- vst(dds, blind=FALSE)
```

7.2 Effects of transformations on the variance

```
ntd <- normTransform(dds)
library("vsn")
meanSdPlot(assay(ntd))
```




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



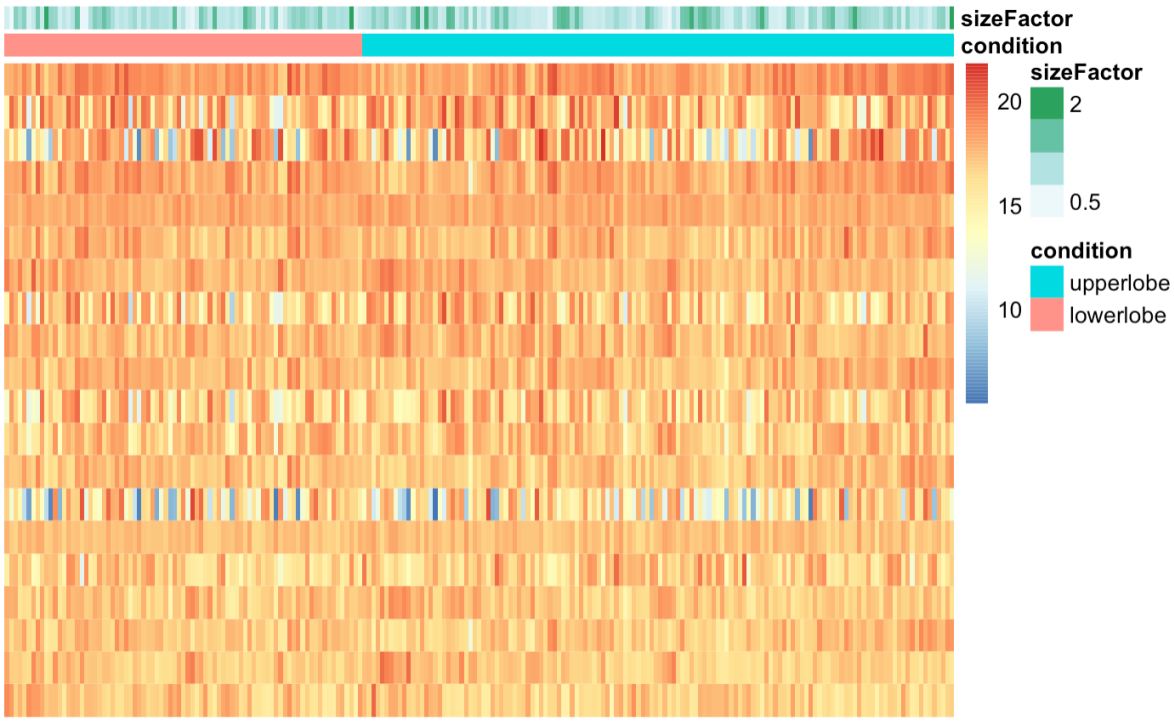
8 Data quality assessment by sample clustering and visualization

8.1 Heatmap of the count matrix

Heatmap of ntd

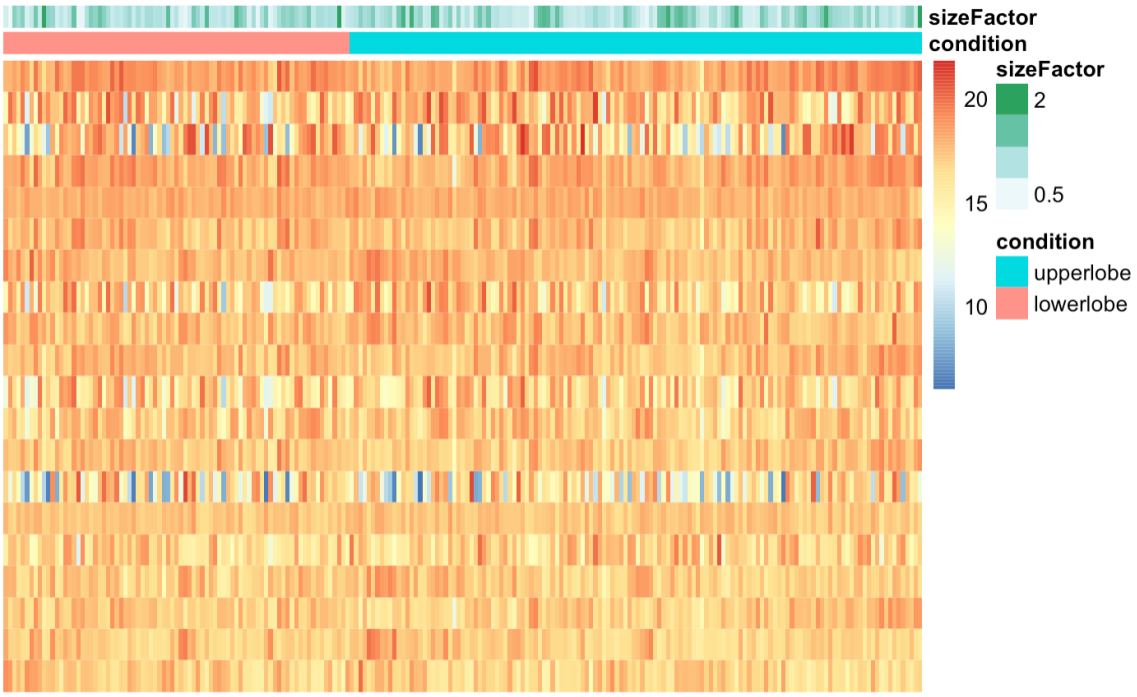
```

```

```
library("pheatmap")
select <- order(rowMeans(counts(dds,normalized=TRUE)),
                decreasing=TRUE)[1:20]
df <- as.data.frame(colData(dds)[,c("condition", "sizeFactor")])
pheatmap(assay(ntd)[select,], cluster_rows=FALSE, show_rownames=FALSE,
         cluster_cols=FALSE, annotation_col=df, show_colnames = FALSE)
```



Heatmap of vsd

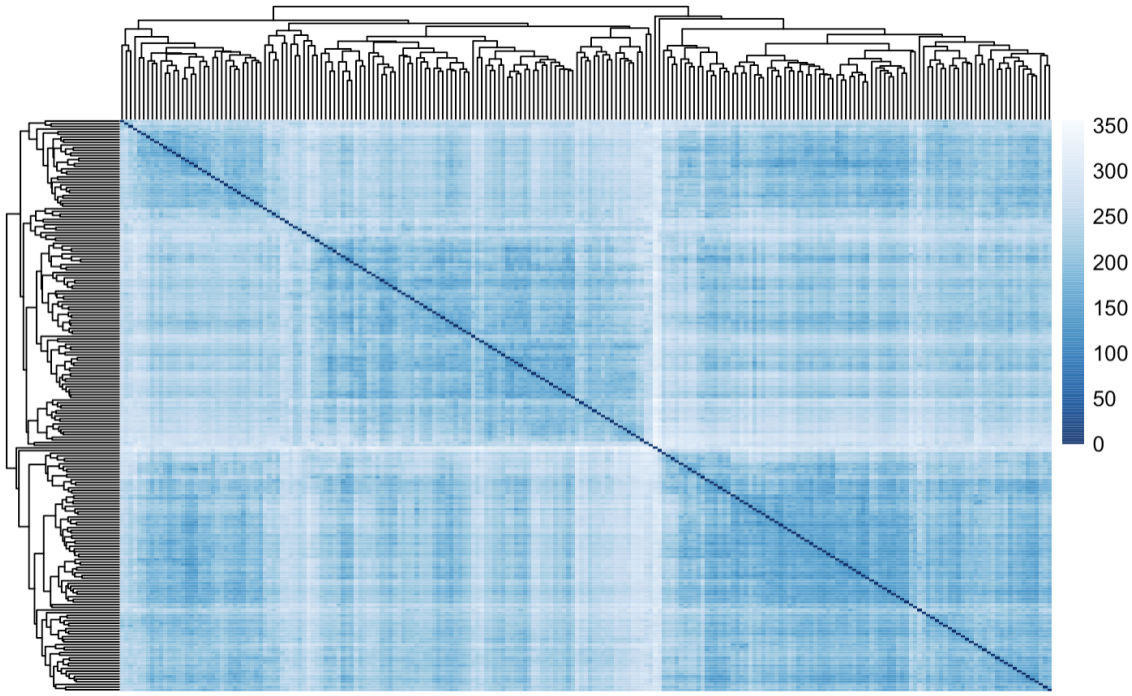
```
pheatmap(assay(vsd)[select,], cluster_rows=FALSE, show_rownames=FALSE,
         cluster_cols=FALSE, annotation_col=df, show_colnames = FALSE)
```



8.2 Heatmap of the sample-to-sample distances

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

```
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$condition, vsd$type, 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 = FALSE)
```



8.3 Principal component plot of the samples

```
plotPCA(vsd, intgroup="condition")
```