

Final_Project

Title

Study the gene expression pattern in TCGA of different age men with prostate cancer using DeSeq2

Author

Cheng-Hsiang Lu

Overview of project

I will study differentially expressed genes between two groups. One group of people is younger than 65 years old, while the other group is older than 65 years old. This analysis will utilize the package DESeq2 and follow the specific vignette: [link](#)

For this analysis, I will use the TCGA cohort and have identified 331 RNA-seq counts files for tumors that fit within my cohort. Saperated by the age of 65 years old, 128 samples are from the group beyond 65 years old and 203 samples are from the group under 65 years old. Within the analysis, I will control for race and primary gleason grade.

Data

I will use the data from [GDC](#) Examining clinical data, there are total 331 cases from 55 to 75 years old, and each group has 128 (65-75 year-olds) and 203 samples (55-64 year-olds).

Milestone_1

Data filtering

First, go to [GDC](#) and click on "**Repository**".

On the left side "**Files**" filters:

Data Category - "**transcriptome profiling**".

Data Type - "**Gene Expression Quantification**".

Experimental Strategy - "**RNA-Seq**".

Workflow Type - "**HTSeq - Counts**".

Access - "**open**".



[Add a File Filter](#)

✓ Search Files



✓ Data Category

☒ transcriptome profiling

Files

17,753

✓ Data Type

☒ Gene Expression Quantification

Files

17,753

✓ Experimental Strategy

☒ RNA-Seq

Files

17,753

✓ Workflow Type

☒ HTSeq - Counts

Files

17,753

☐ HTSeq - FPKM

17,753

☐ HTSeq - FPKM-UQ

17,753

☐ STAR - Counts

6,630

✓ Data Format

☐ txt

Files

17,753

✓ Platform

Files

No data for this field

▼ Access 

☒ open

Files
17,753

On the left side "**Cases**" filters:

First, click "**Add a Case/Biospecimen Filter**"

Then, type "**Gleason Grade**" and select "**primary_gleason_grade**".

Diagnoses Primary Gleason Grade - "**pattern 3**" and "**pattern 4**".

Primary Site - "**prostate gland**".

Program - "**TCGA**".

Disease Type - "**adenomas and adenocarcinomas**".

Gender - "**male**".


Age at Diagnosis - From "**55**" to "**64**".

Vital Status - "**alive**".


Race - "**white**" and "**black or african american**".

Files Cases

[Reset](#) | [Add a Case/Biospecimen Filter](#)

▼ Diagnoses Primary Gleason Grade 

	# Cases
<input checked="" type="checkbox"/> pattern 4	224
<input checked="" type="checkbox"/> pattern 3	184
<input type="checkbox"/> pattern 5	40
<input type="checkbox"/> pattern 2	1

▼ Search Cases 

Upload Case Set

4 / 24

In this group, I got 225 files but only 203 cases.

Later, open a new webpage of [GDC](#). I will select another group with all the same filters except Age at Diagnosis (From "65" to "75"). I got 146 files but only 128 cases.

Data downloading

After selecting all files to Cart in GDC, I have downloaded TCGA data by clicking **Manifest**.

The screenshot shows the GDC Data Portal interface. On the left, there are filters for 'Diagnoses Primary Gleason Grade' (pattern 4, 3, 5) and 'Search Cases' (TCGA-A5-A0G2, 432fe49-2...). The main area displays search results for 'Primary Site' (prostate gland) and 'Program' (TCGA). A table of files is shown with columns: Access, File Name, Cases Project, Data Category, Data Format, File Size, and Annotations. The table lists 203 cases and 225 files, all of which are HTSeq counts from TCGA-PRAD.

Access	File Name	Cases Project	Data Category	Data Format	File Size	Annotations
open	4d4fe2e3-617b-4280-a582-8d5d59420c6a.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	254.07 KB	0
open	d98aa6a3-c555-40d8-adc8-5dc74f24a9e.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	252.81 KB	1
open	a7711d68-2d12-46f4-aaca-68ac2cd921d8.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	253.45 KB	0
open	e36f84b0-a02f-4d11-94a8-a6cac14b1d7.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	250.71 KB	0
open	4839d8d3-b631-44bc-a4d4-2c7ba456596a.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	258.21 KB	0
open	1c3e2b57-9927-4821-9842-c62ed8c2907.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	251.08 KB	0
open	ee7a57f1-661d-4455-86a6-c279588a390a.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	254.44 KB	0
open	ac3e352-b255-4c41-b90f-5e5ed273b06.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	251.35 KB	0
open	2e7b3bb4-a198-4816-b3fa-f23d79574ddd.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	255.56 KB	0
open	e9a165a5-1afe-4dca-b64a-9b4b6bd3ac74.htseq.counts.gz	1 TCGA-PRAD	Transcriptome Profiling	TXT	252.28 KB	0

You have to download "gdc-client" from [GDC Data Transfer Tool](#) by choosing **gdc-client_v1.6.1_OSX_x64.zip** and put it in your work directory and copy your work directory path into the ".zshrc" file like this:

```
vi ~/.zshrc
```

```
export PATH="/path to your gdc-client:${PATH}"
```

Then, after reopening the terminal, please use the command:

```
nohup gdc-client download -m ~/path_to_your_file/your_manifest.txt &
```

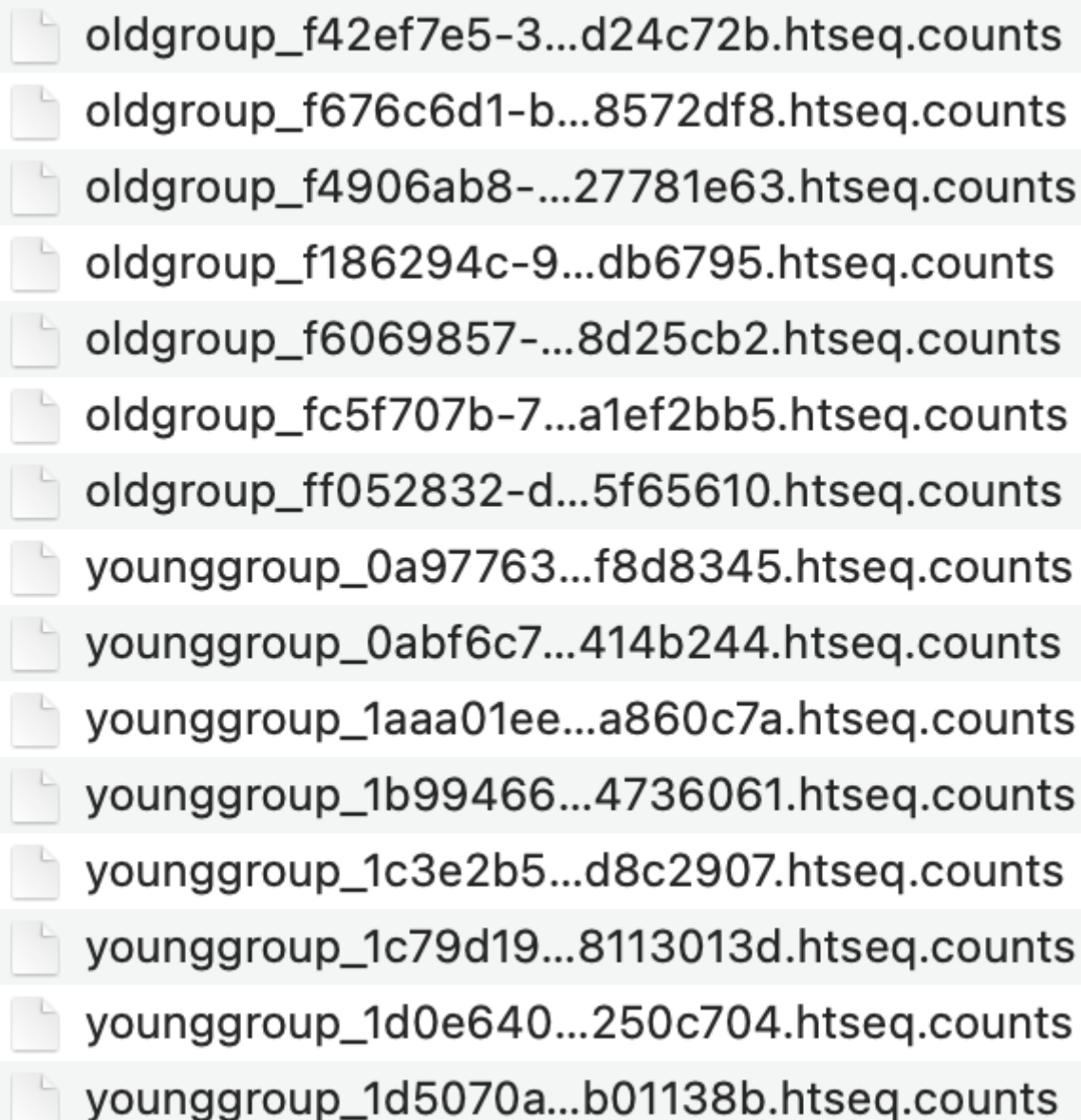
I will put all files in new directory.

unzip all files in by using the command:

```
gunzip *htseq.counts
```

The first group which age between 55-64, I will put them in a folder called "young" and change all their names with the prefix "younggroup".

The second group which age between 65-75, I will put them in a folder called "old" and change all their names with the prefix "oldgroup".



- oldgroup_f42ef7e5-3...d24c72b.htseq.counts
- oldgroup_f676c6d1-b...8572df8.htseq.counts
- oldgroup_f4906ab8-...27781e63.htseq.counts
- oldgroup_f186294c-9...db6795.htseq.counts
- oldgroup_f6069857-...8d25cb2.htseq.counts
- oldgroup_fc5f707b-7...a1ef2bb5.htseq.counts
- oldgroup_ff052832-d...5f65610.htseq.counts
- younggroup_0a97763...f8d8345.htseq.counts
- younggroup_0abf6c7...414b244.htseq.counts
- younggroup_1aaa01ee...a860c7a.htseq.counts
- younggroup_1b99466...4736061.htseq.counts
- younggroup_1c3e2b5...d8c2907.htseq.counts
- younggroup_1c79d19...8113013d.htseq.counts
- younggroup_1d0e640...250c704.htseq.counts
- younggroup_1d5070a...b01138b.htseq.counts

Then, merge all files into a new folder called "all".

Next Steps

I will run through the SOP I presented above and try to reduce errors within my contexts. Maybe run more data to test my script. Then, I will start to create plots from the vignette.

Data

I have uploaded "Sample_young.csv" and "result.txt" in the "Other_files" folder. All my "htseq.counts" files are in the "HTseq_counts_files" folder.

Known Issues

I have met issue with the content in DESeq2 guidelines. However, after discussing with Dr. Craig, problems solved but still need to retest my whole testing scripts.

It is hard to put all files into the scripts that I run, but I will put more data and samples into my scripts eventually.

Milestone2

Modified my Milestone_1(optional)

I have modified my Milestone_1 with more details about how to download the data step by step. Then, I reloaded the data and put more screenshots to follow through.

Input all samples

After testing with more files, now I will start putting all my samples in my script. All samples are in the "HTseq_counts_files" folder. I create a "all" folder which keeps all my samples in "GDC" folder on my Desktop.

```
#htseq-count input
directory <- "~/Desktop/GDC/all"
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)
library("DESeq2")
dds <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable,
                                   directory = directory,
                                   design= ~ condition)
```

Pre-filtering: remove rows in which there are reads less than 10.

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

Note on factor levels: tell results which comparison to make.

```
dds$condition <- factor(dds$condition, levels =
c("younggroup","oldgroup"))
```

Speed-up and parallelization thoughts

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

Differential expression analysis

###The standard differential expression analysis steps are wrapped into a single function, DESeq.(It may take a while.)

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

All kinds of Result tables

You can specify the contrast and build a results table.

```
res <- results(dds, contrast=c("condition","younggroup","oldgroup"))
```

You can summarize some basic tallies using the summary function.

```
summary(res)
```

Or check how many adjusted p-values were less than 0.01.

```
sum(res$padj < 0.01, na.rm=TRUE)
```

Log fold change shrinkage for visualization and ranking.

```
resultsNames(dds)
library(apegglm)
resLFC <- lfcShrink(dds, coef="condition_oldgroup_vs_younggroup",
type="apeglm")
```

P-values and adjusted p-values

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

Set the adjusted p value cutoff to 0.05.

```
res05 <- results(dds, alpha=0.05)
summary(res05)
```


Independent hypothesis weighting: A generalization of the idea of p-value filtering is to weight hypotheses to optimize power.

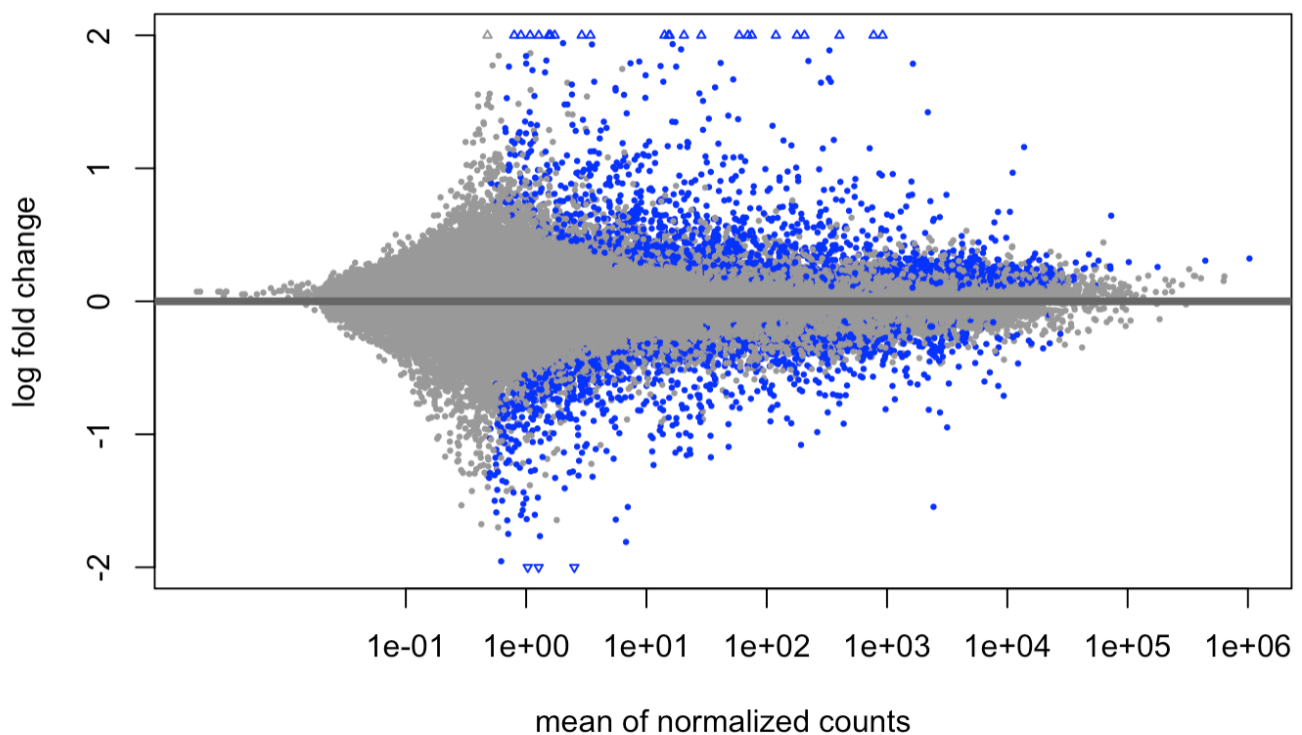
```
library("IHW")
resIHW <- results(dds, filterFun=ihw,
contrast=c("condition","younggroup","oldgroup"), alpha=0.05,)
summary(resIHW)
```

Exploring and exporting results

MA-plot

It is a normal plot if it looks symmetrical from the line in the middle.

```
library(ggplot2)
plotMA(res, ylim=c(-2,2))
```

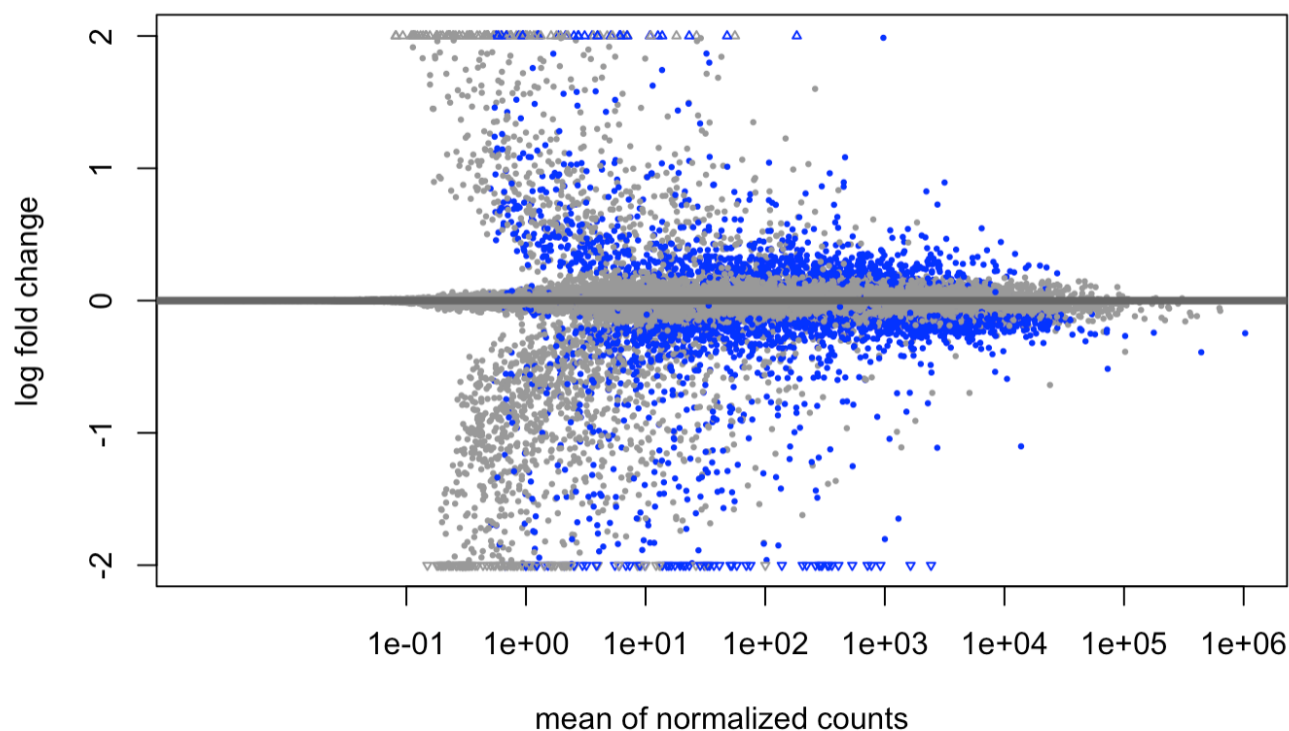


With this plot, I remove the noise associated with log2 fold changes from low count genes without requiring

arbitrary filtering thresholds.

```
{r}
plotMA(resLFC, ylim=c(-2,2))

```



Alternative shrinkage estimators

In DESeq2 version 1.18, they include two additional adaptive shrinkage estimators, available via the type argument of `lfcShrink`.

I can specify the coefficient by the order that it appears in:

```
resultsNames(dds)
```

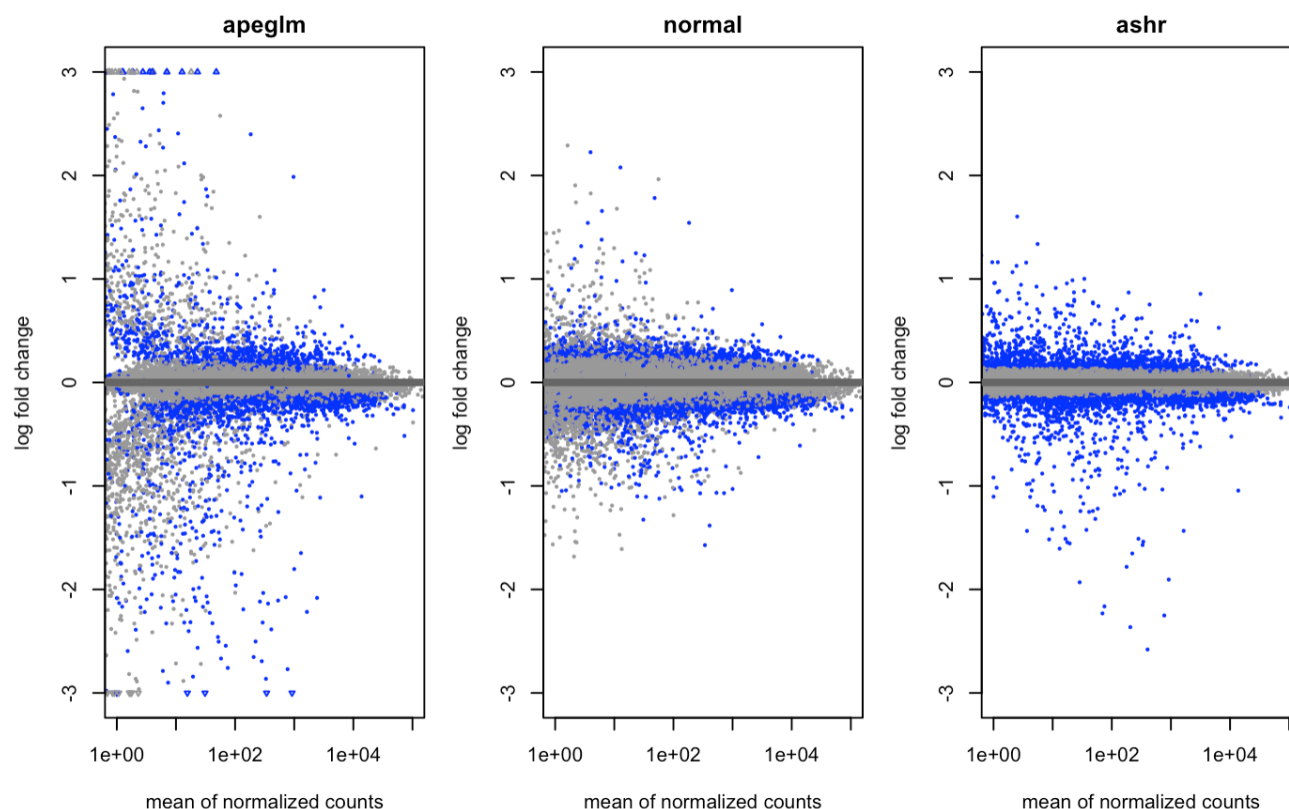
In this case I use `coef=2`.

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

```

{r}
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")

```



The options for **type** are:

apeglm is the adaptive t prior shrinkage estimator from the apegm package.

ashr is the adaptive shrinkage estimator from the ashhr package.

normal is the the original DESeq2 shrinkage estimator, an adaptive Normal distribution as prior.

Plot counts

It can also be useful to examine the counts of reads for a single gene across the groups.

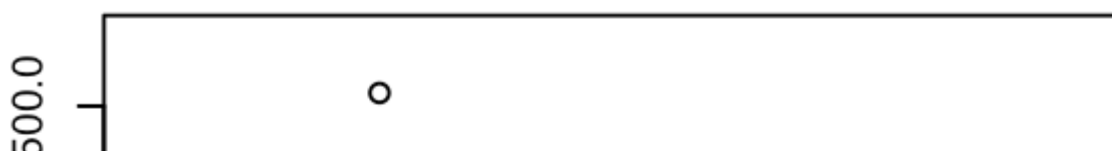
I select a few genes that is related to prostate cancer.

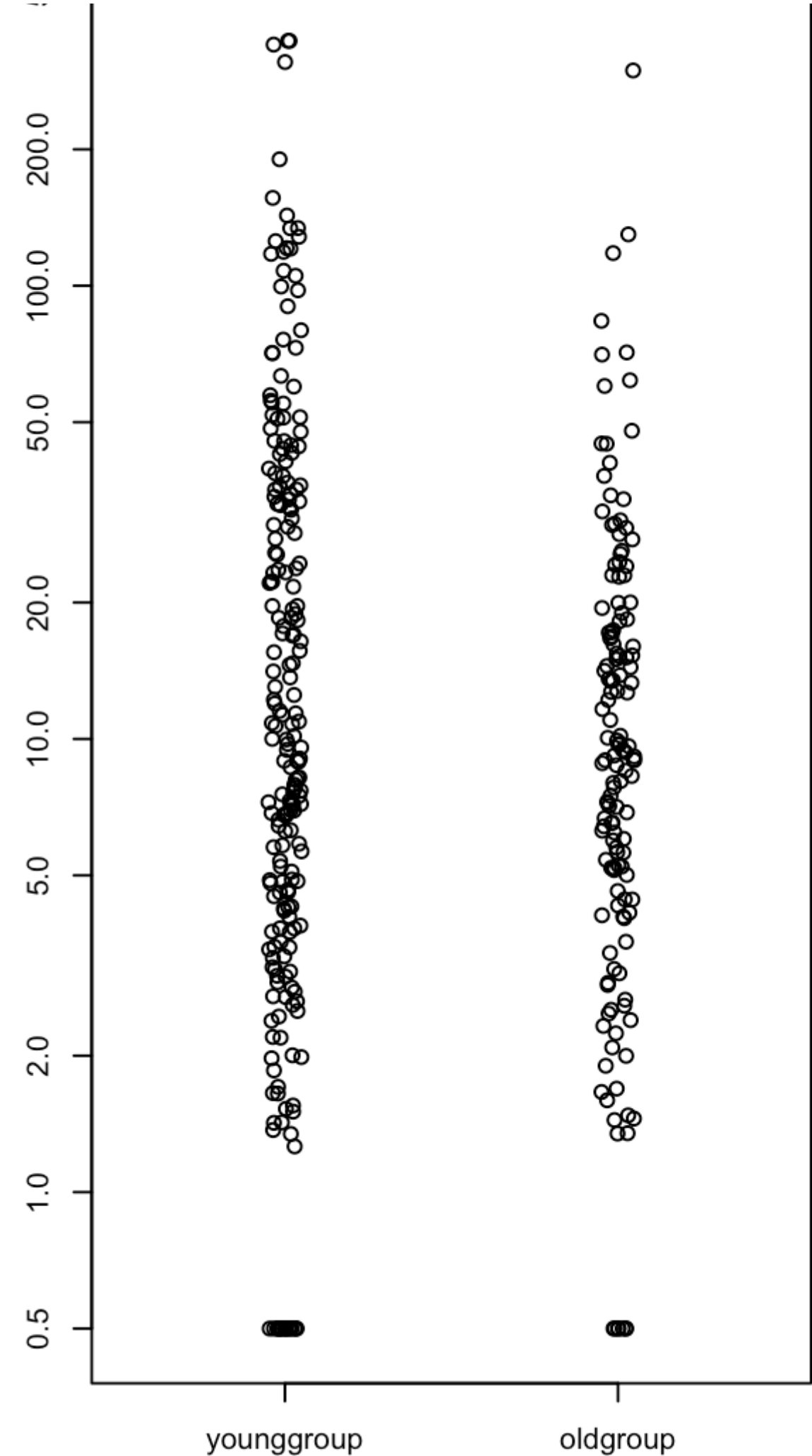
```

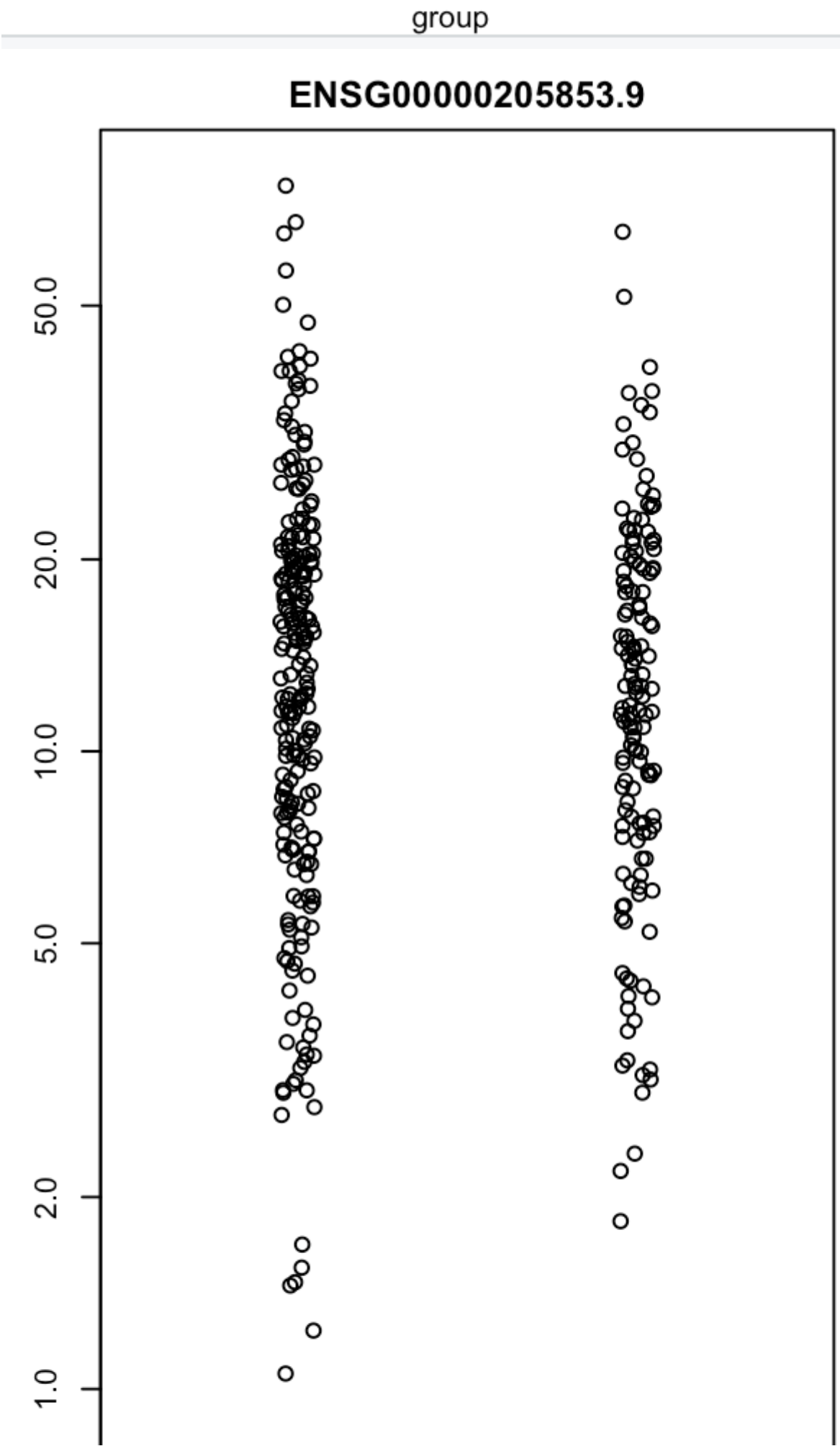
plotCounts(dds, "ENSG00000004809.12", intgroup="condition") #padj<0.01
plotCounts(dds, "ENSG000000205853.9", intgroup="condition") #RFPL3S

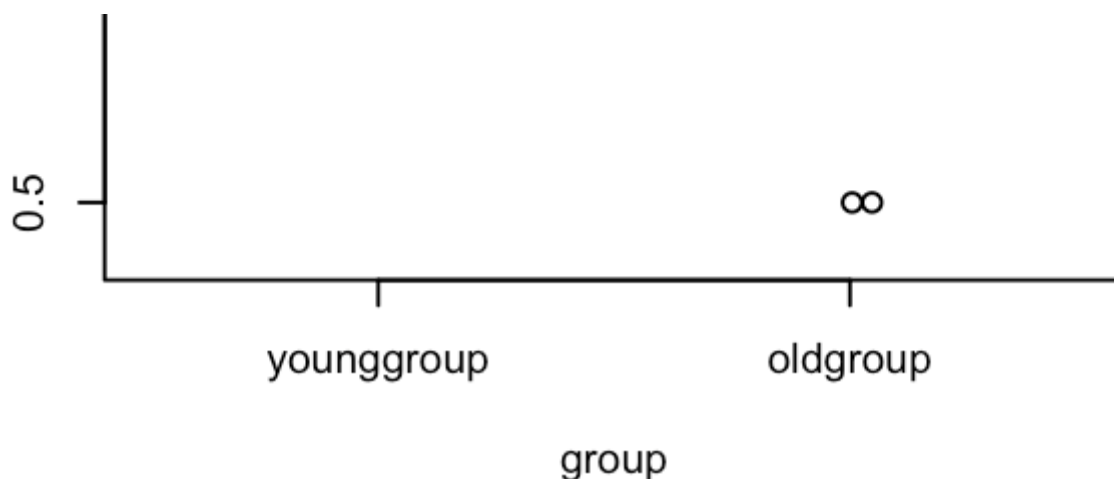
```

ENSG00000004809.12









customized plotting.

I do not see any difference between younggroup and oldgroup by plot counts right now. #####More information on results columns

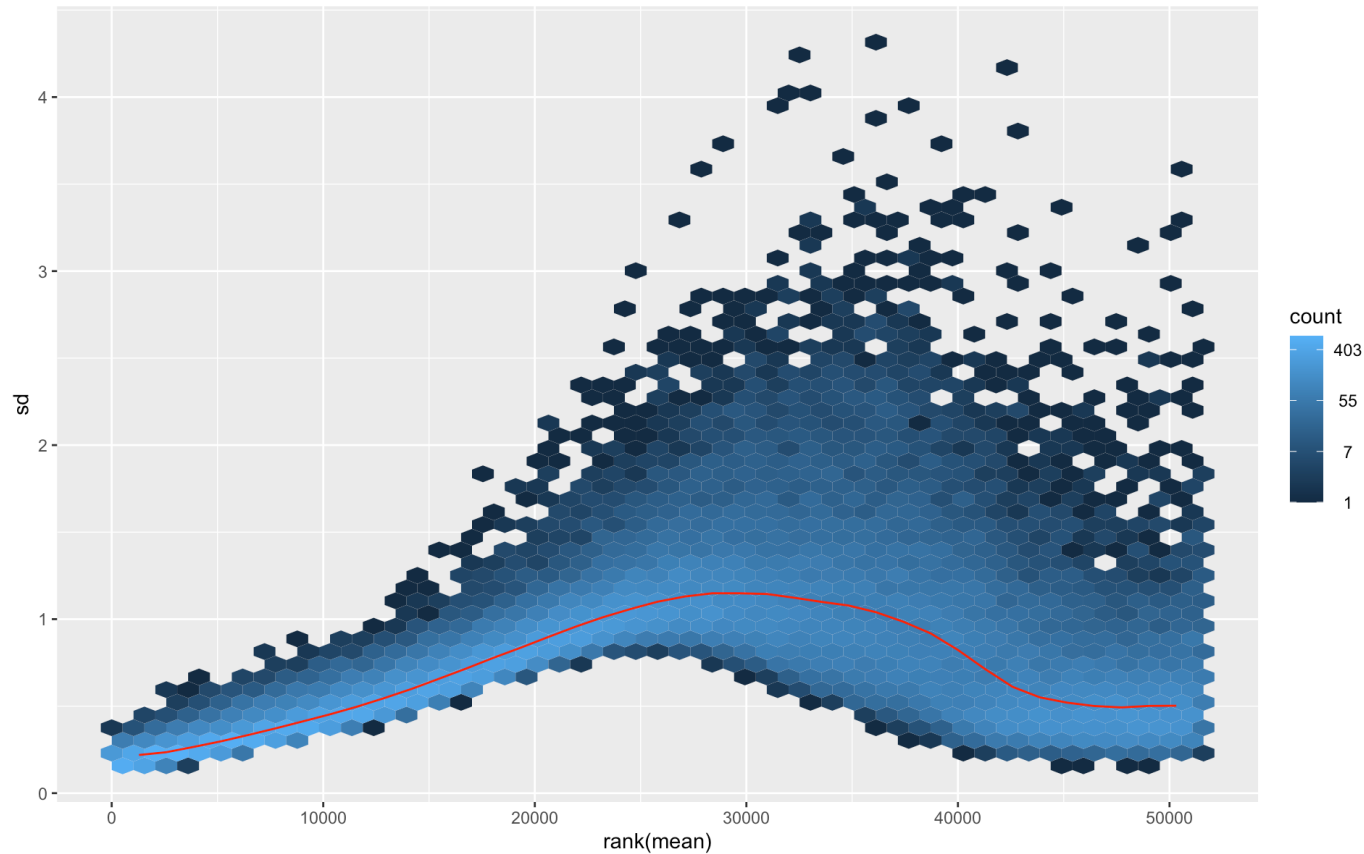
`mcols(res)$description` ##Data transformations and visualization

Extracting transformed values

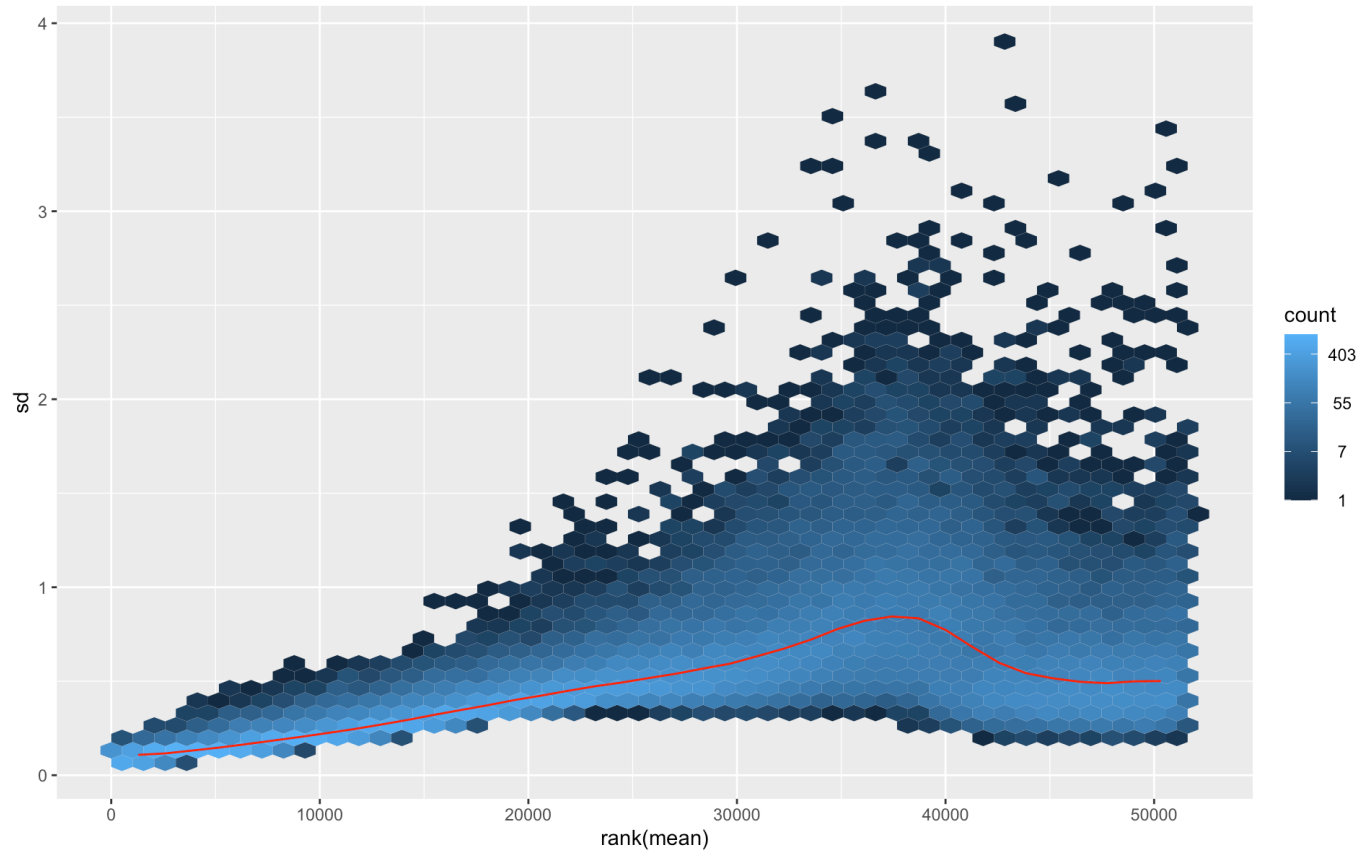
```
vsd <- vst(dds, blind=FALSE)
head(assay(vsd), 3)
```

This gives $\log_2(n + 1)$.

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



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



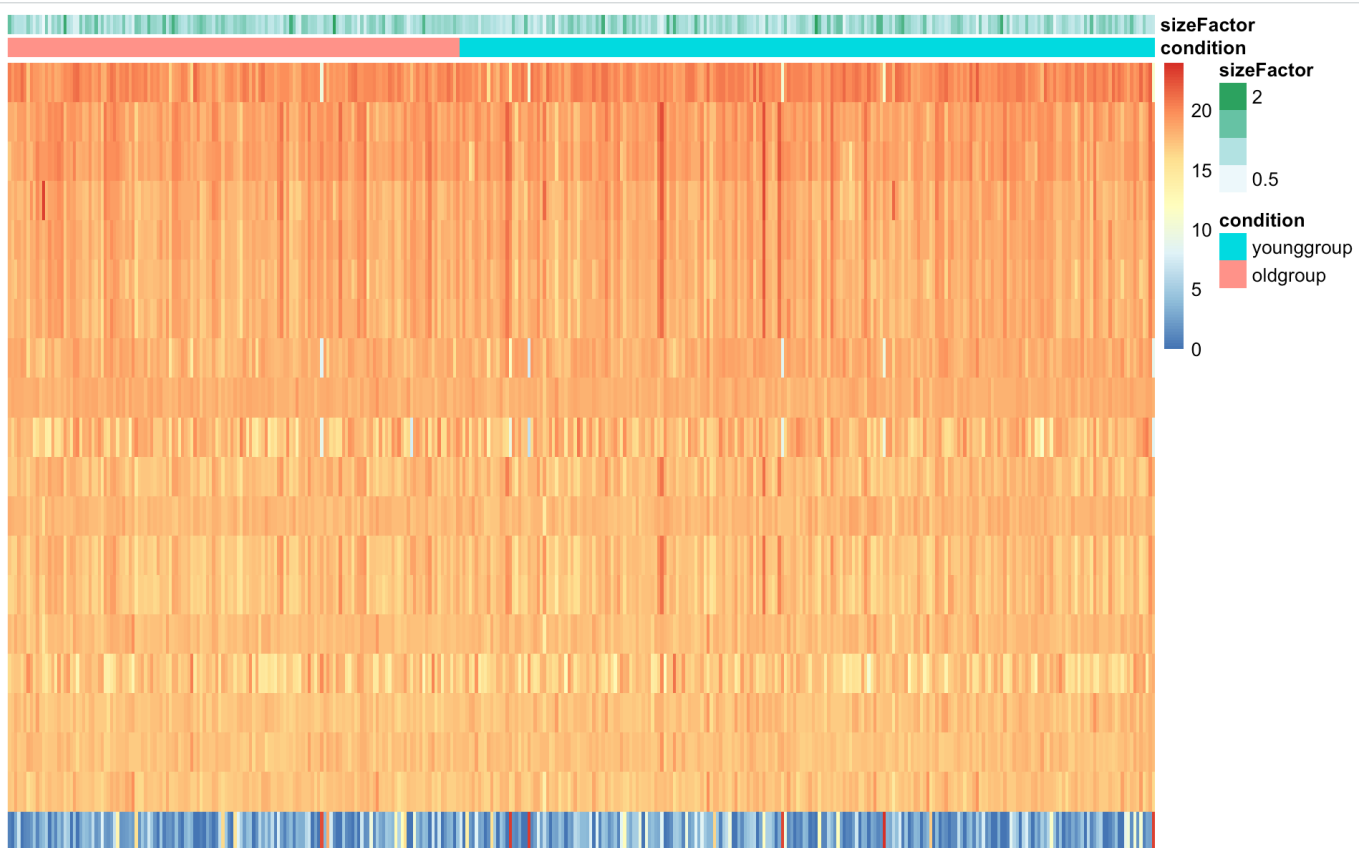
Standard deviation and mean are calculated row-wise from the expression matrix.

Heatmap of the count matrix.

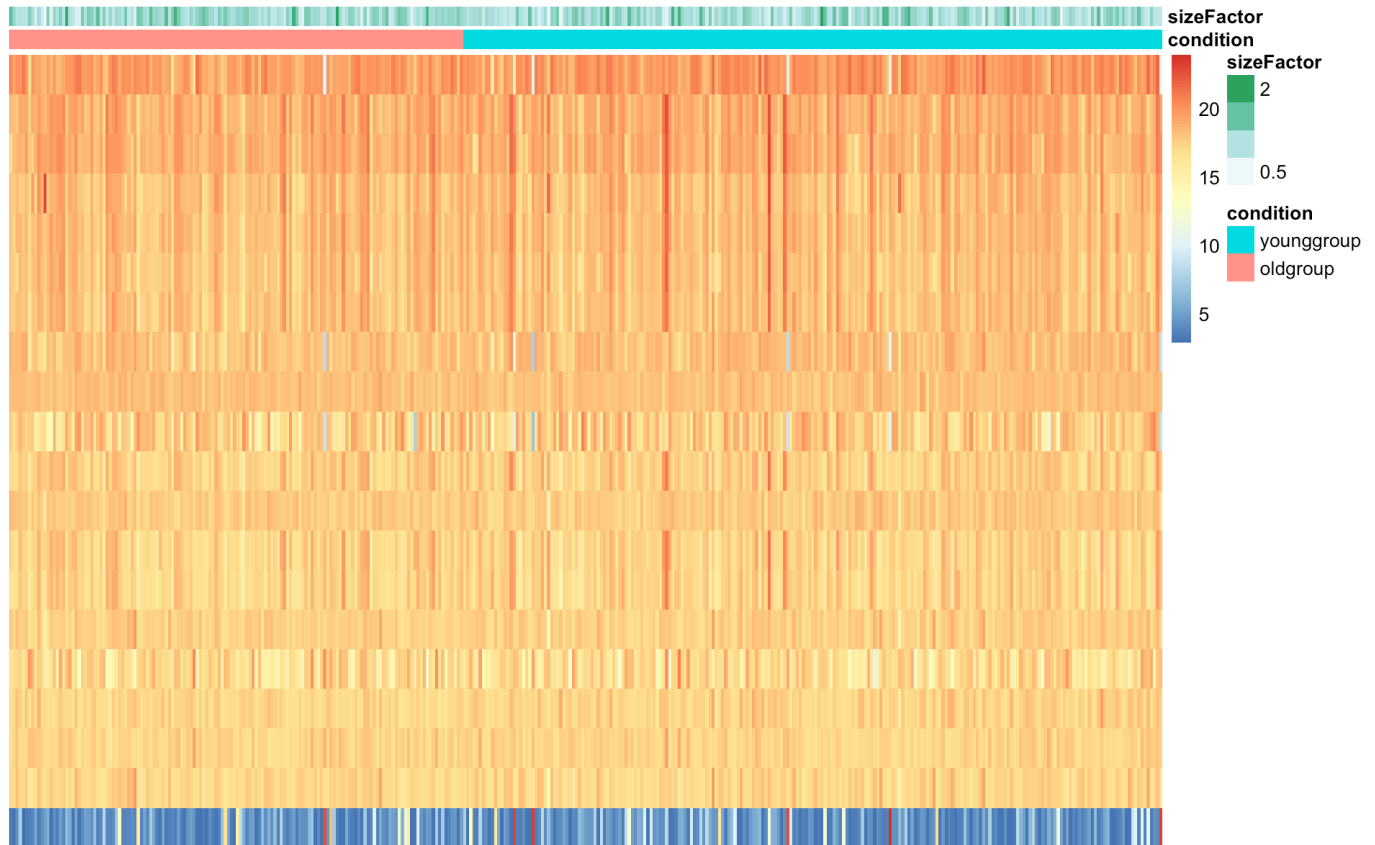
To explore a count matrix, it is often instructive to look at it as a heatmap.

```
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)
```



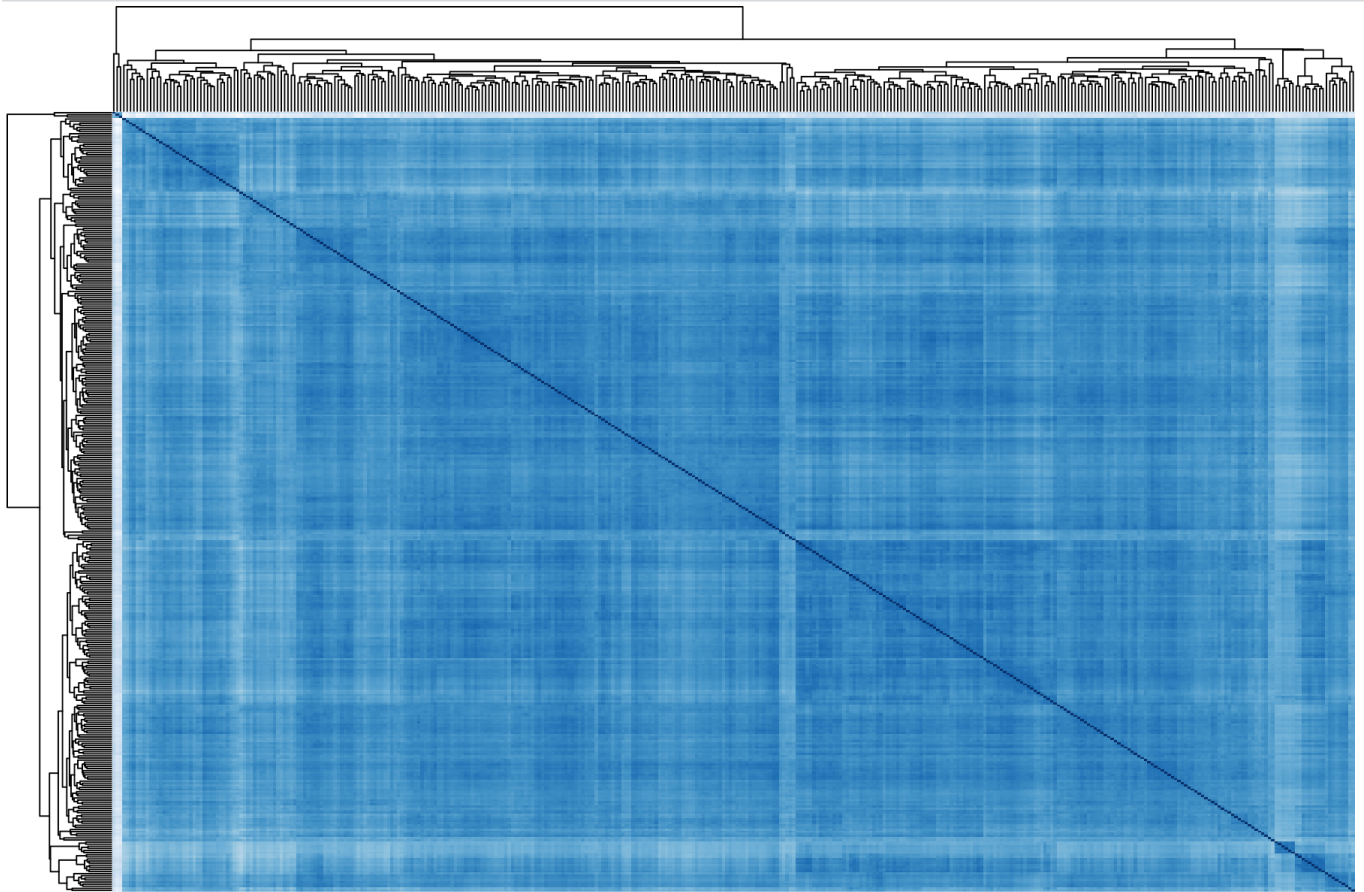
```
pheatmap(assay(vsd)[select,], cluster_rows=FALSE, show_rownames=FALSE,
          cluster_cols=FALSE, annotation_col=df, show_colnames = FALSE)
```

From these heatmaps, younggroup and oldgroup look similar. I think the dataset might be too large to analyze.

Heatmap of the sample-to-sample distances.

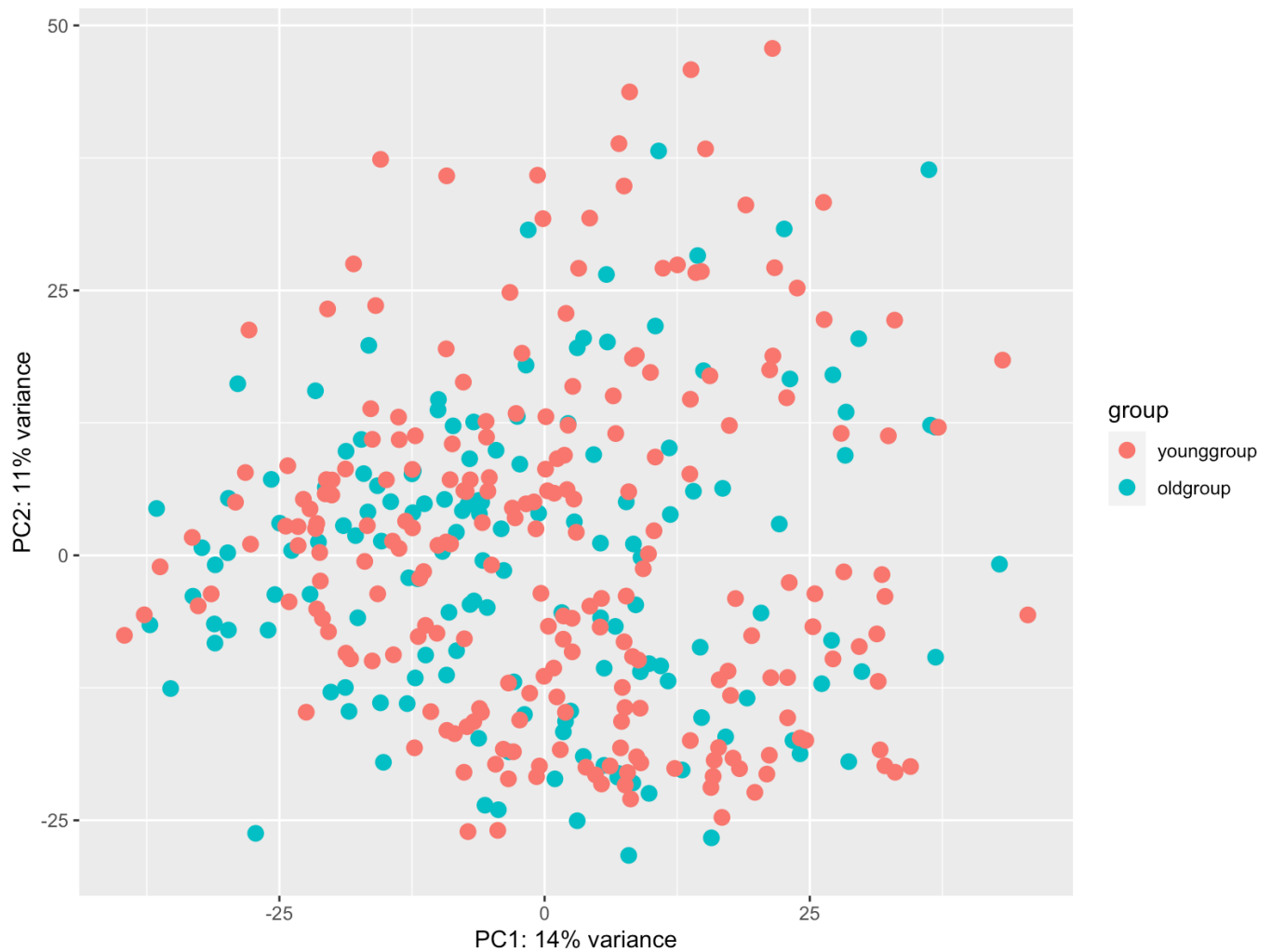
```
sampleDists <- dist(t(assay(vsd)))
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$condition, vsd$type, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
          clustering_distance_rows=sampleDists,
          clustering_distance_cols=sampleDists,
          col=colors, show_rownames=FALSE)
```



Principal component plot.

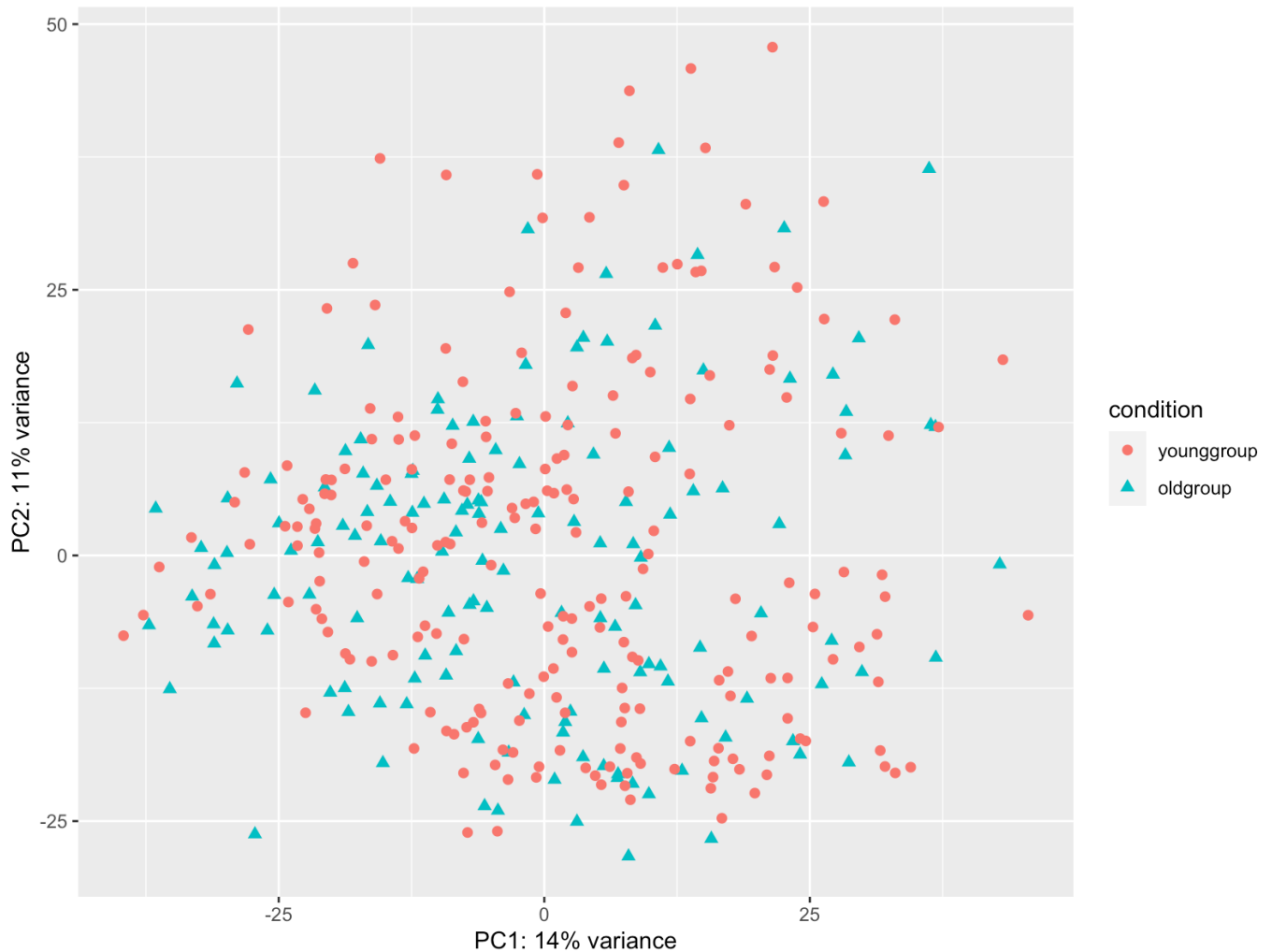
It shows the samples in the 2D plane spanned by their first two principal components.

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



I customize the PCA plot using the ggplot function.

```
pcaData <- plotPCA(vsd, intgroup=c("condition"), returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2, color=condition, shape=condition)) +
  geom_point(size=2) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  coord_fixed()
```



From these PCA plots, we can see that younggroup and oldgroup are clustered together.

Data

All of my data will be uploaded to my GitHub account.

Feedback

Change ensembl id to real gene names.

See how many genes are significantly different (up-regulated or down-regulated).

Try to get no more than 500 genes.

Look into the link Dr. Craig gave us to look into the biology part(put in the gene list I have).

Heatmaps need to be fixed.

Known issues

I will change my p-value to 0.05 and 2foldchange to 1 and see what will happen with my plots.

I will try to change all my ensemble id to hugo id.

I have faced a problem with the heatmap error. Everytime I try to put the reference of "annotation_col=df" into my code, it will not work.

Last changes

Dataset

Independent hypothesis weighting: A generalization of the idea of p-value filtering is to weight hypotheses to optimize power. (The result table that I select to manage my data.)

```
library("IHW")
resIHW <- results(dds, filterFun=ihw,
contrast=c("condition","younggroup","oldgroup"), alpha=0.05,)
summary(resIHW)
```

Change Ensembl_id into gene_name.

First, remove ensembl_id version name.

```
ens_id<- substr(row.names(resIHW),1 ,15)
row.names(resIHW) <- ens_id
rawcount<- resIHW
Ensembl_ID <- data.frame(Ensembl_ID = row.names(rawcount))
row.names(Ensembl_ID) <- Ensembl_ID[,1]
rawcount <- cbind(Ensembl_ID, rawcount)
```

Function to Change Ensembl_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_T0_Genename <- data.frame(gene_id = gene_id,
                                       gene_name = gene_name,
                                       stringsAsFactors = FALSE)

  return(Ensembl_ID_T0_Genename)
}
```

Save the list of Ensembl_ids and gene_names into csv file

```
Ensembl_ID_TO_Genename <-  
get_map("~/Desktop/GDC/gencode.v38lift37.annotation.gtf")  
gtf_Ens_ID <- substr(Ensembl_ID_TO_Genename[,1],1,15)  
Ensembl_ID_TO_Genename <- data.frame(gtf_Ens_ID,  
Ensembl_ID_TO_Genename[,2])  
colnames(Ensembl_ID_TO_Genename) <- c("Ensembl_ID","gene_id")  
write.csv(Ensembl_ID_TO_Genename, file =  
"~/Desktop/GDC/Ensembl_ID_TO_Genename.csv")
```

Merge data with "Ensembl_ID".

```
mergeRawCounts <- merge(Ensembl_ID_TO_Genename, rawcount ,by =  
"Ensembl_ID")
```

Remove duplicate data by "gene_id".

```
index <- duplicated(mergeRawCounts$gene_id)  
mergeRawCounts <- mergeRawCounts[!index,]
```

Use gene_id as rownames.

```
rownames(mergeRawCounts) <- mergeRawCounts[, "gene_id"]  
res_new <- mergeRawCounts[, -c(1:2)]
```

####save files.

```
write.csv(as.data.frame(res_new), file = "~/Desktop/GDC/res_new.csv")
```

Create a upregulated genes list and a downregulated genes list.

```
summary(res_new)  
res_df <- as.data.frame(res_new)  
get_upregulated <- function(df){  
  
  key <- intersect(rownames(df)[which(df$log2FoldChange>=1)], rownames(df)  
[which(df$pvalue<=0.05)])
```

```

results <- as.data.frame((df)[which(rownames(df) %in% key),])
return(results)
}

get_downregulated <- function(df){

  key <- intersect(rownames(df)[which(df$log2FoldChange<=-1)],
rownames(df)[which(df$pvalue<=0.05)])

  results <- as.data.frame((df)[which(rownames(df) %in% key),])
  return(results)
}

up <- get_upregulated(res_df)
write.csv(as.data.frame(up), file = "~/Desktop/GDC/up.csv")
down <- get_downregulated(res_df)
write.csv(as.data.frame(down), file = "~/Desktop/GDC/down.csv")

```

After getting up and down csv files, I save them as txt files. Then, I use the [GSEA](#) website and put all my up genes into it.

I got three sets of Gene Set Name.

Gene Set Name [# Genes (K)]	Description	# Genes in Overlap (k)	k/K	p-value	FDR q-value
KRAS.600_UP.V1_UP [278]	Genes up-regulated in four lineages of epithelial cell lines over-expressing an oncogenic form of KRAS [Gene ID=3845] gene.	6		9.76 e ⁻⁵	1.85 e ⁻²
KRAS.300_UP.V1_UP [142]	Genes up-regulated in four lineages of epithelial cell lines over-expressing an oncogenic form of KRAS [Gene ID=3845] gene.	4		5.55 e ⁻⁴	3.88 e ⁻²
ATM_DN.V1_DN [146]	Genes down-regulated in HEK293 cells (kidney fibroblasts) upon knockdown of ATM [Gene ID=472] gene by RNAi.	4		6.16 e ⁻⁴	3.88 e ⁻²

Gene/geneset overlap matrix

Entrez Gene Id	Gene Symbol	KRAS.600_UP.V1_UP	KRAS.300_UP.V1_UP	ATM_DN.V1_DN	Entrez	Ensembl	Gene Description
23676	SMPX						small muscle protein X-linked [Source:HGNC Symbol;Acc:HGNC:11122]
167	CRISP1						cysteine rich secretory protein 1 [Source:HGNC Symbol;Acc:HGNC:304]
2118	ETV4						ETS variant transcription factor 4 [Source:HGNC Symbol;Acc:HGNC:3493]
1081	CGA						"glycoprotein hormones, alpha polypeptide [Source:HGNC Symbol;Acc:HGNC:1885]"
2670	GFAP						glial fibrillary acidic protein [Source:HGNC Symbol;Acc:HGNC:4235]
55079	FEZF2						FEZ family zinc finger 2 [Source:HGNC Symbol;Acc:HGNC:13506]
8557	TCAP						titin-cap [Source:HGNC Symbol;Acc:HGNC:11610]
7224	TRPC5						transient receptor potential cation channel subfamily C member 5 [Source:HGNC Symbol;Acc:HGNC:12337]
25769	SLC24A2						solute carrier family 24 member 2 [Source:HGNC Symbol;Acc:HGNC:10976]
4741	NEFM						neurofilament medium [Source:HGNC Symbol;Acc:HGNC:7734]
4621	MYH3						myosin heavy chain 3 [Source:HGNC Symbol;Acc:HGNC:7573]
57495	NWD2						NACHT and WD repeat domain containing 2 [Source:HGNC Symbol;Acc:HGNC:29229]
57408	LRTM1						leucine rich repeats and transmembrane domains 1 [Source:HGNC Symbol;Acc:HGNC:25023]
5047	PAEP						progesterone associated endometrial protein [Source:HGNC Symbol;Acc:HGNC:8573]

However, when I put all my down genes into the website, I got zero set.

Compute Overlaps for Selected Genes

Converted 74 submitted identifiers into 56 NCBI (Entrez) genes. [click here for details](#).

Collections	# Overlaps Shown	# Gene Sets in Collections	# Genes in Comparison (n)	# Genes in Universe (N)
C6	0	189	56	40312

No overlaps found

Known issues

I cannot use `rld <- rlog(dds, blind=FALSE)` because my samples are too large to use this code. It ran overnight and still got nothing.

I could reduce my data into smaller dataset, so all my plot could look better and relatively easy to analyze.

After changing `ensembl_id` into `gene_names` for my "up" and "down" files, I still don't know how to change dds dataset's names.

Try to fix the problem of down file with zero set in the end.

Conclusion

I should reduce my data to a smaller size that is easier to manage.

I don't see a major difference between two groups. The group I choose should have a larger gap between them, like 30-40 years old to 60-70 years old.

Deliverable

A complete repository with clear documentation and description of my analysis and results.