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 studied 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 utilized the package DESeq2 and follow the specific vignette: link

For this analysis, I used the TCGA cohort and identidfied 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 controled for race and primary gleason grade.

Data

I used 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).

Files manage

I put all csv files in the folder of "Excel".

I put all htseq.counts files in the folder of "HTseq_counts_files".

I put all png images in the folder of "Images".

I put all PDF files in the folder of "PDF".

I put all scripts in the folder of "Scripts".

Other form of files are in the folder of " Other_files".

Milestone_1

Data filtering

First, go to GDC and click on "**Repository**".

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

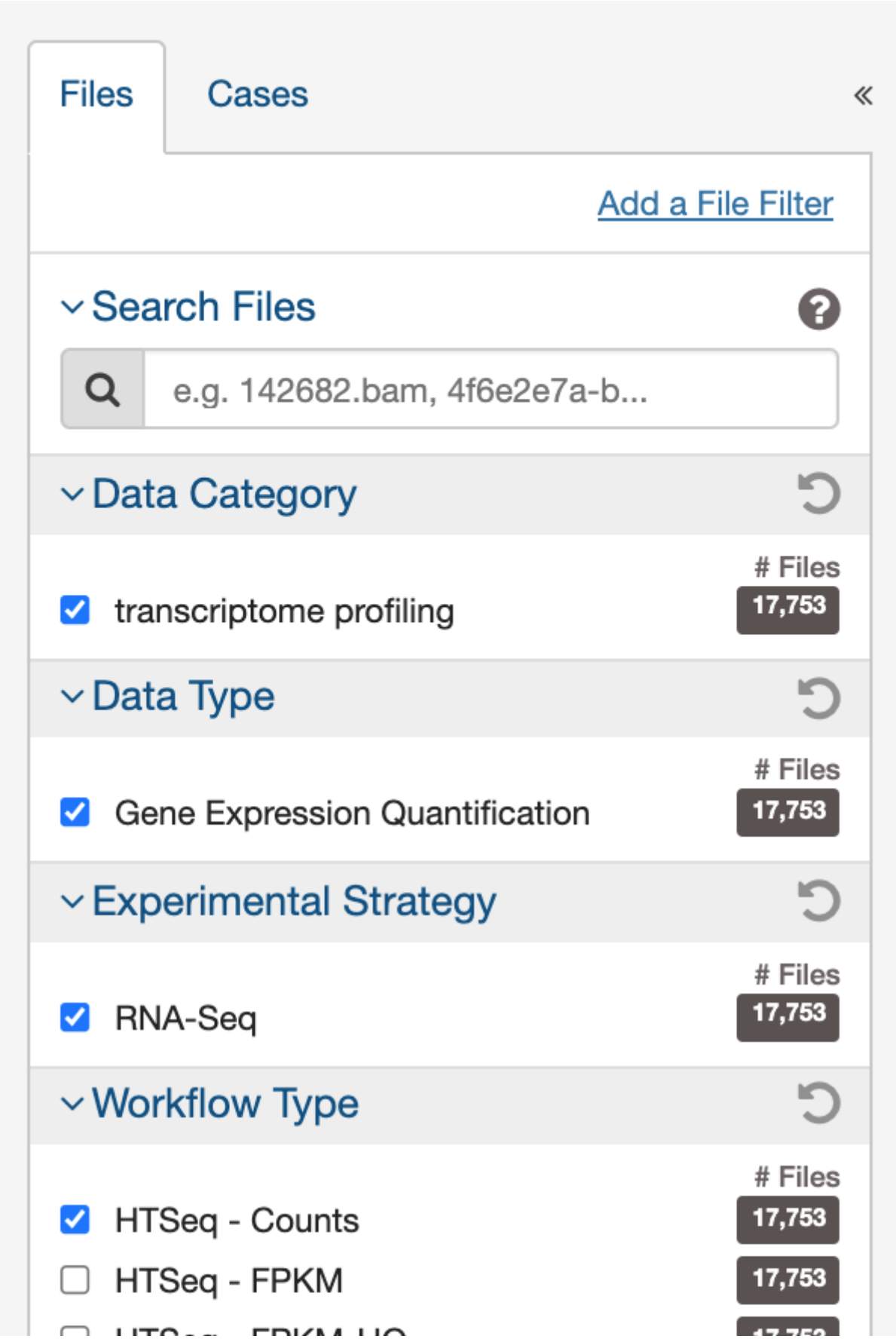
Data Category - "**transcriptome profiling**".

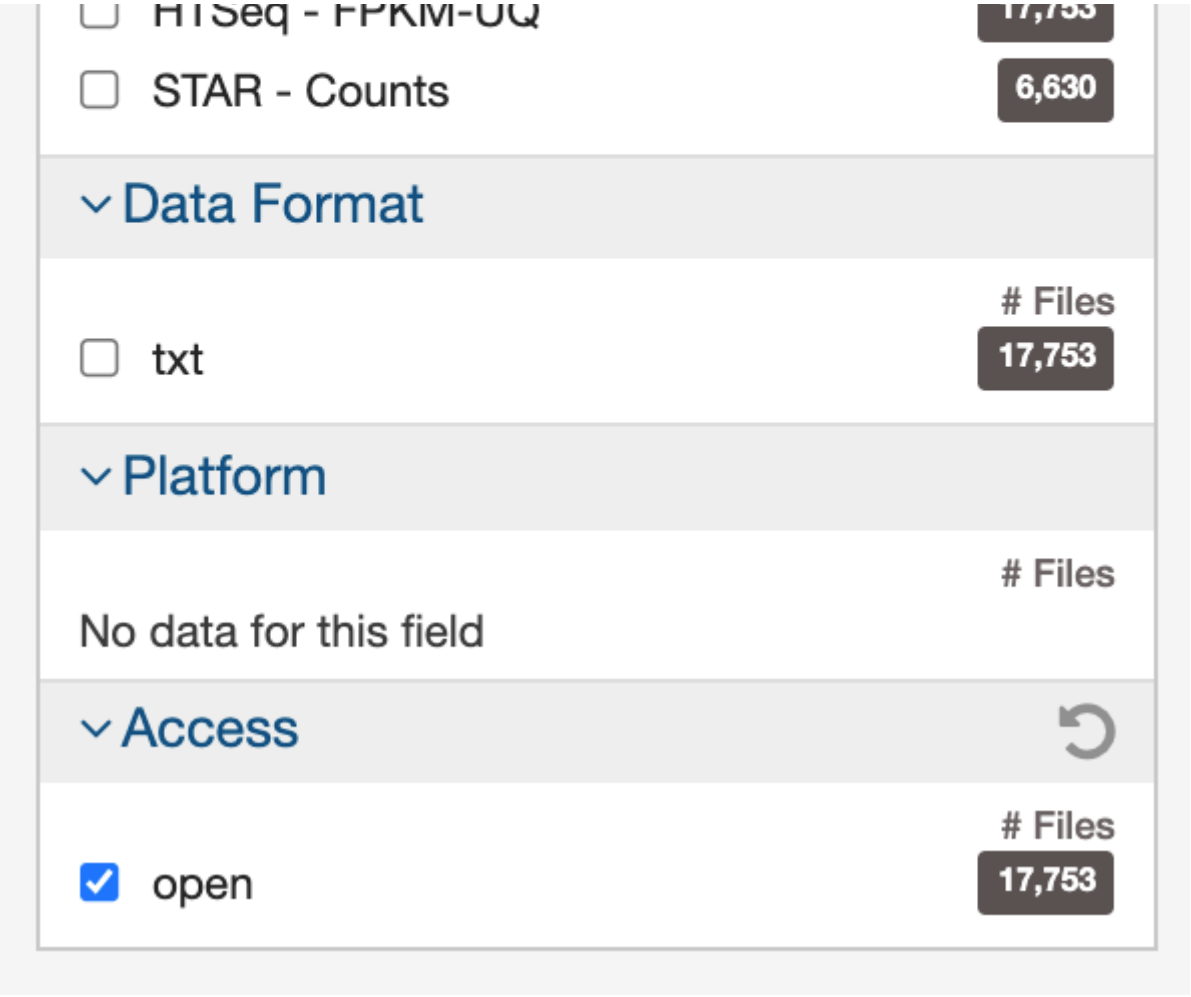
Data Type - "**Gene Expression Quantification**".

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

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

Access - "**open**".





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**".

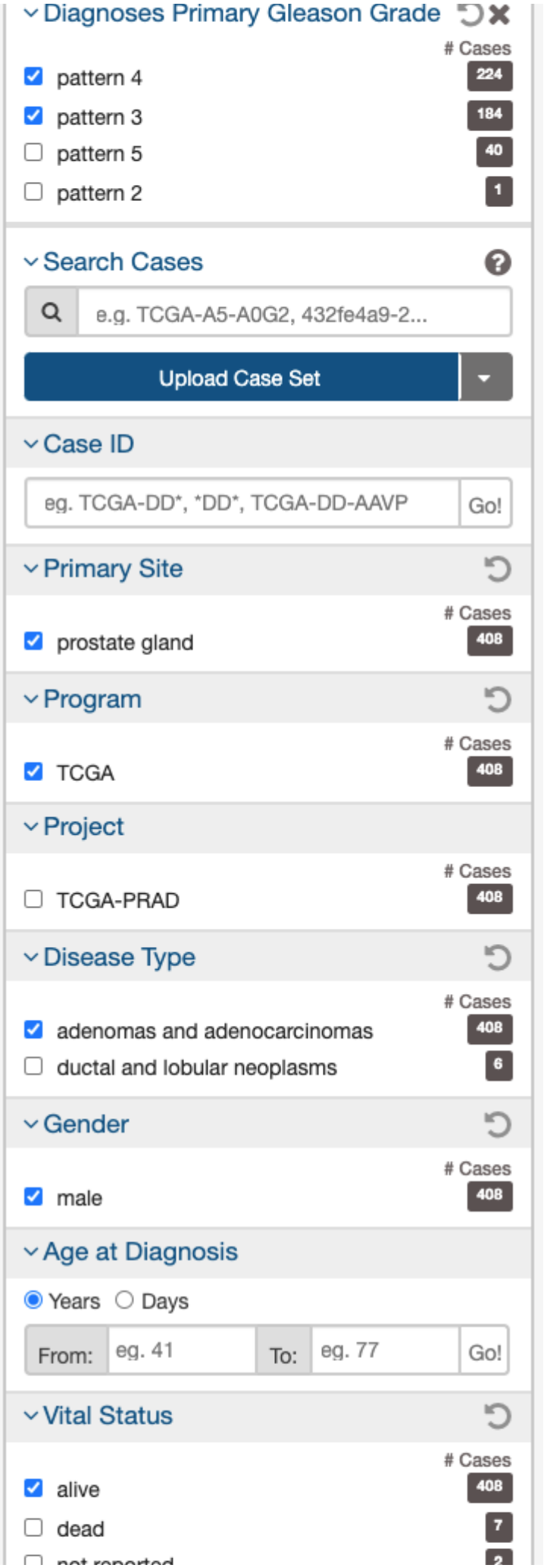
Disease Type - "**adenomas and adenocarcinomas**".

Gender - "**male**".

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

Vital Status - "**alive**".

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

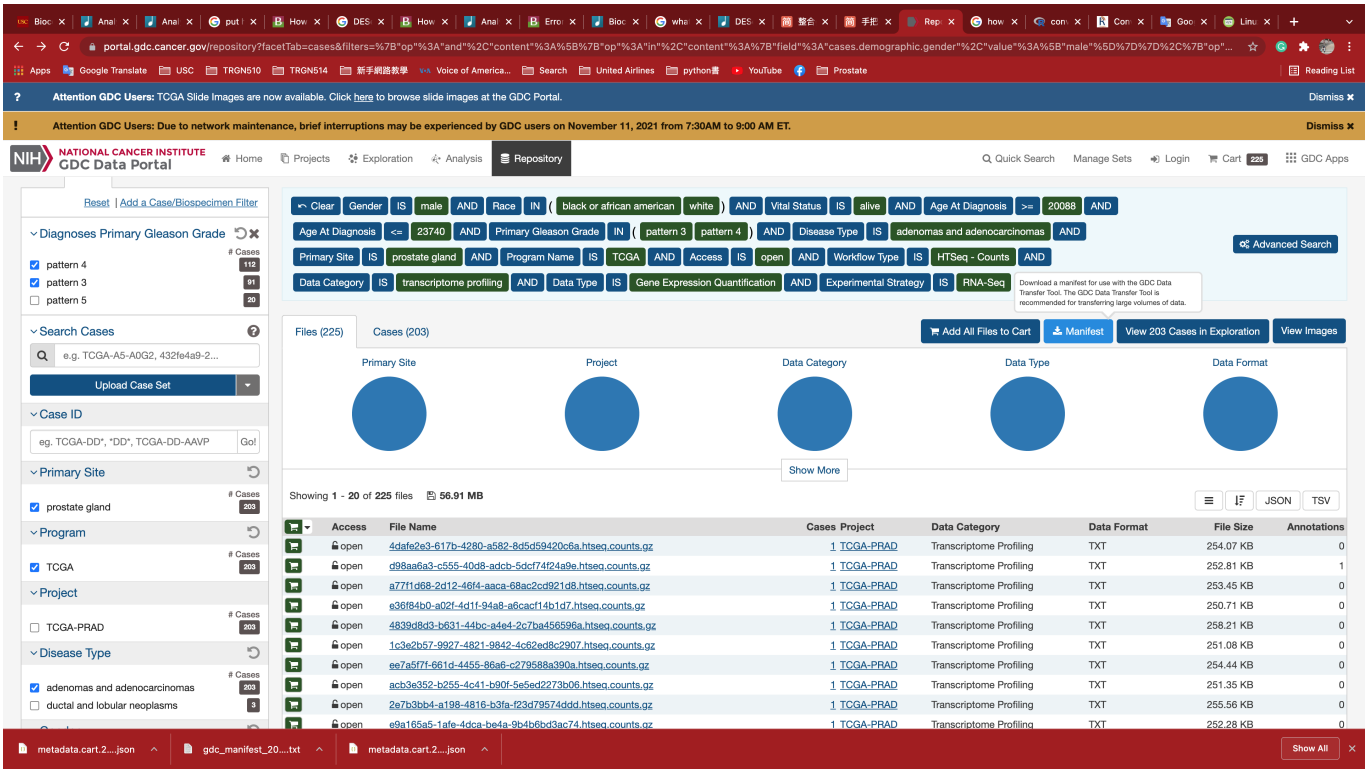

Diagnoses Primary Gleason Grade
Cases
pattern 4 224
pattern 3 184
pattern 5 40
pattern 2 1
Search Cases
e.g. TCGA-A5-A0G2, 432fe4a9-2...
Upload Case Set
Case ID
eg. TCGA-DD*, *DD*, TCGA-DD-AAVP
Go!
Primary Site
Cases
prostate gland 408
Program
Cases
TCGA 408
Project
Cases
TCGA-PRAD 408
Disease Type
Cases
adenomas and adenocarcinomas 408
ductal and lobular neoplasms 6
Gender
Cases
male 408
Age at Diagnosis
Years
Days
From:
eg. 41
To:
eg. 77
Go!
Vital Status
Cases
alive 408
dead 7

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**.



You have to download "gdc-client" form 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 &
```

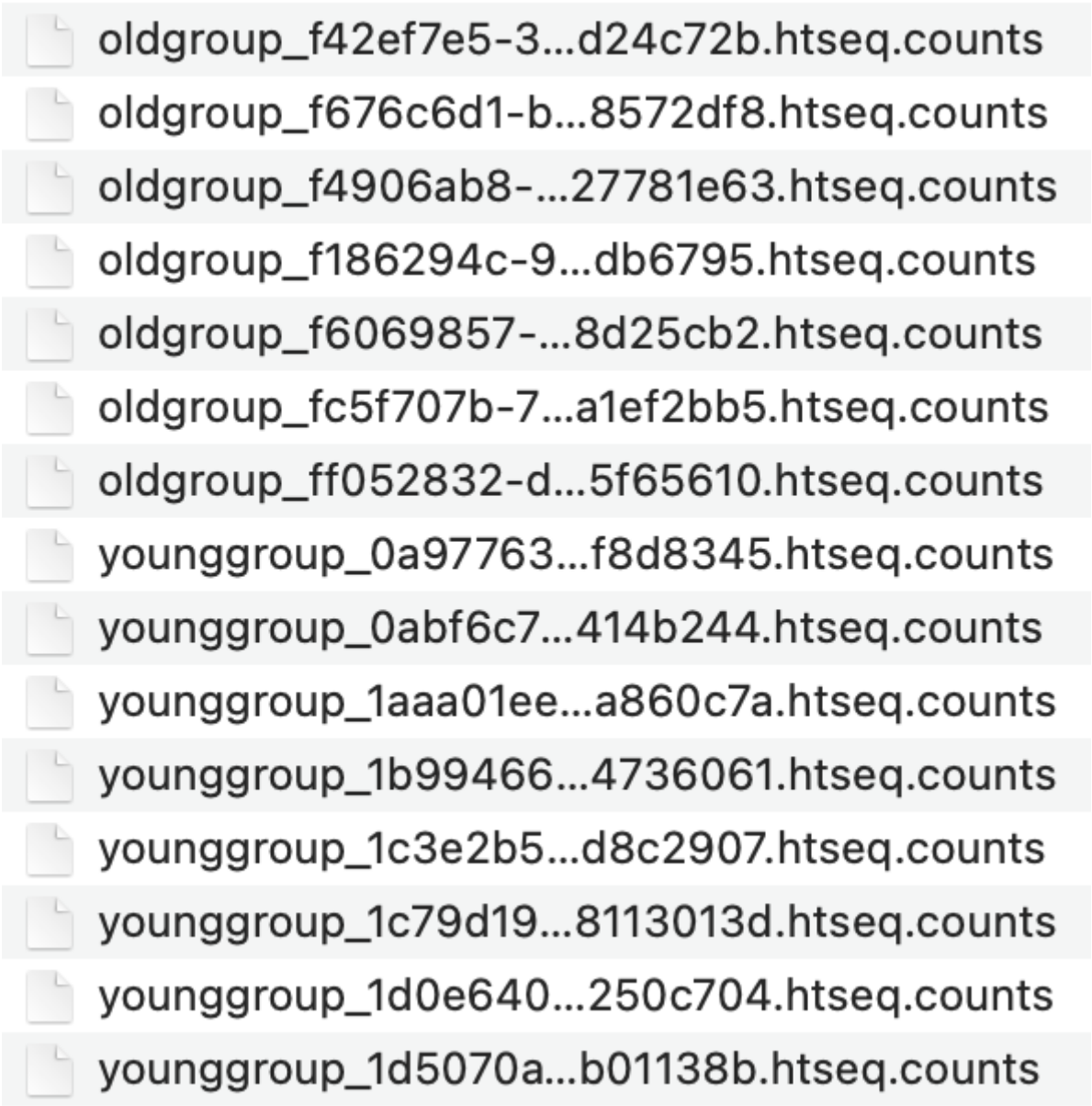
I will put all files in new directory.

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".



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

Next Steps

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

Data

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

Known Issues

I met issue with the content in DESeq2 guildlines. 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 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, I started putting all my samples in my script. All samples are in the "HTseq_counts_files" folder. I created 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(apeglm)
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.

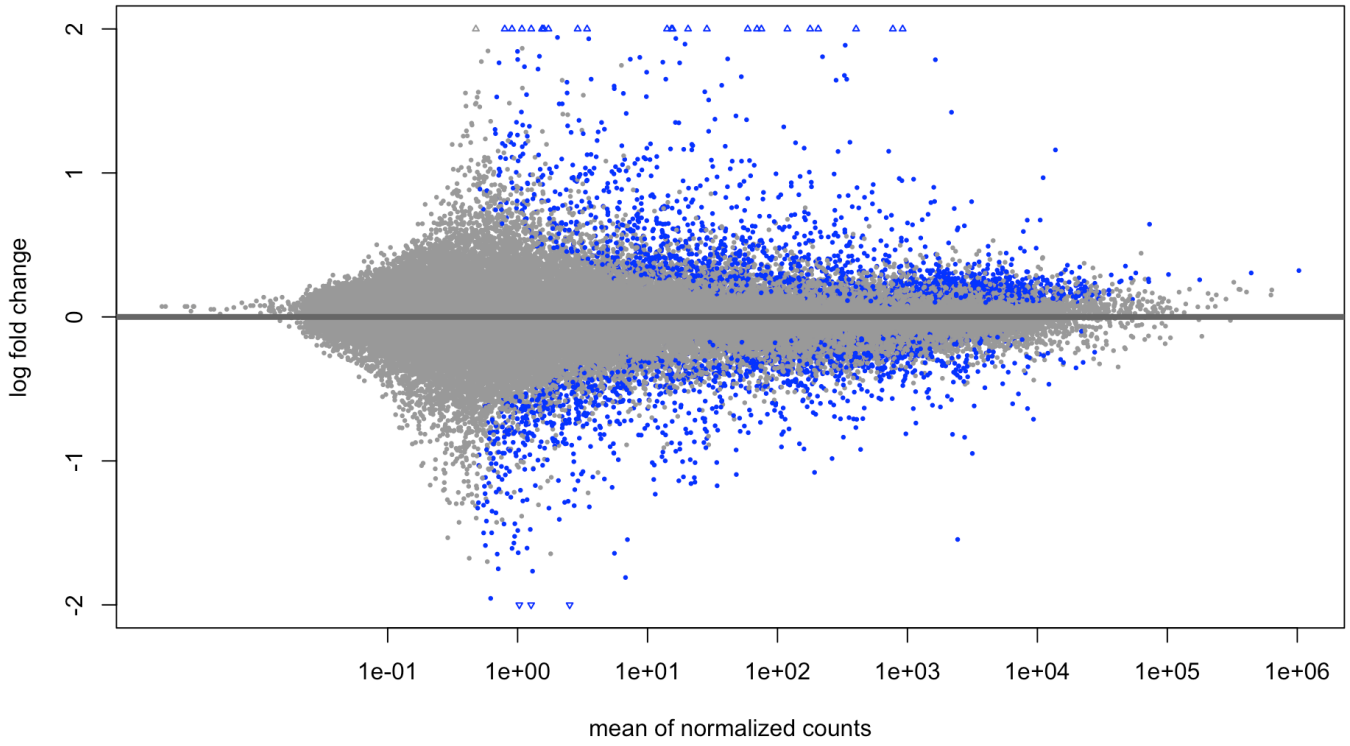
```
library("IHW")
resIHW <- results(dds, filterFun=ihw,
contrast=c("condition","younggroup","oldgroup"), alpha=0.05,)
summary(resIHW)
```

Exploring and exporting results

MA-plot

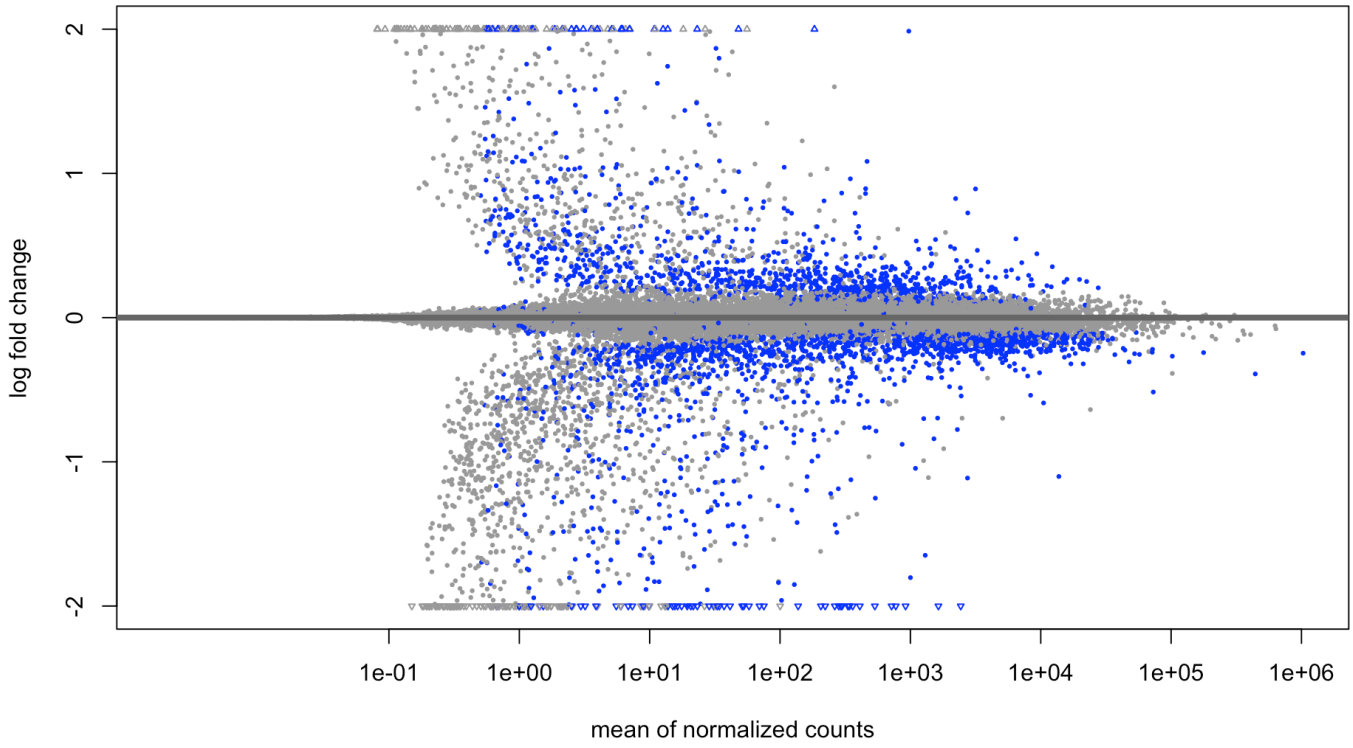
It is a normal plot if it looks symmetrical from the line in the middle.

```
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.

```
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.

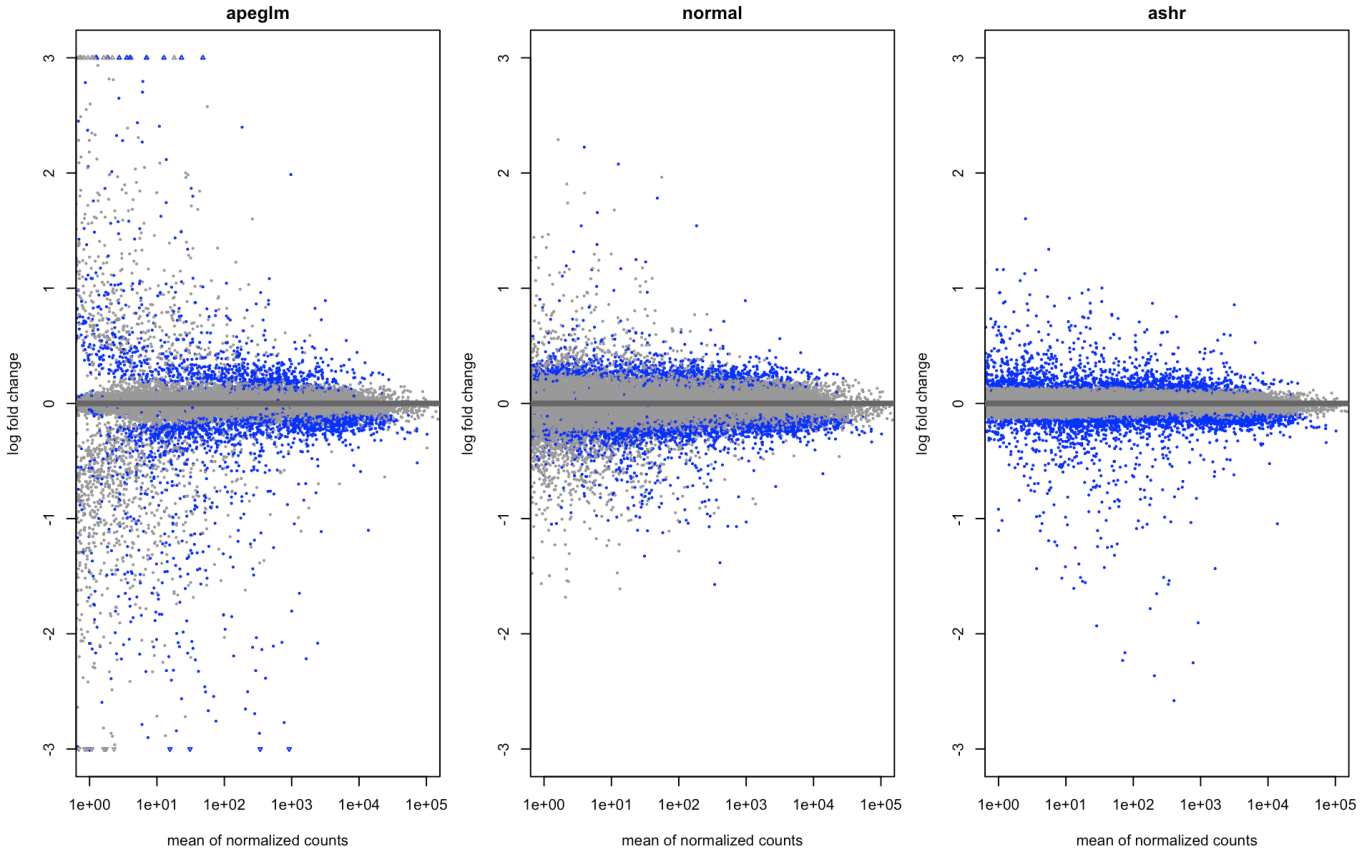
I can specify the coefficient by the order that it appears in:

```
resultsNames(dds)
```

In this case I use coef=2.

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

```
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 apeglm package.

`ashr` is the adaptive shrinkage estimator from the ashr package.

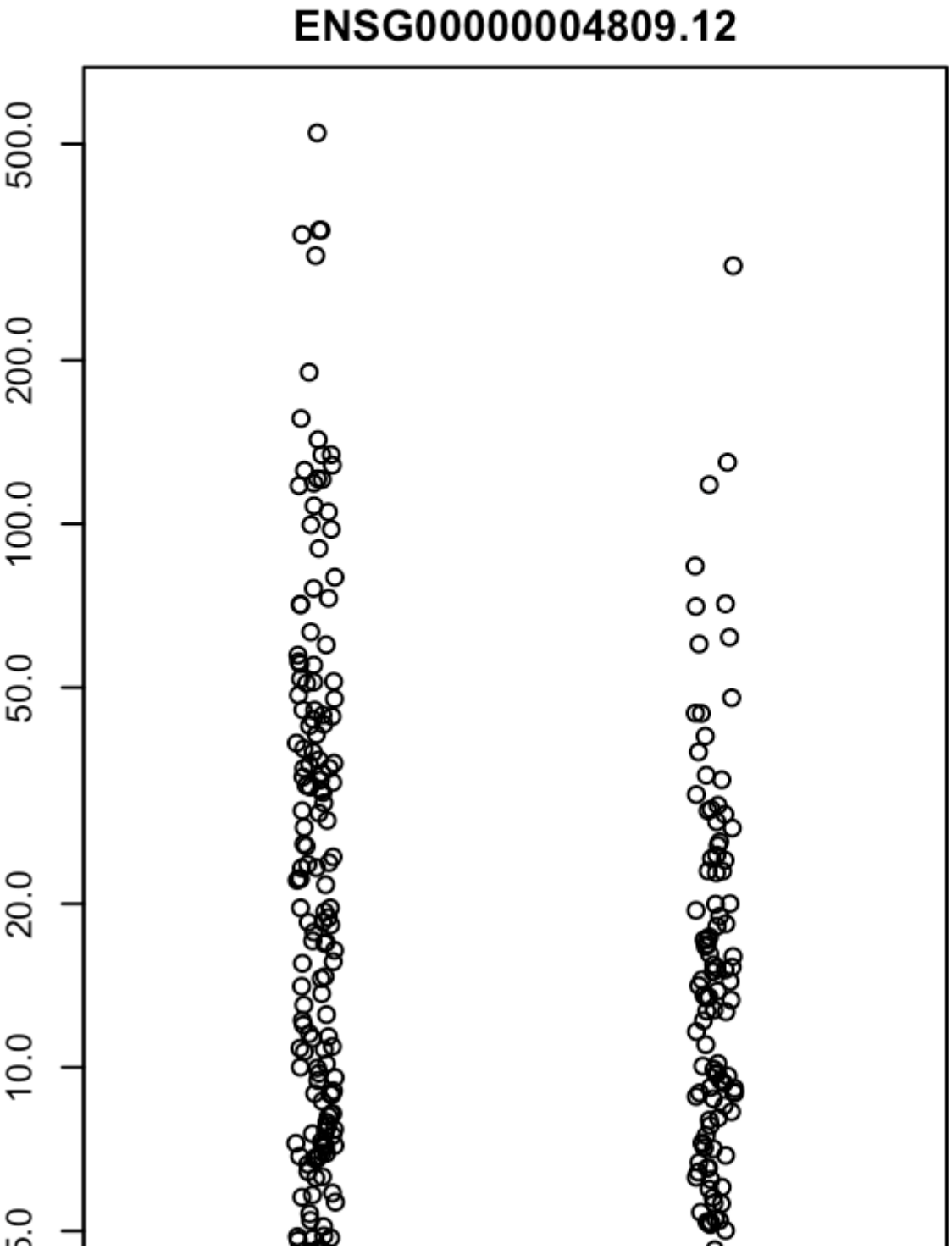
`normal` is the the original DESeq2 shrinkage estimator, an adaptive Normal distribution as prior.

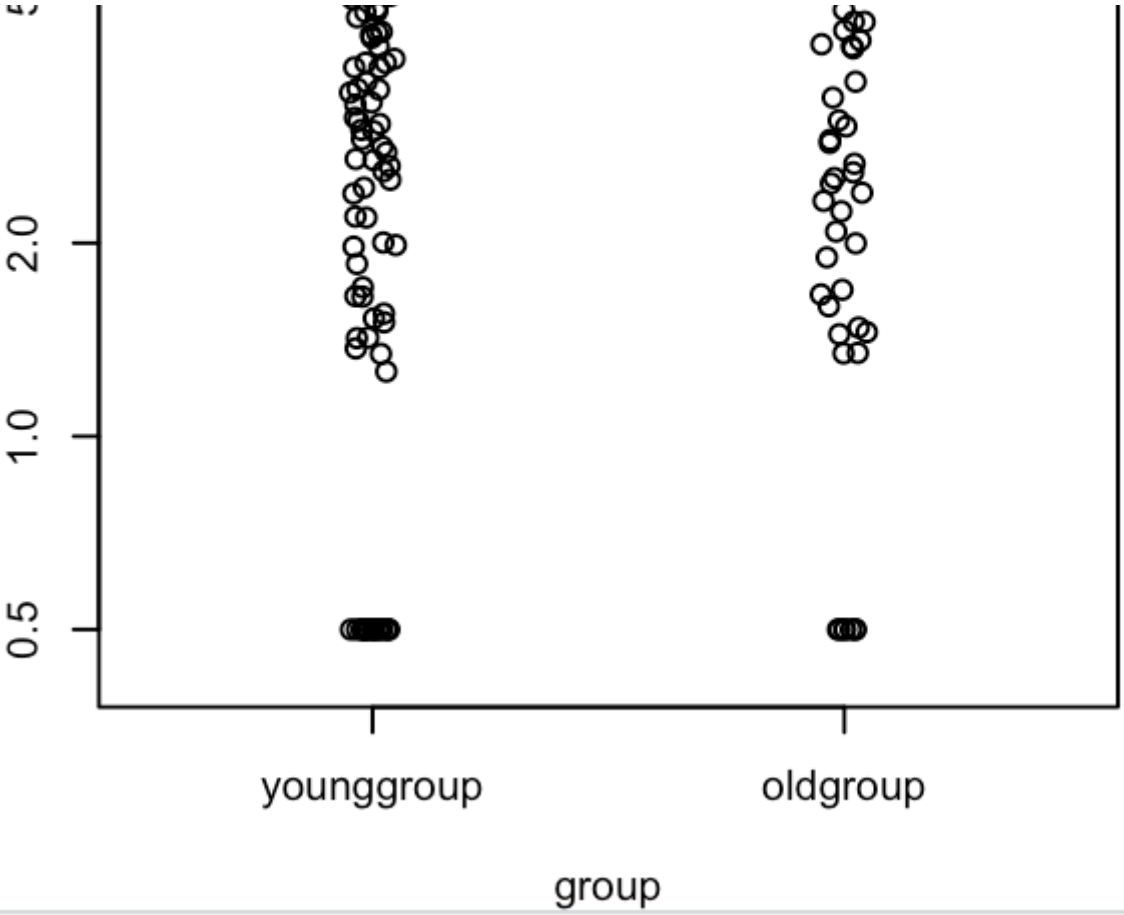
Plot counts

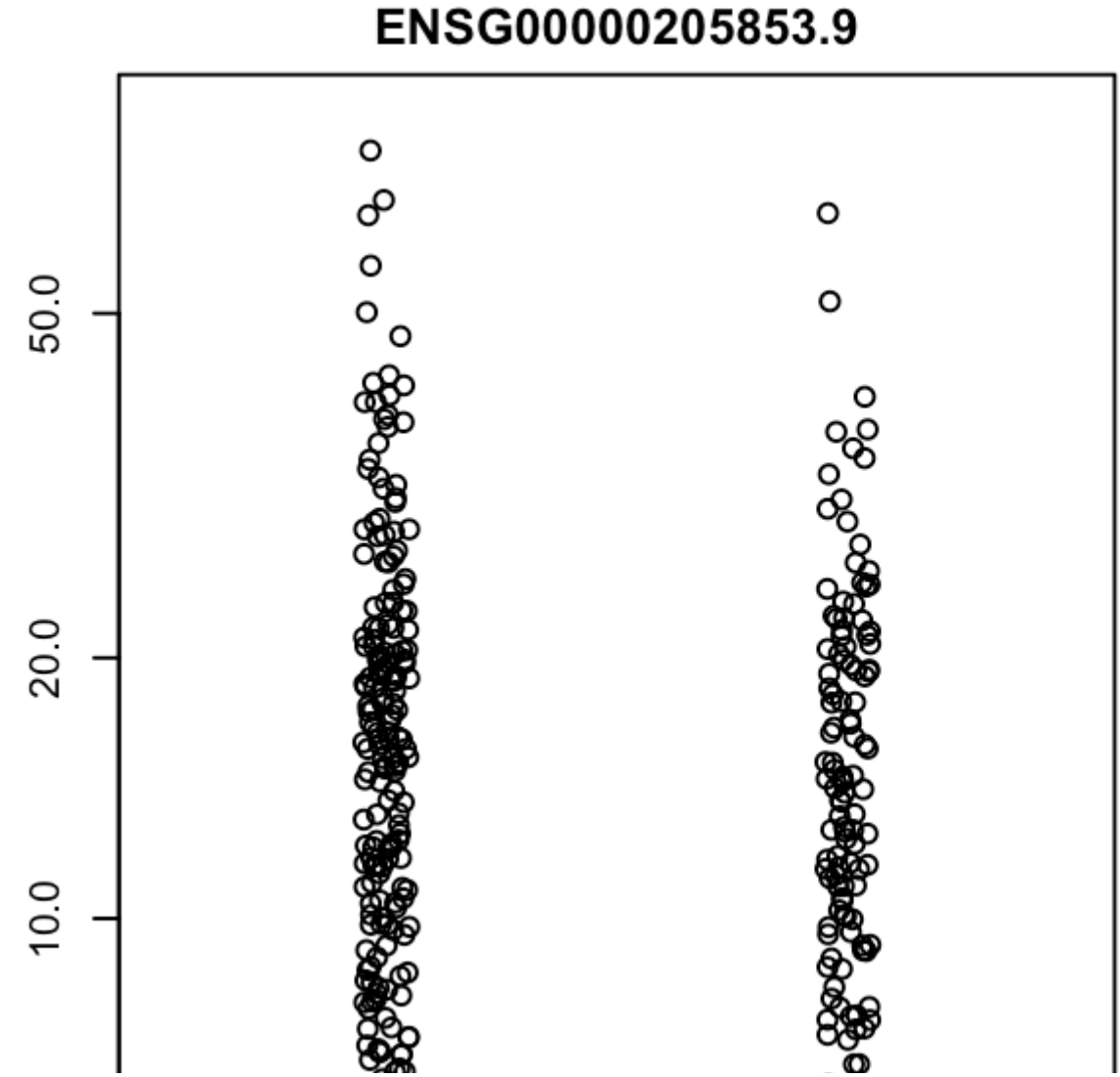
It can also be useful to examine the counts of reads for a single gene across the groups.

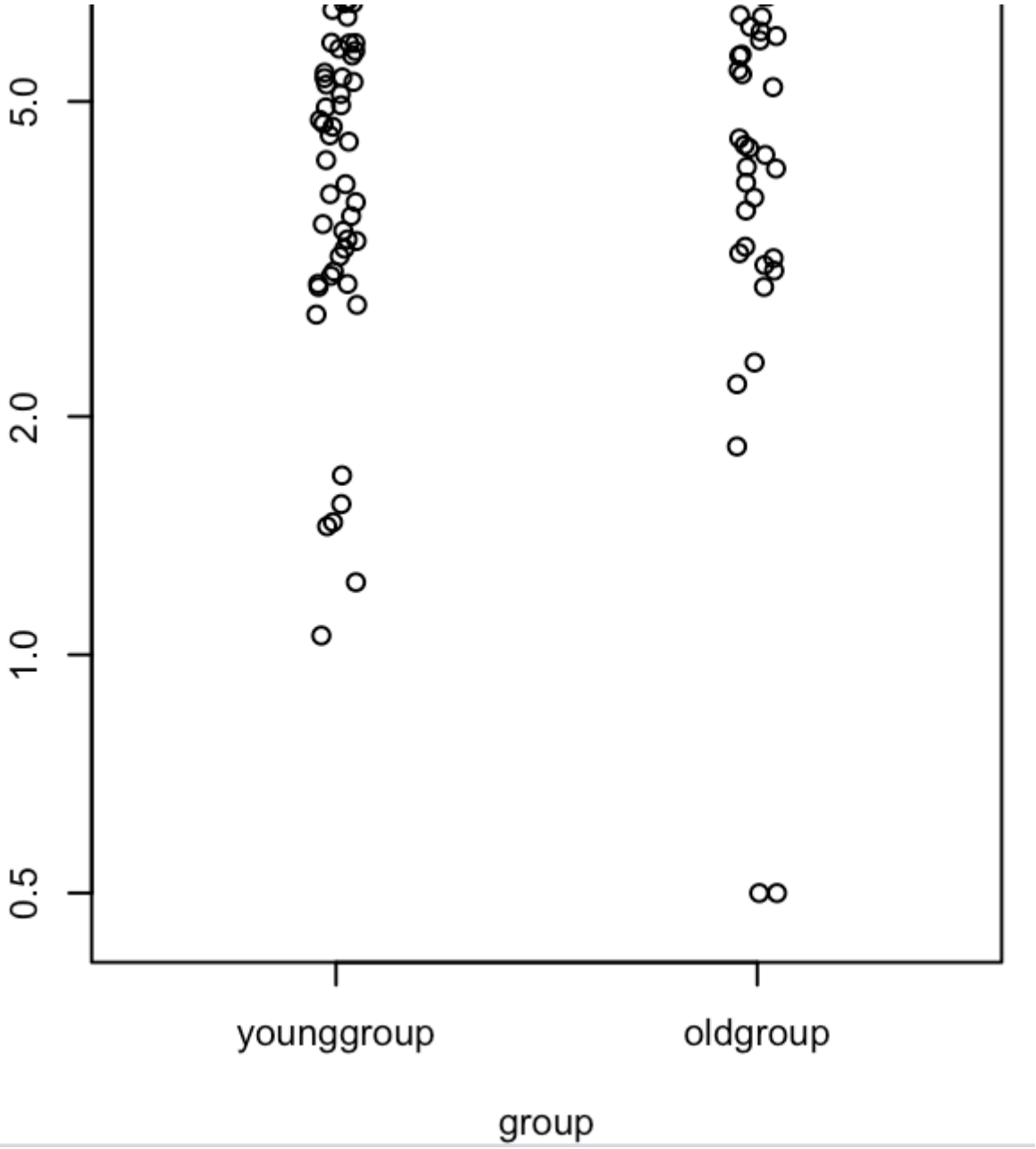
I select a few genes that is related to prostate cancer.

```
plotCounts(dds, "ENSG00000004809.12", intgroup="condition")
plotCounts(dds, "ENSG00000205853.9", intgroup="condition")
```



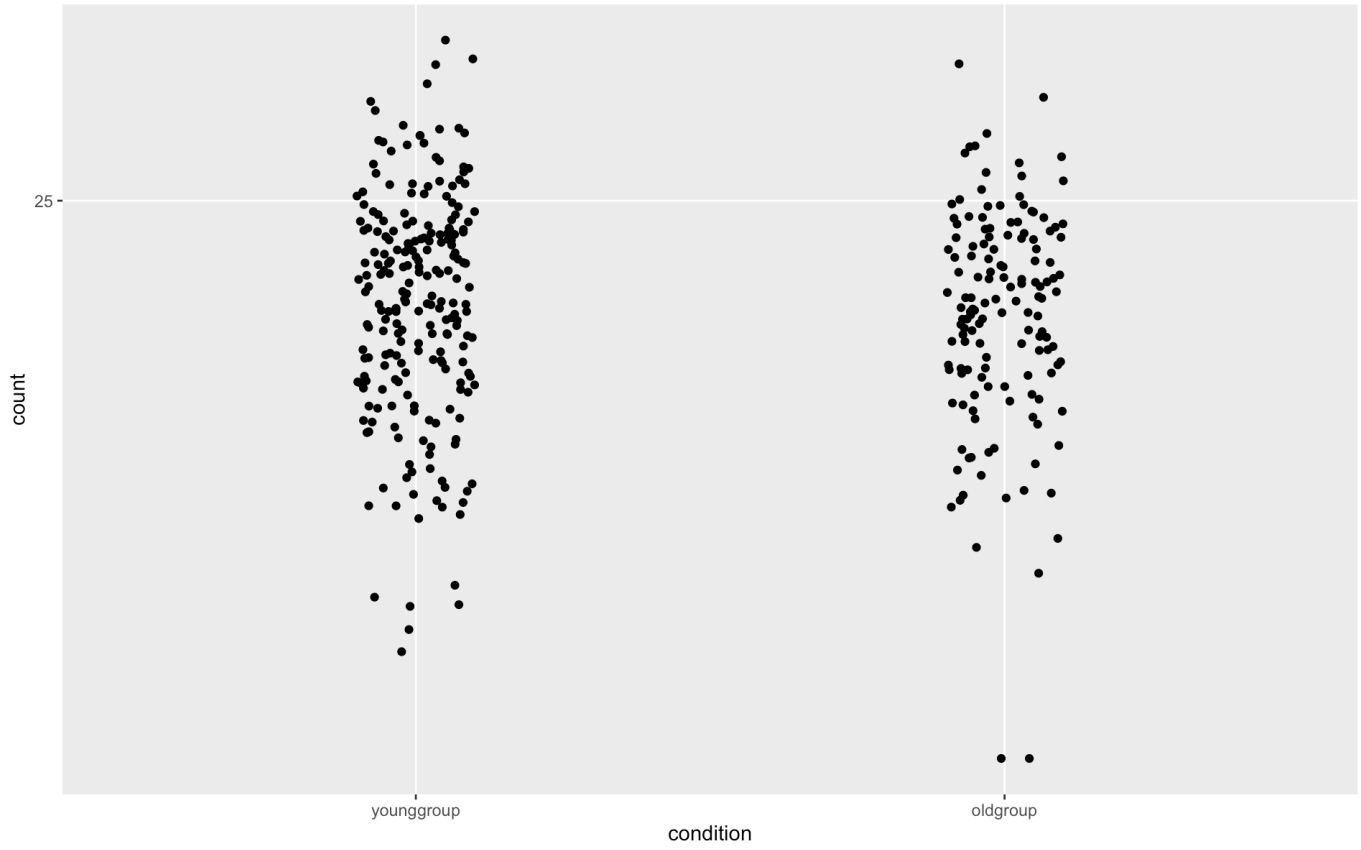

0.5
1.0
2.0
younggroup
oldgroup
group


ENSG00000205853.9
10.0
20.0
50.0



customized plotting.

```
d <- plotCounts(dds, "ENSG00000205853.9", 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))
```



I do not see any difference between younggroup and oldgroup by plot counts right now.

More information on results columns

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

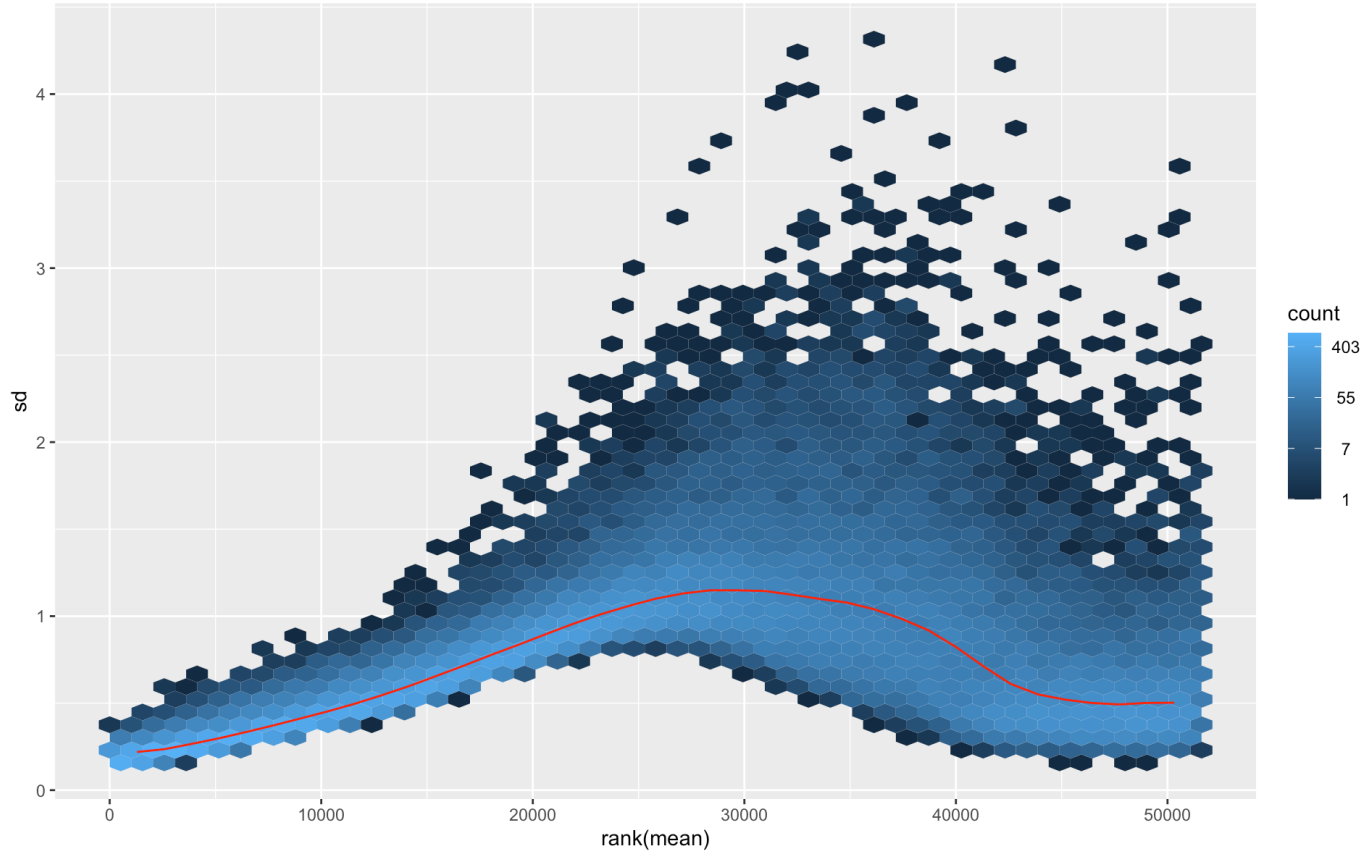
Data transformations and visualization

Extracting transformed values

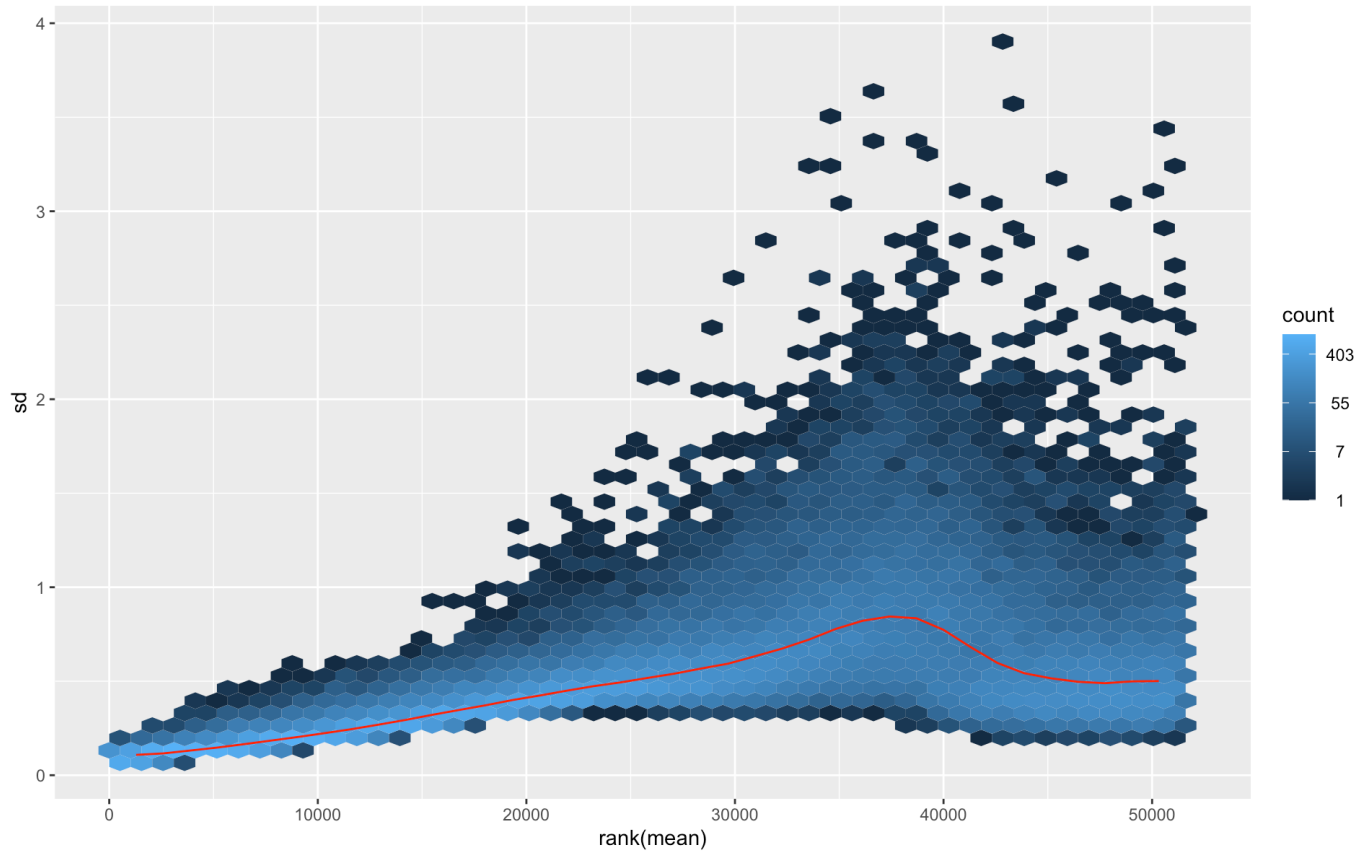
```
vsd <- vst(dds, blind=FALSE)
head(assay(vsd), 3)
```

This gives log2(n + 1).

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




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



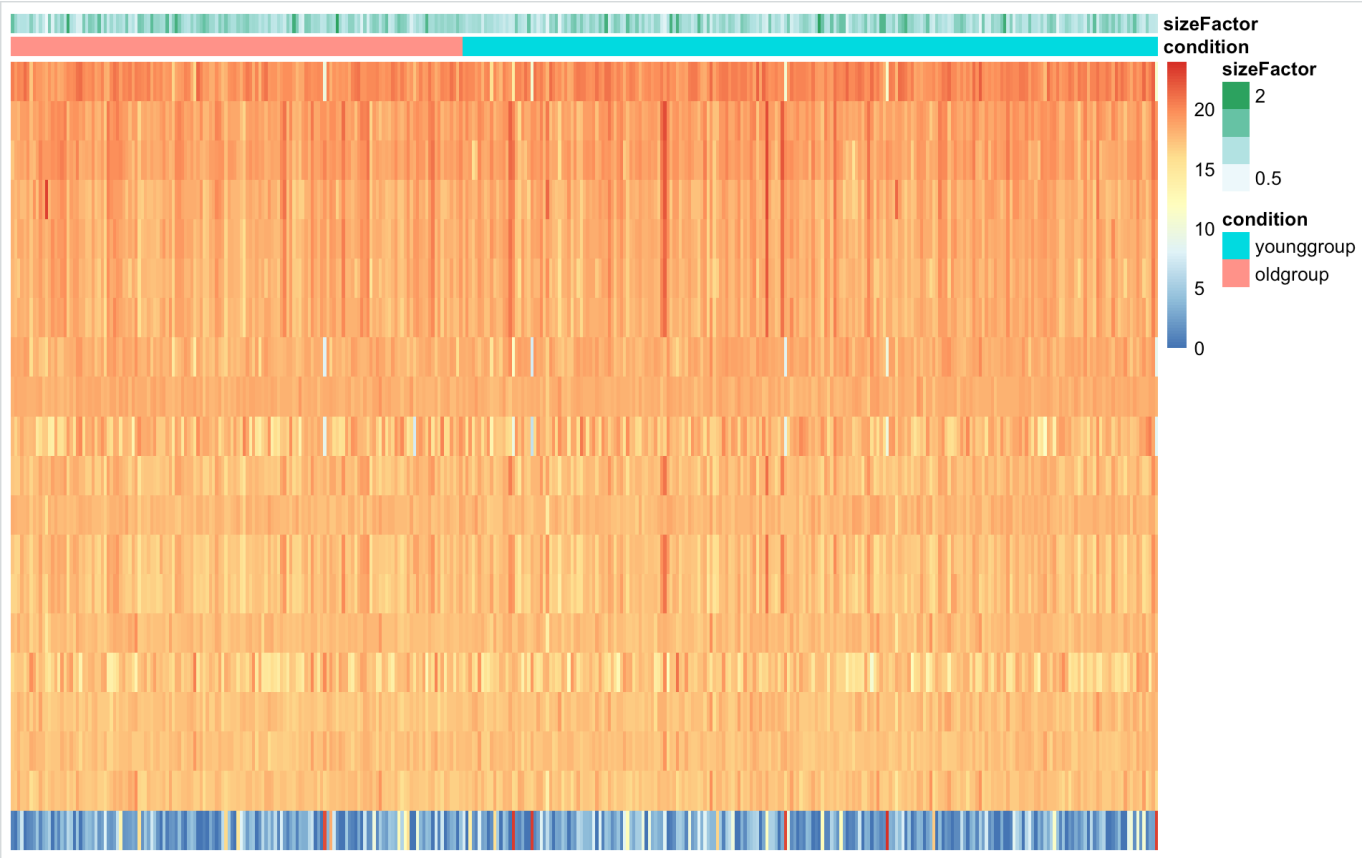
Standard deviation and mean are calculated row-wise from the expression matrix.

Heatmap of the count matrix.

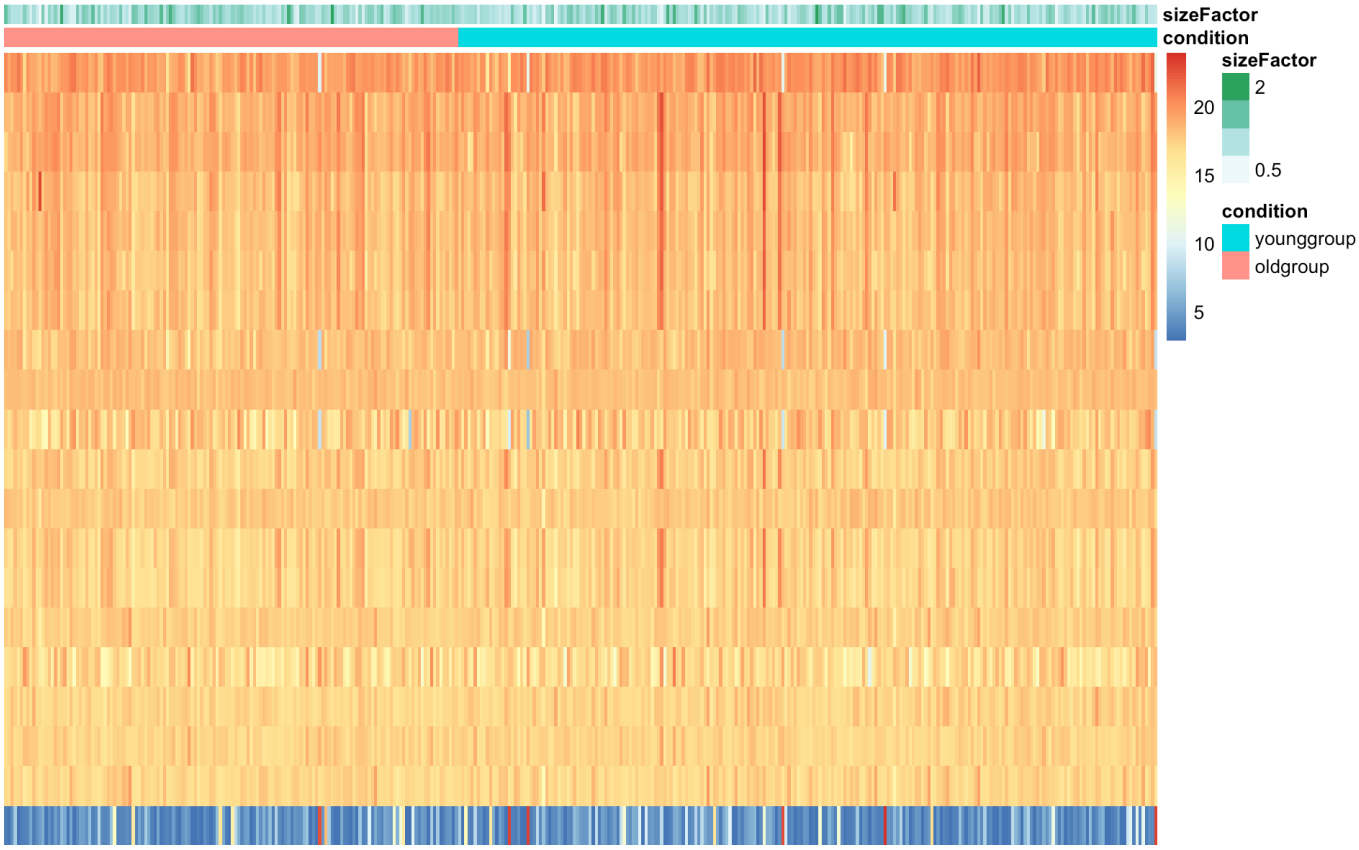
To explore a count matrix, it is often instructive to look at it as a heatmap.

```
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)
```



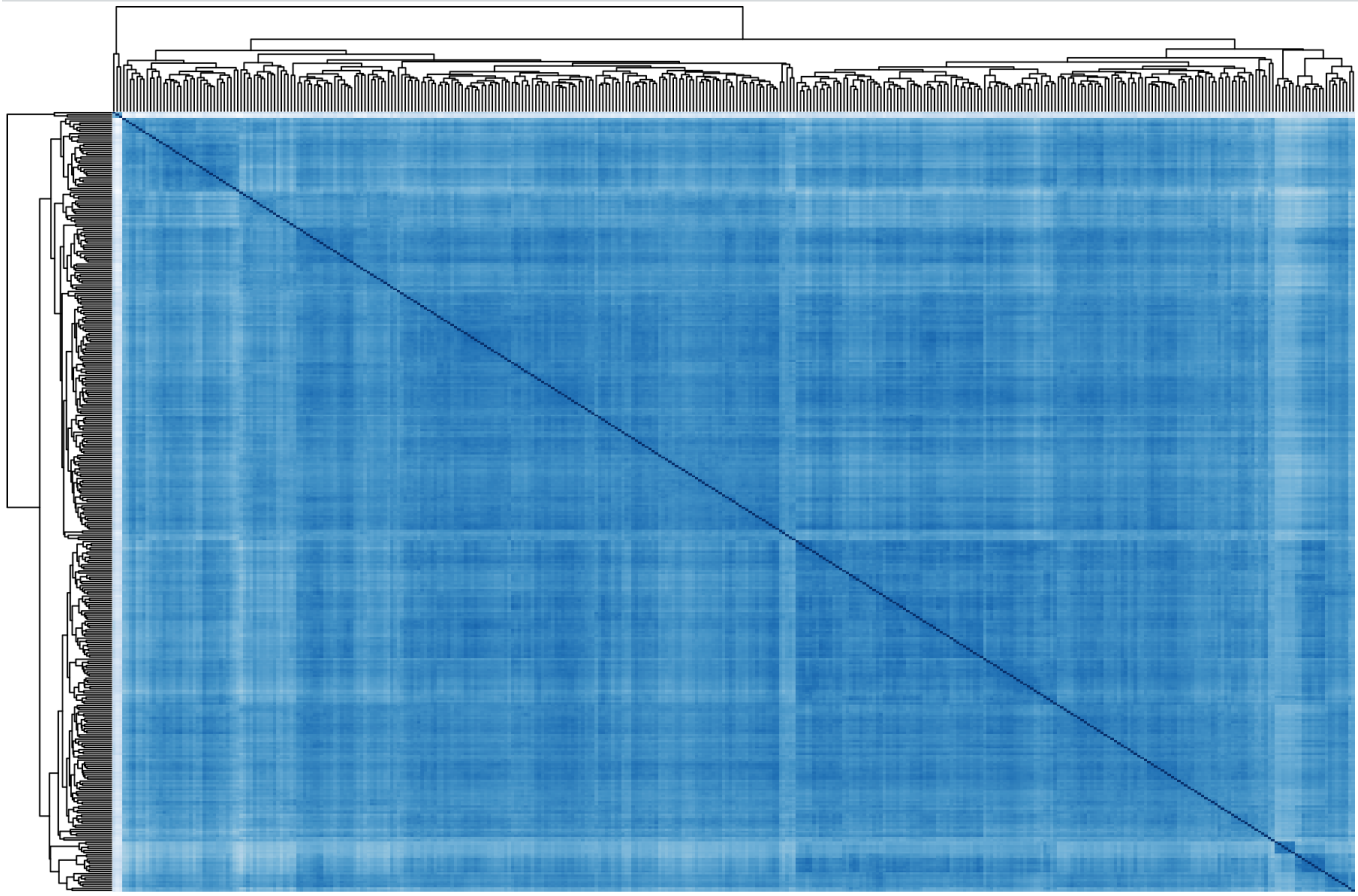

```
pheatmap(assay(vsd)[select,], cluster_rows=FALSE, show_rownames=FALSE,
         cluster_cols=FALSE, annotation_col=df, show_colnames = FALSE)
```



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

Heatmap of the sample-to-sample distances.

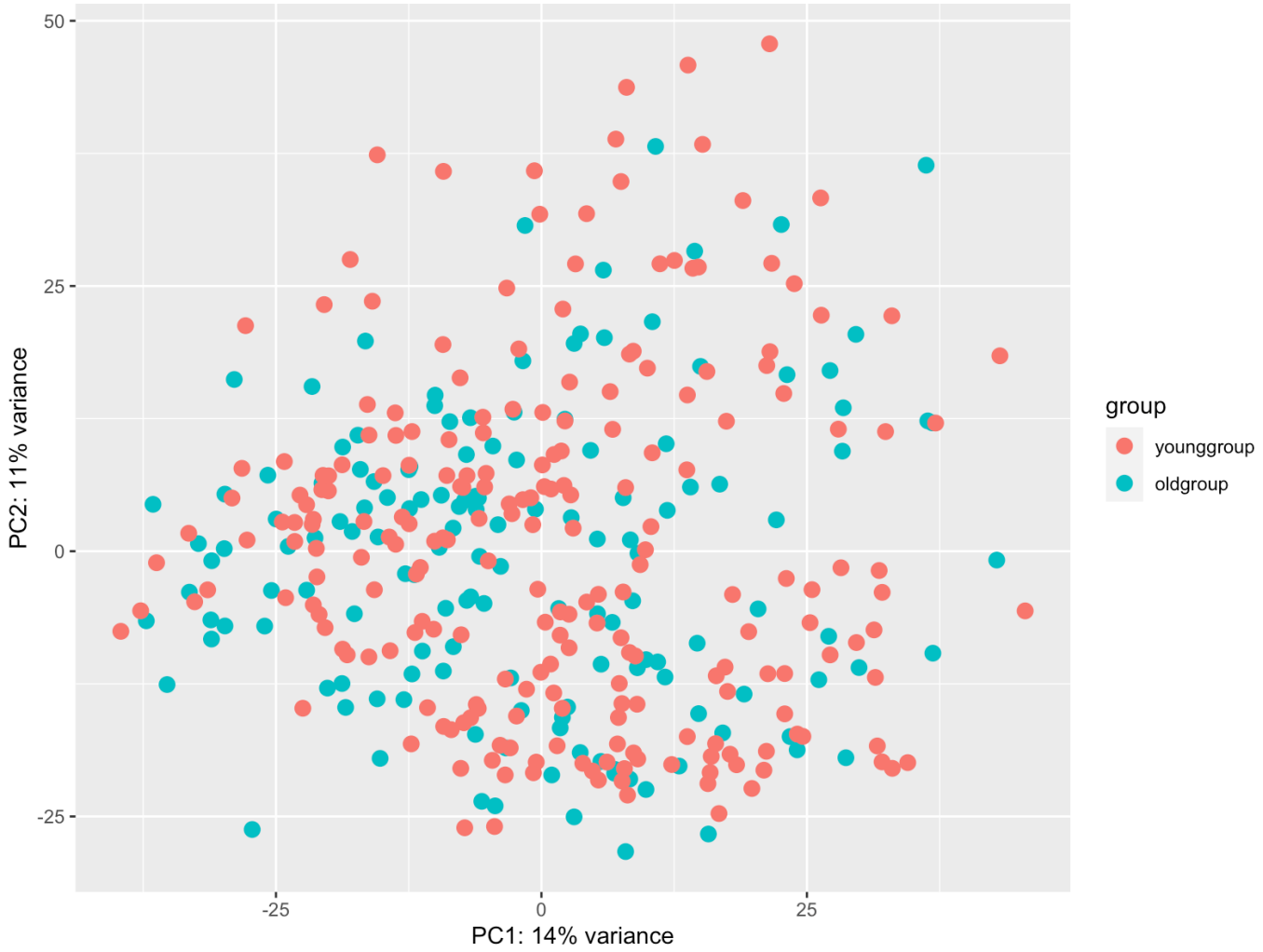
```
sampleDists <- dist(t(assay(vsd)))
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$condition, vsd$type, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,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



Principal component plot.

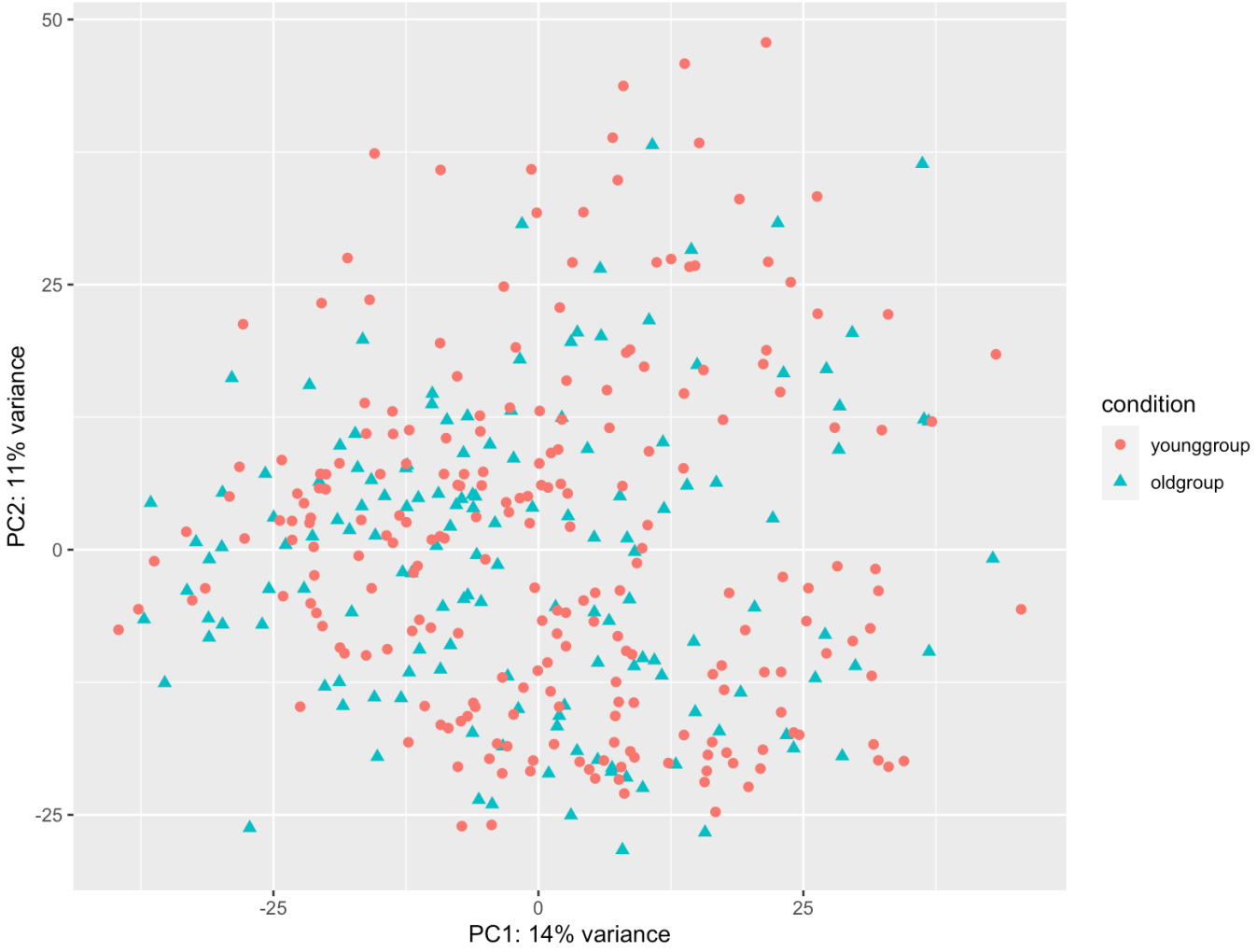
It shows the samples in the 2D plane spanned by their first two principal components.

```
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 acount.

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 gene lists I have).

Heatmaps need to be fixed. (I fixed it with the sizeFactor and condition bar above the plot and on the right side.)

Known issues

I will change my p-value to 0.05 and log2foldchange 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)
rownames(resIHW) <- ens_id
rawcount<- resIHW
Ensembl_ID <- data.frame(Ensembl_ID = row.names(rawcount))
rownames(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_TO_Genename <- data.frame(gene_id = gene_id,
                                       gene_name = gene_name,
```

```
                                          stringsAsFactors = FALSE)
  return(Ensembl_ID_TO_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 with p-value < 0.05, log2FoldChange>=1 or log2FoldChange<=-1.

```

```

```
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")
```

127 genes are listed in "up.csv" and 74 genes are listed in "down.csv", 201 genes in total.

After getting up and down csv files, I save them as txt files. Then, I use the GSEA website and put all my up list 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						progestagen associated endometrial protein [Source:HGNC Symbol;Acc:HGNC:8573]

However, when I put all my down list 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. With samples less than 30 would be better to try this code.

I try to put genes name into the `plotCounts(dds, "ENSG00000004809.12", intgroup="condition")` code, however,it did not work.

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

In this study, my aim is to study the gene expression pattern in TCGA of different age men with prostate cancer by using DeSEQ2. After Downloading Data from GDC, I got 331 cases in total. (203 cases in young

group/ 128 cases in oldgroup.) . Using the DEseq2 allows me to interpret my dataset. Then, I set my p-value < 0.05 and log2foldchange >=1 or log2foldchange <=1. I got 127 unregulated genes and 74 down regulated genes, 201 genes in total. Then, I put my up regulated and down regulated gene list in GSEA respectively. In my upregulated gene list, I got 3 gene sets but only 4-6 genes overlaped. On the other side, after inputting the down regulated gene list, no overlaps found. In my opinion, I think that it would be interesting to see whether there is any difference by using a larger gap of age, like 30-40 years old compare to 60-70 years old. In the future, we should also include more samples or different datasets to confirm the results.

Files

I put all csv files in the folder of "Excel".

I put all htseq.counts files in the folder of "HTseq_counts_files".

I put all png images in the folder of "Images".

I put all PDF files in the folder of "PDF".

I put all scripts in the folder of "Scripts".

Other form of files are in the folder of " Other_files".

Deliverable

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