NGS

Alasia

Introduction:

For my project, I used differential gene expression (DGE) analysis with RNA-seq, which is both pertinent and a useful tool for the advancements of biomedicine. The data we are using is extracted from research done by Beth Israel Deaconess Medical Center and whose methods consisted of transfecting MDA-MB-231 breast cancer cells with 20nM control or NRDE2-targeting siRNAs, and collecting RNA after 48h for RNA-seq analysis. In other words, I used RNA-seq libraries from three biological replicates from control cell lines and three biological replicates in which the gene NRDE2 was silenced with RNAi. The libraries were single-end (SE) sequenced on an Illumina NextSeq platform. To conduct a DGE analysis, I processed the raw data and implemented a Salmon + tximport + DESeq2 workflow. The objective of this project is to characterize differentially expressed genes that may be impacted by knocking down NRDE2.

A synopsis of the experiment can be found here:

http://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

Materials:

The following is a table on Greene with the samples and their fastq files:

/scratch/work/courses/BI7653/project.2023/project_fastqs.txt

The fastq files are located in:

/scratch/work/courses/BI7653/project.2023/fastqs

GRCh38 genome "Gene Annotation" section of the Ensembl website:

https://www.ensembl.org/Homo_sapiens/Info/Index

GRCh38 file "Homo_sapiens.GRCh38.cdna.all.fa.gz"

Salmon index: https://salmon.readthedocs.io/en/latest/salmon.html

mapping file: /scratch/work/courses/BI7653/project.2023/tx2gene.csv

Reference human transcriptome file which reads were aligned against using using Salmon software package: (ftp://ftp.ensembl.org/pub/release-98/fasta/homo_sapiens/cdna/Homo_sapiens.GRCh38.cdna.abinitio.fa.gz)

Methods:

I first downloaded and unzipped all my fastq files from the project repository. I trimmed the fastqs with fastp using length_required 75 and –n_base_limit 50 to automatically remove adapters from single end reads and

polyG sequences introduced on NextSeq platform. After Trimming the reads, I downloaded the human reference transcriptome from the Ensembl website and then ran fastqc on the processed RNA-seq reads separately on each sample. Next, I generated a MultiQC report. Then, I ran Picard tools NormalizeFasta to strip everything after the transcript id, which is the first identifier in the Fasta header lines for each transcript on both the reference files and reads that were aligned against the reference human transcriptome that I retrieved from the Ensembl website. After, I created a Salmon index and then ran Salmon in mapping-based mode using a command appropriate for single-end data. Finally, I conducted a standard analysis of differential gene expression using DESEQ. Included in my analysis is a table with the total number of reads and the mapping rate for each sample, the number of statistically significant genes at 0.05 FDR, the number of biologically relevant differentially expressed genes defined using a change in gene expression of two-fold or greater, a table with the 10 most highly significant differentially expressed genes, a sample PCA, an MA plot, and dispersion-by-mean plot, and a raw p-value histogram. I am reporting shrunken log fold-change estimates to normalize errors from sequencing and sampling and give better estimates of the log2 fold-change. The statistical approach used for the multiple-test correction method was to use a false discovery rate (FDR) adjusted p value, and I chose level of significance to be 0.05, so that genes with an adjusted p value greater than .05 were rejected to avoid false positives. What was unusual in the multiqc report was that none of the samples passed the sequence duplication levels and the per base sequence content. The library type Salmon inferred for the input reads was stranded, derived from the reverse strand (SR).

Specifically, I did the following:

```
cp /scratch/work/courses/BI7653/project.2023/fastqs/*.fastq.gz .

[am12179@cs061 ngs.finalproject]$ gunzip -c  SRR7819990.fastq.gz > SRR7819990.fastq
[am12179@cs061 ngs.finalproject]$ gunzip -c  SRR7819991.fastq.gz > SRR7819991.fastq
[am12179@cs061 ngs.finalproject]$ gunzip -c  SRR7819992.fastq.gz > SRR7819992.fastq
[am12179@cs061 ngs.finalproject]$ gunzip -c  SRR7819993.fastq.gz > SRR7819993.fastq
[am12179@cs061 ngs.finalproject]$ gunzip -c  SRR7819994.fastq.gz > SRR7819994.fastq
[am12179@cs061 ngs.finalproject]$ gunzip -c  SRR7819995.fastq.gz > SRR7819995.fastq
```

Task 1: Trim the fastqs with fastp using appropriate settings to automatically remove adapters from single end reads and polyG sequences introduced on NextSeq platforms (see Week 2) The following is a table on Greene with the samples and their fastq files:

```
#!/bin/bash
#
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --time=8:00:00
#SBATCH --mem=16GB
#SBATCH --job-name=slurm_template
#SBATCH --mail-type=FAIL
#SBATCH --mail-user=am12179@nyu.edu

module purge
module load fastp/intel/0.20.1

fastp -i SRR7819990.fastq -o  SRR7819990_out.fastq \
        --length_required 75 \
        --n_base_limit 50 --html SRR7819990.fastp.html \
        --json SRR7819990.fastp.json
```

```
fastp -i    SRR7819991.fastq -o SRR7819991_out.fastq \
        --length_required  \
        --n_base_limit 50 --html SRR7819991.fastp.html \
        --json SRR7819991.fastp.json
fastp -i    SRR7819992.fastq -o  SRR7819992_out.fastq \
        --length_required 75 \
        --n_base_limit 50 --html SRR7819992.fastp.html \
        --json SRR7819992.fastp.json
fastp -i    SRR7819993.fastq -o  SRR7819993_out.fastq \
        --length_required 75 \
        --n_base_limit 50 --html SRR7819993.fastp.html \
        --json SRR7819993.fastp.json
fastp -i    SRR7819994.fastq -o  SRR7819994_out.fastq \
        --length_required 75 \
        --n_base_limit 50 --html SRR7819994.fastp.html \
        --json SRR7819994.fastp.json
fastp -i    SRR7819995.fastq -o  SRR7819995_out.fastq \
        --length_required 75 \
        --n_base_limit 50 --html SRR7819995.fastp.html \
        --json SRR7819995.fastp.json
```

Task 2: Run fastqc on the processed RNA-seq reads separately on each sample

```
am12179@log-3 ngs.optiona]$ wget ftp://ftp.ensembl.org/pub/release-98/fasta/homo_sapiens/cdna/Homo_sapi
gz
--2023-04-30 16:20:12--  ftp://ftp.ensembl.org/pub/release-98/fasta/homo_sapiens/cdna/Homo_sapiens.GRCh
193.62.193.139
Connecting to ftp.ensembl.org (ftp.ensembl.org)|193.62.193.139|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /pub/release-98/fasta/homo_sapiens/cdna ... done.
==> SIZE Homo_sapiens.GRCh38.cdna.abinitio.fa.gz ... 20601482
==> PASV ... done.    ==> RETR Homo_sapiens.GRCh38.cdna.abinitio.fa.gz ... done.
Length: 20601482 (20M) (unauthoritative)
Homo_sapiens.GRCh38.cdna.abiniti 100%[=============================================================>]  19.65
```

```
[am12179@cs061 ngs.finalproject]$ module load fastqc/0.11.9
[am12179@cs061 ngs.finalproject]$ fastqc  SRR7819990_out.fastq -
[am12179@cs061 ngs.finalproject]$ fastqc  SRR7819991_out.fastq
[am12179@cs061 ngs.finalproject]$ fastqc  SRR7819992_out.fastq
[am12179@cs061 ngs.finalproject]$ fastqc  SRR7819993_out.fastq
[am12179@cs061 ngs.finalproject]$ fastqc  SRR7819994_out.fastq
[am12179@cs061 ngs.finalproject]$ fastqc  SRR7819995_out.fastq
```

Task 2: Generate a MultiQC report

```
module load multiqc/1.9
[am12179@cs090 ngs.optiona]$ find $PWD -name \*fastqc.zip > fastqc_files.txt
[am12179@cs090 ngs.optiona]$ multiqc --file-list /scratch/am12179/ngs.optiona/fastqc_files.txt
```

Task 3: Run Picard tools NormalizeFasta to strip everything after transcript id, which is the first identifier in the Fasta header lines for each transcript.

```
#!/bin/bash
#
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --time=8:00:00
#SBATCH --mem=8GB
#SBATCH --job-name=slurm_template
#SBATCH --mail-type=FAIL
#SBATCH --mail-user=am12179@nyu.edu

module load picard/2.23.8


java -jar $PICARD_JAR NormalizeFasta \
      I=Homo_sapiens.GRCh38.cdna.all.fa \
      O=Homo_sapiens.GRCh38.cdna.all.norm.fa

module purge
```

```
#!/bin/bash
#
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --time=8:00:00
#SBATCH --mem=8GB
#SBATCH --job-name=slurm_template
#SBATCH --mail-type=FAIL
#SBATCH --mail-user=am12179@nyu.edu

module load picard/2.23.8


java -jar $PICARD_JAR NormalizeFasta \
        I=Homo_sapiens.GRCh38.cdna.abinitio.fa   \
      O=norm_sequence_Homo_sapiens.GRCh38.cdna.abinitio.fa

module purge
```

Task 4: Create Salmon index.

```
#!/bin/bash
#
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --time=8:00:00
#SBATCH --mem=16GB
#SBATCH --job-name=slurm_
#SBATCH --mail-type=FAIL
#SBATCH --mail-user=am12719@nyu.edu

module purge

module load salmon/1.4.0

salmon index -t norm_sequence_Homo_sapiens.GRCh38.cdna.abinitio.fa -i Homo_sapiens_GRCh38_cdna_norm_tra

module purge
```

Task 5: Run Salmon in mapping-based mode using a command appropriate for single-end data.

```
#!/bin/bash
#
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=4
#SBATCH --time=24:00:00
#SBATCH --mem=8GB
#SBATCH --job-name=salmon
#SBATCH --mail-type=FAIL
#SBATCH --mail-user=am12179@nyu.edu
#SBATCH --array=1-6

module purge

module load salmon/1.4.0

salmon quant \
    -i  Homo_sapiens_GRCh38_cdna_norm_transcripts_index \
    -l A \
    -r SRR7819990_out.fastq \
    --validateMappings \
    --gcBias \
    --threads ${SLURM_CPUS_PER_TASK} \
    -o SRR7819990.transcripts_quant

salmon quant \
    -i  Homo_sapiens_GRCh38_cdna_norm_transcripts_index \
```

```
    -l A \
    -r SRR7819991_out.fastq \
    --validateMappings \
    --gcBias \
    --threads ${SLURM_CPUS_PER_TASK} \
    -o SRR7819991.transcripts_quant

salmon quant \
    -i  Homo_sapiens_GRCh38_cdna_norm_transcripts_index \
    -l A \
    -r SRR7819992_out.fastq \
    --validateMappings \
    --gcBias \
    --threads ${SLURM_CPUS_PER_TASK} \
    -o SRR7819992.transcripts_quant

salmon quant \
    -i  Homo_sapiens_GRCh38_cdna_norm_transcripts_index \
    -l A \
    -r SRR7819993_out.fastq \
    --validateMappings \
    --gcBias \
    --threads ${SLURM_CPUS_PER_TASK} \
    -o SRR7819993.transcripts_quant

salmon quant \
    -i  Homo_sapiens_GRCh38_cdna_norm_transcripts_index \
    -l A \
    -r SRR7819994_out.fastq \
    --validateMappings \
    --gcBias \
    --threads ${SLURM_CPUS_PER_TASK} \
    -o SRR7819994.transcripts_quant

salmon quant \
    -i  Homo_sapiens_GRCh38_cdna_norm_transcripts_index \
    -l A \
    -r SRR7819995_out.fastq \
    --validateMappings \
    --gcBias \
    --threads ${SLURM_CPUS_PER_TASK} \
    -o SRR7819995.transcripts_quant

module purge
```

```
library("tximport")
```

Task 6: Results

```
## Warning: package 'tximport' was built under R version 4.2.2
```

```
sample_names = c("SRR7819990",
                 "SRR7819991",
                 "SRR7819992",
                 "SRR7819993",
                 "SRR7819994",
                 "SRR7819995")
sample_condition = c(rep('control',3),rep('treated',3))

files = file.path("C:/Users/Alasi/Downloads",
                  paste(sample_names,".transcripts_quant",sep=""),
                  'quant.sf')

names(files) = sample_names

print(files)
```

```
##                                                                SRR7819990
## "C:/Users/Alasi/Downloads/SRR7819990.transcripts_quant/quant.sf"
##                                                                SRR7819991
## "C:/Users/Alasi/Downloads/SRR7819991.transcripts_quant/quant.sf"
##                                                                SRR7819992
## "C:/Users/Alasi/Downloads/SRR7819992.transcripts_quant/quant.sf"
##                                                                SRR7819993
## "C:/Users/Alasi/Downloads/SRR7819993.transcripts_quant/quant.sf"
##                                                                SRR7819994
## "C:/Users/Alasi/Downloads/SRR7819994.transcripts_quant/quant.sf"
##                                                                SRR7819995
## "C:/Users/Alasi/Downloads/SRR7819995.transcripts_quant/quant.sf"
```

Table with the total number of reads and the mapping rate for each sample

```
tx2gene = read.csv("tx2gene.csv",
                   header=F,
                   sep=",")

tx2gene$V1<- gsub("\\..*","",tx2gene$V1)
tx2gene$V2<- gsub("\\..*","",tx2gene$V2)
head (tx2gene)
```

```
##                 V1              V2
## 1 ENST00000632684 ENSG00000282431
## 2 ENST00000631435 ENSG00000282253
## 3 ENST00000448914 ENSG00000228985
## 4 ENST00000434970 ENSG00000237235
## 5 ENST00000415118 ENSG00000223997
## 6 ENST00000604642 ENSG00000270961
```

```
txi = tximport(files,
               type="salmon",
               tx2gene=tx2gene,
               txOut=TRUE)
```

```
## reading in files with read_tsv
```

```
## 1 2 3 4 5 6
```

```
samples = data.frame(sample_names=sample_names,
                     condition=sample_condition)

row.names(samples) = sample_names
```

```
library("DESeq2")
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Warning: package 'BiocGenerics' was built under R version 4.2.1
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
## Warning: package 'IRanges' was built under R version 4.2.1
```

```
##
## Attaching package: 'IRanges'
```

```
## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 4.2.2

## Loading required package: SummarizedExperiment

## Warning: package 'SummarizedExperiment' was built under R version 4.2.1

## Loading required package: MatrixGenerics

## Warning: package 'MatrixGenerics' was built under R version 4.2.1

## Loading required package: matrixStats

## Warning: package 'matrixStats' was built under R version 4.2.2

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians

## Warning: multiple methods tables found for 'aperm'

## Warning: replacing previous import 'BiocGenerics::aperm' by
## 'DelayedArray::aperm' when loading 'SummarizedExperiment'
```

```
ddsTxi = DESeqDataSetFromTximport(txi,
                                  colData = samples,
                                  design = ~ condition)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors

## using counts and average transcript lengths from tximport
```

```
print(ddsTxi)
```

```
## class: DESeqDataSet
## dim: 49361 6
## metadata(1): version
## assays(2): counts avgTxLength
## rownames(49361): GENSCAN00000000001 GENSCAN00000000002 ...
##   GENSCAN00000056642 GENSCAN00000056643
## rowData names(0):
## colnames(6): SRR7819990 SRR7819991 ... SRR7819994 SRR7819995
## colData names(2): sample_names condition
```

```
keep = rowSums(counts(ddsTxi)) >=10 #filter data
ddsTxi = ddsTxi[keep,]
```

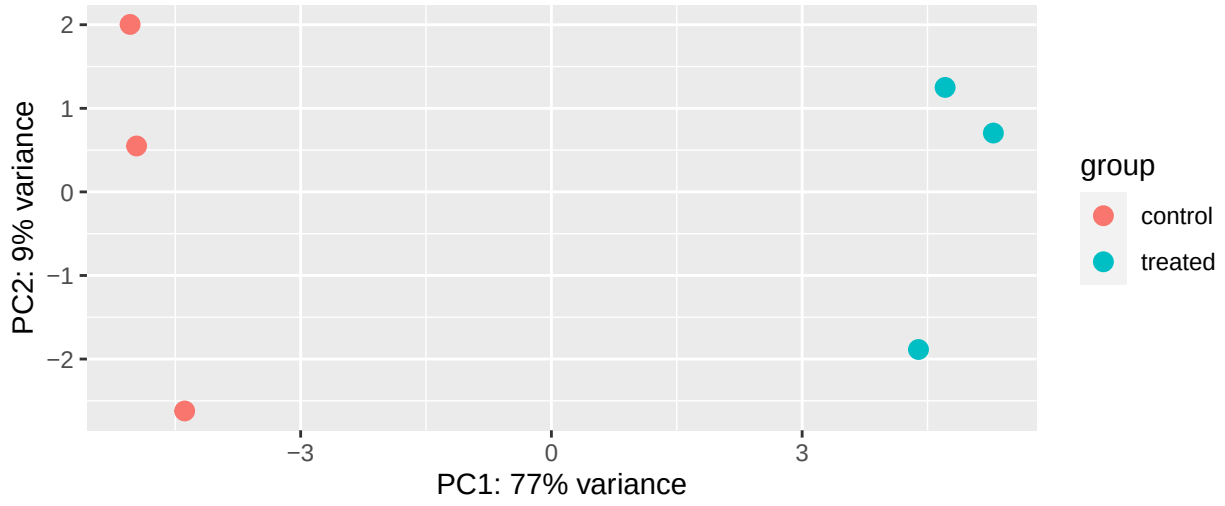
```
ddsTxi = DESeq(ddsTxi)
```

```
## estimating size factors

## using 'avgTxLength' from assays(dds), correcting for library size

## estimating dispersions

## gene-wise dispersion estimates
```

```
## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

PCA

```
rld = rlog(ddsTxi)
plotPCA(object = rld)
```



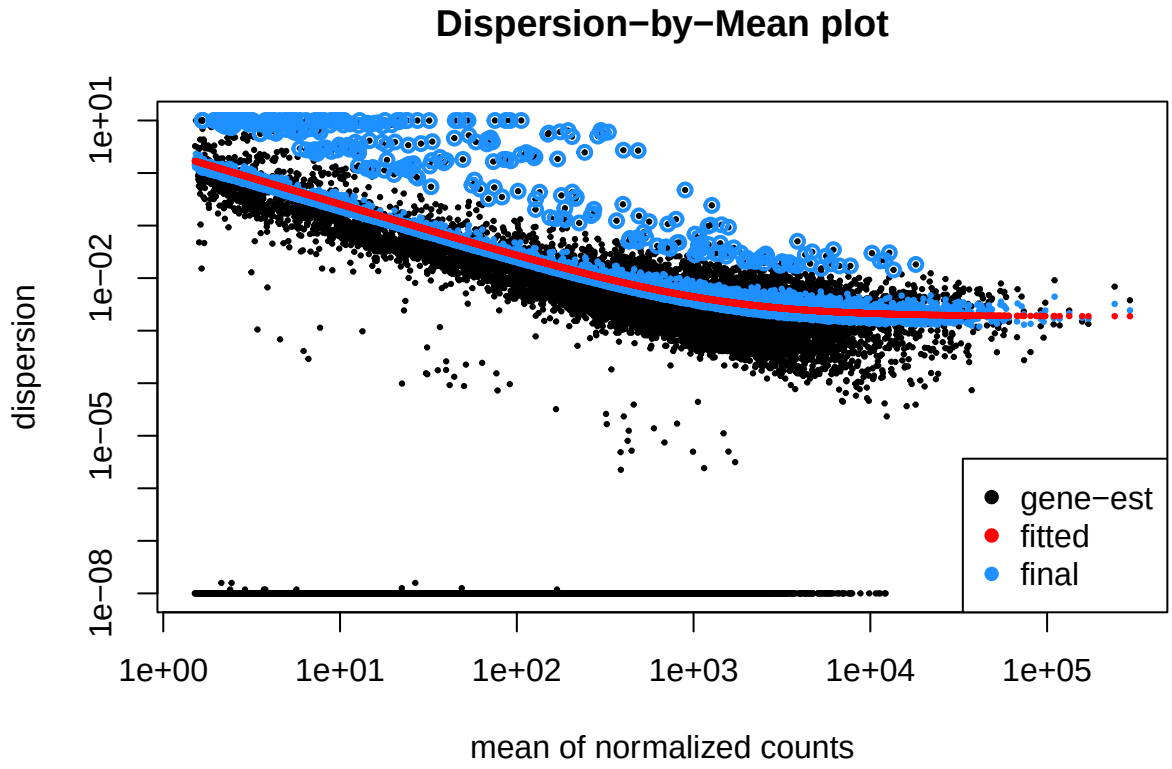
Dispersion-by-mean plot

```
estimateDispersions(ddsTxi, fitType = "parametric")
```

```
## found already estimated dispersions, replacing these

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates
```

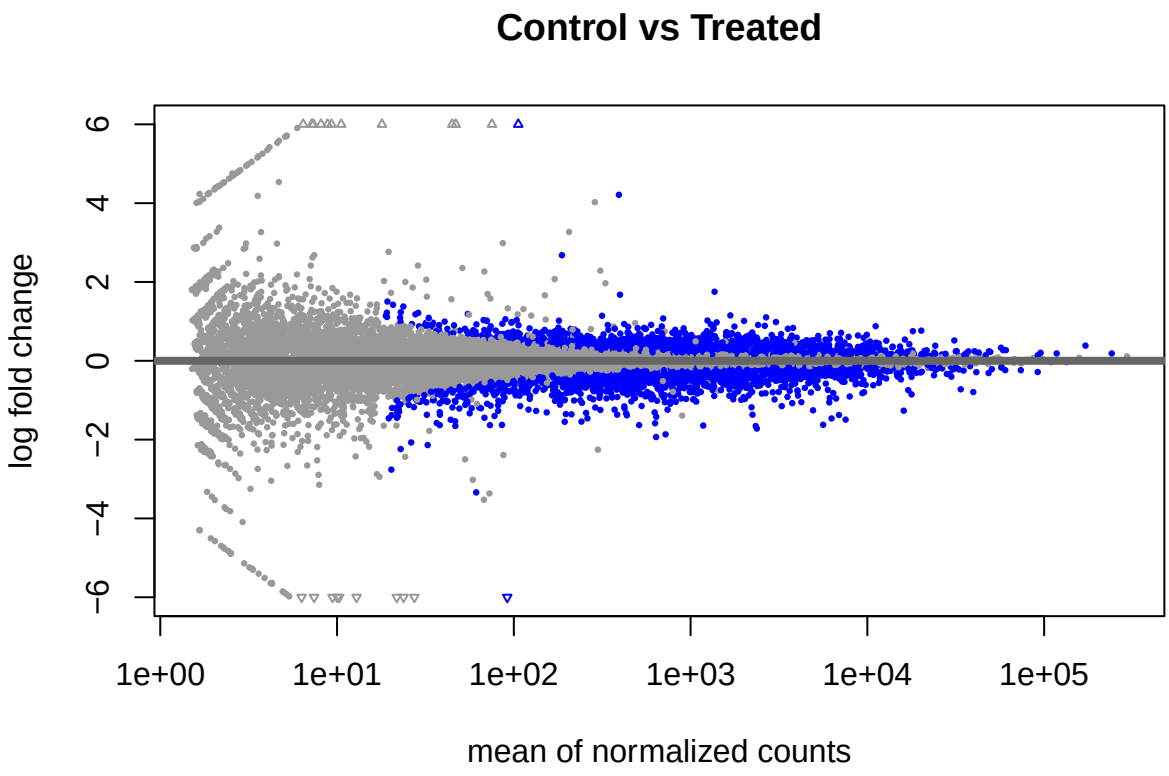
```
## class: DESeqDataSet
## dim: 17387 6
## metadata(1): version
## assays(6): counts avgTxLength ... H cooks
## rownames(17387): GENSCAN00000000001 GENSCAN00000000005 ...
##   GENSCAN00000056636 GENSCAN00000056639
## rowData names(10): baseMean baseVar ... dispOutlier dispMAP
## colnames(6): SRR7819990 SRR7819991 ... SRR7819994 SRR7819995
## colData names(2): sample_names condition
```

```
plotDispEsts(object = ddsTxi,
             main="Dispersion-by-Mean plot")
```



MA Plot

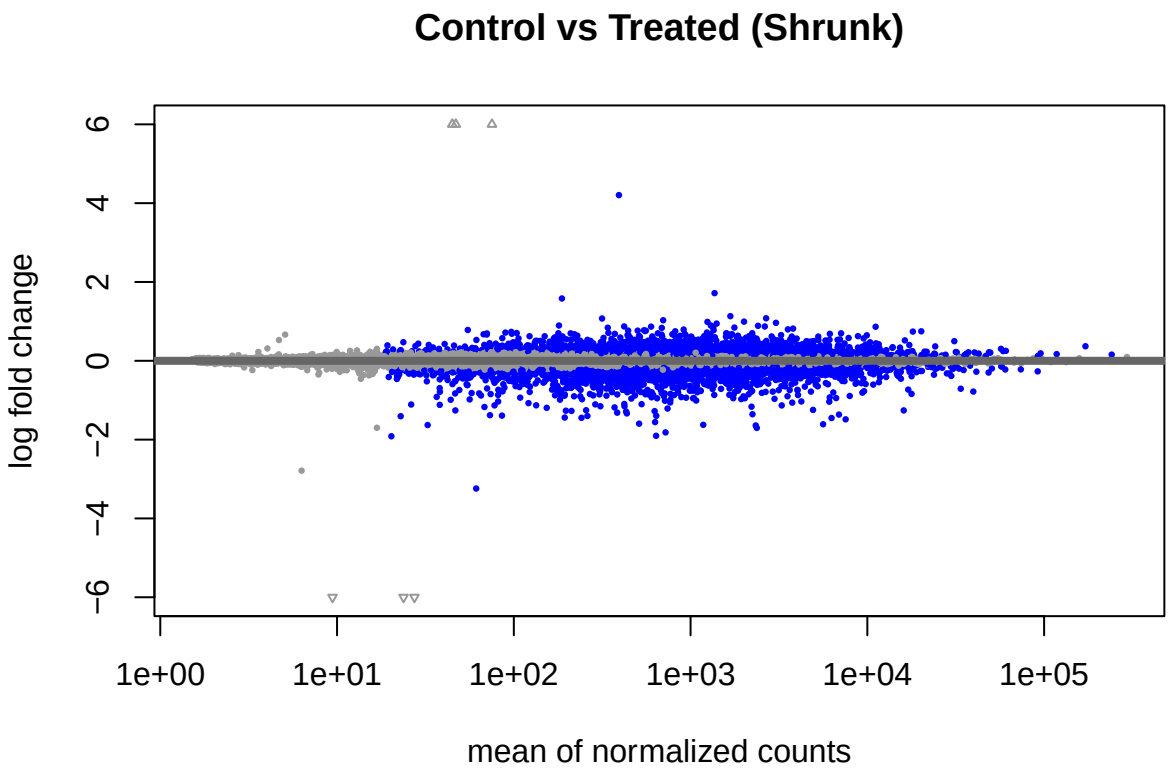
```
res = results(ddsTxi, contrast = c('condition',
                                   'control',
                                   'treated'))
write.table(res,file = "DE_Seq2_res_ALL_genes.txt", sep = "\t")

plotMA(res,
       ylim = c(-6,6),
       main = "Control vs Treated")
```




```
resshrunk = lfcShrink(ddsTxi, contrast = c('condition',
                                           'control',
                                           'treated'), type = "ashr")
```

```
## 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
```

```
plotMA(resshrunk,
       ylim = c(-6,6),
       main = "Control vs Treated (Shrunk)")
```




```
resshrunk = resshrunk[complete.cases(resshrunk),] # remove NAs

resshrunk_ordered = resshrunk[order(resshrunk$padj),]
```

Raw p-value histogram

```
library('ggplot2')
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
ggplot(as.data.frame(resshrunk_ordered),aes(pvalue)) + geom_histogram(fill="light blue",color='black')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

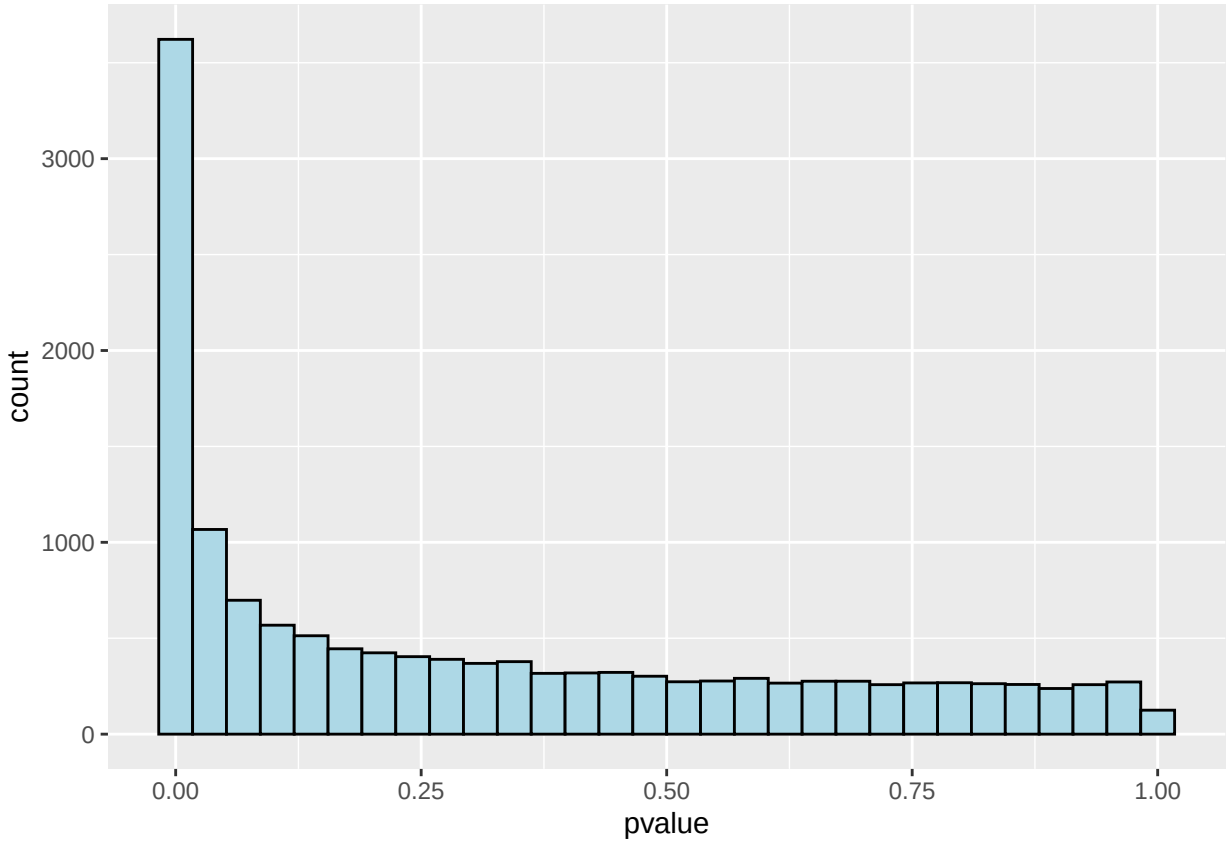


Table with the 10 most highly significant differentially expressed genes

```
head(resshrunk_ordered, n = 10)
```

```
## log2 fold change (MMSE): condition control vs treated
## Wald test p-value: condition control vs treated
## DataFrame with 10 rows and 5 columns
##                     baseMean log2FoldChange     lfcSE       pvalue         padj
##                    <numeric>      <numeric> <numeric>    <numeric>    <numeric>
## GENSCAN00000005890  7534.461       -1.48560 0.0474823 5.98711e-217 8.38494e-213
## GENSCAN00000041049 16060.689       -1.25931 0.0442843 4.28541e-181 3.00086e-177
## GENSCAN00000029520  5620.673       -1.61029 0.0644299 1.29506e-139 6.04575e-136
## GENSCAN00000042588  2337.157       -1.64159 0.0666371 1.01783e-135 3.56368e-132
## GENSCAN00000034116  6275.041       -1.45271 0.0600708 1.93770e-131 5.42751e-128
## GENSCAN00000042880   393.092        4.20433 0.1733653 2.10737e-130 4.91895e-127
## GENSCAN00000023753  6909.312       -1.36500 0.0569802 1.64936e-129 3.29991e-126
## GENSCAN00000042223  2369.357       -1.70265 0.0710233 1.00843e-128 1.76539e-125
## GENSCAN00000045535  4928.876       -1.24700 0.0537174 1.79811e-122 2.79805e-119
## GENSCAN00000032864  2239.065       -1.35663 0.0600102 6.78329e-116 9.50000e-113
```

The number of statistically relevant differentially expressed genes using 0.05 false discovery rate

```
resshrunk_fdr = resshrunk_ordered[resshrunk_ordered$padj<=0.05,]
summary(resshrunk_fdr)
```

```
##
## out of 3348 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 1610, 48%
## LFC < 0 (down)     : 1738, 52%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 17)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

The number of biologically relevant differentially expressed genes with a gene expression two fold or greater

```
resshrunk_fdr.cut_1 = resshrunk_fdr[resshrunk_fdr$log2FoldChange>=1,]
resshrunk_fdr.cut_2 = resshrunk_fdr[resshrunk_fdr$log2FoldChange<= -1,]

diffexp_genes = c(row.names(resshrunk_fdr.cut_1), row.names(resshrunk_fdr.cut_2))

summary(diffexp_genes)
```

```
##    Length     Class      Mode
##        62 character character
```

Discussion

My deseq analysis identified that there are 62 biologically and statistically relevant genes that are impacted by knocking down NRDE2. I came to this conclusion by first developing a table with the total number of reads and the mapping rate for each sample, a PCA plot which provided me with an indication of the distances between sample gene expression profiles and helped me detect batch effect, and an MA plot which showed gene expression distribution comparing my controls and treatments, highlighting differentially expressed genes. The dispersion-by-mean plot I included in my analysis demonstrated the gene which do not follow the modeling assumptions and thus have higher variability than others for biological (or technical) reasons, and my histogram plot revealed an enrichment of low p-values. Ultimately, I found that after setting my adjusted p value to less than 0.05 to reveal the number of statistically relevant differentially expressed, I had 3348 genes left. Then, I found that the number of biologically relevant differentially expressed genes defined using a change in gene expression of two-fold was 62.