Analysis of Control and Treated Breast Cancer: Utilization of Differential Gene Expression (DGE)

Zarifa Kabir

BI7653 - Spring 2024

Abstract

This study aimed to distinguish genes with differential expression (DGEs) in control and treated breast cancer cells through RNA sequencing data analysis. Employing the nf-core/rnaseq pipeline and DESeq2, the study detected notable alterations in gene expression, offering potential insights into the molecular mechanisms underlying the efficacy of breast cancer treatment. To complete the project, I downloaded the fastqs, processed the raw data, and conducted differential gene expression (DGE) analysis using Salmon, tximport, and DESeq2 workflow. Specifically, the nf-core/rnaseq pipeline was utilized to trim reads and execute Salmon to generate quant.sf files. Proper configuration of DESeq2 with the correct statistical design allowed for testing of differential gene expression between the RNAi lines and the control lines.

Introduction

A study published by Cold Spring Harbor Laboratory Press for the RNA Society investigated the functional consequences of NRDE2 depletion in MDA-MB-231 breast cancer cells through differential gene expression (DGE) analysis. RNA-sequencing experiments were conducted to delineate transcriptome profiles of control and NRDE2-depleted cell populations, followed by bioinformatic analysis to identify significant expression changes in key genes. The study revealed a notable shift in the transcriptome landscape post-NRDE2 depletion, suggesting a potential regulatory role for NRDE2 in cancer cell biology. Modulation was observed in genes governing essential cellular processes such as cell cycle progression, DNA damage response, and centrosome maturation. These findings highlight NRDE2 as a prospective therapeutic target for disrupting critical pathways in tumorigenesis and metastasis, contributing to a deeper understanding of its involvement in cancer. Based on the insights from the study, I conducted a study to focus on understanding the changes in gene expression, along with the method and effectiveness of the treatment. Several bioinformatics tools, discussed in Appendix B, are used to identify the gene expression between control and treated breast cancer cells and pinpoint differentially expressed genes (DGEs). To perform this analysis, I utilized various tools, including Salmon, tximport, and the DESeq2 workflow and produced statistical results.

Materials

- A script (download.sh) was used to download single-end RNA-sequencing data (FASTQ files) from an Illumina NextSeq sequencing run. The downloaded files can be grouped into two categories:
    - **Control Group:** These files (Control1, Control2, Control3) contain sequence data from replicates of the control cell line.
    - **Treated Group:** These files (Treated1, Treated2, Treated3) contain sequence data from replicates of the NRDE2 RNAi cell line (treated with RNAi).
- To analyze the RNA-sequencing data, I used the most recent human reference transcriptome and its corresponding annotation file from ENSEMBL (specific commands are detailed in Appendix A). These files are in FASTA and GTF formats.
    - **FASTA File**: Homo_sapiens.GRCh38.dna_sm.primary_assembly.fa.gz
    - **GTF File:** Homo_sapiens.GRCh38.111.gtf

Methods

Sequencing was performed on the samples with a specific focus on mRNA transcripts. The data obtained were single-read and did not follow a paired design. To maintain the integrity of the raw sequencing data, reads underwent pre-processing utilizing the fastp tool. This tool was instrumental in trimming adapters and eliminating low-quality sequences. This preliminary cleaning process was essential to ensure the accuracy and reliability of subsequent bioinformatics analyses.

The v3.14.0 version of nf-core/rnaseq pipeline was utilized to handle primary data processing tasks. This pipeline integrates TrimGalore! for read trimming, employing FastQC for adapter trimming and quality control. Following trimming, the reads were aligned to the human reference genome using Salmon, a tool adept at mapping reads to a transcriptome and quantifying their abundance based on specifications provided in the JSON config file. Salmon quantifies each transcript in the transcriptome assembly for each sample. Expression levels for each transcript are reported as Transcripts Per Million (TPMs). The nf-core pipeline also facilitated indexing of the human reference genome and execution of Salmon. Human reference genome and annotation files, available in GTF format from ENSEMBL, were utilized to ensure compatibility and precision in transcript recognition and quantification. DESeq2 transformed gene expression values into read counts, applied a method (apeglm) to improve fold-change calculations for genes with low expression, and controlled for false discoveries using a correction procedure (Benjamini-Hochberg). Finally, it calculated log2 fold-change (LFC) to precisely represent changes in gene expression due to the RNAi treatment relative to the control group.

To improve the reliability and accuracy of the analysis, I filtered out genes with low expression (less than 10 reads across all samples) before performing PCA. This step reduces noise and computational burden by focusing on genes with more reliable expression levels. Following this initial filtering, DESeq2 was used to preprocess the data. First, it normalized the raw counts using the median of ratios method to account for differences in library sizes. Then, it transformed the data using the "regularized log" (rlog) method. This transformation stabilizes variance across genes with varying expression levels, making the data more suitable for PCA. Finally, PCA was applied to the rlog-transformed data. This technique reduces the data's dimensionality by identifying a smaller set of uncorrelated components (principal components) that capture the most significant variation in the original data. The plotPCA function within DESeq2 was used to select the top 500 most variable genes, which are likely the most influential in driving the PCA and contributing to the observed differences between samples.

To improve the accuracy of gene expression changes, particularly for genes with low expression, the apeglm method was applied. It adjusts log2 fold-change values and borrows information from all genes to provide more stable and reliable estimates, especially for genes with low counts where fold-change calculations can be inflated. This helps reduce noise and improve the reliability of the results. DESeq2's model considers differences in the amount of RNA sequenced between samples (library size) and inherent variation in gene expression (dispersion). It calculates a normalization factor during dispersion estimation to account for library size. Additionally, it refines individual gene dispersion estimates by considering the overall trend across all genes, leading to more accurate and stable values.

Since analyzing high-throughput genomics data involves many tests simultaneously, there's a high risk of falsely identifying significant results. To address this challenge, I employed the Benjamin-Hochberg (BH) correction method. This technique adjusts p-values, which are statistical measures of significance, to account for the multiple tests performed. This helps control the false discovery rate (FDR), or the rate of mistakenly identifying non-significant genes as significant.

The analysis first calculated raw p-values for each gene using DESeq2, a software package that fits a specialized statistical model (negative binomial distribution) to the data. The BH method was then applied to these raw p-values to compute adjusted p-values (padj). Typically, a significance level of 5% is used, meaning a gene is considered differentially expressed if its adjusted p-value is lower than 0.05. This approach ensures a reliable selection of genes that are truly different between the treated and control groups, allowing for further investigation.

Results & Discussion

The original files for the MultiQC and execution reports are provided and submitted in html format, as specified in Appendix E.

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

Sample	Number of Reads	Mapping Rate %
SRR7819990	55663891	90.90073440405694 %
SRR7819991	58717637	92.08544122651704 %
SRR7819992	52054034	93.05537822873117 %
SRR7819993	53290204	92.39376166783193 %
SRR7819994	54595800	92.68125391697545 %
SRR7819995	42975036	92.65945411973398 %

Figure 1. Number of Reads and Mapping Rate

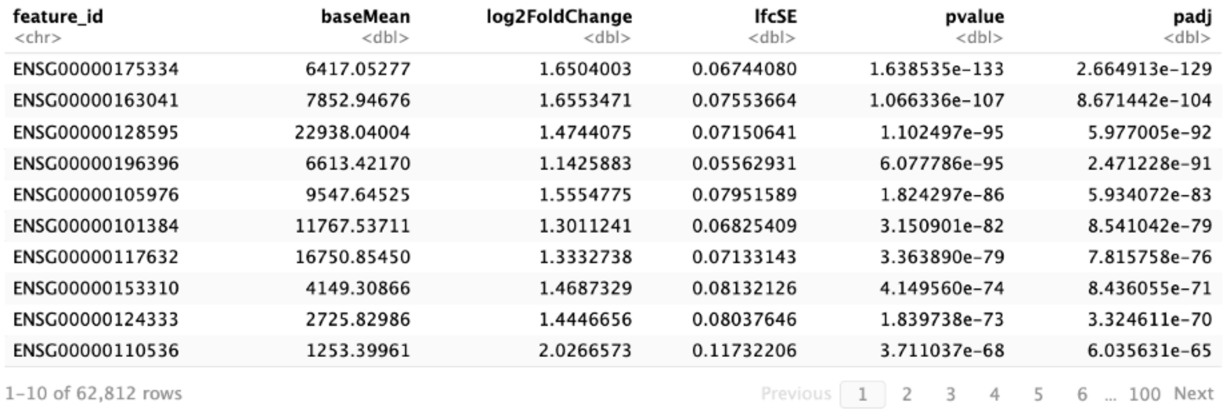
Table with 10 most highly significant differentially expressed genes(DGEs)

feature_id	**baseMean**	**L2FC**	**LFC Std. Error**	**P value**	**P-value adjusted**
ENSG00000175334	6423	1.66	0.0655	2.80e-142	4.53e-138
ENSG00000163041	7972	1.66	0.0717	1.47e-120	1.19e-116
ENSG00000196396	6618	1.15	0.0526	5.18e-107	2.79e-103
ENSG00000105976	9566	1.57	0.0758	2.24e- 96	9.07e- 93
ENSG00000128595	22967	1.48	0.0719	1.57e- 95	5.09e- 92
ENSG00000101384	11784	1.31	0.0660	5.47e- 89	1.48e- 85
ENSG00000124333	2742	1.48	0.0748	8.41e- 89	1.94e- 85
ENSG00000117632	1978	1.34	0.0703	3.26e- 82	6.59e- 79
ENSG00000180398	20588	0.909	0.0515	6.39e- 71	1.15e- 67
ENSG00000213281	6962	1.18	0.0675	1.11e- 69	1.79e- 66

Figure 2. 10 Highly significant DGEs

Number of genes with significantly higher and lower expression in RNAi vs. control

Figure 3 presents a comprehensive DESeq2 results table with gene IDs, base means, log2 fold-changes, standard errors, p-values, and adjusted p-values (FDR).



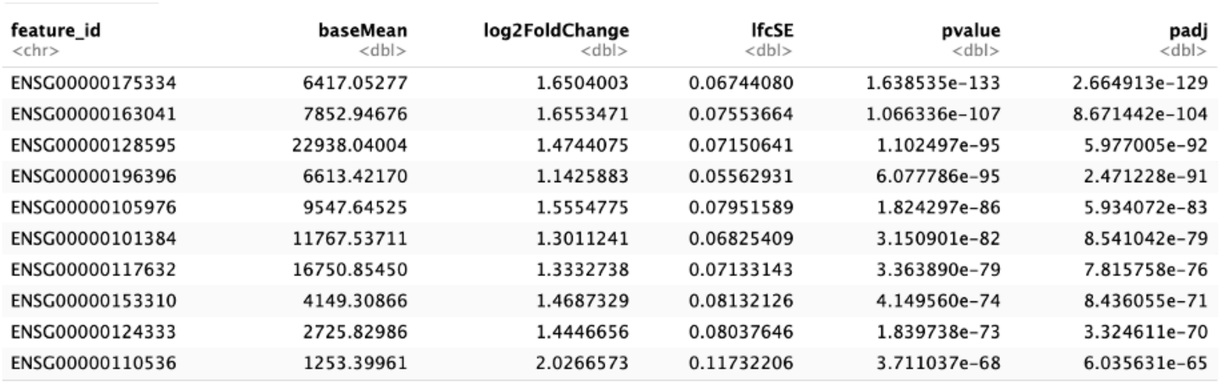
feature_id <chr>	baseMean <dbl>	log2FoldChange <dbl>	lfcSE <dbl>	pvalue <dbl>	padj <dbl>
ENSG00000175334	6417.05277	1.6504003	0.06744080	1.638535e-133	2.664913e-129
ENSG00000163041	7852.94676	1.6553471	0.07553664	1.066336e-107	8.671442e-104
ENSG00000128595	22938.04004	1.4744075	0.07150641	1.102497e-95	5.977005e-92
ENSG00000196396	6613.42170	1.1425883	0.05562931	6.077786e-95	2.471228e-91
ENSG00000105976	9547.64525	1.5554775	0.07951589	1.824297e-86	5.934072e-83
ENSG00000101384	11767.53711	1.3011241	0.06825409	3.150901e-82	8.541042e-79
ENSG00000117632	16750.85450	1.3332738	0.07133143	3.363890e-79	7.815758e-76
ENSG00000153310	4149.30866	1.4687329	0.08132126	4.149560e-74	8.436055e-71
ENSG00000124333	2725.82986	1.4446656	0.08037646	1.839738e-73	3.324611e-70
ENSG00000110536	1253.39961	2.0266573	0.11732206	3.711037e-68	6.035631e-65

1–10 of 62,812 rows

Previous 1 2 3 4 5 6 ... 100 Next

Figure 3. DESeq2 Results

Figure 4 presents a comprehensive DESeq2 results table with gene IDs, base means, log2 fold-changes, standard errors, p-values, and adjusted p-values (FDR).The calculation method for log2 fold-change (log2(RNAi/control)) guarantees an accurate representation of how RNAi treatment affects gene expression compared to the control group.



feature_id <chr>	baseMean <dbl>	log2FoldChange <dbl>	lfcSE <dbl>	pvalue <dbl>	padj <dbl>
ENSG00000175334	6417.05277	1.6504003	0.06744080	1.638535e-133	2.664913e-129
ENSG00000163041	7852.94676	1.6553471	0.07553664	1.066336e-107	8.671442e-104
ENSG00000128595	22938.04004	1.4744075	0.07150641	1.102497e-95	5.977005e-92
ENSG00000196396	6613.42170	1.1425883	0.05562931	6.077786e-95	2.471228e-91
ENSG00000105976	9547.64525	1.5554775	0.07951589	1.824297e-86	5.934072e-83
ENSG00000101384	11767.53711	1.3011241	0.06825409	3.150901e-82	8.541042e-79
ENSG00000117632	16750.85450	1.3332738	0.07133143	3.363890e-79	7.815758e-76
ENSG00000153310	4149.30866	1.4687329	0.08132126	4.149560e-74	8.436055e-71
ENSG00000124333	2725.82986	1.4446656	0.08037646	1.839738e-73	3.324611e-70
ENSG00000110536	1253.39961	2.0266573	0.11732206	3.711037e-68	6.035631e-65

Figure 4. Sorted Table of Significant DGEs

My analysis identified 3,515 genes potentially affected by the RNAi treatment. To focus on the most impactful changes, we narrowed down this list to genes with statistically significant expression differences (FDR < 5%). This subset likely represents genes most relevant to the biological effects of the RNAi treatment. The provided table ranks these genes by their adjusted p-value, highlighting the genes most dramatically affected under the experimental condition.

Principal Component Analysis Plot

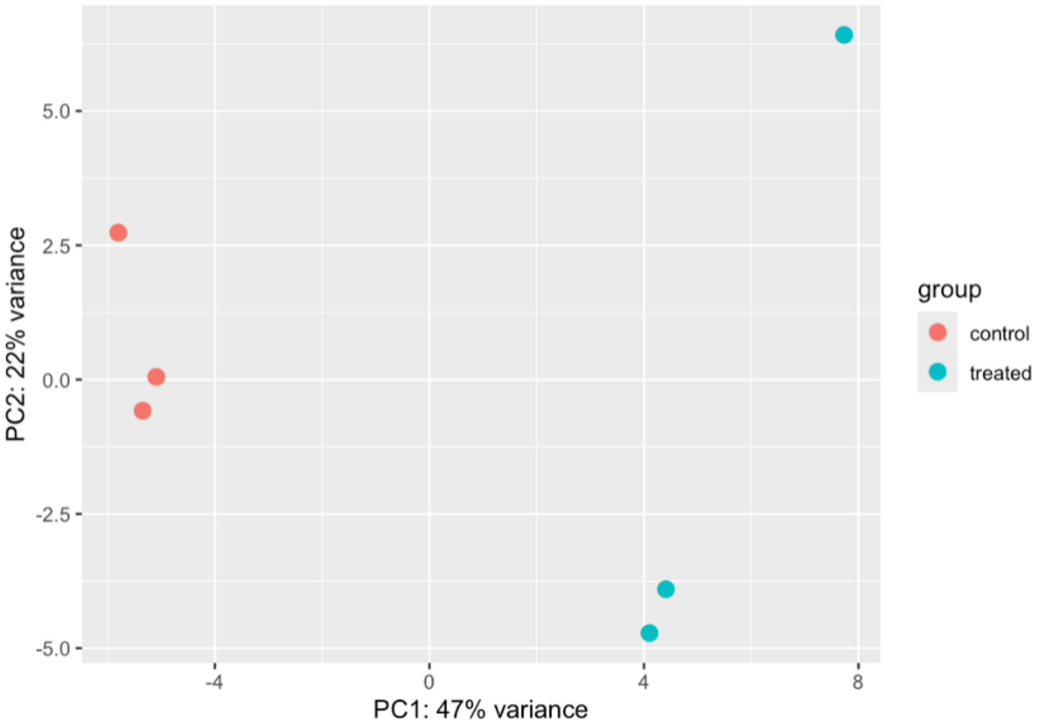


Figure 5. PCA Plot

The PCA plot highlights distinct gene expression profiles between control and treated samples. The first two principal components (PC1 and PC2) indicate a substantial portion of the variation in gene expression (69% combined, with PC1 contributing the most at 47%). This suggests these components effectively capture the key differences between the groups. Visually, the control samples (red) cluster tightly on the left side of the plot, while treated samples (blue) cluster on the right. This separation along PC1 indicates the treatment significantly impacts gene expression profiles. The distinct clustering suggests the treatment likely has a broad biological effect, influencing a large number of genes. This clear separation between groups reinforces the effectiveness of the treatment in altering gene expression and justifies further analysis to identify differentially expressed genes.

MA plot (after LFC shrinkage)

The MA plot in Figure 5 illustrates a trend where genes with low average expression (mean normalized counts) tend to have more scattered log2 fold changes. This dispersion reflects the higher inherent variability in measuring expression for genes with low counts. Despite the scatter, the plot also reveals a population of genes with substantial apparent up-regulation or down-regulation, suggesting potential biological effects for further investigation.

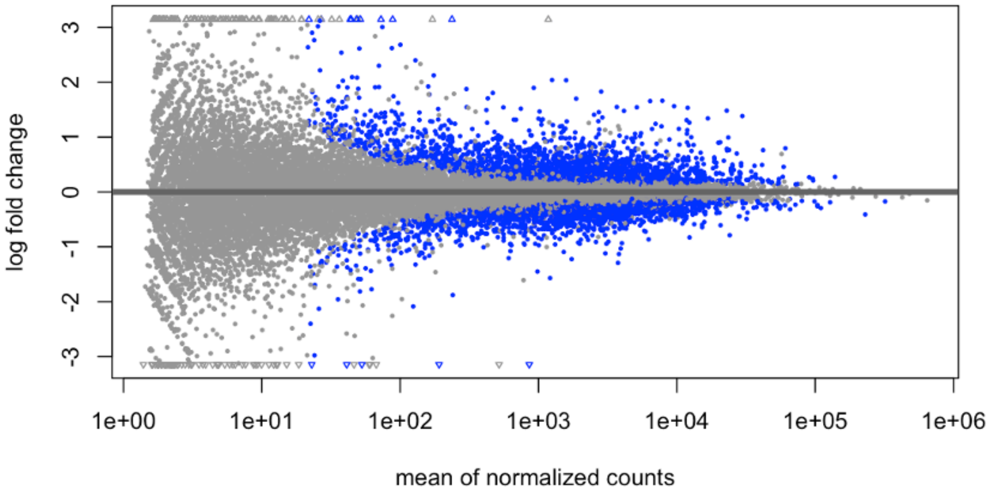


Figure 6. Pre-Shrinkage

Shrinking the data with the apeglm method (Figure 7) dramatically reduced the scatter in log2 fold changes, especially for genes with low expression. This suggests the method successfully filtered out noisy data in these genes. As a result, the data points bunch tighter around zero, indicating fewer unreliable genes with extreme fold-change values. The blue points represent genes identified as statistically significant after multiple testing adjustments, while grey points are non-significant. The shrinkage process led to fewer genes with dramatic fold-changes, reflecting a more rigorous and reliable assessment of differential expression. This is crucial to minimize false positives, which can arise from the inherent variability of genes expressed at low levels. Overall, shrinkage provides a more conservative and dependable set of differentially expressed genes for further analysis.

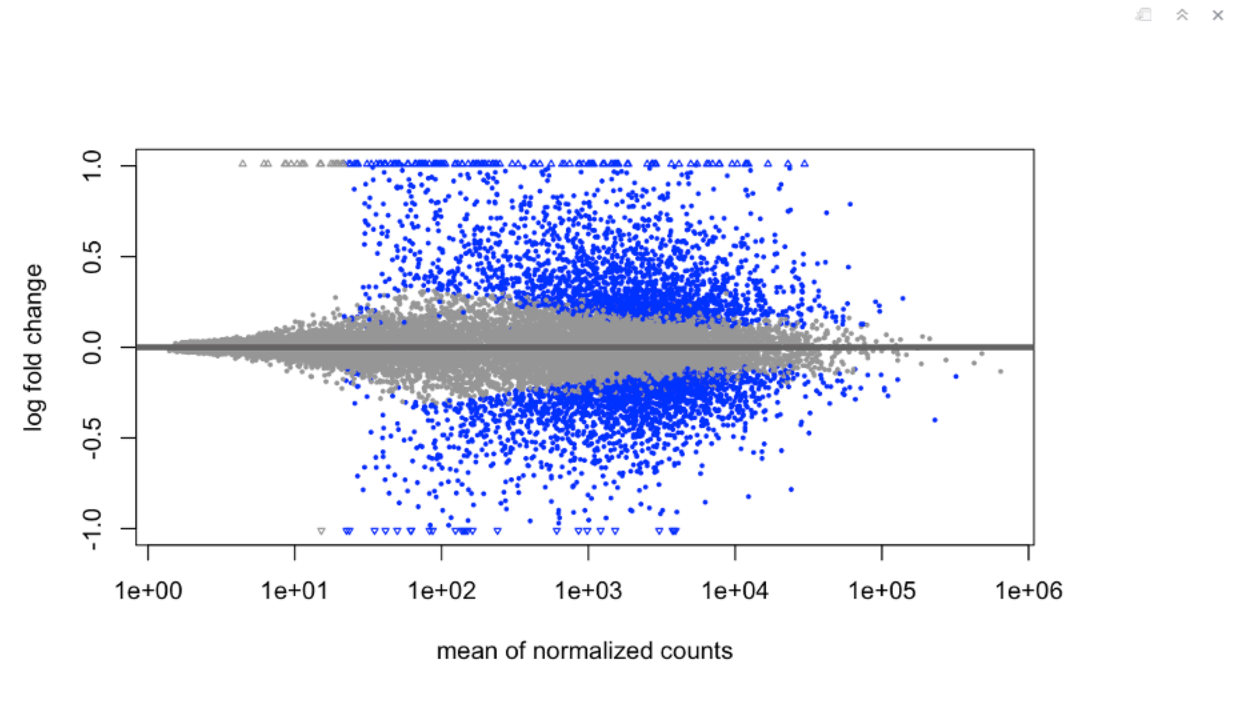


Figure 7. Post-Shrinkage

Dispersion-by-mean plot

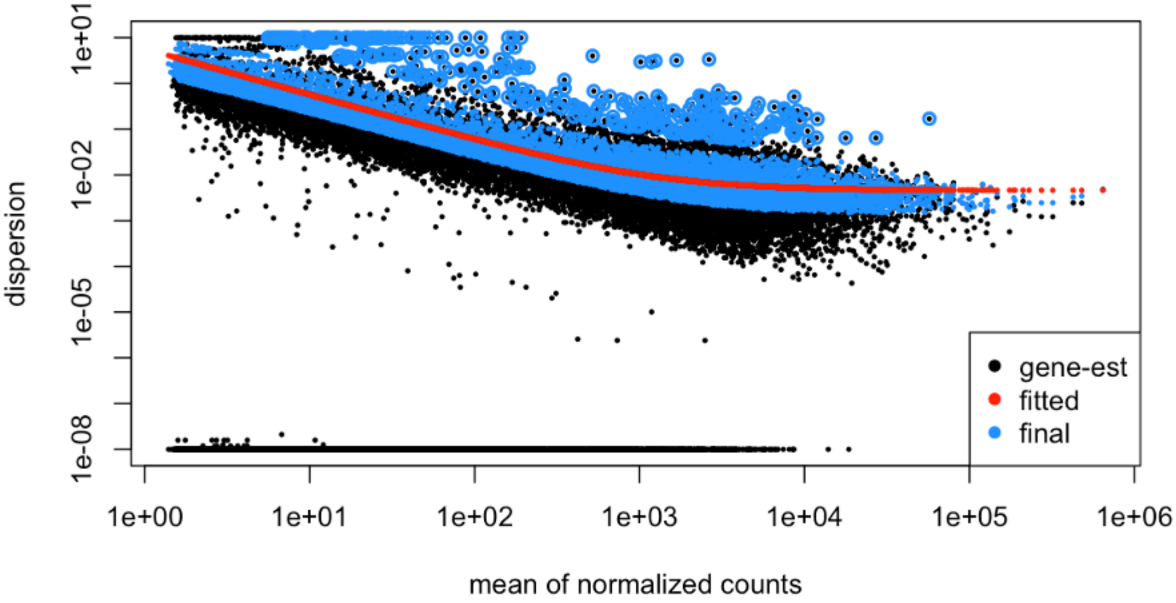


Figure 8. Dispersion-by-mean plot

The graph (Figure 8) showcases how the gene variations were accounted for during analysis. Black dots show the initial estimates of how much each gene's expression varied. Blue dots represent these estimates after adjustment, making them more accurate. The red line depicts the expected variation based on the average expression levels.

Raw P-value histogram

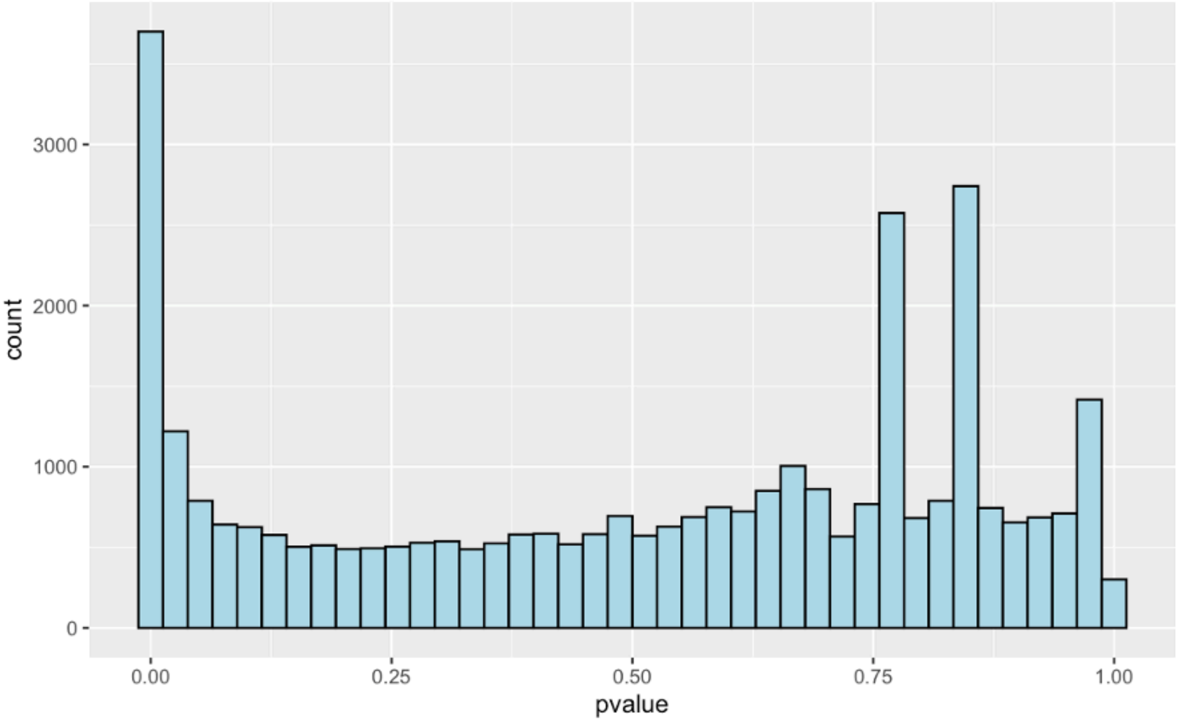


Figure 9. Raw P-value Data

The p-value histogram (Figure 9) is like a fingerprint of the gene expression changes in this experiment. Ideally, if no genes were differentially expressed, the p-values would be spread out evenly (uniform distribution). In this case, the histogram leans to the left, with a bump at very low p-values (close to zero). This means many genes have highly significant changes, suggesting the treatment strongly affected gene expression.There's also a spread of p-values in the middle and higher ranges. While not all genes changed, the histogram is not completely flat, meaning some genes likely have no significant difference. This suggests the treatment impacted some genes more than others. Overall, the slant towards zero and the varied distribution across the p-value range indicate the treatment had a measurable effect on many genes, with varying degrees of change in their expression.

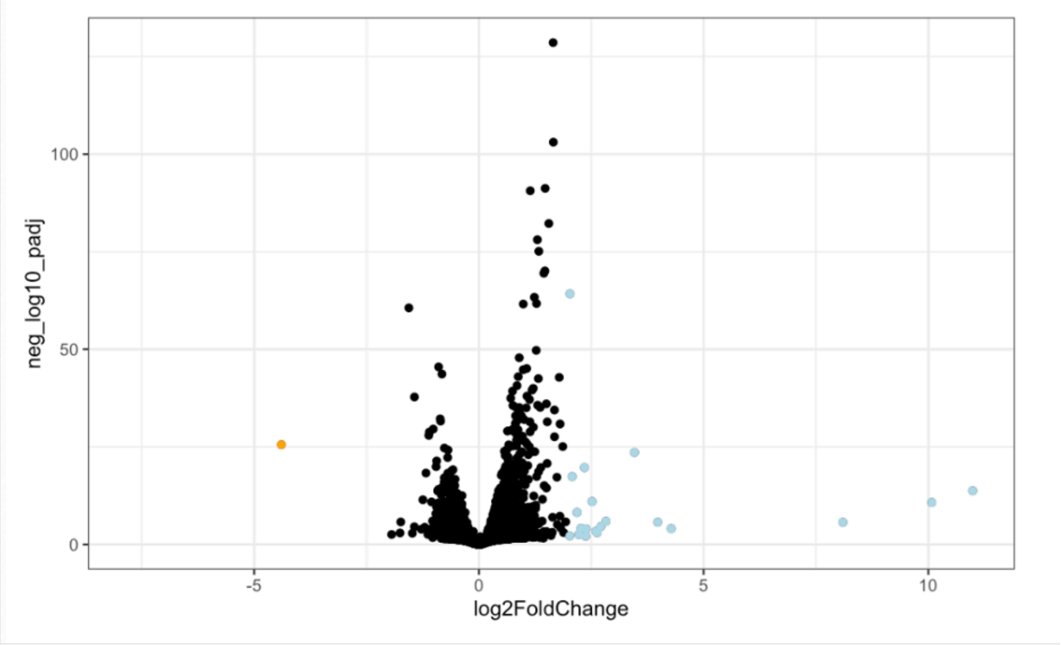


Figure 10. Volcano Scatterplot

Most genes cluster near the center, indicating little to no change in expression between treated and control groups. Genes with very low adjusted p-values (statistically significant) appear higher on the y-axis. The higher they are, the stronger the evidence for a change. Genes located further away from zero on the x-axis show larger changes in expression. Dots to the right represent genes expressed at a higher level in the treated group compared to the control (upregulated). Conversely, dots to the left represent genes with lower expression in the treated group (downregulated). The lone orange dot far left is likely an outlier, a gene with a very substantial decrease in expression due to the treatment.

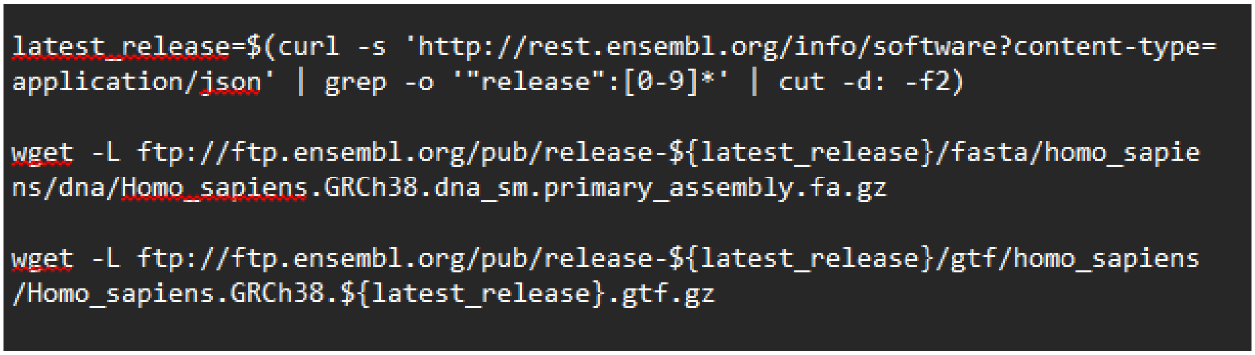
Appendix

Table of Contents

Title:
Executed Scripts/Commands and Generated Reports

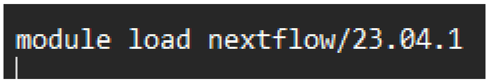
Appendix A: Execution of NextFlow

- Download.sh was downloaded
    - After “cd” in scratch/zmk256/FinalProjectA, I copied download.sh to the directory, and ran “bash download.sh” to download the contents.
- Download the recommended reference transcriptome and annotation file from provided links.



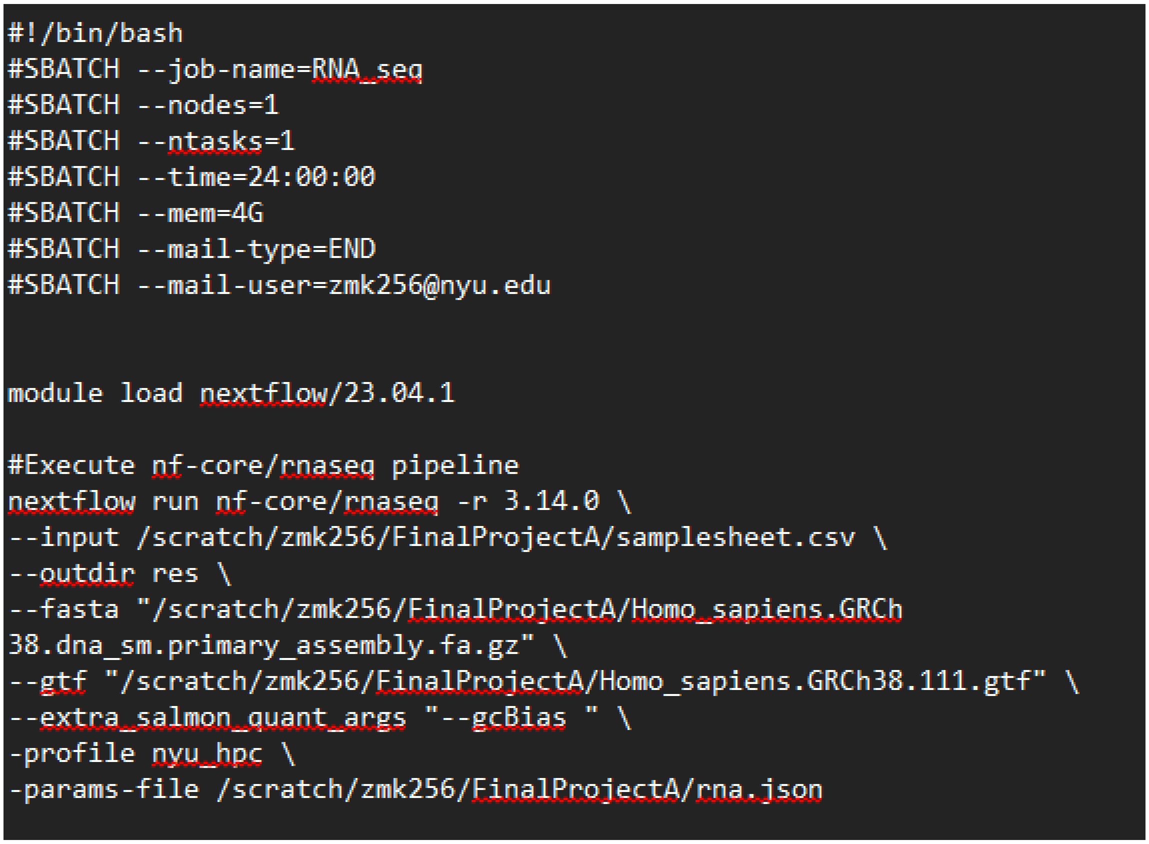
```
latest_release=$(curl -s 'http://rest.ensembl.org/info/software?content-type=
application/json' | grep -o '"release":[0-9]*' | cut -d: -f2)

wget -L ftp://ftp.ensembl.org/pub/release-${latest_release}/fasta/homo_sapie
ns/dna/Homo_sapiens.GRCh38.dna_sm.primary_assembly.fa.gz

wget -L ftp://ftp.ensembl.org/pub/release-${latest_release}/gtf/homo_sapiens
/Homo_sapiens.GRCh38.${latest_release}.gtf.gz
```

- Create a slurm script to execute nextflow
    - Load the most recent Nextflow module:



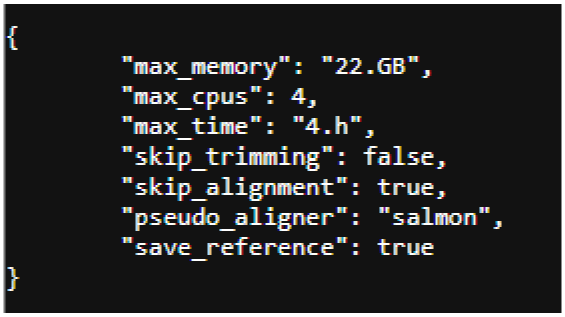
```
module load nextflow/23.04.1
```

    - Enter the execution slurm script by using the nano command.



```
#!/bin/bash
#SBATCH --job-name=RNA_seq
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=24:00:00
#SBATCH --mem=4G
#SBATCH --mail-type=END
#SBATCH --mail-user=zmk256@nyu.edu


module load nextflow/23.04.1

#Execute nf-core/rnaseq pipeline
nextflow run nf-core/rnaseq -r 3.14.0 \
--input /scratch/zmk256/FinalProjectA/samplesheet.csv \
--outdir res \
--fasta "/scratch/zmk256/FinalProjectA/Homo_sapiens.GRCh
38.dna_sm.primary_assembly.fa.gz" \
--gtf "/scratch/zmk256/FinalProjectA/Homo_sapiens.GRCh38.111.gtf" \
--extra_salmon_quant_args "--gcBias " \
-profile nyu_hpc \
-params-file /scratch/zmk256/FinalProjectA/rna.json
```

- Create a configuration file (rna.json)

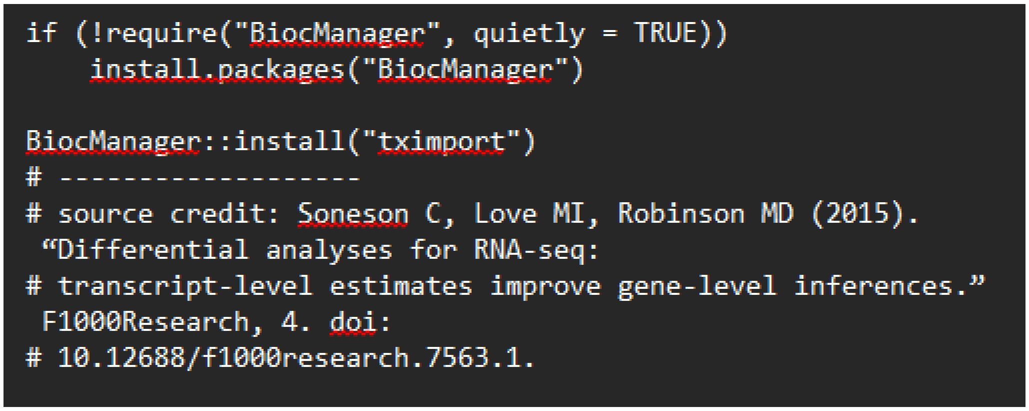


```
{
        "max_memory": "22.GB",
        "max_cpus": 4,
        "max_time": "4.h",
        "skip_trimming": false,
        "skip_alignment": true,
        "pseudo_aligner": "salmon",
        "save_reference": true
}
```

- MultiQC and execution reports were generated after the successful execution of Nextflow workflow, along with the trimmed reads using TrimGalore!/Cutadept (versions provided in Appendix D).

Appendix B: Main tools for RStudio

- Four tools were used to process the generated files from the nextflow execution.
  - Generated files:
    - Quant.sf files for each sample (six in total)
    - Tx2gene.tsv in scratch/zmk256/FinalPrjectA/res/salmon
  - Four necessary tools
    - RStudio
    - DESeq1
    - Tximport

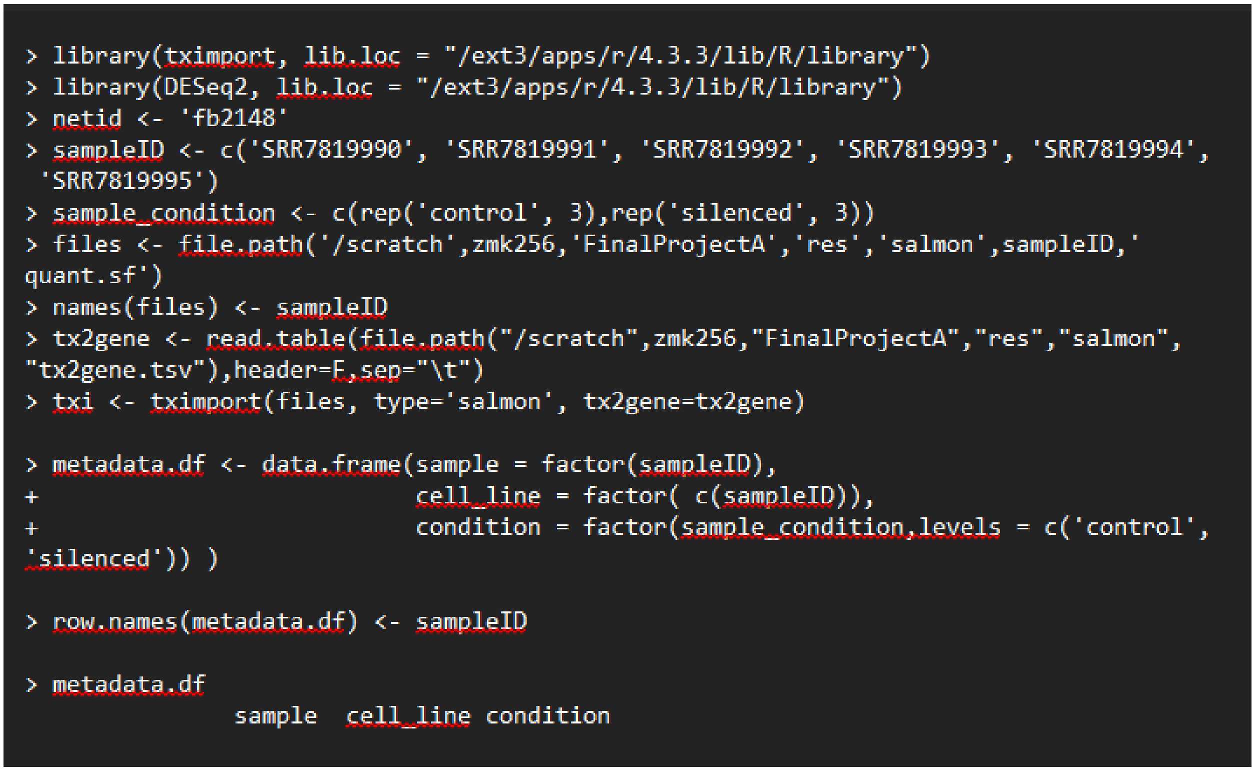


```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("tximport")
# --------------------
# source credit: Soneson C, Love MI, Robinson MD (2015).
 “Differential analyses for RNA-seq:
# transcript-level estimates improve gene-level inferences.”
 F1000Research, 4. doi:
# 10.12688/f1000research.7563.1.
```

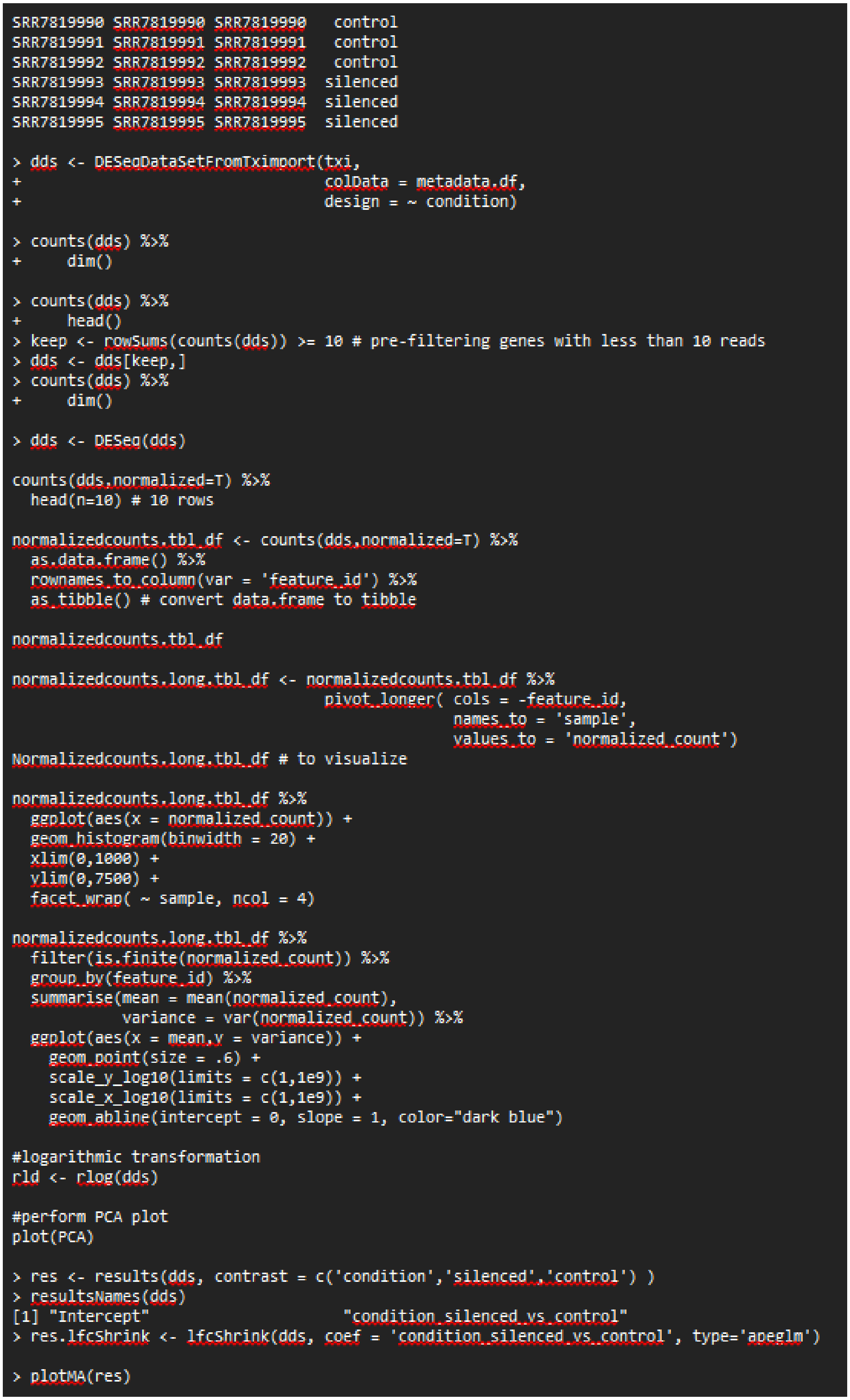
-
  -
    - Tidyverse

Appendix C: R scripts

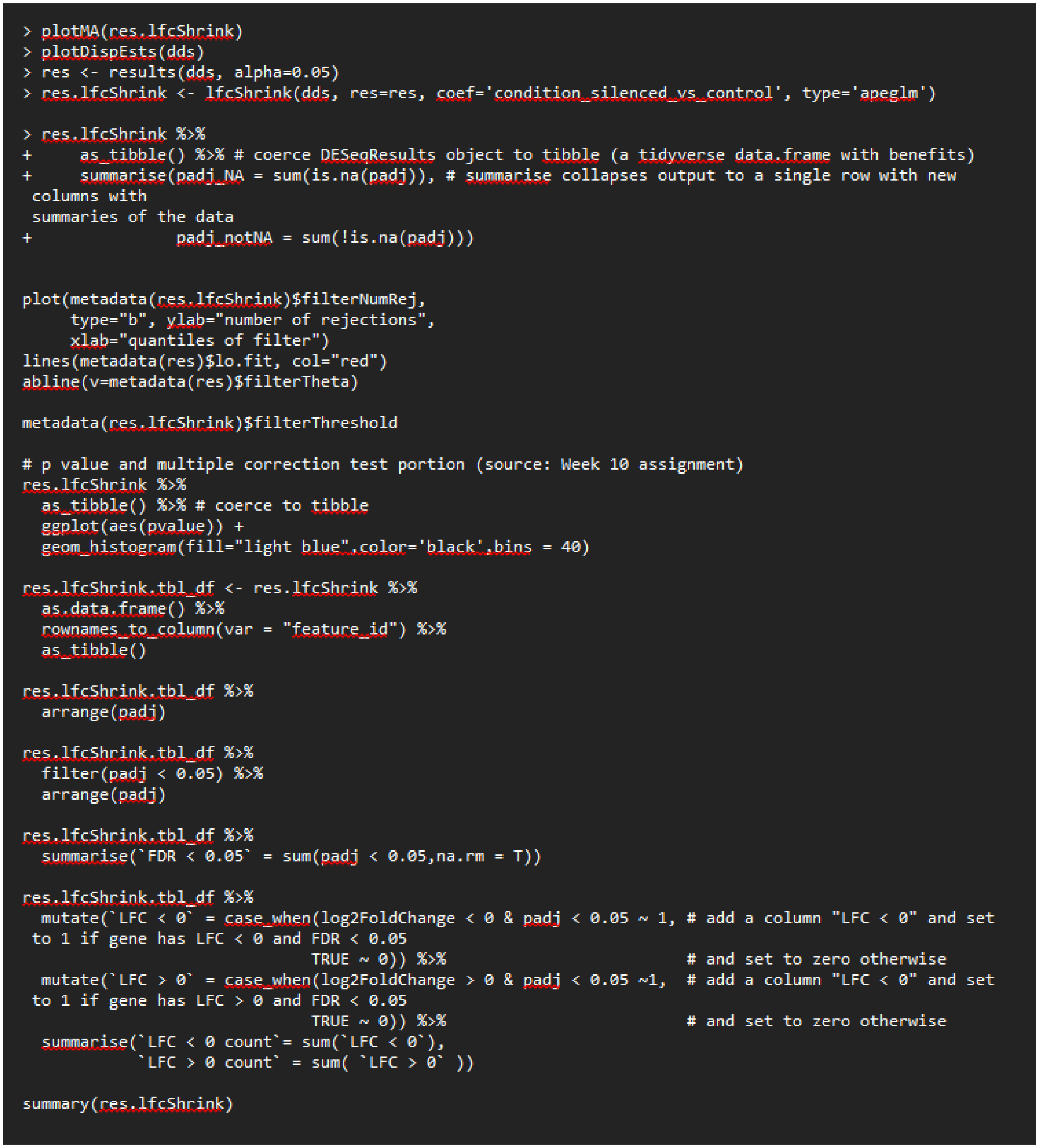
- With the assistance of Week 9 and 10 Assignments in the course, R scripts were executed while substituting with information from the current project.



```
> library(tximport, lib.loc = "/ext3/apps/r/4.3.3/lib/R/library")
> library(DESeq2, lib.loc = "/ext3/apps/r/4.3.3/lib/R/library")
> netid <- 'fb2148'
> sampleID <- c('SRR7819990', 'SRR7819991', 'SRR7819992', 'SRR7819993', 'SRR7819994',
 'SRR7819995')
> sample_condition <- c(rep('control', 3),rep('silenced', 3))
> files <- file.path('/scratch',zmk256,'FinalProjectA','res','salmon',sampleID,'
quant.sf')
> names(files) <- sampleID
> tx2gene <- read.table(file.path("/scratch",zmk256,"FinalProjectA","res","salmon",
"tx2gene.tsv"),header=F,sep="\t")
> txi <- tximport(files, type='salmon', tx2gene=tx2gene)

> metadata.df <- data.frame(sample = factor(sampleID),
+                            cell_line = factor( c(sampleID)),
+                            condition = factor(sample_condition,levels = c('control',
'silenced')) )

> row.names(metadata.df) <- sampleID

> metadata.df
               sample  cell_line condition
```



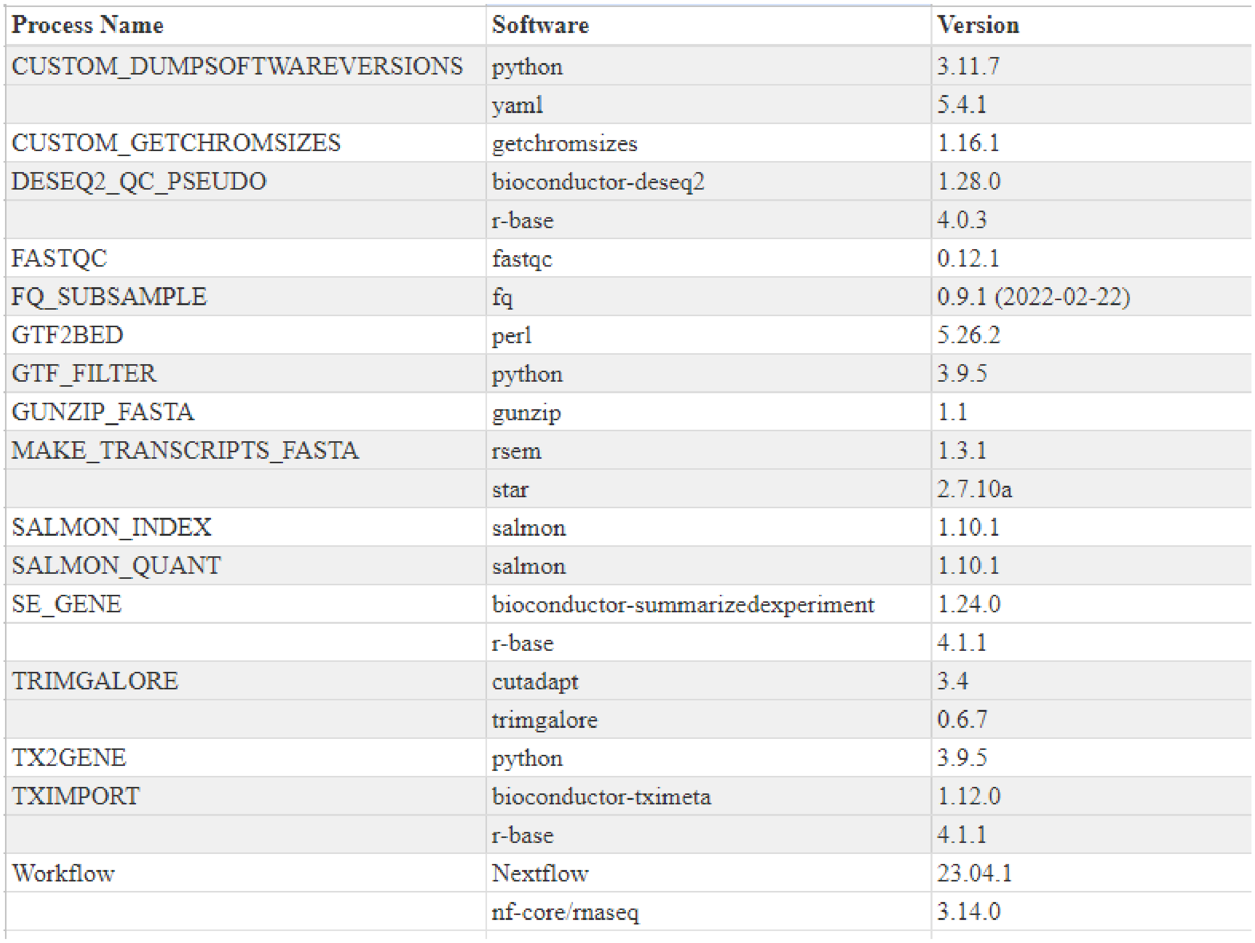
```
SRR7819990 SRR7819990 SRR7819990   control
SRR7819991 SRR7819991 SRR7819991   control
SRR7819992 SRR7819992 SRR7819992   control
SRR7819993 SRR7819993 SRR7819993  silenced
SRR7819994 SRR7819994 SRR7819994  silenced
SRR7819995 SRR7819995 SRR7819995  silenced

> dds <- DESeqDataSetFromTximport(txi,
+                                  colData = metadata.df,
+                                  design = ~ condition)

> counts(dds) %>%
+     dim()

> counts(dds) %>%
+     head()
> keep <- rowSums(counts(dds)) >= 10 # pre-filtering genes with less than 10 reads
> dds <- dds[keep,]
> counts(dds) %>%
+     dim()

> dds <- DESeq(dds)

counts(dds,normalized=T) %>%
  head(n=10) # 10 rows

normalizedcounts.tbl_df <- counts(dds,normalized=T) %>%
  as.data.frame() %>%
  rownames_to_column(var = 'feature_id') %>%
  as_tibble() # convert data.frame to tibble

normalizedcounts.tbl_df

normalizedcounts.long.tbl_df <- normalizedcounts.tbl_df %>%
                                  pivot_longer( cols = -feature_id,
                                                names_to = 'sample',
                                                values_to = 'normalized_count')
Normalizedcounts.long.tbl_df # to visualize

normalizedcounts.long.tbl_df %>%
  ggplot(aes(x = normalized_count)) +
  geom_histogram(binwidth = 20) +
  xlim(0,1000) +
  ylim(0,7500) +
  facet_wrap( ~ sample, ncol = 4)

normalizedcounts.long.tbl_df %>%
  filter(is.finite(normalized_count)) %>%
  group_by(feature_id) %>%
  summarise(mean = mean(normalized_count),
            variance = var(normalized_count)) %>%
  ggplot(aes(x = mean,y = variance)) +
    geom_point(size = .6) +
    scale_y_log10(limits = c(1,1e9)) +
    scale_x_log10(limits = c(1,1e9)) +
    geom_abline(intercept = 0, slope = 1, color="dark blue")

#logarithmic transformation
rld <- rlog(dds)

#perform PCA plot
plot(PCA)

> res <- results(dds, contrast = c('condition','silenced','control') )
> resultsNames(dds)
[1] "Intercept"                     "condition_silenced_vs_control"
> res.lfcShrink <- lfcShrink(dds, coef = 'condition_silenced_vs_control', type='apeglm')

> plotMA(res)
```



```
> plotMA(res.lfcShrink)
> plotDispEsts(dds)
> res <- results(dds, alpha=0.05)
> res.lfcShrink <- lfcShrink(dds, res=res, coef='condition_silenced_vs_control', type='apeglm')

> res.lfcShrink %>%
+     as_tibble() %>% # coerce DESeqResults object to tibble (a tidyverse data.frame with benefits)
+     summarise(padj_NA = sum(is.na(padj)), # summarise collapses output to a single row with new
 columns with
 summaries of the data
+               padj_notNA = sum(!is.na(padj)))


plot(metadata(res.lfcShrink)$filterNumRej,
     type="b", ylab="number of rejections",
     xlab="quantiles of filter")
lines(metadata(res)$lo.fit, col="red")
abline(v=metadata(res)$filterTheta)

metadata(res.lfcShrink)$filterThreshold

# p value and multiple correction test portion (source: Week 10 assignment)
res.lfcShrink %>%
  as_tibble() %>% # coerce to tibble
  ggplot(aes(pvalue)) +
  geom_histogram(fill="light blue",color='black',bins = 40)

res.lfcShrink.tbl_df <- res.lfcShrink %>%
  as.data.frame() %>%
  rownames_to_column(var = "feature_id") %>%
  as_tibble()

res.lfcShrink.tbl_df %>%
  arrange(padj)

res.lfcShrink.tbl_df %>%
  filter(padj < 0.05) %>%
  arrange(padj)

res.lfcShrink.tbl_df %>%
  summarise(`FDR < 0.05` = sum(padj < 0.05,na.rm = T))

res.lfcShrink.tbl_df %>%
  mutate(`LFC < 0` = case_when(log2FoldChange < 0 & padj < 0.05 ~ 1, # add a column "LFC < 0" and set
 to 1 if gene has LFC < 0 and FDR < 0.05
                              TRUE ~ 0)) %>%                        # and set to zero otherwise
  mutate(`LFC > 0` = case_when(log2FoldChange > 0 & padj < 0.05 ~1,  # add a column "LFC < 0" and set
 to 1 if gene has LFC > 0 and FDR < 0.05
                              TRUE ~ 0)) %>%                        # and set to zero otherwise
  summarise(`LFC < 0 count`= sum(`LFC < 0`),
            `LFC > 0 count` = sum( `LFC > 0` ))

summary(res.lfcShrink)
```

Appendix D: Utilized Versions/Packages

- Chart has been retrieved from the MultiQC report.
- All utilized modules/packages and their versions



Process Name	Software	Version
CUSTOM_DUMPSOFTWAREVERSIONS	python	3.11.7
	yaml	5.4.1
CUSTOM_GETCHROMSIZES	getchromsizes	1.16.1
DESEQ2_QC_PSEUDO	bioconductor-deseq2	1.28.0
	r-base	4.0.3
FASTQC	fastqc	0.12.1
FQ_SUBSAMPLE	fq	0.9.1 (2022-02-22)
GTF2BED	perl	5.26.2
GTF_FILTER	python	3.9.5
GUNZIP_FASTA	gunzip	1.1
MAKE_TRANSCRIPTS_FASTA	rsem	1.3.1
	star	2.7.10a
SALMON_INDEX	salmon	1.10.1
SALMON_QUANT	salmon	1.10.1
SE_GENE	bioconductor-summarizedexperiment	1.24.0
	r-base	4.1.1
TRIMGALORE	cutadapt	3.4
	trimgalore	0.6.7
TX2GENE	python	3.9.5
TXIMPORT	bioconductor-tximeta	1.12.0
	r-base	4.1.1
Workflow	Nextflow	23.04.1
	nf-core/rnaseq	3.14.0

Appendix E: External Reports

- MultiQC and Execution reports are attached with the report as individual files.