Assignment 2: Identifying Differentially Expressed Genes between Samples

BINF*6110: Genomic Methods for Bioinformatics

Isha Baxi

INTRODUCTION

Differential gene expression (DGE) analysis plays a crucial role in understanding the molecular mechanisms underlying biological processes, particularly in comparing gene expression levels across different experimental conditions or time points. This project focuses on identifying differentially expressed genes (DEGs) between various developmental stages of *Saccharomyces cerevisiae* (yeast) during velum development. RNA-Seq technology, a powerful tool for transcriptome-wide gene expression profiling, will be employed to generate comprehensive gene expression data, enabling the comparison of gene activity across different stages of velum development (Wang et al., n.d.).

The primary challenge in conducting DGE research is ensuring the accuracy and reliability of the results, as various factors, such as technical variations, sample quality, and experimental biases, can influence gene expression data. To address these challenges, multiple steps are incorporated into the analysis pipeline, including quality control (using FastQC and MultiQC), adapter trimming (using Trimmomatic), alignment (via STAR), and gene quantification (with featureCounts). These steps help to improve the quality and accuracy of the resulting data and reduce potential sources of bias.

However, there are several potential limitations and considerations with this approach. One key challenge is the selection of appropriate parameters for differential expression testing, such as the threshold for fold change (logFC) and false discovery rate (FDR). Stringent thresholds might miss genes that are biologically relevant but only modestly differentially expressed, while lenient thresholds may lead to an increased rate of false positives. Additionally, RNA-Seq data can suffer from batch effects, sequencing biases, and other technical artifacts, which require careful normalization and statistical adjustment to minimize their impact (Wang et al., n.d.).

Despite these challenges, this approach offers the advantage of providing high-resolution data on gene expression, allowing for a detailed analysis of gene regulation during velum development. By leveraging high-quality RNA-Seq data and rigorous statistical methods, this research aims to uncover critical insights into the molecular mechanisms governing yeast development, contributing to a deeper understanding of gene expression dynamics in model organisms.

METHODS AND RESULTS

Software Tools Used

- **FastQC** for quality control checks on raw RNA-Seq data to identify issues like adapter contamination and low-quality reads.
- **Trimmomatic** for trimming low-quality bases and removing adapter sequences to improve read accuracy.
- **STAR** for mapping RNA-Seq reads to the reference genome efficiently.

- **SAMtools** for manipulating alignment files, sorting, indexing, and checking mapping quality.
- **Subread (featureCounts)** for counting reads mapped to genes, enabling gene expression quantification.
- **edgeR** in RStudio for statistical analysis of differential gene expression, including normalization and dispersion estimation.

Workflow of Analysis

1. **Data Preparation** – Retrieve RNA-Seq sequence from Compute Canada and decompress FASTQ files for processing.
2. **Quality Control** – Use FastQC to assess sequencing quality and MultiQC for a summarized quality report for visualizations.
3. **Adaptor Trimming** – Use Trimmomatic to remove adapters and low-quality bases and processed reads were stored for downstream analysis.
4. **Sequence Alignment** – STAR aligns to referenced genome, generates and sorts BAM files using SAMtools.
5. **Gene Quantification** – Use featureCounts to assign reads to genomic features and produce count matrix for differential expression analysis (DEA).
6. **DEA** – Load count matrix for DEA and filter low-expression genes, normalize counts, create design matrix and estimate dispersion. Fit the model and perform various differential expression tests.
7. **Visualizations** – Create visualizations in RStudio to analyze DEA.

The differential gene expression analysis revealed significant changes in transcript levels across the three stages of yeast velum development. The Venn diagram (Figure 3) illustrates the overlap of differentially expressed genes (DEGs) between the early biofilm, thin biofilm, and mature biofilm stages. A total of 806 genes were found to be differentially expressed in all pairwise comparisons, indicating a core set of genes that may play crucial roles in velum development. The volcano plots (Figure 1) further elucidate these findings by highlighting upregulated and downregulated genes for each comparison. For instance, in the early vs thin comparison, several genes showed high log fold changes (logFC) with low p-values, suggesting strong differential expression. Similarly, the early vs later comparison identified key genes with substantial logFC values, such as Gene 3568 with a logFC of 3.28 and a highly significant p-value of 1.37e-12. These results underscore the dynamic nature of gene expression during velum formation.

The smear plot (Figure 2) provides additional insights into the distribution of log fold changes against average log counts per million (CPM). It shows a clear separation between significantly differentially expressed genes and those that are not, validating the robustness of our statistical approach. The top 10 DEGs for each pairwise comparison are summarized in Tables 1 and 2. For example, in the early vs thin comparison, Gene 1 exhibits a logFC of 1.71 and a p-value of 1.71e-12, indicating strong upregulation. In the early vs later comparison, Gene 3568 has a logFC of 3.28 and a p-value of 1.37e-12, suggesting it is highly upregulated in the later stages. These tables provide a concise overview of the most significant changes in gene expression, facilitating further investigation into their biological functions.

DISCUSSION

The differential gene expression analysis conducted in this assignment provides a comprehensive view of the transcriptomic changes occurring during yeast velum development. The identification of 806 genes commonly differentially expressed across all pairwise comparisons highlights a core set of genes that likely play essential roles in the progression of velum formation. These genes may represent key regulators of biological pathways involved in biofilm development, such as cell adhesion, extracellular matrix production, and stress response. For instance, Gene YKL164C exhibits a logFC of 3.28 and an FDR of 5.67e-09, suggesting its potential involvement in the transition from early to mature biofilm stages. Similarly, Gene YDR403W shows a logFC of 4.09 and an FDR of 5.67e-09, indicating its role in the initial stages of velum formation. These findings underscore the dynamic nature of gene expression during velum development and provide a strong foundation for further functional studies.

The robustness of our approach is evident in the clear separation of significant DEGs from non-significant ones in the smear plot and the consistency observed in the volcano plots. The use of edgeR for normalization, dispersion estimation, and statistical modeling ensures that the results are both accurate and reproducible. However, it is important to acknowledge potential limitations in the analysis. While statistically significant DEGs are identified, it does not provide direct evidence of their biological functions. Functional validation through experimental techniques such as qPCR, RNA interference, or CRISPR-based knockouts would be necessary to confirm the roles of these genes (Sokol et al., 2023). Additionally, incorporating pathway enrichment analysis could reveal broader biological themes and interactions among the identified DEGs, offering deeper insights into the molecular mechanisms driving velum development. The inclusion of adjusted p-values (FDR) in the tables helps control for multiple testing and reduces the likelihood of false positives, enhancing the reliability of the results. Furthermore, the high F values observed for many DEGs, such as Gene YKL164C with an F value of 861.0337, indicate strong statistical support for the differential expression patterns, reinforcing the significance of the findings.

Another consideration is the integration of additional genomic information provided in the tables, such as chromosome location, strand orientation, and gene length. This information can offer context regarding the structural and regulatory features of the identified DEGs. For example, Gene YBR117C, located on the negative strand with a length of 2046 base pairs, shows a logFC of 3.92 and an FDR of 1.05e-08, suggesting its potential involvement in specific regulatory networks. Understanding the chromosomal organization and structural attributes of these genes can aid in predicting their functional roles and interactions within the cellular environment.

Finally, the implications of this assignment extend beyond the immediate context of yeast velum development. Understanding the genetic basis of biofilm formation has broader applications in biotechnology, food science, and even medicine. Furthermore, the methods employed in this assignment—ranging from quality control and read alignment to differential expression analysis—serve as a robust framework for similar transcriptomic investigations. By capturing both the magnitude and significance of gene expression changes, this approach effectively identifies key regulators of biological processes. In conclusion, this assignment not only advances our understanding of yeast velum development but also demonstrates the power of RNA-Seq analysis in uncovering the molecular underpinnings of complex biological phenomena.

REFERENCES

Babraham Bioinformatics. (n.d.). FastQC A Quality Control Tool for High Throughput Sequence Data. Retrieved March 11, 2025, from http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., & Gingeras, T. R. (n.d.). STAR: Ultrafast universal RNA-seq aligner - PMC. *Bioinformatics*, *29*(1). https://doi.org/10.1093/bioinformatics/bts635

Genetic analyses of bacterial biofilm formation. (n.d.). *Current Opinion in Microbiology*, *2*(6), 598–603. https://doi.org/10.1016/S1369-5274(99)00028-4

Mardanov, A. V., Eldarov, M. A., Beletsky, A. V., Tanashchuk, T. N., Kishkovskaya, S. A., & Ravin, N. V. (n.d.). Frontiers. *Frontiers in Microbiology*, *11*. https://doi.org/10.3389/fmicb.2020.00538

Reynolds, T. B., & Fink, G. R. (2001). Bakers' Yeast, a model for fungal biofilm formation. *Science*, *291*(5505), 878–881. https://doi.org/10.1126/science.291.5505.878

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2009). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. https://doi.org/10.1093/bioinformatics/btp616

Sokol, L., Cuypers, A., Truong, A.-C. K., Bouché, A., Brepoels, K., Souffreau, J., Rohlenova, K., Vinckier, S., Schoonjans, L., Eelen, G., Dewerchin, M., de Rooij, L. P. M. H., & Carmeliet, P. (2023). Prioritization and functional validation of target genes from single-cell transcriptomics studies. *Communications Biology*, *6*(1), 1–13. https://doi.org/10.1038/s42003-023-05006-7

USADELLAB.org - Trimmomatic: A flexible read trimming tool for Illumina NGS data. (n.d.). USADELLAB.Org. Retrieved March 11, 2025, from http://www.usadellab.org/cms/?page=trimmomatic

Wang, Z., Gerstein, M., & Snyder, M. (n.d.). RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews Genetics*, *10*(1), 57–63. https://doi.org/10.1038/nrg2484

APPENDIX

Table #1: *Top 10 Differentially Expressed Genes in Early vs Later Comparison*
This table lists the top 10 genes from the Early vs Later pairwise comparison, along with their corresponding log fold change (logFC), p-value, and regulation status (upregulated or downregulated).

Gene	logFC	P-Value	Regulation
3568	3.281201	1.3667E-12	Upregulated
1388	4.094397	2.0222E-12	Upregulated
5338	5.563521	5.2514E-12	Upregulated
6339	3.075086	1.1229E-11	Upregulated
346	3.918998	1.1505E-11	Upregulated
3172	3.108573	1.1754E-11	Upregulated
1053	3.295412	1.4920E-11	Upregulated
1529	6.194781	1.5041E-11	Upregulated
3814	3.203784	1.7174E-11	Upregulated
3119	7.539114	3.8284E-11	Upregulated

Table #2: *Top 10 Differentially Expressed Genes in Early vs Thin Comparison*
This table lists the top 10 genes from the Early vs Thin pairwise comparison, along with their corresponding log fold change (logFC), p-value, and regulation status (upregulated or downregulated). Results for comparisons with other stages are provided in the attached text file.

Gene	logFC	P-Value	Regulation
2740	-5.2693	1.7112E-12	Downregulated
2347	-4.9284	2.6655E-12	Downregulated
3568	4.0922	3.4286E-12	Upregulated
3172	4.3416	9.5974E-12	Upregulated
5338	7.9224	1.0492E-11	Upregulated
5339	4.9411	1.4746E-11	Upregulated
1530	5.9411	3.3035E-11	Upregulated
2195	-4.7071	3.3245E-11	Downregulated

3123	-5.3230	4.0096E-11	Downregulated
1388	4.0527	4.8906E-11	Upregulated

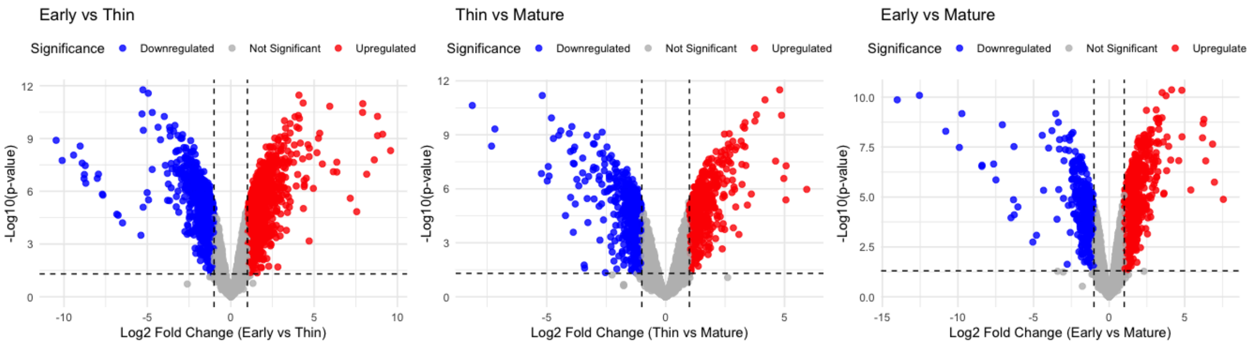

Figure 1: *Volcano Plots of Differentially Expressed Genes Across Biofilm Development Stages*
Volcano plots depicting the differential expression of genes in three pairwise comparisons during biofilm development: Early vs Thin, Thin vs Mature, and Early vs Mature. Each plot shows the log2 fold change (x-axis) against the -log10(p-value) (y-axis). Genes are color-coded based on their significance and regulation status: upregulated (red), downregulated (blue), and not significant (gray). The dashed lines indicate the thresholds for statistical significance (p-value < 0.05) and biological relevance (fold change > ±1).

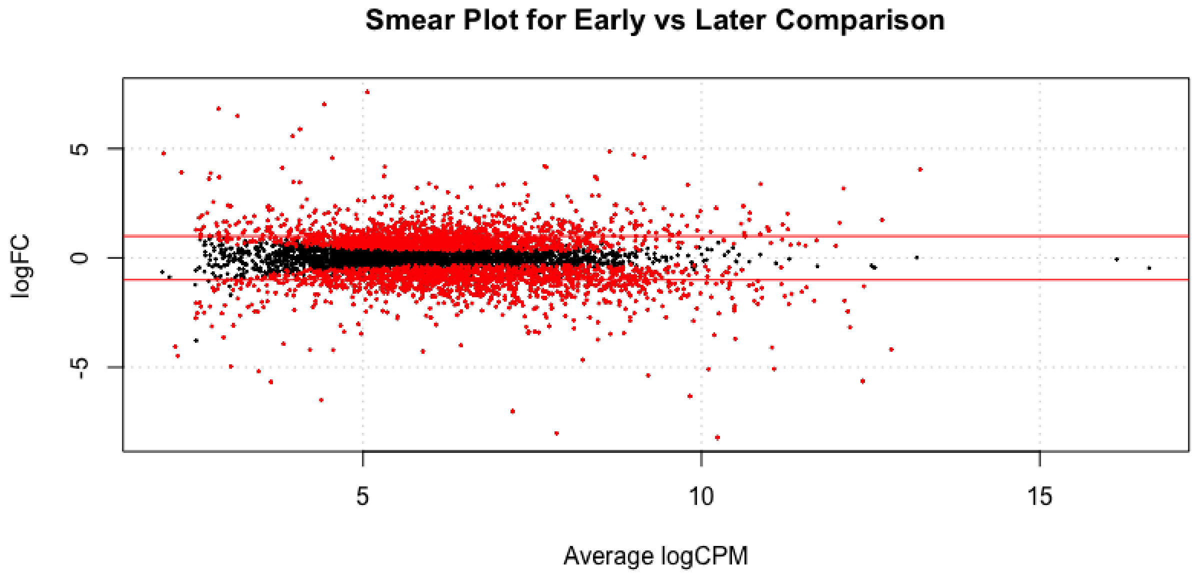

Figure 2: *Smear Plot of Gene Expression in Early vs Later Comparison*
This smear plot illustrates the distribution of log fold changes (logFC) against the average log counts per million (logCPM) for genes in the "Early vs Later" comparison. The red points represent differentially expressed genes, while the black points indicate non-differentially expressed genes. The horizontal red lines denote the thresholds for significant upregulation and downregulation (±1 logFC). This visualization helps identify genes with substantial expression changes between early and later stages of biofilm development.

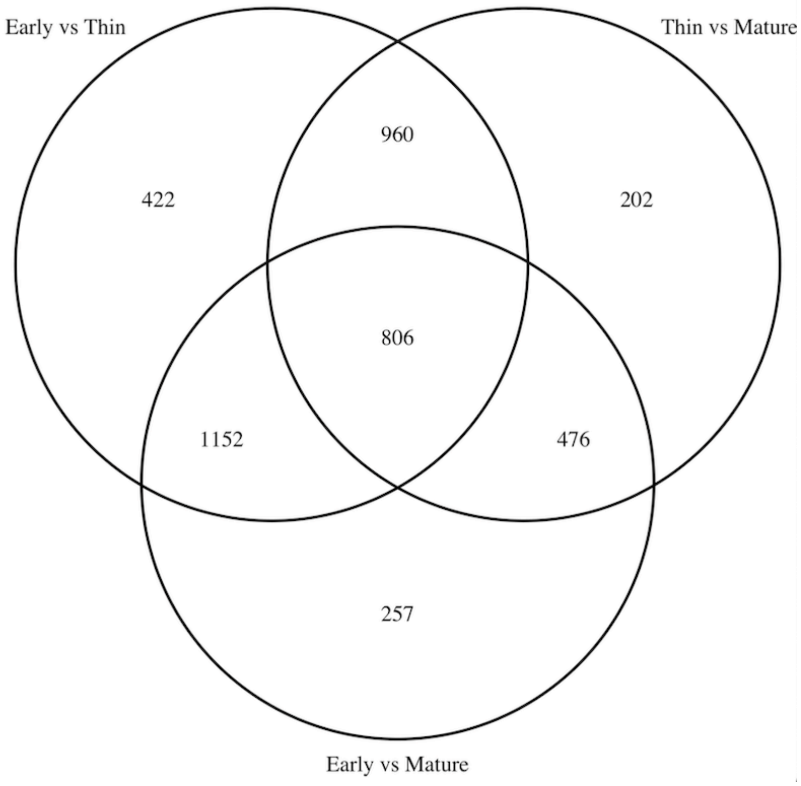

Figure 3: *Venn Diagram of Differentially Expressed Genes Across Biofilm Development Stages*
This Venn diagram illustrates the overlap of differentially expressed genes (DEGs) across three pairwise comparisons during biofilm development: Early vs Thin, Thin vs Mature, and Early vs Mature. The numbers within each section represent the count of DEGs unique to or shared between the respective comparisons. Specifically:

- 422 genes are uniquely differentially expressed in the "Early vs Thin" comparison.
- 202 genes are uniquely differentially expressed in the "Thin vs Mature" comparison.
- 257 genes are uniquely differentially expressed in the "Early vs Mature" comparison.
- 960 genes are shared between the "Early vs Thin" and "Thin vs Mature" comparisons.
- 1152 genes are shared between the "Early vs Thin" and "Early vs Mature" comparisons.
- 476 genes are shared between the "Thin vs Mature" and "Early vs Mature" comparisons.
- 806 genes are commonly differentially expressed across all three comparisons.

Bash Commands

```
# Create a directory for Assignment 2
mkdir a2

# Copy all sequence files from the shared directory to the current working directory
cp /scratch/lukens/Assignment_2_Seqs/* .
```

```
# Unzip all FASTQ files
gunzip *.fastq.gz

# Rename paired-end FASTQ files by removing '_1' suffix
for file in *_1.fastq; do
    mv "$file" "${file/_1.fastq/.fastq}"
done

# Load necessary computational modules
module load StdEnv/2023 gcc/12.2.0 r/4.2.2 star/2.7.10a samtools/1.16
fastqc/0.11.9 subread/2.0.3

# Create a directory for quality control (QC) reports
mkdir -p QC_reports

# Allocate computational resources for 2 hours with 32GB memory and 8 CPUs
salloc --time=02:00:00 --mem=32G --cpus-per-task=8

# Run FastQC on all FASTQ files for quality assessment
for file in *.fastq; do
    echo "Running FastQC on $file..."
    fastqc -t 8 "$file" -o QC_reports
done


# Generate a consolidated QC report using MultiQC
multiqc QC_reports -o QC_reports

# Unzip all FastQC reports for inspection
for file in *_fastqc.zip; do
    echo "Unzipping $file..."
    unzip -o "$file" -d .
done

# Remove zipped FastQC reports to save space
rm -v *_fastqc.zip

# Transfer QC reports to local machine
scp -r ibaxi@graham.computecanada.ca:/home/ibaxi/scratch/genomics/a2/QC_reports
.

scp -r
ibaxi@graham.computecanada.ca:/home/ibaxi/scratch/genomics/a2/trimmed_fastq/QC_
output/multiqc_report.html .

# Organize FASTQ files into separate directories
mkdir unzipped zipped
mv SRR*.fastq unzipped/
```

```
# Load Trimmomatic for adapter and quality trimming
module load trimmomatic

# Run Trimmomatic for each FASTQ file to remove low-quality bases
for file in SRR*.fastq; do
    base=$(basename "$file" .fastq)
    java -jar $EBROOTTRIMMOMATIC/trimmomatic-0.39.jar SE -threads 8 \
        "$file" "${base}_trimmed.fastq" HEADCROP:20
    # HEADCROP:20 removes the first 20 bases from each read
    # Single-end (SE) mode is used
    # Uses 8 threads for faster processing

done

# Check available STAR aligner versions
module spider star/2.7.9a

# Perform read alignment using STAR for each FASTQ file
for i in *.fastq; do
    j=$(basename "$i")  # Extracts the filename from the path
    STAR --runMode alignReads \
         --runThreadN 8 \
         --genomeDir /scratch/lukens/Assignment_2_Genome \
         --readFilesIn "$i" \
         --outFileNamePrefix "$j"
done

# Create directories to organize output files
mkdir -p fastq_files aligned_sam_files logs_final logs_progress logs_out
splice_junctions

# Move files into respective directories
mv SRR*.fastq fastq_files/  # Move FASTQ files
mv SRR*.fastqAligned.out.sam aligned_sam_files/  # Move aligned SAM files
mv SRR*.fastqLog.final.out logs_final/  # Move final log files
mv SRR*.fastqLog.progress.out logs_progress/  # Move progress logs
mv SRR*.fastqLog.out logs_out/  # Move general log files
mv SRR*.fastqSJ.out.tab splice_junctions/  # Move splice junction files

# Convert SAM files to BAM format using samtools
for file in *.sam; do
    samtools view -S -b "$file" > "${file%.sam}.bam"
done

# Sort BAM files for downstream analysis
for file in *.bam; do
    samtools sort "$file" -o "${file%.bam}.sorted.bam"
done
```

```
# Load Subread package for feature counting
module load subread

# Perform gene expression quantification using featureCounts
featureCounts -T 8 -a /scratch/lukens/Assignment_2_Genome/genomic.gtf -o
gene_counts.txt *.sorted.bam

# Transfer the gene count file to the local machine
scp -r
ibaxi@graham.computecanada.ca:/home/ibaxi/scratch/genomics/a2/trimmed_fastq/sam
_files/gene_counts.txt .
```

R Script

```
# Load required libraries
install.packages("ggplot2")
install.packages("VennDiagram")
install.packages("EnhancedVolcano")
BiocManager::install("edgeR")

library(ggplot2)
library(VennDiagram)
library(EnhancedVolcano)
library(edgeR)

library(ggplot2)
library(dplyr)
library(gridExtra)

# Load count data
gene_counts <- read.delim("featureGenecount.txt", comment.char="#",
check.names=FALSE, stringsAsFactors=FALSE)

# Define sample groups
Stage <- factor(rep(c("Early biofilm", "Thin biofilm", "Mature biofilm"),
each=3))
y <- DGEList(counts=gene_counts[,7:ncol(gene_counts)], group=Stage,
genes=gene_counts[,1:6])

# Filter and normalize
keep <- filterByExpr(y)
y <- y[keep,,keep.lib.sizes=FALSE]
y <- calcNormFactors(y, method="TMM")

# Create design matrix
designmatrix <- model.matrix(~0 + Stage)
rownames(designmatrix) <- colnames(y)

# Estimate dispersion
```

```
y <- estimateDisp(y, designmatrix, robust=TRUE)

# Fit the model
fit <- glmQLFit(y, designmatrix)

# Differential Expression Analysis
deg_early_vs_thin <- glmQLFTest(fit, contrast = c(1, -1, 0))
deg_thin_vs_mature <- glmQLFTest(fit, contrast = c(0, 1, -1))
deg_early_vs_mature <- glmQLFTest(fit, contrast = c(1, 0, -1))

# Extract top DEGs
top_early_vs_thin <- topTags(deg_early_vs_thin, n = 10, p.value = 0.05)$table
top_thin_vs_mature <- topTags(deg_thin_vs_mature, n = 10, p.value = 0.05)$table
top_early_vs_mature <- topTags(deg_early_vs_mature, n = 10, p.value =
0.05)$table

# Add regulation status (Upregulated/Downregulated) based on logFC
top_early_vs_thin$Regulation <- ifelse(top_early_vs_thin$logFC > 0,
"Upregulated", "Downregulated")
top_thin_vs_mature$Regulation <- ifelse(top_thin_vs_mature$logFC > 0,
"Upregulated", "Downregulated")
top_early_vs_mature$Regulation <- ifelse(top_early_vs_mature$logFC > 0,
"Upregulated", "Downregulated")

# Combine results
list_anytwotreatments <- data.frame(
  Gene = rownames(top_early_vs_thin),
  Regulation_Early_vs_Thin = top_early_vs_thin$Regulation,
  logFC_Early_vs_Thin = top_early_vs_thin$logFC,
  PValue_Early_vs_Thin = top_early_vs_thin$PValue,
  Regulation_Thin_vs_Mature = top_thin_vs_mature$Regulation,
  logFC_Thin_vs_Mature = top_thin_vs_mature$logFC,
  PValue_Thin_vs_Mature = top_thin_vs_mature$PValue,
  Regulation_Early_vs_Mature = top_early_vs_mature$Regulation,
  logFC_Early_vs_Mature = top_early_vs_mature$logFC,
  PValue_Early_vs_Mature = top_early_vs_mature$PValue
)

# Early vs Later (Thin & Mature)
early_vs_later <- c(1, -0.5, -0.5)
deg_early_vs_later <- glmQLFTest(fit, contrast=early_vs_later)

# Extract DEGs
top_early_vs_later <- topTags(deg_early_vs_later, n=10, p.value=0.05)$table

# Add regulation status (Upregulated/Downregulated) based on logFC
top_early_vs_later$Regulation <- ifelse(top_early_vs_later$logFC > 0,
"Upregulated", "Downregulated")
```

```
# Combine results
list_early_vs_later <- data.frame(
  Gene = rownames(top_early_vs_later),
  Regulation_Early_vs_Later = top_early_vs_later$Regulation,
  logFC_Early_vs_Later = top_early_vs_later$logFC,
  PValue_Early_vs_Later = top_early_vs_later$PValue,
  FDRValue_Early_vs_Later = top_early_vs_later$FDR
)

# Remove unnecessary columns (chr, start, end, strand, length) from the top 10
lists
#list_anytwotreatments <- list_anytwotreatments %>%
  select(-matches("chr|start|end|strand|length"))

list_early_vs_later <- list_early_vs_later %>%
  select(-matches("chr|start|end|strand|length"))

top_early_vs_mature <- top_early_vs_mature %>%
  select(-matches("chr|start|end|strand|length"))

top_early_vs_thin <- top_early_vs_thin %>%
  select(-matches("chr|start|end|strand|length"))

top_thin_vs_mature <- top_thin_vs_mature %>%
  select(-matches("chr|start|end|strand|length"))

# Save results as text files
write.table(list_anytwotreatments, "DEGs_Pairwise.txt", row.names=FALSE,
quote=FALSE, sep="\t")
write.table(list_early_vs_later, "DEGs_Early_vs_Later.txt", row.names=FALSE,
quote=FALSE, sep="\t")

### Visualization 1 ----

# Extract all DEGs for each pairwise comparison
all_early_vs_thin <- topTags(deg_early_vs_thin, n = Inf, p.value = 0.05)$table
all_thin_vs_mature <- topTags(deg_thin_vs_mature, n = Inf, p.value =
0.05)$table
all_early_vs_mature <- topTags(deg_early_vs_mature, n = Inf, p.value =
0.05)$table

# Create a Venn diagram for all DEGs
venn_data_all <- list(
  'Early vs Thin' = rownames(all_early_vs_thin),
  'Thin vs Mature' = rownames(all_thin_vs_mature),
  'Early vs Mature' = rownames(all_early_vs_mature)
)

venn.diagram(venn_data_all,
             filename = 'venn_diagram_all_genes.png',
             category.names = c("Early vs Thin", "Thin vs Mature", "Early vs
Mature"),
             output = TRUE)

### Visualization #2 ----
# Smear Plot
dt_early_vs_later <- decideTestsDGE(deg_early_vs_later)
plotSmear(deg_early_vs_later, de.tags =
rownames(y)[as.logical(dt_early_vs_later)], main = "Smear Plot for Early vs
Later Comparison")
abline(h = c(-1, 1), col = "red")

# Visualization #3 -------

# Combine all pairwise results into a single data frame for volcano plots
volcano_data <- data.frame(
  Gene = rownames(y$genes),
  logFC_Early_vs_Thin = deg_early_vs_thin$table$logFC,
  PValue_Early_vs_Thin = deg_early_vs_thin$table$PValue,
  logFC_Thin_vs_Mature = deg_thin_vs_mature$table$logFC,
  PValue_Thin_vs_Mature = deg_thin_vs_mature$table$PValue,
  logFC_Early_vs_Mature = deg_early_vs_mature$table$logFC,
  PValue_Early_vs_Mature = deg_early_vs_mature$table$PValue
)

# Define thresholds for significance
p_threshold <- 0.05
fc_threshold <- 1

# Add columns to classify genes as upregulated, downregulated, or not
significant for each comparison
volcano_data <- volcano_data %>%
  mutate(
    Significance_Early_vs_Thin = case_when(
      PValue_Early_vs_Thin < p_threshold & abs(logFC_Early_vs_Thin) >
fc_threshold ~ ifelse(logFC_Early_vs_Thin > 0, "Upregulated", "Downregulated"),
      TRUE ~ "Not Significant"
    ),
    Significance_Thin_vs_Mature = case_when(
      PValue_Thin_vs_Mature < p_threshold & abs(logFC_Thin_vs_Mature) >
fc_threshold ~ ifelse(logFC_Thin_vs_Mature > 0, "Upregulated",
"Downregulated"),
      TRUE ~ "Not Significant"
    ),
    Significance_Early_vs_Mature = case_when(
      PValue_Early_vs_Mature < p_threshold & abs(logFC_Early_vs_Mature) >
fc_threshold ~ ifelse(logFC_Early_vs_Mature > 0, "Upregulated",
"Downregulated"),
      TRUE ~ "Not Significant"
    )
  )

# Volcano Plot for Early vs Thin
volcano_plot_Early_vs_Thin <- ggplot(volcano_data, aes(x = logFC_Early_vs_Thin,
y = -log10(PValue_Early_vs_Thin), color = Significance_Early_vs_Thin)) +
  geom_point(alpha = 0.8, size = 2) +
  scale_color_manual(values = c("Upregulated" = "red", "Downregulated" =
"blue", "Not Significant" = "grey")) +
  geom_hline(yintercept = -log10(p_threshold), linetype = "dashed", color =
"black") +
  geom_vline(xintercept = c(fc_threshold, -fc_threshold), linetype = "dashed",
color = "black") +
  labs(
    title = "Early vs Thin",
    x = "Log2 Fold Change (Early vs Thin)",
    y = "-Log10(p-value)",
    color = "Significance"
  ) +
  theme_minimal() +
  theme(legend.position = "top")

# Volcano Plot for Thin vs Mature
volcano_plot_Thin_vs_Mature <- ggplot(volcano_data, aes(x =
logFC_Thin_vs_Mature, y = -log10(PValue_Thin_vs_Mature), color =
Significance_Thin_vs_Mature)) +
  geom_point(alpha = 0.8, size = 2) +
  scale_color_manual(values = c("Upregulated" = "red", "Downregulated" =
"blue", "Not Significant" = "grey")) +
  geom_hline(yintercept = -log10(p_threshold), linetype = "dashed", color =
"black") +
  geom_vline(xintercept = c(fc_threshold, -fc_threshold), linetype = "dashed",
color = "black") +
  labs(
    title = "Thin vs Mature",
    x = "Log2 Fold Change (Thin vs Mature)",
    y = "-Log10(p-value)",
    color = "Significance"
  ) +
  theme_minimal() +
  theme(legend.position = "top")

# Volcano Plot for Early vs Mature
volcano_plot_Early_vs_Mature <- ggplot(volcano_data, aes(x =
logFC_Early_vs_Mature, y = -log10(PValue_Early_vs_Mature), color =
Significance_Early_vs_Mature)) +
  geom_point(alpha = 0.8, size = 2) +
  scale_color_manual(values = c("Upregulated" = "red", "Downregulated" =
"blue", "Not Significant" = "grey")) +
  geom_hline(yintercept = -log10(p_threshold), linetype = "dashed", color =
"black") +
  geom_vline(xintercept = c(fc_threshold, -fc_threshold), linetype = "dashed",
color = "black") +
  labs(
    title = "Early vs Mature",
    x = "Log2 Fold Change (Early vs Mature)",
    y = "-Log10(p-value)",
    color = "Significance"
  ) +
  theme_minimal() +
  theme(legend.position = "top")

# Combine the plots side by side
grid.arrange(volcano_plot_Early_vs_Thin, volcano_plot_Thin_vs_Mature,
volcano_plot_Early_vs_Mature, ncol = 3)
```