TCGA RNA-seq Analysis using DESeq2

Albert Wang

Introduction

Lung adenocarcinoma (LUAD) is the most common subtype of non-small cell lung cancer and a major contributor to cancer-related mortality in the United States (Siegel et al., 2024; Herbst et al., 2018). Understanding changes in gene expression during tumor development and progression is essential for identifying potential biomarkers and therapeutic targets. **RNA sequencing (RNA-seq)** is a widely used technique for measuring gene expression by quantifying the RNA produced from active genes, helping researchers determine which genes are turned on or off under different biological conditions. In this analysis, I used RNA-seq data from **The Cancer Genome Atlas (TCGA)** to investigate transcriptomic differences between tumor and normal tissues, as well as between early-stage and late-stage LUAD. TCGA is a comprehensive, large-scale public database initiated by the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) to systematically document tumor molecular profiles (Chang et al., 2013). By integrating genomic, epigenomic, transcriptomic, and clinical data across more than 30 cancer types, TCGA has become an invaluable platform for studying cancer biology and discovering clinically relevant molecular features. Identifying transcriptomic changes in LUAD may improve our understanding of tumor progression and support the development of stage-specific diagnostic and therapeutic strategies. This notebook outlines the full analysis pipeline—from data acquisition and filtering to differential expression analysis, visualization, and functional enrichment using Gene Ontology (GO) analysis.

Approach

In this analysis, I leveraged RNA-seq data from TCGA's LUAD cohort (Collisson et al., 2014) to explore gene expression differences between tumor and normal tissues, as well as between early-stage and late-stage tumors. After downloading and preparing the dataset, I performed quality control and filtering to remove low-quality and duplicate samples. Differential gene expression analysis was carried out using DESeq2 (Love et al., 2014), a statistical method designed to identify genes that show significant changes in expression between conditions. This was followed by log2 fold change shrinkage to reduce noise in low-count genes and provide more stable estimates. Variance stabilizing transformation (VST) was then applied to the count data, which normalizes for sequencing depth and reduces variance across genes, making the data suitable for exploratory analyses. This transformation prepared the result for downstream visualizations, including PCA plots (to assess sample clustering), volcano plots (to highlight significant gene expression changes), and heatmaps (to visualize expression patterns across samples). Lastly, Gene Ontology (GO) enrichment analysis was conducted to identify biological processes that are statistically overrepresented among the differential expressed genes, providing functional insights into the observed expression changes.

Key Findings

- Lung adenocarcinoma and normal lung tissue exhibit distinct gene expression profiles.
- Genes associated with key cancer hallmarks (Hanahan, 2022), including cell proliferation, extracellular matrix remodeling, and immune evasion, are significantly altered in tumor samples compared to normal tissue.
- In addition to these classical hallmarks, late-stage tumors also show differential expression of genes that may help metastatic cancer cells adapt to and colonize secondary sites.

Transcriptomic Profiling of Tumor and Normal Samples

Download TCGA-LUAD RNA-seq dataset

The Cancer Genome Atlas (TCGA) Lung Adenocarcinoma (LUAD) RNA-seq data was accessed and downloaded using the TCGAbiolinks package (Colaprico et al., 2016). The analysis utilized the "STAR - Counts" workflow, which provides gene level expression (raw counts) data generated using the STAR aligner (Dobin et al., 2012). The data was retrieved directly from the Genomic Data Commons (GDC) and includes both gene expression counts and associated clinical metadata.

```
## Download and clean TCGA data (Lung Adenocarcinoma) ==========================

# Query and download TCGA-LUAD RNA-seq gene expression data
query.luad <- GDCquery(project = "TCGA-LUAD",
                       data.category = "Transcriptome Profiling",
                       data.type = "Gene Expression Quantification",
                       experimental.strategy = "RNA-Seq",
                       workflow.type = "STAR - Counts")

GDCdownload(query.luad)
data.luad <- GDCprepare(query.luad)
```

Data Cleaning and Sample Filtering

To ensure data quality and consistency, I performed filtering to remove samples derived from formalin-fixed paraffin-embedded (FFPE) tissues, which can introduce RNA degradation artifacts. Additionally, for patients with multiple technical replicates, I retained only a single representative sample based on Broad Institute Genome Data Analysis Center (GDAC) barcode—prioritizing analyte type (H > R > T), followed by portion and plate numbers when necessary. This filtering logic was implemented using a custom-written R function to identify and retain the most appropriate replicate per patient. This step minimizes redundancy and ensures that each patient is represented only once in downstream analyses.

```
# Number of samples prior to filtering
length(data.luad$barcode)
```

```
## [1] 600
```

```
### filtering of FFPE (due to potential RNA degradation in old FFPE samples) and replicates

data.luad = data.luad[, !data.luad$is_ffpe] #remove FFPE samples

barcode.all = data.luad$barcode # all barcode ID

# Use the tcga_replicateFilter function defined earlier
# to get the one barcode for each patient (no replicate)
barcode.filt = tcga_replicateFilter(barcode.all)
```

```
## Filter barcodes successfully based on plate number
```

```
data.luad = data.luad[, data.luad$barcode %in% barcode.filt] # obtain filtered data

# Number of samples after filtering out low-quality or duplicate entries
length(data.luad$barcode)
```

```
## [1] 577
```

Samples were further filtered to exclude those lacking appropriate clinical information.

```
# Construct clinical data table and filter out tumor/normal samples without clinical info

# Extract clinical info
clinical.info = data.frame(
  ID = data.luad$barcode,

  # Used to indicate whether the patients were dead or alive at the end of study
  vital_status = data.luad$vital_status,

  # Used only for patients that were dead
  days_to_death = data.luad$days_to_death,

  # Used only for patients that were alive at the end of study
  days_to_last_follow_up = data.luad$days_to_last_follow_up,

  patientID = data.luad$patient,

  FFPE = data.luad$is_ffpe,

  stage = data.luad$ajcc_pathologic_stage,

  tissue_type = data.luad$tissue_type,

  tumorType = data.luad$definition
  # ex. Primary solid Tumor, Recurrent Solid Tumor, etc.
)

# Create a unified "days" column by prioritizing "days_to_death"
# and using "days_to_last_follow_up" if "days_to_death" is missing
clinical.info = clinical.info %>%
  mutate(days = coalesce(days_to_death, days_to_last_follow_up))
# Merge sub-stages (e.g., IA, IB, IIA, IIB, IIIA, IIIB)
# into their corresponding main stages (I, II, III, IV)
clinical.info = clinical.info %>%
  mutate(overall.Stage = case_when(stage == "Stage I" ~ "Stage I",
                                   stage == "Stage IA" ~ "Stage I",
                                   stage == "Stage IB" ~ "Stage I",
                                   stage == "Stage II" ~ "Stage II",
                                   stage == "Stage IIA" ~ "Stage II",
                                   stage == "Stage IIB" ~ "Stage II",
                                   stage == "Stage IIIA" ~ "Stage III",
                                   stage == "Stage IIIB" ~ "Stage III",
                                   stage == "Stage IV" ~ "Stage IV"))

clinical.info.subset = clinical.info %>%
  #filter(overall.Stage != "Stage IV") %>% # (optional) remove certain stages
  #filter(tissue_type == "Tumor") %>%      # (optional) remove Normal or Tumor samples
  filter(!is.na(days)) %>%                 # (optional) remove samples without survival data
  #filter(days != 0) %>%                   # (optional) remove samples with survival day = 0
  filter(!is.na(overall.Stage)) %>%        # remove samples without tumor stage
```

```
  filter(tumorType !=
           "Recurrent Solid Tumor")        # remove recurrent tumor

data.luad = data.luad[, data.luad$barcode %in% clinical.info.subset$ID]


# Number of total samples after filtering based on clinical info
length(data.luad$barcode)
```

```
## [1] 432
```

```
# Number of tumor samples
length(data.luad[, data.luad$tissue_type == "Tumor"]$barcode)
```

```
## [1] 388
```

```
# Number of normal samples
length(data.luad[, data.luad$tissue_type == "Normal"]$barcode)
```

```
## [1] 44
```

DESeq2 Analysis: Tumor vs. Normal tissue

To identify differential expressed genes between tumor and normal tissues, I used the DESeq2 package. DESeq2 is a widely used R package designed for analyzing count-based RNA sequencing data (Love et al., 2014). It models gene expression counts using a negative binomial distribution, allowing for accurate identification of differentially expressed genes between conditions. The method accounts for variation in sequencing depth and biological variability by estimating size factors and dispersion parameters.

To begin, I obtained the TCGA raw count data, mapped Ensembl IDs to gene symbols, and ensured that all samples in the count matrix were present in the clinical metadata with matching sample order.

```
### Use Unstranded data as input (count data) for DESeq2 ###
# Filter samples based on spearman correlation
# (low correlation as possible outlier)
dataPrep.luad <- TCGAanalyze_Preprocessing(object = data.luad,cor.cut = 0.6,
                                           datatype = "unstranded")
```

```
## Number of outliers: 0
```

```
# Get the official gene name
extName = data.frame(Symbol = rowData(data.luad)$gene_name,
                     Ensembl = rownames(data.luad))

# Reorder extName to match rownames of dataPrep.luad
extName = extName[match(rownames(dataPrep.luad), extName$Ensembl),]

# If some Ensembl IDs don't have matching symbols, this will result in NAs in that row
if (anyNA(extName)){
  warning("Some Ensembl IDs don't have matching symbols")
```

```
  extName = extName[!is.na(extName$Symbol),] # remove those rows
}
rownames(dataPrep.luad) = extName$Symbol # change Ensembl ID to gene name
dataPrep.luad = as.data.frame(dataPrep.luad)

# DESeq2 input data format (count data)
dataPrep.luad[1:7, 1:10]
```

```
##          TCGA-05-4244-01A-01R-1107-07 TCGA-05-4249-01A-01R-1107-07
## TSPAN6                           5001                         4383
## TNMD                                0                            0
## DPM1                             1452                         2006
## SCYL3                            1308                         1632
## C1orf112                          789                          482
## FGR                              1963                         1209
## CFH                              2969                         2742
##          TCGA-05-4250-01A-01R-1107-07 TCGA-05-4389-01A-01R-1206-07
## TSPAN6                           5316                         9134
## TNMD                                5                            1
## DPM1                             2886                         2215
## SCYL3                             631                         1106
## C1orf112                          716                          770
## FGR                              1468                         1592
## CFH                              4517                         2822
##          TCGA-05-4390-01A-02R-1755-07 TCGA-05-4395-01A-01R-1206-07
## TSPAN6                           2311                         3131
## TNMD                                0                            0
## DPM1                             1331                         3193
## SCYL3                             385                         1326
## C1orf112                          499                          483
## FGR                               540                          580
## CFH                               921                        17271
##          TCGA-05-4396-01A-21R-1858-07 TCGA-05-4397-01A-01R-1206-07
## TSPAN6                           1107                         7805
## TNMD                                0                           19
## DPM1                              776                         9358
## SCYL3                             495                         1789
## C1orf112                          115                         2743
## FGR                               135                         1775
## CFH                              4044                         8455
##          TCGA-05-4398-01A-01R-1206-07 TCGA-05-4402-01A-01R-1206-07
## TSPAN6                           6052                        12949
## TNMD                                3                            6
## DPM1                             3726                         2197
## SCYL3                            1336                         1530
## C1orf112                         1243                          701
## FGR                              3469                         1531
## CFH                              3208                        22644
```

```
rownames(clinical.info.subset) = clinical.info.subset$ID

# Verify that all samples in the clinical information
# are present in the count data and that naming is consistent
```

```r
all(rownames(clinical.info.subset) %in% colnames(dataPrep.luad))
```

```
## [1] TRUE
```

```r
# Reorder the count matrix to match the sample order in the clinical information
dataPrep.luad <- dataPrep.luad[, rownames(clinical.info.subset)]
# Confirm that the sample order is identical;
# this is critical to prevent misalignment between counts and sample information in DESeq2
all(rownames(clinical.info.subset) == colnames(dataPrep.luad))
```

```
## [1] TRUE
```

General steps of DESeq2 analysis:

- Prior to analysis, a filtering step was applied to remove genes with low total counts (fewer than 10 reads across all samples), which improves both computational efficiency and the reliability of results. I also set the tissue type to "Normal" as the reference level, allowing the model to compute log fold changes for tumor samples relative to normal tissue.
- To address the problem that lowly expressed genes tend to have high variability, shrinkage of log2FC was performed using the apeglm package (Zhu et al., 2018).
- To prepare the data for downstream visualization and clustering, variance stabilizing transformation (VST) was applied to the normalized count matrix. VST adjusts for differences in sequencing depth and stabilizes the variance across genes, especially those with low counts. This transformation makes the data more suitable for methods like principal component analysis (PCA) and heatmaps by reducing the influence of highly variable genes and making expression values more comparable across samples.

```r
### Set up DESeq data set ###
dds <- DESeqDataSetFromMatrix(countData = dataPrep.luad,
                              colData = clinical.info.subset,
                              # Compare tissue_type (Tumor vs. Normal)
                              design = ~ tissue_type)
dds
```

```
## class: DESeqDataSet
## dim: 60660 432
## metadata(1): version
## assays(1): counts
## rownames(60660): TSPAN6 TNMD ... AL391628.1 AP006621.6
## rowData names(0):
## colnames(432): TCGA-05-4396-01A-21R-1858-07
##   TCGA-38-4629-01A-02R-1206-07 ... TCGA-44-7667-01A-31R-2066-07
##   TCGA-MP-A4T8-01A-11R-A24X-07
## colData names(11): ID vital_status ... days overall.Stage
```

```r
### Pre-filtering ###
# It's not necessary to pre-filter low count genes.
# Two useful reasons: 1) reduce memory size of the dds data object,
#                        thus increase speed within DESeq2
#                     2) improve visualizations as low counts gene are not plotted
# filter by sum of counts
keep <- rowSums(counts(dds)) >= 10
```

```r
dds <- dds[keep,]


### factor levels ###
#specify reference level (ex. "untreated")
dds$tissue_type <- relevel(dds$tissue_type, ref = "Normal")

### Differential expression analysis ###
#Starting in version 1.16, betaPrior is F by default,
# this means that DESeq2 will not perform any shrinkage of log2FC.
# This is moved to the lfcShrink function
# so that newer shrinkage methods can be more easily apply as needed.
# Turn betaPrior on if want to match the result of earlier versions of DESEq2.

# The purpose of shrinkage of log2FC is to address the problem
# that lowly expressed genes tend to have high variability

# It looks at the largest fold changes that are not due to low counts
# and uses these to inform a prior distribution.
# So the large fold changes from genes with lots of statistical information are not shrunk,
# while the imprecise fold changes are shrunk.
# This allows you to compare all estimated LFC across experiments,
# for example, which is not really feasible without the use of a prior.

dds <- DESeq(dds, betaPrior = F, quiet = T)
res <- results(dds)
res
```

```
## log2 fold change (MLE): tissue type Tumor vs Normal
## Wald test p-value: tissue type Tumor vs Normal
## DataFrame with 52854 rows and 6 columns
##              baseMean log2FoldChange     lfcSE       stat      pvalue
##             <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## TSPAN6     3184.65530       1.155908 0.1232271    9.38031 6.57804e-21
## TNMD          6.48473       1.535432 0.5284641    2.90546 3.66713e-03
## DPM1       1578.88932       0.285052 0.0891316    3.19811 1.38333e-03
## SCYL3       763.31161       0.480260 0.0764081    6.28546 3.26877e-10
## C1orf112    369.31510       1.664402 0.1138296   14.62187 2.03725e-48
## ...               ...            ...       ...        ...         ...
## AC078856.1    0.200534     -0.0228658  0.762941 -0.0299705 9.76091e-01
## AC008763.4    0.150537     -0.3252293  0.831425 -0.3911707 6.95671e-01
## AL592295.6  257.422865      0.3362808  0.106073  3.1702710 1.52297e-03
## AL391628.1    7.165437     -0.2021930  0.154686 -1.3071172 1.91173e-01
## AP006621.6   16.337285      0.9744893  0.148125  6.5788358 4.74146e-11
##                   padj
##              <numeric>
## TSPAN6     1.00916e-19
## TNMD       7.86581e-03
## DPM1       3.21324e-03
## SCYL3      1.85136e-09
## C1orf112   1.95921e-46
## ...                ...
## AC078856.1          NA
```

```
## AC008763.4          NA
## AL592295.6 3.51270e-03
## AL391628.1 2.67517e-01
## AP006621.6 2.93947e-10
```

```
## LFC (log fold change) shrinkage
# Unlike betaprior, all estimators in lfcShrink does not change padj.
# BetaPrior will give the p-value for the shrunken LFC,
# while lfcShrink is only giving the shruken LFC, and keeping the original p-value
resLFC = lfcShrink(dds, coef = 2, type="apeglm", res = res, quiet = T)
# The newer shrinkage methods/estimators (apeglm and ashr) outperform
# the Normal prior (or betaprior) in most cases. The default is "apeglm".

resLFC
```

```
## log2 fold change (MAP): tissue type Tumor vs Normal
## Wald test p-value: tissue type Tumor vs Normal
## DataFrame with 52854 rows and 5 columns
##              baseMean log2FoldChange     lfcSE      pvalue        padj
##             <numeric>      <numeric> <numeric>   <numeric>   <numeric>
## TSPAN6     3184.65530       1.143226 0.1236426 6.57804e-21 1.00916e-19
## TNMD          6.48473       2.706448 0.5489898 3.66713e-03 7.86581e-03
## DPM1       1578.88932       0.282168 0.0887606 1.38333e-03 3.21324e-03
## SCYL3       763.31161       0.476760 0.0762754 3.26877e-10 1.85136e-09
## C1orf112    369.31510       1.654385 0.1142361 2.03725e-48 1.95921e-46
## ...               ...            ...       ...         ...         ...
## AC078856.1   0.200534       0.357160  0.690028 9.76091e-01          NA
## AC008763.4   0.150537      -0.173560  0.637010 6.95671e-01          NA
## AL592295.6 257.422865       0.332023  0.105489 1.52297e-03 3.51270e-03
## AL391628.1   7.165437      -0.195096  0.152463 1.91173e-01 2.67517e-01
## AP006621.6  16.337285       0.955199  0.148722 4.74146e-11 2.93947e-10
```

```
## Export full DESeq2 result
res.DE = as.data.frame(resLFC)

# DESeq2 sets the gene name with "." instead of "-" (ex. HLA.E instead of HLA-E).
# Here change gene names with "." to "-" (i.e. HLA.E becomes HLA-E)
rownames(res.DE) <- gsub("\\.", "-", rownames(res.DE))

DEgroups_export = resultsNames(dds)[2]
write.csv(as.data.frame(res.DE),
          file=paste("TCGA-LUAD_DESeq2result_", DEgroups_export,".csv", sep = ""))
```

```
### Differential expressed gene (DEG) threshold ###
FC = 2 # log2 fold change
adjp = 0.01 # adjusted p-values

# Determine significant DEGs based on fold-change and adjusted p-value cut-off
sigGenes <- rownames(subset(res.DE, (abs(log2FoldChange)>=FC & padj<=adjp )))

# Extract RNAseq data for the significant genes
sig.res.DE = subset(res.DE, rownames(res.DE) %in% sigGenes)
```

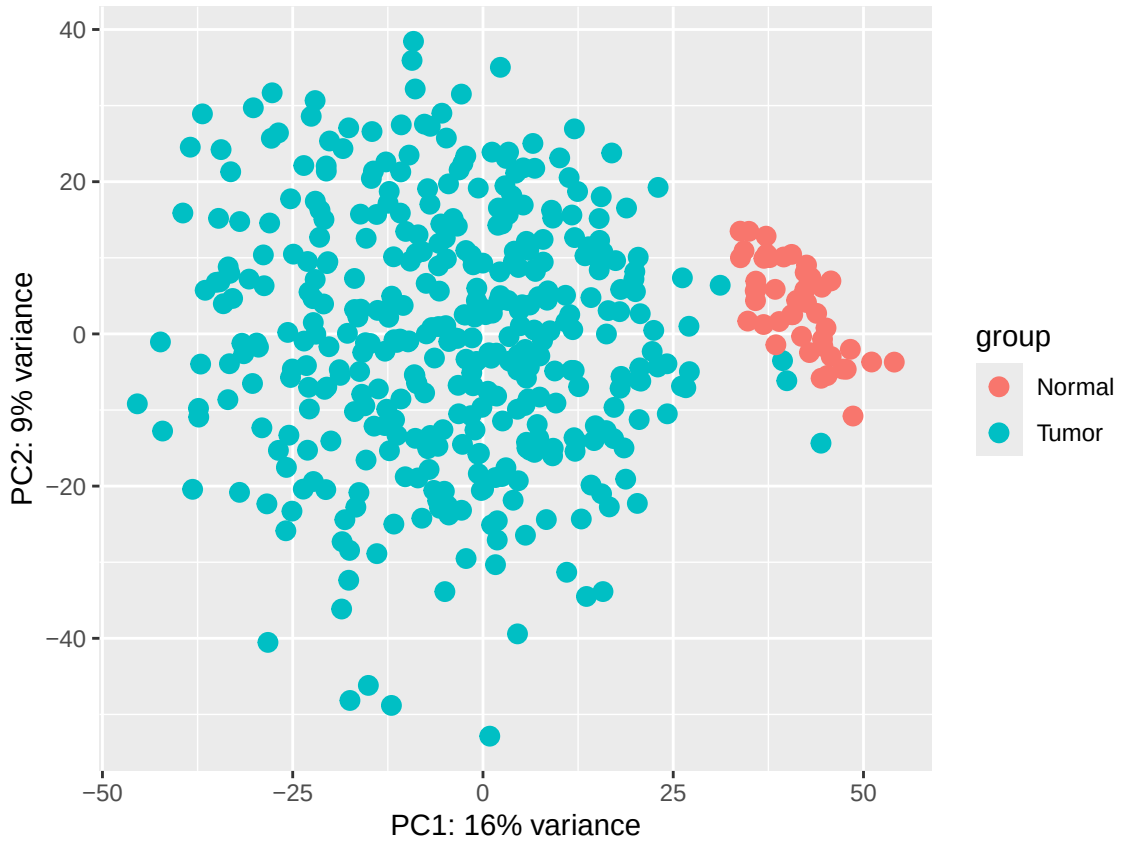
```r
### Variance Stablizing Transformation (VST) ###
# Adjusts for sequencing depth and stabilizes variance across the range of mean values.
# This transformation is important for visualization and clustering,
# as it reduces the influence of highly variable genes
# and makes expression values more comparable across samples.
vsd = vst(dds, blind = F)
# This transform counts into log2 scale (modeled to stabilize variance) for visualization

# Extract the matrix of VST-transformed values
vst_mat = assay(vsd)
sig.vst_mat = subset(vst_mat, rownames(vst_mat) %in% sigGenes)
```

Visualization (Tumor vs. Normal tissue)

To explore overall patterns of gene expression, a Principal Component Analysis (PCA) plot was generated using the VST-transformed data. The PCA plot provides a visual summary of variation across samples based on the gene expression profiles (most variable genes). By reducing high-dimensional RNA-seq data into principal components, the PCA plot highlights sample clustering and separation between groups (ex. tumor and normal tissues).

```r
### Visualization ###

### PCA plot ###
plotPCA(vsd, intgroup = "tissue_type", ntop = 500)
```

```
## using ntop=500 top features by variance
```

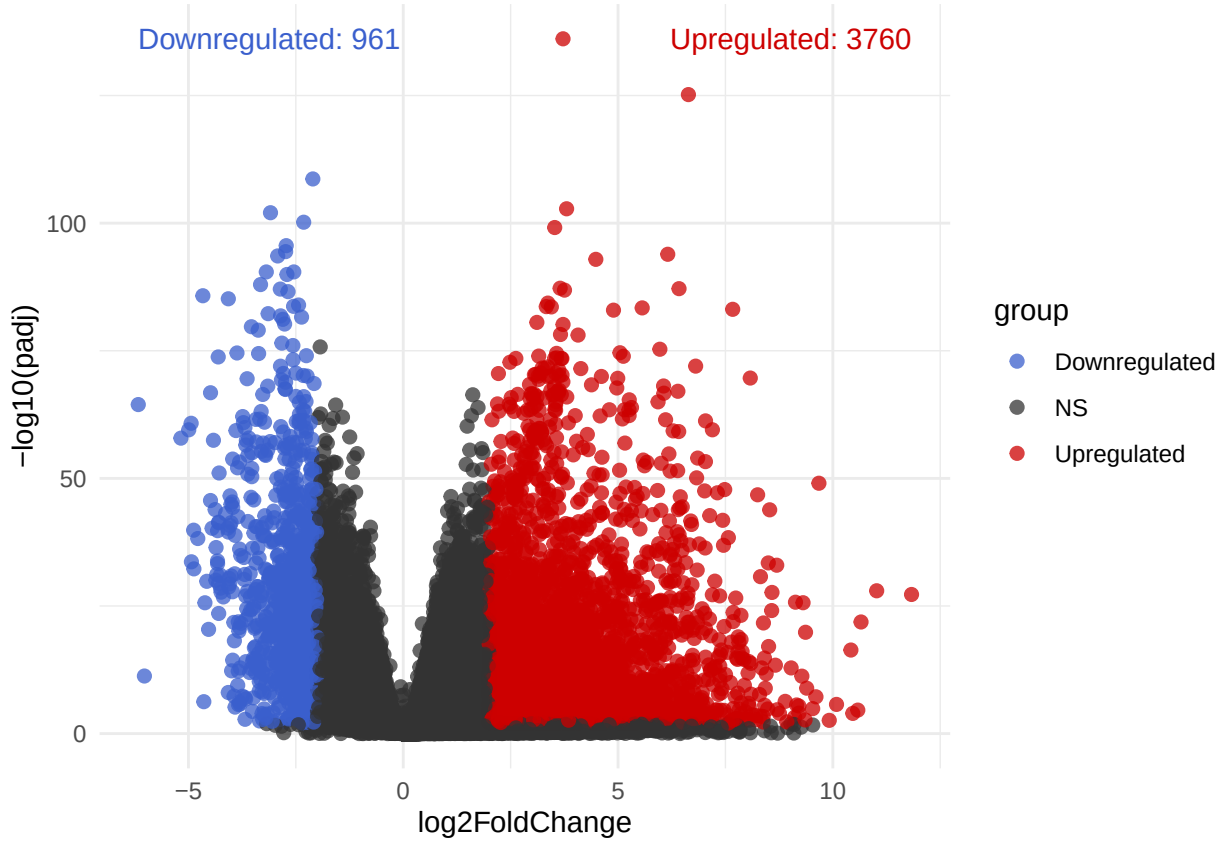



```
# ntop specify the number of top genes to sue for PCA, selected by highest row variance
```

Additionally, a volcano plot was created to highlight genes that are significantly differentially expressed, with the x-axis representing log2 fold change and the y-axis showing the -log10 of the adjusted p-value. This provides a visual summary of statistically significant genes, with those showing large fold changes and low p-values appearing in the corners of the plot. Red indicates significantly upregulated genes and blue indicates downregulated ones, with the number of genes in each category also displayed.

```
### Volcano plot ###
vol = res.DE %>% filter(!is.na(padj)) # Exclude genes with NA padj
# Determine which genes are upregulated, downregulated,
# or not significant (NS) based on the DEG cutoff
vol$group = with(vol, ifelse(padj < adjp & log2FoldChange > FC, "Upregulated",
                             ifelse(padj < adjp & log2FoldChange < -FC,
                                    "Downregulated", "NS")))
# Count the number of genes that are upregulated, downregulated, or NS
numDEG = table(vol$group)

ggplot(vol, aes(x=log2FoldChange, y=-log10(padj), color = group)) +
  geom_point(size = 2, alpha = 0.75) +
  scale_color_manual(values = c("Upregulated" = "red3",
                                "Downregulated" = "royalblue3",
                                "NS" = "gray20")) +
  annotate("text", x = max(vol$log2FoldChange),
           y = max(-log10(vol$padj), na.rm = TRUE),
```

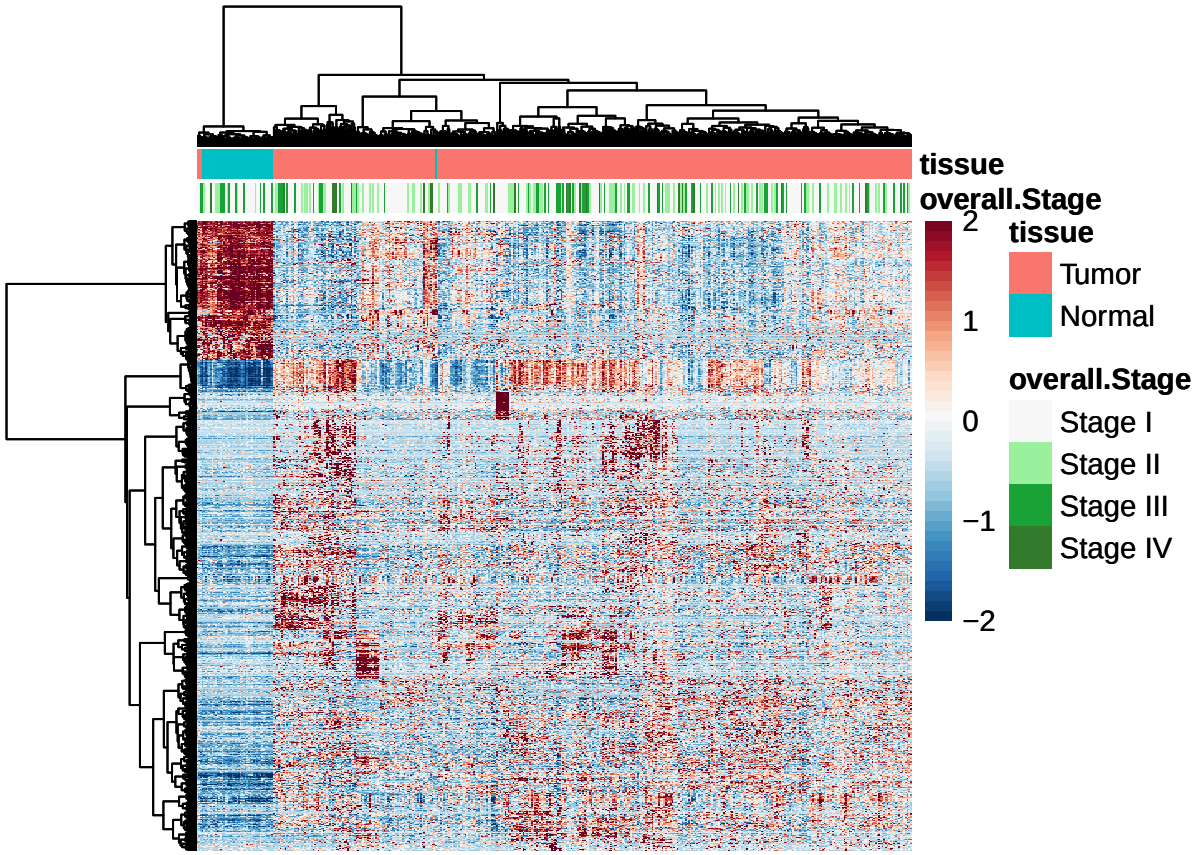
```
          label = paste("Upregulated:",
                        numDEG["Upregulated"]),
          color = "red3", hjust = 1) +
  annotate("text", x = min(vol$log2FoldChange),
          y = max(-log10(vol$padj), na.rm = TRUE),
          label = paste("Downregulated:",
                        numDEG["Downregulated"]),
          color = "royalblue3", hjust = 0) +
  theme_minimal()
```



To further examine patterns of gene expression across samples, a heatmap of the differential expressed genes was generated. The VST-transformed data were scaled by row to highlight relative expression changes. Clustering was applied to both genes and samples to identify potential subgroups, and clinical annotations such as tumor stage and tissue type were overlaid to aid biological interpretation. The resulting heatmap showed clear separation between tumor and normal samples, indicating distinct gene expression profiles between the two groups.

```
### Heatmap annotation ###
# Define annotation data
AnnData <- data.frame(overall.Stage = clinical.info.subset$overall.Stage,
                      tissue = clinical.info.subset$tissue_type)
rownames(AnnData) <- clinical.info.subset$ID

# Define annotation color
Ann_color = list("overall.Stage" = c("Stage I" = "grey97",
                                     "Stage II" = "#9bf09d",
```

```
                                    "Stage III" = "#19a337",
                                    "Stage IV" = "#337a2c"),
                  "tissue" = c("Tumor" = "#F8766D","Normal" = "#00BFC4"))



# Define cluster distance
clusterDist = "euclidean" #options: "euclidean", "correlation", and more
breaksList = seq(-2, 2, by = 0.1)

TCGAheatmap_TvN = pheatmap(sig.vst_mat, scale="row", cluster_row=T, cluster_col=T,
                           clustering_distance_rows=clusterDist,
                           clustering_distance_cols=clusterDist,
                           clustering_method="ward.D2",
                           color=colorRampPalette(
                             rev(brewer.pal(n = 11,
                                            name ="RdBu")))(length(breaksList)),
                           border_color="grey10", show_rownames = F,
                           show_colnames = F, fontsize = 11,
                           annotation_col = AnnData, breaks = breaksList,
                           treeheight_row = 70,annotation_colors = Ann_color)

TCGAheatmap_TvN
```



Gene Ontology analysis (Tumor vs. Normal tissue)

To gain insight into the biological functions associated with the differential expressed genes, I performed Gene Ontology (GO) enrichment analysis using the clusterProfiler package (Yu et al, 2012; Xu et al., 2024). This analysis focused on the Biological Process (BP) ontology to identify pathways and processes that are significantly overrepresented among the significant genes.

To visualize the results, I used multiple plot types:

- The dot plot provides an overview of enriched terms along with their gene ratios and adjusted p-values.
  - Many genes involved in cell division are highly dysregulated in tumor tissue compared to normal tissue, suggesting disrupted control of the cell cycle and contributing to unchecked tumor growth.
- The tree plot displays hierarchical relationships between GO terms to highlight functional clusters.
  - The tree plot highlights multiple functional clusters, including regulation of mitotic division, and other processes involved in cell division. It also reveals additional key pathways related to immune response and extracellular matrix organization, both of which play important roles in cancer progression.
- The cnetplot (category network plot) shows how individual genes are connected to multiple enriched terms, offering insights into shared functional roles.

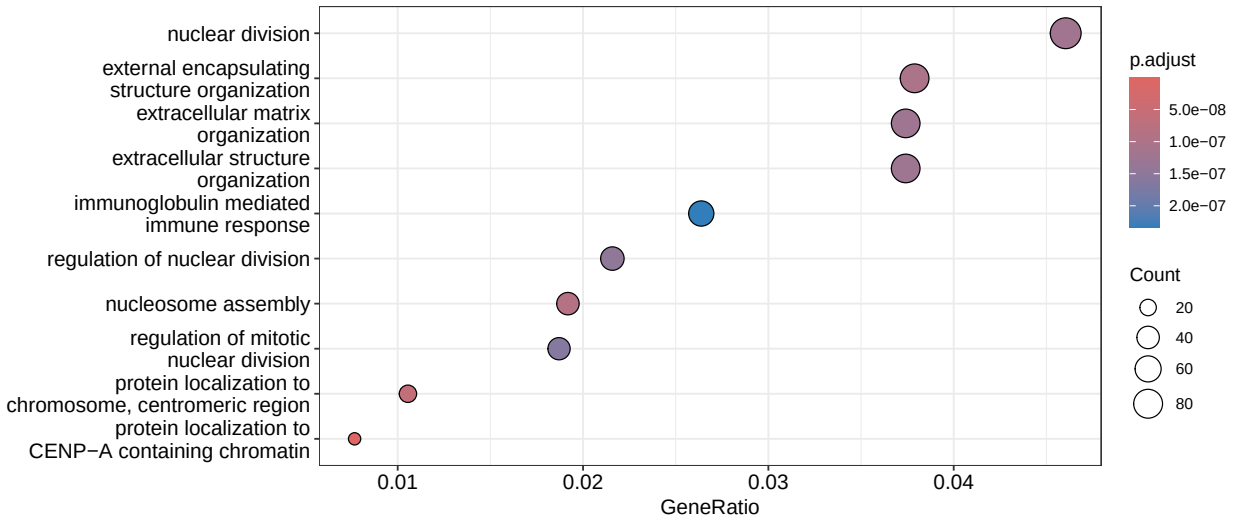
These visualizations collectively help contextualize the transcriptional changes observed across conditions.

```
### GO analysis ###

go_enrich = enrichGO(
  gene = sigGenes,
  OrgDb = org.Hs.eg.db,
  keyType = "SYMBOL",
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05)

go_results = as.data.frame(go_enrich)
head(go_results, n = c(6,10))
```

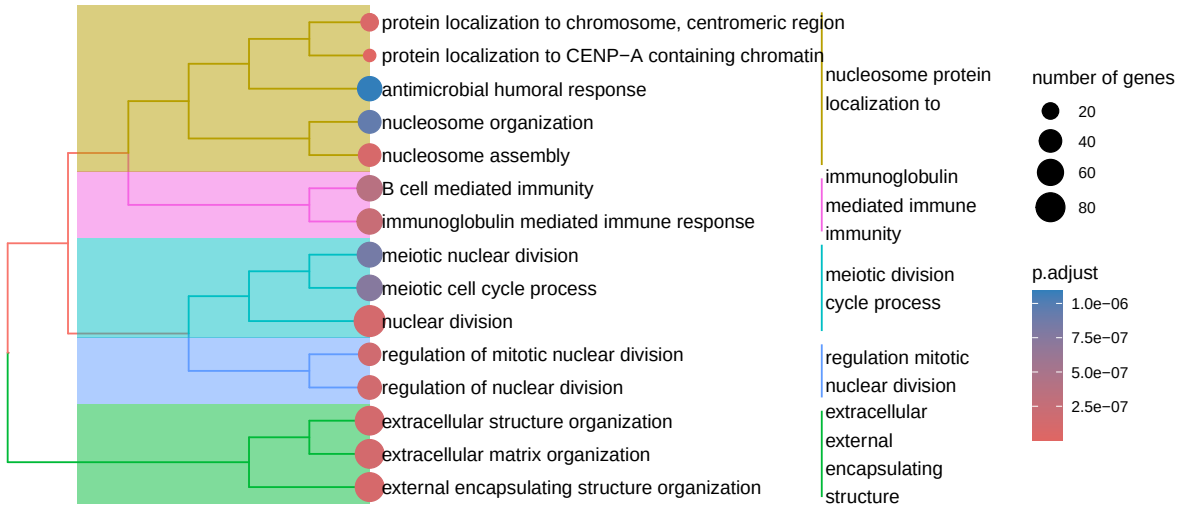
```
##                    ID                                          Description
## GO:0061644 GO:0061644    protein localization to CENP-A containing chromatin
## GO:0071459 GO:0071459 protein localization to chromosome, centromeric region
## GO:0006334 GO:0006334                                 nucleosome assembly
## GO:0045229 GO:0045229          external encapsulating structure organization
## GO:0000280 GO:0000280                                     nuclear division
## GO:0030198 GO:0030198                        extracellular matrix organization
##            GeneRatio   BgRatio RichFactor FoldEnrichment    zScore       pvalue
## GO:0061644   16/2085  18/18986  0.8888889       8.094218 10.576260 5.236977e-14
## GO:0071459   22/2085  41/18986  0.5365854       4.886144  8.749145 2.164101e-11
## GO:0006334   40/2085 120/18986  0.3333333       3.035332  7.855757 4.777422e-11
## GO:0045229   79/2085 341/18986  0.2316716       2.109600  7.262140 7.299758e-11
## GO:0000280   96/2085 451/18986  0.2128603       1.938305  7.083341 1.078501e-10
## GO:0030198   78/2085 339/18986  0.2300885       2.095185  7.146360 1.364601e-10
##                 p.adjust       qvalue
## GO:0061644 2.909665e-10 2.547927e-10
```

```
## GO:0071459 6.011873e-08 5.264461e-08
## GO:0006334 8.847786e-08 7.747806e-08
## GO:0045229 1.013936e-07 8.878811e-08
## GO:0000280 1.198430e-07 1.049438e-07
## GO:0030198 1.257411e-07 1.101086e-07
```

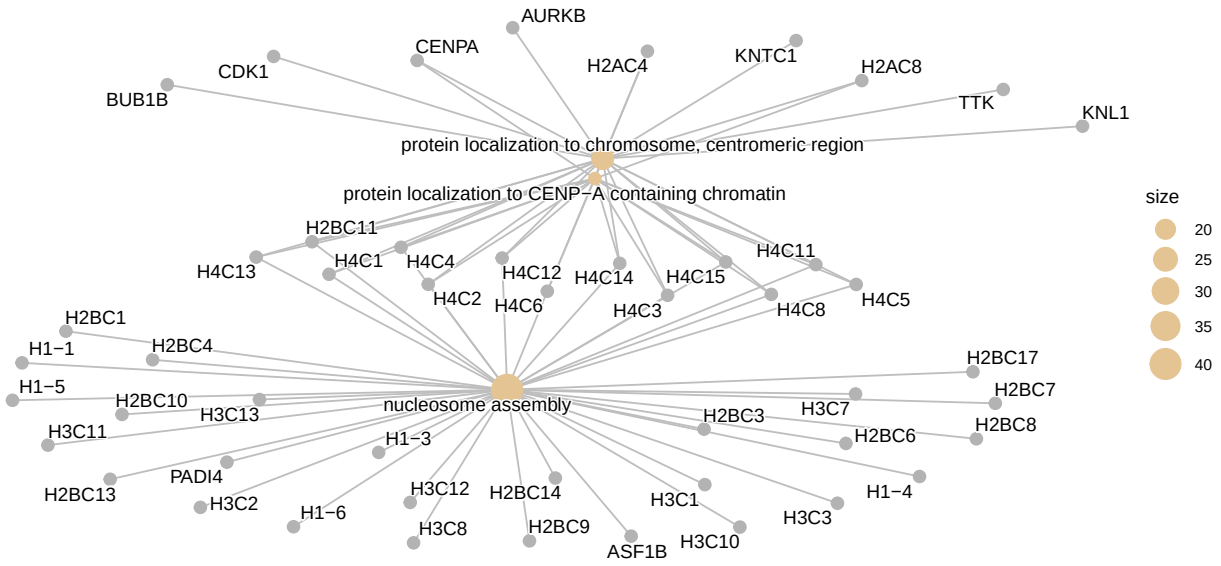
```r
# Dot plot
dotplot(go_enrich, showCategory = 10)
```




```r
# Enrichment Map
enrich_result = pairwise_termsim(go_enrich)
treeplot(enrich_result, showCategory = 15, label_format = 20)
```




```r
# Network plot
cnetplot(go_enrich, foldChange = NULL, showCategory = 3, node_label = "all")
```



Gene Expression Analysis by Tumor Stage

To investigate how gene expression varies across tumor progression, I performed differential expression analysis comparing late-stage (Stage IV) and early-stage (Stage II) lung adenocarcinoma tumors. By categorizing samples based on clinical staging information, I used DESeq2 to identify genes that are significantly up- or downregulated in late-stage tumors relative to early-stage cases. This analysis provides insight into molecular changes associated with tumor advancement and may highlight genes relevant to metastasis or aggressive disease behavior.

The samples were filtered to retain only the relevant expression data required for downstream analysis:

```
# Only keep tumor samples
clinical.info.tumor = clinical.info.subset %>%
  filter(tissue_type == "Tumor") %>%
  filter(overall.Stage != "Stage I") %>%
  filter(overall.Stage != "Stage III")

# Classify tumor samples into early-stage or late-stage based on overall stage
clinical.info.tumor = clinical.info.tumor %>%
  mutate(tumorCategory = case_when(overall.Stage == "Stage II" ~ "Early_Stage",
                                   overall.Stage == "Stage IV" ~ "Late_Stage"))

dataPrep.luad.tumor = dataPrep.luad[, colnames(dataPrep.luad) %in%
                                      clinical.info.tumor$ID]

# Verify that all samples in the clinical information
# are present in the count data and that naming is consistent
all(rownames(clinical.info.tumor) %in% colnames(dataPrep.luad.tumor))
```

```
## [1] TRUE
```

```
# Reorder the count matrix to match the sample order in the clinical information
dataPrep.luad.tumor <- dataPrep.luad.tumor[, rownames(clinical.info.tumor)]
# Confirm that the sample order is identical;
```

```r
# this is critical to prevent misalignment between counts and sample information in DESeq2
all(rownames(clinical.info.tumor) == colnames(dataPrep.luad.tumor))
```

```
## [1] TRUE
```

Following the same DESeq2 workflow described for the tumor vs. normal comparison:

```r
### Set up DESeq data set ###
dds.tumor <- DESeqDataSetFromMatrix(countData = dataPrep.luad.tumor,
                              colData = clinical.info.tumor,
                              design = ~ tumorCategory)
dds.tumor
```

```
## class: DESeqDataSet
## dim: 60660 111
## metadata(1): version
## assays(1): counts
## rownames(60660): TSPAN6 TNMD ... AL391628.1 AP006621.6
## rowData names(0):
## colnames(111): TCGA-38-4629-01A-02R-1206-07
##   TCGA-91-7771-01A-11R-2170-07 ... TCGA-05-4427-01A-21R-1858-07
##   TCGA-44-7667-01A-31R-2066-07
## colData names(12): ID vital_status ... overall.Stage tumorCategory
```

```r
### Pre-filtering ###
keep <- rowSums(counts(dds.tumor)) >= 10
dds.tumor <- dds.tumor[keep,]


### factor levels ###
#specify reference level (ex. "untreated")
dds.tumor$tumorCategory <- relevel(dds.tumor$tumorCategory, ref = "Early_Stage")

### Differential expression analysis ###
dds.tumor <- DESeq(dds.tumor, betaPrior = F, quiet = T)

res.tumor <- results(dds.tumor)
res.tumor
```

```
## log2 fold change (MLE): tumorCategory Late Stage vs Early Stage
## Wald test p-value: tumorCategory Late Stage vs Early Stage
## DataFrame with 47604 rows and 6 columns
##               baseMean log2FoldChange     lfcSE      stat    pvalue      padj
##              <numeric>      <numeric> <numeric> <numeric> <numeric> <numeric>
## TSPAN6      3461.32898     -0.2870798  0.196462 -1.461248 0.1439473  0.619608
## TNMD           9.30947     -2.2270851  0.884394 -2.518203 0.0117955  0.238933
## DPM1        1786.29099     -0.1293544  0.142592 -0.907161 0.3643216  0.802910
## SCYL3        815.32116      0.0398546  0.116454  0.342235 0.7321740  0.943065
## C1orf112     458.82422      0.1585012  0.192345  0.824045 0.4099140  0.827796
## ...                ...            ...       ...       ...       ...       ...
## AC078856.1   0.2409036      0.2691080  0.999057  0.269362  0.787651        NA
## AC008763.4   0.0823549     -0.2403514  2.143735 -0.112118  0.910730        NA
```

```
## AL592295.6 287.3517710      0.0387073  0.162817  0.237735  0.812087  0.961572
## AL391628.1   6.7765061      0.3444673  0.221909  1.552291  0.120593  0.589124
## AP006621.6  16.7352887     -0.0342971  0.177163 -0.193591  0.846496  0.969069
```

```
## LFC (log fold change) shrinkage
resLFC.tumor = lfcShrink(dds.tumor, coef = 2, type="apeglm",
                         res = res.tumor, quiet = T)
resLFC.tumor
```

```
## log2 fold change (MAP): tumorCategory Late Stage vs Early Stage
## Wald test p-value: tumorCategory Late Stage vs Early Stage
## DataFrame with 47604 rows and 5 columns
##               baseMean log2FoldChange      lfcSE    pvalue      padj
##              <numeric>      <numeric>  <numeric> <numeric> <numeric>
## TSPAN6      3461.32898   -3.95722e-05 0.00144294 0.1439473  0.619608
## TNMD           9.30947   -9.59325e-07 0.00144269 0.0117955  0.238933
## DPM1        1786.29099   -6.44057e-06 0.00144263 0.3643216  0.802910
## SCYL3        815.32116    3.08796e-06 0.00144258 0.7321740  0.943065
## C1orf112     458.82422    3.01957e-06 0.00144266 0.4099140  0.827796
## ...                ...            ...        ...       ...       ...
## AC078856.1   0.2409036    5.43665e-07 0.00144269  0.787651        NA
## AC008763.4   0.0823549   -1.91454e-07 0.00144269  0.910730        NA
## AL592295.6 287.3517710    1.67698e-05 0.00144269  0.812087  0.961572
## AL391628.1   6.7765061    7.48971e-06 0.00144267  0.120593  0.589124
## AP006621.6  16.7352887   -1.13341e-06 0.00144265  0.846496  0.969069
```

```
## Export full DESeq2 result
res.DE.tumor = as.data.frame(resLFC.tumor)
rownames(res.DE.tumor) <- gsub("\\.", "-", rownames(res.DE.tumor))

DEgroups_export = resultsNames(dds.tumor)[2]
write.csv(as.data.frame(res.DE.tumor),
          file=paste("TCGA-LUAD_DESeq2result_",
                     DEgroups_export,".csv", sep = ""))
```

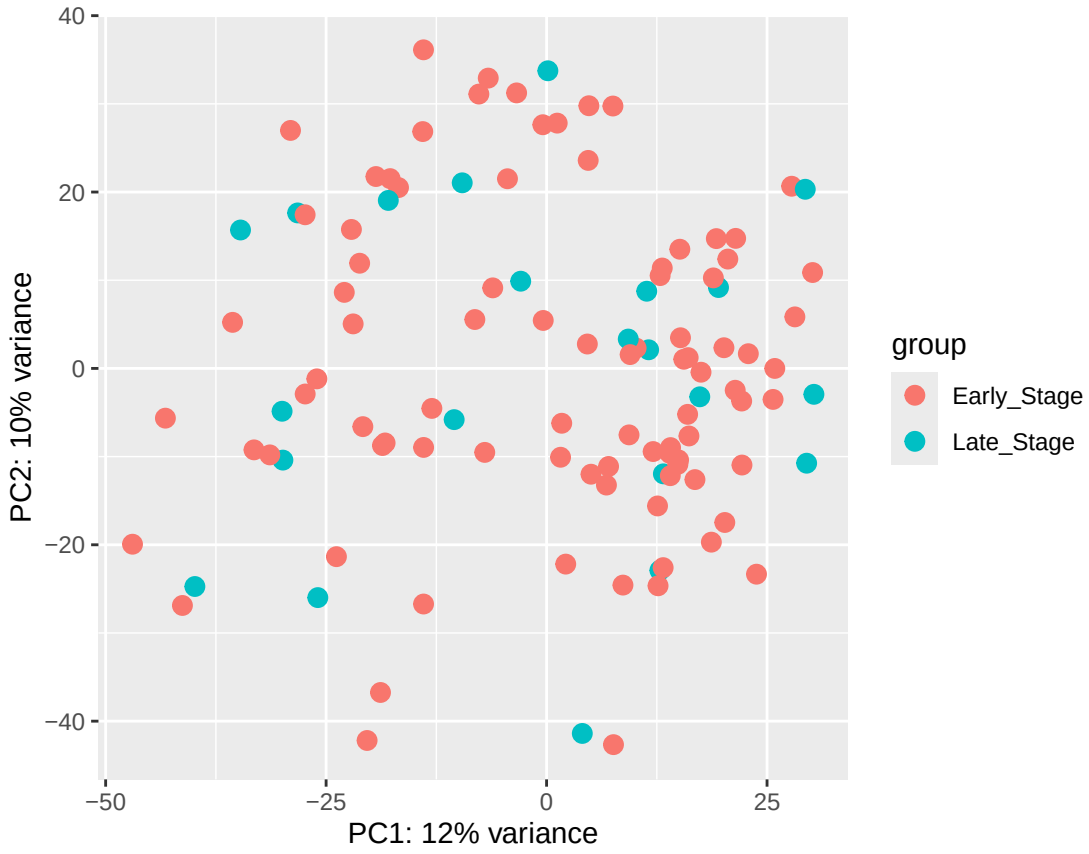
```
### Differential expressed gene (DEG) threshold ###
FC = 1 # log2 fold change
adjp = 0.05 # adjusted p-values

# Determine significant DEGs based on fold-change and adjusted p-value cut-off
sigGenes.tumor <- rownames(subset(res.DE.tumor,
                                  (abs(log2FoldChange)>=FC & padj<=adjp )))


### Variance Stablizing Transformation (VST) ###
vsd.tumor = vst(dds.tumor, blind = F)

# Extract the matrix of VST-transformed values
vst_mat.tumor = assay(vsd.tumor)
sig.vst_mat.tumor = subset(vst_mat.tumor,
                           rownames(vst_mat.tumor) %in% sigGenes.tumor)
```

Visualization (late-stage vs. early-stage tumors)

The PCA plot for the late-stage vs. early-stage tumor comparison does not show clear separation between the two groups, suggesting that their overall gene expression profiles are not as distinct as those observed in the tumor vs. normal comparison. This is further reflected in the volcano plot, where the number of differential expressed genes is considerably lower. One possible explanation is that late-stage and early-stage tumors share many core tumor-associated gene expression patterns, resulting in fewer transcriptomic differences between them.

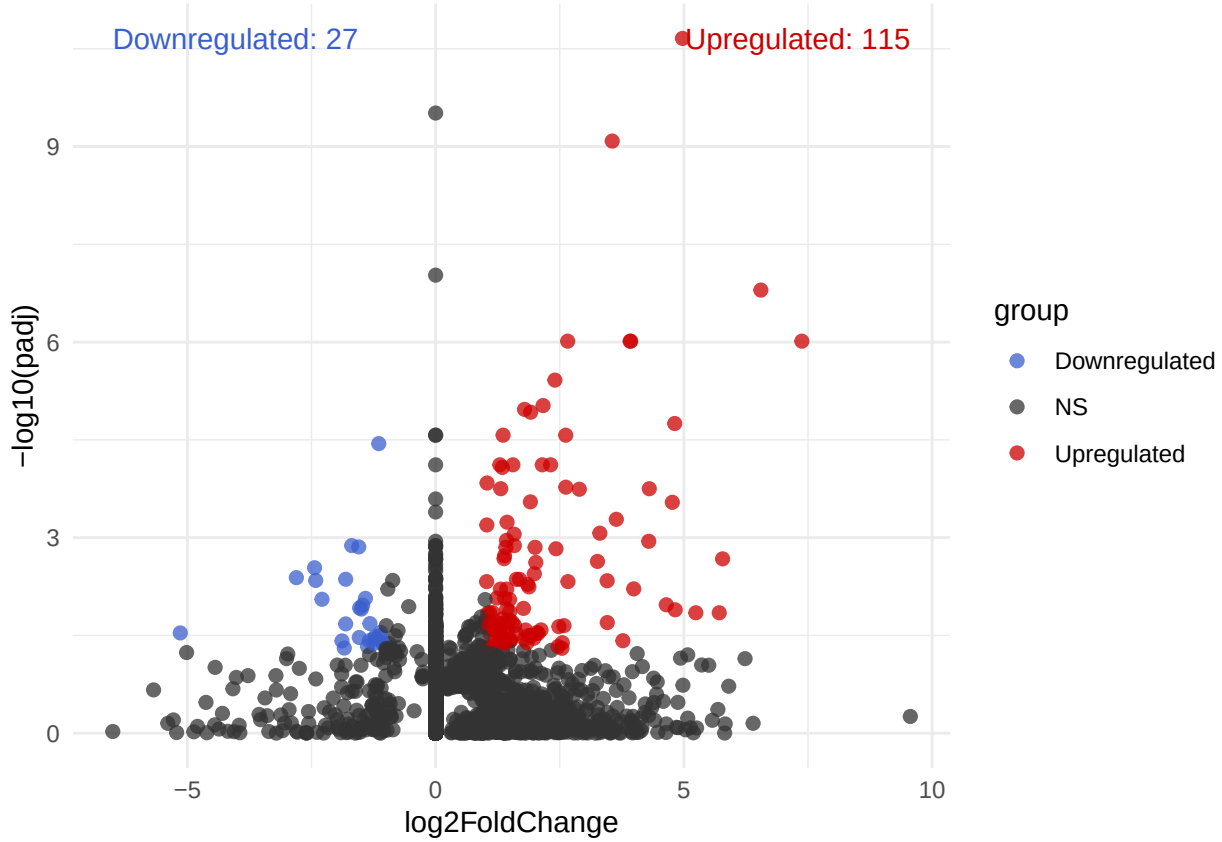
```
### Visualization ###

### PCA plot ###
plotPCA(vsd.tumor, intgroup = "tumorCategory", ntop = 500)
```

```
## using ntop=500 top features by variance
```




```
### Volcano plot ###
vol = res.DE.tumor %>% filter(!is.na(padj)) # Exclude genes with NA padj
# Determine which genes are upregulated, downregulated,
# or not significant (NS) based on the DEG cutoff
vol$group = with(vol, ifelse(padj < adjp & log2FoldChange > FC, "Upregulated",
                             ifelse(padj < adjp & log2FoldChange < -FC,
                                    "Downregulated", "NS")))

# Count the number of genes that are upregulated, downregulated, or NS
```

```
numDEG = table(vol$group)

ggplot(vol, aes(x=log2FoldChange, y=-log10(padj), color = group)) +
  geom_point(size = 2, alpha = 0.75) +
  scale_color_manual(values = c("Upregulated" = "red3",
                                "Downregulated" = "royalblue3",
                                "NS" = "gray20")) +
  annotate("text", x = max(vol$log2FoldChange),
           y = max(-log10(vol$padj), na.rm = TRUE),
           label = paste("Upregulated:",
                         numDEG["Upregulated"]),
           color = "red3", hjust = 1) +
  annotate("text", x = min(vol$log2FoldChange),
           y = max(-log10(vol$padj), na.rm = TRUE),
           label = paste("Downregulated:",
                         numDEG["Downregulated"]),
           color = "royalblue3", hjust = 0) +
  theme_minimal()
```



Gene Ontology analysis (late-stage vs. early-stage tumors) Despite the overall similarity in gene expression profiles, additional changes were still observed in late-stage tumors compared to early-stage tumors. These alterations may play a critical role in the metastatic process, as late-stage (Stage IV) cancer is defined by the spread of tumor cells beyond the original site to distant organs, whereas early-stage (Stage II) remains largely confined to the primary location. Gene Ontology (GO) analysis revealed that many differentially expressed genes are traditionally associated with neurological signaling and regulatory processes. Interestingly, the central nervous system is one of the most common metastatic sites for lung

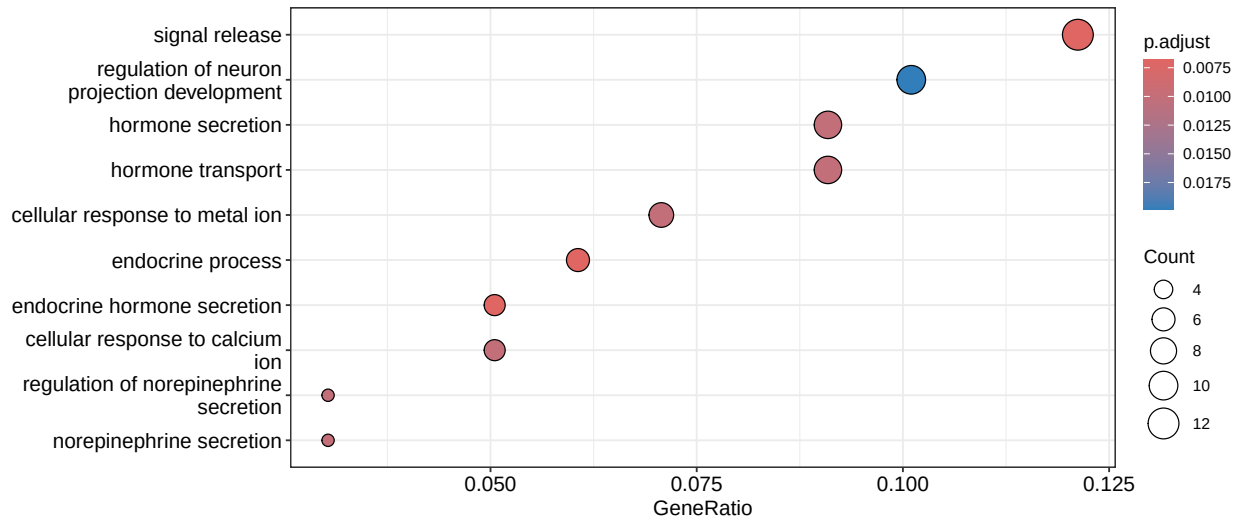
adenocarcinoma (Cagney et al., 2017; Suh et al., 2020; Soffietti et al., 2020). This finding suggests that the differential expressed genes may reflect adaptations that enable tumor cells to survive and thrive within the microenvironment of the central nervous system. The cnetplot highlighted several candidate genes that may warrant further investigation. For example, NLGN1 has been implicated in promoting cancer–nerve interactions, while FZD4 and KCNB1 have been associated with epithelial-to-mesenchymal transition and glioma progression, respectively(Bizzozero et al., 2022; Sompel et al., 2021; Wang et al., 2017).

```
### GO analysis ###
go_enrich = enrichGO(
  gene = sigGenes.tumor,
  OrgDb = org.Hs.eg.db,
  keyType = "SYMBOL",
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05
)

go_results = as.data.frame(go_enrich)
head(go_results)
```

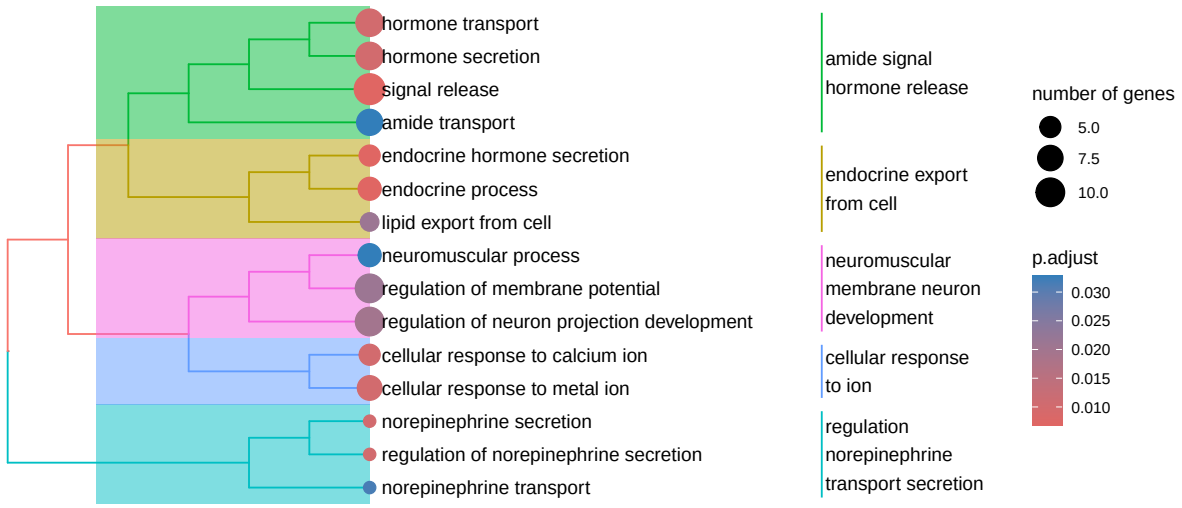
```
##                    ID                          Description GeneRatio
## GO:0050886 GO:0050886                     endocrine process      6/99
## GO:0023061 GO:0023061                       signal release     12/99
## GO:0060986 GO:0060986           endocrine hormone secretion      5/99
## GO:0014061 GO:0014061 regulation of norepinephrine secretion      3/99
## GO:0046879 GO:0046879                     hormone secretion      9/99
## GO:0048243 GO:0048243              norepinephrine secretion      3/99
##              BgRatio RichFactor FoldEnrichment    zScore       pvalue
## GO:0050886  93/18986 0.06451613      12.372760  7.959729 9.117470e-06
## GO:0023061 493/18986 0.02434077       4.668019  5.974379 9.800512e-06
## GO:0060986  58/18986 0.08620690      16.532567  8.577215 1.281603e-05
## GO:0014061  13/18986 0.23076923      44.256410 11.295242 3.786862e-05
## GO:0046879 319/18986 0.02821317       5.410658  5.751788 4.384730e-05
## GO:0048243  14/18986 0.21428571      41.095238 10.865299 4.801402e-05
##               p.adjust      qvalue
## GO:0050886 0.006788224 0.006043770
## GO:0023061 0.006788224 0.006043770
## GO:0060986 0.006788224 0.006043770
## GO:0014061 0.010059378 0.008956182
## GO:0046879 0.010059378 0.008956182
## GO:0048243 0.010059378 0.008956182
##                                                                    geneID
## GO:0050886                                        ECRG4/WNK4/CGA/CRHBP/FZD4/AVPR1B
## GO:0023061 SNCAIP/OXT/ECRG4/WNK4/VGF/CGA/ADCYAP1/CRHBP/ADRA2A/KCNB1/NLGN1/FZD4
## GO:0060986                                            ECRG4/WNK4/CGA/CRHBP/FZD4
## GO:0014061                                                       OXT/ADRA2A/KCNB1
## GO:0046879                  ECRG4/WNK4/VGF/CGA/ADCYAP1/CRHBP/ADRA2A/KCNB1/FZD4
## GO:0048243                                                       OXT/ADRA2A/KCNB1
##            Count
## GO:0050886     6
## GO:0023061    12
## GO:0060986     5
## GO:0014061     3
## GO:0046879     9
```

```
## GO:0048243      3
```

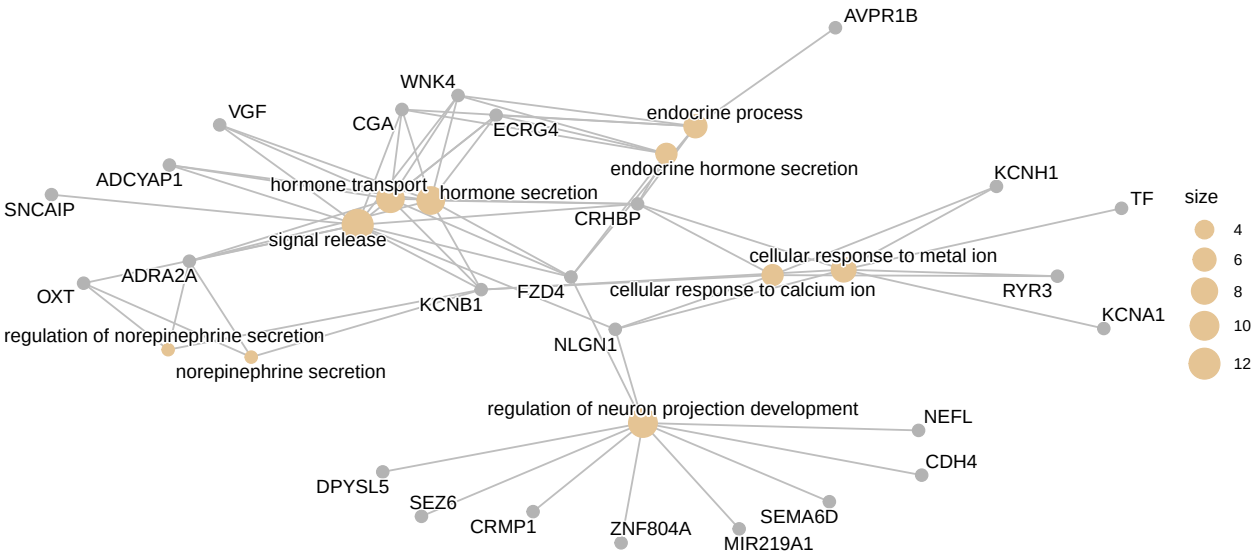
```
# Dot plot
dotplot(go_enrich, showCategory = 10)
```




```
# Enrichment Map
enrich_result = pairwise_termsim(go_enrich)
treeplot(enrich_result, showCategory = 15, label_format = 20)
```




```
# Network plot
cnetplot(go_enrich, foldChange = NULL, showCategory = 10, node_label = "all")
```



Conclusion

This analysis utilized RNA sequencing data from TCGA to investigate transcriptomic changes in lung adenocarcinoma, identifying distinct gene expression patterns between tumor and normal tissues, as well as between early stage and late stage tumors. Differential gene expression and Gene Ontology enrichment analyses identified genes involved in critical cancer-related processes such as cell cycle regulation, extracellular matrix organization, and immune system modulation, highlighting the dysregulated pathways that drive tumor progression. Although early stage and late stage tumors share many expression features, late stage samples showed additional changes in genes related to neurological signaling, which may reflect adaptations for metastasis to the central nervous system. These results enhance our understanding of lung adenocarcinoma biology and may inform future efforts to develop stage-specific diagnostics and targeted therapies.

References

Bizzozero, L., Pergolizzi, M., Pascal, D., Maldi, E., Villari, G., Erriquez, J., Volante, M., Serini, G., Marchiò, C., Bussolino, F., et al. (2022). Tumoral Neuroligin 1 Promotes Cancer–Nerve Interactions and Synergizes with the Glial Cell Line-Derived Neurotrophic Factor. Cells 11, 280.

Cagney, D.N., Martin, A.M., Catalano, P.J., Redig, A.J., Lin, N.U., Lee, E.Q., Wen, P.Y., Dunn, I.F., Bi, W.L., Weiss, S.E., et al. (2017). Incidence and prognosis of patients with brain metastases at diagnosis of systemic malignancy: a population-based study. Neuro-Oncology 19, 1511-1521.

Chang, K., Creighton, C.J., Davis, C., Donehower, L., Drummond, J., Wheeler, D., Ally, A., Balasundaram, M., Birol, I., Butterfield, Y.S.N., et al. (2013). The Cancer Genome Atlas Pan-Cancer analysis project. Nature genetics 45, 1113-1120.

Colaprico, A., Silva, T.C., Olsen, C., Garofano, L., Cava, C., Garolini, D., Sabedot, T.S., Malta, T.M., Pagnotta, S.M., Castiglioni, I., et al. (2016). TCGAbiolinks: an R/Bioconductor package for integrative analysis of TCGA data. Nucleic Acids Research 44, e71-e71.

Collisson, E.A., Campbell, J.D., Brooks, A.N., Berger, A.H., Lee, W., Chmielecki, J., Beer, D.G., Cope, L., Creighton, C.J., Danilova, L., et al. (2014). Comprehensive molecular profiling of lung adenocarcinoma. Nature 511, 543-550.

Dobin, A., Davis, C.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T.R. (2012). STAR: ultrafast universal RNA-seq aligner. Bioinformatics (Oxford, England) 29, 15-21.

Hanahan, D. (2022). Hallmarks of Cancer: New Dimensions. Cancer Discovery 12, 31-46.

Herbst, R.S., Morgensztern, D., and Boshoff, C. (2018). The biology and management of non-small cell lung cancer. Nature 553, 446.

Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology 15, 550.

Siegel, R.L., Giaquinto, A.N., and Jemal, A. (2024). Cancer statistics, 2024. CA: a cancer journal for clinicians 74, 12-49.

Soffietti, R., Ahluwalia, M., Lin, N., and Rudà, R. (2020). Management of brain metastases according to molecular subtypes. Nature Reviews Neurology 16, 557-574.

Sompel, K., Elango, A., Smith, A.J., and Tennis, M.A. (2021). Cancer chemoprevention through Frizzled receptors and EMT. Discover Oncology 12, 32.

Suh, J.H., Kotecha, R., Chao, S.T., Ahluwalia, M.S., Sahgal, A., and Chang, E.L. (2020). Current approaches to the management of brain metastases. Nature Reviews Clinical Oncology 17, 279-299.

Wang, H.-Y., Wang, W., Liu, Y.-W., Li, M.-Y., Liang, T.-Y., Li, J.-Y., Hu, H.-M., Lu, Y., Yao, C., Ye, Y.-Y., et al. (2017). Role of KCNB1 in the prognosis of gliomas and autophagy modulation. Scientific reports 7, 14.

Xu, S., Hu, E., Cai, Y., Xie, Z., Luo, X., Zhan, L., Tang, W., Wang, Q., Liu, B., Wang, R., et al. (2024). Using clusterProfiler to characterize multiomics data. Nature Protocols 19, 3292-3320.

Yu, G., Wang, L.-G., Han, Y., and He, Q.-Y. (2012). clusterProfiler: an R Package for Comparing Biological Themes Among Gene Clusters. OMICS: A Journal of Integrative Biology 16, 284-287.

Zhu, A., Ibrahim, J.G., and Love, M.I. (2018). Heavy-tailed prior distributions for sequence count data: removing the noise and preserving large differences. Bioinformatics (Oxford, England) 35, 2084-2092.