Analysis of TCGA-LIHC RNASeq dataset

Prakash Sah

2025-01-18

This document presents an analysis of RNA-seq data from the TCGA-LIHC (Liver Hepatocellular Carcinoma) cohort, part of The Cancer Genome Atlas (TCGA) project. TCGA provides a valuable resource of publicly available genomic, epigenomic, and transcriptomic datasets across various cancer types. For this analysis, pre-processed STAR-aligned gene count data was used, rather than raw FASTQ files, to reduce computational demands. The workflow focuses on commonly performed downstream RNA-seq analyses, such as normalization, differential expression, and visualization, rather than alignment and read processing.

To begin, ensure that all necessary libraries are loaded for the analysis. These include: TCGAbiolinks – for accessing and downloading TCGA data SummarizedExperiment – for managing and manipulating assay data DESeq2 – for differential expression analysis AnnotationDbi and org.Hs.eg.db – for gene annotation and ID mapping EnhancedVolcano – for creating volcano plots ComplexHeatmap – for generating detailed and customizable heatmaps These packages form the foundation for preprocessing, analysis, and visualization of the TCGA-LIHC RNA-seq dataset.

Download and prepare TCGA-LIHC dataset In this analysis, RNA-seq data from the TCGA-LIHC (Liver Hepatocellular Carcinoma) cohort has been used, with gene-level counts generated using the STAR aligner. These STAR count files can be downloaded directly from the TCGA database using the TCGAbiolinks package. Alternatively, users can manually download the dataset from the TCGA GDC portal and load it using the GDCprepare() function.

For demonstration purposes, the data download code is included but commented out, as the dataset was already available locally for this analysis.

```
# create query
query_LIHC = GDCquery(project = "TCGA-LIHC", data.category = "Transcriptome Profiling", data.type = "Ge
```

```
## --------------------------------------

## o GDCquery: Searching in GDC database

## --------------------------------------

## Genome of reference: hg38

## --------------------------------------------

## oo Accessing GDC. This might take a while...

## --------------------------------------------

## ooo Project: TCGA-LIHC
```

```
## --------------------

## oo Filtering results

## --------------------

## ooo By data.type

## ----------------

## oo Checking data

## ----------------

## ooo Checking if there are duplicated cases

## ooo Checking if there are results for the query

## -------------------

## o Preparing output

## -------------------
```

```
#GDCdownload(query)
TCGA_LIHC_data = GDCprepare(query = query_LIHC, directory = "/Users/prakashsah/Github/DESeq2/GDCdata", s
```

```
## |                                                  |  0%                      |

## Starting to add information to samples

##  => Add clinical information to samples

##  => Adding TCGA molecular information from marker papers

##  => Information will have prefix 'paper_'

## Available assays in SummarizedExperiment :
##   => unstranded
##   => stranded_first
##   => stranded_second
##   => tpm_unstrand
##   => fpkm_unstrand
##   => fpkm_uq_unstrand
```

```r
#Extract count matrix and column data
LIHC_counts_mat = assay(TCGA_LIHC_data)
coldata= colData(TCGA_LIHC_data)
```

Differential expression analysis

We begin by creating a DESeq2 object and performing differential expression analysis using the DESeq() function. TCGA RNA-seq datasets typically use Ensembl gene IDs with version numbers as row names. To make downstream interpretation and visualization easier, it's useful to convert these to gene symbols. At a minimum, the version numbers should be stripped from the Ensembl IDs to allow compatibility with gene set analysis tools. This step can be done either before or after differential expression analysis.

```r
# Create DESeq2 object using raw counts and sample metadata
dds = DESeqDataSetFromMatrix(countData = LIHC_counts_mat, colData = coldata, design = ~ tissue_type)
```

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

```r
## Filter out genes with low counts (less than 10 reads across all samples)
dds = dds[rowSums(counts(dds)) >= 10, ]

# Check the levels of 'tissue_type' to ensure the correct reference level is set
# The reference group should be "Normal" for comparison; relevel if necessary
dds$tissue_type
```

```
##   [1] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Normal Tumor
##  [11] Normal Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
##  [21] Tumor  Tumor  Normal Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor
##  [31] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal
##  [41] Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor
##  [51] Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor
##  [61] Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Normal Tumor
##  [71] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
##  [81] Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Tumor
##  [91] Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Tumor
## [101] Tumor  Normal Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor
## [111] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [121] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [131] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor
## [141] Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Normal Tumor  Tumor
## [151] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [161] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [171] Normal Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Tumor
## [181] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal Normal Tumor
## [191] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal
## [201] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor
## [211] Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal
## [221] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [231] Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [241] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [251] Normal Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [261] Normal Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor
## [271] Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor
```

```
## [281] Normal Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [291] Tumor  Normal Tumor  Tumor  Tumor  Normal Tumor  Normal Tumor  Tumor
## [301] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [311] Normal Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor
## [321] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor
## [331] Tumor  Normal Tumor  Tumor  Normal Normal Tumor  Tumor  Normal Tumor
## [341] Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Normal
## [351] Tumor  Tumor  Normal Normal Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [361] Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor  Tumor
## [371] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [381] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor
## [391] Tumor  Tumor  Tumor  Tumor  Normal Normal Tumor  Tumor  Tumor  Tumor
## [401] Tumor  Tumor  Tumor  Tumor  Tumor  Normal Tumor  Tumor  Tumor  Tumor
## [411] Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor  Tumor
## [421] Tumor  Tumor  Tumor  Tumor
## Levels: Normal Tumor
```

```
# To relevel (if needed): dds$tissue_type <- relevel(dds$tissue_type, ref = "Normal")


#Run differential expression analysis
dds = DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 5115 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

```
res = results(dds)
res #examine results
```

```
## log2 fold change (MLE): tissue type Tumor vs Normal
## Wald test p-value: tissue type Tumor vs Normal
## DataFrame with 49319 rows and 6 columns
##                      baseMean log2FoldChange     lfcSE       stat      pvalue
##                     <numeric>      <numeric> <numeric>  <numeric>   <numeric>
```

```
## ENSG00000000003.15  4975.70344     -0.1954794 0.1150055  -1.699740 8.91798e-02
## ENSG00000000005.6      3.36591     -0.9808918 0.3641989  -2.693286 7.07516e-03
## ENSG00000000419.13  1208.83144      0.0181800 0.0666717   0.272680 7.85099e-01
## ENSG00000000457.14   540.30964      0.0634797 0.0823683   0.770681 4.40896e-01
## ENSG00000000460.17   253.84426      0.8954536 0.1599579   5.598057 2.16768e-08
## ...                        ...            ...       ...        ...         ...
## ENSG00000288667.1    0.0696463     -0.5213852 1.2442589  -0.419033 6.75192e-01
## ENSG00000288669.1    2.3581873     -4.9167203 0.4878131 -10.079108 6.83439e-24
## ENSG00000288670.1  336.7530619      0.0727496 0.0984628   0.738854 4.59996e-01
## ENSG00000288674.1    5.6944966      0.7536527 0.1920161   3.924945 8.67495e-05
## ENSG00000288675.1   11.8429417      0.9884342 0.1874447   5.273204 1.34062e-07
##                           padj
##                      <numeric>
## ENSG00000000003.15 1.39159e-01
## ENSG00000000005.6  1.46069e-02
## ENSG00000000419.13 8.38471e-01
## ENSG00000000457.14 5.38014e-01
## ENSG00000000460.17 1.12077e-07
## ...                        ...
## ENSG00000288667.1           NA
## ENSG00000288669.1  1.99962e-22
## ENSG00000288670.1  5.55984e-01
## ENSG00000288674.1  2.59001e-04
## ENSG00000288675.1  6.21273e-07
```

```
# Add gene symbols to results using ENSEMBL IDs
# First, remove version numbers from ENSEMBL IDs
ensembl_ids_ver = rownames(res) #ensembl ids with version number
ensembl_ids <- gsub("\\..*$", "", ensembl_ids_ver)
head(ensembl_ids) # Confirm version numbers were removed
```

```
## [1] "ENSG00000000003" "ENSG00000000005" "ENSG00000000419" "ENSG00000000457"
## [5] "ENSG00000000460" "ENSG00000000938"
```

```
# Map ENSEMBL IDs to gene symbols
Symbol = mapIds(org.Hs.eg.db, keys = ensembl_ids, keytype = "ENSEMBL", column = "SYMBOL")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
res$Symbol = Symbol # Add gene symbols to results table
head(res)
```

```
## log2 fold change (MLE): tissue type Tumor vs Normal
## Wald test p-value: tissue type Tumor vs Normal
## DataFrame with 6 rows and 7 columns
##                     baseMean log2FoldChange     lfcSE      stat      pvalue
##                    <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000000003.15 4975.70344     -0.1954794 0.1150055 -1.699740 8.91798e-02
## ENSG00000000005.6     3.36591     -0.9808918 0.3641989 -2.693286 7.07516e-03
## ENSG00000000419.13 1208.83144      0.0181800 0.0666717  0.272680 7.85099e-01
## ENSG00000000457.14  540.30964      0.0634797 0.0823683  0.770681 4.40896e-01
## ENSG00000000460.17  253.84426      0.8954536 0.1599579  5.598057 2.16768e-08
```

```
## ENSG00000000938.13  243.30929     -0.8774605 0.1598811 -5.488207 4.06035e-08
##                           padj      Symbol
##                      <numeric> <character>
## ENSG00000000003.15 1.39159e-01      TSPAN6
## ENSG00000000005.6  1.46069e-02        TNMD
## ENSG00000000419.13 8.38471e-01        DPM1
## ENSG00000000457.14 5.38014e-01       SCYL3
## ENSG00000000460.17 1.12077e-07       FIRRM
## ENSG00000000938.13 2.02364e-07         FGR
```

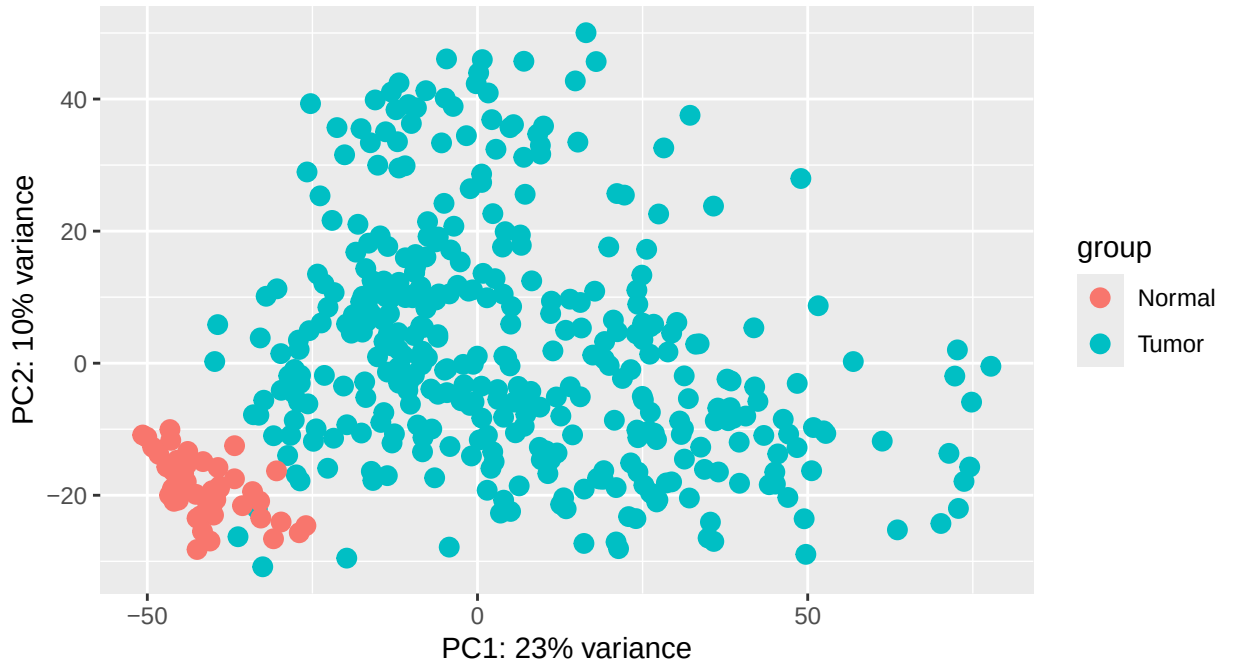
Principal Component Analysis (PCA)

DESeq2 provides tools to perform Principal Component Analysis (PCA) on transformed count data using either regularized log transformation (rlog) or variance stabilizing transformation (VST). In this analysis, we use VST to stabilize variance across the range of counts, making the data more suitable for visualization and clustering.

PCA is a dimensionality reduction technique that helps uncover major sources of variation in the dataset. It is particularly useful for exploring sample relationships, detecting batch effects, and identifying potential outliers.In the context of this TCGA-LIHC dataset, PCA can help assess whether tumor and normal liver tissue samples cluster separately, providing a quality check and confirming that the primary variation aligns with biological condition.

```
#Perform PCA analysis
vsd = vst(dds, blind = FALSE)
plotPCA(vsd, intgroup= "tissue_type")
```

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



Heatmap of Top Differentially Expressed Genes (DEGs)

Following the PCA analysis, we can further visualize the differential expression by generating a heatmap of the top differentially expressed genes (DEGs). A heatmap provides a clear, visual representation of the gene expression patterns across samples, with hierarchical clustering revealing how tumor and normal liver tissue samples group based on their gene expression profiles. The heatmap can highlight the most significantly different genes between conditions, helping to identify key biomarkers and assess the quality of the data.

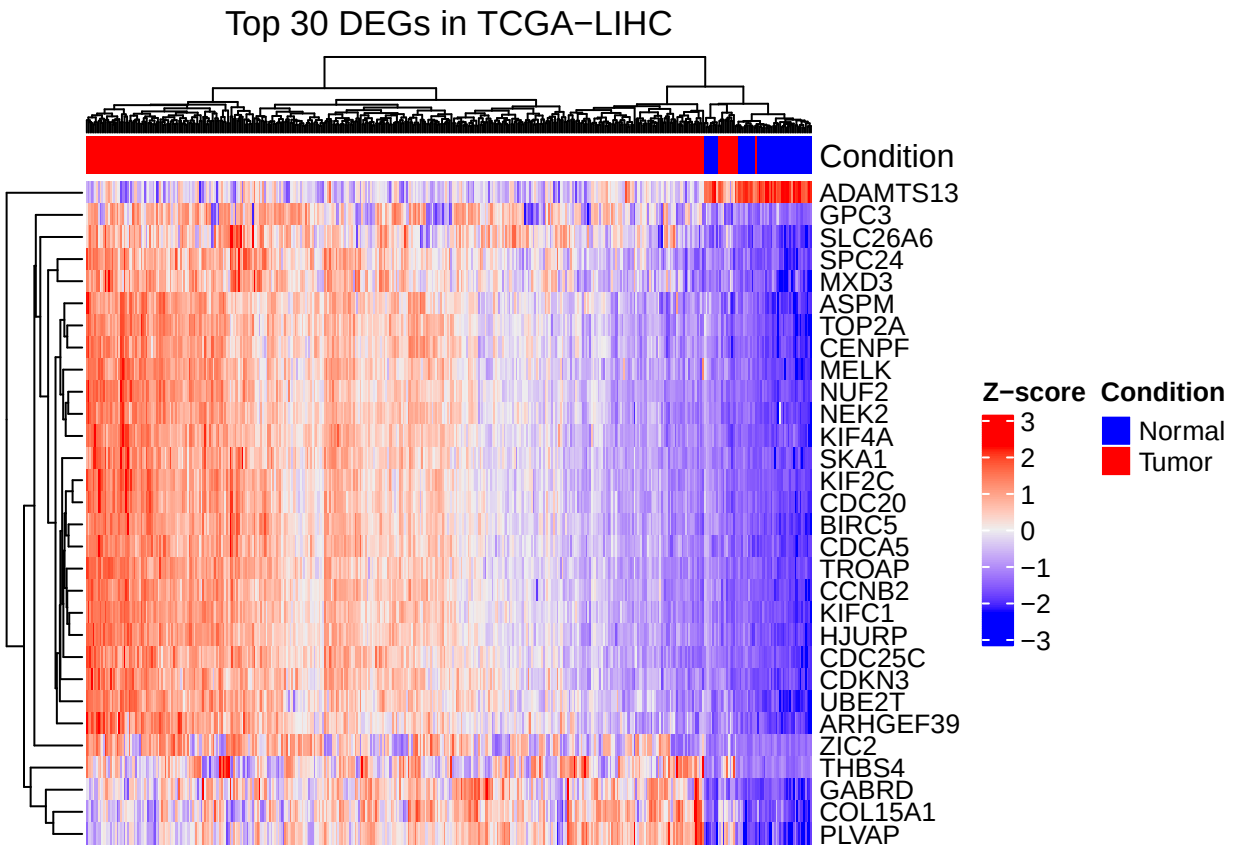
This heatmap will be created using the ComplexHeatmap package, which allows for customized and informative visualizations of gene expression data.

```
#significant DEGs
sig_degs <- res %>%
as.data.frame() %>%
filter(padj < 0.05, abs(log2FoldChange) > 1) %>%
arrange(padj)

top_30_degs = head(sig_degs, 30)

#heatmap of top 30 DEGs
norm_counts = assay(vsd)
top30 = rownames(top_30_degs)
top30_norm_counts = norm_counts[top30, ]

#column annotation for heatmap
coldata_tissue_type = data.frame(Condition = coldata$tissue_type)
ha = HeatmapAnnotation(df = coldata_tissue_type,
col = list(Condition = c("Tumor" = "red", "Normal" = "blue")))
```

```r
#change rownames to gene symbol from ensembl IDs
top30_gene_names = top_30_degs$Symbol
rownames(top30_norm_counts)=top30_gene_names
rownames(top30_norm_counts)
```

```
##  [1] "GABRD"    "PLVAP"    "CDKN3"    "CDC25C"   "UBE2T"    "CENPF"
##  [7] "NUF2"     "SKA1"     "ZIC2"     "TROAP"    "KIFC1"    "BIRC5"
## [13] "KIF4A"    "HJURP"    "ARHGEF39" "CDCA5"    "NEK2"     "ADAMTS13"
## [19] "CCNB2"    "GPC3"     "MELK"     "CDC20"    "COL15A1"  "SPC24"
## [25] "MXD3"     "KIF2C"    "ASPM"     "TOP2A"    "SLC26A6"  "THBS4"
```

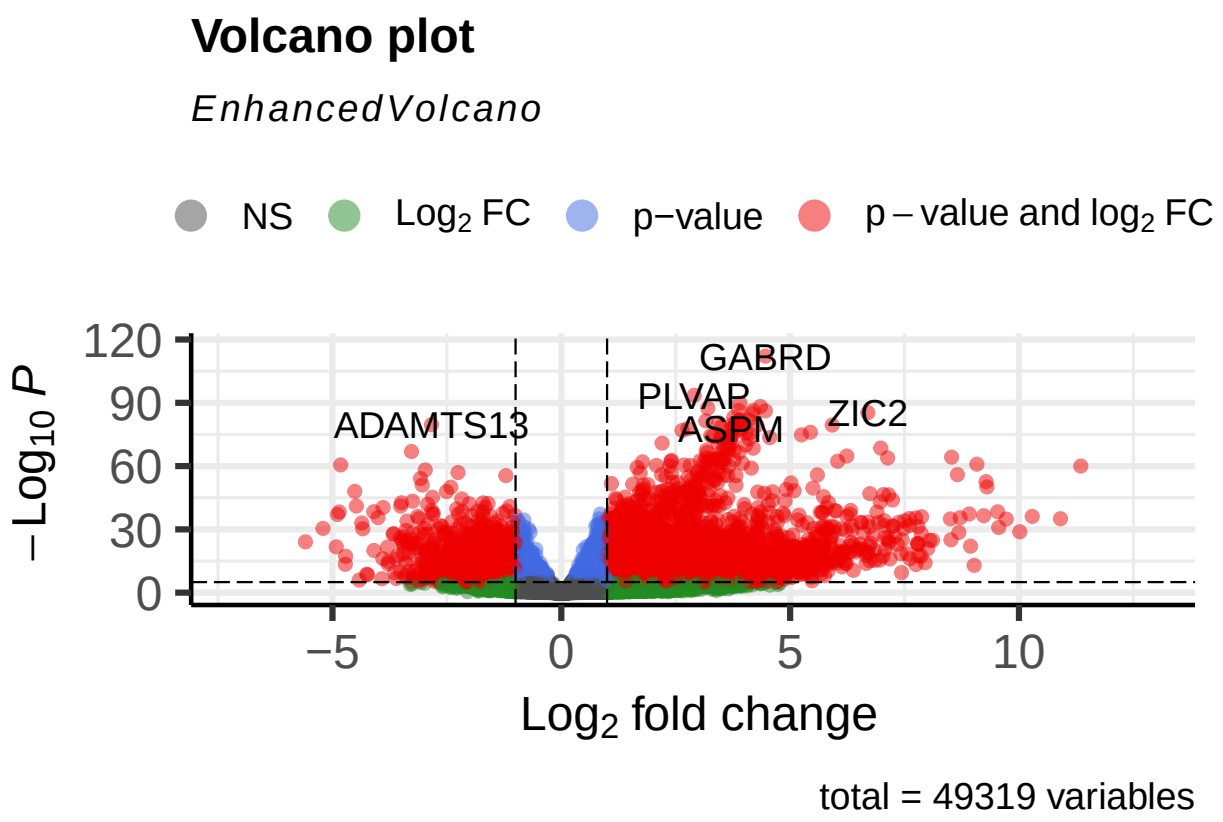
```r
#plot heatmap
Heatmap(t(scale(t(top30_norm_counts))),
name = "Z-score",
top_annotation = ha,
cluster_columns = TRUE,
show_column_names = FALSE,
row_names_gp = gpar(fontsize = 10),
column_title = "Top 30 DEGs in TCGA-LIHC")
```



A volcano plot is a widely used visualization for RNA-seq data, as it effectively displays the relationship between statistical significance (adjusted p-value) and magnitude of change (log2 fold change) for each gene. In this analysis, all genes from the res object are plotted as an example, but typically, a threshold for adjusted p-value (padj) and log2 fold change is applied to filter out genes with low significance or minimal expression changes.

Here, the plot is generated using the EnhancedVolcano package, which offers a simple and informative way to display DEGs with customizable labels and color-coding.Volcano plots can also be created using base R or, alternatively, with ggplot2 for more flexibility in customizing the plot's appearance and adding additional annotations or highlights.

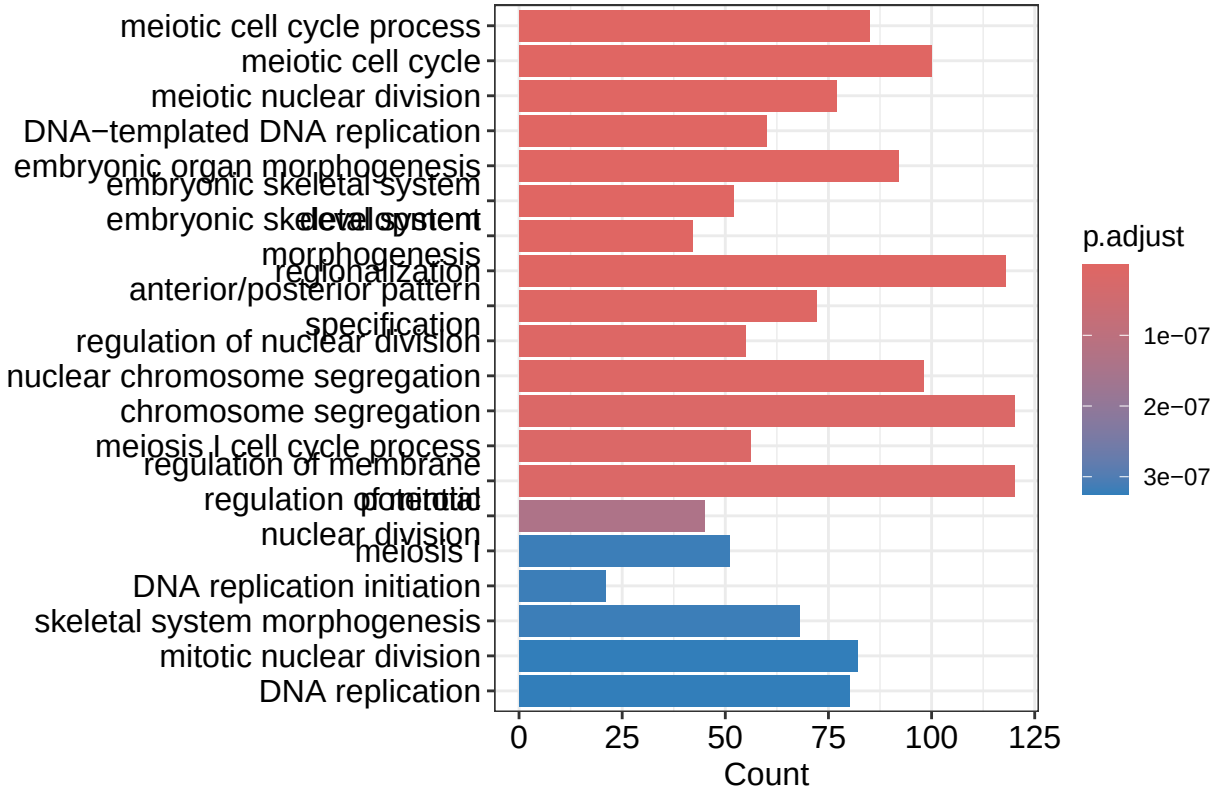
```
EnhancedVolcano(res, x = "log2FoldChange", y = "padj", lab = res$Symbol, selectLab = top_30_degs$Symbol
```



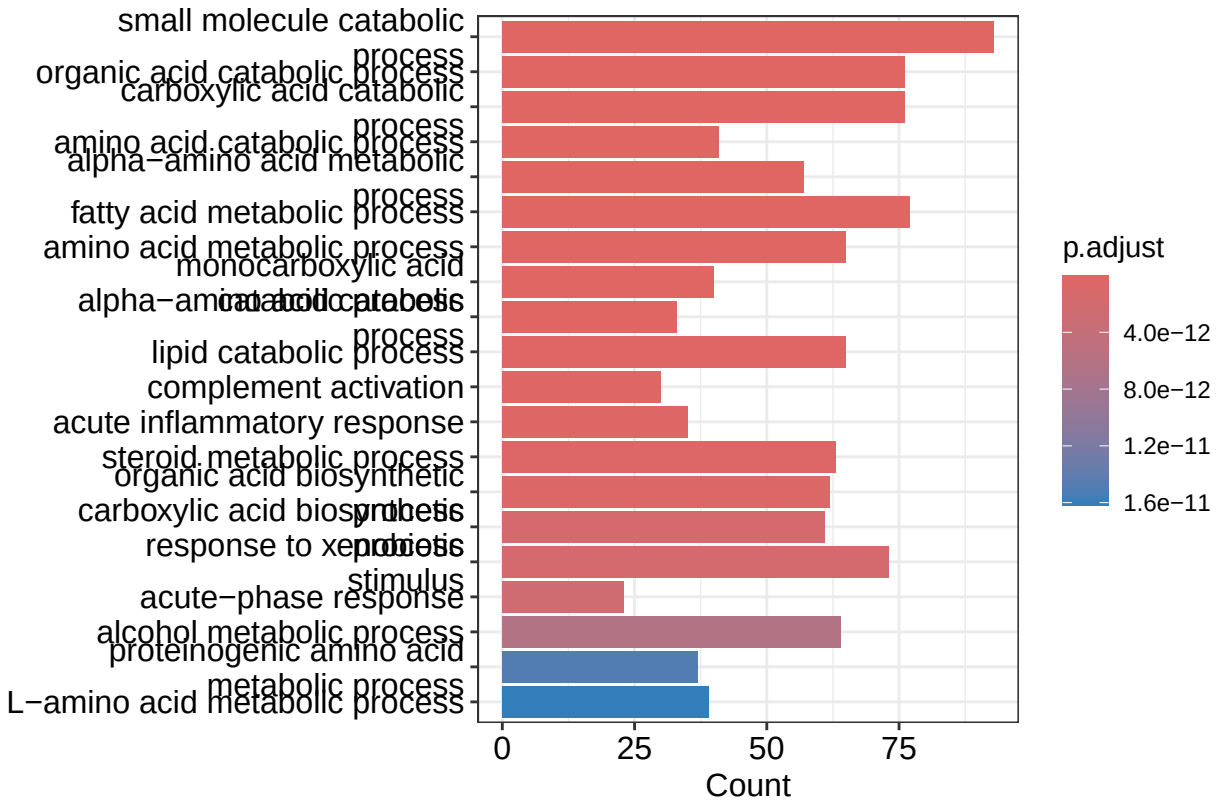
Gene ontology analysis

GO enrichment analysis is performed using the clusterProfiler package to explore biological processes associated with the differentially expressed genes. Ensembl gene IDs (with version numbers removed) are used as input. Alternatively, gene symbols can be used by setting keyType = "SYMBOL". The analysis highlights functional themes enriched among up- or downregulated genes.

```
# Gene Ontology (GO) enrichment analysis using clusterProfiler

# Separate significantly upregulated and downregulated genes
sig_up = sig_degs[sig_degs$log2FoldChange>0,]
sig_down = sig_degs[sig_degs$log2FoldChange<0,]

# Extract Ensembl gene IDs (remove version numbers) for GO analysis
genes_up = gsub("\\..*$", "", rownames(sig_up))
genes_down = gsub("\\..*$", "", rownames(sig_down))

# Perform GO enrichment analysis for upregulated genes (Biological Process ontology)
GO_up = enrichGO(gene = genes_up, OrgDb = "org.Hs.eg.db", keyType = "ENSEMBL", ont = "BP")
plot(barplot(GO_up, showCategory = 20)) # Plot top 20 enriched GO terms
```




```
# Perform GO enrichment analysis for downregulated genes (Biological Process ontology)
GO_down = enrichGO(gene = genes_down, OrgDb = "org.Hs.eg.db", keyType = "ENSEMBL", ont = "BP")
plot(barplot(GO_down, showCategory = 20)) # Plot top 20 enriched GO terms
```


small molecule catabolic process
organic acid catabolic process
carboxylic acid catabolic process
amino acid catabolic process
alpha−amino acid metabolic process
fatty acid metabolic process
amino acid metabolic process
monocarboxylic acid metabolic process
alpha−amino acid catabolic process
lipid catabolic process
complement activation
acute inflammatory response
steroid metabolic process
organic acid biosynthetic process
carboxylic acid biosynthetic process
response to xenobiotic stimulus
acute−phase response
alcohol metabolic process
proteinogenic amino acid metabolic process
L−amino acid metabolic process
0
25
50
75
Count
p.adjust
4.0e−12
8.0e−12
1.2e−11
1.6e−11