Lab-2

Load Dataset

In this step, we load the raw counts and sample mapping data for further analysis.

```
rawCounts <- read.delim("E-GEOD-50760-raw-counts.tsv")
sampleData <- read.delim("E-GEOD-50760-experiment-design.tsv")
```

```
# Prepare countData and colData
countData <- subset(rawCounts, select = -c(Gene.Name, Gene.ID))
rownames(countData) <- rawCounts$Gene.ID

colData <- data.frame(condition = sampleData$Sample.Characteristic.biopsy.site.)
rownames(colData) <- sampleData$Run
```

Differential Expression Analysis using DESeq2

Read the DESeq2 manual for reference: DESeq2 Manual

```
# Load DESeq2
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

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

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

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

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

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb
## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

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

## Loading required package: Biobase

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

##
## Attaching package: 'Biobase'

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

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

```r
# Create DESeq2 dataset
dds <- DESeqDataSetFromMatrix(countData = countData,
                              colData = colData,
                              design = ~condition)
```

```
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```r
# Run DESeq2 to identify differentially expressed genes
dds <- DESeq(dds)
```

```
## estimating size factors
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
## -- replacing outliers and refitting for 1071 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

```
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```r
res <- results(dds)

# Summary of results
summary(res)
```

```
##
## out of 44314 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 1456, 3.3%
## LFC < 0 (down)     : 1676, 3.8%
## outliers [1]       : 0, 0%
## low counts [2]     : 16758, 38%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```r
# Filter and save DE genes by p-value
DE_genes <- subset(res, padj < 0.05)
write.csv(as.data.frame(DE_genes), file = "DE_genes.csv")  # Save as .csv
```

Gene Set Enrichment Analysis

Identify functional enrichment of the DE genes using clusterProfiler and org.Hs.eg.db packages.

```r
# Load necessary libraries
library(clusterProfiler)
```

```
##
## clusterProfiler v4.14.0 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
## Guangchuang Yu, Li-Gen Wang, Yanyan Han and Qing-Yu He.
## clusterProfiler: an R package for comparing biological themes among
## gene clusters. OMICS: A Journal of Integrative Biology. 2012,
## 16(5):284-287
```

```
##
## Attaching package: 'clusterProfiler'
```

```
## The following object is masked from 'package:IRanges':
##
##     slice
```

```
## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(org.Hs.eg.db)
```

```
## Loading required package: AnnotationDbi
```

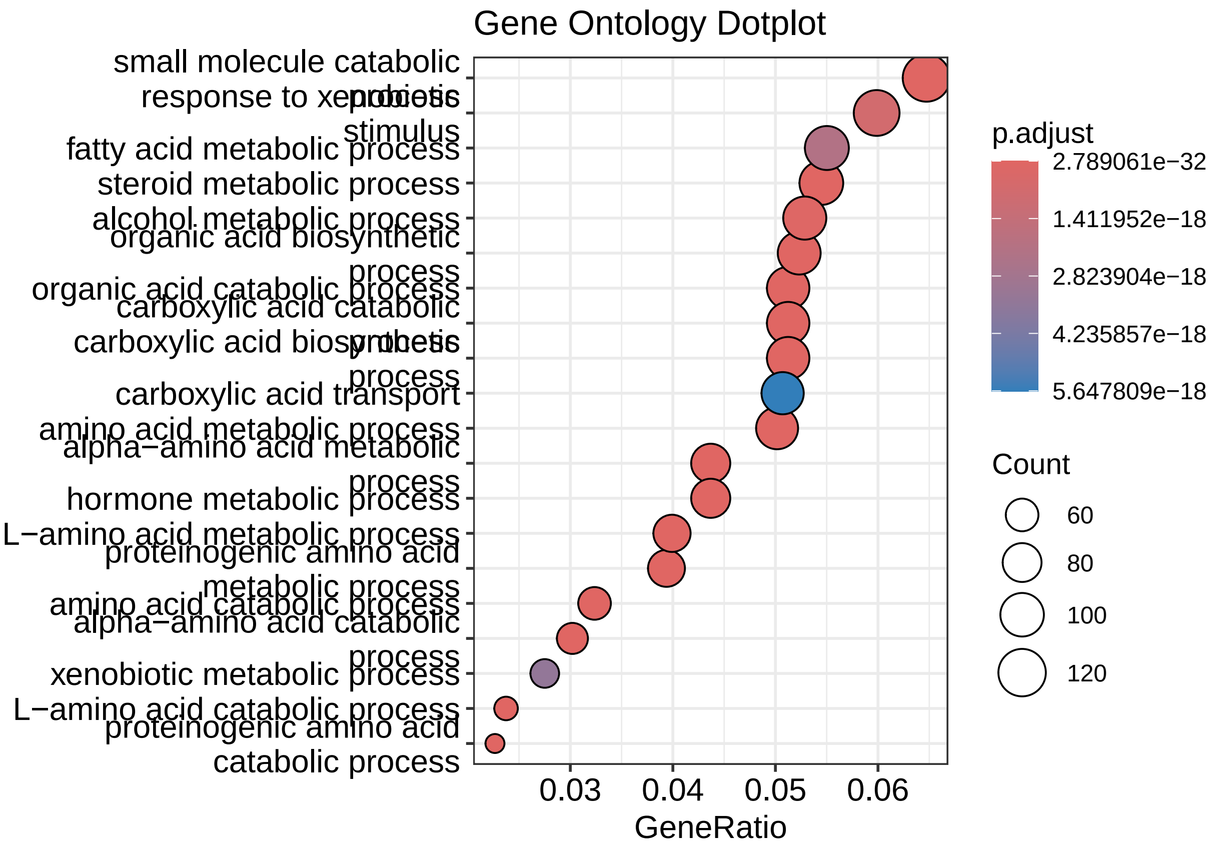
```
##
## Attaching package: 'AnnotationDbi'
```

```
## The following object is masked from 'package:clusterProfiler':
##
##     select
```

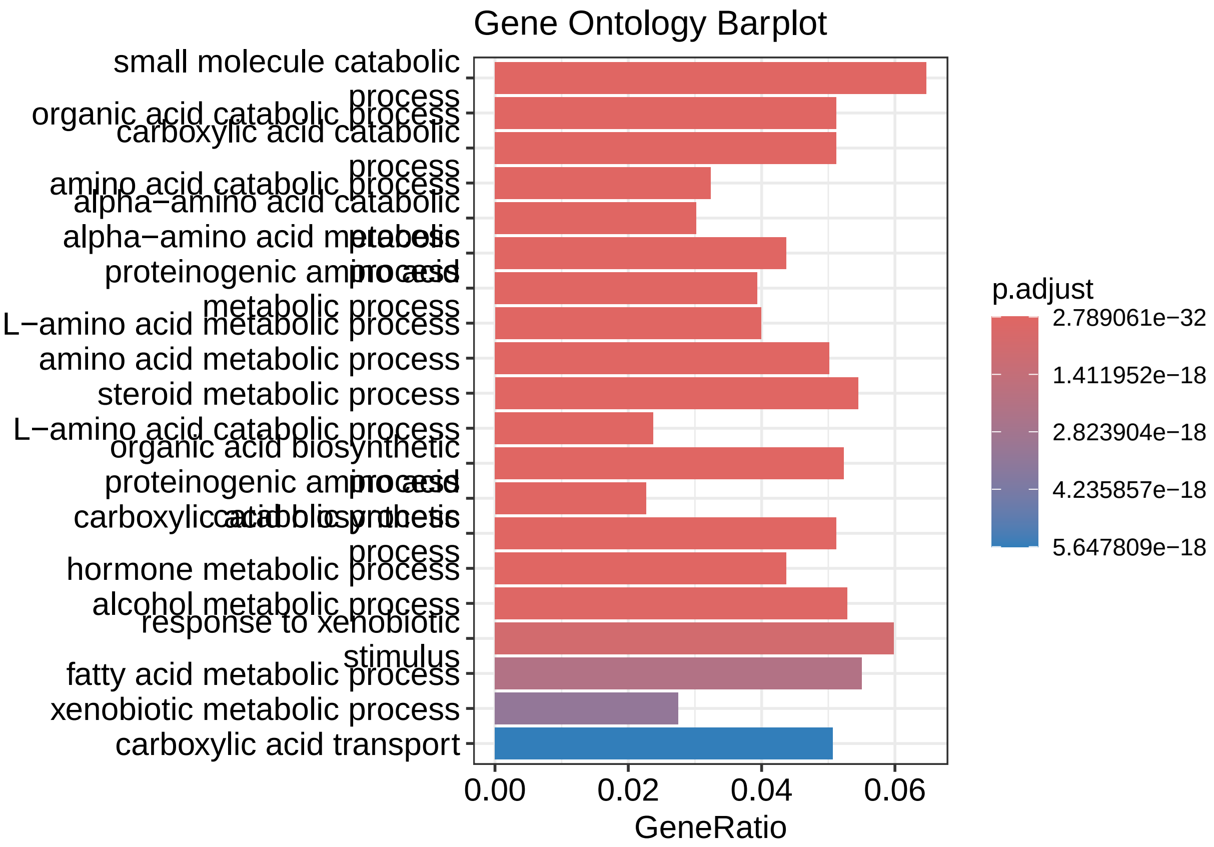
```
##
```

```r
# Gene set enrichment analysis
gene_id_list <- rownames(DE_genes)
GO_result <- enrichGO(gene_id_list,
                      OrgDb = org.Hs.eg.db,
                      ont = "ALL",
                      pvalueCutoff = 0.05,
                      pAdjustMethod = "BH",
                      keyType = 'ENSEMBL')

# Plotting gene set enrichment results
dotplot(GO_result, x = "GeneRatio", color = "p.adjust", showCategory = 20, title = "Gene Ontology Dotpl
```

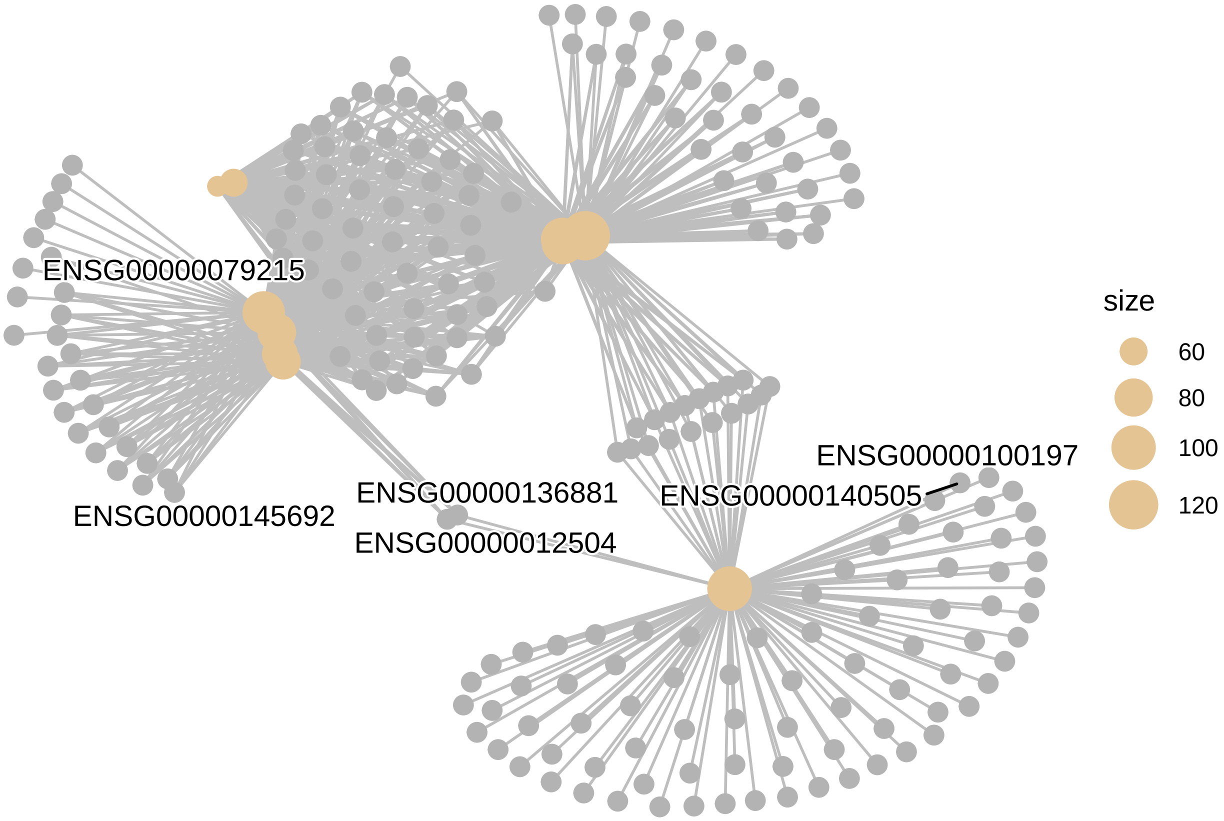


```
barplot(GO_result, x = "GeneRatio", color = "p.adjust", showCategory = 20, title = "Gene Ontology Barpl
```



```r
cnetplot(GO_result, showCategory = 10)
```

```
## Warning: ggrepel: 234 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

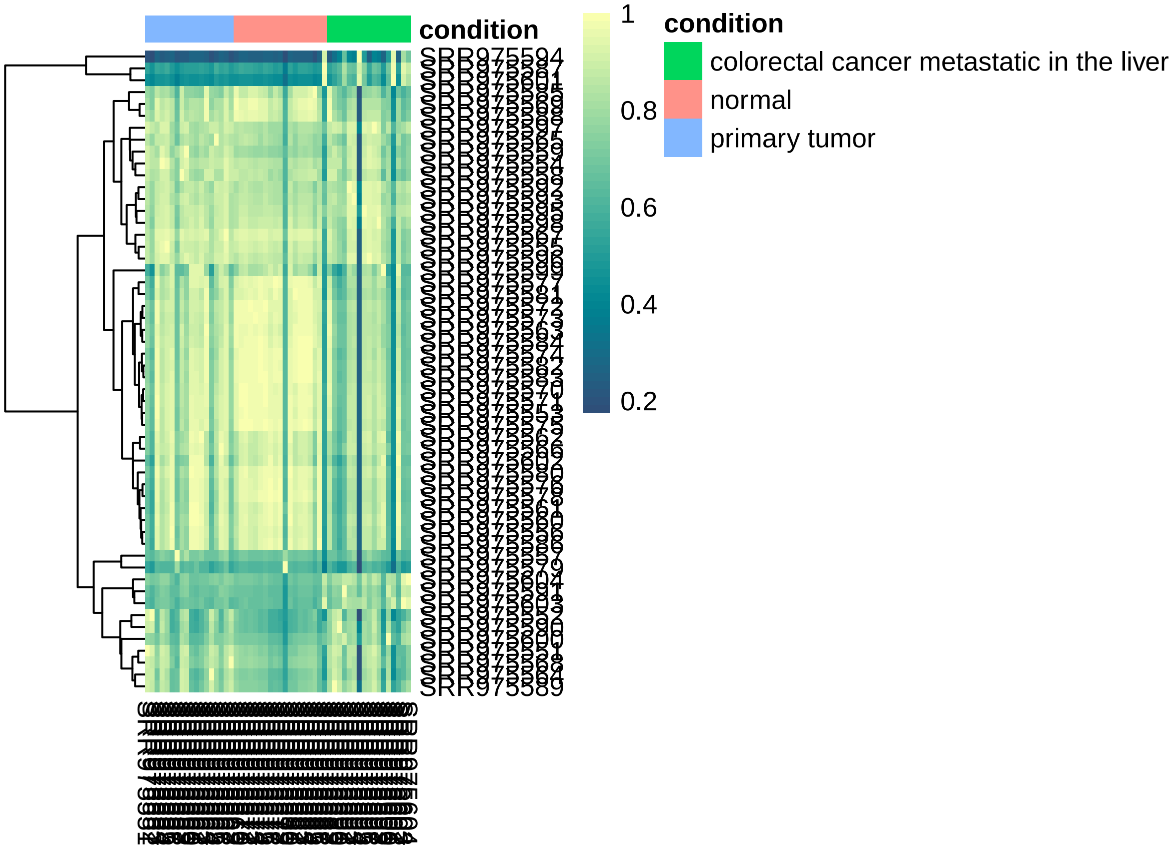


Hierarchical Clustering and Heatmap

Apply hierarchical clustering to genes and generate a heatmap.

```r
# Load pheatmap
library(pheatmap)

# Create heatmap
pheatmap(cor(countData),
         annotation = colData,
         cluster_cols = FALSE,
         color = hcl.colors(50, "BluYl"))  # Adjust color for heatmap
```



```r
library(rmarkdown)
```