Bioinformatics Assignment 2

```r
# Define file paths
data_dir <- file.path("../../data", "SRP164913")
data_file <- file.path(data_dir, "SRP164913_HUGO.tsv")
metadata_file <- file.path(data_dir, "metadata_SRP164913.tsv")
results_dir <- file.path("../../results")
plots_dir <- file.path("plots")

# Libraries
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
library(ggplot2)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:GenomicRanges':
##
##     subtract
```

```r
library(M3C)
library("umap")
```

```
##
## Attaching package: 'umap'

## The following object is masked from 'package:M3C':
##
##     umap
```

```r
# Set seed for reproducible results
set.seed(12345)
```

Group: 5, **Date:** 09/25/2024 Order to run R scripts:
(task 1) preprocessing.R
(task 2) PCA_Plot.R
(task 3) DifferentialAnalysis.R
(task 4) heatmap.R
(task 5-Hannah) gprofiler2.R
(task 5-Luca) topGO.R

Task 1 - Data analysis

Project report on the analysis of our RNA-seq data set on Neurological Immune-mediated Disorders. The data set is composed of 88 total samples out of which 34 are Multiple Sclerosis, 14 are HAM/TSP and 20 are healthy controls. URL: https://www.refine.bio/experiments/SRP164913/comprehensive-analysis-of-tcr-s-repertoire-in-patients-with-neurological-immune-mediated-disorders ### Preprocessing At first we downloaded the quantile normalized version of our data set. After a discussion with our TA we switched to the non-normalized version, since some further analysis will use that Data. The following preprocessing steps have been taken.

1. Map Ensemble to HUGO Id's using the following tutorial: https://alexslemonade.github.io/refinebio-examples/03-rnaseq/gene-id-annotation_rnaseq_01_ensembl.html
2. Some Ensemble ID's did not map to HUGO, these rows have been dropped.
3. Some Ensemble ID's mapped to multiple HUGO ID's we took the first match
4. A few Ensemble ID's mapped to the same HUGO ID's creating duplicate rows. These have been aggregated by the median into single rows.

Of 43'363 Genes in the original data set, 29'589 where used for the analysis. 13'774 rows from the original dataset were dropped or aggregated.

During the Preprocessing steps we also found out that even the non-quartile-normalized dataset from refine.bio was pre normalized. The consequences of this will be apparent in further chapters.

Sample Size

```r
data_analysis_df <- read.delim("../../data/SRP164913/SRP164913_HUGO.tsv",header = TRUE, row.names = 1, s
cat("Number of Genes in the expression matrix: ", dim(data_analysis_df)[1], "\n")
```

```
## Number of Genes in the expression matrix:  29589
```

```r
cat("Number of Samples in the expression matrix: ", dim(data_analysis_df)[2], "\n")
```

```
## Number of Samples in the expression matrix:  88
```

There are 88 samples, however, there are three disease groups: healthy controls, MS subjects, and ham/tsp subjects. To make this a binary problem, from Task 2 onward, we removed the ham/tsp subjects to compare only MS vs healthy controls (hc). This reduces the number of samples to 62.

Density Plot of Gene Expressions

```r
# Get the median of expressions by gene
gene_median <- apply(data_analysis_df, 1, median)
head(gene_median)
```

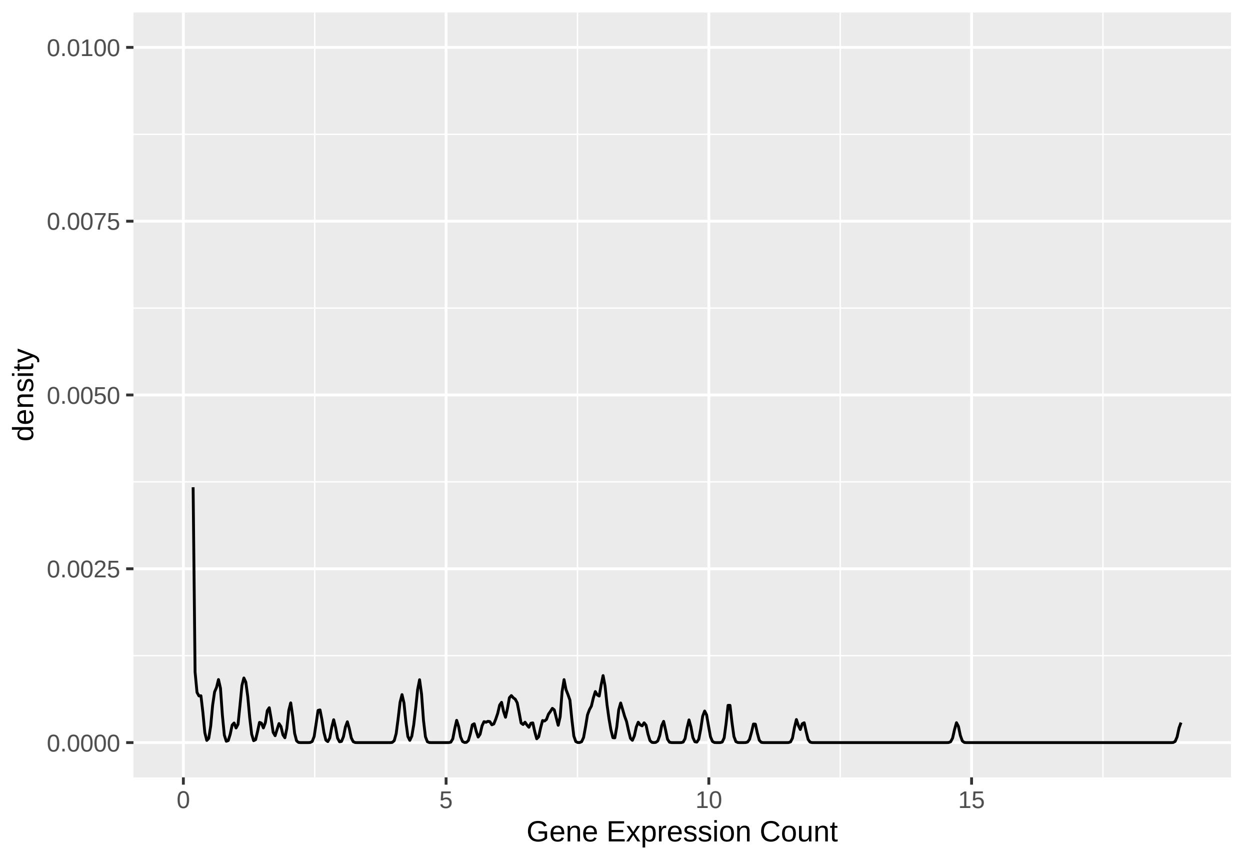
```
##     A1BG A1BG-AS1     A1CF      A2M  A2M-AS1    A2ML1
##        0        0        0        0        0        0
```

```r
gene_median = log2(gene_median +1)
cat("Variance between gene expression medians:", var(gene_median, na.rm = TRUE))
```

```
## Variance between gene expression medians: 0.1234497
```

```r
# Create a Data frame from the numerical array
gene_median_df <- data.frame(Median = gene_median)
# Plot the values
ggplot(gene_median_df, aes(x = Median)) + geom_density() + xlab("Gene Expression Count") + ylim(0,1e-2)
```

```
## Warning: Removed 72 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```



The density plot shows a left skewed distribution of gene expression counts with most of the expressions being between 0 and 10 but with a few outliers between 10 and 15. We downloaded the un-normalized data, however when we opened the raw file, most count values were 0 or very close to 0. We suspect the data came pre-normalized, and this is causing some issues. This carries over through the entire assignment.

Task 2 - Principal Component Analysis

We were able to get a PCA, tsne, and umap plot following the tutorials. All three show a clear difference between the healthy control (hs) and MS (ms) groups, with a few samples in the midst of the other cluster. The number of crossovers is different.

PCA Plot

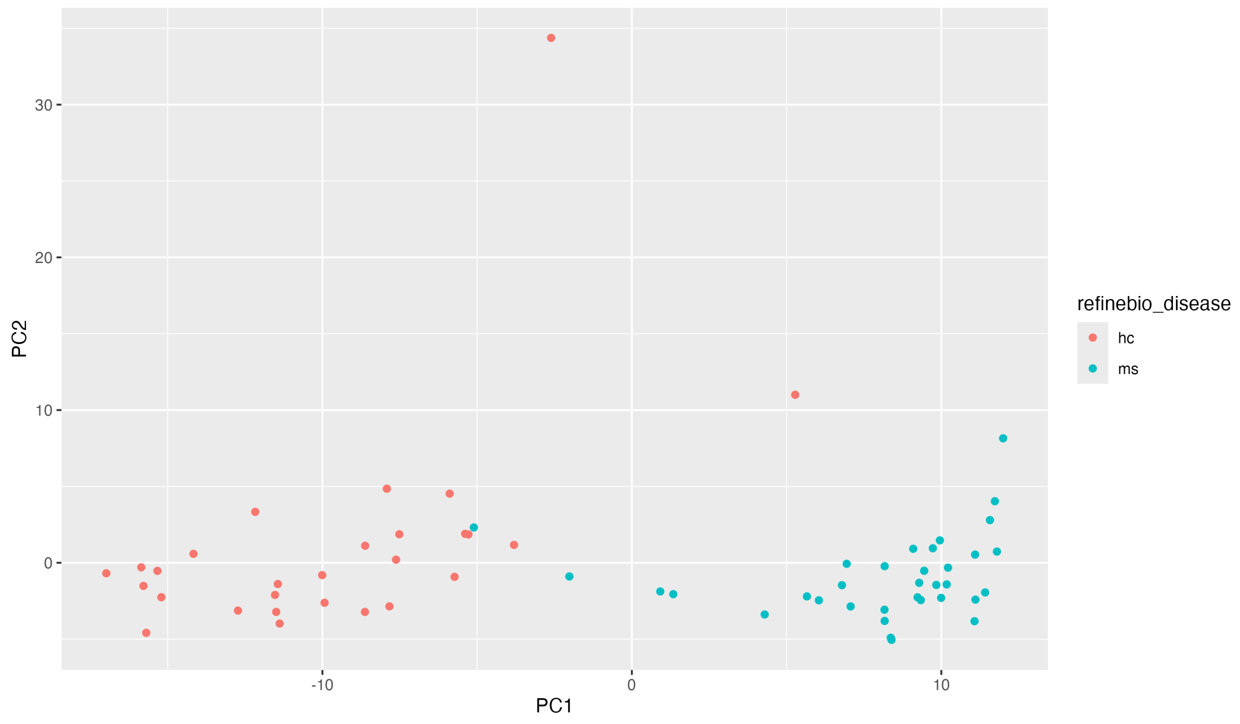


Figure 1: PCA Plot

TSNE Plot

The TSNE (t-distributed stochastic neighbor embedding) plot visualizes n-dimensional data in two dimensions. Using this method, our data is nicely split in two clusters, with a few outliers in the wrong cluster. This gives a first hint that there are some differentially expressed genes among the two groups, Multiple Sclerosis and Human Control.

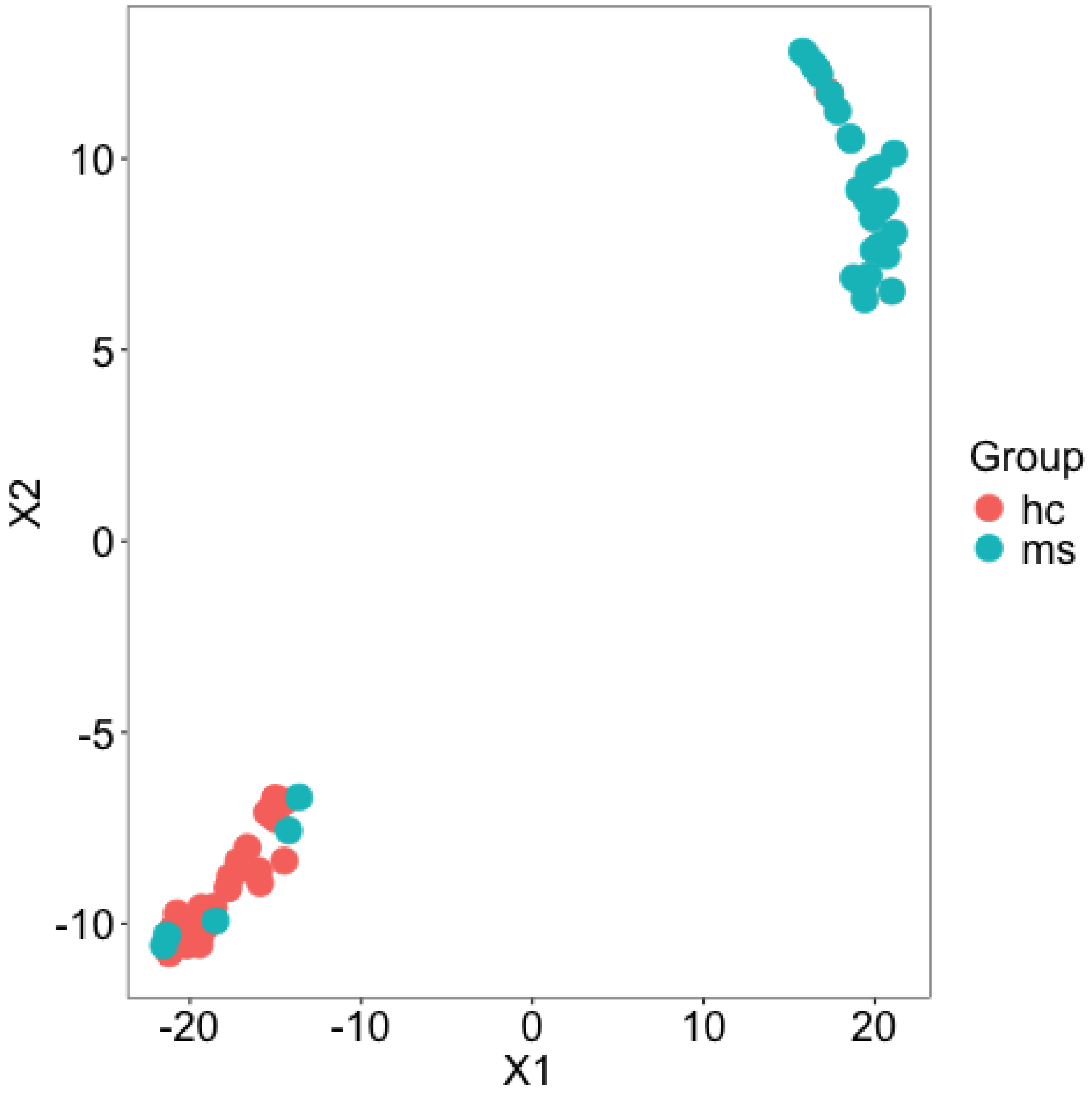


Figure 2: tsne Plot

UMAP Plot

Similar to t-SNE UMAP also reduces n-dimensional data to two dimensions. In our case it yields similar results to the t-SNE plot with slightly less outliers.

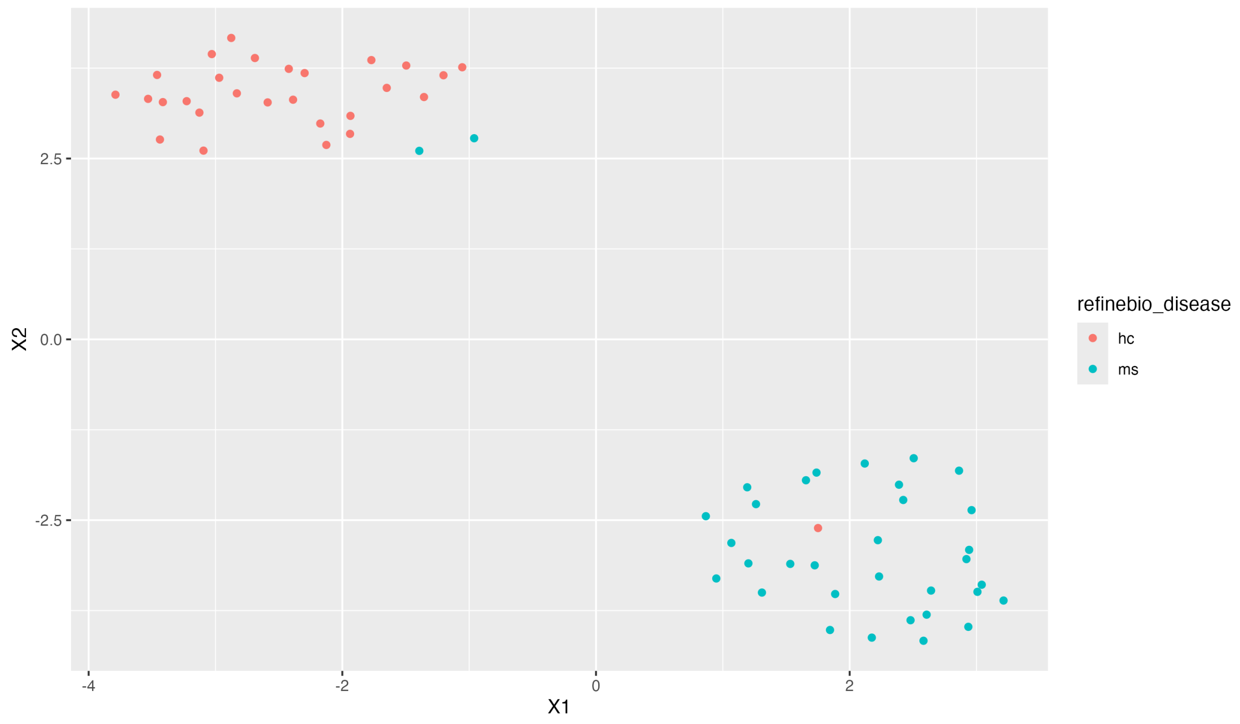


Figure 3: umap Plot

Task 3 - Differential Analysis

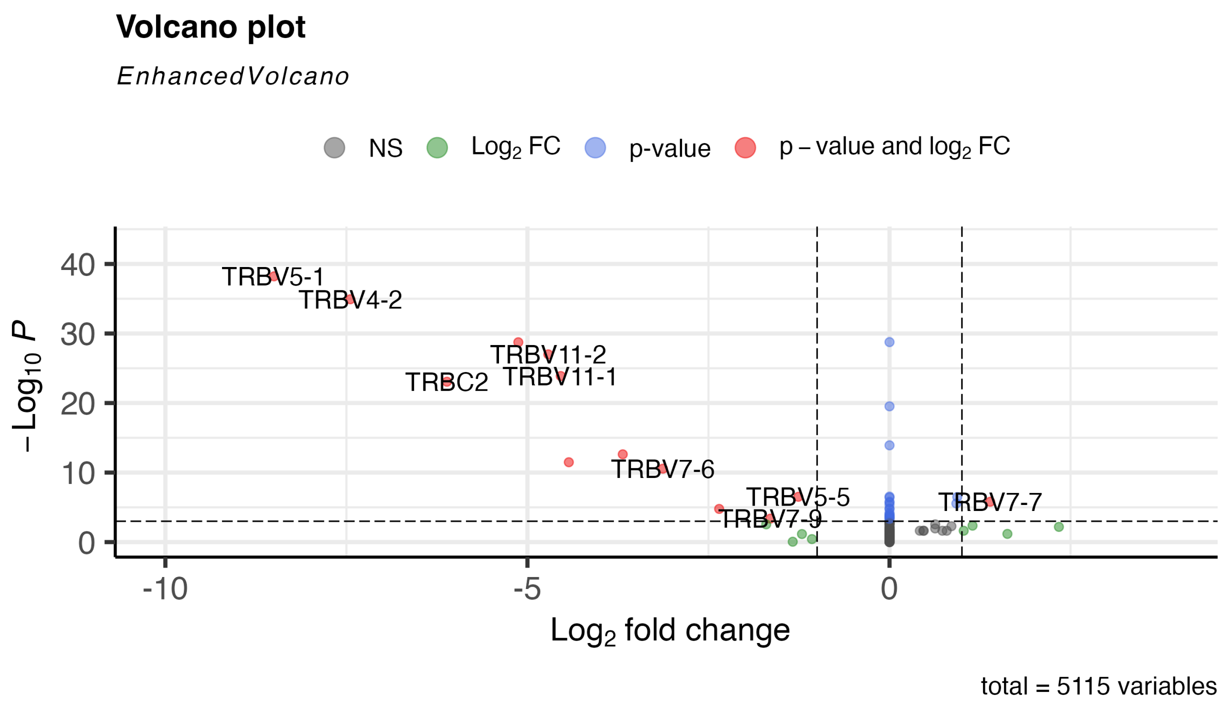
When we did our differential analysis, because the data was so small and normalized, we had to adjust the filter value to 0.01 or else only ~1000 rows were making it to the DESeqDataSetFromMatrix() step. This was causing our volcano plot to look very sparse. We were able to get a list of the top 50 differentially expressed genes. The results, both the entire table and only the top 50, are stored in the results folder.

```r
top_50_file <- file.path(results_dir, "SRP164913_diff_expr_top_50_results.tsv")
top_50 <- readr::read_tsv(top_50_file)
```

```
## Rows: 50 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): Gene
## dbl (5): baseMean, log2FoldChange, lfcSE, pvalue, padj
## lgl (1): threshold
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(top_50)
```

```
## # A tibble: 6 x 7
##   Gene      baseMean log2FoldChange lfcSE        pvalue        padj threshold
##   <chr>        <dbl>          <dbl> <dbl>         <dbl>       <dbl> <lgl>
## 1 TRBV23-1      9.45          2.34  0.614 0.000168      0.00652     TRUE
## 2 TRBV16       18.9           1.63  0.424 0.0633        0.0663      FALSE
## 3 TRBV7-7     557.            1.39  0.269 0.0000000137  0.00000158  TRUE
## 4 TRBV7-4      31.7           1.15  0.362 0.000108      0.00435     TRUE
## 5 TRBV12-4    389.            1.02  0.509 0.000837      0.0234      TRUE
## 6 TRBV15      729.            0.936 0.172 0.00000000247 0.000000326 TRUE
```



The volcano plot, while still somewhat sparse, is far more populated than it was with the highly filtered data. Most significant results have a negative log fold change.

Task 4 - Heatmap

A heatmap was generated for the top 50 differentially expressed genes. The counts for each gene in the top 50 were extracted and put into the heatmap plot. It was annotated with the two disease groups: ms subjects and healthy control subjects. Since many of our counts are 0 and therefore are not expressed differently, the map is largely blue.

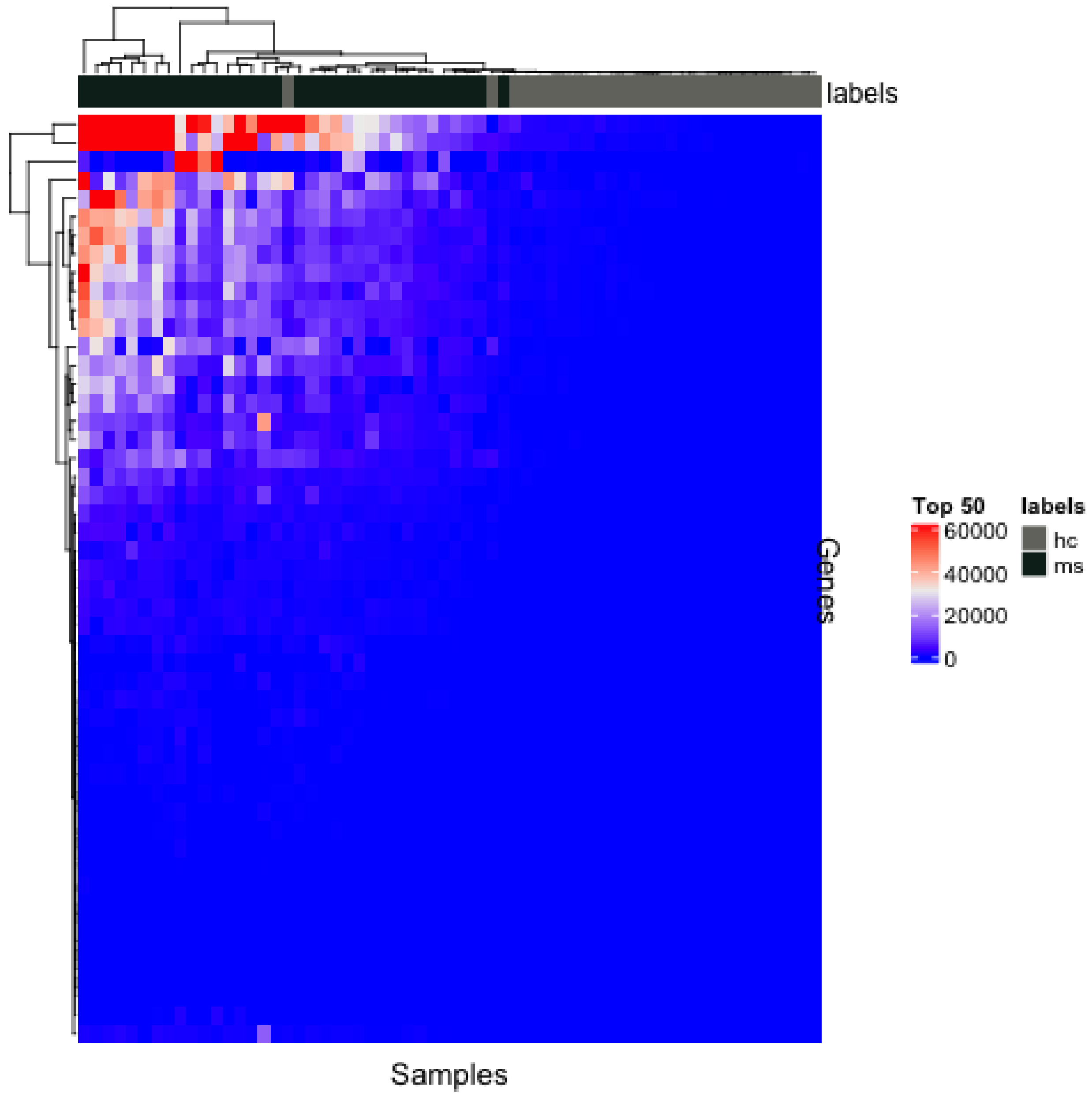


Figure 4: Heatmap

Task 5 - Enrichment Analysis

Hannah- gprofiler2

The function gost was used from gprofiler2 to perform enrichment analysis on the differentially expressed data. Disease ontology was used, as the data was differentially expressed between the MS subjects and healthy controls. Only genes with adjusted p-values below 0.05 were put into the function. Two gost results were generated: one that contains all results, significant or not, and one that contained only significant results. The plot (created with gostplot) for the significant results is shown below. It seems there are a few points above the significance threshold.

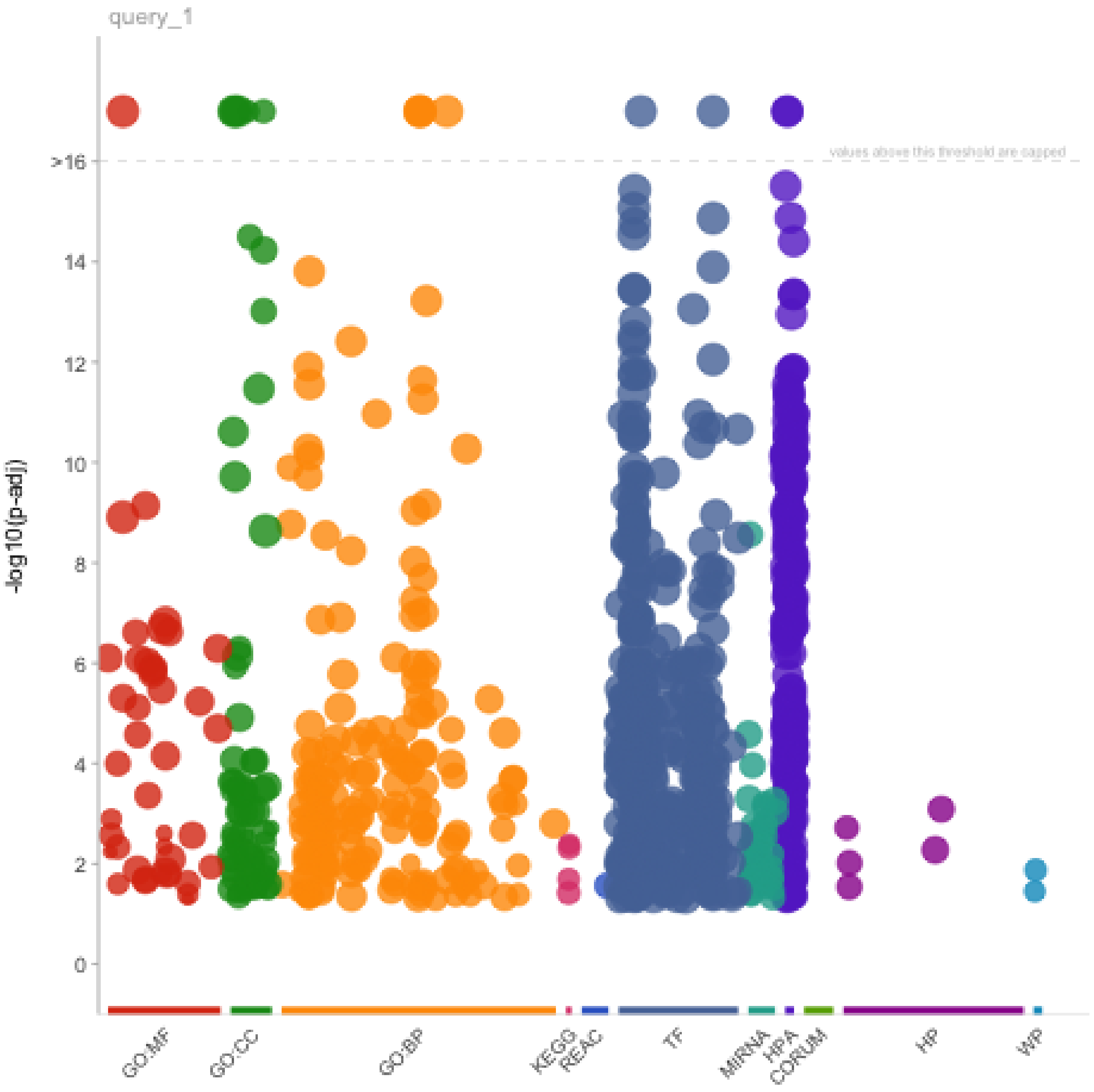


Figure 5: gprofiler2 Plot

However, when publish_gosttable(gost_res3) was run, the function would not complete. It would continue to run for over 15 minutes and never produce results. I was unable to generate a table from the results, and when I tried to print the results, it would return NULL for both significant and non-significant gost results. The NULL result is shown below along with the gost function that was run. It is possible that no truly significant results were generated and so the results were empty.

```r
results_dir <- file.path("../../results")
diff_expr_file <- file.path(results_dir, "SRP164913_diff_expr_results.tsv")
library(gprofiler2)
diff_expr_df <- readr::read_tsv(diff_expr_file)
```

```
## Rows: 5115 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): Gene
## dbl (5): baseMean, log2FoldChange, lfcSE, pvalue, padj
## lgl (1): threshold
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
gost_res3 <- gost(
  query = unlist(diff_expr_df[diff_expr_df$padj<0.05, 'Gene']),
  significant=TRUE,
  organism = "hsapiens"
  )
head(gost_res3$results)
```

```
## NULL
```

When I had initially tried to run gost with every gene name, no p-value factor, it would produce a result that could be printed, but that is not as relevant to our dataset. This does prove that something is off with specifically the differential data, and the function does work.

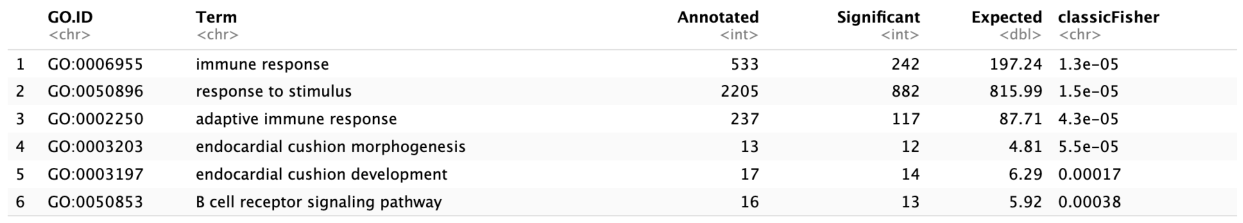
Unfortunatly, due to the circumstances described above, no table was generated using the gprofiler2 method.

Luca - topGO

Given a set of genes and a gene differential expression analysis, topGo enriches the most relevant Genes with Gene Ontology (GO) terms. This helps to find out what function the differential expressed genes perform in the Human Body.

Methodology The Mothodology described is analogous to the Quick Start Guide found here "https://bioconductor.org/packages/release/bioc/vignettes/topGO/inst/doc/topGO.pdf". The Output of Task 1-4 was a list of differential expressed genes, with their probability, only trivial transformations were required to input the data into the topGO function which returns an object containing all gene identifiers and their scores, GO annotations and further information. The cutoff for expressed gene selection has been set to the default of $p < 0.05$.

Using the Fisher exact test a enrichment analysis is performed, testing for the over-representation of GO terms within the differentially expressed genes. Due to time constraints only the classic method, testing each GO category independently was used.



	GO.ID <chr>	Term <chr>	Annotated <int>	Significant <int>	Expected <dbl>	classicFisher <chr>
1	GO:0006955	immune response	533	242	197.24	1.3e−05
2	GO:0050896	response to stimulus	2205	882	815.99	1.5e−05
3	GO:0002250	adaptive immune response	237	117	87.71	4.3e−05
4	GO:0003203	endocardial cushion morphogenesis	13	12	4.81	5.5e−05
5	GO:0003197	endocardial cushion development	17	14	6.29	0.00017
6	GO:0050853	B cell receptor signaling pathway	16	13	5.92	0.00038

Figure 6: Top 5 GO Terms

Results: The Table above shows the first few rows of the enrichment analysis. Although the issues with the pre-normalized data mentioned in the introduction, parts of the result seem plausible to the untrained eye in regard to the disease Multiple Sclerosis (this is the first time I actually experienced joy during this assignment). In multiple sclerosis MS, the immune system attacks the protective cover on nerve, causing communication problems between the brain and the rest of the body. Due to this pathology it is coherent, that the top differential expressed genes found during the enrichment analysis, are genes and receptors linked to the immune system.

Below the most significant 5 nodes are represented as a tree.

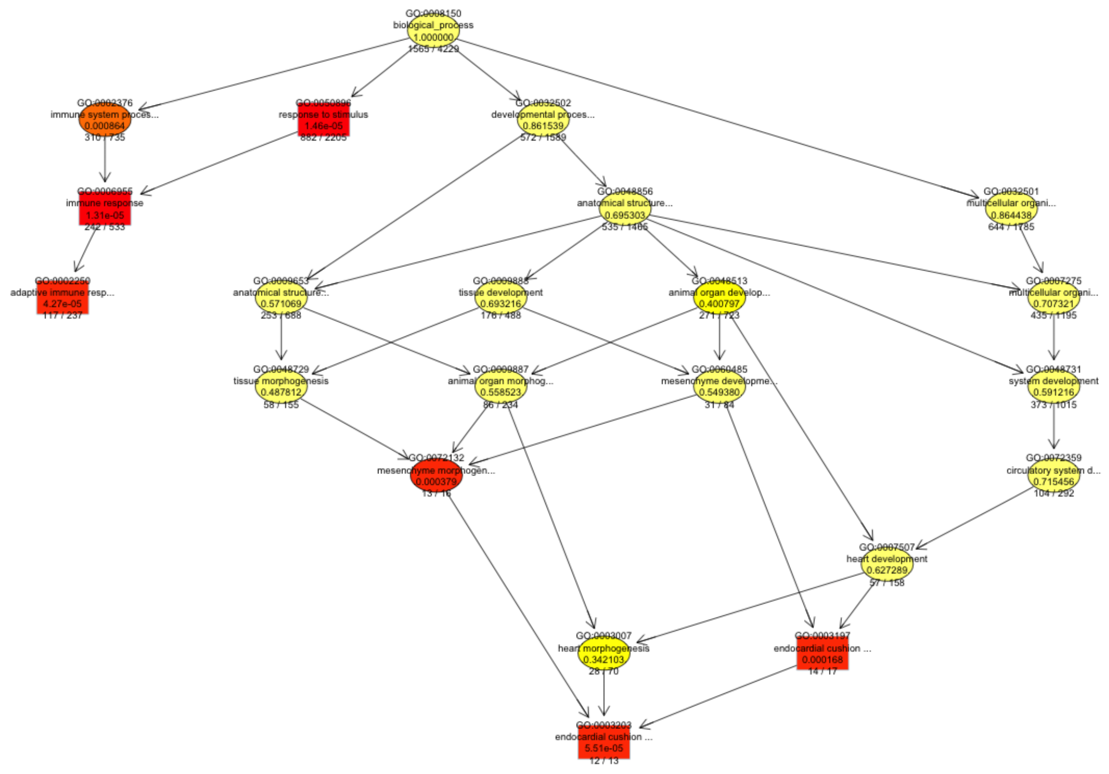


Figure 7: Gene Ontology Tree

Fisher vs Elim Due to time constraint, only Fisher's exact test was used to generate the above results.

To visualize the potential differences the following plot, shows that the Elim test would yield higher p-values for multiple genes, while some are identical for both methods.

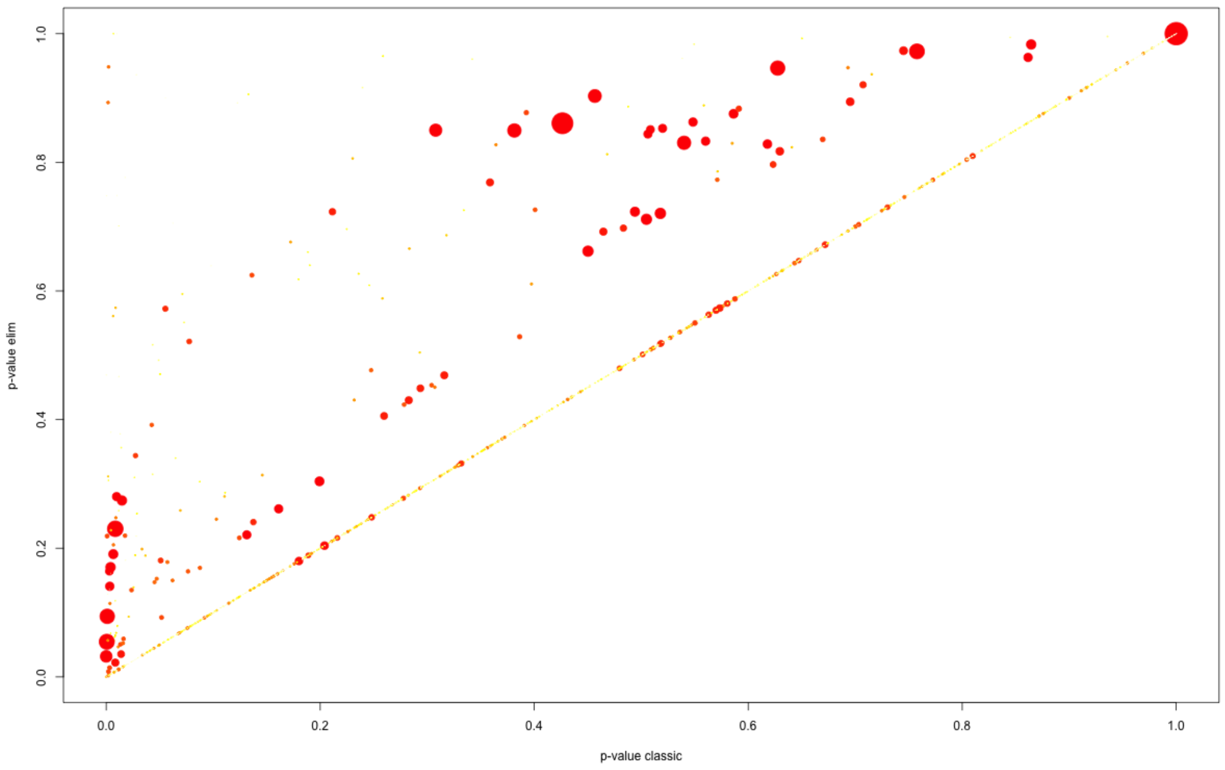


Figure 8: Fisher vs Elim

Peter- ClusterProfiler

Cluster Profiler is designed for the analysis and visualization of functional profiles of genes and gene clusters. It helps researchers identify biological themes, functions, and pathways that are overrepresented in gene sets. In this specific example, it was used to identify the enriched biological processes, and molecular functions associated with the gene set.

Here is the plot I was able to generate after using the enrichGO function to find overrepresentation.

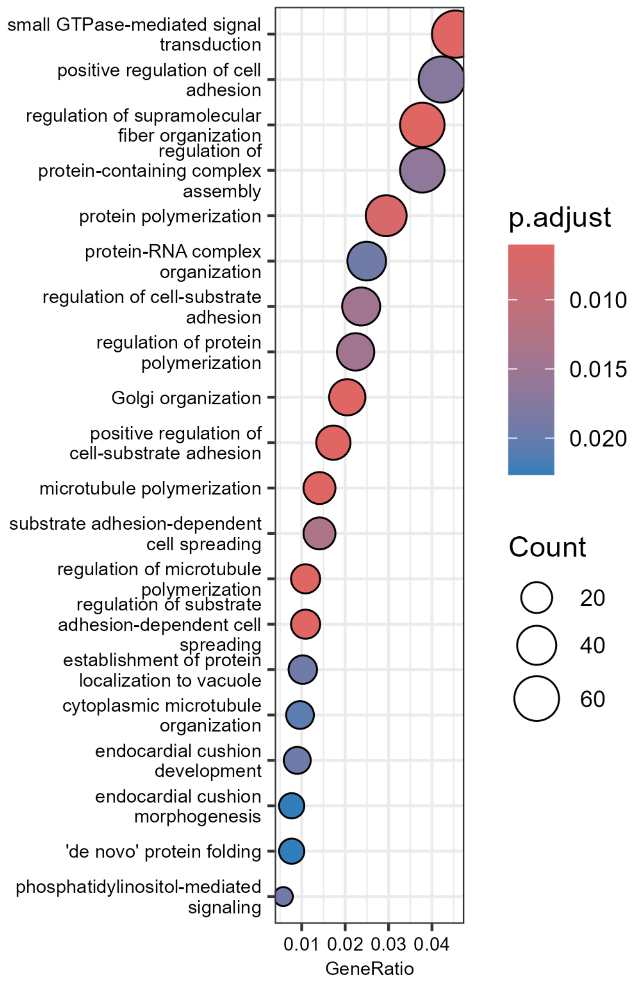


Figure 9: ClusterProfiler Plot

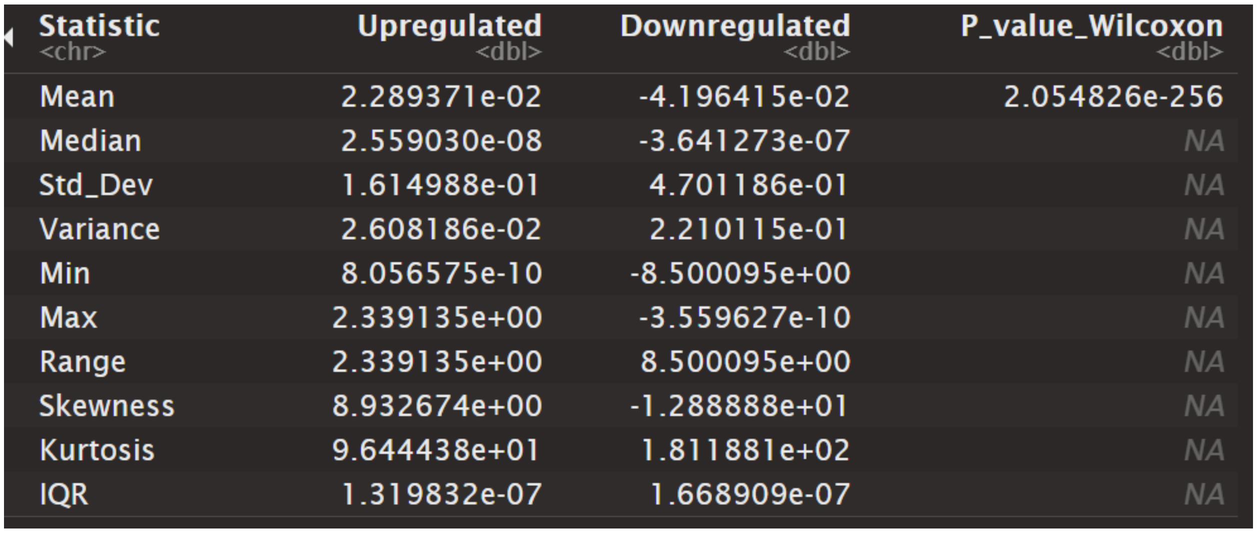
As shown, the top result was small GTPase-mediated signal transduction, which I found to play a significant role in signal transduction pathways that regulate various cellular functions, including cell growth and immune responses. Small GTPases like Rho, Rac, and Cdc42 are vital for activating and migrating immune cells. In MS, these cells lead to neuroinflammation and tissue damage.They also regulate cytokine production in immune and glial cells, with dysregulation potentially worsening MS-related inflammation.

Esha - Wilcoxon Rank-Sum Test

To understand the Wilcoxon Rank-Sum Test, the University of Auckland's Department of Statistics' PDF was utilized (https://www.stat.auckland.ac.nz/~wild/ChanceEnc/Ch10.wilcoxon.pdf). Furthermore, the

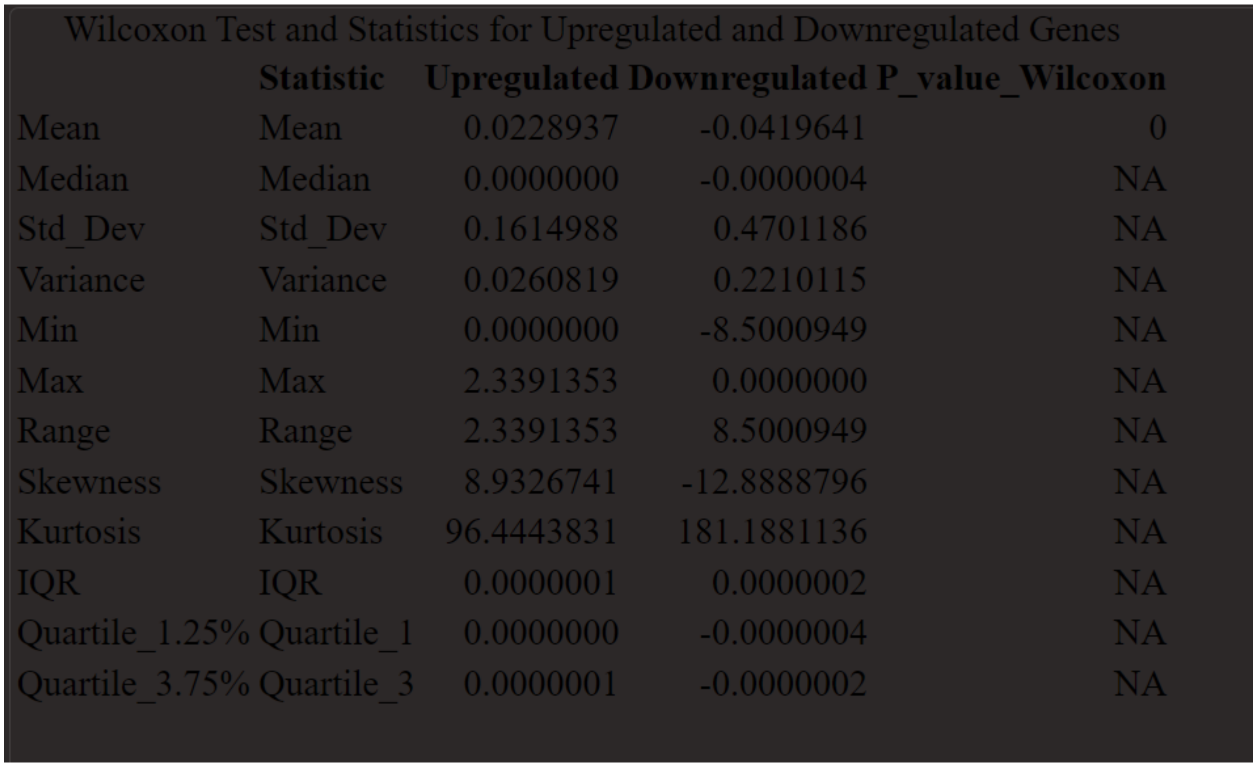
Wilcoxon Rank-Sum Test CSV is located in the results folder within the Wilcoxon's respective folder.

Furthermore, in order to understand different functions and libraries in R, I utilized these citations per this document (https://docs.google.com/document/d/1WnJGxTHmfK8Lc3xXH0t1TL96tsdu-CYl5K2yj2KjyU4/edit?usp=sharing)



Statistic <chr>	Upregulated <dbl>	Downregulated <dbl>	P_value_Wilcoxon <dbl>
Mean	2.289371e-02	-4.196415e-02	2.054826e-256
Median	2.559030e-08	-3.641273e-07	NA
Std_Dev	1.614988e-01	4.701186e-01	NA
Variance	2.608186e-02	2.210115e-01	NA
Min	8.056575e-10	-8.500095e+00	NA
Max	2.339135e+00	-3.559627e-10	NA
Range	2.339135e+00	8.500095e+00	NA
Skewness	8.932674e+00	-1.288888e+01	NA
Kurtosis	9.644438e+01	1.811881e+02	NA
IQR	1.319832e-07	1.668909e-07	NA

Figure 10: Additional Statistics



Wilcoxon Test and Statistics for Upregulated and Downregulated Genes

	Statistic	Upregulated	Downregulated	P_value_Wilcoxon
Mean	Mean	0.0228937	-0.0419641	0
Median	Median	0.0000000	-0.0000004	NA
Std_Dev	Std_Dev	0.1614988	0.4701186	NA
Variance	Variance	0.0260819	0.2210115	NA
Min	Min	0.0000000	-8.5000949	NA
Max	Max	2.3391353	0.0000000	NA
Range	Range	2.3391353	8.5000949	NA
Skewness	Skewness	8.9326741	-12.8888796	NA
Kurtosis	Kurtosis	96.4443831	181.1881136	NA
IQR	IQR	0.0000001	0.0000002	NA
Quartile_1.25%	Quartile_1	0.0000000	-0.0000004	NA
Quartile_3.75%	Quartile_3	0.0000001	-0.0000002	NA

Figure 11: Additional Statistics

These are additional statistics that were ran. Note, the P-Value does not display for all of the statistics since it is the same for all of them.

One of the plots that I created shows the 10 Gene Ontology concepts in relation to biology. The X-axis shows the gene count that corrrelates with each of the biology process words. So, since the bar is longer, that means that the genes play a bigger role in that specific process. The Y-axis represents the enrichment. The colors showcase the P-values (this uses the Benjamini-Hochberg Methodology) since the red indicates smaller P-values and blue means that the P-values are higher. The words that are the most "shown" means that their presence is incredibly signficant in the dataset.

The second plot is an extension of the first plot. For this, Gene Ratio (https://www.spandidos-publications.com/10.3892/etm.2018.6884#:~:text='Gene%20ratio'%20is%20the%20percentage,function%3B%20CC%2C%20cellular%20c was utilized. Gene ratio, as defined by Spandidos Publications is defined as "the percentage of total DEG's in the given GO term". The X-axis is the Gene Ratio in this plot, and the Y-axis are the GO terms.

Justin - Genomic Super Signature

Justin was unable to complete his portion of the assignment.

Task 6/7 - Joint Enrichment Analysis Tables

We had 3 successful gene enrichment tables created. The top ten results of each analysis are shown below.

```r
#combining results
results_dir <- file.path("../../results")
topgo_results <- file.path(results_dir, "SRP164913_topGO_results.tsv")
clusterprofiler_results <- file.path(results_dir, "clusterprofiler_results.tsv")
```

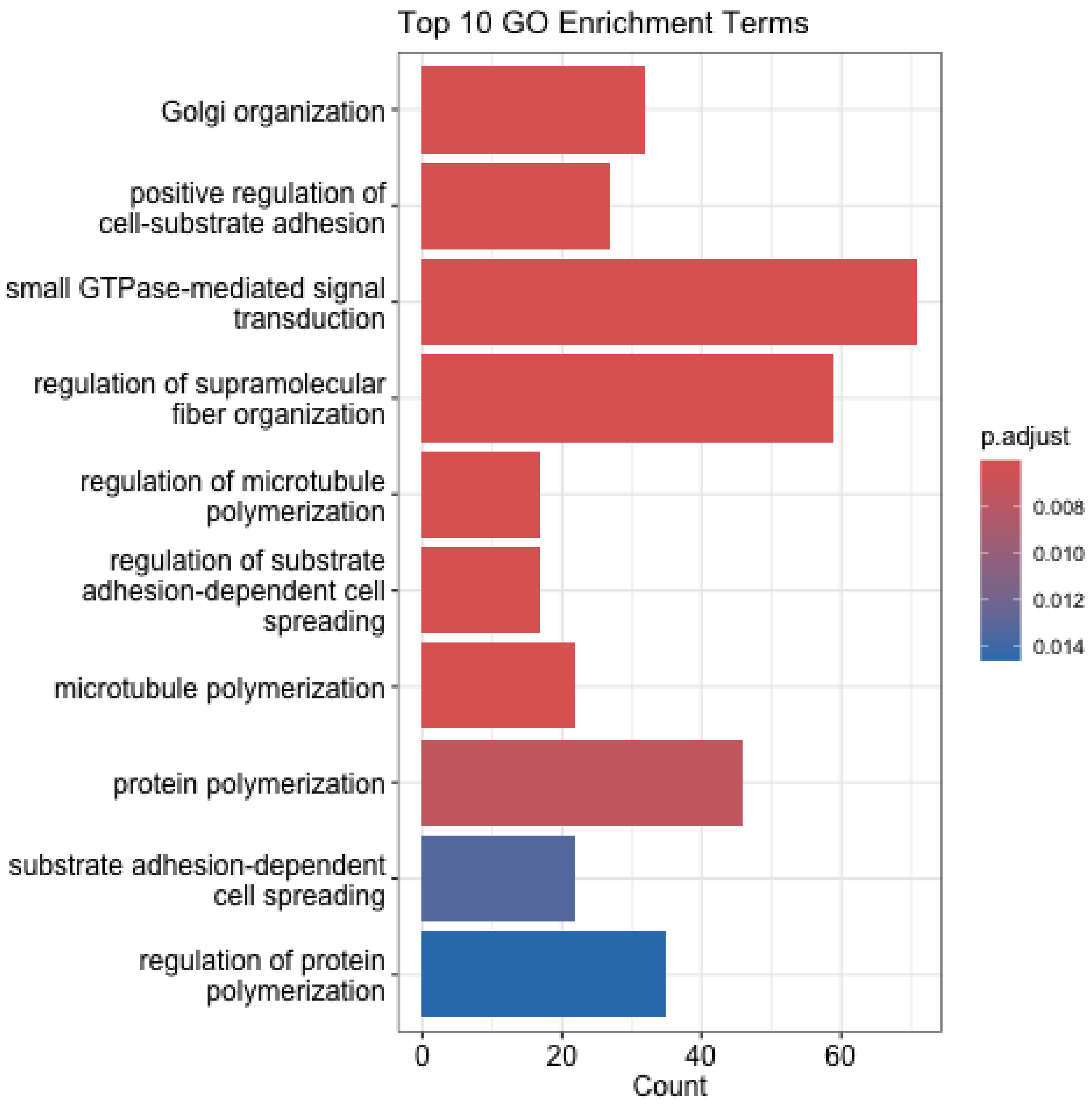


Figure 12: GO Enrichment Bar Graph

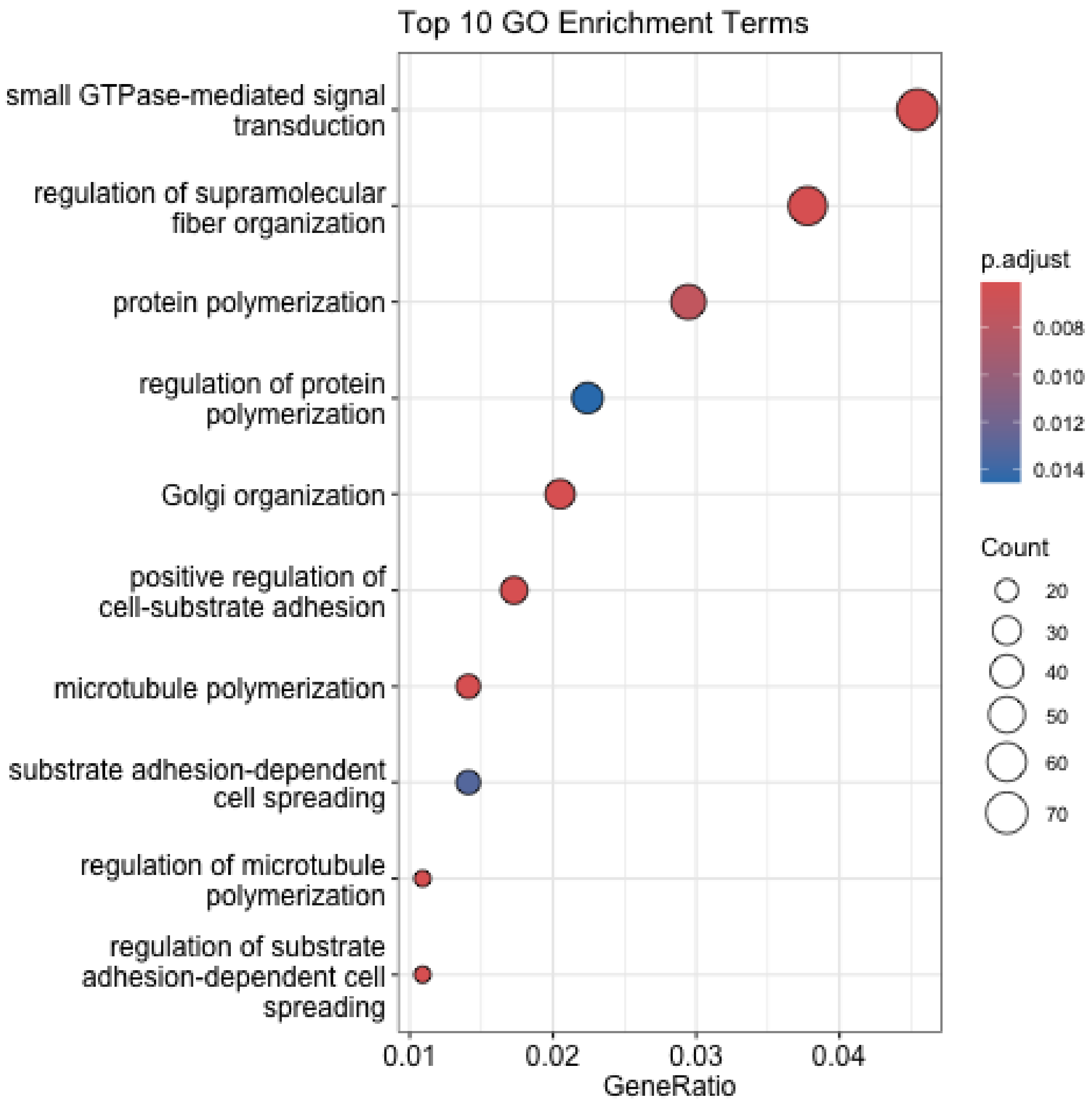


Figure 13: GO Enrichment Dot Plot

```r
wilcoxon_results <- file.path(results_dir, "Wilcoxon_GeneOnt/Enrichment.csv")

#read in files
topgo_df <- readr::read_tsv(topgo_results)
```

```
## Rows: 50 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): GO.ID, Term
## dbl (4): Annotated, Significant, Expected, classicFisher
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
clusterprofiler_df <- readr::read_tsv(clusterprofiler_results)
```

```
## Rows: 90 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (6): ONTOLOGY, ID, Description, GeneRatio, BgRatio, geneID
## dbl (7): RichFactor, FoldEnrichment, zScore, pvalue, p.adjust, qvalue, Count
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
wilcoxon_df <- readr::read_csv(wilcoxon_results)
```

```
## Rows: 5638 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): ID, Description, GeneRatio, BgRatio, geneID
## dbl (7): RichFactor, FoldEnrichment, zScore, pvalue, p.adjust, qvalue, Count
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(topgo_df, 10)
```

```
## # A tibble: 10 x 6
##    GO.ID      Term                   Annotated Significant Expected classicFisher
##    <chr>      <chr>                      <dbl>       <dbl>    <dbl>         <dbl>
##  1 GO:0006955 immune response              533         242   197.        0.000013
##  2 GO:0050896 response to stimulus        2205         882   816.        0.000015
##  3 GO:0002250 adaptive immune resp~        237         117    87.7       0.000043
##  4 GO:0003203 endocardial cushion ~         13          12     4.81      0.000055
##  5 GO:0003197 endocardial cushion ~         17          14     6.29      0.00017
##  6 GO:0050853 B cell receptor sign~         16          13     5.92      0.00038
##  7 GO:0072132 mesenchyme morphogen~         16          13     5.92      0.00038
##  8 GO:0050789 regulation of biolog~       2847        1102  1054.        0.00055
##  9 GO:0010811 positive regulation ~         44          27    16.3       0.00084
## 10 GO:0002376 immune system process        735         310   272         0.00086
```

```r
clusterprofiler_sorted <- clusterprofiler_df[order(clusterprofiler_df$p.adjust,
                                    decreasing = FALSE), ]
head(clusterprofiler_sorted, 10)
```

```
## # A tibble: 10 x 13
##    ONTOLOGY ID    Description GeneRatio BgRatio RichFactor FoldEnrichment zScore
##    <chr>    <chr> <chr>       <chr>     <chr>        <dbl>          <dbl>  <dbl>
##  1 CC       GO:0~ T cell rec~ 70/1624   150/19~      0.467           5.72  17.3
##  2 CC       GO:0~ plasma mem~ 80/1624   325/19~      0.246           3.02  10.9
##  3 CC       GO:0~ cell-subst~ 71/1624   431/19~      0.165           2.02   6.37
##  4 CC       GO:0~ focal adhe~ 69/1624   421/19~      0.164           2.01   6.23
##  5 CC       GO:0~ ruffle      32/1624   182/19~      0.176           2.15   4.66
##  6 CC       GO:0~ ficolin-1-~ 32/1624   185/19~      0.173           2.12   4.56
##  7 CC       GO:0~ cell leadi~ 59/1624   426/19~      0.138           1.70   4.33
##  8 CC       GO:1~ microtubul~ 11/1624   34/198~      0.324           3.96   5.16
##  9 CC       GO:0~ microtubul~ 9/1624    24/198~      0.375           4.59   5.25
## 10 CC       GO:0~ lysosomal ~ 60/1624   443/19~      0.135           1.66   4.18
## # i 5 more variables: pvalue <dbl>, p.adjust <dbl>, qvalue <dbl>, geneID <chr>,
## #   Count <dbl>
```

```r
wilcoxon_sorted <- wilcoxon_df[order(clusterprofiler_df$p.adjust,
                                     decreasing = FALSE), ]
head(wilcoxon_df, 10)
```

```
## # A tibble: 10 x 12
##    ID     Description GeneRatio BgRatio RichFactor FoldEnrichment zScore  pvalue
##    <chr>  <chr>       <chr>     <chr>        <dbl>          <dbl>  <dbl>   <dbl>
##  1 GO:00~ Golgi orga~ 32/1562   156/18~      0.205           2.48   5.57 1.28e-6
##  2 GO:00~ positive r~ 27/1562   124/18~      0.218           2.63   5.48 2.58e-6
##  3 GO:00~ small GTPa~ 71/1562   500/18~      0.142           1.72   4.88 4.76e-6
##  4 GO:19~ regulation~ 59/1562   392/18~      0.151           1.82   4.93 4.98e-6
##  5 GO:00~ regulation~ 17/1562   61/188~      0.279           3.37   5.57 5.79e-6
##  6 GO:19~ regulation~ 17/1562   62/188~      0.274           3.32   5.48 7.37e-6
##  7 GO:00~ microtubul~ 22/1562   95/188~      0.232           2.80   5.28 7.50e-6
##  8 GO:00~ protein po~ 46/1562   287/18~      0.160           1.94   4.81 1.05e-5
##  9 GO:00~ substrate ~ 22/1562   101/18~      0.218           2.63   4.94 2.11e-5
## 10 GO:00~ regulation~ 35/1562   204/18~      0.172           2.07   4.63 2.74e-5
## # i 4 more variables: p.adjust <dbl>, qvalue <dbl>, geneID <chr>, Count <dbl>
```