Final Project

1. Install Bioconductor packages

https://www.bioconductor.org/install/

In [1]:
```
# note that 3.18 is not the latest Bioconductor release
# But we are running R 4.3.3 on UW-IT JupyterHub
# Installing these packages will take a few minutes. Make sure you include the inst
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install(version = "3.18")
BiocManager::install(c("edgeR", "ggplot2", "DESeq2", "rtracklayer", "GenomicRanges"
library(ggplot2)
library(edgeR)
library (DESeq2)
library(rtracklayer)
library(GenomicRanges)
```

```
Updating HTML index of packages in '.Library'

Making 'packages.html' ...
 done

'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.18 (BiocManager 1.30.25), R 4.3.3 (2024-02-29)

Installing package(s) 'BiocVersion'

Updating HTML index of packages in '.Library'

Making 'packages.html' ...
 done

Old packages: 'askpass', 'backports', 'bit', 'bit64', 'bitops', 'broom',
  'bslib', 'caret', 'class', 'cli', 'clock', 'colorspace', 'commonmark',
  'cpp11', 'crayon', 'credentials', 'curl', 'data.table', 'DBI', 'dials',
  'downlit', 'e1071', 'evaluate', 'fontawesome', 'forecast', 'fs', 'future',
  'future.apply', 'gert', 'gower', 'gtable', 'hardhat', 'hexbin', 'highr',
  'httr2', 'ipred', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lava', 'lhs',
  'lubridate', 'modeldata', 'modelenv', 'nlme', 'nnet', 'openssl',
  'parallelly', 'parsnip', 'patchwork', 'pbdZMQ', 'pillar', 'pkgbuild',
  'pkgdown', 'pkgload', 'processx', 'prodlim', 'profvis', 'progressr',
  'promises', 'ps', 'purrr', 'R6', 'ragg', 'randomForest', 'Rcpp',
  'RcppArmadillo', 'RCurl', 'readxl', 'recipes', 'reprex', 'rlang',
  'rmarkdown', 'RODBC', 'roxygen2', 'rpart', 'RSQLite', 'rstudioapi',
  'sessioninfo', 'shiny', 'slider', 'survival', 'sys', 'systemfonts',
  'testthat', 'textshaping', 'tidymodels', 'timeDate', 'tinytex', 'tseries',
  'tune', 'urca', 'usethis', 'uuid', 'waldo', 'withr', 'workflows', 'xfun',
  'xml2', 'xts', 'yaml', 'yardstick', 'zip', 'zoo'

'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.18 (BiocManager 1.30.25), R 4.3.3 (2024-02-29)

Warning message:
“package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'ggplot2'”
Installing package(s) 'edgeR', 'DESeq2', 'rtracklayer', 'GenomicRanges'

also installing the dependencies ‘formatR’, ‘abind’, ‘SparseArray’, ‘lambda.r’, ‘fut
ile.options’, ‘statmod’, ‘S4Arrays’, ‘DelayedArray’, ‘futile.logger’, ‘snow’, ‘BH’,
‘GenomeInfoDbData’, ‘Rhtslib’, ‘rjson’, ‘limma’, ‘locfit’, ‘S4Vectors’, ‘IRanges’,
‘SummarizedExperiment’, ‘BiocGenerics’, ‘Biobase’, ‘BiocParallel’, ‘matrixStats’, ‘M
atrixGenerics’, ‘XML’, ‘XVector’, ‘GenomeInfoDb’, ‘Biostrings’, ‘zlibbioc’, ‘Rsamtoo
ls’, ‘GenomicAlignments’, ‘BiocIO’, ‘restfulr’
```

```
Updating HTML index of packages in '.Library'

Making 'packages.html' ...
 done

Old packages: 'askpass', 'backports', 'bit', 'bit64', 'bitops', 'broom',
  'bslib', 'caret', 'class', 'cli', 'clock', 'colorspace', 'commonmark',
  'cpp11', 'crayon', 'credentials', 'curl', 'data.table', 'DBI', 'dials',
  'downlit', 'e1071', 'evaluate', 'fontawesome', 'forecast', 'fs', 'future',
  'future.apply', 'gert', 'gower', 'gtable', 'hardhat', 'hexbin', 'highr',
  'httr2', 'ipred', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lava', 'lhs',
  'lubridate', 'modeldata', 'modelenv', 'nlme', 'nnet', 'openssl',
  'parallelly', 'parsnip', 'patchwork', 'pbdZMQ', 'pillar', 'pkgbuild',
  'pkgdown', 'pkgload', 'processx', 'prodlim', 'profvis', 'progressr',
  'promises', 'ps', 'purrr', 'R6', 'ragg', 'randomForest', 'Rcpp',
  'RcppArmadillo', 'RCurl', 'readxl', 'recipes', 'reprex', 'rlang',
  'rmarkdown', 'RODBC', 'roxygen2', 'rpart', 'RSQLite', 'rstudioapi',
  'sessioninfo', 'shiny', 'slider', 'survival', 'sys', 'systemfonts',
  'testthat', 'textshaping', 'tidymodels', 'timeDate', 'tinytex', 'tseries',
  'tune', 'urca', 'usethis', 'uuid', 'waldo', 'withr', 'workflows', 'xfun',
  'xml2', 'xts', 'yaml', 'yardstick', 'zip', 'zoo'

Loading required package: limma

Loading required package: S4Vectors

Loading required package: stats4

Loading required package: BiocGenerics


Attaching package: ‘BiocGenerics’


The following object is masked from ‘package:limma’:

    plotMA


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, sort,
    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
```

In [2]:
```
# install additional Bioconductor packages
BiocManager::install(c('cluster', 'gplots'))
```

```
'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.18 (BiocManager 1.30.25), R 4.3.3 (2024-02-29)

Installing package(s) 'cluster', 'gplots'

also installing the dependencies ‘gtools’, ‘caTools’


Updating HTML index of packages in '.Library'

Making 'packages.html' ...
 done

Old packages: 'askpass', 'backports', 'bit', 'bit64', 'bitops', 'broom',
  'bslib', 'caret', 'class', 'cli', 'clock', 'colorspace', 'commonmark',
  'cpp11', 'crayon', 'credentials', 'curl', 'data.table', 'DBI', 'dials',
  'downlit', 'e1071', 'evaluate', 'fontawesome', 'forecast', 'fs', 'future',
  'future.apply', 'gert', 'gower', 'gtable', 'hardhat', 'hexbin', 'highr',
  'httr2', 'ipred', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lava', 'lhs',
  'lubridate', 'modeldata', 'modelenv', 'nlme', 'nnet', 'openssl',
  'parallelly', 'parsnip', 'patchwork', 'pbdZMQ', 'pillar', 'pkgbuild',
  'pkgdown', 'pkgload', 'processx', 'prodlim', 'profvis', 'progressr',
  'promises', 'ps', 'purrr', 'R6', 'ragg', 'randomForest', 'Rcpp',
  'RcppArmadillo', 'RCurl', 'readxl', 'recipes', 'reprex', 'rlang',
  'rmarkdown', 'RODBC', 'roxygen2', 'rpart', 'RSQLite', 'rstudioapi',
  'sessioninfo', 'shiny', 'slider', 'survival', 'sys', 'systemfonts',
  'testthat', 'textshaping', 'tidymodels', 'timeDate', 'tinytex', 'tseries',
  'tune', 'urca', 'usethis', 'uuid', 'waldo', 'withr', 'workflows', 'xfun',
  'xml2', 'xts', 'yaml', 'yardstick', 'zip', 'zoo'
```

2. Read in the counts and meta data

In [2]:
```
library(edgeR)
library(dplyr)
```

```
Attaching package: ‘dplyr’


The following object is masked from ‘package:Biobase’:

    combine


The following object is masked from ‘package:matrixStats’:

    count


The following objects are masked from ‘package:GenomicRanges’:

    intersect, setdiff, union


The following object is masked from ‘package:GenomeInfoDb’:

    intersect


The following objects are masked from ‘package:IRanges’:

    collapse, desc, intersect, setdiff, slice, union


The following objects are masked from ‘package:S4Vectors’:

    first, intersect, rename, setdiff, setequal, union


The following objects are masked from ‘package:BiocGenerics’:

    combine, intersect, setdiff, union


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

In [51]:
```
install.packages("pheatmap")
BiocManager::install("EnhancedVolcano")
BiocManager::install("ComplexHeatmap")
```

```
Updating HTML index of packages in '.Library'

Making 'packages.html' ...
 done

'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.18 (BiocManager 1.30.25), R 4.3.3 (2024-02-29)

Warning message:
“package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'EnhancedVolcano'”
Old packages: 'askpass', 'backports', 'bit', 'bit64', 'bitops', 'broom',
  'bslib', 'caret', 'class', 'cli', 'clock', 'colorspace', 'commonmark',
  'cpp11', 'crayon', 'credentials', 'curl', 'data.table', 'DBI', 'dials',
  'downlit', 'e1071', 'evaluate', 'fontawesome', 'forecast', 'fs', 'future',
  'future.apply', 'gert', 'gower', 'gtable', 'hardhat', 'hexbin', 'highr',
  'httr2', 'ipred', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lava', 'lhs',
  'lubridate', 'modeldata', 'modelenv', 'nlme', 'nnet', 'openssl',
  'parallelly', 'parsnip', 'patchwork', 'pbdZMQ', 'pillar', 'pkgbuild',
  'pkgdown', 'pkgload', 'processx', 'prodlim', 'profvis', 'progressr',
  'promises', 'ps', 'purrr', 'R6', 'ragg', 'randomForest', 'Rcpp',
  'RcppArmadillo', 'RCurl', 'readxl', 'recipes', 'reprex', 'rlang',
  'rmarkdown', 'RODBC', 'roxygen2', 'rpart', 'RSQLite', 'rstudioapi',
  'sessioninfo', 'shiny', 'slider', 'survival', 'sys', 'systemfonts',
  'testthat', 'textshaping', 'tidymodels', 'timeDate', 'tinytex', 'tseries',
  'tune', 'urca', 'usethis', 'uuid', 'waldo', 'withr', 'workflows', 'xfun',
  'xml2', 'xts', 'yaml', 'yardstick', 'zip', 'zoo'

'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cran.r-project.org

Bioconductor version 3.18 (BiocManager 1.30.25), R 4.3.3 (2024-02-29)

Warning message:
“package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'ComplexHeatmap'”
Old packages: 'askpass', 'backports', 'bit', 'bit64', 'bitops', 'broom',
  'bslib', 'caret', 'class', 'cli', 'clock', 'colorspace', 'commonmark',
  'cpp11', 'crayon', 'credentials', 'curl', 'data.table', 'DBI', 'dials',
  'downlit', 'e1071', 'evaluate', 'fontawesome', 'forecast', 'fs', 'future',
  'future.apply', 'gert', 'gower', 'gtable', 'hardhat', 'hexbin', 'highr',
  'httr2', 'ipred', 'jsonlite', 'KernSmooth', 'knitr', 'later', 'lava', 'lhs',
  'lubridate', 'modeldata', 'modelenv', 'nlme', 'nnet', 'openssl',
  'parallelly', 'parsnip', 'patchwork', 'pbdZMQ', 'pillar', 'pkgbuild',
  'pkgdown', 'pkgload', 'processx', 'prodlim', 'profvis', 'progressr',
  'promises', 'ps', 'purrr', 'R6', 'ragg', 'randomForest', 'Rcpp',
  'RcppArmadillo', 'RCurl', 'readxl', 'recipes', 'reprex', 'rlang',
  'rmarkdown', 'RODBC', 'roxygen2', 'rpart', 'RSQLite', 'rstudioapi',
  'sessioninfo', 'shiny', 'slider', 'survival', 'sys', 'systemfonts',
  'testthat', 'textshaping', 'tidymodels', 'timeDate', 'tinytex', 'tseries',
```

```
'tune', 'urca', 'usethis', 'uuid', 'waldo', 'withr', 'workflows', 'xfun',
'xml2', 'xts', 'yaml', 'yardstick', 'zip', 'zoo'
```

In [88]:
```
# Step 1: Load the raw count data
raw_cnts <- read.csv("GSE288289_study2_genesCounts.csv",
                     sep = "\t",
                     header = TRUE,
                     row.names = 1,
                     check.names = FALSE)
dim(raw_cnts)
raw_cnts[1:5, 1:5]
```

38592 · 82

A data.frame: 5 × 5

	MEIS2_CE_A06_GT23-12174_TGCGAGAC-CAACAATG_S1_L001	NC0A3_KO_A12_GT24-01167_TTGGACTC-CTGCTTCC_S147_L005	BMLHE40_CE_H05_GT24-00868_TAAGGTCA-CTACGACA_S192_L007	N AG
	<int>	<int>	<int>	
ENSG00000268674	0	0	0	
ENSG00000271254	793	897	752	
ENSG00000275063	0	0	0	
ENSG00000277856	0	0	0	
ENSG00000276345	0	0	5	

In [90]:
```
# Step 2: Read sample descriptions from series matrix
lines <- readLines("GSE288289_series_matrix.txt")
desc_line <- lines[53]  # Row 53 based on your input
desc_fields <- strsplit(desc_line, "\t")[[1]]
sample_descriptions <- desc_fields[-1]  # Remove "!Sample_description" label
length(sample_descriptions)  # Should be 82
```

82

In [91]:
```
# Step 3: Create metadata data frame
# Assuming raw_cnts colnames are the sample IDs (e.g., "GT23-12159", not "GSM...")
meta_data <- data.frame(
  SampleID = colnames(raw_cnts),
  Description = sample_descriptions,
  stringsAsFactors = FALSE
)
```

In [92]:
```
# Step 4: Relabel conditions per your requirements
meta_data$Group <- ifelse(grepl("_KO_", meta_data$Description), "KO",
                          ifelse(grepl("_WT_|_KOLF2_", meta_data$Description), "WT"

# Check distribution
table(meta_data$Group, useNA = "ifany")  # Should show KO, WT, and NA counts
```

```
KO   WT <NA>
19   15   48
```

In [93]:
```
# Step 5: Filter to WT and KO only
meta_data_goal1 <- meta_data[!is.na(meta_data$Group), ]
raw_cnts_goal1 <- raw_cnts[, meta_data_goal1$SampleID]
dim(raw_cnts_goal1)  # [38592, 34]
dim(meta_data_goal1)  # [34, 3]
```

38592 · 34

34 · 3

Goal #1: compare all KO vs all WT

In [94]:
```
# Step 6: Goal 1 - All KO vs. All WT
group_goal1 <- factor(meta_data_goal1$Group, levels = c("WT", "KO"))
y_goal1 <- DGEList(counts = raw_cnts_goal1, group = group_goal1)
```

In [95]:
```
keep_goal1 <- filterByExpr(y_goal1)
y_goal1 <- y_goal1[keep_goal1, , keep.lib.sizes = FALSE]
```

In [96]:
```
# Normalize
y_goal1 <- calcNormFactors(y_goal1, method = "TMM")
```

In [97]:
```
# Estimate dispersion
y_goal1 <- estimateDisp(y_goal1)
```

```
Using classic mode.
```

In [98]:
```
et_goal1 <- exactTest(y_goal1, pair = c("WT", "KO"))
top_tags_goal1 <- topTags(et_goal1, n = Inf)$table
```

In [99]:
```
head(top_tags_goal1)
write.csv(top_tags_goal1, "DEG_All_KO_vs_All_WT.csv", row.names = TRUE)
```

A data.frame: 6 × 4

	logFC	logCPM	PValue	FDR
	<dbl>	<dbl>	<dbl>	<dbl>
ENSG00000117245	-1.340758	1.86903257	3.084251e-06	0.02094764
ENSG00000173261	-1.485721	0.00487537	4.113282e-06	0.02094764
ENSG00000228793	-1.717669	1.69487871	4.232953e-06	0.02094764
ENSG00000108551	-1.478372	0.61347980	5.359165e-06	0.02094764
ENSG00000184271	-1.758124	3.32194297	7.521795e-06	0.02177415
ENSG00000070669	-1.533315	7.12407722	1.004804e-05	0.02177415

In [46]:
```
library (gplots)
```

```
Attaching package: ‘gplots’


The following object is masked from ‘package:rtracklayer’:

    space


The following object is masked from ‘package:IRanges’:

    space


The following object is masked from ‘package:S4Vectors’:

    space


The following object is masked from ‘package:stats’:

    lowess
```

In [52]:
```
library(edgeR)
library(readr)
library(ggplot2)
library(pheatmap)
library(EnhancedVolcano)
library(ComplexHeatmap)
```

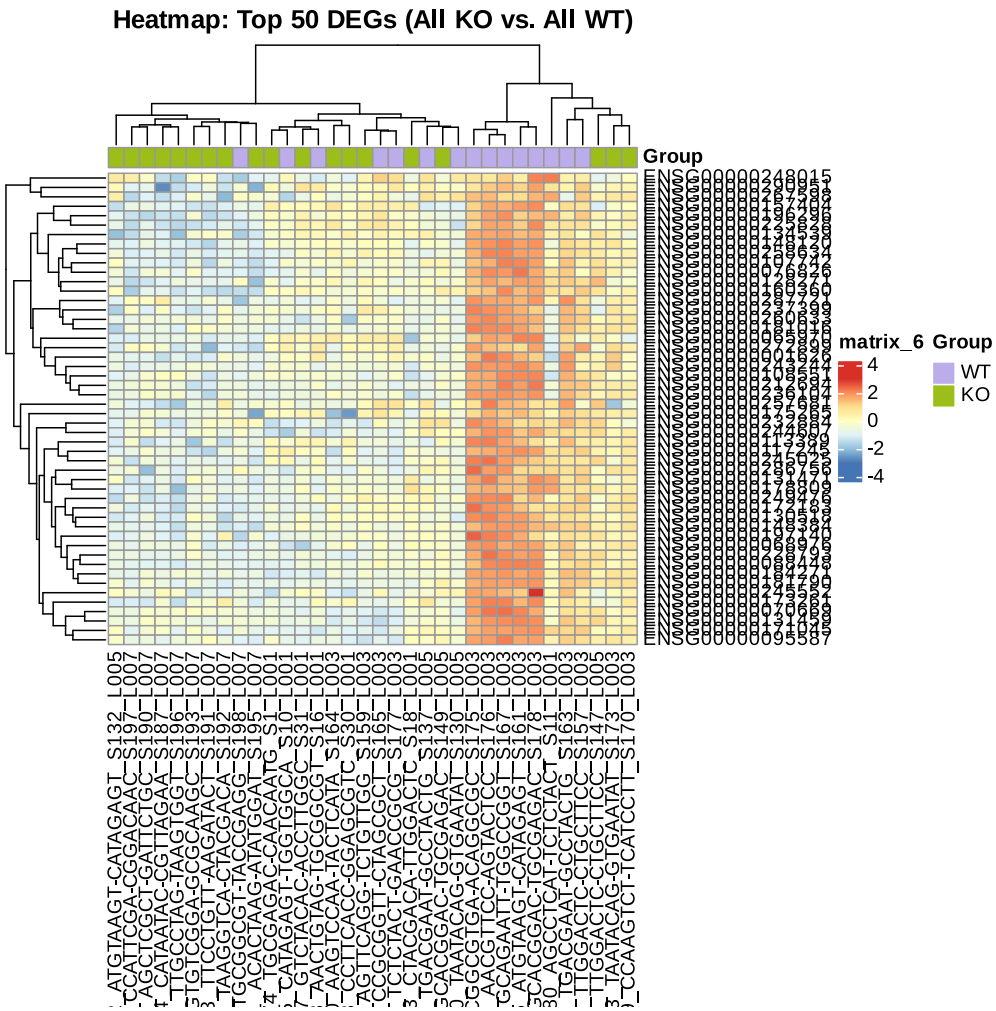
```
Loading required package: ggrepel

Loading required package: grid

========================================
ComplexHeatmap version 2.18.0
Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
Github page: https://github.com/jokergoo/ComplexHeatmap
Documentation: http://jokergoo.github.io/ComplexHeatmap-reference

If you use it in published research, please cite either one:
- Gu, Z. Complex Heatmap Visualization. iMeta 2022.
- Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
    genomic data. Bioinformatics 2016.


The new InteractiveComplexHeatmap package can directly export static
complex heatmaps into an interactive Shiny app with zero effort. Have a try!

This message can be suppressed by:
  suppressPackageStartupMessages(library(ComplexHeatmap))
========================================
! pheatmap() has been masked by ComplexHeatmap::pheatmap(). Most of the arguments
   in the original pheatmap() are identically supported in the new function. You
   can still use the original function by explicitly calling pheatmap::pheatmap().



Attaching package: 'ComplexHeatmap'


The following object is masked from 'package:pheatmap':

    pheatmap
```

In [100…

```r
# Step 6b: Visualizations for Goal 1
# Prepare normalized counts for heatmaps
logcpm_goal1 <- cpm(y_goal1, log = TRUE)
top_genes_goal1 <- head(rownames(top_tags_goal1), 50)  # Top 50 DEGs
heatmap_data_goal1 <- logcpm_goal1[top_genes_goal1, ]
```

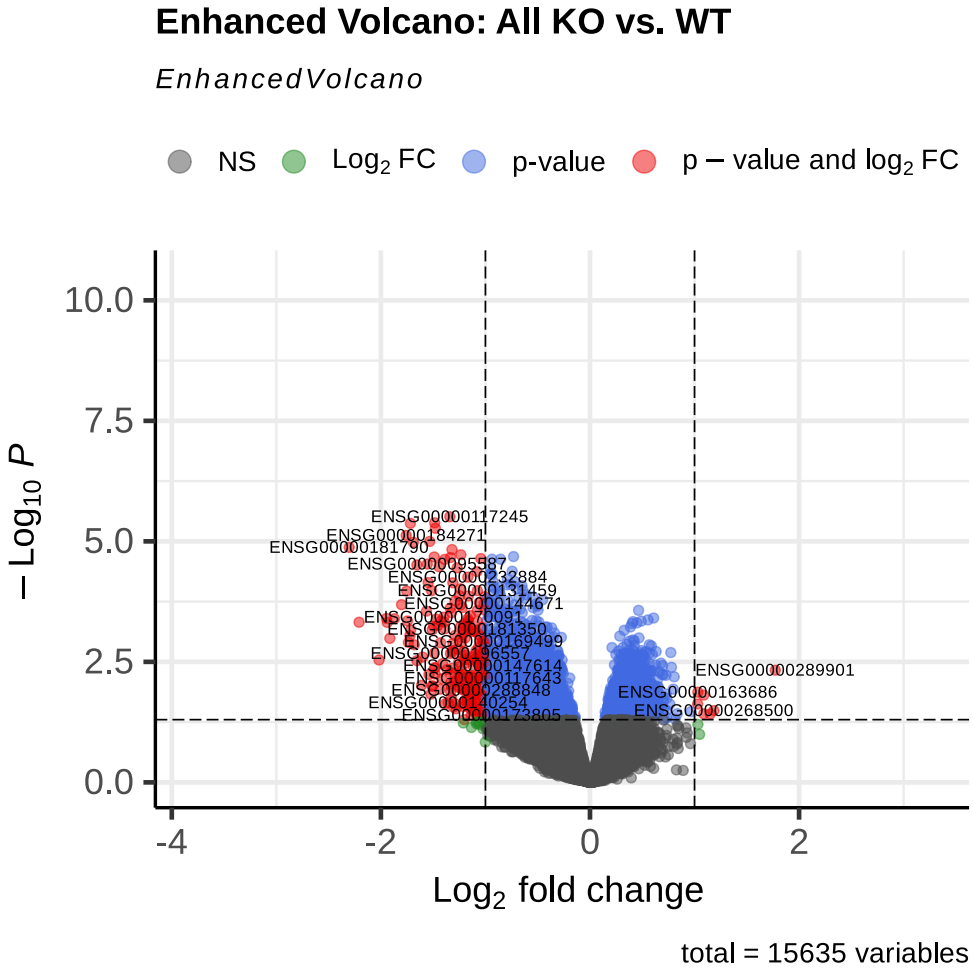
In [101…

```r
# 1. pheatmap
pheatmap(heatmap_data_goal1,
         scale = "row",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         annotation_col = data.frame(Group = group_goal1, row.names = colnames(heat
         main = "Heatmap: Top 50 DEGs (All KO vs. All WT)")
         #,filename = "pheatmap_All_KO_vs_WT.png")
```



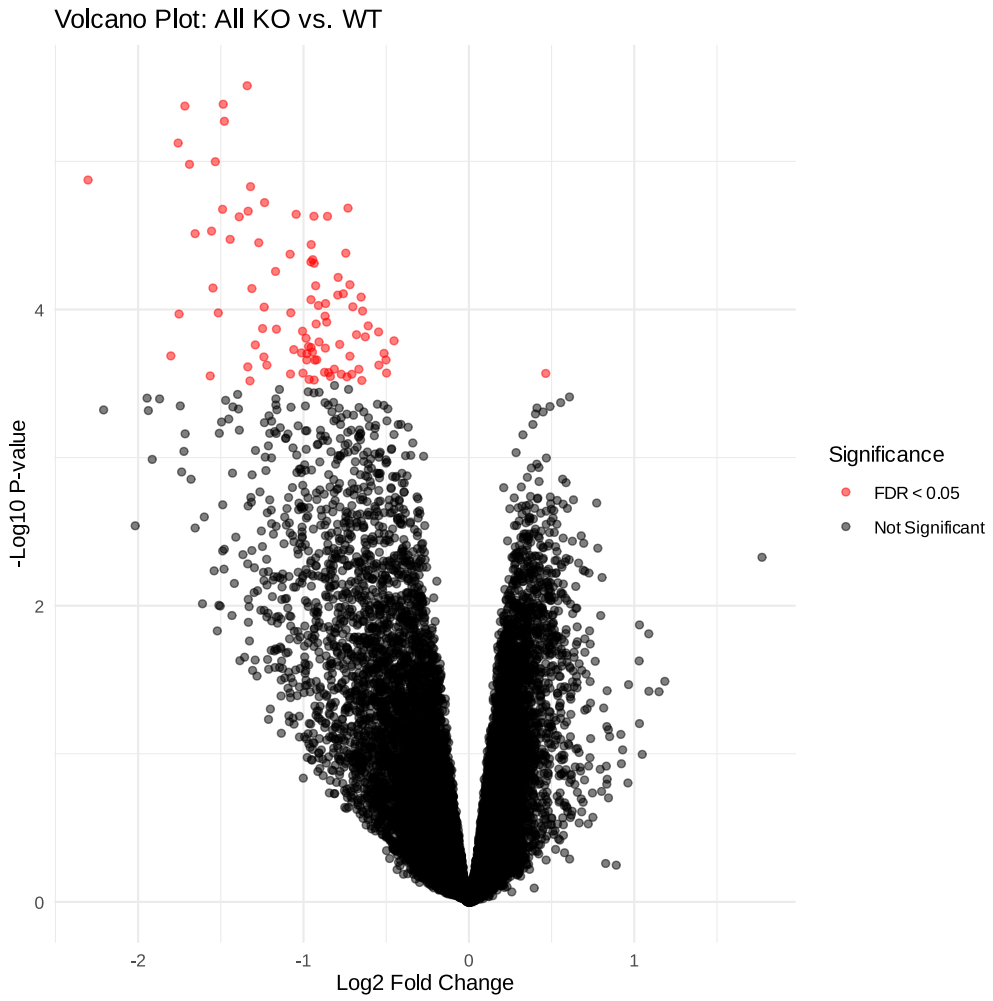
In [56]:

```
# 2. EnhancedVolcano
EnhancedVolcano(top_tags_goal1,
                lab = rownames(top_tags_goal1),
                x = 'logFC',
                y = 'PValue',
                title = 'Enhanced Volcano: All KO vs. WT',
                pCutoff = 0.05,
                FCcutoff = 1.0,
                pointSize = 2.0,
                labSize = 3.0)
#ggsave("EnhancedVolcano_All_KO_vs_WT.png")
```

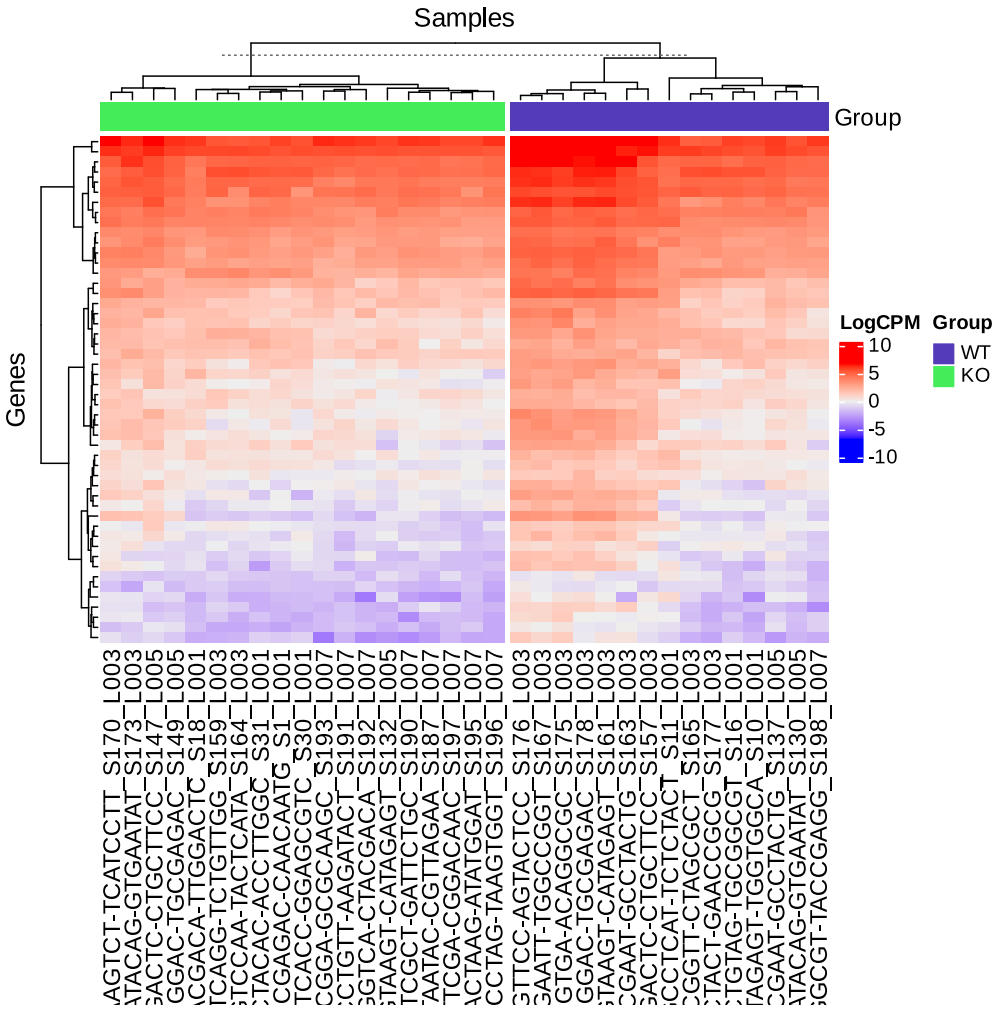


In [60]:

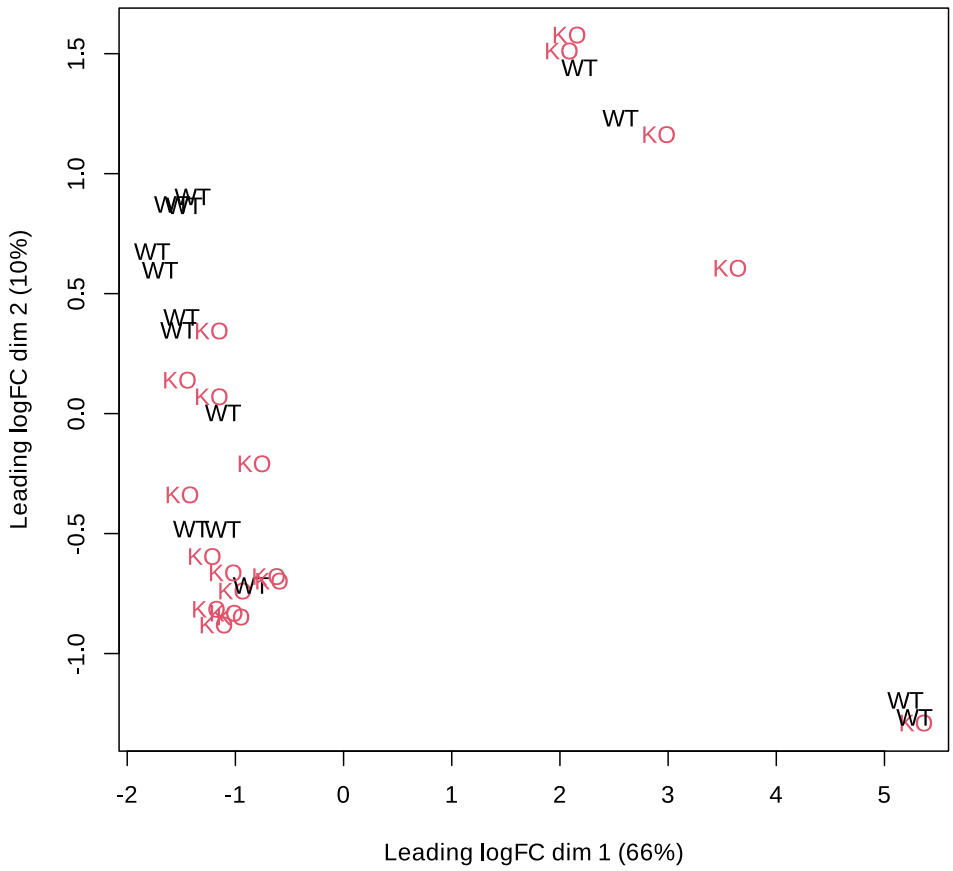
```
ggplot(volcano_data_goal1, aes(x = logFC, y = -log10(PValue), color = Significance)
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("FDR < 0.05" = "red", "Not Significant" = "black"))
  labs(title = "Volcano Plot: All KO vs. WT", x = "Log2 Fold Change", y = "-Log10 P
  theme_minimal()
#ggsave("Volcano_Plot_All_KO_vs_WT.png")
```



In [57]:
```
# Generate and display the heatmap
Heatmap(heatmap_data_goal1,
        name = "LogCPM",
        row_title = "Genes",
        column_title = "Samples",
        cluster_rows = TRUE,
        cluster_columns = TRUE,
        column_split = group_goal1,
        show_row_names = FALSE,
        top_annotation = HeatmapAnnotation(Group = group_goal1))
```



In [58]:
```
# Generate and display the MDS plot
plotMDS(y_goal1,
        main = "MDS Plot: All KO vs. WT",
        col = as.numeric(group_goal1),
        labels = group_goal1)
```



In [ ]:

Goal #2:

In [103…
```
# Goal 2 - MDX1 KO vs. MDX1 WT
# Subset MDX1 samples
meta_data_mdx1 <- meta_data[grepl("MDX1_KO_|MDX1_KOLF2_", meta_data$Description), ]
meta_data_mdx1$Group <- ifelse(grepl("MDX1_KO_", meta_data_mdx1$Description), "MDX1
```

In [104…
```
# Subset counts
raw_cnts_mdx1 <- raw_cnts[, meta_data_mdx1$SampleID, drop = FALSE]
print("Dimensions of raw_cnts_mdx1:")
dim(raw_cnts_mdx1)
```

```
[1] "Dimensions of raw_cnts_mdx1:"
```

38592 · 6

In [105…
```
# Proceed with edgeR
group_mdx1 <- factor(meta_data_mdx1$Group, levels = c("MDX1_WT", "MDX1_KO"))
y_mdx1 <- DGEList(counts = raw_cnts_mdx1, group = group_mdx1)
keep_mdx1 <- filterByExpr(y_mdx1)
y_mdx1 <- y_mdx1[keep_mdx1, , keep.lib.sizes = FALSE]
y_mxx1 <- calcNormFactors(y_mdx1, method = "TMM")
```

```
y_mdx1 <- estimateDisp(y_mdx1)
et_mdx1 <- exactTest(y_mdx1, pair = c("MDX1_WT", "MDX1_KO"))
top_tags_mdx1 <- topTags(et_mdx1, n = Inf)$table
head(top_tags_mdx1)
write.csv(top_tags_mdx1, "DEG_MDX1_KO_vs_MDX1_WT.csv", row.names = TRUE)
```

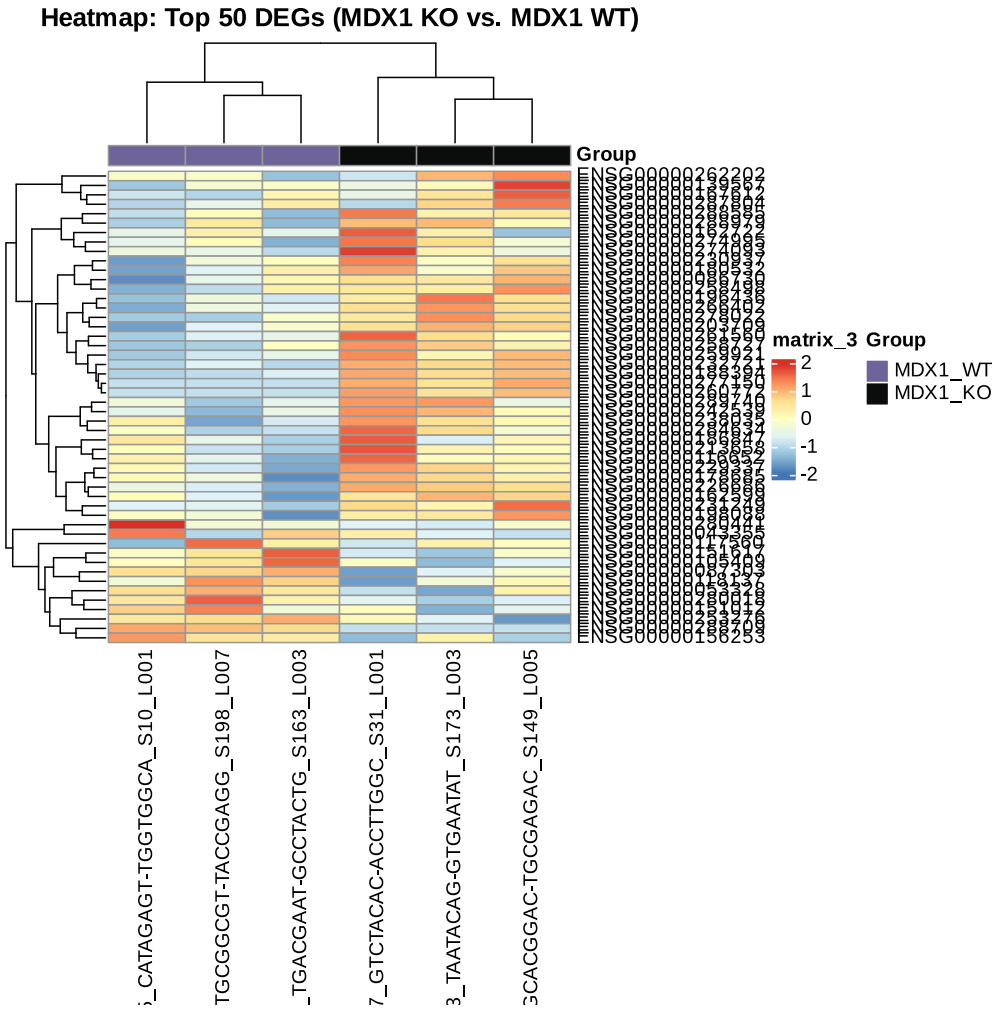
```
Using classic mode.
```

A data.frame: 6 × 4

	logFC	logCPM	PValue	FDR
	<dbl>	<dbl>	<dbl>	<dbl>
ENSG00000288709	-9.360011	0.7599687	2.292324e-11	3.507944e-07
ENSG00000277150	9.731498	1.1223018	1.955343e-10	1.496131e-06
ENSG00000260772	9.394729	0.7983569	5.321627e-10	2.714562e-06
ENSG00000196436	1.986708	-0.3915383	8.526406e-03	1.000000e+00
ENSG00000289740	1.727436	4.7196744	1.178506e-02	1.000000e+00
ENSG00000226686	1.551394	-0.6351895	1.707694e-02	1.000000e+00

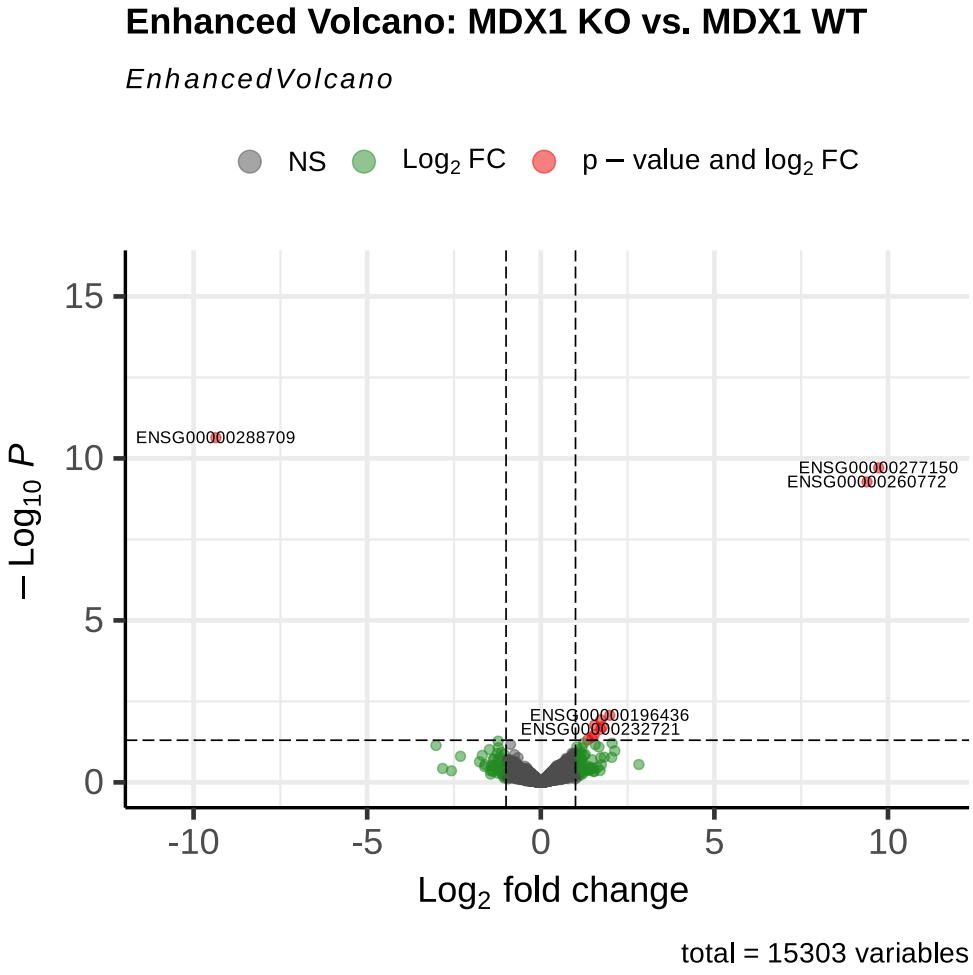
In [61]:
```
# Prepare normalized counts for heatmaps
logcpm_mdx1 <- cpm(y_mdx1, log = TRUE)
top_genes_mdx1 <- head(rownames(top_tags_mdx1), 50)  # Top 50 DEGs
heatmap_data_mdx1 <- logcpm_mdx1[top_genes_mdx1, ]
```

In [62]:
```
# 1. pheatmap
pheatmap(heatmap_data_mdx1,
         scale = "row",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         annotation_col = data.frame(Group = group_mdx1, row.names = colnames(heatm
         main = "Heatmap: Top 50 DEGs (MDX1 KO vs. MDX1 WT)")
         #,filename = "pheatmap_MDX1_KO_vs_MDX1_WT.png")
```



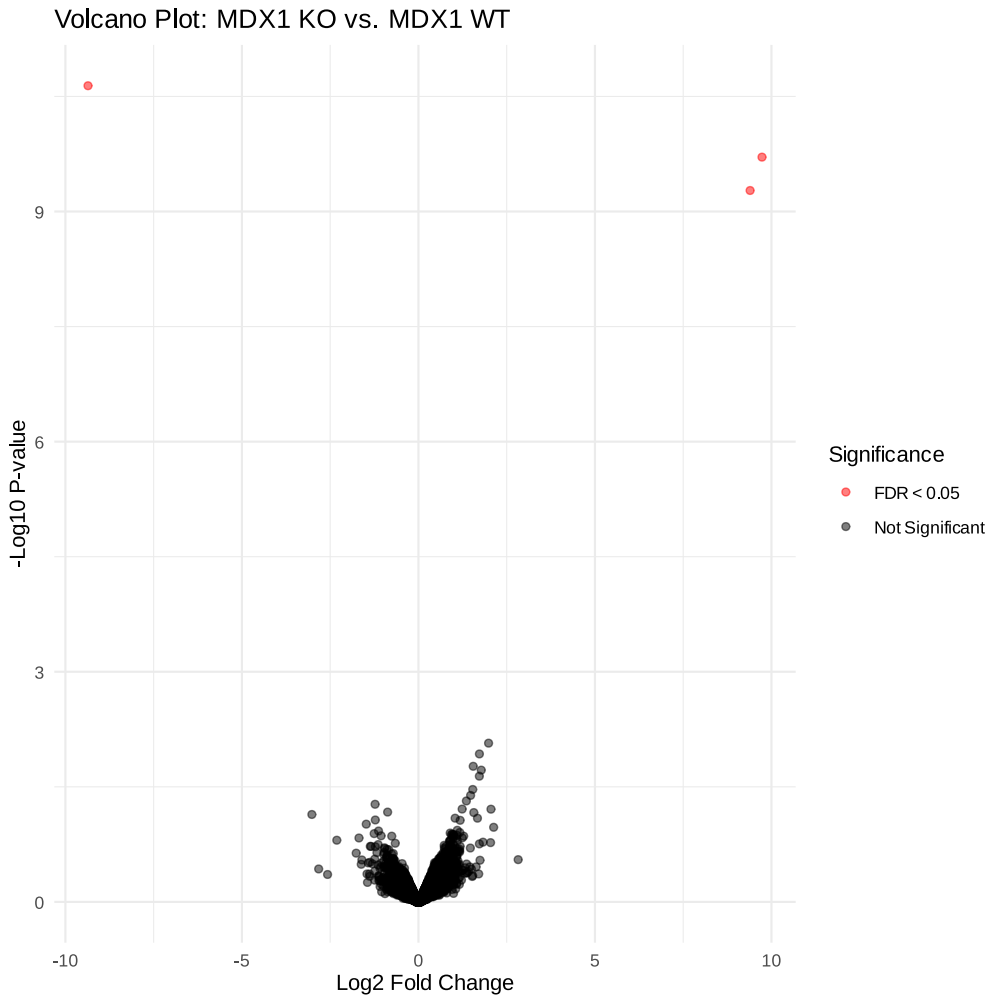
In [63]:

```
# 2. EnhancedVolcano
EnhancedVolcano(top_tags_mdx1,
                lab = rownames(top_tags_mdx1),
                x = 'logFC',
                y = 'PValue',
                title = 'Enhanced Volcano: MDX1 KO vs. MDX1 WT',
                pCutoff = 0.05,
                FCcutoff = 1.0,
                pointSize = 2.0,
                labSize = 3.0)
#ggsave("EnhancedVolcano_MDX1_KO_vs_MDX1_WT.png")
```

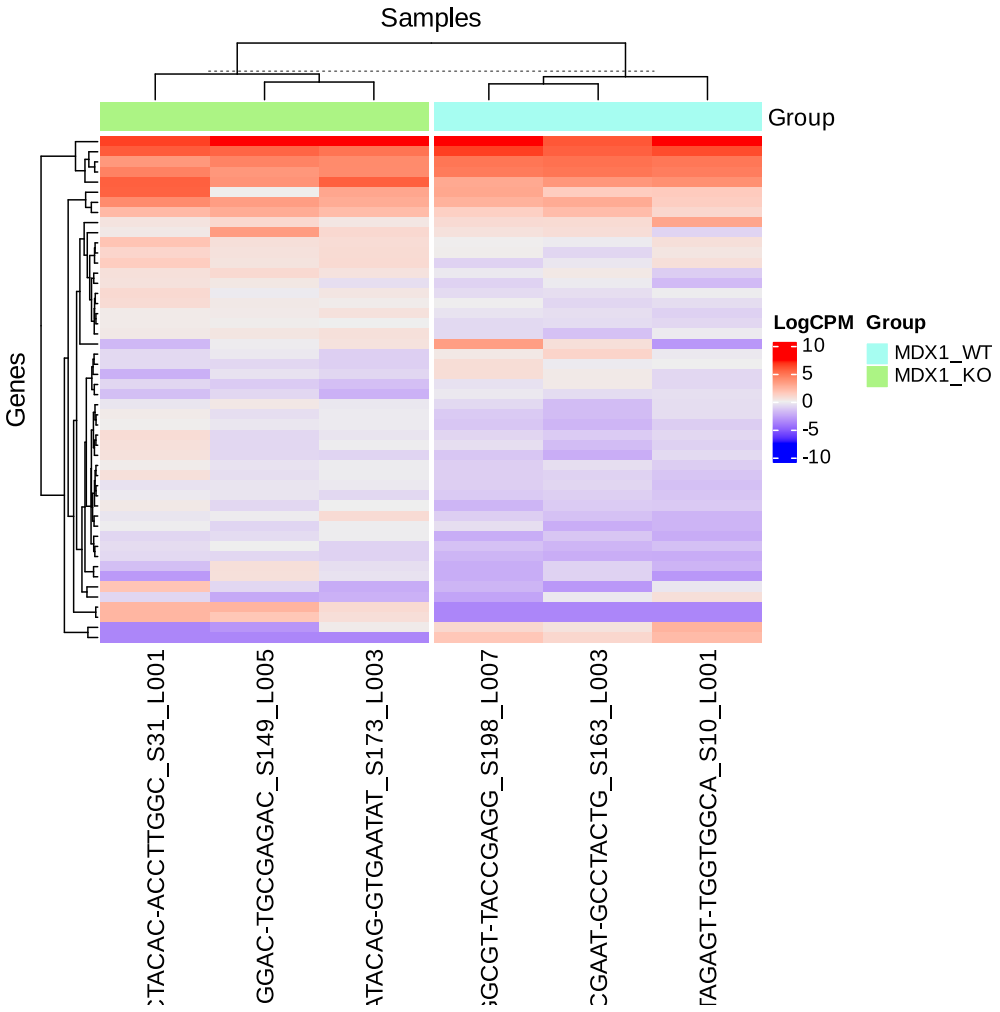


In [64]:

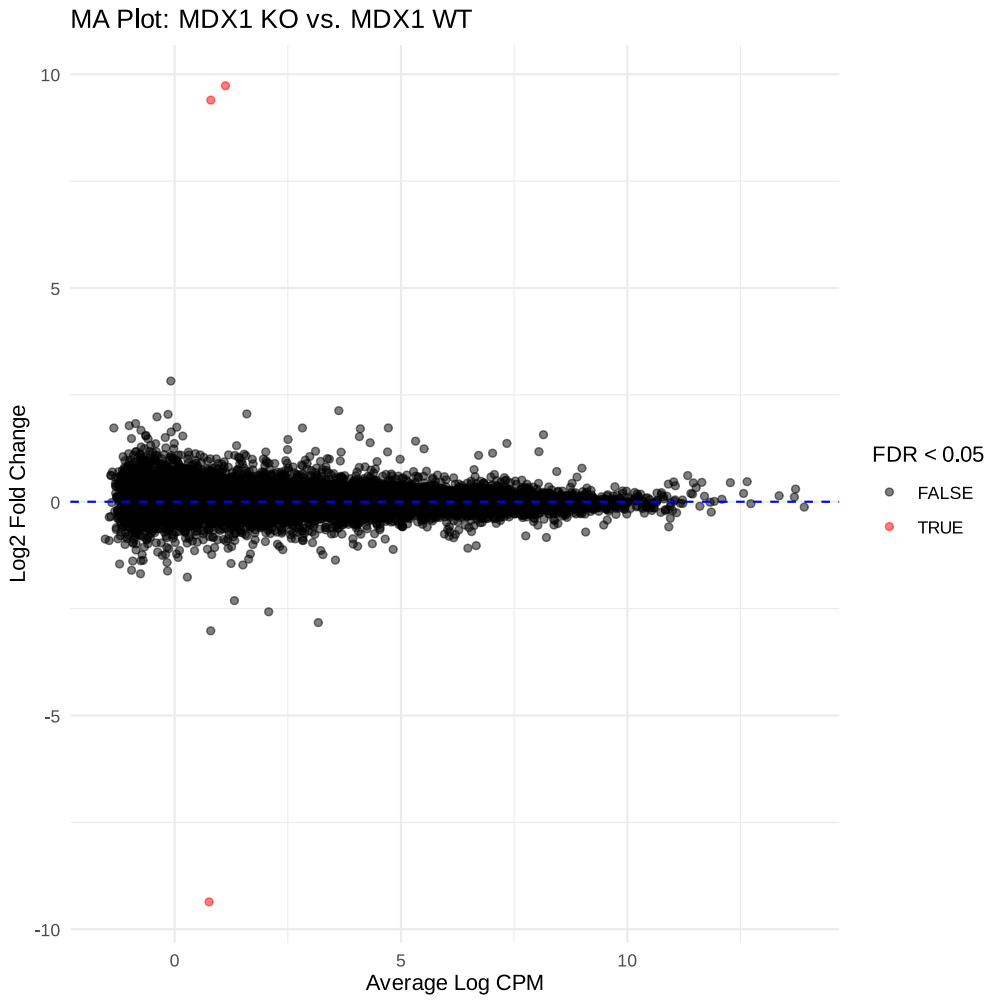
```
# Volcano Plot
volcano_data_mdx1 <- top_tags_mdx1
volcano_data_mdx1$Significance <- ifelse(volcano_data_mdx1$FDR < 0.05, "FDR < 0.05"
ggplot(volcano_data_mdx1, aes(x = logFC, y = -log10(PValue), color = Significance))
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("FDR < 0.05" = "red", "Not Significant" = "black"))
  labs(title = "Volcano Plot: MDX1 KO vs. MDX1 WT", x = "Log2 Fold Change", y = "-L
  theme_minimal()
#ggsave("Volcano_Plot_MDX1_KO_vs_MDX1_WT.png")
```



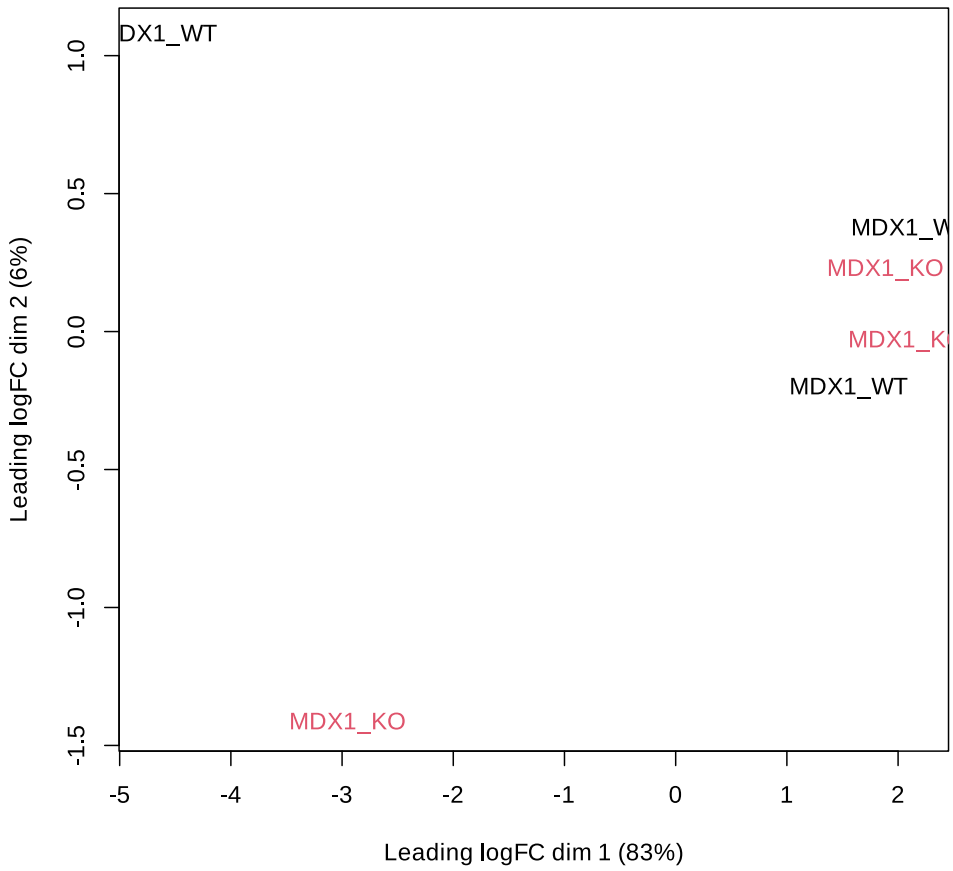
In [65]:
```
# Generate and display the heatmap
Heatmap(heatmap_data_mdx1,
        name = "LogCPM",
        row_title = "Genes",
        column_title = "Samples",
        cluster_rows = TRUE,
        cluster_columns = TRUE,
        column_split = group_mdx1,
        show_row_names = FALSE,
        top_annotation = HeatmapAnnotation(Group = group_mdx1))

# png("ComplexHeatmap_MDX1_KO_vs_MDX1_WT.png") # Saving is commented out
# dev.off()
```



In [42]:
```
# MA Plot
ggplot(volcano_data_mdx1, aes(x = logCPM, y = logFC, color = FDR < 0.05)) +
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("TRUE" = "red", "FALSE" = "black"), name = "FDR < 0
  geom_hline(yintercept = 0, linetype = "dashed", color = "blue") +
  labs(title = "MA Plot: MDX1 KO vs. MDX1 WT", x = "Average Log CPM", y = "Log2 Fol
  theme_minimal()
#ggsave("MA_Plot_MDX1_KO_vs_MDX1_WT.png")
```



In [43]:
```
# Generate and display the MDS plot
plotMDS(y_mdx1,
        main = "MDS Plot: MDX1 KO vs. MDX1 WT",
        col = as.numeric(group_mdx1),
        labels = group_mdx1)
```



T-Test - We using "GSEXXXXXX_studyX_genesTPM.csv.gz" as TPM normalized

In [106…

```
# Step 1: Load the TPM normalized count data (distinct from raw counts)
tpm_raw <- read.csv("GSE288289_study2_genesTPM.csv",
                    sep = "\t",
                    header = TRUE,
                    row.names = 1,
                    check.names = FALSE)
dim(tpm_raw)  # Expected: [38592, 82] or similar
head(tpm_raw[1:5, 1:5])
```

38592 · 82

A data.frame: 5 × 5

	MEIS2_CE_A06_GT23-12174_TGCGAGAC-CAACAATG_S1_L001	NC0A3_KO_A12_GT24-01167_TTGGACTC-CTGCTTCC_S147_L005	BMLHE40_CE_H05_GT24-00868_TAAGGTCA-CTACGACA_S192_L007	N AG
	<dbl>	<dbl>	<dbl>	
ENSG00000268674	0.0000	0.0000	0.0000	
ENSG00000271254	21.7165	19.0753	17.0863	
ENSG00000275063	0.0000	0.0000	0.0000	
ENSG00000277856	0.0000	0.0000	0.0000	
ENSG00000276345	0.0000	0.0000	0.6943	

In [107…
```r
# Step 2: Read sample descriptions from series matrix (distinct variable)
tpm_lines <- readLines("GSE288289_series_matrix.txt")
tpm_desc_line <- tpm_lines[53]
tpm_desc_fields <- strsplit(tpm_desc_line, "\t")[[1]]
tpm_sample_descriptions <- tpm_desc_fields[-1]
length(tpm_sample_descriptions)  # Should be 82
```

82

In [108…
```r
# Step 3: Create metadata data frame for TPM
tpm_meta_data <- data.frame(
  SampleID = colnames(tpm_raw),
  Description = tpm_sample_descriptions,
  stringsAsFactors = FALSE
)
```

In [109…
```r
# Step 4: Relabel conditions for Goal 1 (TPM)
tpm_meta_data$Group <- ifelse(grepl("_KO_", tpm_meta_data$Description), "KO",
                              ifelse(grepl("_WT_|_KOLF2_", tpm_meta_data$Descriptio
print("Goal 1 - Group distribution before filtering (TPM):")
table(tpm_meta_data$Group, useNA = "ifany")  # Should show KO: 19, WT: 15, NA: 48
```

```
[1] "Goal 1 - Group distribution before filtering (TPM):"
 KO   WT <NA>
 19   15   48
```

In [113…
```r
# Step 5: Filter to WT and KO only (Goal 1, TPM)
tpm_meta_data_goal1 <- tpm_meta_data[!is.na(tpm_meta_data$Group), ]
tpm_data_goal1 <- tpm_raw[, tpm_meta_data_goal1$SampleID]
dim(tpm_data_goal1)  # [38592, 34]
dim(tpm_meta_data_goal1)  # [34, 3]
```

38592 · 34

34 · 3

In [114…
```r
# Pre-filter genes with mean TPM > 1 to reduce NA/Inf issues
tpm_data_goal1 <- tpm_data_goal1[rowMeans(tpm_data_goal1) > 1, ]
```

```
print("Goal 1 - Dimensions after filtering low-expression genes (TPM):")
dim(tpm_data_goal1)
```

```
[1] "Goal 1 - Dimensions after filtering low-expression genes (TPM):"
```

14454 · 34

In [115…

```
# Step 6: Goal 1 - T-Tests for All KO vs. WT (TPM)
tpm_t_test_goal1 <- apply(tpm_data_goal1, 1, function(x) {
  ko_vals <- x[tpm_meta_data_goal1$Group == "KO"]
  wt_vals <- x[tpm_meta_data_goal1$Group == "WT"]
  if (length(ko_vals) > 1 & length(wt_vals) > 1 & mean(wt_vals) > 0) {  # Avoid div
    test <- t.test(ko_vals, wt_vals)
    log_fc <- log2(mean(ko_vals) / mean(wt_vals))
    if (is.finite(log_fc)) {  # Only return finite logFC
      return(c(logFC = log_fc, PValue = test$p.value))
    }
  }
  return(c(logFC = NA, PValue = NA))
})
tpm_t_test_goal1 <- as.data.frame(t(tpm_t_test_goal1))
tpm_t_test_goal1$FDR <- p.adjust(tpm_t_test_goal1$PValue, method = "BH")
```

In [116…

```
# Post-filter to remove NA values
tpm_t_test_goal1 <- tpm_t_test_goal1[!is.na(tpm_t_test_goal1$logFC) & !is.na(tpm_t_
print("Goal 1 - Number of genes after NA filtering:")
nrow(tpm_t_test_goal1)
write.csv(tpm_t_test_goal1, "TTest_All_KO_vs_WT_TPM.csv", row.names = TRUE)
```

```
[1] "Goal 1 - Number of genes after NA filtering:"
```

14453

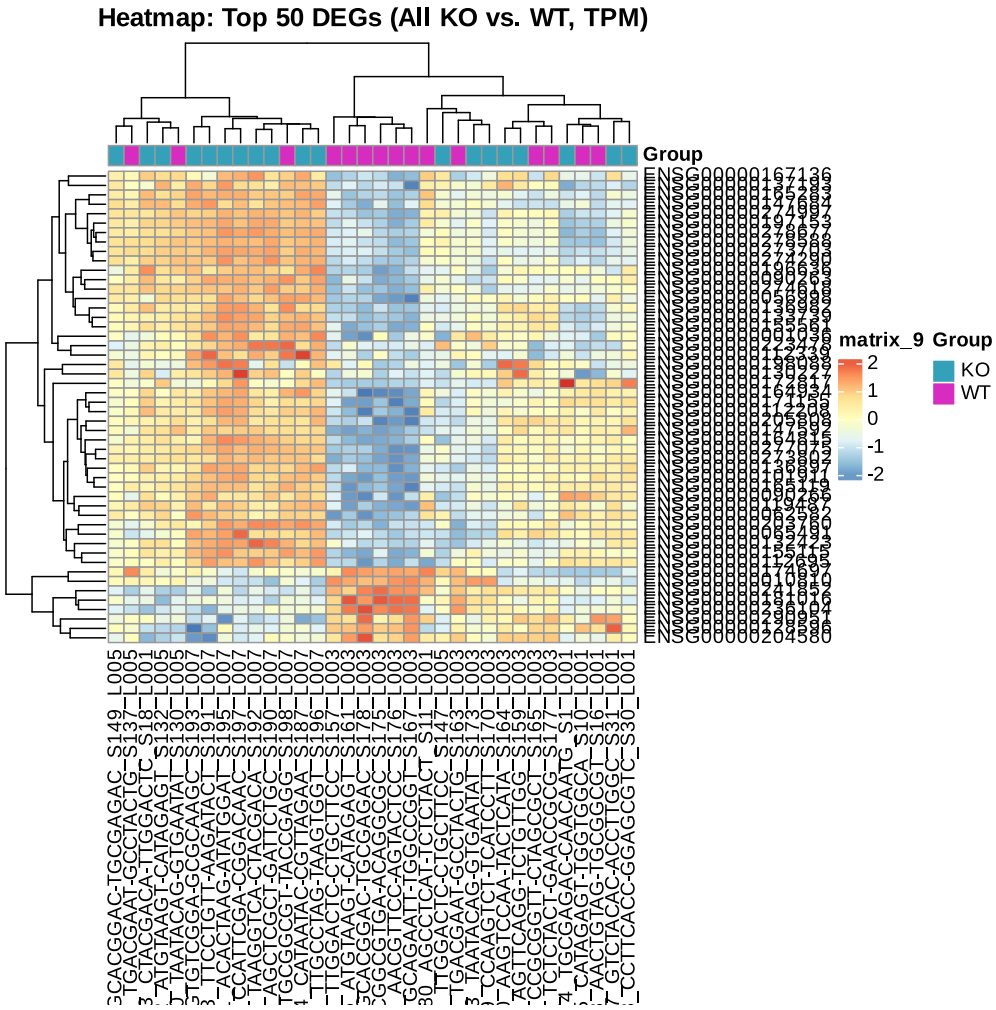
In [117…

```
# Step 6b: Visualizations for Goal 1 (TPM)
# Prepare logTPM for heatmaps (using filtered data)
tpm_logtpm_goal1 <- log2(tpm_data_goal1 + 1)
tpm_top_genes_goal1 <- head(rownames(tpm_t_test_goal1[order(tpm_t_test_goal1$FDR),
tpm_heatmap_data_goal1 <- tpm_logtpm_goal1[tpm_top_genes_goal1, ]
```

In [120…

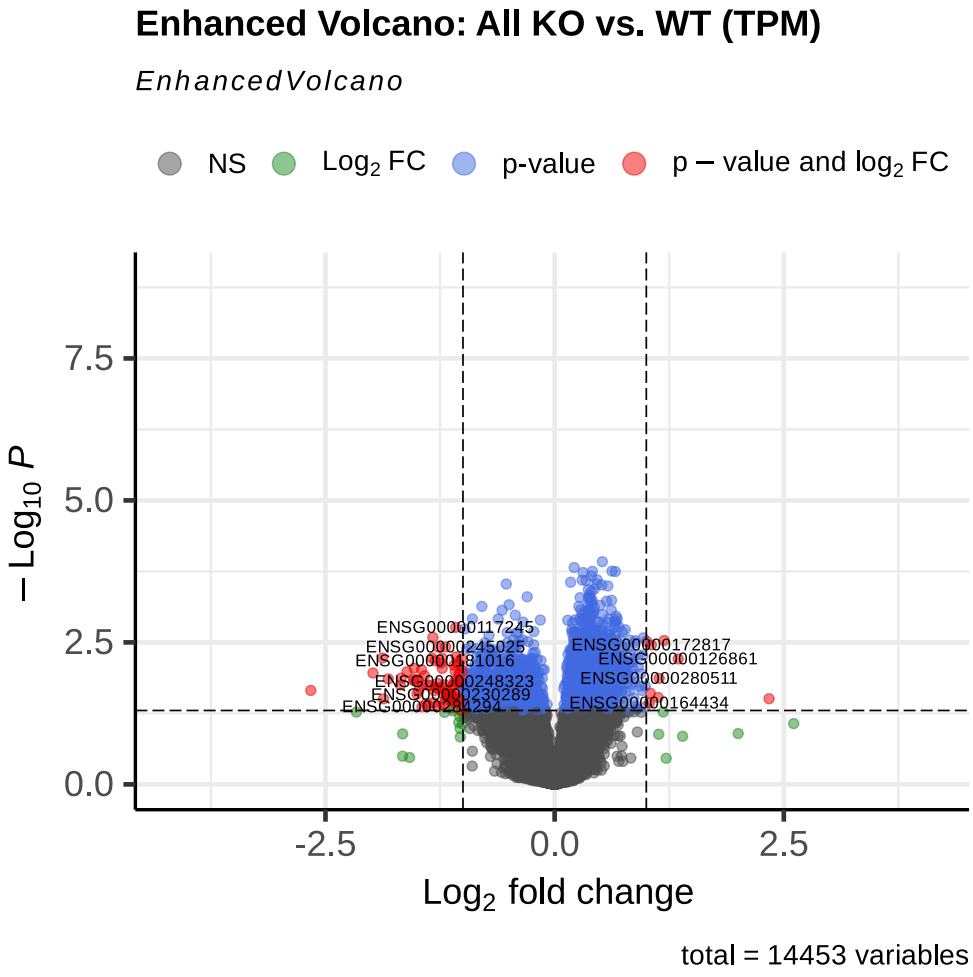
```
# 1. pheatmap
pheatmap(tpm_heatmap_data_goal1,
         scale = "row",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         annotation_col = data.frame(Group = tpm_meta_data_goal1$Group, row.names =
         main = "Heatmap: Top 50 DEGs (All KO vs. WT, TPM)")
         #,filename = "pheatmap_All_KO_vs_WT_TPM.png")
```

```
Warning message:
“The input is a data frame, convert it to the matrix.”
```



In [121…

```
# 2. EnhancedVolcano
EnhancedVolcano(tpm_t_test_goal1,
                lab = rownames(tpm_t_test_goal1),
                x = 'logFC',
                y = 'PValue',
                title = 'Enhanced Volcano: All KO vs. WT (TPM)',
                pCutoff = 0.05,
                FCcutoff = 1.0,
                pointSize = 2.0,
                labSize = 3.0)
#ggsave("EnhancedVolcano_All_KO_vs_WT_TPM.png")
```



In [124…
```r
# Step 7: Goal 2 - MDX1 KO vs. MDX1 WT (TPM)
tpm_meta_data_mdx1 <- tpm_meta_data[grepl("MDX1_KO_|MDX1_KOLF2_", tpm_meta_data$Des
tpm_meta_data_mdx1$Group <- ifelse(grepl("MDX1_KO_", tpm_meta_data_mdx1$Description
```

In [125…
```r
# Diagnostics
print("Goal 2 - Number of MDX1 samples found (TPM):")
print(nrow(tpm_meta_data_mdx1))
print("Goal 2 - MDX1 sample descriptions (TPM):")
print(tpm_meta_data_mdx1$Description)
```

```
[1] "Goal 2 - Number of MDX1 samples found (TPM):"
[1] 6
[1] "Goal 2 - MDX1 sample descriptions (TPM):"
[1] "\"Library name: MDX1_KO_C03_GT23-10879_CGTTAGAA-GACCTGAA_S5\""
[2] "\"Library name: MDX1_KO_C04_GT23-10880_AGCCTCAT-TCTCTACT_S11\""
[3] "\"Library name: MDX1_KO_D01_GT23-10881_GATTCTGC-CTCTCGTC_S9\""
[4] "\"Library name: MDX1_KOLF2_1_GT23-10876_GGCATTCT-CAAGCTAG_S8\""
[5] "\"Library name: MDX1_KOLF2_2_GT23-10877_AATGCCTC-TGGATCGA_S3\""
[6] "\"Library name: MDX1_KOLF2_3_GT23-10878_TACCGAGG-AGTTCAGG_S13\""
```

In [126…
```r
# Subset TPM data
tpm_data_mdx1 <- tpm_raw[, tpm_meta_data_mdx1$SampleID, drop = FALSE]
tpm_data_mdx1 <- tpm_data_mdx1[rowMeans(tpm_data_mdx1) > 1, ]  # Pre-filter low-exp
print("Goal 2 - Dimensions of tpm_data_mdx1 after filtering:")
dim(tpm_data_mdx1)
```

```
[1] "Goal 2 - Dimensions of tpm_data_mdx1 after filtering:"
14446 · 6
```

In [127…
```r
# Step 7b: T-Tests for Goal 2 (TPM)
tpm_t_test_mdx1 <- apply(tpm_data_mdx1, 1, function(x) {
  ko_vals <- x[tpm_meta_data_mdx1$Group == "MDX1_KO"]
  wt_vals <- x[tpm_meta_data_mdx1$Group == "MDX1_WT"]
  if (length(ko_vals) > 1 & length(wt_vals) > 1 & mean(wt_vals) > 0) {  # Avoid div
    test <- t.test(ko_vals, wt_vals)
    log_fc <- log2(mean(ko_vals) / mean(wt_vals))
    if (is.finite(log_fc)) {  # Only return finite logFC
      return(c(logFC = log_fc, PValue = test$p.value))
    }
  }
  return(c(logFC = NA, PValue = NA))
})
tpm_t_test_mdx1 <- as.data.frame(t(tpm_t_test_mdx1))
tpm_t_test_mdx1$FDR <- p.adjust(tpm_t_test_mdx1$PValue, method = "BH")
```

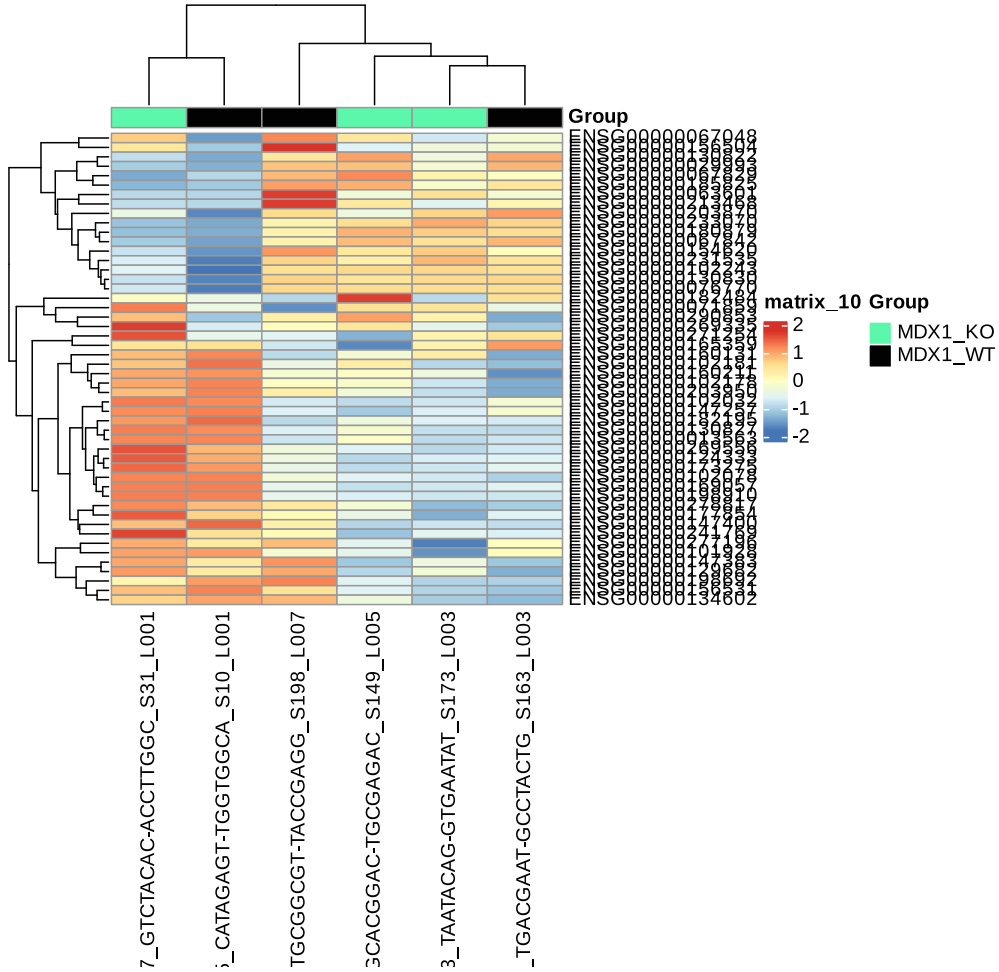
In [128…
```r
# Post-filter to remove NA values
tpm_t_test_mdx1 <- tpm_t_test_mdx1[!is.na(tpm_t_test_mdx1$logFC) & !is.na(tpm_t_tes
print("Goal 2 - Number of genes after NA filtering:")
nrow(tpm_t_test_mdx1)
write.csv(tpm_t_test_mdx1, "TTest_MDX1_KO_vs_MDX1_WT_TPM.csv", row.names = TRUE)
```

```
[1] "Goal 2 - Number of genes after NA filtering:"
14437
```

In [129…
```r
# Step 7c: Visualizations for Goal 2 (TPM)
# Prepare logTPM for heatmaps
tpm_logtpm_mdx1 <- log2(tpm_data_mdx1 + 1)
tpm_top_genes_mdx1 <- head(rownames(tpm_t_test_mdx1[order(tpm_t_test_mdx1$FDR), ]),
tpm_heatmap_data_mdx1 <- tpm_logtpm_mdx1[tpm_top_genes_mdx1, ]
```

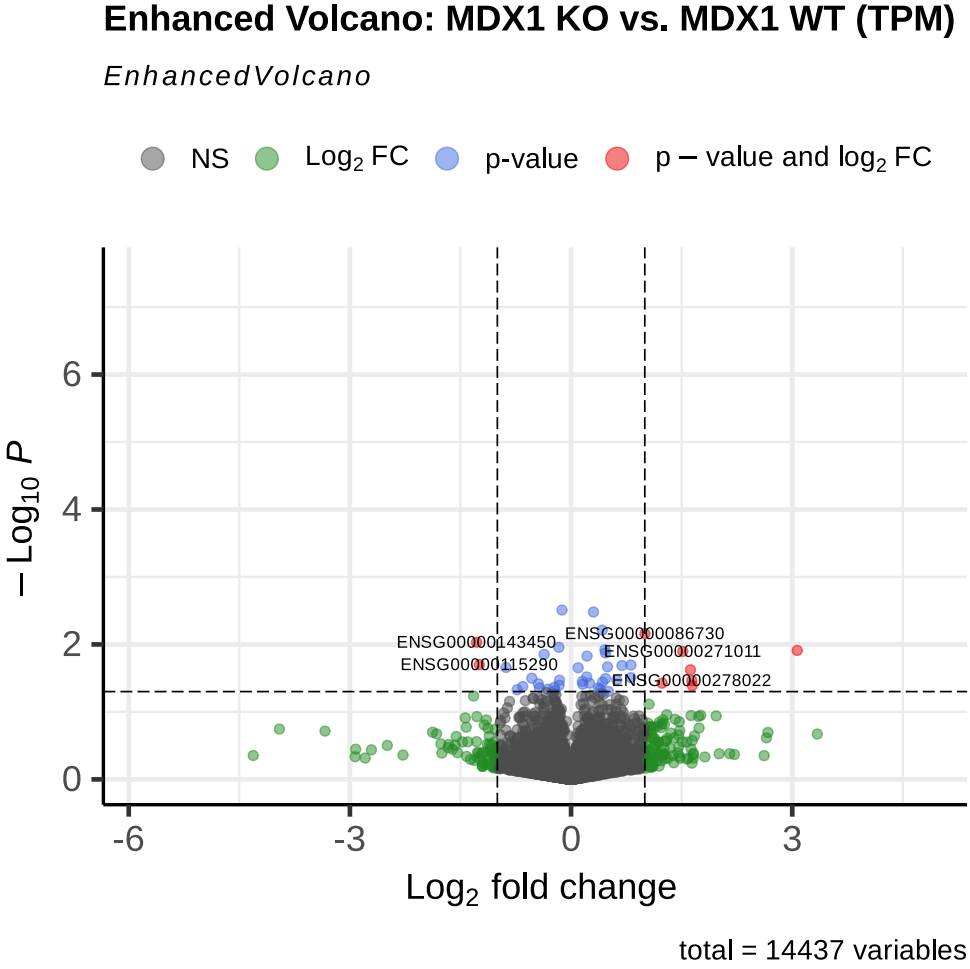
In [130…
```r
# 1. pheatmap
pheatmap(tpm_heatmap_data_mdx1,
         scale = "row",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         annotation_col = data.frame(Group = tpm_meta_data_mdx1$Group, row.names =
         main = "Heatmap: Top 50 DEGs (MDX1 KO vs. MDX1 WT, TPM)")
         #,filename = "pheatmap_MDX1_KO_vs_MDX1_WT_TPM.png")
```

```
Warning message:
“The input is a data frame, convert it to the matrix.”
```



In [131…

```
# 2. EnhancedVolcano
EnhancedVolcano(tpm_t_test_mdx1,
                lab = rownames(tpm_t_test_mdx1),
                x = 'logFC',
                y = 'PValue',
                title = 'Enhanced Volcano: MDX1 KO vs. MDX1 WT (TPM)',
                pCutoff = 0.05,
                FCcutoff = 1.0,
                pointSize = 2.0,
                labSize = 3.0)
#ggsave("EnhancedVolcano_MDX1_KO_vs_MDX1_WT_TPM.png")
```



Now we compare the edgeR vs T-Test for both goal #1 and goal #2

In [136…
```
install.packages("VennDiagram")
```

```
Updating HTML index of packages in '.Library'

Making 'packages.html' ...
 done
```

In [137…
```
library(VennDiagram)
library(readr)
```

```
Loading required package: futile.logger
```

In [138…
```
library(readr)
library(dplyr)
```

```
Attaching package: ‘dplyr’


The following object is masked from ‘package:Biobase’:

    combine


The following object is masked from ‘package:matrixStats’:

    count


The following objects are masked from ‘package:GenomicRanges’:

    intersect, setdiff, union


The following object is masked from ‘package:GenomeInfoDb’:

    intersect


The following objects are masked from ‘package:IRanges’:

    collapse, desc, intersect, setdiff, slice, union


The following objects are masked from ‘package:S4Vectors’:

    first, intersect, rename, setdiff, setequal, union


The following objects are masked from ‘package:BiocGenerics’:

    combine, intersect, setdiff, union


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

In [182…

```r
edger_goal1 <- read.csv("DEG_All_KO_vs_All_WT.csv", row.names = 1)
# Sort by PValue and get top 20
edger_goal1_top20 <- edger_goal1 %>%
  arrange(PValue) %>%
  head(20) %>%
  select(PValue)
```

In [183…
```
head(edger_goal1_top20)
```

A data.frame: 6 × 1

	PValue <dbl>
ENSG00000117245	3.084251e-06
ENSG00000173261	4.113282e-06
ENSG00000228793	4.232953e-06
ENSG00000108551	5.359165e-06
ENSG00000184271	7.521795e-06
ENSG00000070669	1.004804e-05

In [181…
```
#genes_edger_goal1_top20 <- edger_goal1_top20$Gene
#genes_edger_goal1_top20
```

NULL

In [184…
```
edger_goal1_top20 <- data.frame(GeneID = rownames(edger_goal1_top20), PValue = edge
write.csv(edger_goal1_top20, "Top20_edgeR_All_KO_vs_WT.csv", row.names = FALSE)
```

In [185…
```
# Load t-test results
ttest_goal1 <- read.csv("TTest_All_KO_vs_WT_TPM.csv", row.names = 1)
# Sort by PValue and get top 20
ttest_goal1_top20 <- ttest_goal1 %>%
  arrange(PValue) %>%
  head(20) %>%
  select(PValue)
```

In [186…
```
head(ttest_goal1_top20)
```

A data.frame: 6 × 1

	PValue <dbl>
ENSG00000008324	0.0001198973
ENSG00000164919	0.0001521374
ENSG00000165490	0.0001761297
ENSG00000178449	0.0001773089
ENSG00000120820	0.0001787403
ENSG00000153485	0.0001862630

In [187…
```
#genes_ttest_goal1_top20 <- ttest_goal1_top20$Gene
#genes_ttest_goal1_top20
```

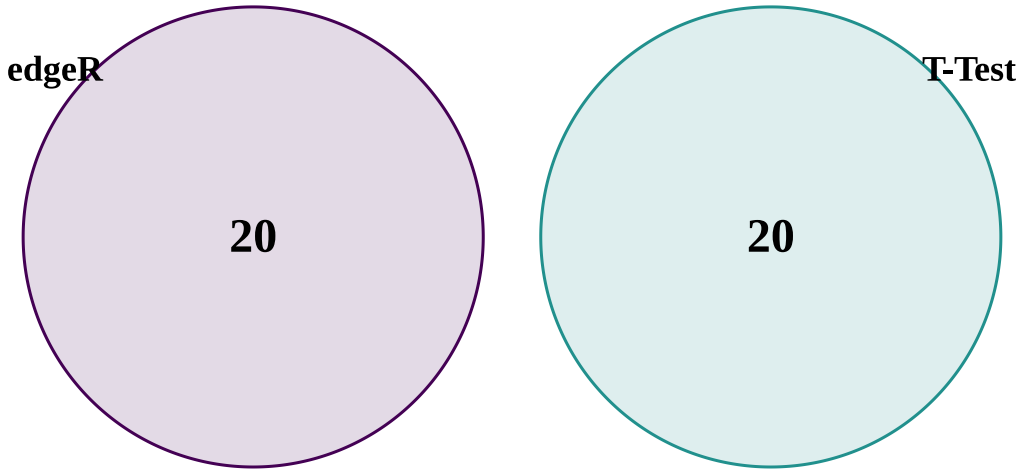
In [188…

```
# Add gene ID as a column
ttest_goal1_top20 <- data.frame(GeneID = rownames(ttest_goal1_top20), PValue = ttes
write.csv(ttest_goal1_top20, "Top20_TTest_All_KO_vs_WT_TPM.csv", row.names = FALSE)
```

In [189…

```
# Venn diagram
venn_goal1 <- venn.diagram(
  x = list(
    edgeR = edger_goal1_top20$GeneID,
    TTest = ttest_goal1_top20$GeneID
  ),
  category.names = c("edgeR", "T-Test"),
  filename = NULL,  # No file saving
  output = TRUE,
  main = "Goal 1: Top 20 DEGs (All KO vs. WT)",
  col = c("#440154ff", "#21908dff"),  # Colors for edgeR and T-Test
  fill = c(alpha("#440154ff", 0.3), alpha("#21908dff", 0.3)),
  cex = 2,
  fontface = "bold",
  cat.cex = 1.5,
  cat.fontface = "bold"
)

grid.draw(venn_goal1)
```

Goal 1: Top 20 DEGs (All KO vs. WT)



In [190…
```r
# --- Goal 2: MDX1 KO vs. MDX1 WT ---
# Load edgeR results
edger_goal2 <- read.csv("DEG_MDX1_KO_vs_MDX1_WT.csv", row.names = 1)
# Sort by PValue and get top 20
edger_goal2_top20 <- edger_goal2 %>%
  arrange(PValue) %>%
  head(20) %>%
  select(PValue)
```

In [191…
```r
head(edger_goal2_top20)
```

A data.frame: 6 × 1

	PValue
	<dbl>
ENSG00000288709	2.292324e-11
ENSG00000277150	1.955343e-10
ENSG00000260772	5.321627e-10
ENSG00000196436	8.526406e-03
ENSG00000289740	1.178506e-02
ENSG00000226686	1.707694e-02

In [193…
```r
#genes_edger_goal2_top20 <- edger_goal2_top20$Gene
#genes_edger_goal2_top20
```

In [194…
```r
edger_goal2_top20 <- data.frame(GeneID = rownames(edger_goal2_top20), PValue = edge
write.csv(edger_goal2_top20, "Top20_edgeR_MDX1_KO_vs_MDX1_WT.csv", row.names = FALS
```

In [195…
```r
# Load t-test results
ttest_goal2 <- read.csv("TTest_MDX1_KO_vs_MDX1_WT_TPM.csv", row.names = 1)
# Sort by PValue and get top 20
ttest_goal2_top20 <- ttest_goal2 %>%
  arrange(PValue) %>%
  head(20) %>%
  select(PValue)
```

In [196…
```r
head(ttest_goal2_top20)
```

A data.frame: 6 × 1

	PValue
	<dbl>
ENSG00000169100	0.003093085
ENSG00000270629	0.003308031
ENSG00000125375	0.006144911
ENSG00000086730	0.006927725
ENSG00000143450	0.009292603
ENSG00000115514	0.011029560

In [197…

```
#genes_ttest_goal2_top20 <- ttest_goal2_top20$Gene
#genes_ttest_goal2_top20
```

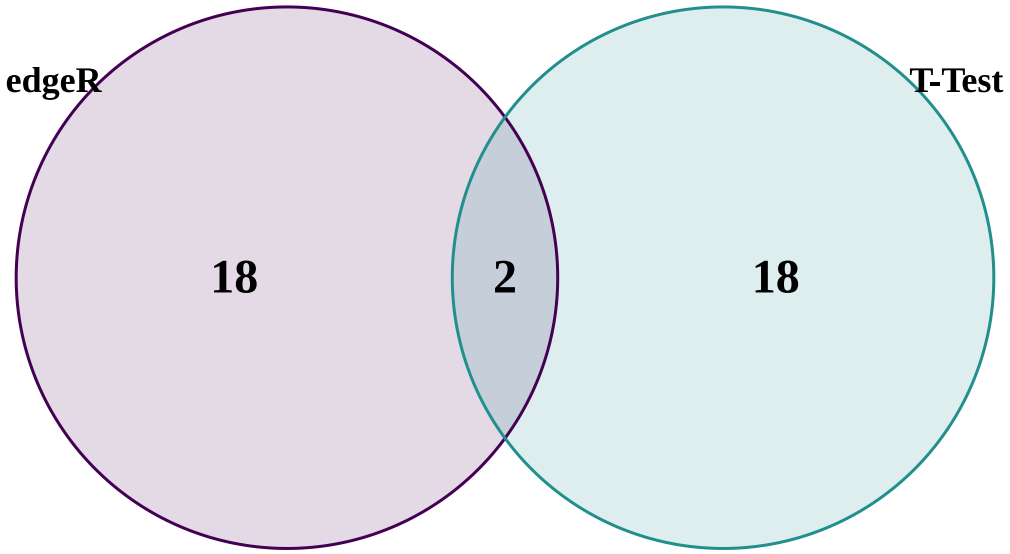
In [198…

```
ttest_goal2_top20 <- data.frame(GeneID = rownames(ttest_goal2_top20), PValue = ttes
write.csv(ttest_goal2_top20, "Top20_TTest_MDX1_KO_vs_MDX1_WT_TPM.csv", row.names =
```

In [199…

```
venn_goal2 <- venn.diagram(
  x = list(
    edgeR = edger_goal2_top20$GeneID,
    TTest = ttest_goal2_top20$GeneID
  ),
  category.names = c("edgeR", "T-Test"),
  filename = NULL,
  output = TRUE,
  main = "Goal 2: Top 20 DEGs (MDX1 KO vs. MDX1 WT)",
  col = c("#440154ff", "#21908dff"),
  fill = c(alpha("#440154ff", 0.3), alpha("#21908dff", 0.3)),
  cex = 2,
  fontface = "bold",
  cat.cex = 1.5,
  cat.fontface = "bold"
)

grid.draw(venn_goal2)
```

Goal 2: Top 20 DEGs (MDX1 KO vs. MDX1 WT)



Acknowledges references

1. edgeR Analysis (Raw Counts, DEG Calculation) Source: The edgeR workflow for differential expression analysis (used in your initial Goal 1 and Goal 2 analyses with GSE288289_study2_genesCounts.csv). Link: Official edgeR documentation: https://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuid Original Paper: Robinson, M.D., McCarthy, D.J., Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics, 26(1), 139-140. doi:10.1093/bioinformatics/btp616 (https://academic.oup.com/bioinformatics/article/26/1/139/182977) Influence: Steps like DGEList(), filterByExpr(), calcNormFactors(), estimateDisp(), and exactTest() follow the standard edgeR pipeline as outlined in the user guide.
2. T-Test Analysis (TPM Data) Source: R's base t.test() function for comparing TPM values between KO and WT groups. Link: R Documentation: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/t.test R Core Team (2023). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/ Influence: The t-test implementation with apply() to loop over genes and calculate logFC

(log2(mean(KO) / mean(WT))) is a standard approach, adapted from basic R statistical tutorials.

3. Data Manipulation (dplyr) Source: Used for sorting, filtering, and selecting data (e.g., top 20 genes by PValue). Link: Official dplyr documentation: https://dplyr.tidyverse.org/ Hadley Wickham et al., dplyr: A Grammar of Data Manipulation. https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html Influence: Functions like arrange(), head(), and select() were used as per dplyr's standard practices.
4. Visualization: ggplot2 (Volcano and MA Plots) Source: Early volcano and MA plots for edgeR results (e.g., March 11, 2025 response). Link: Official ggplot2 documentation: https://ggplot2.tidyverse.org/ Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. https://ggplot2-book.org/ Influence: Code like ggplot(aes(x = logFC, y = -log10(PValue))) + geom_point() follows ggplot2 examples from the documentation.
5. Visualization: pheatmap Source: Heatmaps for top DEGs (introduced later in the conversation). Link: Official pheatmap documentation: https://cran.r-project.org/web/packages/pheatmap/pheatmap.pdf Raivo Kolde (2019). pheatmap: Pretty Heatmaps. R package version 1.0.12. https://cran.r-project.org/package=pheatmap Influence: The pheatmap() function with clustering and annotations was adapted from the package's vignette.
6. Visualization: EnhancedVolcano Source: Enhanced volcano plots for both edgeR and t-test results. Link: Official EnhancedVolcano documentation: https://bioconductor.org/packages/release/bioc/vignettes/EnhancedVolcano/inst/doc/Enha Love, M.I., et al. (2019). EnhancedVolcano: Publication-ready volcano plots with enhanced colouring and labeling. Bioconductor package. https://bioconductor.org/packages/EnhancedVolcano/ Influence: The EnhancedVolcano() call with parameters like pCutoff and FCcutoff follows the package's examples.
7. Visualization: ComplexHeatmap Source: Advanced heatmaps with annotations and clustering. Link: Official ComplexHeatmap documentation: https://jokergoo.github.io/ComplexHeatmap-reference/book/ Gu, Z., Eils, R., Schlesner, M. (2016). Complex heatmaps reveal patterns and correlations in multidimensional genomic data. Bioinformatics, 32(18), 2847-2849. doi:10.1093/bioinformatics/btw313 (https://academic.oup.com/bioinformatics/article/32/18/2847/1743595) Influence: The Heatmap() function with column_split and top_annotation was inspired by the package's reference manual.
8. Visualization: VennDiagram Source: Venn diagrams comparing edgeR vs. t-test top 20 genes. Link: Official VennDiagram documentation: https://cran.r-project.org/web/packages/VennDiagram/VennDiagram.pdf Chen, H. (2018). VennDiagram: Generate High-Resolution Venn and Euler Plots. R package version 1.6.20. https://cran.r-project.org/package=VennDiagram Influence: The venn.diagram() function and styling options were adapted from the package's vignette.

9. General RNA-seq Workflow Source: Overall structure of loading data, filtering, analyzing, and visualizing RNA-seq data. Link: Bioconductor RNA-seq workflow: https://www.bioconductor.org/packages/devel/workflows/vignettes/rnaseqGene/inst/doc/rn Huber, W., et al. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. Nature Methods, 12(2), 115-121. doi:10.1038/nmeth.3252 (https://www.nature.com/articles/nmeth.3252) Influence: The general approach (e.g., filtering samples, normalizing, differential analysis) aligns with Bioconductor's recommended practices.
10. TPM Normalization Context Source: Understanding TPM data usage (per your professor's directive). Link: Wagner, G.P., Kin, K., Lynch, V.J. (2012). Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. Theory in Biosciences, 131(4), 281-285. doi:10.1007/s12064-012-0162-3 (https://link.springer.com/article/10.1007/s12064-012-0162-3) Influence: While we didn't normalize TPM ourselves (it was pre-normalized), this paper informed the context of using TPM for t-tests.

In [ ]: