A bio-functional polymer that prevents retinal scarring through modulation of NRF2 signalling pathway

```
library(GEOquery)
```

```
## Loading required package: Biobase
```

```
## 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, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

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

```
## Setting options('download.file.method.GEOquery'='auto')
```

```
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```
library(DESeq2)
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
##
## 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
```

```
## Warning: package 'IRanges' was built under R version 4.3.1
```

```
## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 4.3.1

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

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

##
## 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

## The following object is masked from 'package:Biobase':
##
##     rowMedians
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## 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 object is masked from 'package:Biobase':
##
##     combine

## 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
```

```
library(ComplexHeatmap)
```

```
## Loading required package: grid

## ========================================
## ComplexHeatmap version 2.16.0
## 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))
## ========================================
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.2     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::%within%() masks IRanges::%within%()
## x dplyr::collapse()     masks IRanges::collapse()
## x dplyr::combine()      masks Biobase::combine(), BiocGenerics::combine()
```

```
## x dplyr::count()        masks matrixStats::count()
## x dplyr::desc()         masks IRanges::desc()
## x tidyr::expand()       masks S4Vectors::expand()
## x dplyr::filter()       masks stats::filter()
## x dplyr::first()        masks S4Vectors::first()
## x dplyr::lag()          masks stats::lag()
## x ggplot2::Position()   masks BiocGenerics::Position(), base::Position()
## x purrr::reduce()       masks GenomicRanges::reduce(), IRanges::reduce()
## x dplyr::rename()       masks S4Vectors::rename()
## x lubridate::second()   masks S4Vectors::second()
## x lubridate::second<-() masks S4Vectors::second<-()
## x dplyr::slice()        masks IRanges::slice()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(vsn)
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following object is masked from 'package:purrr':
##
##     transpose
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:SummarizedExperiment':
##
##     shift
##
## The following object is masked from 'package:GenomicRanges':
##
##     shift
##
## The following object is masked from 'package:IRanges':
##
##     shift
##
## The following objects are masked from 'package:S4Vectors':
##
##     first, second
```

```
library(ggplot2)
library(RColorBrewer)
library(pheatmap)
```

```
##
## Attaching package: 'pheatmap'
```

```
##
## The following object is masked from 'package:ComplexHeatmap':
##
##     pheatmap
```

```
# Set the working directory
setwd("/Users/seymour/Desktop/Bioinformatics/Advanced_Computational_Biology_BIOL0050/Week3/Gene_Exp_anal
```

```
gse <- getGEO(filename = "GSE176513_series_matrix.txt.gz", GSEMatrix = TRUE, getGPL = FALSE)
```

```
# Create a data frame with samples and conditions
sample_info <- pData(gse)
condition_df <- data.frame(Sample_ID = sample_info$description, Condition_name = sample_info$title) %>%
  mutate(Sample_ID = gsub("gene_count_matrix.csv --", "", Sample_ID)) %>%
  mutate(Condition = substr(Sample_ID, 1, 1)) %>%
  mutate(Duration = substr(Sample_ID, 4, nchar(Sample_ID))) %>%
  mutate(cond_duration = paste(Condition, Duration, sep = "_"))  %>%
  mutate(Condition_name = substr(Condition_name, 1, nchar(Condition_name) - 4)) %>%
  mutate(Condition_name_short = gsub(" \\(8h\\)| \\(24h\\)", "", Condition_name))
```

```
#Separate the 8h and 24h samples
condition_df_8h <- condition_df[condition_df$Duration == "8h", ]
condition_df_24h <- condition_df[condition_df$Duration == "24h", ]
```

```
#Read in the count data
count_data <- read.csv(gzfile("GSE176513_gene_count_matrix.csv.gz"))
```

```
col_order<- condition_df$Sample_ID
rownames(count_data) <- count_data$EnsemblID
aligned_count_data <- count_data[, col_order]
```

```
all_dds <- DESeqDataSetFromMatrix(countData = aligned_count_data, colData = condition_df, design = ~ Co
```

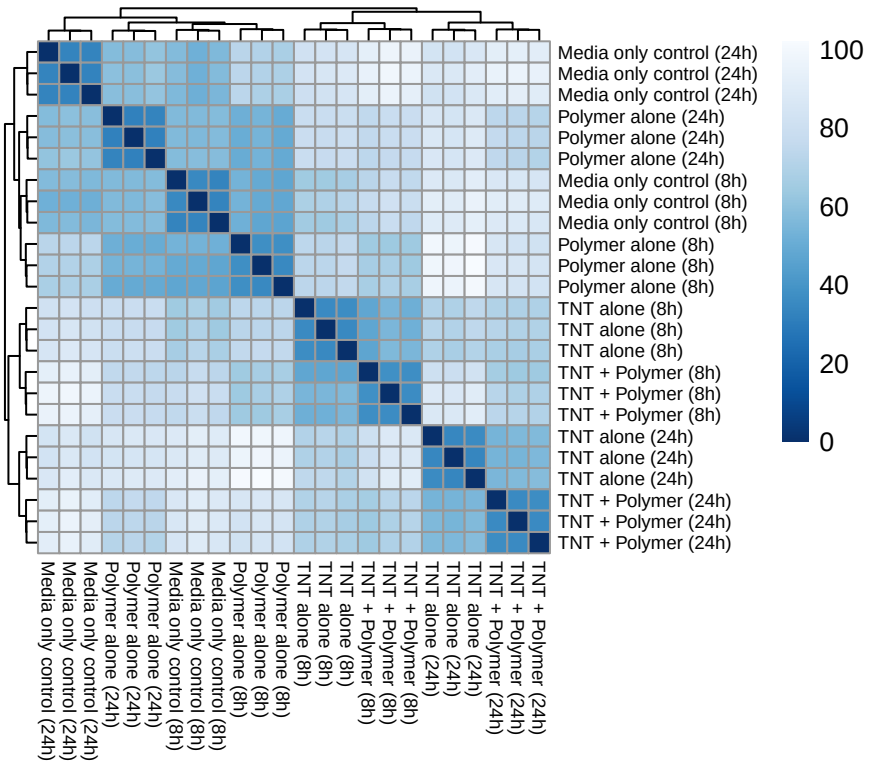
```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
all_dds <- all_dds[ rowSums(counts(all_dds)) > 1, ]
all_dds <- estimateSizeFactors(all_dds)
```

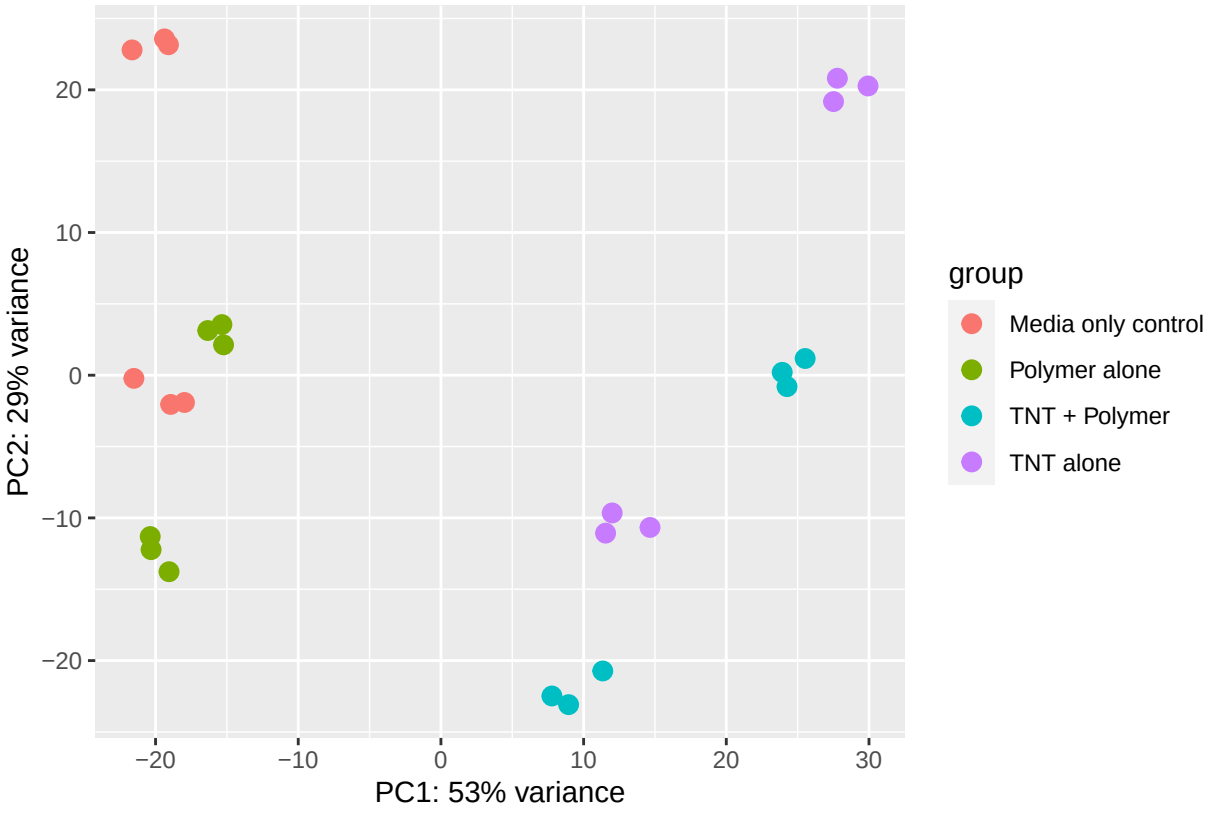
```
vsd <- vst(all_dds, blind = FALSE)
sampleDists_vsd <- dist(t(assay(vsd)))
sampleDistMatrix_vsd <- as.matrix( sampleDists_vsd )
rownames(sampleDistMatrix_vsd) <- paste(vsd$Condition_name)
colnames(sampleDistMatrix_vsd) <- paste(vsd$Condition_name)
```

Distance matix of gene expression profiles from RNA sequencing of all four groups (Media only, 1 wt% poly(CEP), TNT and 1 wt% poly(CEP) + TNT at two time points (8 and 24 h), each done in triplicates. 24 h), each done in triplicates. The intensity of each box corresponds to the distance value shown in the legend.

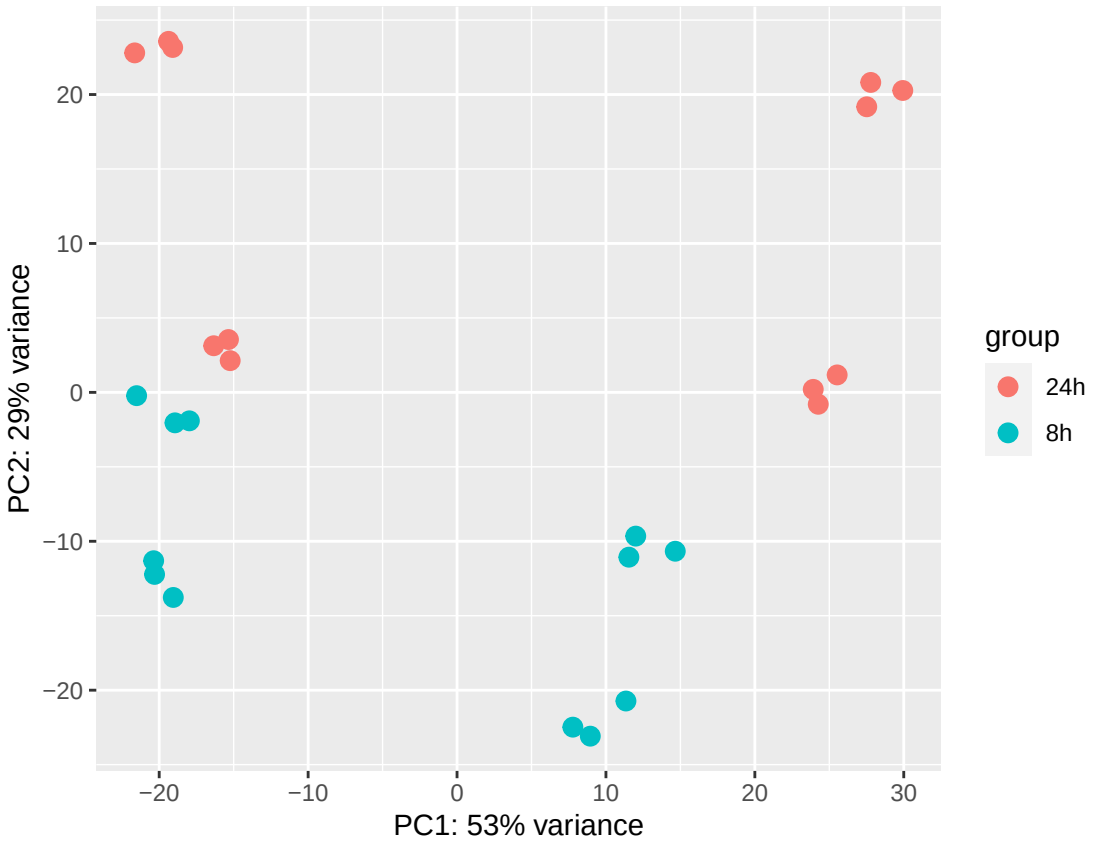
```
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
heatmap <- pheatmap(sampleDistMatrix_vsd,
                    clustering_distance_rows = sampleDists_vsd,
                    clustering_distance_cols = sampleDists_vsd,
                    col = colors,cellwidth=8,
                    cellheight=8,treeheight_row = 8, treeheight_col = 8, fontsize_row = 7,fontsize_col =
```




```
pca_data_condition<-plotPCA(vsd, intgroup="Condition_name_short")
pca_data_condition
```



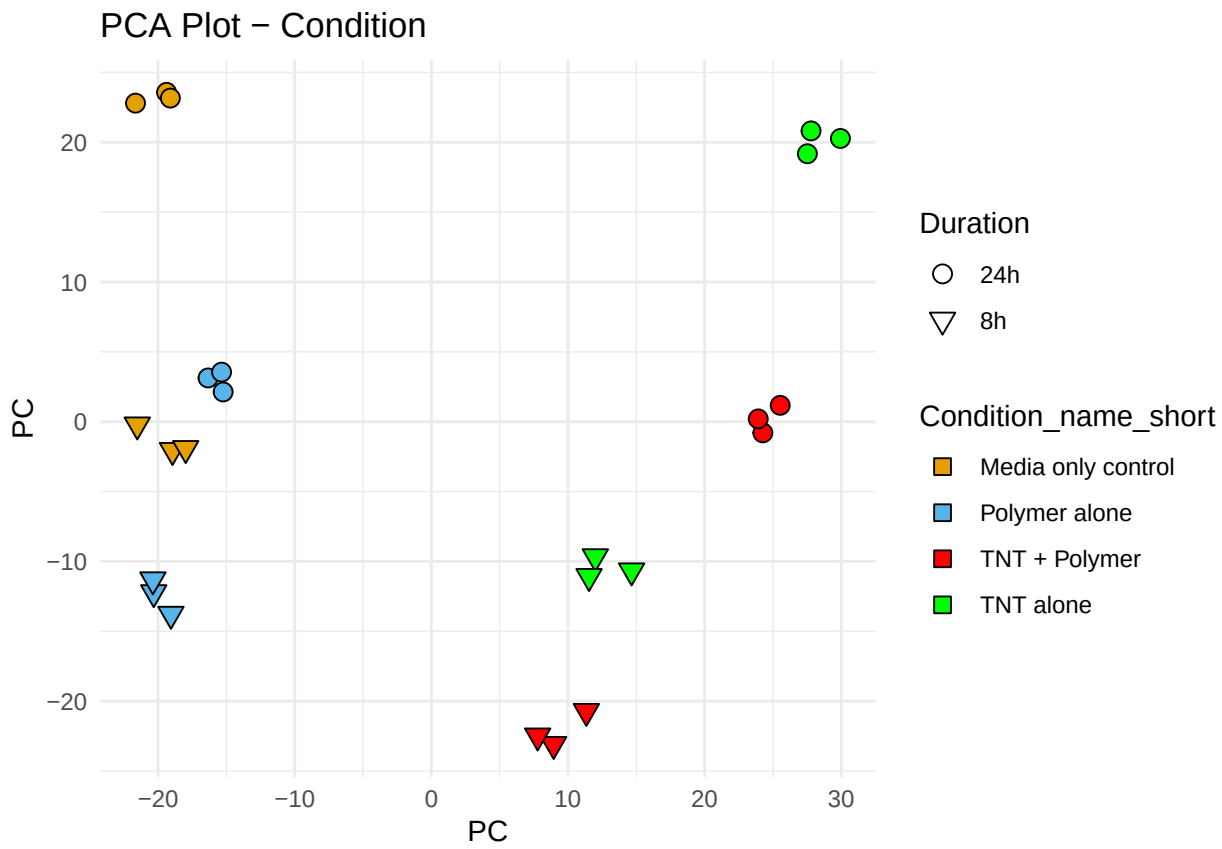

```
pca_data_duration<-plotPCA(vsd, intgroup="Duration")
pca_data_duration
```



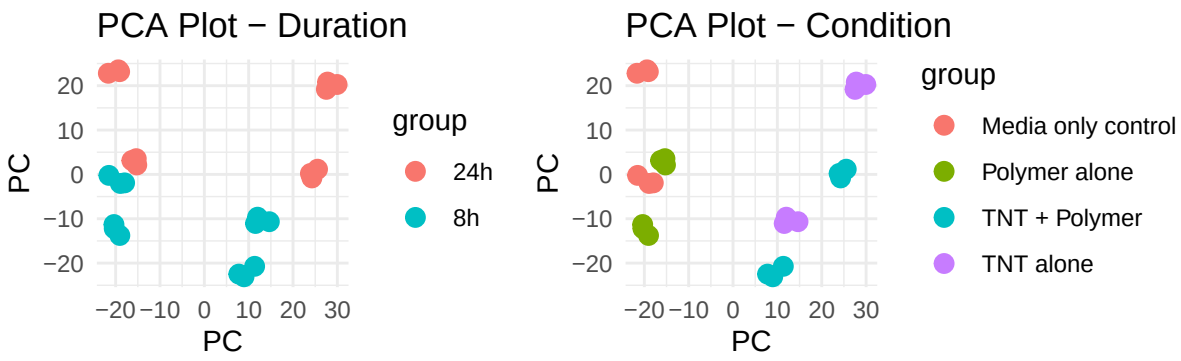

```
library(ggfortify)
library(patchwork)

pca_data_duration <- plotPCA(vsd, intgroup = "Duration")
pca_data_condition <- plotPCA(vsd, intgroup = "Condition_name_short")
match(pca_data_duration$data$name,pca_data_condition$data$name)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
```

```
pca_data_condition$data$Duration<-pca_data_duration$data$Duration
pcaData<-pca_data_condition$data
ggplot(pcaData, aes(x= PC1, y = PC2))+geom_point(size= 3, aes(shape=Duration, fill=Condition_name_short)
```




```
plot_duration <- autoplot(pca_data_duration, data = vsd$Duration, shape  = "Duration") +
  labs(x = paste0("PC", pca_data_duration$x[, 1]), y = paste0("PC", pca_data_duration$x[, 2])) +
  ggtitle("PCA Plot - Duration") +
  theme_minimal()

# PCA plot with intgroup="Condition_name_short"
pca_data_condition <- plotPCA(vsd, intgroup = "Condition_name_short")
plot_condition <- autoplot(pca_data_condition, data = vsd$Condition_name_short, colour = "Condition_nam
  labs(x = paste0("PC", pca_data_condition$x[, 1]), y = paste0("PC", pca_data_condition$x[, 2])) +
  ggtitle("PCA Plot - Condition") +
  theme_minimal()

# Combine the two plots
combined_plot <- plot_duration + plot_condition

# Display the combined plot
print(combined_plot)
```




```
#Create a function to investigate number of significant genes that are differentially expressed between
create_dds <- function(condition_xh) {
  col_order <- condition_xh$Sample_ID
  aligned_count_data <- count_data[, col_order]
  dds <- DESeqDataSetFromMatrix(countData = aligned_count_data, colData = condition_xh, design = ~ Condi
  dds <- dds[ rowSums(counts(dds)) > 1, ]
  dds <- estimateSizeFactors(dds)

  dds <- DESeq(dds)

  res<-results(dds, contrast=c("Condition","A","B"))
  sigs<-na.omit(res)
  sigs<-sigs[sigs$padj<0.05,]
  cat("Number of genes sig in poly(CEP) alone compared to media only:", nrow(sigs), "\n")
  #print(nrow(sigs))

  res1<-results(dds, contrast=c("Condition","C","D"))
  sigs1<-na.omit(res1)
  sigs1<-sigs1[sigs1$padj<0.05,]
  cat("Number of genes sig in poly(CEP)+TNT compared to TNT:", nrow(sigs1), "\n")

  return(dds)
}
```

```
dds_8h <- create_dds(condition_df_8h)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
```

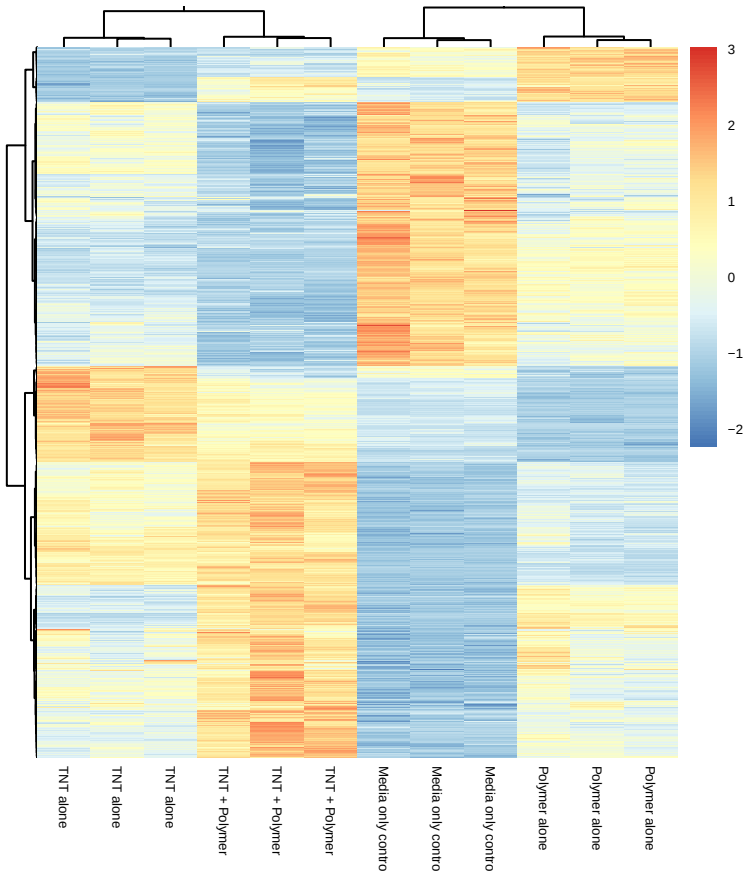
```
## design formula are characters, converting to factors
## using pre-existing size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## Number of genes sig in poly(CEP) alone compared to media only: 3472
## Number of genes sig in poly(CEP)+TNT compared to TNT: 3790
```

```
dds_24h <- create_dds(condition_df_24h)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
## using pre-existing size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## Number of genes sig in poly(CEP) alone compared to media only: 6646
## Number of genes sig in poly(CEP)+TNT compared to TNT: 5162
```

```
res_8h1<-results(dds_8h,contrast=c("Condition","B","A"))
res_8h2<-results(dds_8h,contrast=c("Condition","C","A"))
res_8h3<-results(dds_8h,contrast=c("Condition","D","A"))

create_sigs_df <- function(res_xh) {
  sigs_xh <- na.omit(res_xh)
  sigs_xh <- sigs_xh[sigs_xh$padj < 0.05,]
  df_sigs_xh <- as.data.frame(sigs_xh)
  return(df_sigs_xh)
}

sigs_8h1 <- create_sigs_df(res_8h1)
sigs_8h2 <- create_sigs_df(res_8h2)
sigs_8h3 <- create_sigs_df(res_8h3)

mat_8h1<-counts(dds_8h,normalized=T)[rownames(sigs_8h1),]
common_row_names1 <- intersect(rownames(mat_8h1), rownames(sigs_8h2))

mat_8h2<-mat_8h1[common_row_names1,]
common_row_names2 <- intersect(rownames(mat_8h2), rownames(sigs_8h3))

mat_8h3<-mat_8h2[common_row_names2,]
z_mat_8h3<-t(apply(mat_8h3,1,scale))
```

```
colnames(z_mat_8h3) <- condition_df_8h$Condition_name_short
pheatmap(z_mat_8h3,clustering_distance_rows = "euclidean",clustering_distance_cols = "euclidean",cluste
```




```
# Your existing lists
Media_data <- rownames(sigs_8h1)
Polymer_data <- rownames(sigs_8h1)

print(length(Media_data))
```

```
## [1] 3472
```

```
TNT_data <- rownames(sigs_8h2)
Polymer_and_TNT_data <- rownames(sigs_8h3)

# Update Media with new rownames from sigs_8h2 and sigs_8h3
Media_data <- union(Media_data, TNT_data)
print(length(Media_data))
```

```
## [1] 7543
```

```
Media_data <- union(Media_data, Polymer_and_TNT_data)
print(length(Media_data))
```

```
## [1] 9358
```

```
venn_df_Media <- data.frame(Media=Media_data)
venn_df_Polymer <- data.frame(Polymer=Polymer_data)
```

```
venn_df_Media$RowName <- venn_df_Media$Media
venn_df_Polymer$RowName <-venn_df_Polymer$Polymer

dat1 <- merge(venn_df_Media , venn_df_Polymer, by="RowName", all = T)


venn_df_TNT <- data.frame(TNT=TNT_data)
venn_df_Polymer_and_TNT <- data.frame(Polymer_and_TNT=Polymer_and_TNT_data)

venn_df_TNT$RowName <- venn_df_TNT$TNT
venn_df_Polymer_and_TNT$RowName <-venn_df_Polymer_and_TNT$Polymer_and_TNT

dat2 <- merge(venn_df_TNT, venn_df_Polymer_and_TNT, by="RowName", all = T)
dat3 <- merge(dat1, dat2, by="RowName", all = T)

library(VennDiagram)
```

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

```
#venn.diagram(dat3, filename = "venn-4-dimensions.png")

#Error in `[[<-.data.frame`(`*tmp*`, i, value = c(NA, "ENSG00000000460", :
#replacement has 3473 rows, data has 9358
```

```
write.csv(dat3, file = "dat3.csv", row.names = FALSE)
```

```
prepro_res <- function(res_xh) {
  sigs_xh <-res_xh
  sigs_xh <- na.omit(res_xh)
  #sigs_xh <- sigs_xh[sigs_xh$padj < 0.05,]
  df_sigs_xh <- as.data.frame(sigs_xh)
  return(df_sigs_xh)
}

volc_8h1 <- prepro_res(res_8h1)
volc_8h2 <- prepro_res(res_8h2)
volc_8h3 <- prepro_res(res_8h3)
```

```
#Fucntion to plot volcano plot
make_volc_plot <- function(volc_xhx,plot_name) {
  sigs_xhx_results <- volc_xhx[order(volc_xhx$pvalue),]
  results_order <- as.data.frame(dplyr::mutate(as.data.frame(sigs_xhx_results), sig=ifelse(sigs_xhx_resu

  volcanoP <- ggplot(results_order, ggplot2::aes(log2FoldChange, -log10(pvalue))) +
    geom_point(ggplot2::aes(col = sig)) +
    scale_color_manual(values = c("red", "black")) +
    ggtitle(plot_name)

  volcanoP
}
```

```
make_volc_plot(volc_8h1,"Media vs Polymer")
```

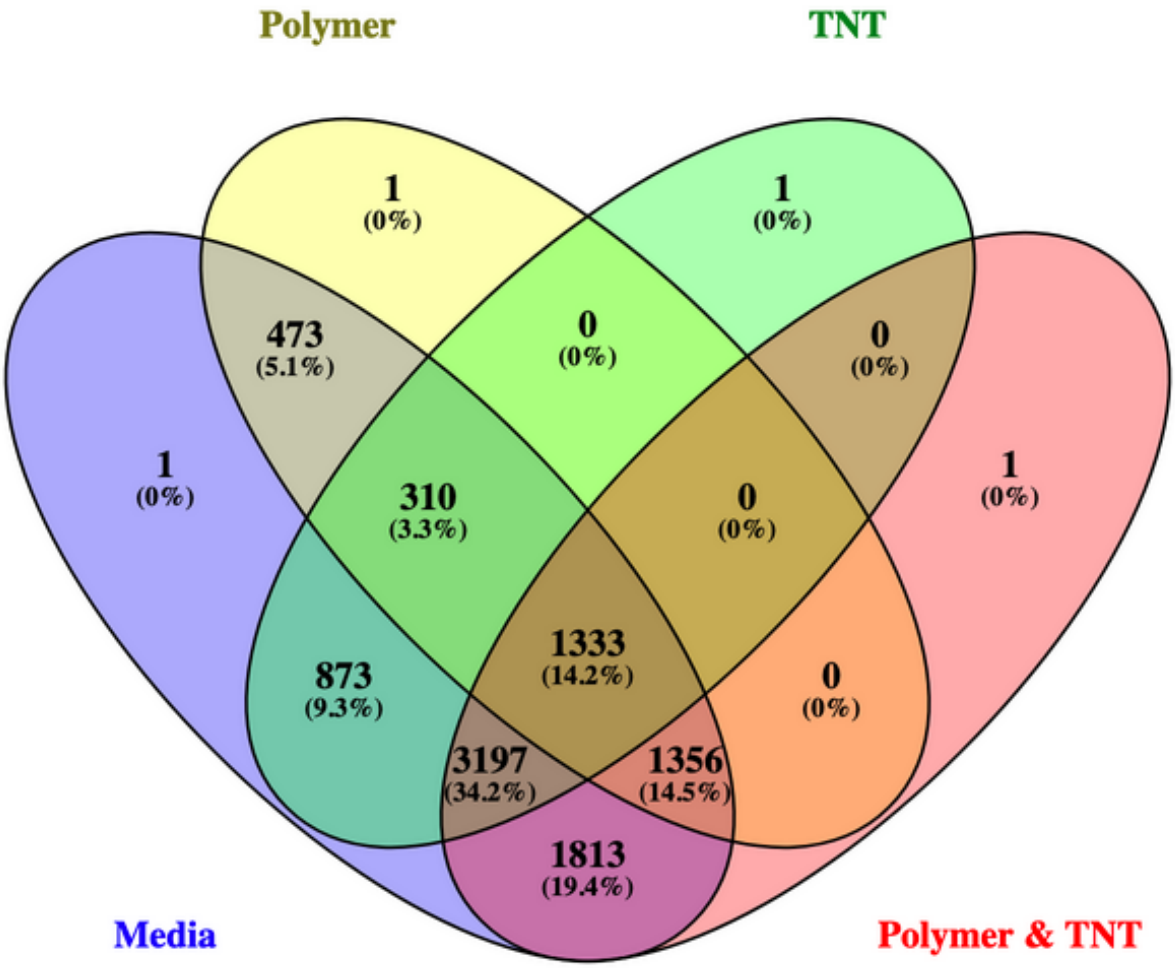
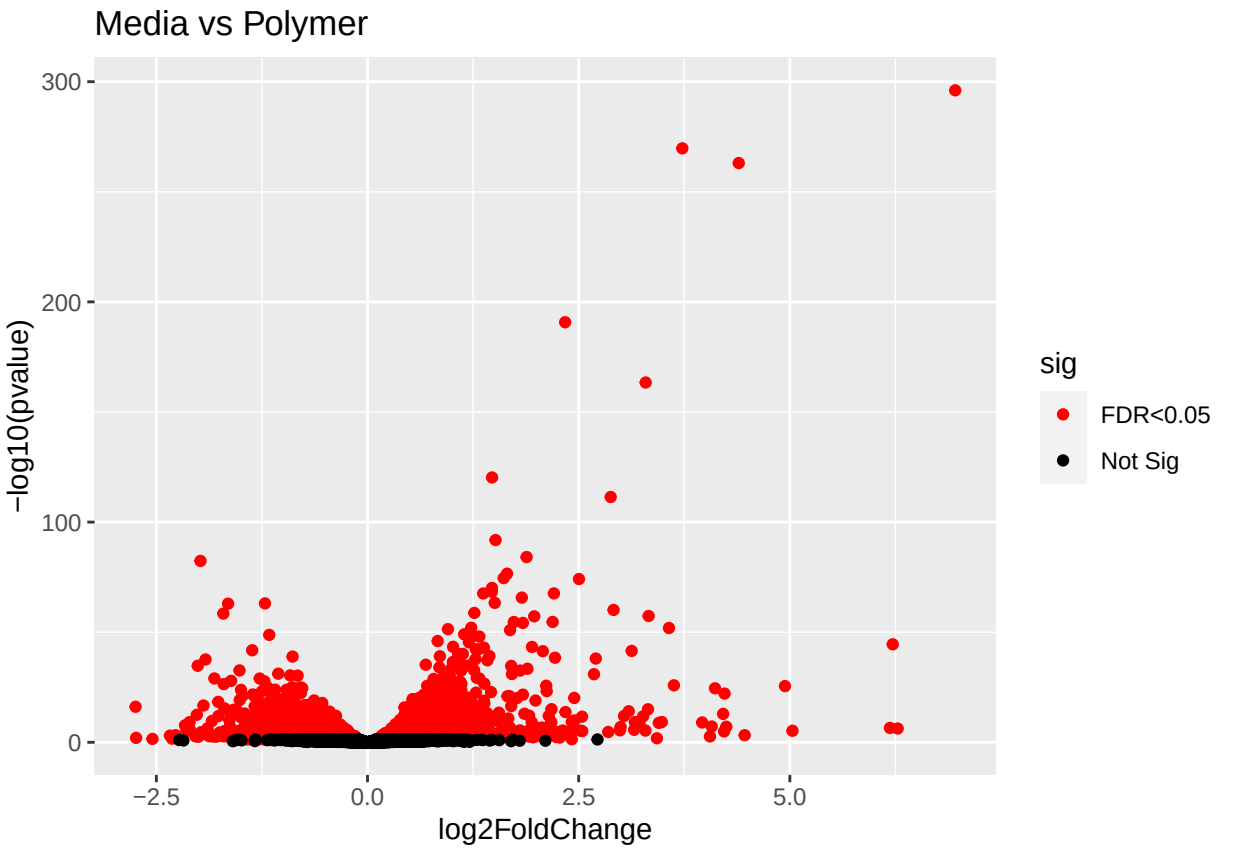
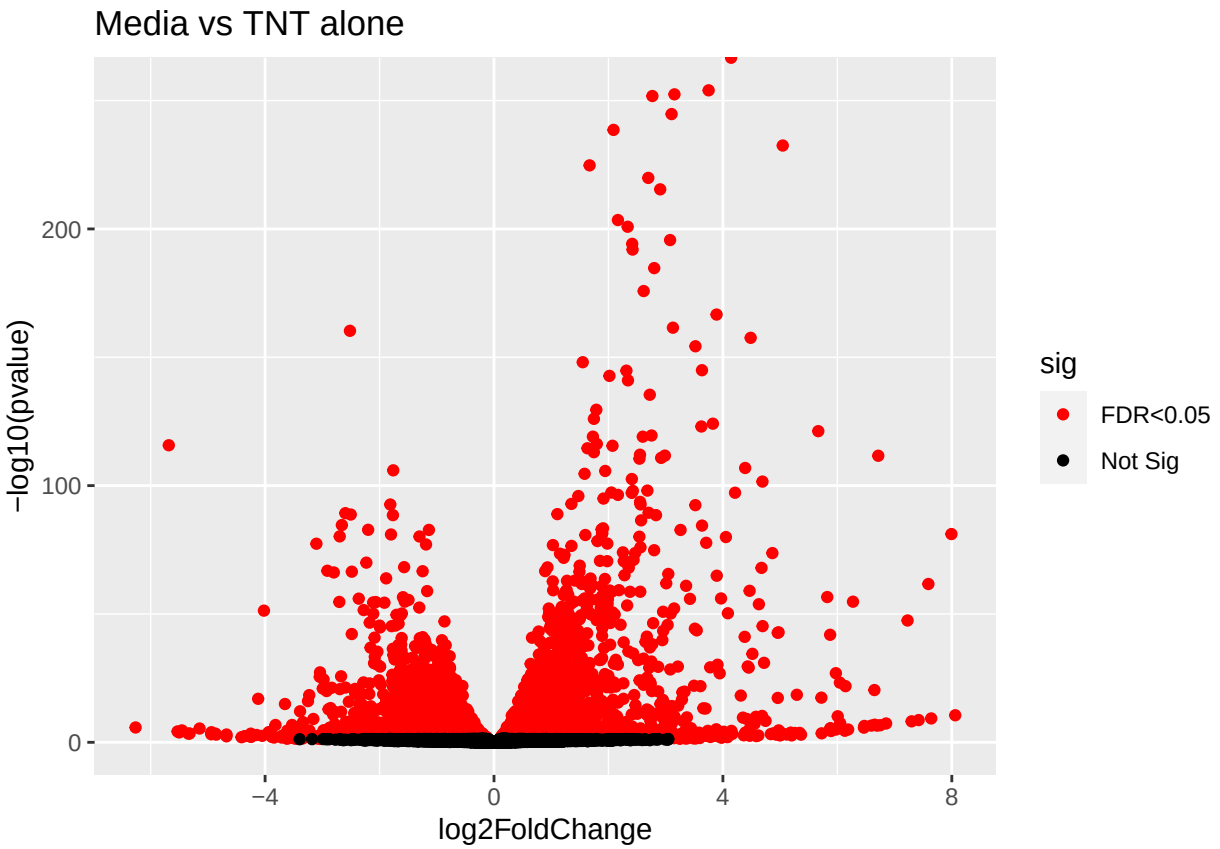

Polymer
TNT
1
(0%)
1
(0%)
473
(5.1%)
0
(0%)
0
(0%)
1
(0%)
310
(3.3%)
0
(0%)
1
(0%)
1333
(14.2%)
873
(9.3%)
0
(0%)
3197
(34.2%)
1356
(14.5%)
1813
(19.4%)
Media
Polymer & TNT

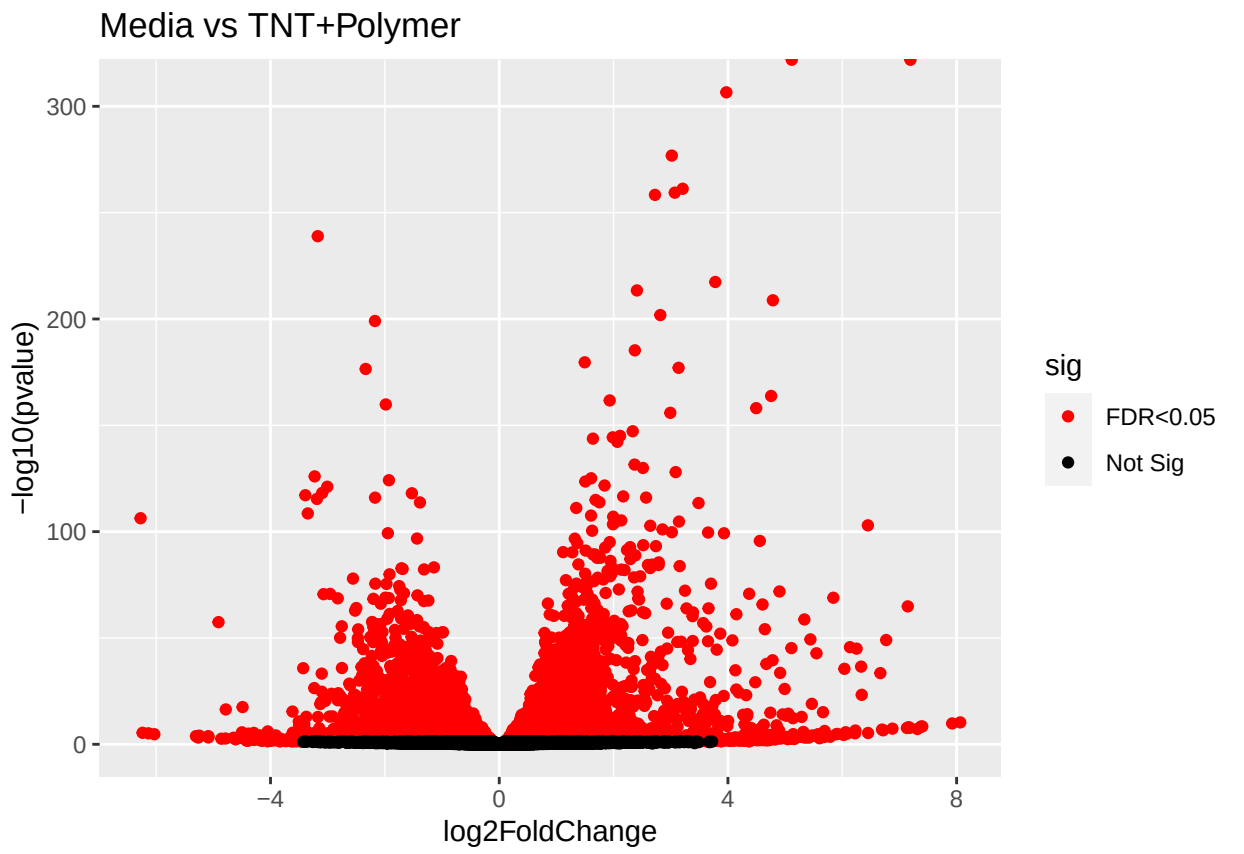
Figure 1: Fig




```
make_volc_plot(volc_8h2,"Media vs TNT alone")
```




```
make_volc_plot(volc_8h3,"Media vs TNT+Polymer ")
```




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

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

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

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

```
##
```

```
library(fgsea)

pathways.hallmark <- gmtPathways("h.all.v7.2.symbols.gmt")

plot_pathways<-function(res_xhx,plot_title){

  ens2symbol <- AnnotationDbi::select(org.Hs.eg.db,
                                      key=rownames(res_xhx),
                                      columns="SYMBOL",
                                      keytype="ENSEMBL")
  ens2symbol <- as_tibble(ens2symbol)

  sigs_xhx_results <- res_xhx[order(res_xhx$pvalue),]
  sigs_xhx_results$ENSEMBL <- rownames(sigs_xhx_results)
  sigs_xhx_results <- inner_join(sigs_xhx_results, ens2symbol, by=c("ENSEMBL"="ENSEMBL"))
```

```r
sigs_xhx_results <- sigs_xhx_results %>%
  mutate(updown = ifelse(log2FoldChange > 0, "upregulated", "downregulated"))

res2 <- sigs_xhx_results %>%
dplyr::select(SYMBOL, stat,updown) %>%
na.omit() %>%
distinct() %>%
group_by(SYMBOL) %>%
summarize(stat=mean(stat), updown = first(updown))

ranks <- deframe(res2)

fgseaRes <- fgseaMultilevel(pathways=pathways.hallmark, stats=ranks)

# Tidy the results:
fgseaResTidy <- fgseaRes %>%
  as_tibble() %>%
  arrange(desc(NES)) # order by normalized enrichment score (NES)

# To see what genes are in each of these pathways:
gene.in.pathway <- pathways.hallmark %>%
  enframe("pathway", "SYMBOL") %>%
  unnest(cols = c(SYMBOL)) %>%
  inner_join(res2, by="SYMBOL")

  # Merge "updown" information with gene.in.pathway
gene.in.pathway <- gene.in.pathway %>%
  inner_join(res2, by = "SYMBOL")

# Merge the enriched pathways information with fgseaResTidy
fgseaResTidy <- fgseaResTidy %>%
  inner_join(gene.in.pathway, by = "pathway")

 # Define the size breaks for the number of genes
size_breaks <- c(10, 50, 100, 200,300, Inf)

# Calculate the circle size based on the number of genes in each pathway
fgseaResTidy$gene_count_group <- cut(fgseaResTidy$size, breaks = size_breaks,
                                     labels = c("10", "50", "100", "200","300"))

# Create a color palette for upregulation/downregulation
colors <- c("downregulated" = "blue", "upregulated" = "brown")

# Map colors to upregulation/downregulation
fgseaResTidy$circle_color <- colors[fgseaResTidy$updown.x]

# Create the plot
ggplot(fgseaResTidy, aes(-log10(padj), reorder(pathway, NES))) +
  geom_point(aes(color = circle_color, size = gene_count_group)) +
  scale_color_manual(values = c("blue", "brown")) +
  scale_size_manual(values = c("10" = 1,  "50" = 1.6, "100" = 2.4, "200" = 5, "300" = 6)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1, size = 6),
```

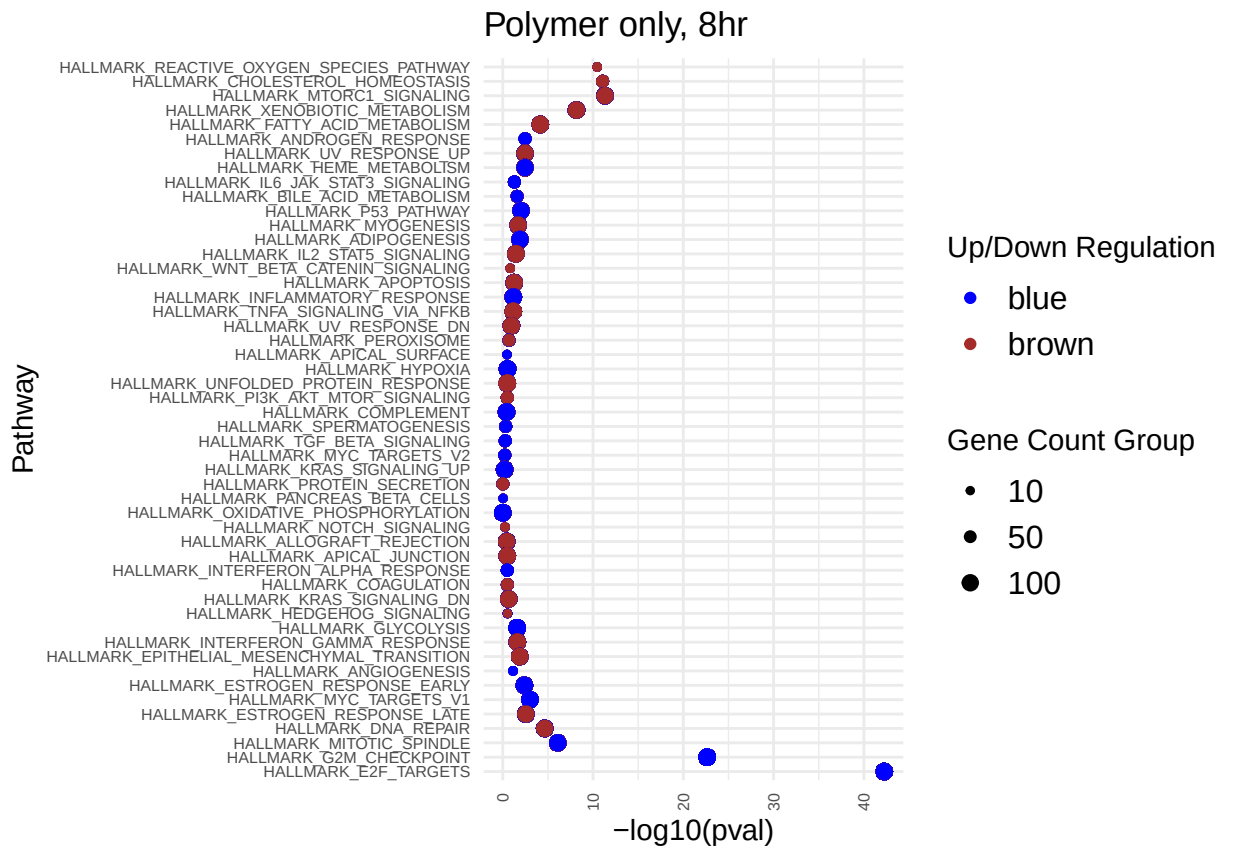
```
          axis.text.y = element_text(size = 6),
    legend.text = element_text(size = 12))+
    labs(x = "-log10(pval)", y = "Pathway",
         title = plot_title) +
    guides(size = guide_legend(title = "Gene Count Group"),
           color = guide_legend(title = "Up/Down Regulation"))
}
```

```
plot_pathways(volc_8h1,"Polymer only, 8hr")
```

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

```
## Warning: `x` must be a one- or two-column data frame in `deframe()`.
```

```
## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize, gseaParam, : There are ties in
## The order of those tied genes will be arbitrary, which may produce unexpected results.
```

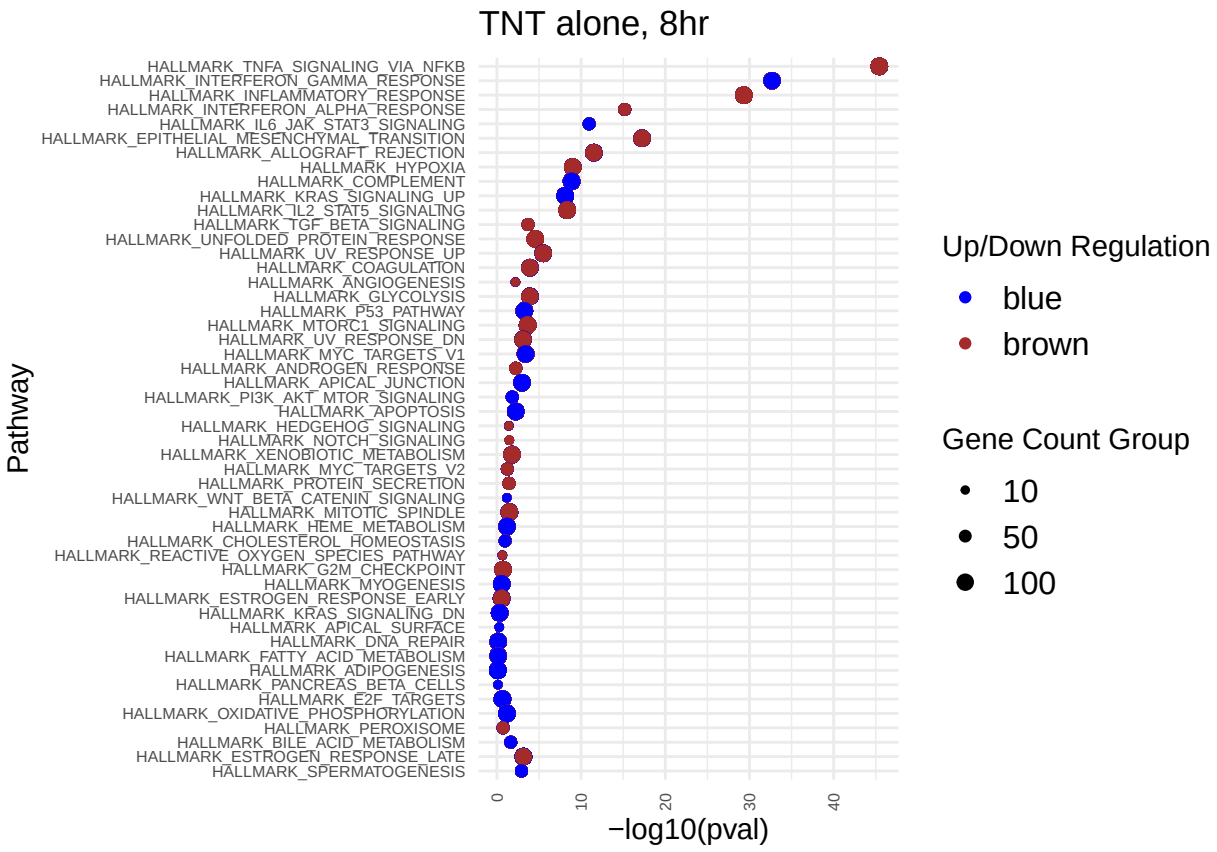



```
plot_pathways(volc_8h2,"TNT alone, 8hr")
```

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

```
## Warning: `x` must be a one- or two-column data frame in `deframe()`.
```

```
## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize, gseaParam, : There are ties in
## The order of those tied genes will be arbitrary, which may produce unexpected results.
```

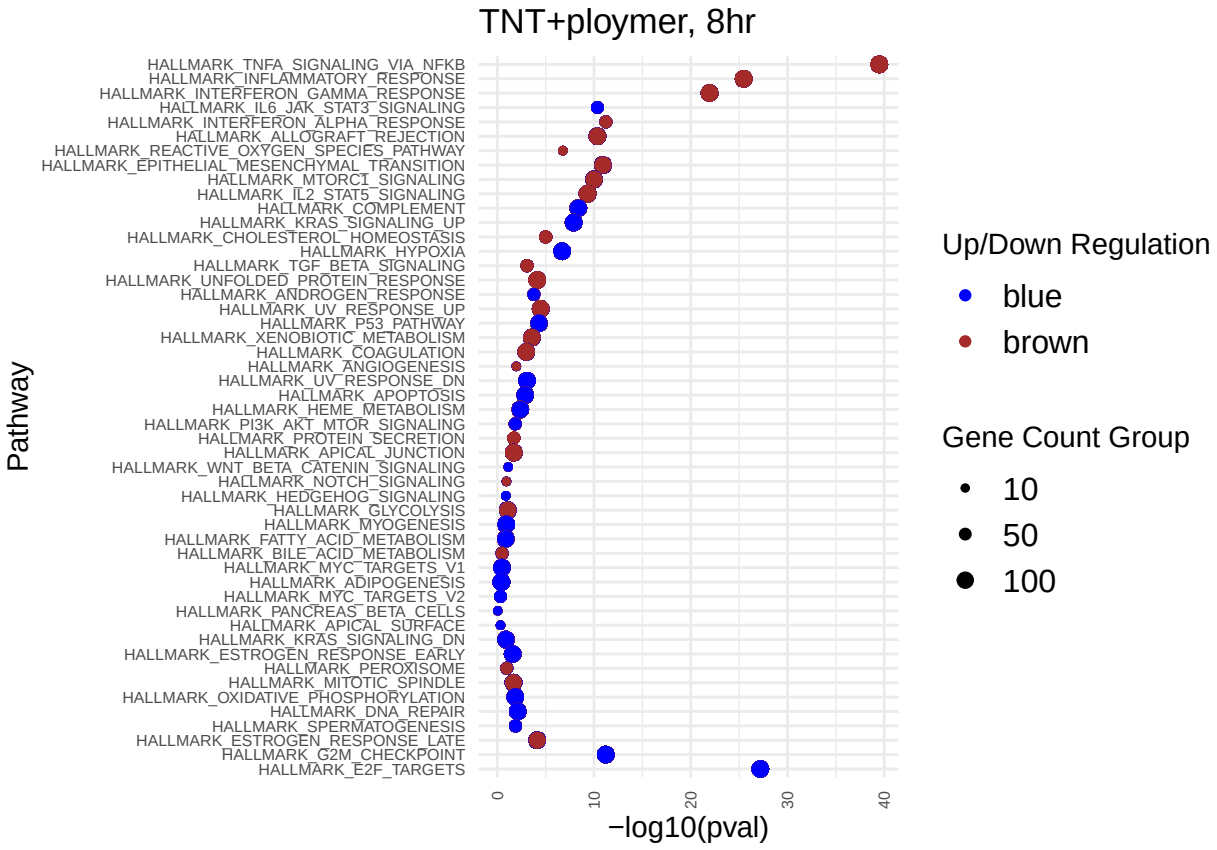



```
plot_pathways(volc_8h3,"TNT+ploymer, 8hr")
```

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

```
## Warning: `x` must be a one- or two-column data frame in `deframe()`.
```

```
## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize, gseaParam, : There are ties in
## The order of those tied genes will be arbitrary, which may produce unexpected results.
```


TNT+ploymer, 8hr
HALLMARK_TNFA_SIGNALING_VIA_NFKB
HALLMARK_INFLAMMATORY_RESPONSE
HALLMARK_INTERFERON_GAMMA_RESPONSE
HALLMARK_IL6_JAK_STAT3_SIGNALING
HALLMARK_INTERFERON_ALPHA_RESPONSE
HALLMARK_ALLOGRAFT_REJECTION
HALLMARK_REACTIVE_OXYGEN_SPECIES_PATHWAY
HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION
HALLMARK_MTORC1_SIGNALING
HALLMARK_IL2_STAT5_SIGNALING
HALLMARK_COMPLEMENT
HALLMARK_KRAS_SIGNALING_UP
HALLMARK_CHOLESTEROL_HOMEOSTASIS
HALLMARK_HYPOXIA
HALLMARK_TGF_BETA_SIGNALING
HALLMARK_UNFOLDED_PROTEIN_RESPONSE
HALLMARK_ANDROGEN_RESPONSE
HALLMARK_UV_RESPONSE_UP
HALLMARK_P53_PATHWAY
HALLMARK_XENOBIOTIC_METABOLISM
HALLMARK_COAGULATION
HALLMARK_ANGIOGENESIS
HALLMARK_UV_RESPONSE_DN
HALLMARK_APOPTOSIS
HALLMARK_HEME_METABOLISM
HALLMARK_PI3K_AKT_MTOR_SIGNALING
HALLMARK_PROTEIN_SECRETION
HALLMARK_APICAL_JUNCTION
HALLMARK_WNT_BETA_CATENIN_SIGNALING
HALLMARK_NOTCH_SIGNALING
HALLMARK_HEDGEHOG_SIGNALING
HALLMARK_GLYCOLYSIS
HALLMARK_MYOGENESIS
HALLMARK_FATTY_ACID_METABOLISM
HALLMARK_BILE_ACID_METABOLISM
HALLMARK_MYC_TARGETS_V1
HALLMARK_ADIPOGENESIS
HALLMARK_MYC_TARGETS_V2
HALLMARK_PANCREAS_BETA_CELLS
HALLMARK_APICAL_SURFACE
HALLMARK_KRAS_SIGNALING_DN
HALLMARK_ESTROGEN_RESPONSE_EARLY
HALLMARK_PEROXISOME
HALLMARK_MITOTIC_SPINDLE
HALLMARK_OXIDATIVE_PHOSPHORYLATION
HALLMARK_DNA_REPAIR
HALLMARK_SPERMATOGENESIS
HALLMARK_ESTROGEN_RESPONSE_LATE
HALLMARK_G2M_CHECKPOINT
HALLMARK_E2F_TARGETS
Pathway
0
10
20
30
40
−log10(pval)
Up/Down Regulation
blue
brown
Gene Count Group
10
50
100