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_an
```

```
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
```

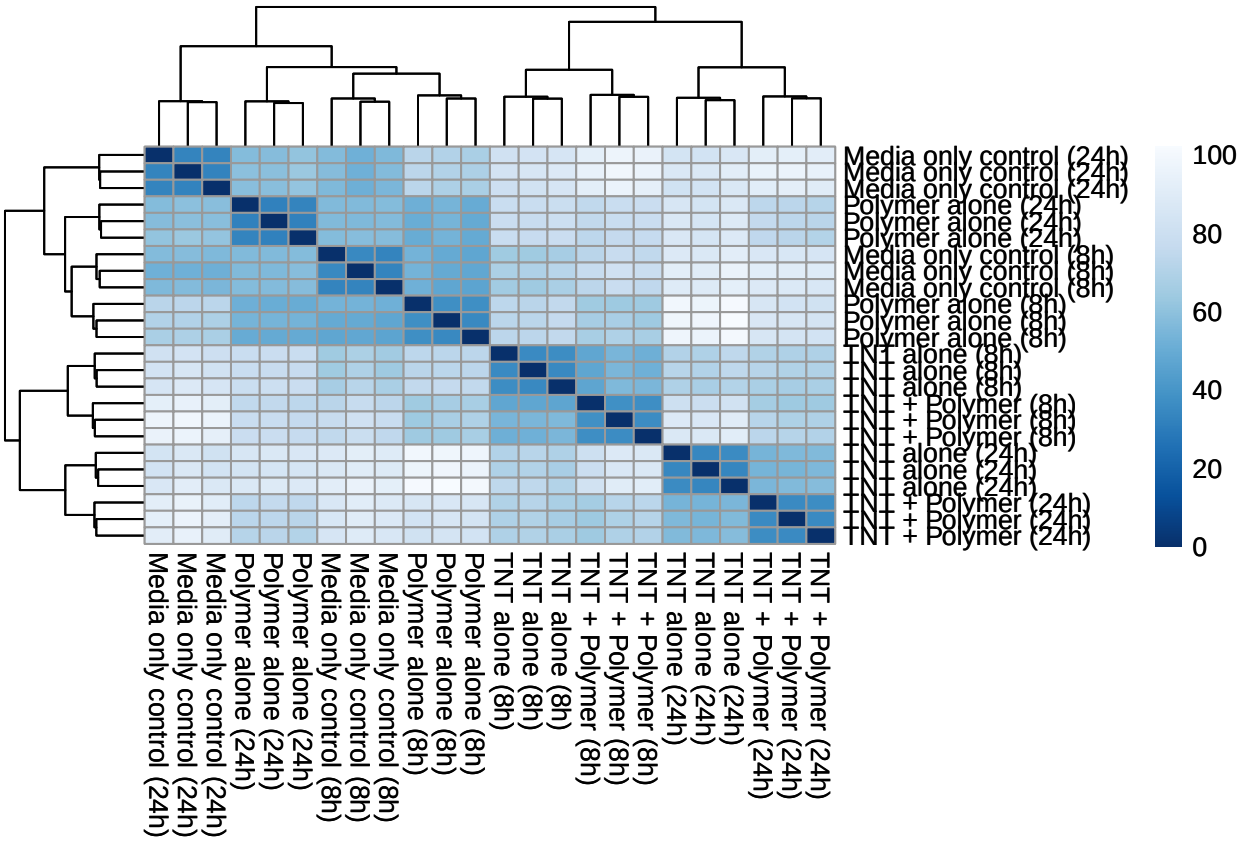
```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

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

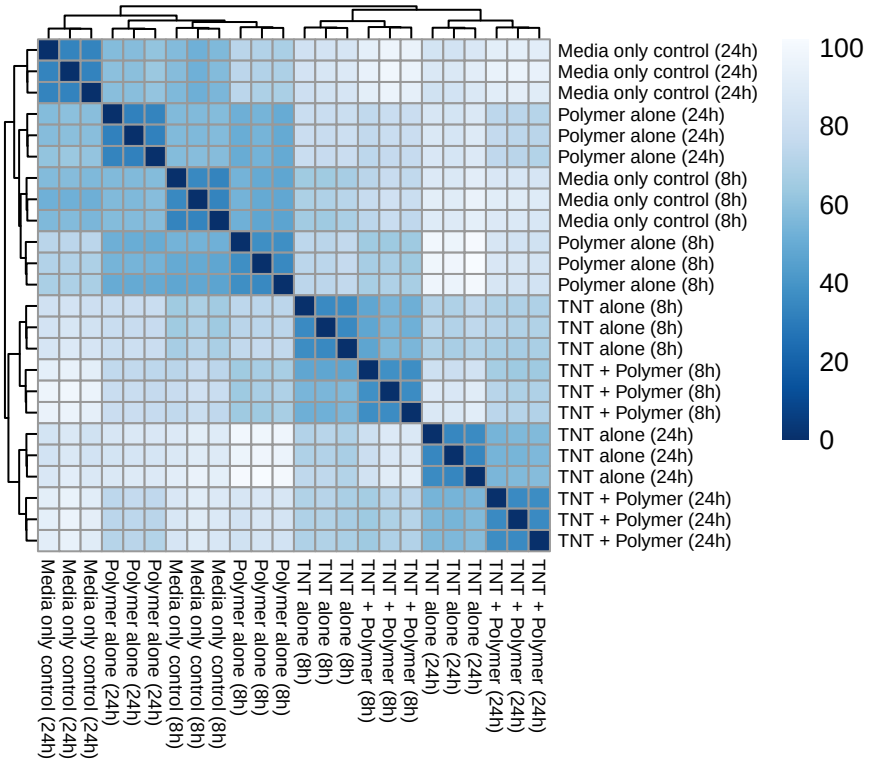
```
vsd <- vst(all_dds, blind = FALSE)
sampleDists_vsd <- dist(t(assay(vsd)))
sampleDistMatrix_vsd <- as.matrix( sampleDists_vsd )
rownames(sampleDistMatrix_vsd) <- paste(vsd$Condition_name)
colnames(sampleDistMatrix_vsd) <- paste(vsd$Condition_name)
```

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.

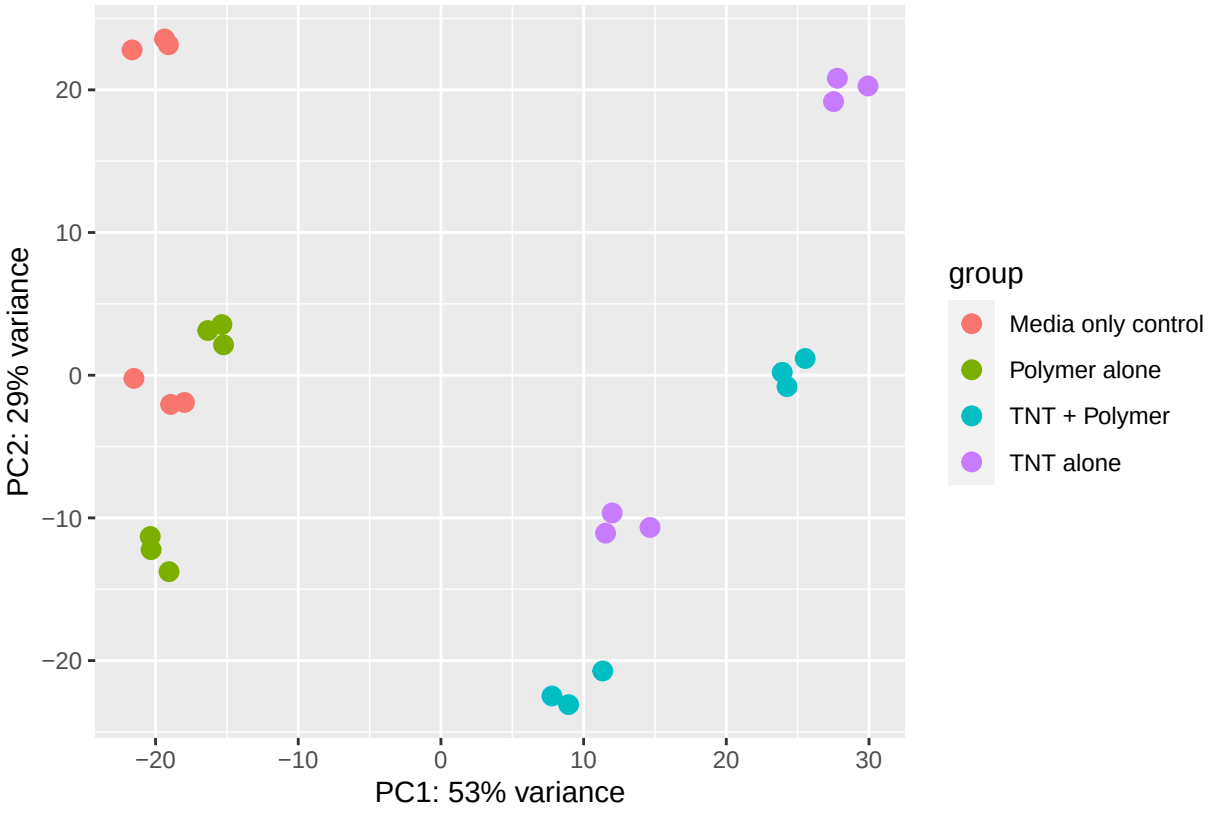
```
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
heatmap <- pheatmap(sampleDistMatrix_vsd,
                    clustering_distance_rows = sampleDists_vsd,
                    clustering_distance_cols = sampleDists_vsd,
                    col = colors)
heatmap
```



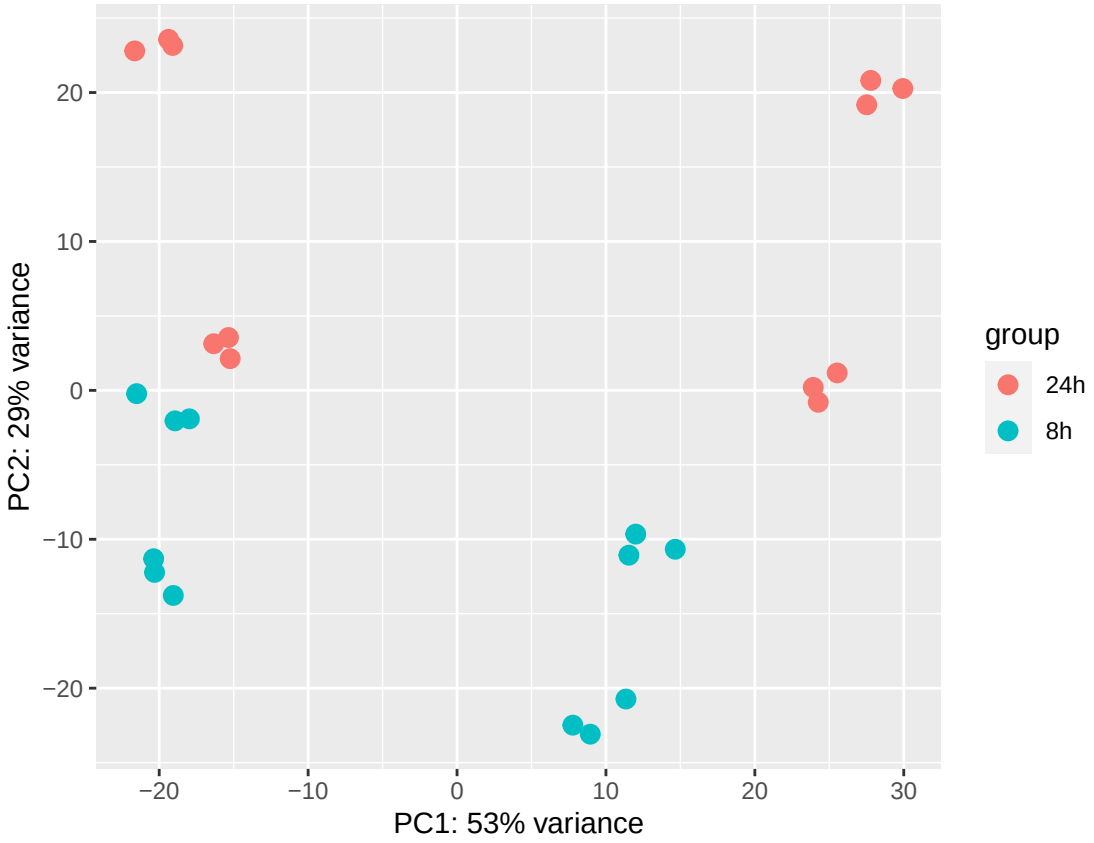

```
heatmap <- pheatmap(sampleDistMatrix_vsd,
                    clustering_distance_rows = sampleDists_vsd,
                    clustering_distance_cols = sampleDists_vsd,
                    col = colors,cellwidth=8,
                    cellheight=8,treeheight_row = 8, treeheight_col = 8, fontsize_row = 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
```

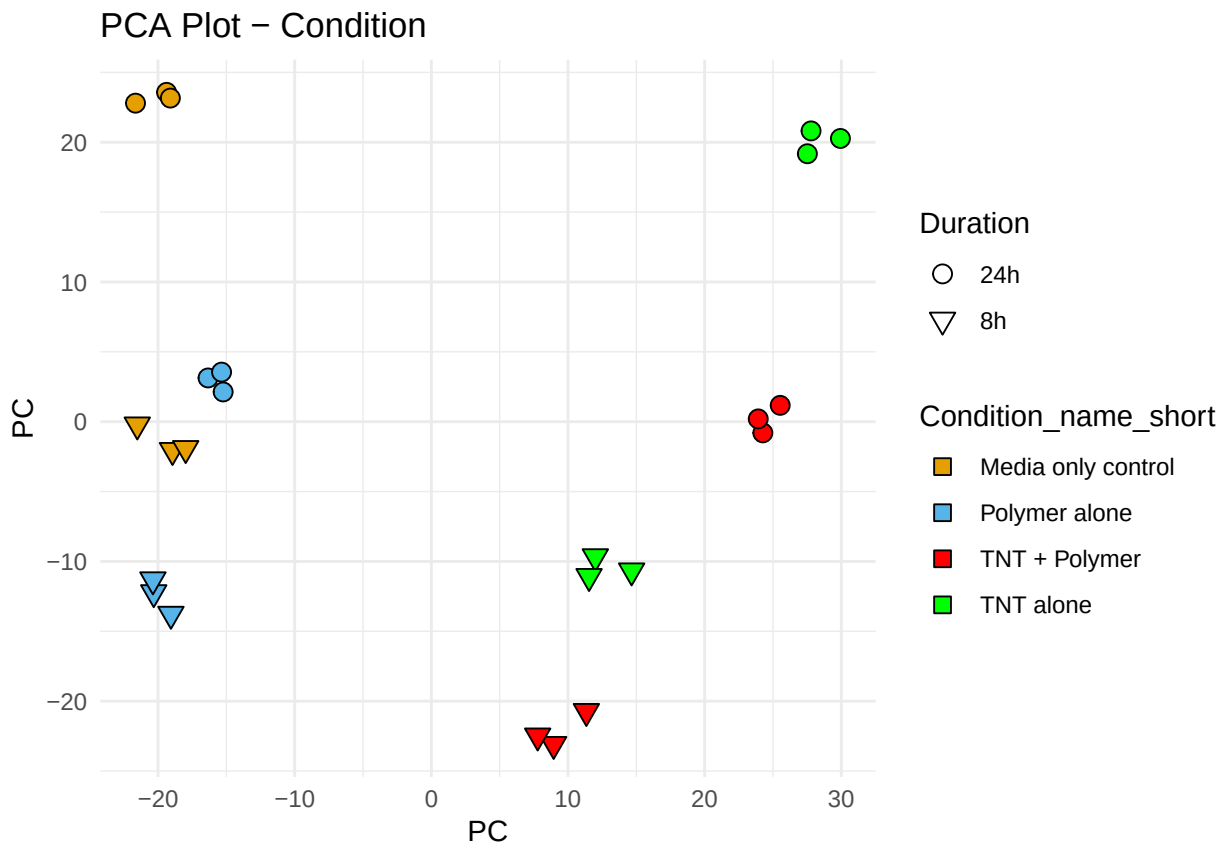



```
library(ggfortify)
library(patchwork)

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)
```

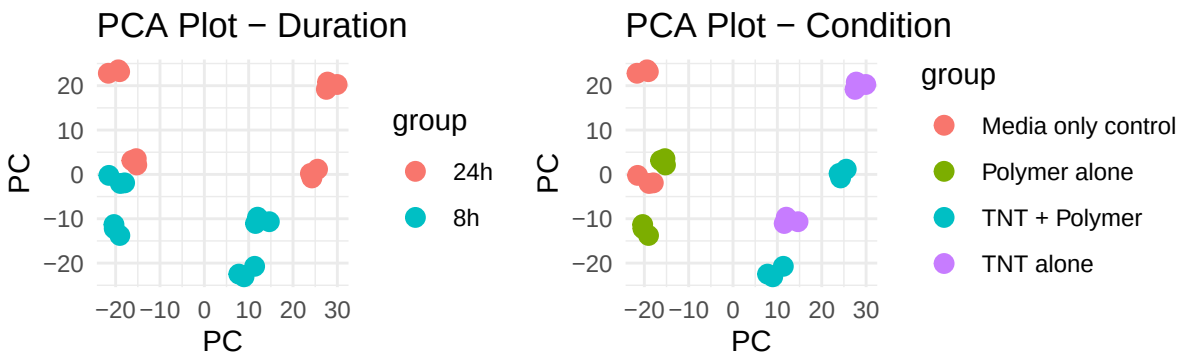



```
plot_duration <- autoplot(pca_data_duration, data = vsd$Duration, shape  = "Duration") +
  labs(x = paste0("PC", pca_data_duration$x[, 1]), y = paste0("PC", pca_data_duration$x[, 2])) +
  ggtitle("PCA Plot - Duration") +
  theme_minimal()

# 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
```