```
############################################################
## Build a gene x sample count matrix from your 18 files
## /opt/app-root/src/home/hannah
############################################################

# 1. Load packages
library(readr)
library(dplyr)
```

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

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)

# 2. Directory where your files live
data_dir <- "/opt/app-root/src/home/hannah"   # change if needed

# 3. List all .txt.gz files in that directory
files <- list.files(
  path   = data_dir,
  pattern = "\\.txt\\.gz$",
  full.names = TRUE
)

cat("Number of .txt.gz files found:", length(files), "\n")
```

```
## Number of .txt.gz files found: 18
```

```
cat("Example file names:\n")
```

```
## Example file names:
```

```
print(basename(files))
```

```
##  [1] "GSM5255692_R1_H1975_ONT.txt.gz"
##  [2] "GSM5255693_R2_H1975_ONT.txt.gz"
##  [3] "GSM5255694_R3_H1975_ONT.txt.gz"
##  [4] "GSM5255695_R1_H1975_Illumina.txt.gz"
##  [5] "GSM5255696_R2_H1975_Illumina.txt.gz"
##  [6] "GSM5255697_R3_H1975_Illumina.txt.gz"
##  [7] "GSM5255698_R1_H1975_PacBio.txt.gz"
##  [8] "GSM5255699_R2_H1975_PacBio.txt.gz"
##  [9] "GSM5255700_R3_H1975_PacBio.txt.gz"
## [10] "GSM5255701_R1_HCC827_ONT.txt.gz"
## [11] "GSM5255702_R2_HCC827_ONT.txt.gz"
## [12] "GSM5255703_R3_HCC827_ONT.txt.gz"
## [13] "GSM5255704_R1_HCC827_Illumina.txt.gz"
## [14] "GSM5255705_R2_HCC827_Illumina.txt.gz"
## [15] "GSM5255706_R3_HCC827_Illumina.txt.gz"
## [16] "GSM5255707_R1_HCC827_PacBio.txt.gz"
## [17] "GSM5255708_R2_HCC827_PacBio.txt.gz"
## [18] "GSM5255709_R3_HCC827_PacBio.txt.gz"
```

```
# 4. Function to read ONE file and split "ENSG... count"
parse_counts_file <- function(f) {
  # read as one-column table (character)
  tbl <- read_tsv(f, col_names = FALSE, show_col_types = FALSE)
  x <- tbl[[1]]  # the only column

  # split into gene_id (before first space) and count (after last space)
  gene_id <- sub(" .*", "", x)
  count   <- as.numeric(sub(".* ", "", x))

  tibble(gene_id = gene_id, count = count)
}

# 5. Read and parse ALL files
parsed_list <- lapply(files, parse_counts_file)
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```
names(parsed_list) <- sub("\\.txt\\.gz$", "", basename(files))

# Quick check of one example
cat("\nExample parsed table:\n")
```

```
##
## Example parsed table:
```

```
print(head(parsed_list[[1]], 10))
```

```
## # A tibble: 10 × 2
##    gene_id            count
##    <chr>              <dbl>
##  1 barcode01.bam         NA
##  2 ENSG00000223972.5   1234
##  3 ENSG00000227232.5   1886
##  4 ENSG00000278267.1      4
##  5 ENSG00000243485.5     28
##  6 ENSG00000284332.1      4
##  7 ENSG00000237613.2      2
##  8 ENSG00000268020.3      0
##  9 ENSG00000240361.2      0
## 10 ENSG00000186092.6      0
```

```
# 6. Create a vector of ALL unique gene IDs across all files
all_genes <- sort(unique(unlist(lapply(parsed_list, function(df) df$gene_id))))
cat("\nTotal unique genes:", length(all_genes), "\n")
```

```
##
## Total unique genes: 78206
```

```
# 7. Build an empty matrix: rows = genes, cols = samples
count_mat <- matrix(
  0,
  nrow = length(all_genes),
  ncol = length(parsed_list),
  dimnames = list(all_genes, names(parsed_list))
)

# 8. Fill the matrix with counts for each sample
for (i in seq_along(parsed_list)) {
  df  <- parsed_list[[i]]
  idx <- match(df$gene_id, all_genes)
  count_mat[idx, i] <- df$count
}

# 9. Convert to data.frame for easier handling
count_df <- as.data.frame(count_mat)

############################################################
## From count_df -> DESeq2 (Illumina, H1975 vs HCC827)
############################################################

# Quick check:
dim(count_df)    # should be ~78206 x 18
```

```
## [1] 78206    18
```

```
## 2. Keep only real genes: rows whose names start with "ENSG"
gene_ids <- rownames(count_df)
is_gene  <- grepl("^ENSG", gene_ids)

count_genes <- count_df[is_gene, ]
dim(count_genes)   # fewer rows now, only ENSG* genes
```

```
## [1] 78112    18
## 3. Build sample metadata (colData) from column names
sample_ids <- colnames(count_genes)

cell_line <- ifelse(grepl("H1975",  sample_ids), "H1975",
                    ifelse(grepl("HCC827", sample_ids), "HCC827", NA))

platform  <- ifelse(grepl("Illumina", sample_ids), "Illumina",
              ifelse(grepl("ONT",      sample_ids), "ONT",
              ifelse(grepl("PacBio",   sample_ids), "PacBio", NA)))

coldata <- data.frame(
  sample_id = sample_ids,
  cell_line = factor(cell_line),
  platform  = factor(platform),
  row.names = sample_ids
)

coldata
```

```
##                                                   sample_id cell_line platform
## GSM5255692_R1_H1975_ONT             GSM5255692_R1_H1975_ONT     H1975      ONT
## GSM5255693_R2_H1975_ONT             GSM5255693_R2_H1975_ONT     H1975      ONT
## GSM5255694_R3_H1975_ONT             GSM5255694_R3_H1975_ONT     H1975      ONT
## GSM5255695_R1_H1975_Illumina   GSM5255695_R1_H1975_Illumina     H1975 Illumina
## GSM5255696_R2_H1975_Illumina   GSM5255696_R2_H1975_Illumina     H1975 Illumina
## GSM5255697_R3_H1975_Illumina   GSM5255697_R3_H1975_Illumina     H1975 Illumina
## GSM5255698_R1_H1975_PacBio       GSM5255698_R1_H1975_PacBio     H1975   PacBio
## GSM5255699_R2_H1975_PacBio       GSM5255699_R2_H1975_PacBio     H1975   PacBio
## GSM5255700_R3_H1975_PacBio       GSM5255700_R3_H1975_PacBio     H1975   PacBio
## GSM5255701_R1_HCC827_ONT           GSM5255701_R1_HCC827_ONT    HCC827      ONT
## GSM5255702_R2_HCC827_ONT           GSM5255702_R2_HCC827_ONT    HCC827      ONT
## GSM5255703_R3_HCC827_ONT           GSM5255703_R3_HCC827_ONT    HCC827      ONT
## GSM5255704_R1_HCC827_Illumina GSM5255704_R1_HCC827_Illumina    HCC827 Illumina
## GSM5255705_R2_HCC827_Illumina GSM5255705_R2_HCC827_Illumina    HCC827 Illumina
## GSM5255706_R3_HCC827_Illumina GSM5255706_R3_HCC827_Illumina    HCC827 Illumina
## GSM5255707_R1_HCC827_PacBio     GSM5255707_R1_HCC827_PacBio    HCC827   PacBio
## GSM5255708_R2_HCC827_PacBio     GSM5255708_R2_HCC827_PacBio    HCC827   PacBio
## GSM5255709_R3_HCC827_PacBio     GSM5255709_R3_HCC827_PacBio    HCC827   PacBio
```

```
## 4. Subset to Illumina samples only
keep_illumina <- coldata$platform == "Illumina"

counts_illumina   <- as.matrix(count_genes[, keep_illumina])
coldata_illumina  <- coldata[keep_illumina, ]

dim(counts_illumina)       # genes x Illumina samples (should be genes x 6)
```

```
## [1] 78112     6
```

```
coldata_illumina
```

```
##                                                    sample_id cell_line platform
## GSM5255695_R1_H1975_Illumina   GSM5255695_R1_H1975_Illumina     H1975 Illumina
## GSM5255696_R2_H1975_Illumina   GSM5255696_R2_H1975_Illumina     H1975 Illumina
## GSM5255697_R3_H1975_Illumina   GSM5255697_R3_H1975_Illumina     H1975 Illumina
## GSM5255704_R1_HCC827_Illumina GSM5255704_R1_HCC827_Illumina    HCC827 Illumina
## GSM5255705_R2_HCC827_Illumina GSM5255705_R2_HCC827_Illumina    HCC827 Illumina
## GSM5255706_R3_HCC827_Illumina GSM5255706_R3_HCC827_Illumina    HCC827 Illumina
```

```
## 5. Load DESeq2 (install if needed)
if (!requireNamespace("DESeq2", quietly = TRUE)) {
  if (!requireNamespace("BiocManager", quietly = TRUE)) {
    install.packages("BiocManager")
  }
  BiocManager::install("DESeq2")
}

library(DESeq2)
```

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

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

```
## Loading required package: BiocGenerics
```

```
## 
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:dplyr':
## 
##     combine, intersect, setdiff, union
```

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

```
## 
## Attaching package: 'S4Vectors'
## The following objects are masked from 'package:dplyr':
##
##     first, rename
```

```
## 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
```

```
##
## Attaching package: 'IRanges'
```

```
## The following object is masked from 'package:purrr':
##
##     reduce
```

```
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: GenomeInfoDb
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: MatrixGenerics
```

```
## Loading required package: matrixStats
```

```
##
## Attaching package: 'matrixStats'
```

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

```
##
## 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
```

```
## 6. Create DESeq2 dataset (design = cell_line)
dds <- DESeqDataSetFromMatrix(
  countData = counts_illumina,
  colData   = coldata_illumina,
  design    = ~ cell_line
)
```

```
## converting counts to integer mode
```

```
## 7. Run DESeq2
dds <- DESeq(dds)
```

```
## estimating size factors
```

```
## estimating dispersions
```

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

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

```
## final dispersion estimates
```

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

```
## 8. Get results: H1975 vs HCC827
res <- results(dds, contrast = c("cell_line", "H1975", "HCC827"))

## 9. Order by adjusted p-value and show top genes
res_ordered <- res[order(res$padj), ]

head(res_ordered)
```

```
## log2 fold change (MLE): cell_line H1975 vs HCC827
## Wald test p-value: cell_line H1975 vs HCC827
## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat    pvalue
##                    <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000100033.16  14591.39       -9.74007 0.1969468  -49.4553         0
## ENSG00000100867.14  10600.17        6.49423 0.1214626   53.4670         0
## ENSG00000102572.14   2099.55       -6.10547 0.1476937  -41.3387         0
## ENSG00000106366.9    4338.76        8.50944 0.2097009   40.5789         0
## ENSG00000111490.14   2205.08       -3.72771 0.0977294  -38.1432         0
## ENSG00000117983.17  13882.70       -9.51244 0.1830708  -51.9605         0
##                         padj
##                    <numeric>
## ENSG00000100033.16         0
## ENSG00000100867.14         0
## ENSG00000102572.14         0
## ENSG00000106366.9          0
## ENSG00000111490.14         0
## ENSG00000117983.17         0
```

```
## 10. Optionally, write results to file
write.csv(
  as.data.frame(res_ordered),
  file = "DESeq2_Illumina_H1975_vs_HCC827.csv"
)

cat("\nDESeq2 analysis finished. Results saved to:\n")
```

```
##
## DESeq2 analysis finished. Results saved to:
```

```
cat("DESeq2_Illumina_H1975_vs_HCC827.csv\n")
```

```
## DESeq2_Illumina_H1975_vs_HCC827.csv
```

```
############################################################
```

```
############################################################
## DESeq2 for ONT samples (H1975 vs HCC827)
############################################################

# 1. We assume you already have:
#    - count_genes   (genes x samples, only ENSG rows)
#    - coldata       (sample metadata: cell_line + platform)

# Quick check: uncomment if needed
# dim(count_genes)
# head(coldata)

# 2. Subset to ONT samples only
keep_ont <- coldata$platform == "ONT"

counts_ont <- as.matrix(count_genes[, keep_ont])
coldata_ont <- coldata[keep_ont, ]

cat("ONT samples found:\n")
```

```
## ONT samples found:
```

```
print(coldata_ont)
```

```
##                                         sample_id cell_line platform
## GSM5255692_R1_H1975_ONT   GSM5255692_R1_H1975_ONT     H1975      ONT
## GSM5255693_R2_H1975_ONT   GSM5255693_R2_H1975_ONT     H1975      ONT
## GSM5255694_R3_H1975_ONT   GSM5255694_R3_H1975_ONT     H1975      ONT
## GSM5255701_R1_HCC827_ONT GSM5255701_R1_HCC827_ONT    HCC827      ONT
## GSM5255702_R2_HCC827_ONT GSM5255702_R2_HCC827_ONT    HCC827      ONT
## GSM5255703_R3_HCC827_ONT GSM5255703_R3_HCC827_ONT    HCC827      ONT
```

```
cat("\nDimensions of ONT count matrix:\n")
```

```
##
## Dimensions of ONT count matrix:
```

```
print(dim(counts_ont))     # should be genes x 6 ONT samples
```

```
## [1] 78112     6
```

```
# 3. Load DESeq2 (install if needed)
if (!requireNamespace("DESeq2", quietly = TRUE)) {
    if (!requireNamespace("BiocManager", quietly = TRUE)) {
        install.packages("BiocManager")
    }
    BiocManager::install("DESeq2")
}

library(DESeq2)

# 4. Create DESeq2 dataset
dds_ont <- DESeqDataSetFromMatrix(
    countData = counts_ont,
    colData   = coldata_ont,
    design    = ~ cell_line
)
```

```
## converting counts to integer mode
```

```
# 5. Run DESeq2
dds_ont <- DESeq(dds_ont)
```

```
## estimating size factors
```

```
## estimating dispersions
```

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

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

```
## final dispersion estimates
```

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

```
# 6. Get results (H1975 vs HCC827)
res_ont <- results(dds_ont, contrast = c("cell_line", "H1975", "HCC827"))

# 7. Order by padj
res_ont_ordered <- res_ont[order(res_ont$padj), ]

# Show top genes
cat("\nTop ONT DE genes:\n")
```

```
##
## Top ONT DE genes:
```

```
print(head(res_ont_ordered))
```

```
## log2 fold change (MLE): cell_line H1975 vs HCC827
## Wald test p-value: cell_line H1975 vs HCC827
## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat       pvalue
##                    <numeric>      <numeric> <numeric> <numeric>    <numeric>
## ENSG00000148346.11  46531.09       -8.76183  0.228848  -38.2867  0.00000e+00
## ENSG00000169715.14  36652.24       12.50857  0.294092   42.5329  0.00000e+00
## ENSG00000162366.7   10847.53       -7.90284  0.214195  -36.8956 5.44543e-298
## ENSG00000100867.14   9250.17        6.46922  0.180656   35.8096 7.82483e-281
## ENSG00000100033.16  12156.14       -9.33039  0.266070  -35.0675 2.11367e-269
## ENSG00000133321.10  26555.07       -5.26807  0.157116  -33.5299 1.76645e-246
##                            padj
##                       <numeric>
## ENSG00000148346.11  0.00000e+00
## ENSG00000169715.14  0.00000e+00
## ENSG00000162366.7  6.60821e-294
## ENSG00000100867.14 7.12177e-277
## ENSG00000100033.16 1.53900e-265
## ENSG00000133321.10 1.07182e-242
```

```
# 8. Save results
write.csv(
    as.data.frame(res_ont_ordered),
    file = "DESeq2_ONT_H1975_vs_HCC827.csv"
)

cat("\nDESeq2 ONT analysis finished. Results saved to:\n")
```

```
##
## DESeq2 ONT analysis finished. Results saved to:
```

```
cat("DESeq2_ONT_H1975_vs_HCC827.csv\n")
```

```
## DESeq2_ONT_H1975_vs_HCC827.csv
```

```
############################################################
```

```
############################################################
## DESeq2 for PacBio samples (H1975 vs HCC827)
############################################################

# 1. We assume you already have:
#    - count_genes   (genes x samples, only ENSG rows)
#    - coldata       (sample metadata: cell_line + platform)

# Quick check (optional)
# dim(count_genes)
# head(coldata)

# 2. Subset to PacBio samples only
keep_pacbio <- coldata$platform == "PacBio"

counts_pacbio  <- as.matrix(count_genes[, keep_pacbio])
coldata_pacbio <- coldata[keep_pacbio, ]

cat("PacBio samples found:\n")
```

```
## PacBio samples found:
```

```
print(coldata_pacbio)
```

```
##                                               sample_id cell_line platform
## GSM5255698_R1_H1975_PacBio   GSM5255698_R1_H1975_PacBio     H1975   PacBio
## GSM5255699_R2_H1975_PacBio   GSM5255699_R2_H1975_PacBio     H1975   PacBio
## GSM5255700_R3_H1975_PacBio   GSM5255700_R3_H1975_PacBio     H1975   PacBio
## GSM5255707_R1_HCC827_PacBio GSM5255707_R1_HCC827_PacBio    HCC827   PacBio
## GSM5255708_R2_HCC827_PacBio GSM5255708_R2_HCC827_PacBio    HCC827   PacBio
## GSM5255709_R3_HCC827_PacBio GSM5255709_R3_HCC827_PacBio    HCC827   PacBio
```

```
cat("\nDimensions of PacBio count matrix (genes x samples):\n")
```

```
##
## Dimensions of PacBio count matrix (genes x samples):
```

```
print(dim(counts_pacbio))     # should be genes x 6 PacBio samples
```

```
## [1] 78112     6
```

```
# 3. Load DESeq2 (install if needed)
if (!requireNamespace("DESeq2", quietly = TRUE)) {
    if (!requireNamespace("BiocManager", quietly = TRUE)) {
        install.packages("BiocManager")
    }
    BiocManager::install("DESeq2")
}
library(DESeq2)

# 4. Create DESeq2 dataset
dds_pacbio <- DESeqDataSetFromMatrix(
    countData = counts_pacbio,
    colData   = coldata_pacbio,
    design    = ~ cell_line
)
```

```
## converting counts to integer mode
```

```
# 5. Run DESeq2
dds_pacbio <- DESeq(dds_pacbio)
```

```
## estimating size factors
```

```
## estimating dispersions
```

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

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

```
## -- note: fitType='parametric', but the dispersion trend was not well captured by t
he
##    function: y = a/x + b, and a local regression fit was automatically substitute
d.
##    specify fitType='local' or 'mean' to avoid this message next time.
```

```
## final dispersion estimates
```

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

```
# 6. Get results (H1975 vs HCC827)
res_pacbio <- results(dds_pacbio, contrast = c("cell_line", "H1975", "HCC827"))

# 7. Order by adjusted p-value
res_pacbio_ordered <- res_pacbio[order(res_pacbio$padj), ]

# Show top genes in console
cat("\nTop PacBio DE genes:\n")
```

```
##
## Top PacBio DE genes:
```

```
print(head(res_pacbio_ordered))
```

```
## log2 fold change (MLE): cell_line H1975 vs HCC827
## Wald test p-value: cell_line H1975 vs HCC827
## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat      pvalue
##                    <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000146648.17   43.5443       -3.13315  0.453849  -6.90350 5.07359e-12
## ENSG00000117983.17   14.2218       -7.33002  1.247280  -5.87680 4.18265e-09
## ENSG00000135506.15  105.7865       -3.78566  0.656398  -5.76733 8.05384e-09
## ENSG00000257342.1   105.7865       -3.78566  0.656398  -5.76733 8.05384e-09
## ENSG00000106366.8    11.8771        7.01151  1.245280   5.63047 1.79716e-08
## ENSG00000115414.18   77.1806        2.55017  0.461127   5.53028 3.19711e-08
##                           padj
##                      <numeric>
## ENSG00000146648.17 3.56318e-08
## ENSG00000117983.17 1.41405e-05
## ENSG00000135506.15 1.41405e-05
## ENSG00000257342.1  1.41405e-05
## ENSG00000106366.8  2.52429e-05
## ENSG00000115414.18 3.74222e-05
```

```
# 8. Save results to CSV
write.csv(
    as.data.frame(res_pacbio_ordered),
    file = "DESeq2_PacBio_H1975_vs_HCC827.csv"
)

cat("\nDESeq2 PacBio analysis finished. Results saved to:\n")
```

```
##
## DESeq2 PacBio analysis finished. Results saved to:
```

```
cat("DESeq2_PacBio_H1975_vs_HCC827.csv\n")
```

```
## DESeq2_PacBio_H1975_vs_HCC827.csv
```

```
############################################################
```

```
############################################################
## STEP 1: Compare Illumina, ONT, PacBio DESeq2 results
############################################################

# 1. Load the DESeq2 results for each platform
ill <- read.csv("DESeq2_Illumina_H1975_vs_HCC827.csv", row.names = 1)
ont <- read.csv("DESeq2_ONT_H1975_vs_HCC827.csv", row.names = 1)
pb  <- read.csv("DESeq2_PacBio_H1975_vs_HCC827.csv", row.names = 1)

# Quick check
dim(ill); dim(ont); dim(pb)
```

```
## [1] 78112     6
```

```
## [1] 78112     6
```

```
## [1] 78112     6
```

```
head(ill[, c("log2FoldChange", "padj")])
```

```
##                    log2FoldChange padj
## ENSG00000100033.16      -9.740068    0
## ENSG00000100867.14       6.494235    0
## ENSG00000102572.14      -6.105466    0
## ENSG00000106366.9        8.509436    0
## ENSG00000111490.14      -3.727710    0
## ENSG00000117983.17      -9.512444    0
```

```
# 2. Keep just log2FC and padj for each
ill_sub <- ill[, c("log2FoldChange", "padj")]
ont_sub <- ont[, c("log2FoldChange", "padj")]
pb_sub  <- pb[,  c("log2FoldChange", "padj")]

colnames(ill_sub) <- c("log2FC_Illumina", "padj_Illumina")
colnames(ont_sub) <- c("log2FC_ONT",      "padj_ONT")
colnames(pb_sub)  <- c("log2FC_PacBio",   "padj_PacBio")

# 3. Merge by gene (rownames = gene IDs)
# Use merge(..., all = TRUE) to keep all genes
merged12 <- merge(ill_sub, ont_sub, by = "row.names", all = TRUE)
rownames(merged12) <- merged12$Row.names
merged12$Row.names <- NULL

merged_all <- merge(merged12, pb_sub, by = "row.names", all = TRUE)
rownames(merged_all) <- merged_all$Row.names
merged_all$Row.names <- NULL

dim(merged_all)
```

```
## [1] 78112     6
```

```
head(merged_all)
```

```
##                    log2FC_Illumina padj_Illumina log2FC_ONT     padj_ONT
## ENSG00000000003.14               NA            NA -1.1619225 1.400388e-07
## ENSG00000000003.15      -1.79986914  8.088531e-56         NA           NA
## ENSG00000000005.5                NA            NA -2.7534602           NA
## ENSG00000000005.6       -0.02116534            NA         NA           NA
## ENSG00000000419.12       0.66340102  6.804318e-06  0.8675107 2.418047e-04
## ENSG00000000457.13               NA            NA -0.1006004 7.013186e-01
##                    log2FC_PacBio padj_PacBio
## ENSG00000000003.14    -1.9611783   0.2492397
## ENSG00000000003.15            NA          NA
## ENSG00000000005.5             NA          NA
## ENSG00000000005.6             NA          NA
## ENSG00000000419.12    -1.0086883          NA
## ENSG00000000457.13    -0.9985465          NA
```

```
# 4. Correlation of log2 fold changes (only where both are not NA)

# Illumina vs ONT
idx_io <- complete.cases(merged_all$log2FC_Illumina, merged_all$log2FC_ONT)
cor_IO <- cor(
  merged_all$log2FC_Illumina[idx_io],
  merged_all$log2FC_ONT[idx_io]
)
cor_IO
```

```
## [1] 0.6874974
```

```
# Illumina vs PacBio
idx_ip <- complete.cases(merged_all$log2FC_Illumina, merged_all$log2FC_PacBio)
cor_IP <- cor(
  merged_all$log2FC_Illumina[idx_ip],
  merged_all$log2FC_PacBio[idx_ip]
)
cor_IP
```

```
## [1] 0.6446656
```

```
# ONT vs PacBio
idx_op <- complete.cases(merged_all$log2FC_ONT, merged_all$log2FC_PacBio)
cor_OP <- cor(
  merged_all$log2FC_ONT[idx_op],
  merged_all$log2FC_PacBio[idx_op]
)
cor_OP
```

```
## [1] 0.6033697
```

```
cat("\nCorrelation of log2FC:\n")
```

```
##
## Correlation of log2FC:
```

```
cat("Illumina vs ONT:    ", cor_IO, "\n")
```

```
## Illumina vs ONT:     0.6874974
```

```
cat("Illumina vs PacBio: ", cor_IP, "\n")
```

```
## Illumina vs PacBio:  0.6446656
```

```
cat("ONT vs PacBio:      ", cor_OP, "\n")
```

```
## ONT vs PacBio:       0.6033697
```

```
# 5. Overlap of significantly DE genes (padj < 0.05 and |log2FC| >= 1)

sig_ill <- rownames(ill_sub)[which(ill_sub$padj_Illumina < 0.05 &
                                   abs(ill_sub$log2FC_Illumina) >= 1)]
sig_ont <- rownames(ont_sub)[which(ont_sub$padj_ONT < 0.05 &
                                   abs(ont_sub$log2FC_ONT) >= 1)]
sig_pb  <- rownames(pb_sub)[which(pb_sub$padj_PacBio < 0.05 &
                                  abs(pb_sub$log2FC_PacBio) >= 1)]

length(sig_ill); length(sig_ont); length(sig_pb)
```

```
## [1] 10263
```

```
## [1] 10971
```

```
## [1] 147
```

```
# pairwise overlaps
overlap_ill_ont <- intersect(sig_ill, sig_ont)
overlap_ill_pb  <- intersect(sig_ill, sig_pb)
overlap_ont_pb  <- intersect(sig_ont, sig_pb)

# triple overlap
overlap_all3 <- Reduce(intersect, list(sig_ill, sig_ont, sig_pb))

cat("\nSignificant DE genes (padj < 0.05 & |log2FC| >= 1):\n")
```

```
##
## Significant DE genes (padj < 0.05 & |log2FC| >= 1):
```

```
cat("Illumina: ", length(sig_ill), "\n")
```

```
## Illumina:  10263
```

```
cat("ONT:      ", length(sig_ont), "\n")
```

```
## ONT:       10971
```

```
cat("PacBio:   ", length(sig_pb),  "\n\n")
```

```
## PacBio:    147
```

```
cat("Overlap Illumina ∩ ONT:    ", length(overlap_ill_ont), "\n")
```

```
## Overlap Illumina ∩ ONT:     3527
```

```
cat("Overlap Illumina ∩ PacBio: ", length(overlap_ill_pb),  "\n")
```

```
## Overlap Illumina ∩ PacBio:  53
```

```
cat("Overlap ONT ∩ PacBio:      ", length(overlap_ont_pb),  "\n")
```

```
## Overlap ONT ∩ PacBio:       127
```

```
cat("Overlap all three:         ", length(overlap_all3),    "\n")
```

```
## Overlap all three:          47
```

```
dim(count_df)
```

```
## [1] 78206    18
```

```
############################################################
## STEP 2: Combined DESeq2 + PCA + Sample Clustering
############################################################

# 0. If needed:
# library(DESeq2)
# install.packages("pheatmap")   # if not installed

library(DESeq2)
library(pheatmap)

# 1. Filter to real genes (ENSG rows) if not already done
gene_ids <- rownames(count_df)
is_gene  <- grepl("^ENSG", gene_ids)

count_genes <- count_df[is_gene, ]
dim(count_genes)   # genes x 18 samples
```

```
## [1] 78112    18
```

```
# 2. Build sample metadata (colData) from column names
sample_ids <- colnames(count_genes)

cell_line <- ifelse(grepl("H1975",  sample_ids), "H1975",
             ifelse(grepl("HCC827", sample_ids), "HCC827", NA))

platform  <- ifelse(grepl("Illumina", sample_ids), "Illumina",
             ifelse(grepl("ONT",      sample_ids), "ONT",
             ifelse(grepl("PacBio",   sample_ids), "PacBio", NA)))

coldata_all <- data.frame(
  sample_id = sample_ids,
  cell_line = factor(cell_line),
  platform  = factor(platform),
  row.names = sample_ids
)

coldata_all
```

```
##                                                   sample_id cell_line platform
## GSM5255692_R1_H1975_ONT             GSM5255692_R1_H1975_ONT     H1975      ONT
## GSM5255693_R2_H1975_ONT             GSM5255693_R2_H1975_ONT     H1975      ONT
## GSM5255694_R3_H1975_ONT             GSM5255694_R3_H1975_ONT     H1975      ONT
## GSM5255695_R1_H1975_Illumina   GSM5255695_R1_H1975_Illumina     H1975 Illumina
## GSM5255696_R2_H1975_Illumina   GSM5255696_R2_H1975_Illumina     H1975 Illumina
## GSM5255697_R3_H1975_Illumina   GSM5255697_R3_H1975_Illumina     H1975 Illumina
## GSM5255698_R1_H1975_PacBio       GSM5255698_R1_H1975_PacBio     H1975   PacBio
## GSM5255699_R2_H1975_PacBio       GSM5255699_R2_H1975_PacBio     H1975   PacBio
## GSM5255700_R3_H1975_PacBio       GSM5255700_R3_H1975_PacBio     H1975   PacBio
## GSM5255701_R1_HCC827_ONT           GSM5255701_R1_HCC827_ONT    HCC827      ONT
## GSM5255702_R2_HCC827_ONT           GSM5255702_R2_HCC827_ONT    HCC827      ONT
## GSM5255703_R3_HCC827_ONT           GSM5255703_R3_HCC827_ONT    HCC827      ONT
## GSM5255704_R1_HCC827_Illumina GSM5255704_R1_HCC827_Illumina    HCC827 Illumina
## GSM5255705_R2_HCC827_Illumina GSM5255705_R2_HCC827_Illumina    HCC827 Illumina
## GSM5255706_R3_HCC827_Illumina GSM5255706_R3_HCC827_Illumina    HCC827 Illumina
## GSM5255707_R1_HCC827_PacBio     GSM5255707_R1_HCC827_PacBio    HCC827   PacBio
## GSM5255708_R2_HCC827_PacBio     GSM5255708_R2_HCC827_PacBio    HCC827   PacBio
## GSM5255709_R3_HCC827_PacBio     GSM5255709_R3_HCC827_PacBio    HCC827   PacBio
```

```
# Quick sanity check
table(coldata_all$cell_line, coldata_all$platform)
```

```
##
##           Illumina ONT PacBio
##   H1975          3   3      3
##   HCC827         3   3      3
```

```
# 3. Create a DESeq2 object for ALL platforms together
dds_all <- DESeqDataSetFromMatrix(
  countData = as.matrix(count_genes),
  colData   = coldata_all,
  design    = ~ platform + cell_line
)
```

```
## converting counts to integer mode
```

```
# Optional: filter out very lowly expressed genes
keep <- rowSums(counts(dds_all)) >= 10
dds_all <- dds_all[keep, ]
dds_all
```

```
## class: DESeqDataSet
## dim: 52725 18
## metadata(1): version
## assays(1): counts
## rownames(52725): ENSG00000000003.14 ENSG00000000003.15 ...
##   ENSG00000288586.1 ENSG00000288587.1
## rowData names(0):
## colnames(18): GSM5255692_R1_H1975_ONT GSM5255693_R2_H1975_ONT ...
##   GSM5255708_R2_HCC827_PacBio GSM5255709_R3_HCC827_PacBio
## colData names(3): sample_id cell_line platform
```

```
# 4. Run DESeq2 (fits model; you can use results later if you want)
dds_all <- DESeq(dds_all)
```

```
## estimating size factors
```

```
## estimating dispersions
```

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

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

```
## final dispersion estimates
```

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

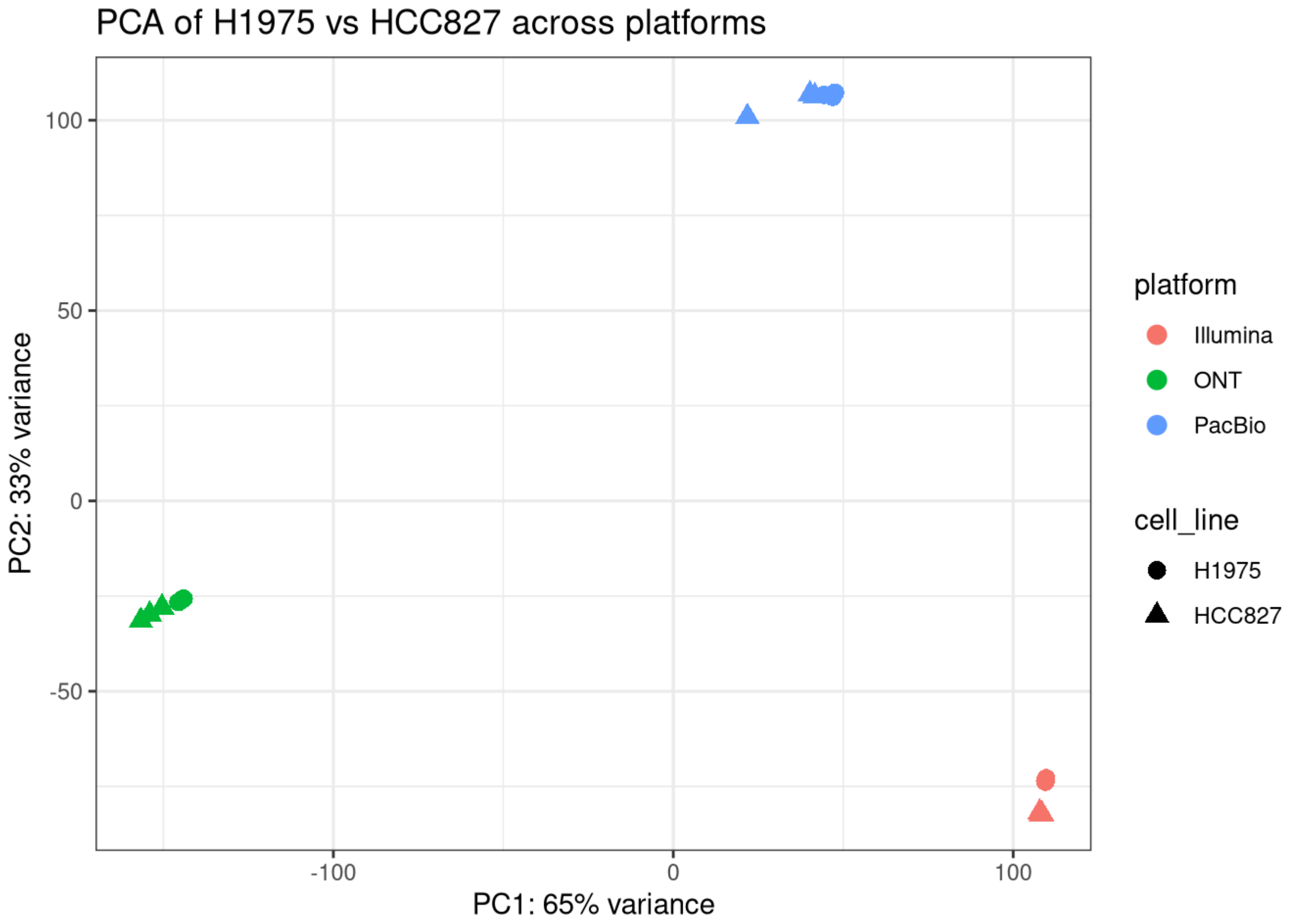
```
# 5. Variance stabilizing transform for PCA & clustering
vsd_all <- vst(dds_all, blind = TRUE)
```

```
## -- note: fitType='parametric', but the dispersion trend was not well captured by t
he
##    function: y = a/x + b, and a local regression fit was automatically substitute
d.
##    specify fitType='local' or 'mean' to avoid this message next time.
```

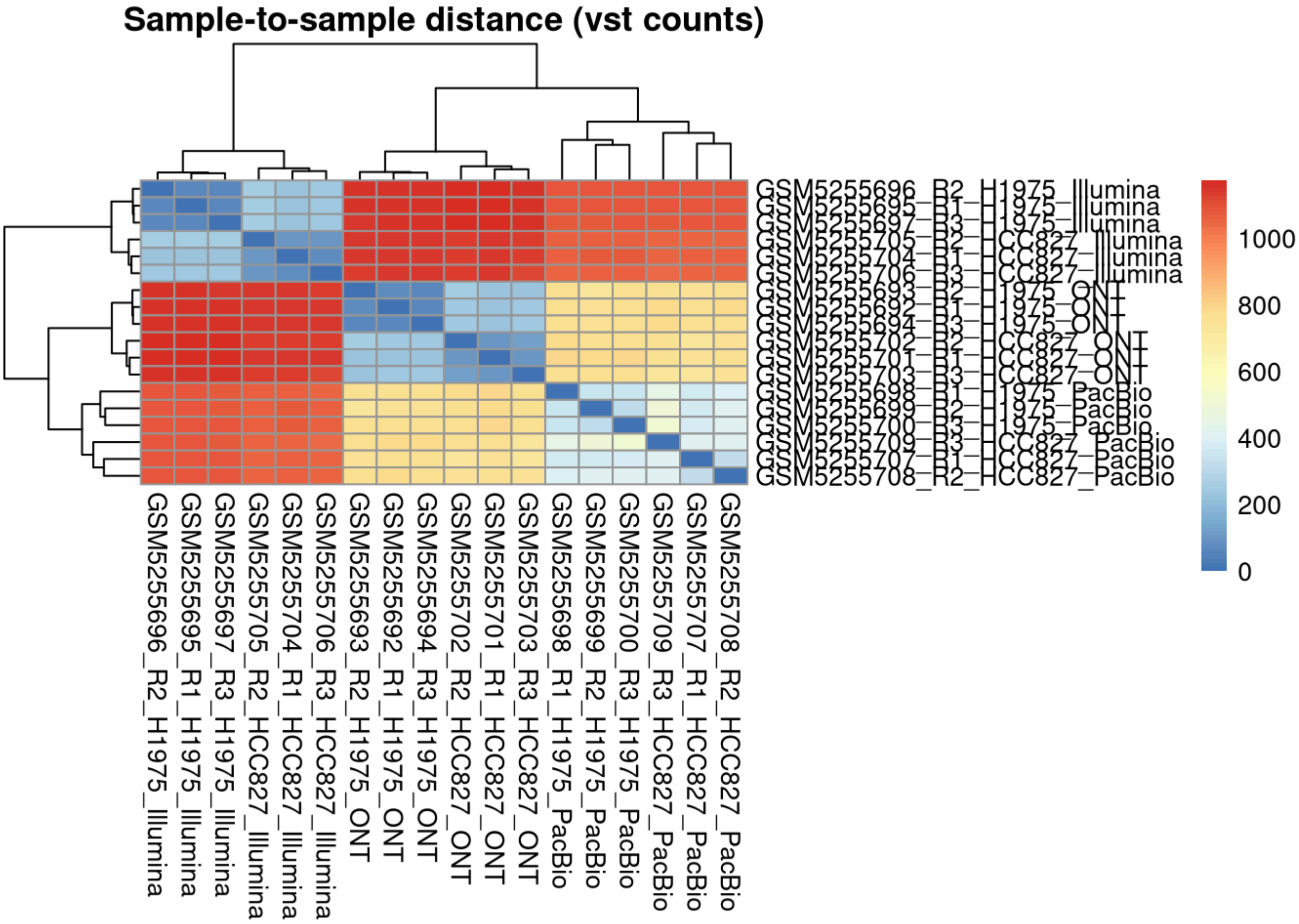
```
# 6. PCA plot (colour by platform, shape by cell line)
pca_data <- plotPCA(vsd_all, intgroup = c("platform", "cell_line"), returnData = TRU
E)
```

```
## using ntop=500 top features by variance
```

```
percentVar <- round(100 * attr(pca_data, "percentVar"))

library(ggplot2)

ggplot(pca_data, aes(PC1, PC2, color = platform, shape = cell_line)) +
  geom_point(size = 3) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  ggtitle("PCA of H1975 vs HCC827 across platforms") +
  theme_bw()
```



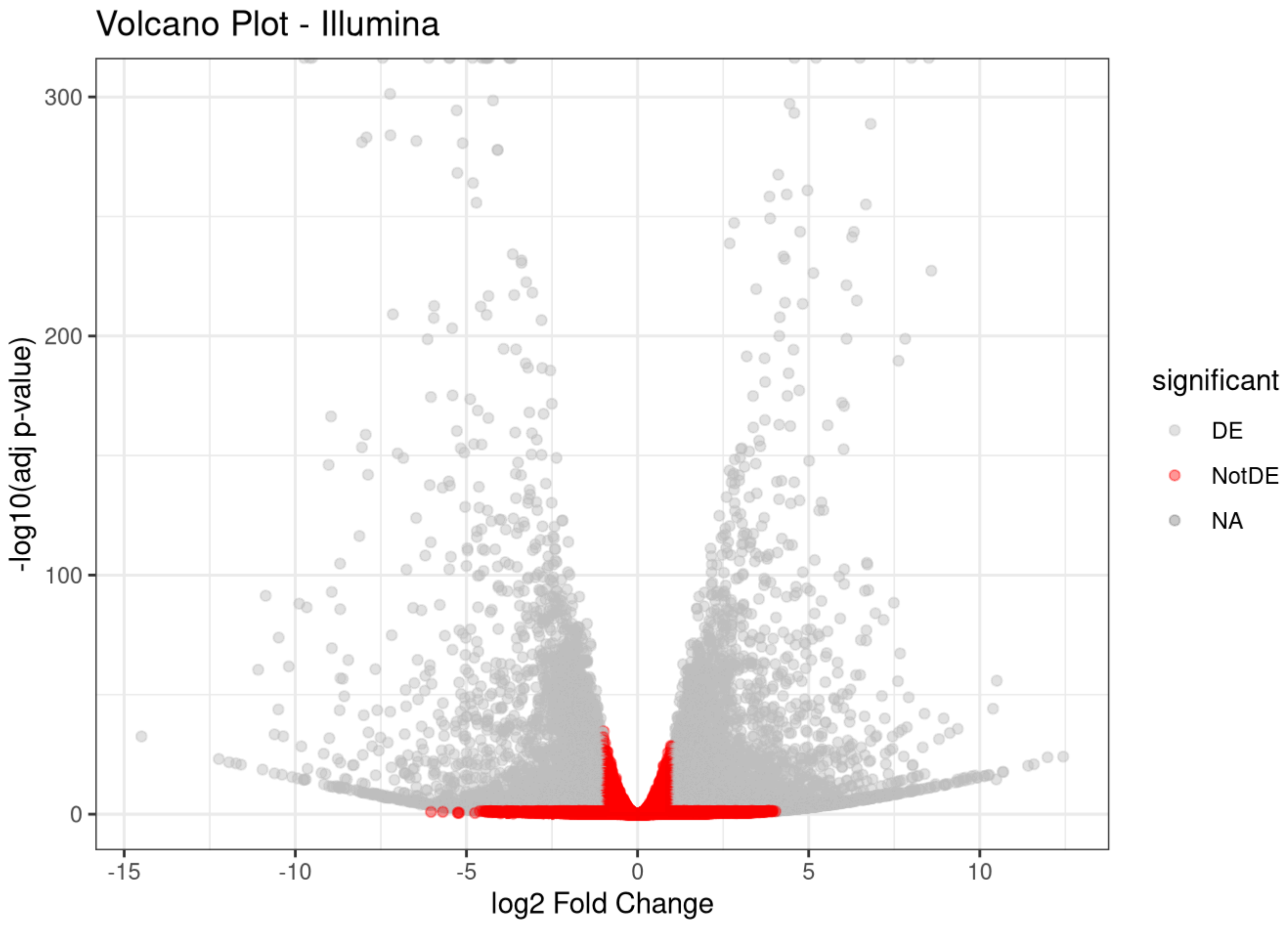
```
# 7. Sample-to-sample distance heatmap
sampleDists <- dist(t(assay(vsd_all)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- colnames(vsd_all)
colnames(sampleDistMatrix) <- colnames(vsd_all)

pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         main = "Sample-to-sample distance (vst counts)")
```


Sample-to-sample distance (vst counts)
GSM5255696_R2_H1975_Illumina
GSM5255695_R1_H1975_Illumina
GSM5255697_R3_H1975_Illumina
GSM5255705_R2_HCC827_Illumina
GSM5255704_R1_HCC827_Illumina
GSM5255706_R3_HCC827_Illumina
GSM5255693_R2_H1975_ONT
GSM5255692_R1_H1975_ONT
GSM5255694_R3_H1975_ONT
GSM5255702_R2_HCC827_ONT
GSM5255701_R1_HCC827_ONT
GSM5255703_R3_HCC827_ONT
GSM5255698_R1_H1975_PacBio
GSM5255699_R2_H1975_PacBio
GSM5255700_R3_H1975_PacBio
GSM5255709_R3_HCC827_PacBio
GSM5255707_R1_HCC827_PacBio
GSM5255708_R2_HCC827_PacBio
1000
800
600
400
200
0

```
####################################################################
```

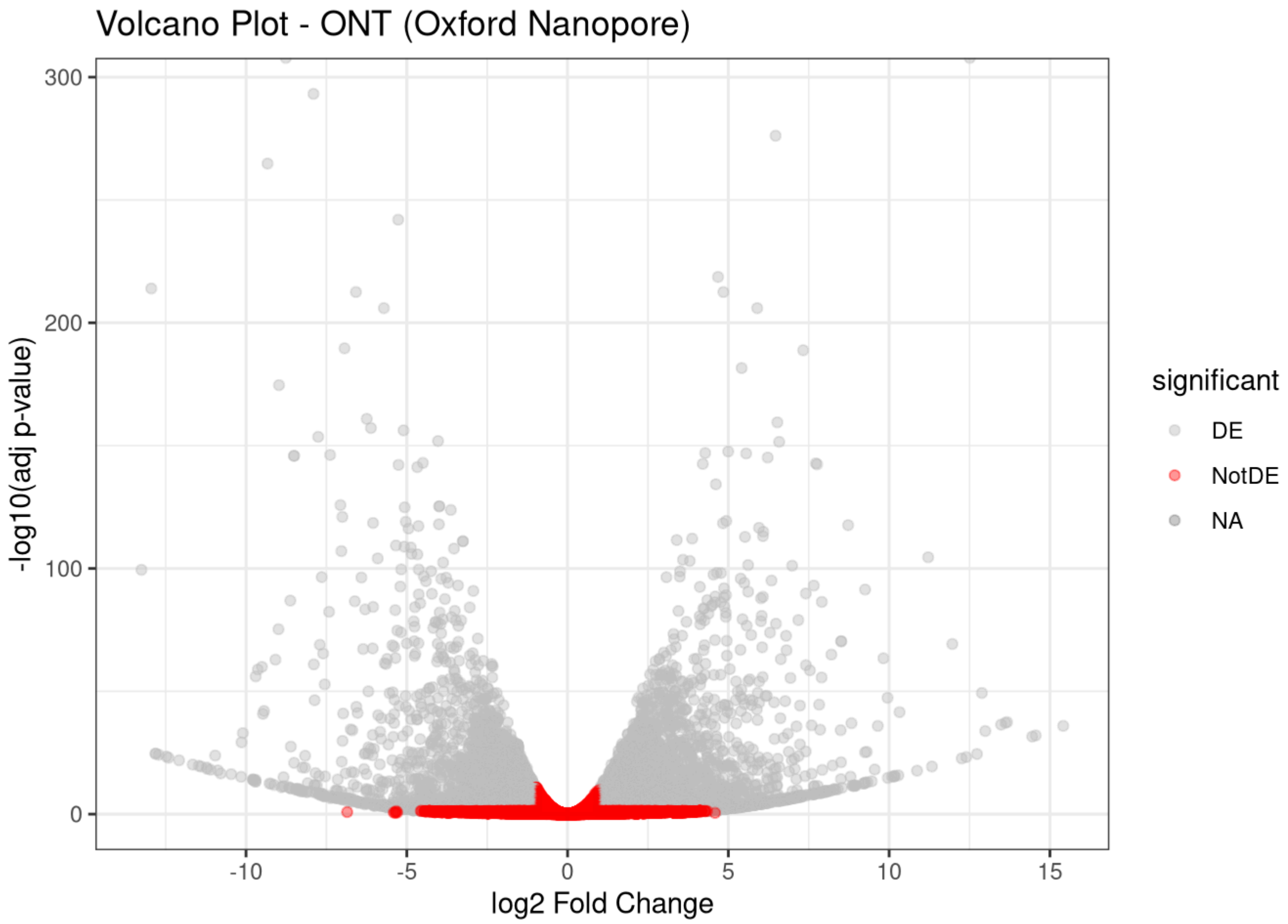
```
############################################################
## STEP 3: Volcano + MA plots for Illumina, ONT, PacBio
############################################################

library(ggplot2)

# Load DESeq2 results
ill <- read.csv("DESeq2_Illumina_H1975_vs_HCC827.csv", row.names = 1)
ont <- read.csv("DESeq2_ONT_H1975_vs_HCC827.csv", row.names = 1)
pb  <- read.csv("DESeq2_PacBio_H1975_vs_HCC827.csv", row.names = 1)

# Function to make volcano plot
make_volcano <- function(df, title) {
  df$neglog10padj <- -log10(df$padj)
  df$significant <- ifelse(df$padj < 0.05 & abs(df$log2FoldChange) >= 1, "DE", "NotD
E")

  ggplot(df, aes(x = log2FoldChange, y = neglog10padj, color = significant)) +
    geom_point(alpha = 0.4) +
    scale_color_manual(values = c("gray", "red")) +
    theme_bw() +
    ggtitle(paste("Volcano Plot -", title)) +
    xlab("log2 Fold Change") +
    ylab("-log10(adj p-value)")
}

# Function to make MA plot
make_ma <- function(df, title) {
  df$significant <- ifelse(df$padj < 0.05 & abs(df$log2FoldChange) >= 1, "DE", "NotD
E")

  ggplot(df, aes(x = baseMean, y = log2FoldChange, color = significant)) +
    geom_point(alpha = 0.4) +
    scale_x_log10() +
    scale_color_manual(values = c("gray", "blue")) +
    theme_bw() +
    ggtitle(paste("MA Plot -", title)) +
    xlab("Mean Expression (baseMean)") +
    ylab("log2 Fold Change")
}

### Volcano plots
make_volcano(ill, "Illumina")
```

```
## Warning: Removed 43091 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



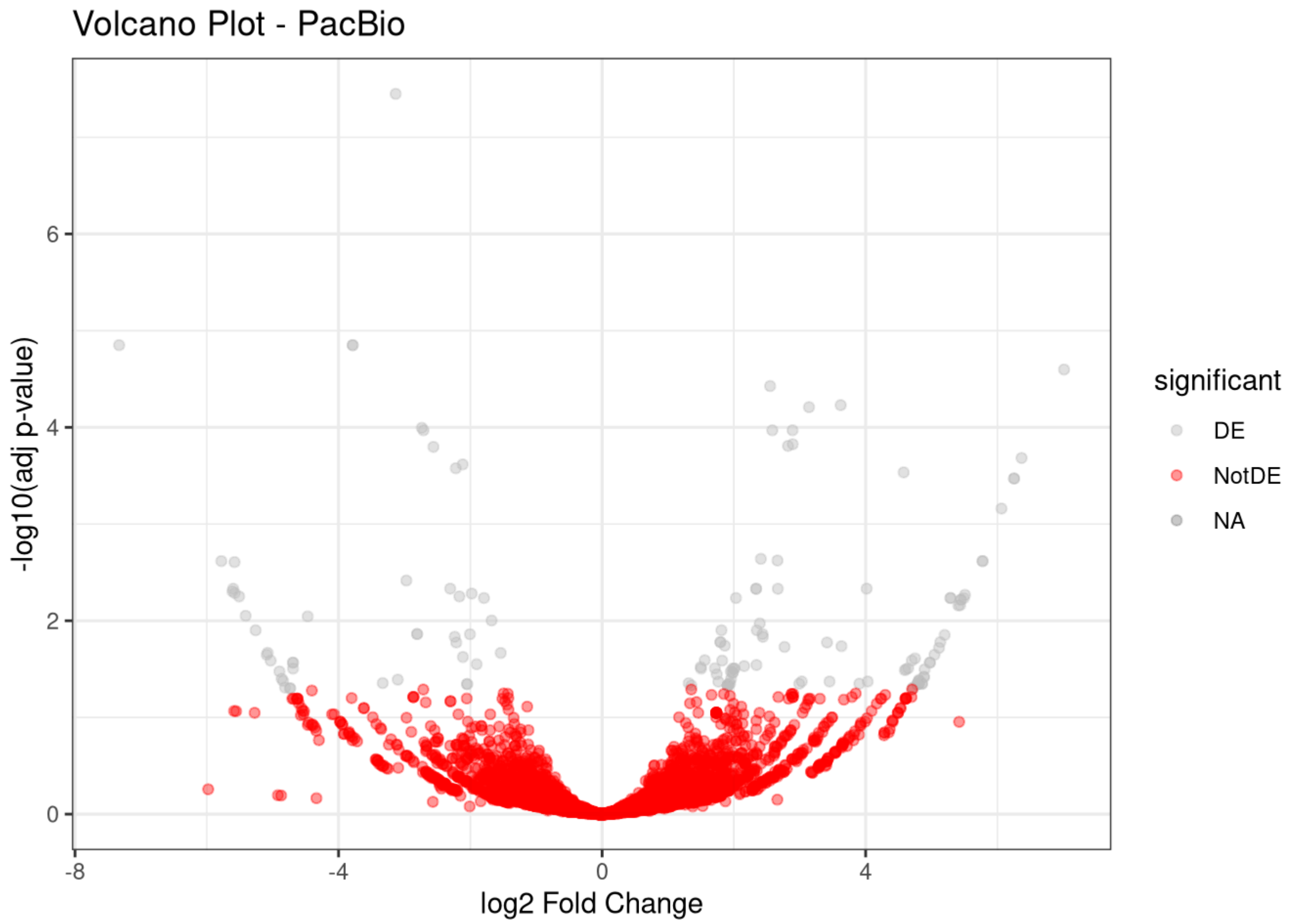
```
make_volcano(ont, "ONT (Oxford Nanopore)")
```

```
## Warning: Removed 41706 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



```
make_volcano(pb,  "PacBio")
```

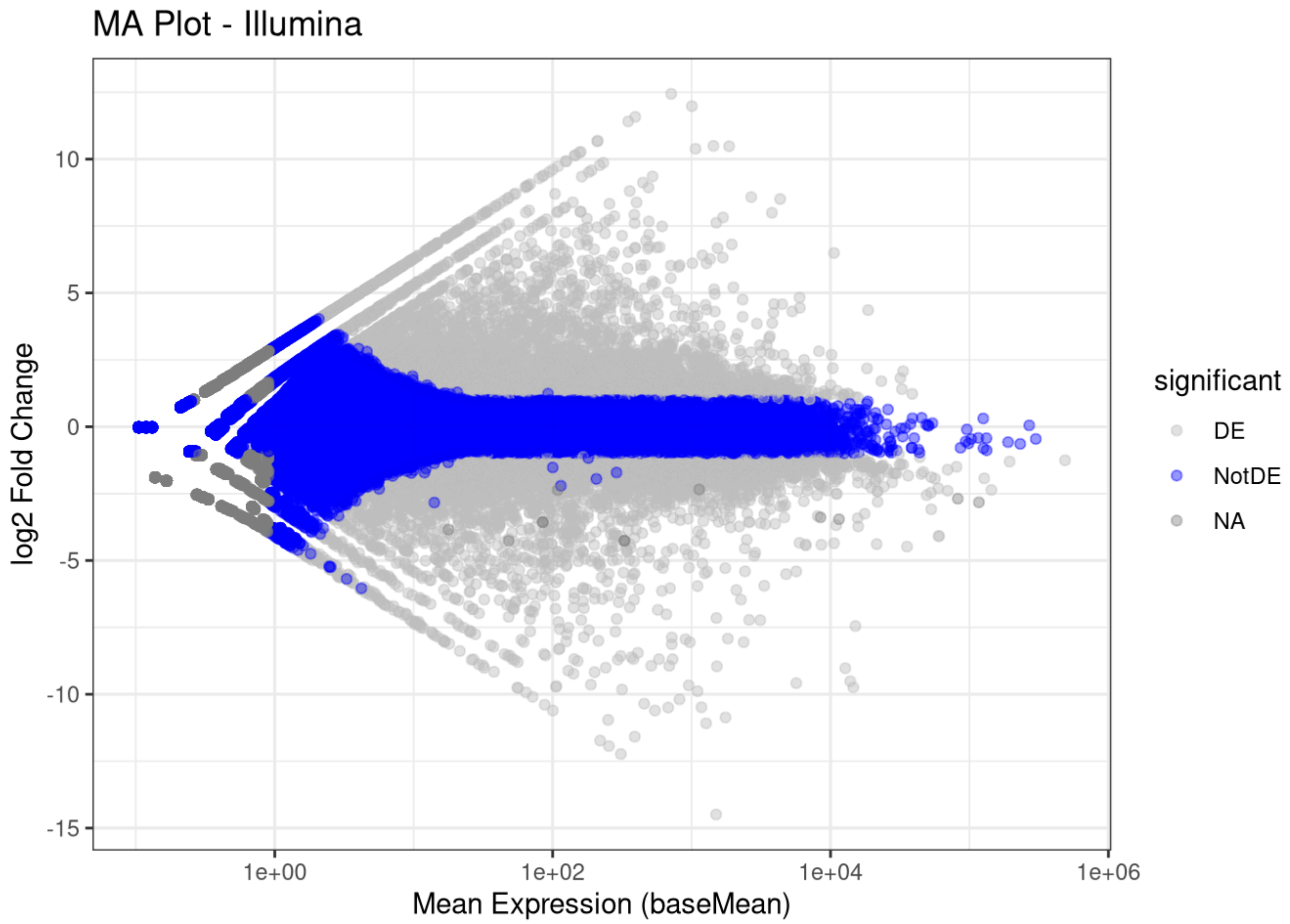
```
## Warning: Removed 71089 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



```
### MA plots
make_ma(ill, "Illumina")
```

```
## Warning in scale_x_log10(): log-10 transformation introduced infinite values.
```

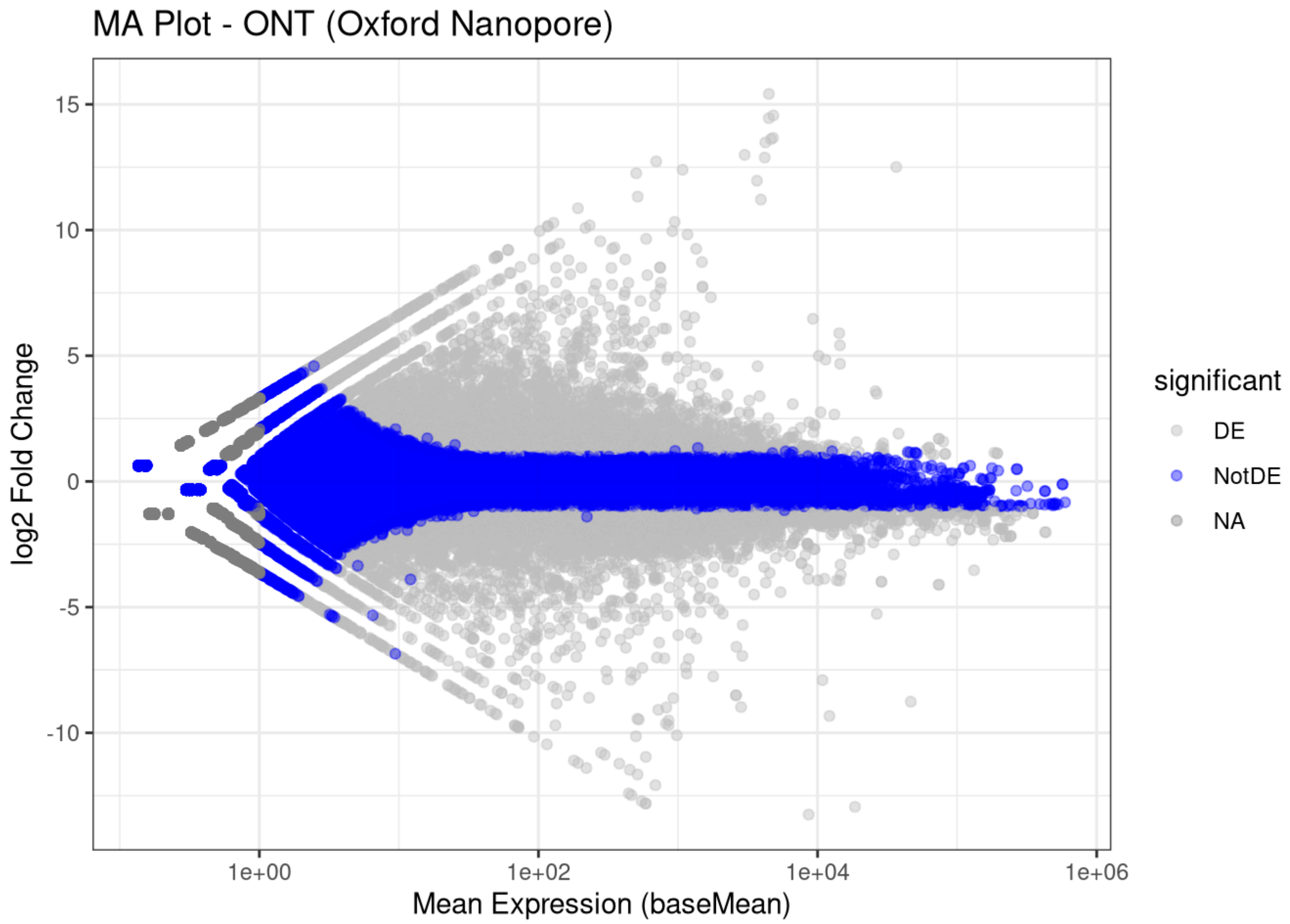
```
## Warning: Removed 31839 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



```
make_ma(ont, "ONT (Oxford Nanopore)")
```

```
## Warning in scale_x_log10(): log-10 transformation introduced infinite values.
```

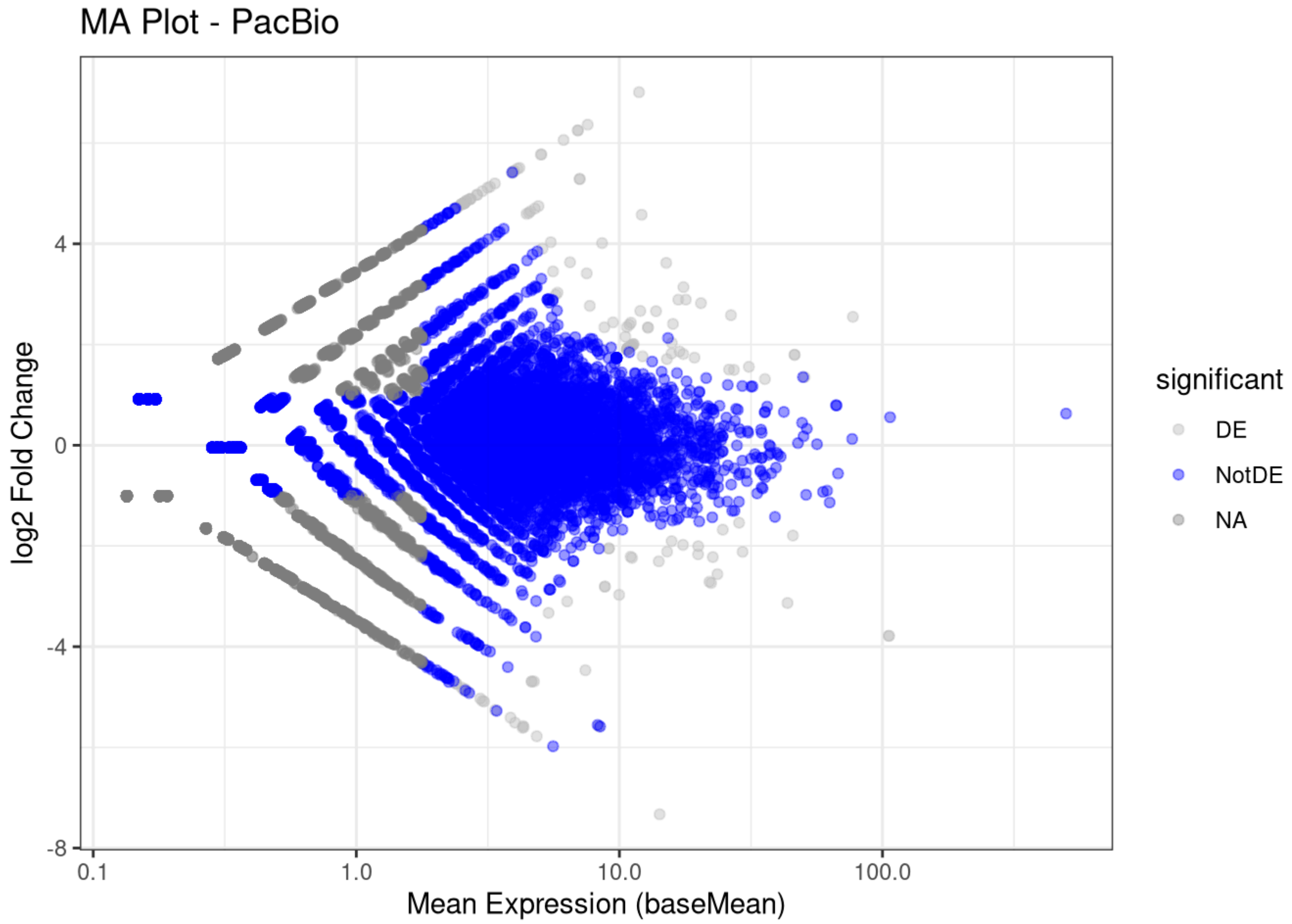
```
## Warning: Removed 33373 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



```
make_ma(pb,  "PacBio")
```

```
## Warning in scale_x_log10(): log-10 transformation introduced infinite values.
```

```
## Warning: Removed 63302 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



```
##############################################################
```

```
##############################################################
## STEP 4A: Venn diagram of DE genes across platforms
##############################################################

# install.packages("VennDiagram")  # run once if needed
library(VennDiagram)
```

```
## Loading required package: grid
```

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

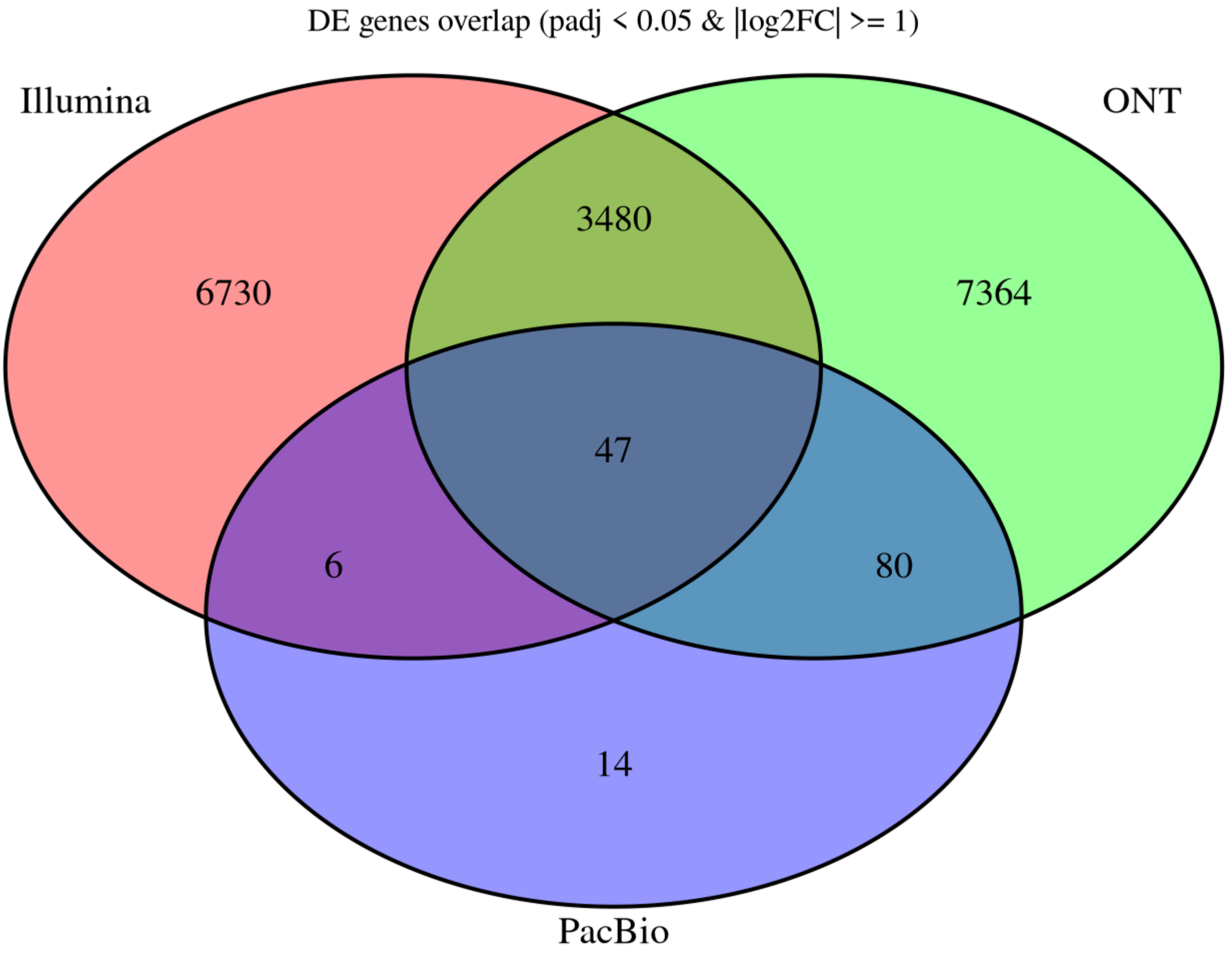
```
# Reload results (or reuse if already in memory)
ill <- read.csv("DESeq2_Illumina_H1975_vs_HCC827.csv", row.names = 1)
ont <- read.csv("DESeq2_ONT_H1975_vs_HCC827.csv", row.names = 1)
pb  <- read.csv("DESeq2_PacBio_H1975_vs_HCC827.csv", row.names = 1)

# Extract DE gene sets
sig_ill <- rownames(ill)[which(ill$padj < 0.05 & abs(ill$log2FoldChange) >= 1)]
sig_ont <- rownames(ont)[which(ont$padj < 0.05 & abs(ont$log2FoldChange) >= 1)]
sig_pb  <- rownames(pb)[which(pb$padj  < 0.05 & abs(pb$log2FoldChange)  >= 1)]

length(sig_ill); length(sig_ont); length(sig_pb)
```

```
## [1] 10263
```

```
## [1] 10971
```

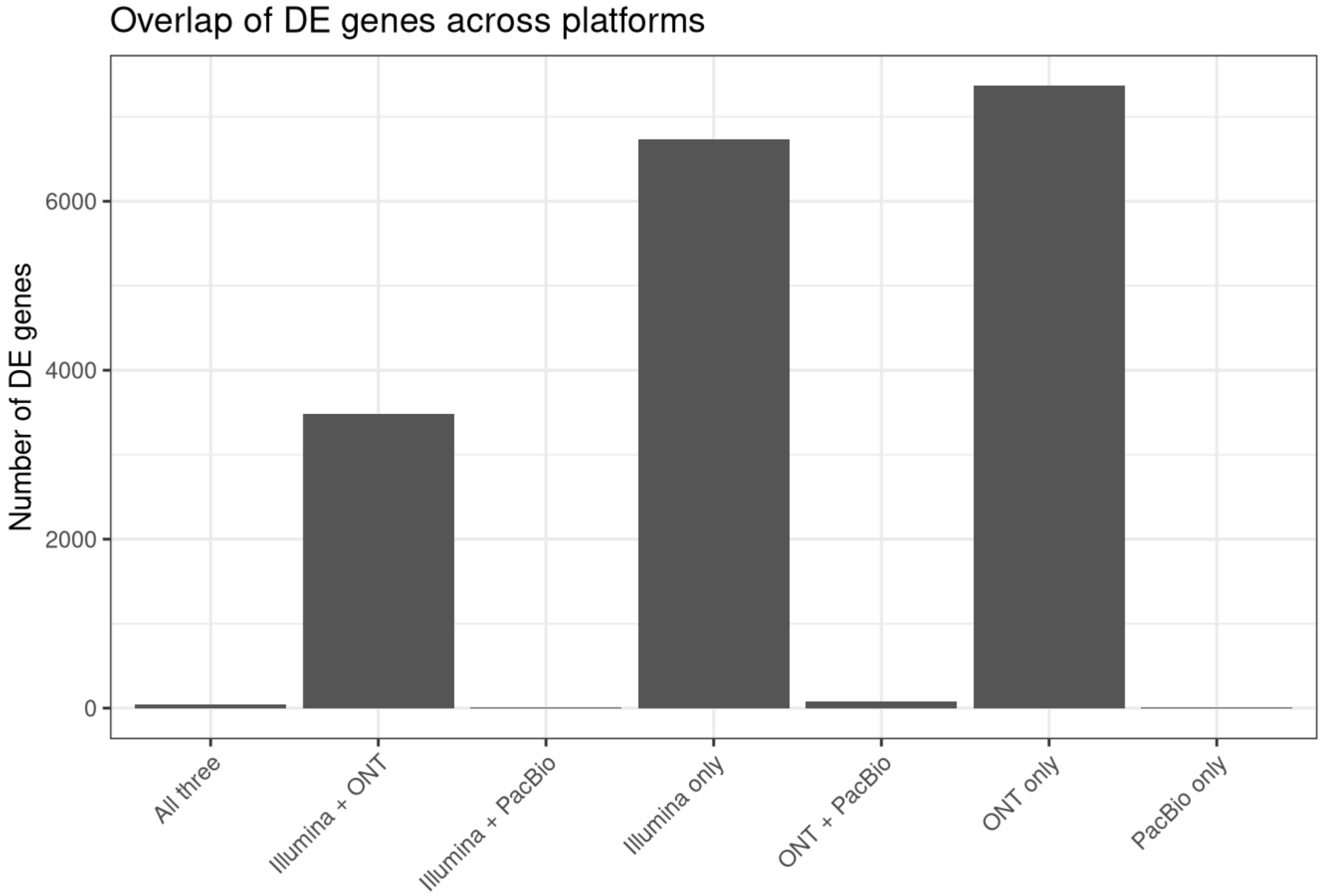
```
## [1] 147
```

```
venn.plot <- venn.diagram(
  x = list(
    Illumina = sig_ill,
    ONT      = sig_ont,
    PacBio   = sig_pb
  ),
  filename = NULL,  # draw to R graphics device
  fill = c("red", "green", "blue"),
  alpha = 0.4,
  cex = 1.2,
  cat.cex = 1.2,
  main = "DE genes overlap (padj < 0.05 & |log2FC| >= 1)"
)

grid::grid.newpage()
grid::grid.draw(venn.plot)
```



```
##################################################################
```

```r
# assumes sig_ill, sig_ont, sig_pb are already defined

library(ggplot2)

ill_only <- setdiff(sig_ill, union(sig_ont, sig_pb))
ont_only <- setdiff(sig_ont, union(sig_ill, sig_pb))
pb_only  <- setdiff(sig_pb,  union(sig_ill, sig_ont))

ill_ont_only <- setdiff(intersect(sig_ill, sig_ont), sig_pb)
ill_pb_only  <- setdiff(intersect(sig_ill, sig_pb),  sig_ont)
ont_pb_only  <- setdiff(intersect(sig_ont, sig_pb),  sig_ill)

all_three <- Reduce(intersect, list(sig_ill, sig_ont, sig_pb))

overlap_df <- data.frame(
  group = c("Illumina only",
            "ONT only",
            "PacBio only",
            "Illumina + ONT",
            "Illumina + PacBio",
            "ONT + PacBio",
            "All three"),
  n = c(length(ill_only),
        length(ont_only),
        length(pb_only),
        length(ill_ont_only),
        length(ill_pb_only),
        length(ont_pb_only),
        length(all_three))
)

ggplot(overlap_df, aes(x = group, y = n)) +
  geom_col() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylab("Number of DE genes") +
  xlab("") +
  ggtitle("Overlap of DE genes across platforms")
```



```
#############################################################
## STEP 5: GO & KEGG Enrichment for the 47 shared DE genes
#############################################################

# Load results
ill <- read.csv("DESeq2_Illumina_H1975_vs_HCC827.csv", row.names = 1)
ont <- read.csv("DESeq2_ONT_H1975_vs_HCC827.csv", row.names = 1)
pb  <- read.csv("DESeq2_PacBio_H1975_vs_HCC827.csv", row.names = 1)

# Define DE genes
sig_ill <- rownames(ill)[which(ill$padj < 0.05 & abs(ill$log2FoldChange) >= 1)]
sig_ont <- rownames(ont)[which(ont$padj < 0.05 & abs(ont$log2FoldChange) >= 1)]
sig_pb  <- rownames(pb)[which(pb$padj  < 0.05 & abs(pb$log2FoldChange)  >= 1)]

# The 47 genes shared by all platforms
shared47 <- Reduce(intersect, list(sig_ill, sig_ont, sig_pb))
length(shared47)
```

```
## [1] 47
```

```
shared47[1:10]
```

```
##  [1] "ENSG00000100867.14" "ENSG00000102572.14" "ENSG00000117983.17"
##  [4] "ENSG00000132434.9"  "ENSG00000108821.13" "ENSG00000268833.1"
##  [7] "ENSG00000257342.1"  "ENSG00000249835.2"  "ENSG00000197608.11"
## [10] "ENSG00000139998.15"
```

```
############################################################
```

```
############################################################
## GO + KEGG enrichment for DE genes shared by all platforms
## (Illumina, ONT, PacBio: H1975 vs HCC827)
############################################################

# Load required packages
library(clusterProfiler)
```

```
##
```

```
## clusterProfiler v4.14.6 Learn more at https://yulab-smu.top/contribution-knowledge
-mining/
##
## Please cite:
##
## Guangchuang Yu, Li-Gen Wang, Yanyan Han and Qing-Yu He.
## clusterProfiler: an R package for comparing biological themes among
## gene clusters. OMICS: A Journal of Integrative Biology. 2012,
## 16(5):284-287
```

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

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

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

```
## The following object is masked from 'package:purrr':
##
##     simplify
```

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

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

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

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

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

```
##
```

```
############################################################
## 1. Load DESeq2 results
############################################################

ill <- read.csv("DESeq2_Illumina_H1975_vs_HCC827.csv", row.names = 1)
ont <- read.csv("DESeq2_ONT_H1975_vs_HCC827.csv", row.names = 1)
pb  <- read.csv("DESeq2_PacBio_H1975_vs_HCC827.csv", row.names = 1)

############################################################
## 2. Define DE gene sets per platform
##    Criteria: padj < 0.05 and |log2FC| >= 1
############################################################

sig_ill <- rownames(ill)[which(ill$padj < 0.05 & abs(ill$log2FoldChange) >= 1)]
sig_ont <- rownames(ont)[which(ont$padj < 0.05 & abs(ont$log2FoldChange) >= 1)]
sig_pb  <- rownames(pb)[which(pb$padj  < 0.05 & abs(pb$log2FoldChange)  >= 1)]

cat("DE genes per platform (padj < 0.05, |log2FC| >= 1):\n")
```

```
## DE genes per platform (padj < 0.05, |log2FC| >= 1):
```

```
cat("Illumina:", length(sig_ill), "\n")
```

```
## Illumina: 10263
```

```
cat("ONT:     ", length(sig_ont), "\n")
```

```
## ONT:      10971
```

```
cat("PacBio:  ", length(sig_pb),  "\n\n")
```

```
## PacBio:   147
```

```
############################################################
## 3. Find genes shared by ALL THREE platforms
############################################################

shared_genes <- Reduce(intersect, list(sig_ill, sig_ont, sig_pb))
cat("Genes shared by all three platforms:", length(shared_genes), "\n\n")
```

```
## Genes shared by all three platforms: 47
```

```
# Look at first few
head(shared_genes)
```

```
## [1] "ENSG00000100867.14" "ENSG00000102572.14" "ENSG00000117983.17"
## [4] "ENSG00000132434.9"  "ENSG00000108821.13" "ENSG00000268833.1"
```

```
############################################################
## 4. Strip Ensembl version suffix (e.g. .16)
##    ENSG00000100033.16 -> ENSG00000100033
############################################################

shared_novers <- sub("\\.\\d+$", "", shared_genes)

head(shared_genes)
```

```
## [1] "ENSG00000100867.14" "ENSG00000102572.14" "ENSG00000117983.17"
## [4] "ENSG00000132434.9"  "ENSG00000108821.13" "ENSG00000268833.1"
```

```
head(shared_novers)
```

```
## [1] "ENSG00000100867" "ENSG00000102572" "ENSG00000117983" "ENSG00000132434"
## [5] "ENSG00000108821" "ENSG00000268833"
```

```
############################################################
## 5. Map Ensembl IDs -> Entrez IDs
############################################################

entrez_shared <- mapIds(
  org.Hs.eg.db,
  keys      = shared_novers,
  keytype   = "ENSEMBL",
  column    = "ENTREZID",
  multiVals = "first"
)
```

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

```
# Drop genes that failed to map
entrez_shared <- na.omit(entrez_shared)
cat("Mapped to Entrez IDs:", length(entrez_shared), "genes\n\n")
```

```
## Mapped to Entrez IDs: 32 genes
```

```
entrez_shared[1:10]
```

```
## ENSG00000100867 ENSG00000102572 ENSG00000117983 ENSG00000132434 ENSG00000108821
##         "10202"          "8428"        "727897"         "55915"          "1277"
## ENSG00000249835 ENSG00000197608 ENSG00000139998 ENSG00000198286 ENSG00000160161
##     "105379054"        "284371"        "376267"         "84433"        "148113"
```

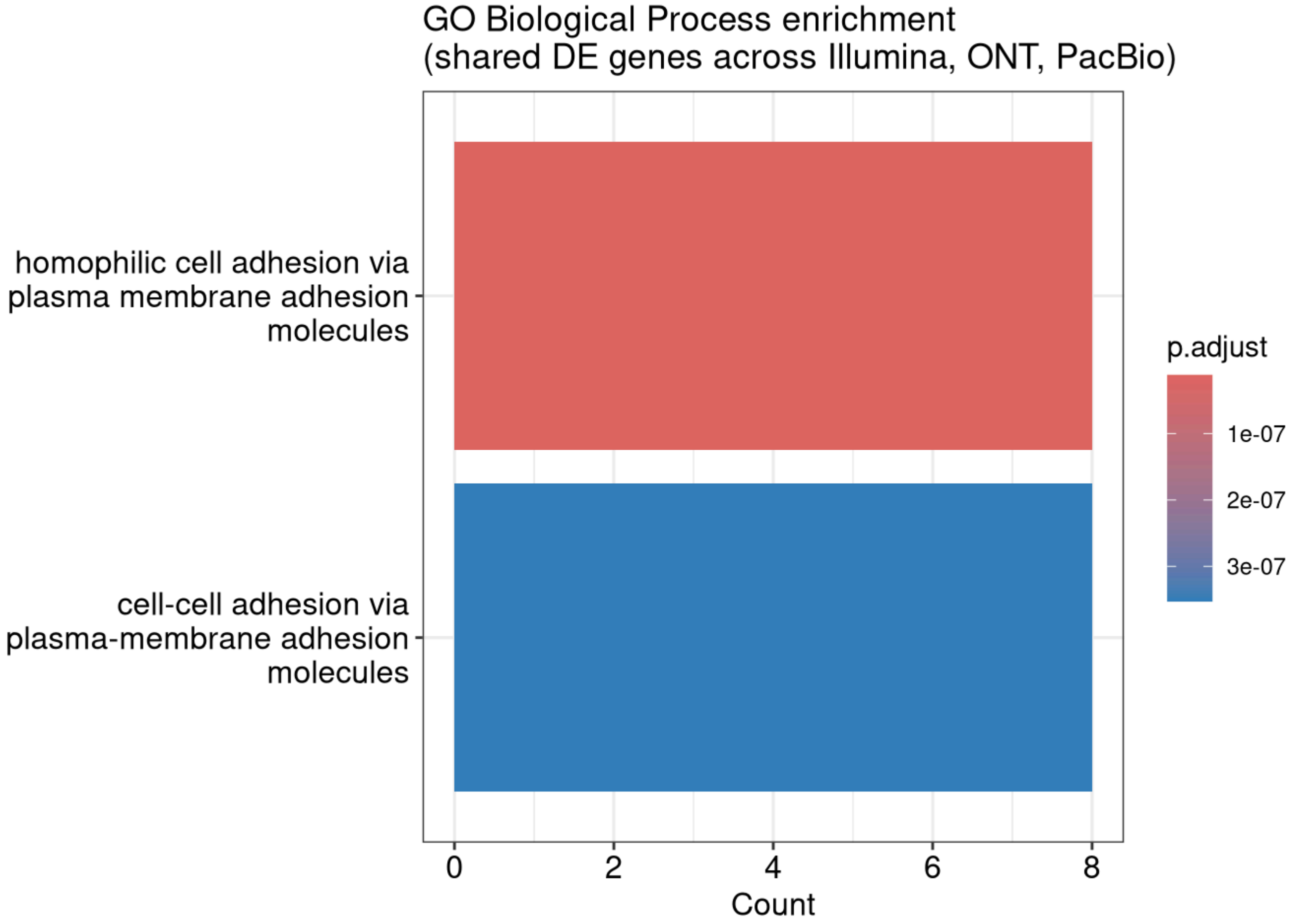
```
############################################################
## 6. GO Biological Process (BP) enrichment
############################################################

ego <- enrichGO(
  gene          = entrez_shared,
  OrgDb         = org.Hs.eg.db,
  keyType       = "ENTREZID",
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.05,
  qvalueCutoff  = 0.05,
  readable      = TRUE
)

cat("Top GO BP terms:\n")
```

```
## Top GO BP terms:
```

```
print(head(ego, 5))
```

```
##                     ID
## GO:0007156 GO:0007156
## GO:0098742 GO:0098742
##                                                                 Description
## GO:0007156 homophilic cell adhesion via plasma membrane adhesion molecules
## GO:0098742       cell-cell adhesion via plasma-membrane adhesion molecules
##            GeneRatio   BgRatio RichFactor FoldEnrichment   zScore       pvalue
## GO:0007156      8/24 168/18986 0.04761905       37.67063 16.98459 2.072416e-11
## GO:0098742      8/24 280/18986 0.02857143       22.60238 12.95564 1.213845e-09
##                p.adjust       qvalue
## GO:0007156 1.204074e-08 9.664005e-09
## GO:0098742 3.526220e-07 2.830176e-07
##                                                                      geneID
## GO:0007156 PCDHGA1/PCDHGA3/PCDHGB1/PCDHGB2/PCDHGA5/PCDHGB4/PCDHGA6/PCDHGA8
## GO:0098742 PCDHGA1/PCDHGA3/PCDHGB1/PCDHGB2/PCDHGA5/PCDHGB4/PCDHGA6/PCDHGA8
##            Count
## GO:0007156     8
## GO:0098742     8
```

```
# Barplot of top 15 GO BP terms
barplot(ego,
        showCategory = 15,
        title = "GO Biological Process enrichment\n(shared DE genes across Illumina, ONT, PacBio)")
```



```
############################################################
## 7. KEGG pathway enrichment
############################################################

ekegg <- enrichKEGG(
  gene         = entrez_shared,
  organism     = "hsa",
  pvalueCutoff = 0.05
)
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/link/hsa/pathway"...
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/hsa"...
```

```
cat("\nTop KEGG pathways:\n")
```

```
##
## Top KEGG pathways:
```

```
print(head(ekegg, 5))
```

```
##  [1] category       subcategory    ID             Description    GeneRatio
##  [6] BgRatio        RichFactor     FoldEnrichment zScore         pvalue
## [11] p.adjust       qvalue         geneID         Count
## <0 rows> (or 0-length row.names)
```

```
# Barplot of top 10 KEGG pathways (if any)
if (nrow(as.data.frame(ekegg)) > 0) {
  barplot(ekegg,
          showCategory = 10,
          title = "KEGG pathway enrichment\n(shared DE genes across Illumina, ONT, PacBio)")
} else {
  cat("\nNo significant KEGG pathways at pvalueCutoff = 0.05\n")
}
```

```
##
## No significant KEGG pathways at pvalueCutoff = 0.05
```

```
############################################################
## END
############################################################
```