

# embryo\_L1\_L3\_intestine\_RNAseq

Rtpw

2/8/2022

Install libraries

Load libraries

```
source(file = "../..../elt2_paper_functions.R")
```

## Analysis outline

- Import counts and format for DESeq2 analysis
- Filter low abundance counts based on counts per million (CPM)
- Filter for protein coding genes
- Perform differential expression analysis
- Visualize pairwise sample correlation
- Categorize gene expression with alternative hypothesis
- Visualize unshrunken pairwise differential expression
- Visualize shrunken pairwise differential expression
- Measure tissue expressed genes
- Visualize developmentally dynamic intestine genes
- Identify novel intestine genes in each stage for smFISH experiments

## Import and format count data

```
# import counts
countsData <- read.delim(file = "../input/all.counts", sep = " ")
# preview counts
head(countsData)

##          chr start stop strand length embryo_cells_rep1
## WBGene00014450 MtDNA     1    55      +    55            0
## WBGene00014451 MtDNA    58   111      +    54            0
## WBGene00010957 MtDNA   113   549      +   437            0
## WBGene00010958 MtDNA   549   783      +   235            0
## WBGene00014452 MtDNA   785   840      +    56            0
## WBGene00014453 MtDNA   842   896      +    55            0
##          embryo_cells_rep2 embryo_GFPminus_rep1 embryo_GFPminus_rep2
## WBGene00014450            0                  0                  0
## WBGene00014451            0                  0                  0
## WBGene00010957            0                  0                  0
## WBGene00010958            0                  0                  0
## WBGene00014452            0                  0                  0
## WBGene00014453            0                  0                  0
##          embryo_GFPminus_rep3 embryo_GFPplus_rep1 embryo_GFPplus_rep2
```

```

## WBGene00014450      0      0      0
## WBGene00014451      0      0      0
## WBGene00010957      0      0      0
## WBGene00010958      0      0      0
## WBGene00014452      0      0      0
## WBGene00014453      0      0      0
##          embryo_GFPplus_rep3 embryo_whole_rep2 embryo_whole_rep3
## WBGene00014450      0      0      0
## WBGene00014451      0      0      0
## WBGene00010957      0      0      0
## WBGene00010958      0      0      0
## WBGene00014452      0      0      0
## WBGene00014453      0      0      0
##          L1_cells_rep1 L1_cells_rep2 L1_cells_rep3 L1_GFPminus_rep1
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0
## WBGene00014453      0      0      0      0
##          L1_GFPminus_rep2 L1_GFPminus_rep3 L1_GFPplus_rep1
## WBGene00014450      0      0      0
## WBGene00014451      0      0      0
## WBGene00010957      0      0      0
## WBGene00010958      0      0      0
## WBGene00014452      0      0      0
## WBGene00014453      0      0      0
##          L1_GFPplus_rep2 L1_GFPplus_rep3 L1_whole_rep1 L1_whole_rep2
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0
## WBGene00014453      0      0      0      0
##          L1_whole_rep3 L3_cells_rep1 L3_cells_rep2 L3_cells_rep3
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0
## WBGene00014453      0      0      0      0
##          L3_GFPminus_rep1 L3_GFPplus_rep2 L3_GFPminus_rep3
## WBGene00014450      0      0      0
## WBGene00014451      0      0      0
## WBGene00010957      0      0      0
## WBGene00010958      0      0      0
## WBGene00014452      0      0      0
## WBGene00014453      0      0      0
##          L3_GFPplus_rep1 L3_GFPminus_rep2 L3_GFPplus_rep3 L3_whole_rep1
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0

```

```

## WBGene00014453          0          0          0
##           L3_whole_rep2 L3_whole_rep3
## WBGene00014450          0          0
## WBGene00014451          0          0
## WBGene00010957          0          0
## WBGene00010958          0          0
## WBGene00014452          0          0
## WBGene00014453          0          0

# print samples
colnames(countsData[6:ncol(countsData)])

## [1] "embryo_cells_rep1"      "embryo_cells_rep2"      "embryo_GFPminus_rep1"
## [4] "embryo_GFPminus_rep2"    "embryo_GFPminus_rep3"    "embryo_GFPplus_rep1"
## [7] "embryo_GFPplus_rep2"    "embryo_GFPplus_rep3"    "embryo_whole_rep2"
## [10] "embryo_whole_rep3"     "L1_cells_rep1"        "L1_cells_rep2"
## [13] "L1_cells_rep3"         "L1_GFPminus_rep1"    "L1_GFPminus_rep2"
## [16] "L1_GFPminus_rep3"     "L1_GFPplus_rep1"    "L1_GFPplus_rep2"
## [19] "L1_GFPplus_rep3"      "L1_whole_rep1"       "L1_whole_rep2"
## [22] "L1_whole_rep3"        "L3_cells_rep1"       "L3_cells_rep2"
## [25] "L3_cells_rep3"        "L3_GFPminus_rep1"   "L3_GFPplus_rep2"
## [28] "L3_GFPminus_rep3"     "L3_GFPplus_rep1"   "L3_GFPminus_rep2"
## [31] "L3_GFPplus_rep3"      "L3_whole_rep1"      "L3_whole_rep2"
## [34] "L3_whole_rep3"

# import metadata and process file
metadata1 <- read.table(file = "../01_input/RWP27_metadata.tsv", header = FALSE, stringsAsFactors = FALSE)
bind_rows(read.table(file = "../01_input/RWP30_metadata.tsv", header = FALSE, stringsAsFactors = FALSE))

colnames(metadata1) <- c("Filename.Fwd", "Filename.Rev", "names")
head(metadata1)

##             Filename.Fwd      Filename.Rev      names
## 1 RW57_S10_L003_R1_001 RW57_S10_L003_R2_001 embryo_cells_rep1
## 2 RW58_S11_L003_R1_001 RW58_S11_L003_R2_001 embryo_GFPplus_rep1
## 3 RW59_S12_L003_R1_001 RW59_S12_L003_R2_001 embryo_GFPminus_rep1
## 4 RW60_S13_L003_R1_001 RW60_S13_L003_R2_001 embryo_whole_rep2
## 5 RW61_S14_L003_R1_001 RW61_S14_L003_R2_001 embryo_cells_rep2
## 6 RW62_S15_L003_R1_001 RW62_S15_L003_R2_001 embryo_GFPplus_rep2

# separate and process sample info
metadata1 <- metadata1 %>% separate(names, sep = "_", into = c("stage", "sample", "rep"), remove = FALSE)
metadata1 <- metadata1 %>% mutate(stage = fct_relevel(stage, c("embryo", "L1", "L3")),
                                    sample = fct_relevel(sample, c("whole", "cells", "GFPplus", "GFPminus")),
                                    rep = fct_relevel(rep, c("rep1", "rep2", "rep3")),
                                    names = fct_relevel(names, metadata1$names)
                                   )

# Order columns according to metadata1 order
countsData <- countsData %>% select(chr:length, sort(metadata1$names))
head(countsData)

##             chr start stop strand length embryo_cells_rep1
## WBGene00014450 MtDNA    1   55    +    55          0
## WBGene00014451 MtDNA   58  111    +    54          0
## WBGene00010957 MtDNA  113  549    +   437          0
## WBGene00010958 MtDNA  549  783    +   235          0

```

## WBGene00014452	MtDNA	785	840	+	56	0
## WBGene00014453	MtDNA	842	896	+	55	0
		embryo_GFPplus_rep1 embryo_GFPminus_rep1 embryo_whole_rep2				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0
## WBGene00010957			0		0	0
## WBGene00010958			0		0	0
## WBGene00014452			0		0	0
## WBGene00014453			0		0	0
		embryo_cells_rep2 embryo_GFPplus_rep2 embryo_GFPminus_rep2				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0
## WBGene00010957			0		0	0
## WBGene00010958			0		0	0
## WBGene00014452			0		0	0
## WBGene00014453			0		0	0
		embryo_whole_rep3 embryo_GFPplus_rep3 embryo_GFPminus_rep3				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0
## WBGene00010957			0		0	0
## WBGene00010958			0		0	0
## WBGene00014452			0		0	0
## WBGene00014453			0		0	0
		L1_whole_rep1 L1_cells_rep1 L1_GFPplus_rep1 L1_GFPminus_rep1				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0
## WBGene00010957			0		0	0
## WBGene00010958			0		0	0
## WBGene00014452			0		0	0
## WBGene00014453			0		0	0
		L1_whole_rep2 L1_cells_rep2 L1_GFPplus_rep2 L1_GFPminus_rep2				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0
## WBGene00010957			0		0	0
## WBGene00010958			0		0	0
## WBGene00014452			0		0	0
## WBGene00014453			0		0	0
		L1_whole_rep3 L1_cells_rep3 L1_GFPplus_rep3 L1_GFPminus_rep3				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0
## WBGene00010957			0		0	0
## WBGene00010958			0		0	0
## WBGene00014452			0		0	0
## WBGene00014453			0		0	0
		L3_whole_rep1 L3_cells_rep1 L3_GFPplus_rep1 L3_GFPminus_rep1				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0
## WBGene00010957			0		0	0
## WBGene00010958			0		0	0
## WBGene00014452			0		0	0
## WBGene00014453			0		0	0
		L3_whole_rep2 L3_cells_rep2 L3_GFPminus_rep2 L3_GFPplus_rep2				
## WBGene00014450			0		0	0
## WBGene00014451			0		0	0

```

## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0
## WBGene00014453      0      0      0      0
##          L3_whole_rep3 L3_cells_rep3 L3_GFPplus_rep3 L3_GFPminus_rep3
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0
## WBGene00014453      0      0      0      0

# Generate a table called "cts" out of the countsData table. Subset the countsData.
cts <- as.matrix(countsData %>% select(metadata1$names))
head(cts)

##          embryo_cells_rep1 embryo_GFPplus_rep1 embryo_GFPminus_rep1
## WBGene00014450      0      0      0
## WBGene00014451      0      0      0
## WBGene00010957      0      0      0
## WBGene00010958      0      0      0
## WBGene00014452      0      0      0
## WBGene00014453      0      0      0
##          embryo_whole_rep2 embryo_cells_rep2 embryo_GFPplus_rep2
## WBGene00014450      0      0      0
## WBGene00014451      0      0      0
## WBGene00010957      0      0      0
## WBGene00010958      0      0      0
## WBGene00014452      0      0      0
## WBGene00014453      0      0      0
##          embryo_GFPminus_rep2 embryo_whole_rep3 embryo_GFPplus_rep3
## WBGene00014450      0      0      0
## WBGene00014451      0      0      0
## WBGene00010957      0      0      0
## WBGene00010958      0      0      0
## WBGene00014452      0      0      0
## WBGene00014453      0      0      0
##          embryo_GFPminus_rep3 L1_whole_rep1 L1_cells_rep1 L1_GFPplus_rep1
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0
## WBGene00014453      0      0      0      0
##          L1_GFPminus_rep1 L1_whole_rep2 L1_cells_rep2 L1_GFPplus_rep2
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0
## WBGene00010958      0      0      0      0
## WBGene00014452      0      0      0      0
## WBGene00014453      0      0      0      0
##          L1_GFPminus_rep2 L1_whole_rep3 L1_cells_rep3 L1_GFPplus_rep3
## WBGene00014450      0      0      0      0
## WBGene00014451      0      0      0      0
## WBGene00010957      0      0      0      0

```

```

## WBGene00010958          0          0          0          0
## WBGene00014452          0          0          0          0
## WBGene00014453          0          0          0          0
##           L1_GFPminus_rep3 L3_whole_rep1 L3_cells_rep1 L3_GFPplus_rep1
## WBGene00014450          0          0          0          0
## WBGene00014451          0          0          0          0
## WBGene00010957          0          0          0          0
## WBGene00010958          0          0          0          0
## WBGene00014452          0          0          0          0
## WBGene00014453          0          0          0          0
##           L3_GFPminus_rep1 L3_whole_rep2 L3_cells_rep2 L3_GFPminus_rep2
## WBGene00014450          0          0          0          0
## WBGene00014451          0          0          0          0
## WBGene00010957          0          0          0          0
## WBGene00010958          0          0          0          0
## WBGene00014452          0          0          0          0
## WBGene00014453          0          0          0          0
##           L3_GFPplus_rep2 L3_whole_rep3 L3_cells_rep3 L3_GFPplus_rep3
## WBGene00014450          0          0          0          0
## WBGene00014451          0          0          0          0
## WBGene00010957          0          0          0          0
## WBGene00010958          0          0          0          0
## WBGene00014452          0          0          0          0
## WBGene00014453          0          0          0          0
##           L3_GFPminus_rep3
## WBGene00014450          0
## WBGene00014451          0
## WBGene00010957          0
## WBGene00010958          0
## WBGene00014452          0
## WBGene00014453          0

# Reorganize the metadata table so the names2 column are now headers
rownames(metadata1) <- metadata1$names
coldata <- metadata1[,c("names", "stage", "sample", "rep")]
rownames(coldata) <- as.vector(metadata1$names)
# make grouping variable
coldata$group <- factor(paste0(coldata$stage, coldata$sample))
coldata

##                               names stage sample rep      group
## embryo_cells_rep1    embryo_cells_rep1 embryo   cells rep1  embryocells
## embryo_GFPplus_rep1  embryo_GFPplus_rep1 embryo GFPplus rep1  embryoGFPplus
## embryo_GFPminus_rep1 embryo_GFPminus_rep1 embryo GFPminus rep1  embryoGFPminus
## embryo_whole_rep2    embryo_whole_rep2 embryo   whole rep2  embryowhole
## embryo_cells_rep2    embryo_cells_rep2 embryo   cells rep2  embryocells
## embryo_GFPplus_rep2  embryo_GFPplus_rep2 embryo GFPplus rep2  embryoGFPplus
## embryo_GFPminus_rep2 embryo_GFPminus_rep2 embryo GFPminus rep2  embryoGFPminus
## embryo_whole_rep3    embryo_whole_rep3 embryo   whole rep3  embryowhole
## embryo_GFPplus_rep3  embryo_GFPplus_rep3 embryo GFPplus rep3  embryoGFPplus
## embryo_GFPminus_rep3 embryo_GFPminus_rep3 embryo GFPminus rep3  embryoGFPminus
## L1_whole_rep1         L1_whole_rep1     L1   whole rep1  L1whole
## L1_cells_rep1         L1_cells_rep1     L1   cells rep1  L1cells
## L1_GFPplus_rep1       L1_GFPplus_rep1   L1   GFPplus rep1 L1GFPplus
## L1_GFPminus_rep1       L1_GFPminus_rep1   L1   GFPminus rep1 L1GFPminus

```

```

## L1_whole_rep2      L1_whole_rep2      L1    whole rep2      L1whole
## L1_cells_rep2     L1_cells_rep2      L1    cells rep2      L1cells
## L1_GFPplus_rep2   L1_GFPplus_rep2   L1    GFPplus rep2    L1GFPplus
## L1_GFPminus_rep2  L1_GFPminus_rep2  L1    GFPminus rep2   L1GFPminus
## L1_whole_rep3     L1_whole_rep3     L1    whole rep3     L1whole
## L1_cells_rep3     L1_cells_rep3     L1    cells rep3     L1cells
## L1_GFPplus_rep3   L1_GFPplus_rep3   L1    GFPplus rep3   L1GFPplus
## L1_GFPminus_rep3  L1_GFPminus_rep3  L1    GFPminus rep3  L1GFPminus
## L3_whole_rep1     L3_whole_rep1     L3    whole rep1     L3whole
## L3_cells_rep1     L3_cells_rep1     L3    cells rep1     L3cells
## L3_GFPplus_rep1   L3_GFPplus_rep1   L3    GFPplus rep1   L3GFPplus
## L3_GFPminus_rep1  L3_GFPminus_rep1  L3    GFPminus rep1  L3GFPminus
## L3_whole_rep2     L3_whole_rep2     L3    whole rep2     L3whole
## L3_cells_rep2     L3_cells_rep2     L3    cells rep2     L3cells
## L3_GFPminus_rep2  L3_GFPminus_rep2  L3    GFPminus rep2  L3GFPminus
## L3_GFPplus_rep2   L3_GFPplus_rep2   L3    GFPplus rep2   L3GFPplus
## L3_whole_rep3     L3_whole_rep3     L3    whole rep3     L3whole
## L3_cells_rep3     L3_cells_rep3     L3    cells rep3     L3cells
## L3_GFPplus_rep3   L3_GFPplus_rep3   L3    GFPplus rep3   L3GFPplus
## L3_GFPminus_rep3  L3_GFPminus_rep3  L3    GFPminus rep3  L3GFPminus

# Check that the names match --> Should be TRUE
all(rownames(coldata) == colnames(cts))

## [1] TRUE

```

## Make DESeqDataSet

Generate the DESeqDataSet. The variables in this design formula will be the type of sample, and the preparation date. This should reduce the variability between the samples based on when they were made.

From the vignette: “In order to benefit from the default settings of the package, you should put the variable of interest at the end of the formula and make sure the control level is the first level.”

The variable of interest is the sample type.

Using `DESeqDataSetFromMatrix` since I used the program `featureCounts`.

```

dds <- DESeqDataSetFromMatrix(countData = cts,
                               colData = coldata,
                               design = ~ group)

```

## Filter genes with sum counts per million $\geq 10$ across all samples

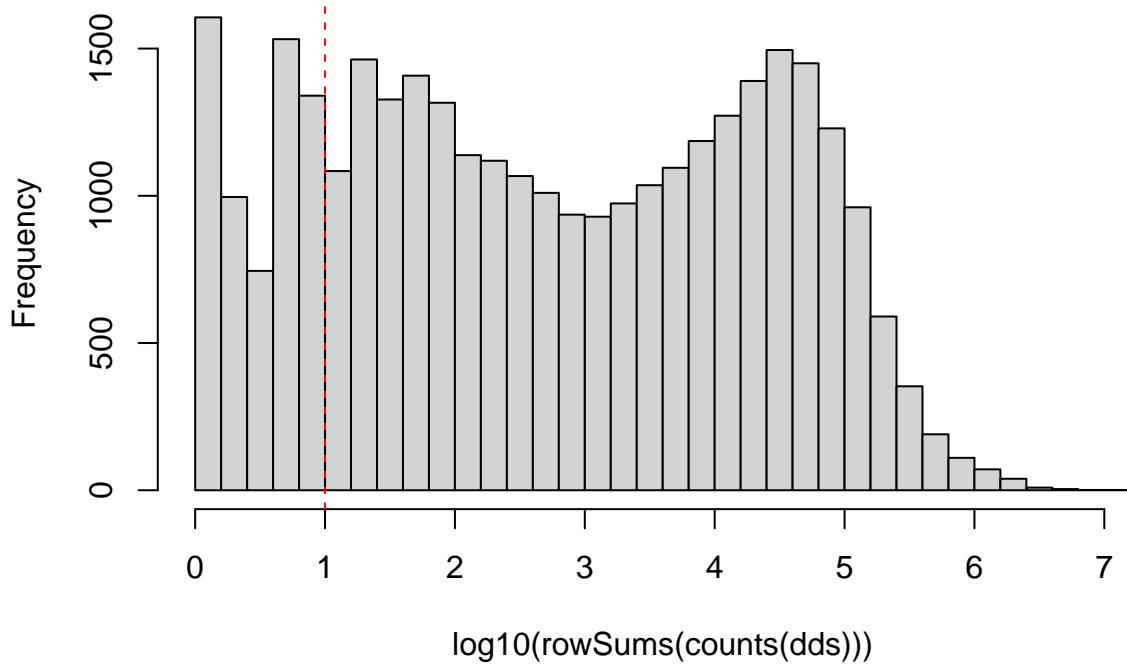
Visualize read count distribution

```

raw_count_threshold <- 10
hist(log10(rowSums(counts(dds))), breaks = 50)
abline(v = log10(raw_count_threshold), col = "red", lty = 2)

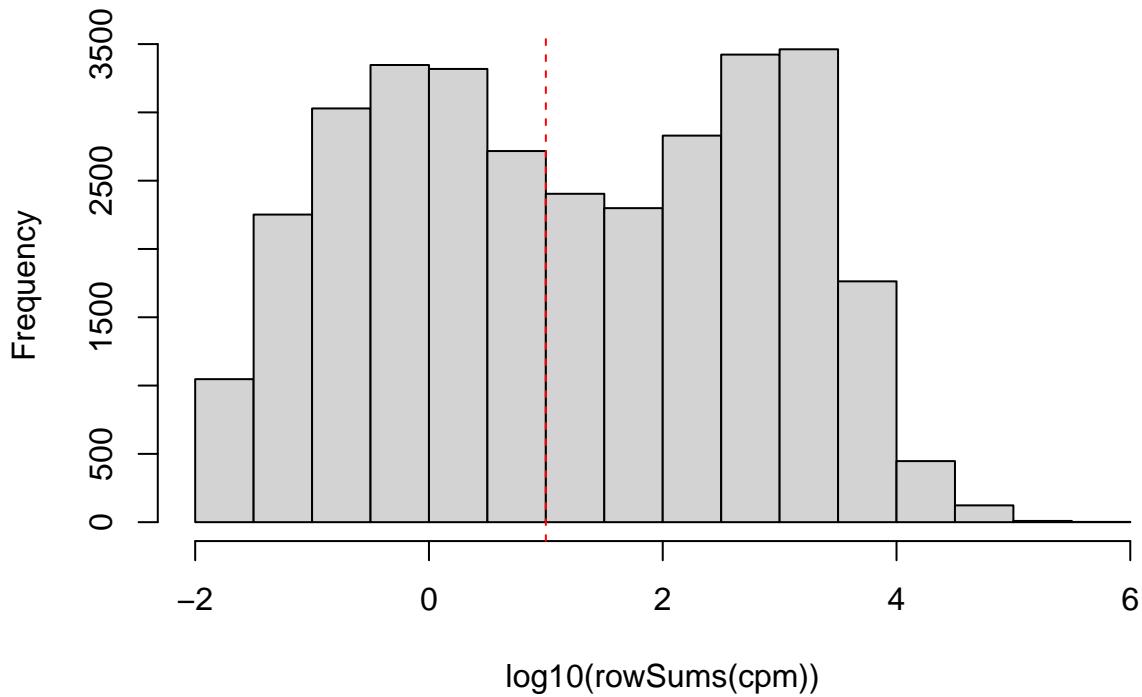
```

### Histogram of $\log_{10}(\text{rowSums}(\text{counts}(dd)))$



```
cpm <- apply(counts(dd), 2, function(x) (x/sum(x))*1000000)
hist(log10(rowSums(cpm)))
abline(v = log10(raw_count_threshold), col = "red", lty = 2)
```

### Histogram of $\log_{10}(\text{rowSums}(cpm))$



Filter to remove genes with low read counts

```

keep <- rowSums(cpm) >= raw_count_threshold
dds <- dds[keep,]
dds

## class: DESeqDataSet
## dim: 16762 34
## metadata(1): version
## assays(1): counts
## rownames(16762): WBGene00021406 WBGene00021407 ... WBGene00199694
##   WBGene00044951
## rowData names(0):
## colnames(34): embryo_cells_rep1 embryo_GFPplus_rep1 ... L3_GFPplus_rep3
##   L3_GFPminus_rep3
## colData names(5): names stage sample rep group

Filter to select for protein-coding genes

transcript_type <- read_csv(file = "../01_input/biomaRt_elegans_transcript_biotype.csv")

## Rows: 59897 Columns: 4
## -- Column specification -----
## Delimiter: ","
## chr (4): Gene stable ID, Genome project, Gene name, Transcript biotype
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

colnames(transcript_type) <- c("WBGeneID", "genome_id", "gene_name", "biotype")
dds <- dds[rownames(dds) %in%
  (transcript_type %>%
    filter(biotype == "protein_coding") %>%
    pull(WBGeneID)),]

```

Perform Differential Expression

```

dds <- DESeq(dds)

## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing

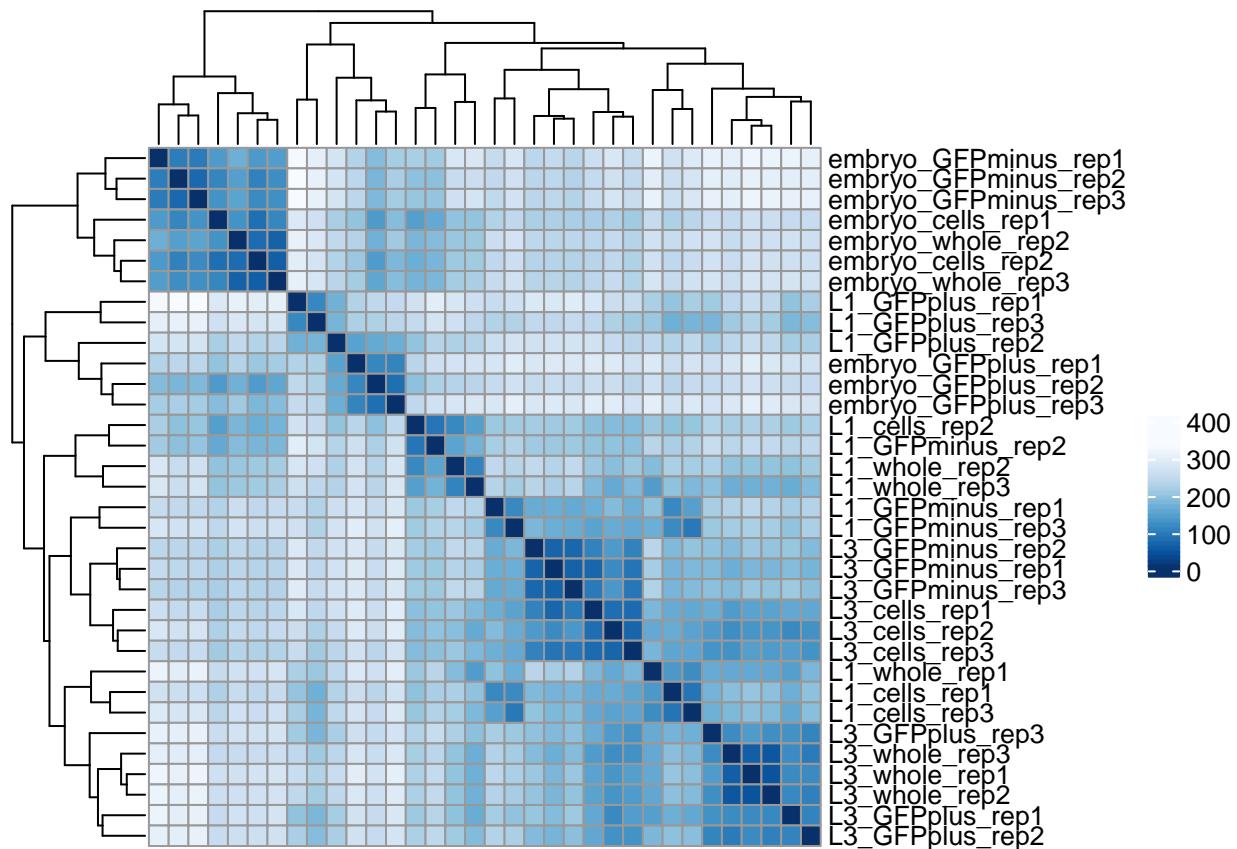
resultsNames(dds)

## [1] "Intercept"                               "group_embryoGFPminus_vs_embryocells"
## [3] "group_embryoGFPplus_vs_embryocells"      "group_embryowhole_vs_embryocells"
## [5] "group_L1cells_vs_embryocells"              "group_L1GFPminus_vs_embryocells"
## [7] "group_L1GFPplus_vs_embryocells"             "group_L1whole_vs_embryocells"
## [9] "group_L3cells_vs_embryocells"               "group_L3GFPminus_vs_embryocells"
## [11] "group_L3GFPplus_vs_embryocells"             "group_L3whole_vs_embryocells"

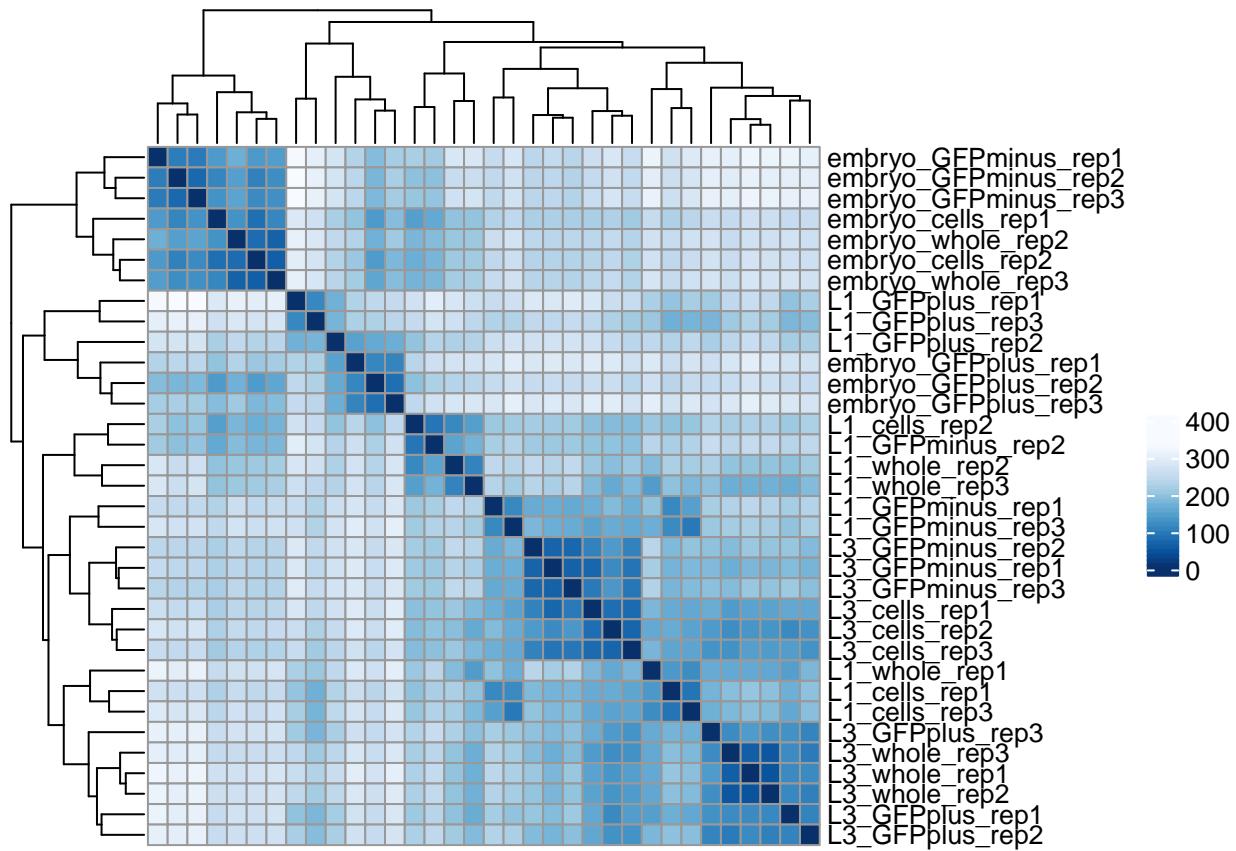
```

## Sample-to-sample distance matrix

```
vsd <- vst(dds, blind = FALSE)
sampleDists <- dist(t(assay(vsd)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- vsd$names
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
mega_cor_plot <- pheatmap(sampleDistMatrix,
                           clustering_distance_rows = sampleDists,
                           clustering_distance_cols = sampleDists,
                           col = colors)
```



```
mega_cor_plot
```

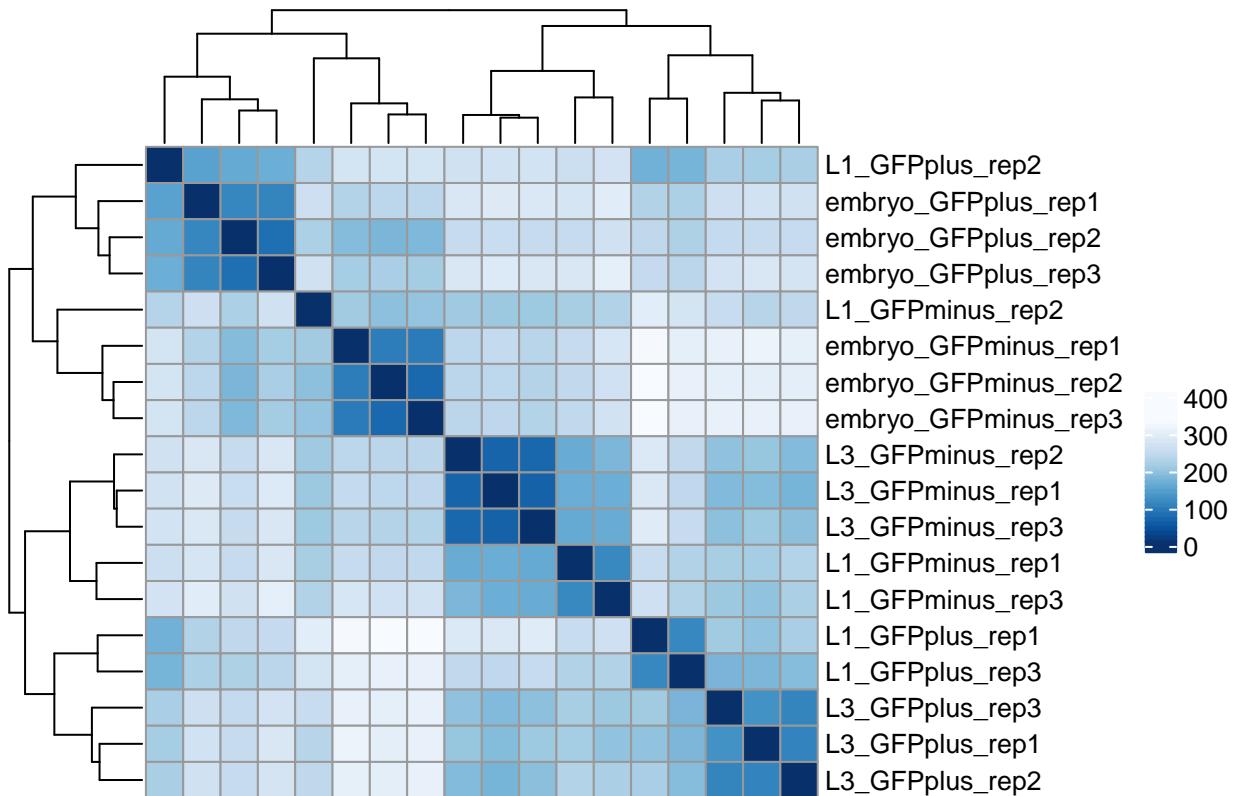


```
myPDFplot(plot = mega_cor_plot, name = "FACS_Correlation_All_Samples", height = 4.5, width = 6, plotdir
## pdf
## 2
```

**Figure S2**

```
all_sorted_samples_cor <- vsd.corr.per.stage("GFPplus|GFPminus", "Correlation of FACS isolated GFP+ and
```

## Correlation of FACS isolated GFP+ and GFP- samples



```
myPDFplot(plot = all_sorted_samples_cor, name = "All_Stage_FACS_Correlation_Sorted_Samples", height = 400)
## pdf
## 2
```

## Remove L1 rep 2

```
remove_samples <- c("L1_whole_rep2", "L1_cells_rep2", "L1_GFPplus_rep2", "L1_GFPminus_rep2")
coldata <- coldata %>% filter(!names %in% remove_samples)
dds <- dds[, !colnames(dds) %in% remove_samples]
```

## Perform Differential Expression

```
dds <- DESeq(dds)

## using pre-existing size factors
## estimating dispersions
## found already estimated dispersions, replacing these
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

```

resultsNames(dds)

## [1] "Intercept"                               "group_embryoGFPminus_vs_embryocells"
## [3] "group_embryoGFPplus_vs_embryocells"      "group_embryowhole_vs_embryocells"
## [5] "group_L1cells_vs_embryocells"              "group_L1GFPminus_vs_embryocells"
## [7] "group_L1GFPplus_vs_embryocells"             "group_L1whole_vs_embryocells"
## [9] "group_L3cells_vs_embryocells"                "group_L3GFPminus_vs_embryocells"
## [11] "group_L3GFPplus_vs_embryocells"              "group_L3whole_vs_embryocells"

```

## Counts table output for supplemental datatable

```

# raw counts
write.table(as.data.frame(counts(dds, normalized = FALSE)) %>% rownames_to_column(var = "WBGeneID"),
            file = "../03_output/count_tables_for_sup/intestine_FACS_RNAseq_raw_counts.txt",
            sep = "\t", quote = FALSE, row.names = FALSE)

# normalized counts
write.table(as.data.frame(counts(dds, normalized = TRUE)) %>% rownames_to_column(var = "WBGeneID"),
            file = "../03_output/count_tables_for_sup/intestine_FACS_RNAseq_norm_counts.txt",
            sep = "\t", quote = FALSE, row.names = FALSE)

# rlog transformed counts
write.table(as.data.frame(assay(rlog(dds, blind=FALSE))) %>% rownames_to_column(var = "WBGeneID"),
            file = "../03_output/count_tables_for_sup/intestine_FACS_RNAseq_rlog_counts.txt",
            sep = "\t", quote = FALSE, row.names = FALSE)

## rlog() may take a few minutes with 30 or more samples,
## vst() is a much faster transformation

```

## Within-stage pairwise comparisons

Set cutoff values

```

thresh = 1
sig = 0.01

```

## Embryo stage pairwise comparisons

```

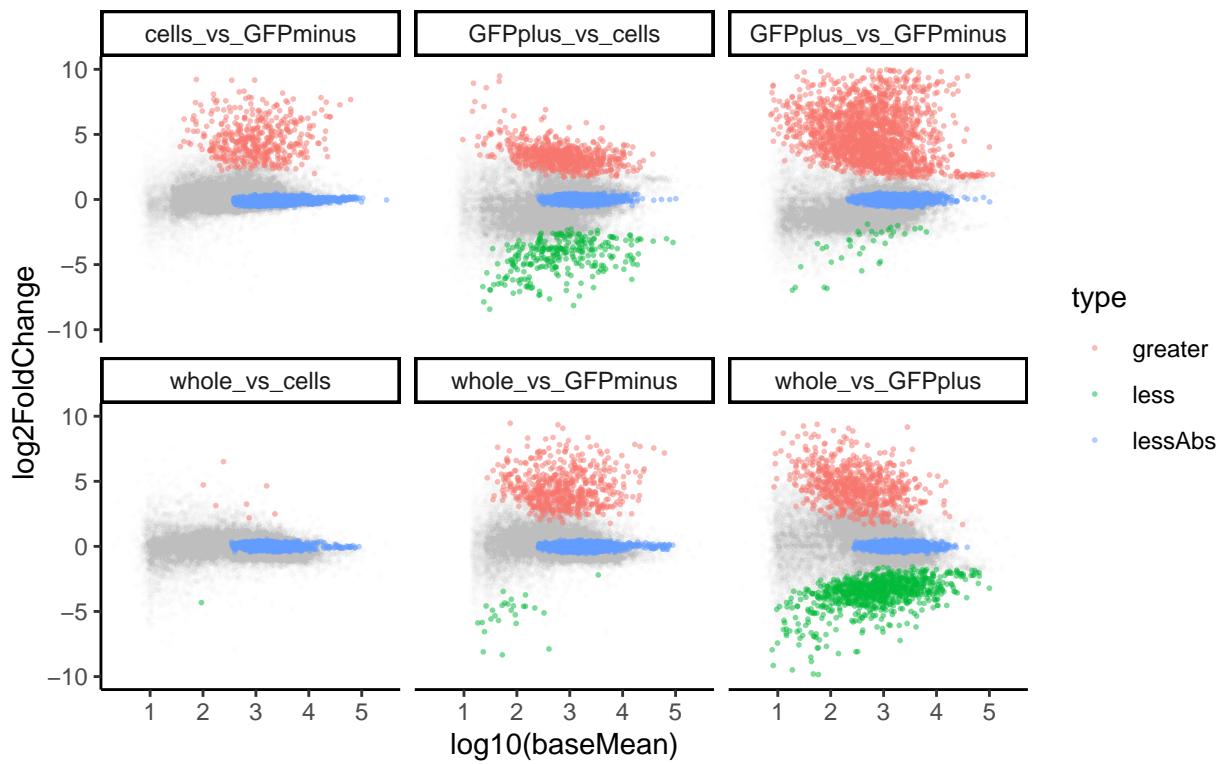
embryo_alt_hyp_res_df <- alt_hyp_res_df("embryo", thresh = thresh, sig = sig)

de_category_MA_plot(embryo_alt_hyp_res_df, paste("Embryo differentially expressed genes\nlfc = ", thresh))

## Warning: Removed 91 rows containing missing values (geom_point).
## Warning: Removed 32 rows containing missing values (geom_point).

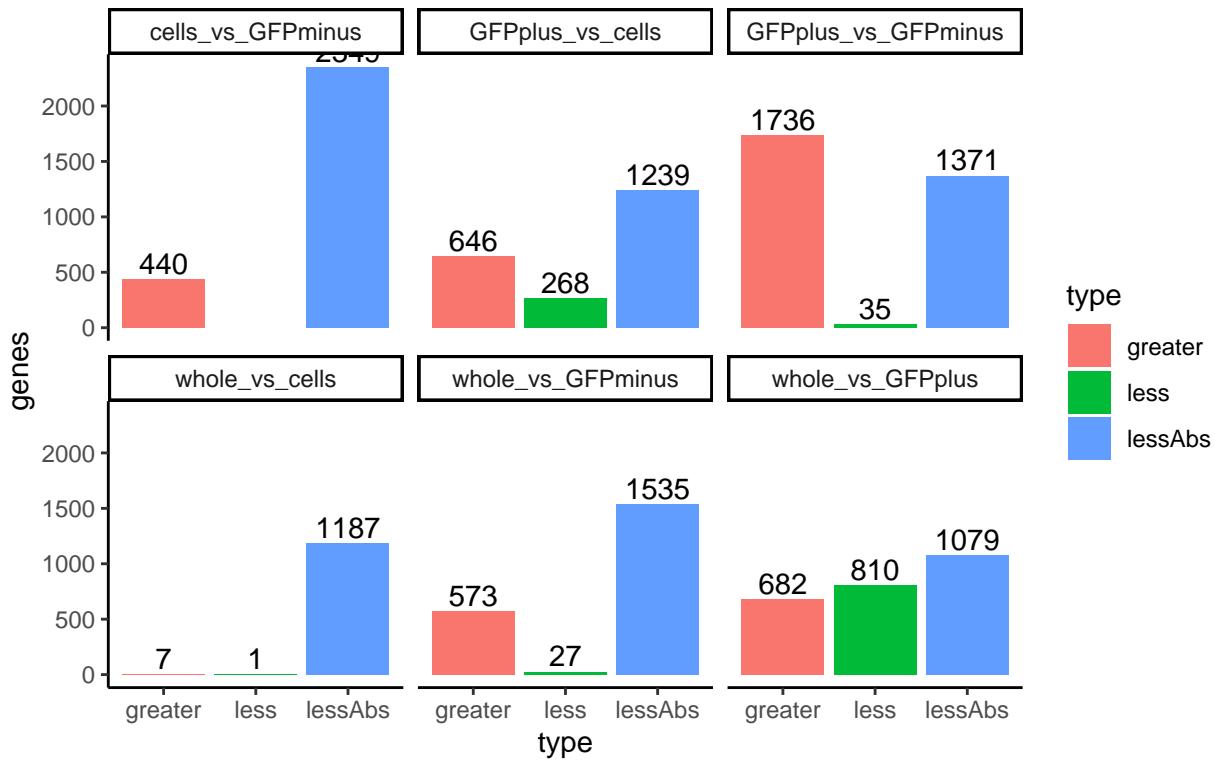
```

Embryo differentially expressed genes  
lfc = 1 & padj < 0.01



```
de_category_bar_plot(embryo_alt_hyp_res_df, paste("Embryo differentially expressed genes\nlfc = ",threshold))
## `summarise()` has grouped output by 'label'. You can override using the
## `.`groups` argument.
```

Embryo differentially expressed genes  
lfc = 1 & padj < 0.01



```
## L1 stage pairwise comparisons
```

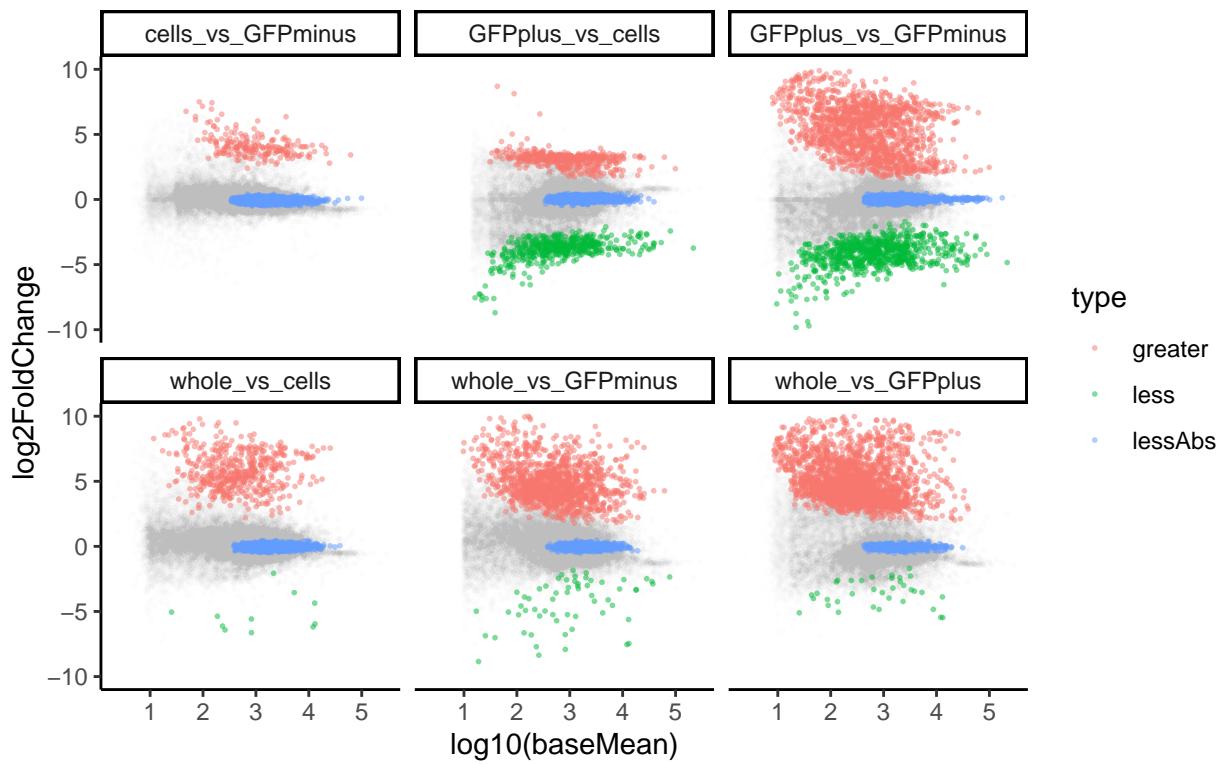
```
L1_alt_hyp_res_df<- alt_hyp_res_df("L1", thresh = thresh, sig = sig)
```

```
de_category_MA_plot(L1_alt_hyp_res_df, paste("L1 differentially expressed genes\nlfc = ",thresh," & padj <
```

```
## Warning: Removed 64 rows containing missing values (geom_point).
```

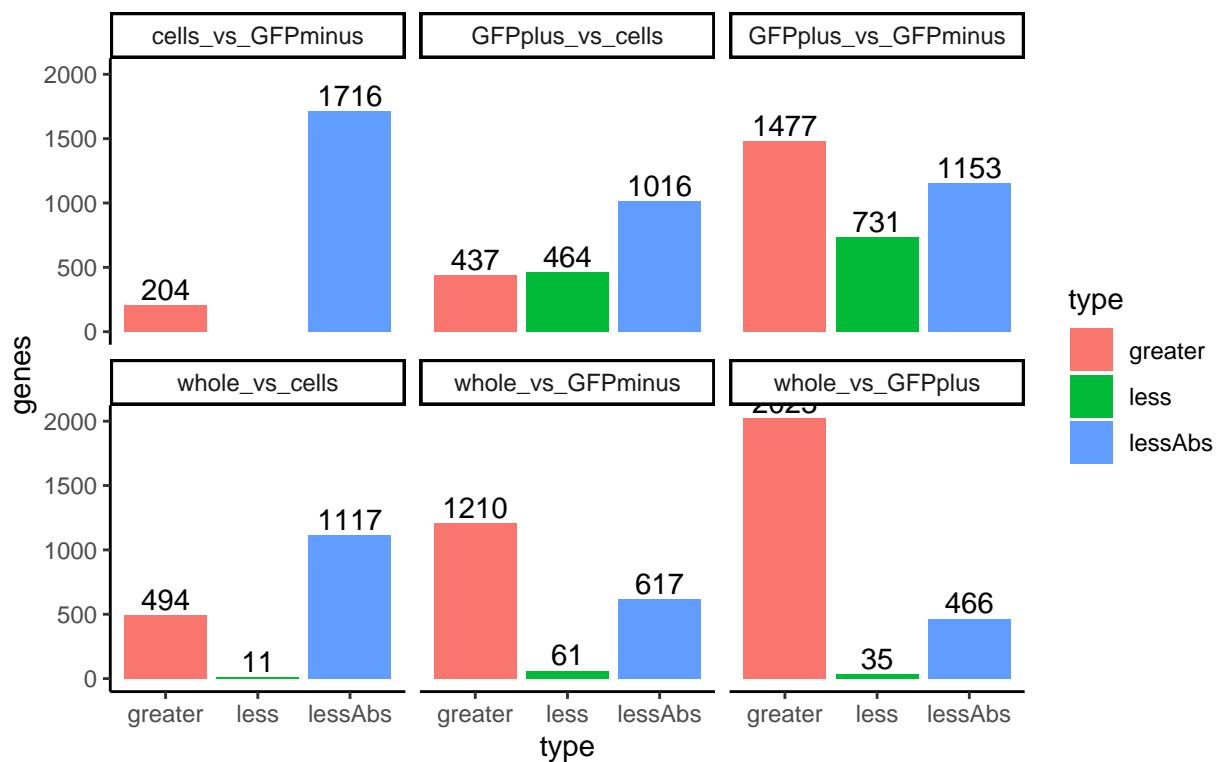
```
## Warning: Removed 31 rows containing missing values (geom_point).
```

L1 differentially expressed genes  
lfc = 1 & padj < 0.01



```
de_category_bar_plot(L1_alt_hyp_res_df, paste("L1 differentially expressed genes\nlfc = ", thresh, " & padj < 0.01"))
## `summarise()` has grouped output by 'label'. You can override using the
## `.` argument.
```

L1 differentially expressed genes  
lfc = 1 & padj < 0.01

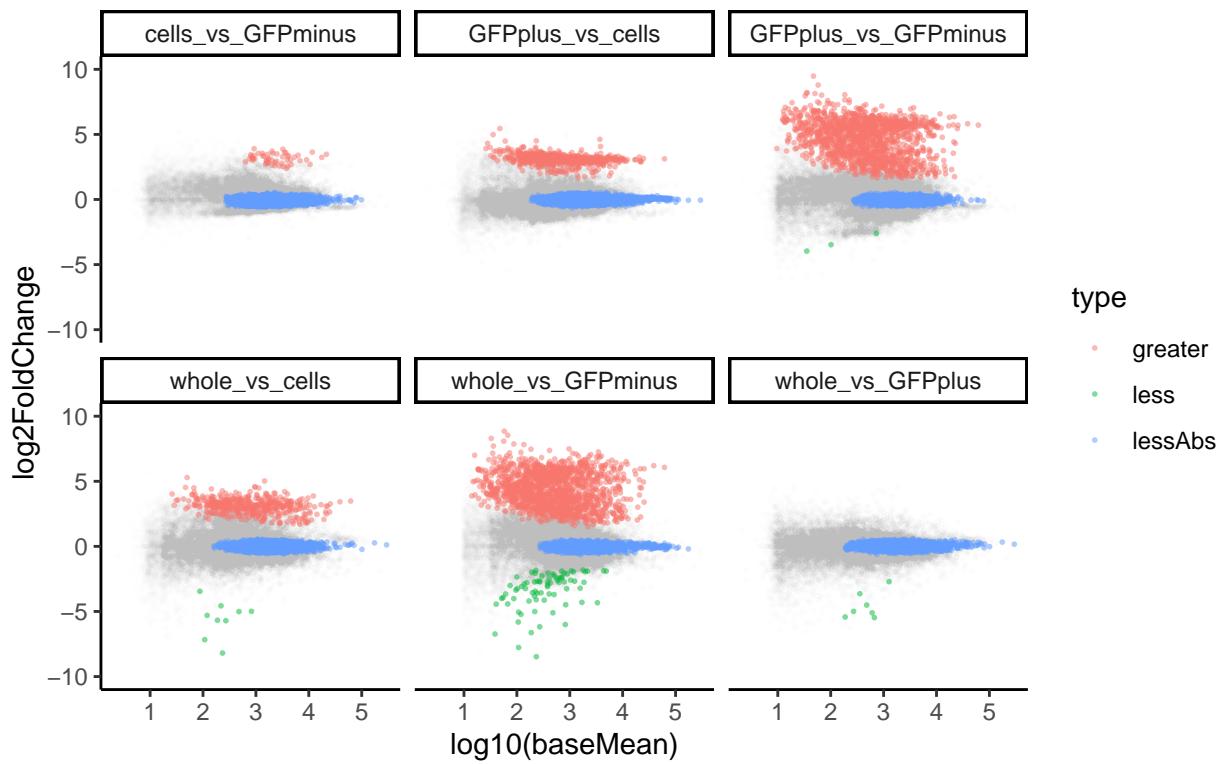


## L3 stage pairwise comparisons

```
L3_alt_hyp_res_df<- alt_hyp_res_df("L3", thresh = thresh, sig = sig)
```

```
de_category_MA_plot(L3_alt_hyp_res_df, paste("L3 differentially expressed genes\nlfc = ",thresh," & padj <
```

L3 differentially expressed genes  
lfc = 1 & padj < 0.01



```
de_category_bar_plot(L3_alt_hyp_res_df, paste("L3 differentially expressed genes\nlfc = ", thresh, " & padj < 0.01"))
## `summarise()` has grouped output by 'label'. You can override using the
## `.groups` argument.
```

L3 differentially expressed genes  
lfc = 1 & padj < 0.01

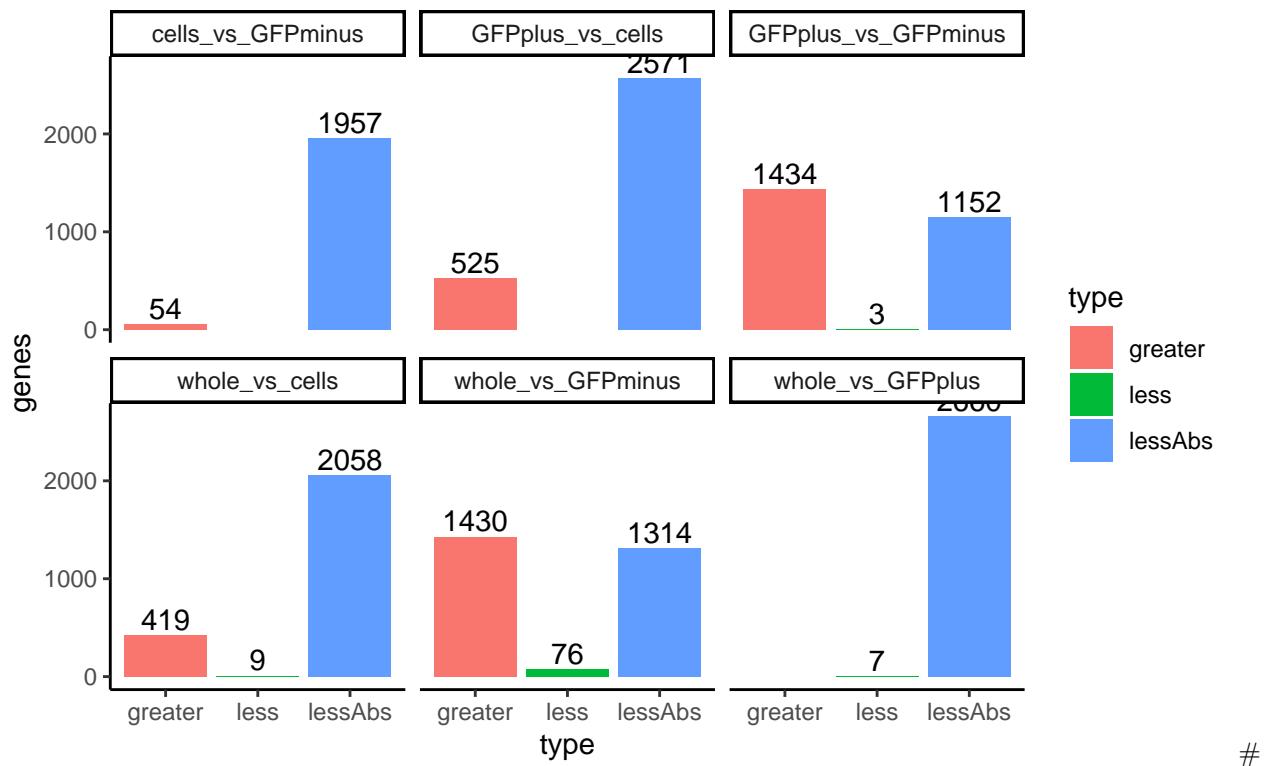


Figure S3

Whole worm vs. dissociated cells analysis

```
cells_vs_whole_bar_df <- L1_alt_hyp_res_df %>% filter(isDE == TRUE, label == "whole_vs_cells") %>% group_by(type) %>% bind_rows(
  L3_alt_hyp_res_df %>% filter(isDE == TRUE, label == "whole_vs_cells") %>% group_by(type) %>% summarise(n = sum(genes))
) %>% bind_rows(
  embryo_alt_hyp_res_df %>% filter(isDE == TRUE, label == "whole_vs_cells") %>% group_by(type) %>% summarise(n = sum(genes))
)
cells_vs_whole_bar_df

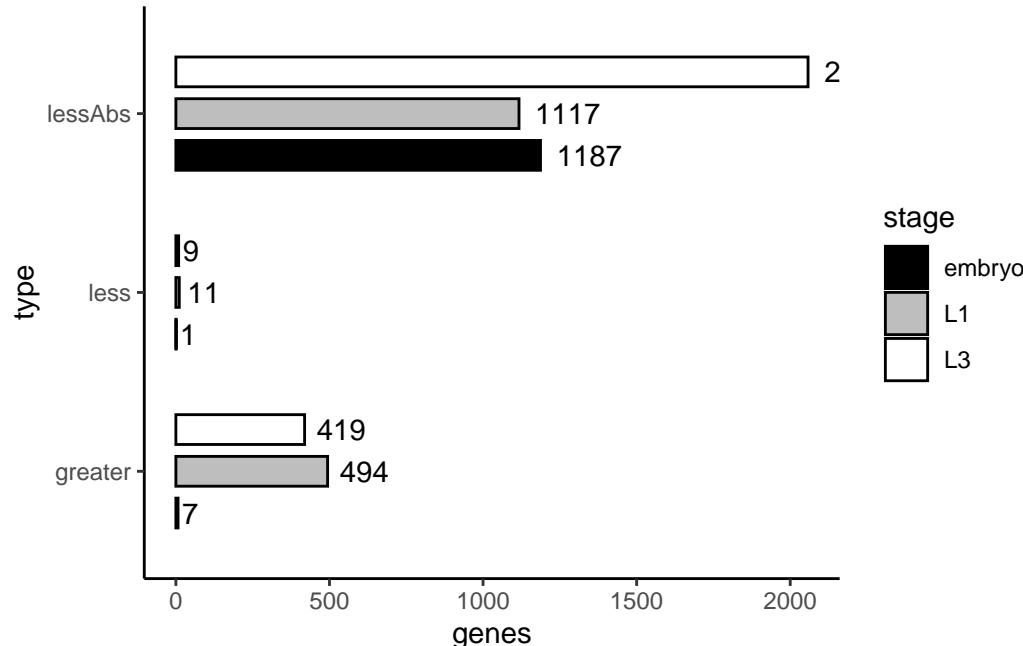
## # A tibble: 9 x 3
##   type     genes stage
##   <chr>    <int> <chr>
## 1 greater     494 L1
## 2 less        11  L1
## 3 lessAbs   1117 L1
## 4 greater     419 L3
## 5 less         9  L3
## 6 lessAbs   2058 L3
## 7 greater      7  embryo
## 8 less         1  embryo
## 9 lessAbs  1187 embryo

cells_vs_whole_bar_plot <- cells_vs_whole_bar_df %>%
  ggplot(aes(x = type, y = genes, fill = stage, label = genes)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.7), width = 0.5, color = "black") +
```

```

geom_text(hjust = -0.25, position = position_dodge(width = 0.7)) +
scale_fill_manual(values = c("black", "grey", "white")) +
theme_classic() +
coord_flip()
cells_vs_whole_bar_plot

```



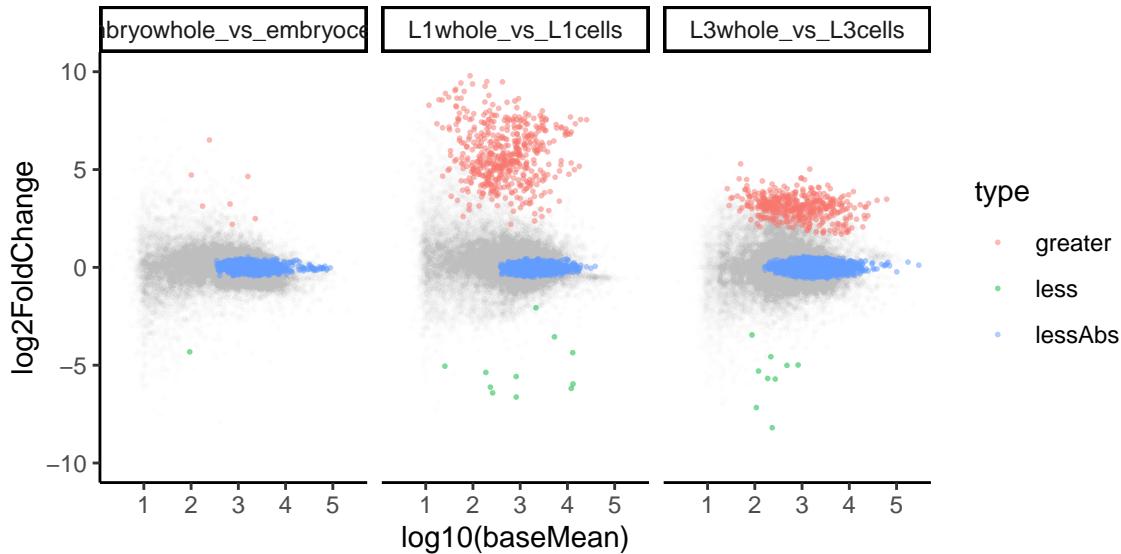
```

ggsave(plot = cells_vs_whole_bar_plot, filename = ".../03_output/plots/cells_vs_whole/cells_vs_whole_bar"

cells_vs_whole_MA_plot <- de_category_MA_plot(
L1_alt_hyp_res_df %>% filter(label == "whole_vs_cells") %>% mutate(label = comparison) %>%
bind_rows(
  L3_alt_hyp_res_df %>% filter(label == "whole_vs_cells") %>% mutate(label = comparison)
) %>%
bind_rows(
  embryo_alt_hyp_res_df %>% filter(label == "whole_vs_cells") %>% mutate(label = comparison)
),
title = NULL
)
cells_vs_whole_MA_plot

## Warning: Removed 10 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_point).

```



```
ggsave(plot = cells_vs_whole_MA_plot, filename = "../03_output/plots/cells_vs_whole/cells_vs_whole_MA_p...  
## Warning: Removed 10 rows containing missing values (geom_point).  
## Removed 2 rows containing missing values (geom_point).
```

## DESeq2 builtin plotting function

```
res_embryoGFPplus_vs_embryoGFPminus <- results(dds, contrast = c("group", "embryoGFPplus", "embryoGFPminus"))  
res_L1GFPplus_vs_L1_GFPminus <- results(dds, contrast = c("group", "L1GFPplus", "L1GFPminus"))  
res_L3GFPplus_vs_L3_GFPminus <- results(dds, contrast = c("group", "L3GFPplus", "L3GFPminus"))  
  
res_embryoGFPplus_vs_embryoGFPminus_ashr <- lfcShrink(dds, contrast = c("group", "embryoGFPplus", "embryoGFPminus"))  
  
## using 'ashr' for LFC shrinkage. If used in published research, please cite:  
## Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.  
## https://doi.org/10.1093/biostatistics/kxw041  
  
res_L1GFPplus_vs_L1GFPminus_ashr <- lfcShrink(dds, contrast = c("group", "L1GFPplus", "L1GFPminus"), type = "ashr")  
  
## using 'ashr' for LFC shrinkage. If used in published research, please cite:  
## Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.  
## https://doi.org/10.1093/biostatistics/kxw041  
  
res_L3GFPplus_vs_L3GFPminus_ashr <- lfcShrink(dds, contrast = c("group", "L3GFPplus", "L3GFPminus"), type = "ashr")  
  
## using 'ashr' for LFC shrinkage. If used in published research, please cite:  
## Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.  
## https://doi.org/10.1093/biostatistics/kxw041
```

## Export rlog counts

```
all_samples_rld <- rlog(dds)  
  
## rlog() may take a few minutes with 30 or more samples,  
## vst() is a much faster transformation
```

```

write_rds(all_samples_rld, file = "../03_output/rlog_counts/all_samples_rlog_counts.rds")
write_tsv(as.data.frame(assay(all_samples_rld)) %>% rownames_to_column(var = "WBGeneID"), file = "../03_
write_tsv(as.data.frame(assay(all_samples_rld)) %>% rownames_to_column(var = "WBGeneID") %>% select(WBGeneID)
all_samples_rld <- read_rds(file = "../03_output/rlog_counts/all_samples_rlog_counts.rds")
all_samples_rld_df <- read_tsv(file = "../03_output/rlog_counts/all_samples_rlog_counts.tsv")

## Rows: 15627 Columns: 31
## -- Column specification -----
## Delimiter: "\t"
## chr (1): WBGeneID
## dbl (30): embryo_cells_rep1, embryo_GFPplus_rep1, embryo_GFPminus_rep1, embr...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
head(all_samples_rld_df)

## # A tibble: 6 x 31
##   WBGeneID   embryo_cells_re~ embryo_GFPplus_~ embryo_GFPminus_~ embryo_whole_re~
##   <chr>       <dbl>        <dbl>        <dbl>        <dbl>
## 1 WBGene000~    6.21        6.19        6.70        5.52
## 2 WBGene000~    1.94        3.43        2.92        2.99
## 3 WBGene000~    3.63        3.35        0.907       1.30
## 4 WBGene000~    1.15        3.27        0.760       1.95
## 5 WBGene000~    9.82       10.1        9.84        8.95
## 6 WBGene000~    1.59        1.06        1.48        2.14
## # ... with 26 more variables: embryo_cells_rep2 <dbl>,
## #   embryo_GFPplus_rep2 <dbl>, embryo_GFPminus_rep2 <dbl>,
## #   embryo_whole_rep3 <dbl>, embryo_GFPplus_rep3 <dbl>,
## #   embryo_GFPminus_rep3 <dbl>, L1_whole_rep1 <dbl>, L1_cells_rep1 <dbl>,
## #   L1_GFPplus_rep1 <dbl>, L1_GFPminus_rep1 <dbl>, L1_whole_rep3 <dbl>,
## #   L1_cells_rep3 <dbl>, L1_GFPplus_rep3 <dbl>, L1_GFPminus_rep3 <dbl>,
## #   L3_whole_rep1 <dbl>, L3_cells_rep1 <dbl>, L3_GFPplus_rep1 <dbl>, ...

```

## Export intestine enrichment assignment

```

embryo_intestine_gene_cats <- embryo_alt_hyp_res_df %>%
  drop_na(padj) %>%
  filter(label == "GFPplus_vs_GFPminus", padj < sig) %>%
  select(WBGeneID, altHyp = "type") %>%
  mutate(intestine_expression = case_when(
    altHyp == "greater" ~ "enriched",
    altHyp == "less" ~ "depleted",
    altHyp == "lessAbs" ~ "equal")) %>%
  mutate(intestine_expression = fct_relevel(intestine_expression, c("enriched", "equal", "depleted")))

L1_intestine_gene_cats <- L1_alt_hyp_res_df %>%
  filter(label == "GFPplus_vs_GFPminus", padj < sig) %>%
  select(WBGeneID, altHyp = "type") %>%
  mutate(intestine_expression = case_when(
    altHyp == "greater" ~ "enriched",
    altHyp == "less" ~ "depleted",
    altHyp == "lessAbs" ~ "equal"))

```

```

  altHyp == "lessAbs" ~ "equal"))%>%
  mutate(intestine_expression = fct_relevel(intestine_expression, c("enriched", "equal", "depleted")))

L3_intestine_gene_cats <- L3_alt_hyp_res_df %>%
  filter(label == "GFPplus_vs_GFPminus", padj < sig) %>%
  select(WBGeneID, altHyp = "type") %>%
  mutate(intestine_expression = case_when(
    altHyp == "greater" ~ "enriched",
    altHyp == "less" ~ "depleted",
    altHyp == "lessAbs" ~ "equal"))%>%
  mutate(intestine_expression = fct_relevel(intestine_expression, c("enriched", "equal", "depleted")))

write_csv(x = embryo_intestine_gene_cats, file = "../03_output/intestine_gene_categories/embryo_intestine_gene_cats.csv")

write_csv(x = L1_intestine_gene_cats, file = "../03_output/intestine_gene_categories/L1_intestine_gene_cats.csv")

write_csv(x = L3_intestine_gene_cats, file = "../03_output/intestine_gene_categories/L3_intestine_gene_cats.csv")

```

## Average embryo GFP+ sample reads

Make function

```

thresh = 1
sig = 0.01

rlog_status <- function(stage, res, hyp_df){
  all_samples_rld_df %>%
    select(WBGeneID, contains(paste(stage, "GFPplus", sep = "_")))) %>%
    pivot_longer(cols = contains("GFPplus"), values_to = "rlog_counts") %>%
    separate(name, sep = "_", into = c("stage", "sample", "rep")) %>%
    group_by(WBGeneID) %>%
    summarise(mean.rlog.counts = mean(rlog_counts), var.rlog.counts = var(rlog_counts)) %>%
    left_join(hyp_df %>% filter(label == "GFPplus_vs_GFPminus")) %>% select(WBGeneID, type, isDE), by = "WBGeneID"
}

embryo_rlog_status_df <- rlog_status(stage = "embryo", res = res_embryoGFPplus_vs_embryoGFPminus, hyp_df = hyp_df)
head(embryo_rlog_status_df)

```

```

## # A tibble: 6 x 5
##   WBGeneID      mean.rlog.counts var.rlog.counts type     isDE
##   <chr>              <dbl>        <dbl> <chr>    <lgl>
## 1 WBGene00000001      10.1        0.00518 greater FALSE
## 2 WBGene00000001      10.1        0.00518 less   FALSE
## 3 WBGene00000001      10.1        0.00518 lessAbs FALSE
## 4 WBGene00000002      10.9        0.135   greater FALSE
## 5 WBGene00000002      10.9        0.135   less   FALSE
## 6 WBGene00000002      10.9        0.135   lessAbs FALSE

write_csv(embryo_rlog_status_df, file = "../03_output/embryo_GFPplus_rlog_counts_status_df.csv", col_names = colnames(embryo_rlog_status_df))
embryo_rlog_status_df <- read_csv(file = "../03_output/embryo_GFPplus_rlog_counts_status_df.csv", col_names = colnames(embryo_rlog_status_df))

## Rows: 40203 Columns: 5
## -- Column specification -----
## Delimiter: ","

```

```

## chr (2): WBGeneID, type
## dbl (2): mean.rlog.counts, var.rlog.counts
## lgl (1): isDE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
L1_rlog_status_df <- rlog_status(stage = "L1", res = res_L1GFPplus_vs_L1_GFPminus, hyp_df = L1_alt_hyp_1)
write_csv(L1_rlog_status_df, file = "../03_output/L1_GFPplus_rlog_counts_status_df.csv", col_names = TRUE)
L1_rlog_status_df <- read_csv(file = "../03_output/L1_GFPplus_rlog_counts_status_df.csv", col_names = TRUE)

## Rows: 38690 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr (2): WBGeneID, type
## dbl (2): mean.rlog.counts, var.rlog.counts
## lgl (1): isDE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
head(L1_rlog_status_df)

## # A tibble: 6 x 5
##   WBGeneID      mean.rlog.counts var.rlog.counts type    isDE
##   <chr>          <dbl>           <dbl> <chr>    <lgl>
## 1 WBGene00000001  10.1            0.0190 greater  FALSE
## 2 WBGene00000001  10.1            0.0190 less    FALSE
## 3 WBGene00000001  10.1            0.0190 lessAbs TRUE
## 4 WBGene00000002  8.34           0.142  greater FALSE
## 5 WBGene00000002  8.34           0.142  less    FALSE
## 6 WBGene00000002  8.34           0.142  lessAbs FALSE

L3_rlog_status_df <- rlog_status(stage = "L3", res = res_L3GFPplus_vs_L3_GFPminus, hyp_df = L3_alt_hyp_1)
write_csv(L3_rlog_status_df, file = "../03_output/L3_GFPplus_rlog_counts_status_df.csv", col_names = TRUE)
L3_rlog_status_df <- read_csv(file = "../03_output/L3_GFPplus_rlog_counts_status_df.csv", col_names = TRUE)

## Rows: 39901 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr (2): WBGeneID, type
## dbl (2): mean.rlog.counts, var.rlog.counts
## lgl (1): isDE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
head(L3_rlog_status_df)

## # A tibble: 6 x 5
##   WBGeneID      mean.rlog.counts var.rlog.counts type    isDE
##   <chr>          <dbl>           <dbl> <chr>    <lgl>
## 1 WBGene00000001  10.1            0.0919 greater FALSE
## 2 WBGene00000001  10.1            0.0919 less    FALSE
## 3 WBGene00000001  10.1            0.0919 lessAbs TRUE
## 4 WBGene00000002  8.99           0.352  greater FALSE
## 5 WBGene00000002  8.99           0.352  less    FALSE

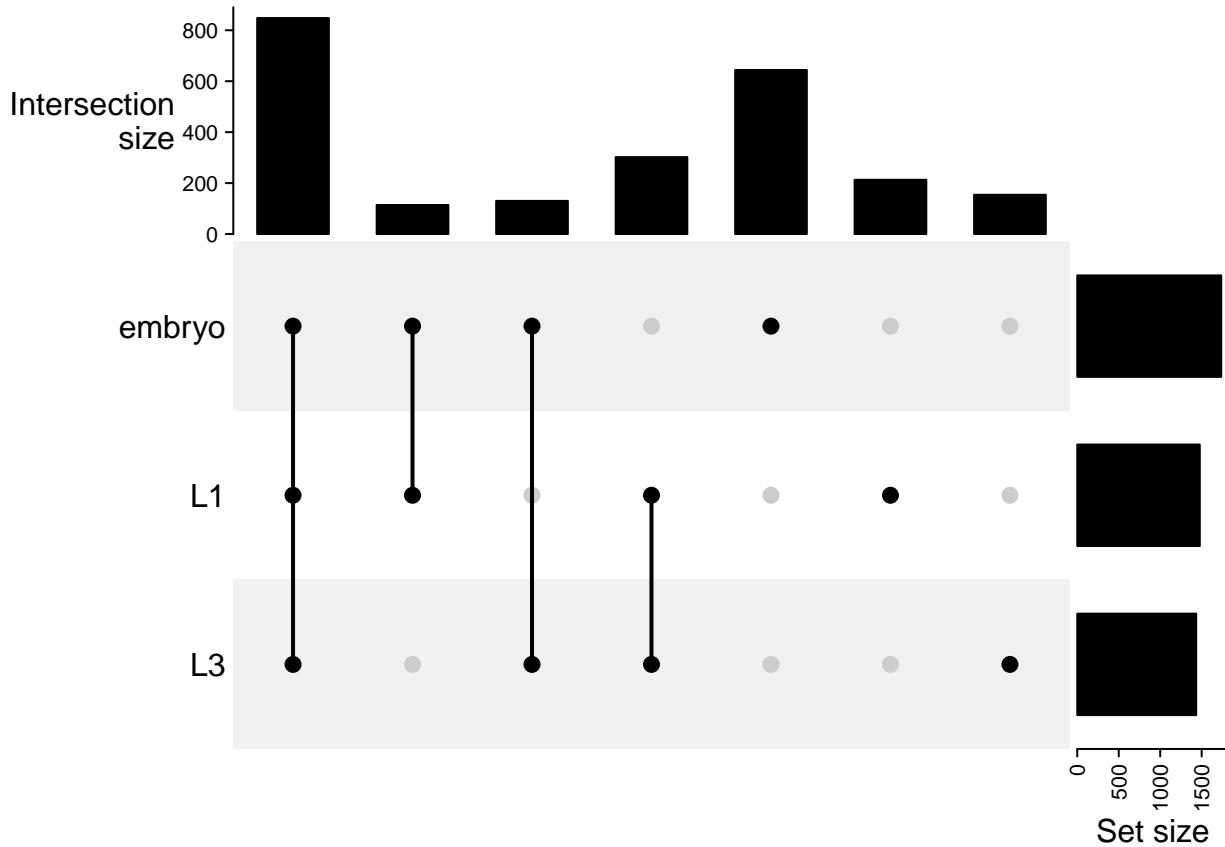
```

```
## 6 WBGene00000002 8.99 0.352 lessAbs FALSE
```

## Intesinte expression across development

### UpSet Plot

```
intestine_enriched_genes <- data.frame(embryo_rlog_status_df, stage = "embryo") %>% bind_rows(data.frame(stage_list<- list(embryo = filter(intestine_enriched_genes, stage == "embryo")$WBGeneID, L1 = filter(intestine_enriched_genes, stage == "L1")$WBGeneID, L3 = filter(intestine_enriched_genes, stage == "L3")$WBGeneID) comb_mat <- make_comb_mat(stage_list) UpSet(comb_mat)
```



```
comb_size(comb_mat)
```

```
## 111 110 101 011 100 010 001  
## 848 114 130 302 644 213 154
```

## Data output

```
# per stage GFP+ vs GFP- differential expression  
write_csv(res_to_df(res_embryoGFPplus_vs_embryoGFPminus), file = ".../03_output/pairwise_DE_results/res_...  
write_csv(res_to_df(res_L1GFPplus_vs_L1_GFPminus), file = ".../03_output/pairwise_DE_results/res_L1GFPpl...  
write_csv(res_to_df(res_L3GFPplus_vs_L3_GFPminus), file = ".../03_output/pairwise_DE_results/res_L3GFPpl...
```

```

# per stage GFP+ vs GFP- differential expression with log2FC shrink (visualization, ranking)
write_csv(res_to_df(res_embryoGFPplus_vs_embryoGFPminus_ashr), file = "../03_output/pairwise_shrunk_DE_results/L1.csv")
write_csv(res_to_df(res_L1GFPplus_vs_L1GFPminus_ashr), file = "../03_output/pairwise_shrunk_DE_results/L1.csv")
write_csv(res_to_df(res_L3GFPplus_vs_L3GFPminus_ashr), file = "../03_output/pairwise_shrunk_DE_results/L3.csv")

# per stage enrichment annotation
write_csv(embryo_alt_hyp_res_df %>% filter(label == "GFPplus_vs_GFPminus") %>% select(WBGeneID:type, isDE),
          file = "L1_enrichment_annotation.csv")
write_csv(L1_alt_hyp_res_df %>% filter(label == "GFPplus_vs_GFPminus") %>% select(WBGeneID:type, isDE),
          file = "L1_enrichment_annotation.csv")
write_csv(L3_alt_hyp_res_df %>% filter(label == "GFPplus_vs_GFPminus") %>% select(WBGeneID:type, isDE),
          file = "L3_enrichment_annotation.csv")

```

## Plot output

```

res_embryoGFPplus_vs_embryoGFPminus_ashr <- read_csv(file = "../03_output/pairwise_shrunk_DE_results/res_embryoGFPplus_vs_embryoGFPminus_ashr.csv")

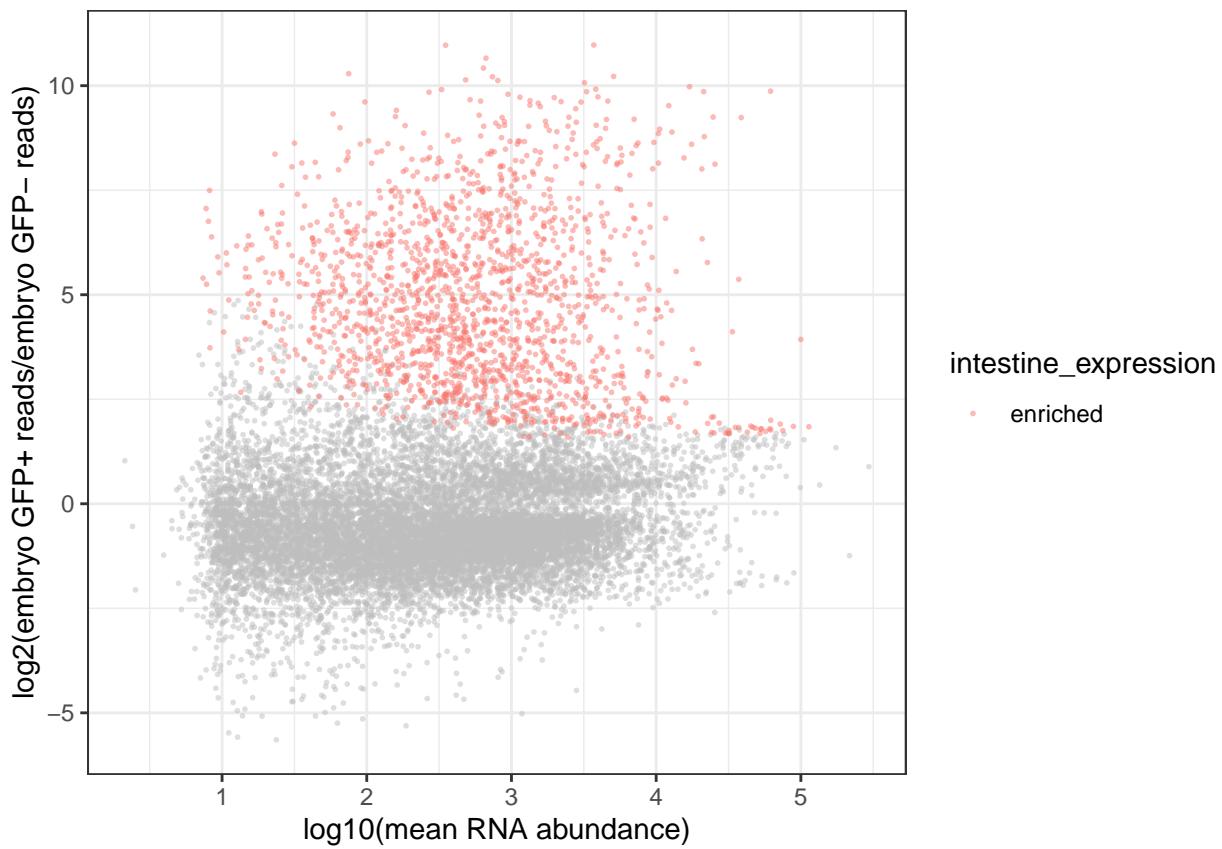
## Rows: 15627 Columns: 6
## -- Column specification -----
## Delimiter: ","
## chr (1): WBGeneID
## dbl (5): baseMean, log2FoldChange, lfcSE, pvalue, padj
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

embryo_intestine_gene_cats <- read_csv(file = "../03_output/intestine_gene_categories/embryo_intestine_gene_cats.csv")

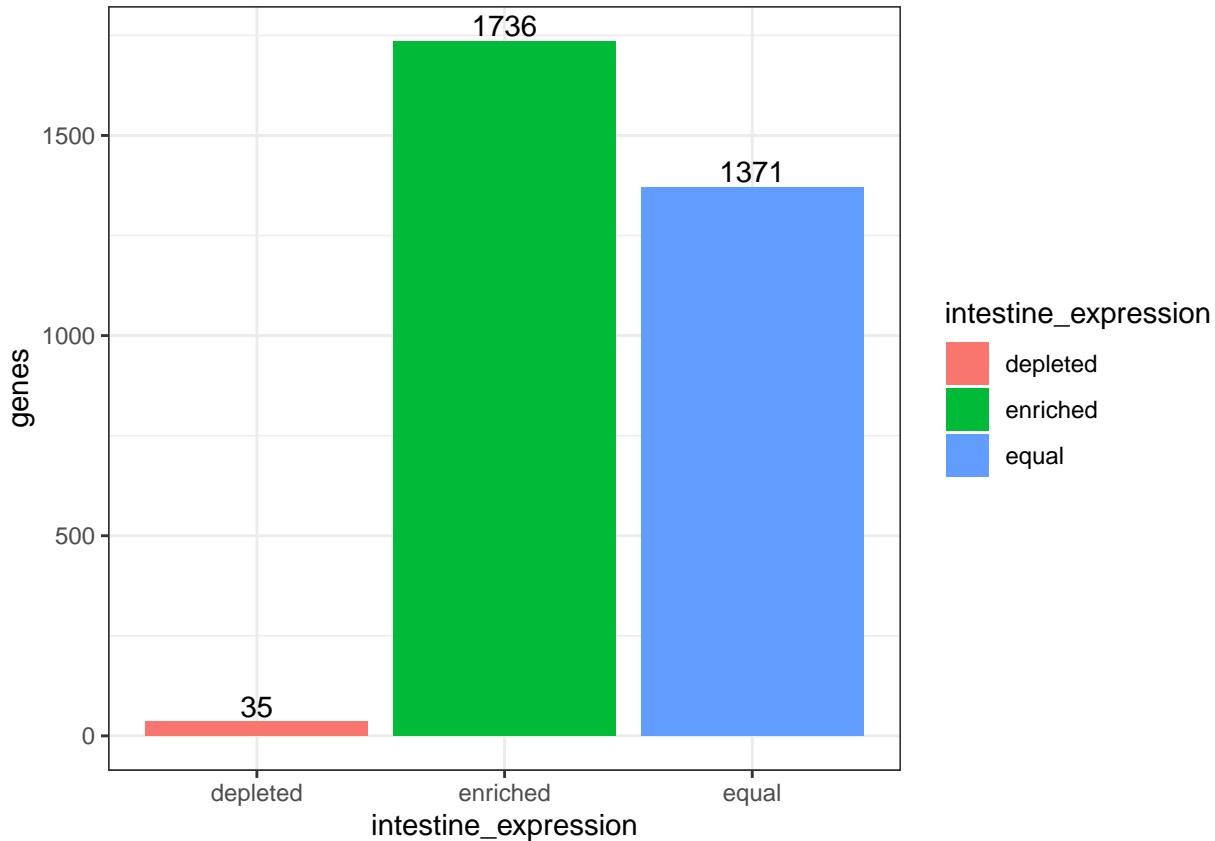
## Rows: 3142 Columns: 3
## -- Column specification -----
## Delimiter: ","
## chr (3): WBGeneID, altHyp, intestine_expression
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

res_embryoGFPplus_vs_embryoGFPminus_ashr %>% left_join(embryo_intestine_gene_cats, by = "WBGeneID") %>%
  ggplot(aes(x = log10(baseMean), y = log2FoldChange)) +
  geom_point(data = res_embryoGFPplus_vs_embryoGFPminus_ashr %>% filter(!(WBGeneID %in% embryo_intestine_gene_cats)),
             shape = 16, alpha = 0.5, stroke = 0, size = 1, color = "grey") +
  geom_point(shape = 16, alpha = 0.5, stroke = 0, size = 1, aes(color = intestine_expression)) +
  theme_bw() +
  labs(x = "log10(mean RNA abundance)",
       y = "log2(embryo GFP+ reads/embryo GFP- reads)")

```



```
ggsave(filename = "../03_output/plots/Intestine_Expression_Category/L3_GFPplus_vs_GFPminus_shrunk_MAplot.pdf", res_to_df(res_embryoGFPplus_vs_embryoGFPminus_ashr) %>% left_join(embryo_intestine_gene_cats, by = "WBGeneID"), geom_bar(stat = "identity") + geom_text(vjust = -0.25) + theme_bw())
```



```

single_MA_plot <- function(in_res, in_cats){
  res_to_df(in_res) %>% left_join(in_cats, by = "WBGeneID") %>% drop_na(intestine_expression) %>%
    filter(log2FoldChange > -20, intestine_expression == "enriched") %>%
  ggplot(aes(x = log10(baseMean), y = log2FoldChange)) +
  geom_point(data = res_to_df(in_res) %>% filter(!(WBGeneID %in% in_cats$WBGeneID)),
             shape = 16, alpha = 0.5, stroke = 0, size = 1, color = "grey") +
  geom_point(shape = 16, alpha = 0.5, stroke = 0, size = 1, aes(color = intestine_expression)) +
  theme_bw() +
  xlim(0.5,5.5) +
  ylim(-6, 10) +
  labs(x = "log10(mean RNA abundance)",
       y = "log2(GFP+ reads/GFP- reads)",
       title = paste("data: ", deparse(substitute(in_res)), sep = ""))
}

single_cat_bar <- function(in_res, in_cats){
  res_to_df(in_res) %>% left_join(in_cats, by = "WBGeneID") %>% drop_na(intestine_expression) %>% group_by(
    geom_bar(stat = "identity") +
    geom_text(vjust = -0.25) +
    ggtitle(paste("data: ", deparse(substitute(in_cats)), sep = "")) +
    theme_bw()
  }

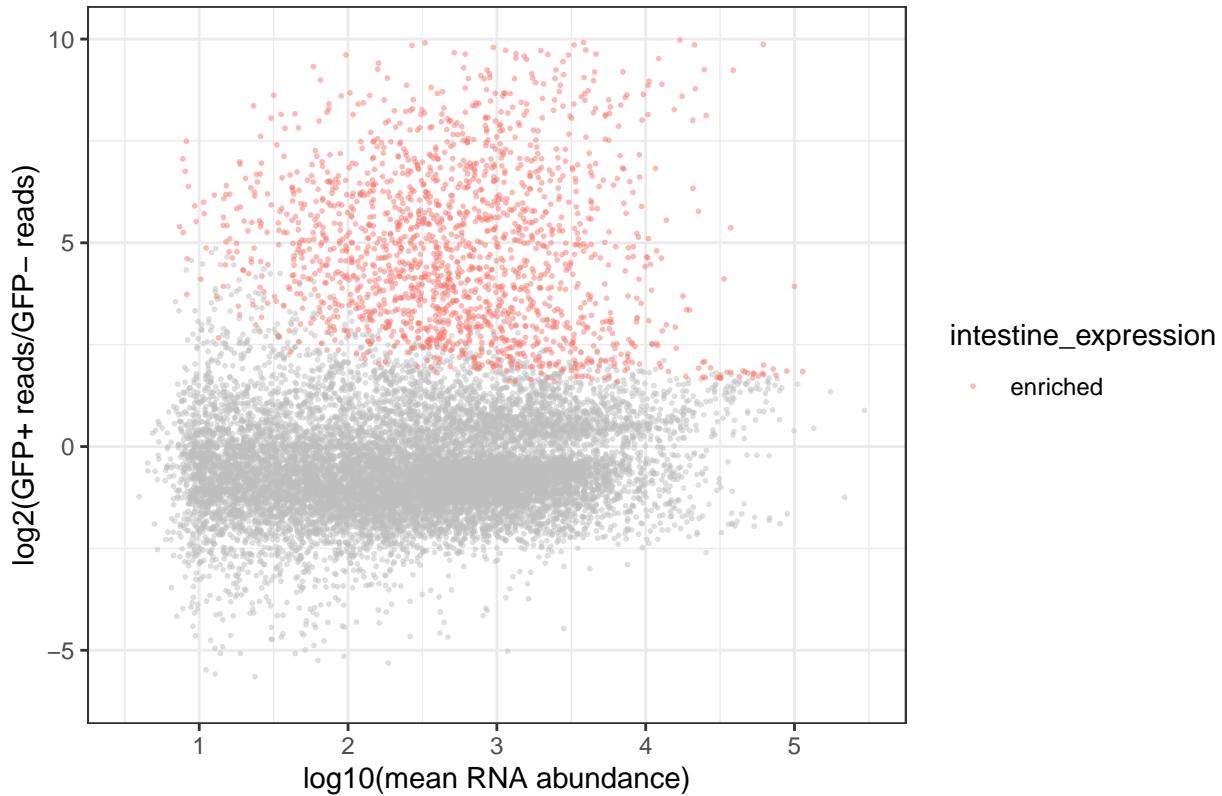
embryo_shrunk_MA <- single_MA_plot(res_embryoGFPplus_vs_embryoGFPminus_ashr, embryo_intestine_gene_cats)
embryo_shrunk_MA

## Warning: Removed 3 rows containing missing values (geom_point).

```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```

```
data: res_embryoGFPplus_vs_embryoGFPminus_ashr
```



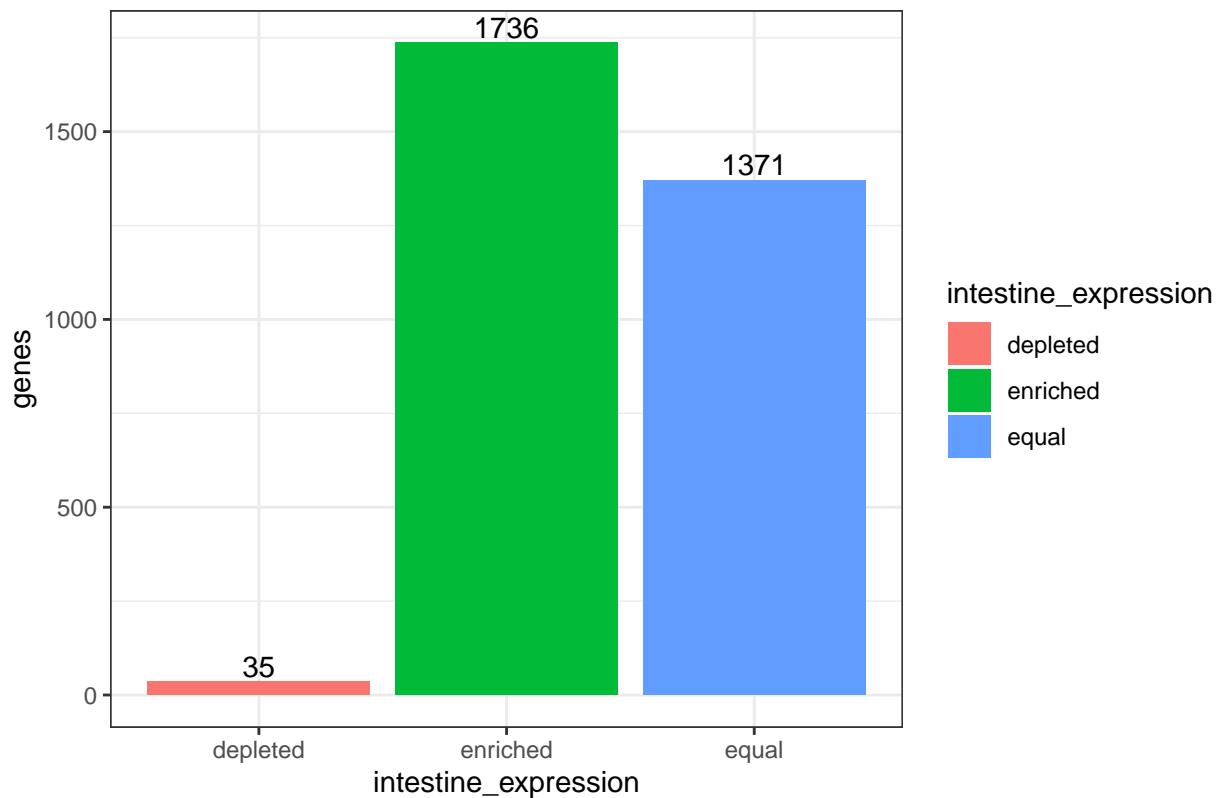
```
ggsave(plot = embryo_shrunk_MA, file = "../03_output/plots/Intestine_Expression_Category/embryo_GFPplus_
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

```
## Removed 10 rows containing missing values (geom_point).
```

```
embryo_category_bar <- single_cat_bar(res_embryoGFPplus_vs_embryoGFPminus_ashr, embryo_intestine_gene_c
```

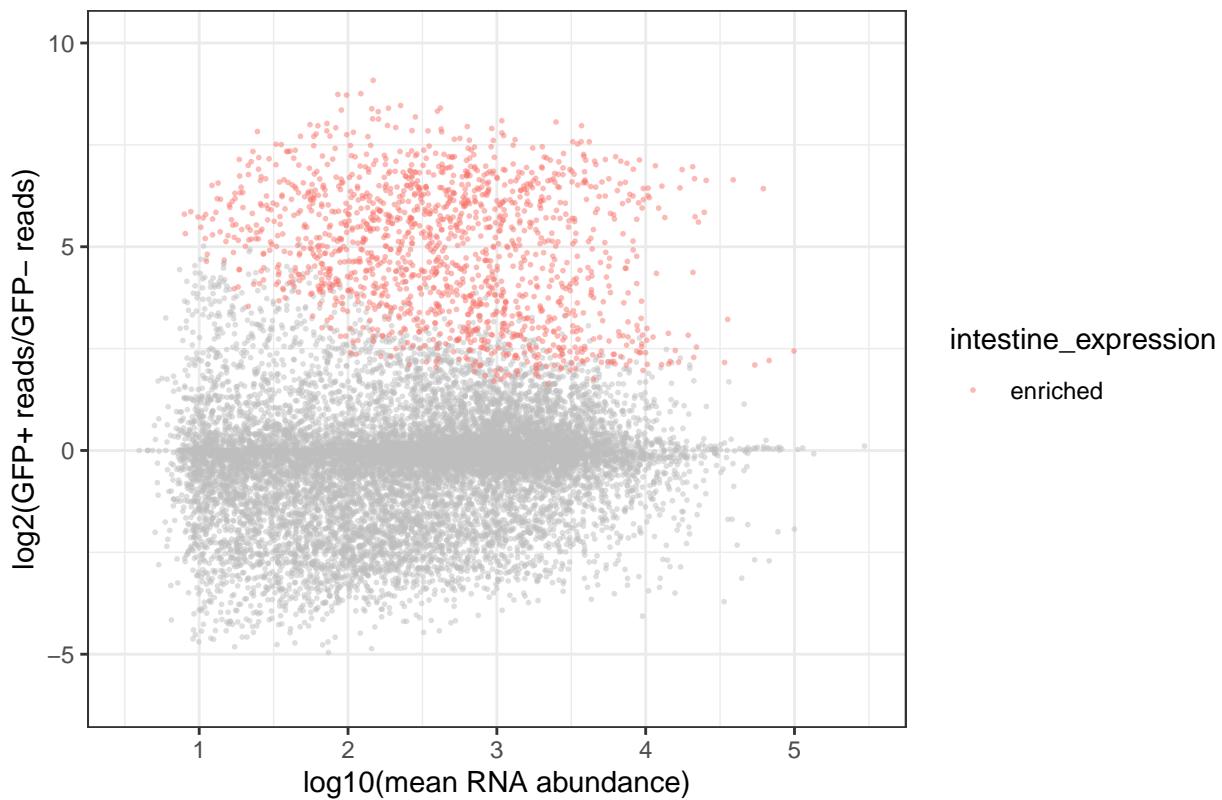
data: embryo\_intestine\_gene\_cats



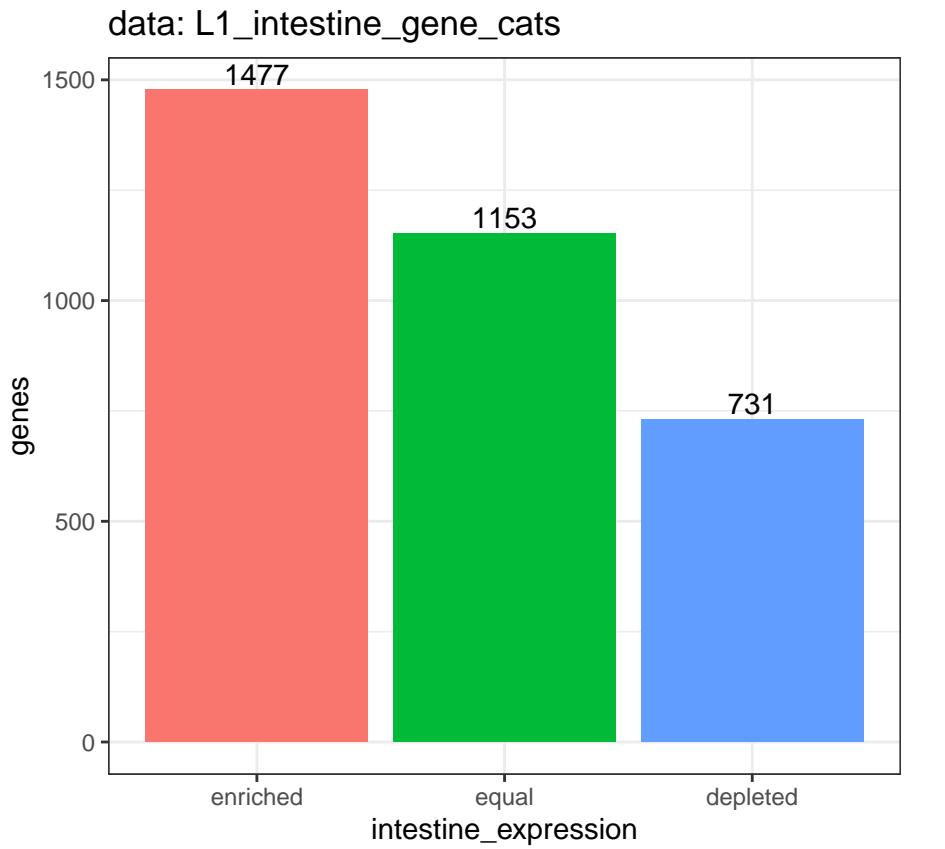
```
ggsave(plot = embryo_category_bar, file = "../03_output/plots/Intestine_Expression_Category/embryo_GFPPplus_vs_GFPPminus_barplot.pdf")
L1_shrunk_MA <- single_MA_plot(res_L1GFPPplus_vs_L1GFPPminus_ashr, L1_intestine_gene_cats)
L1_shrunk_MA

## Warning: Removed 3 rows containing missing values (geom_point).
```

data: res\_L1GFPplus\_vs\_L1GFPminus\_ashr



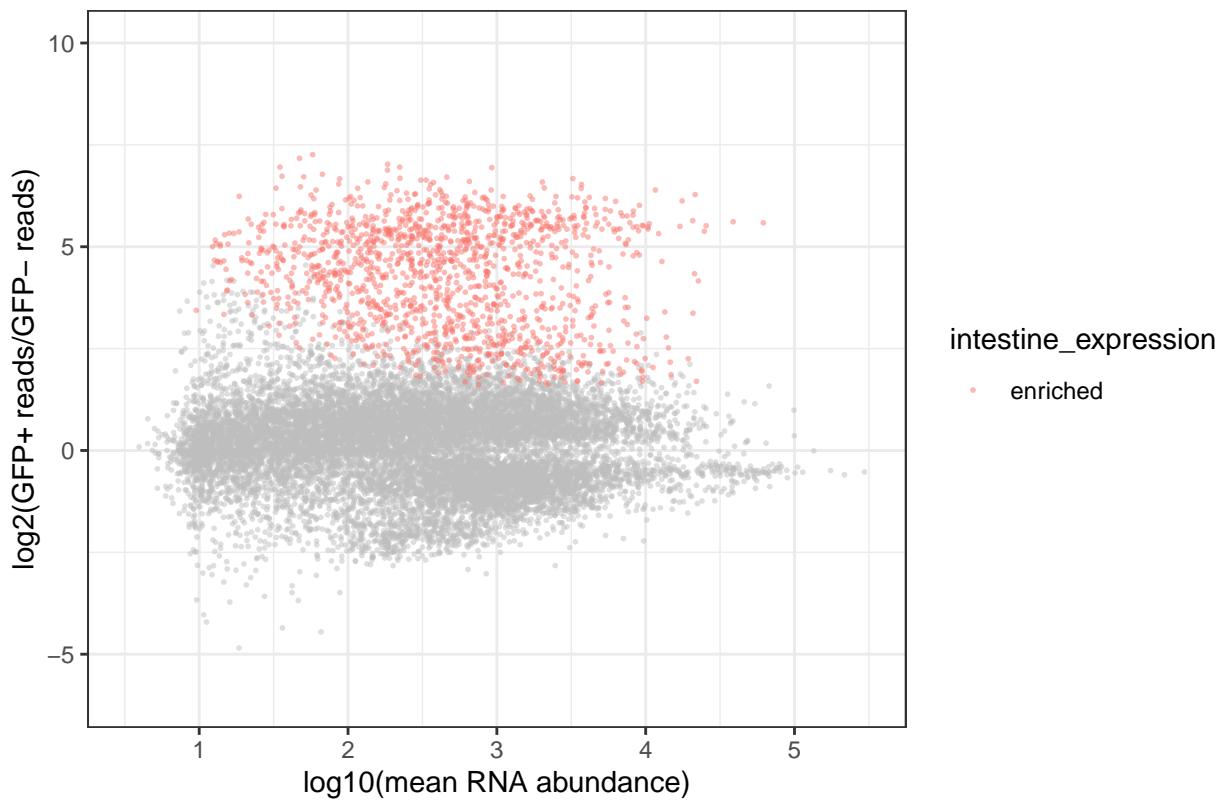
```
ggsave(plot = L1_shrunk_MA, file = "../03_output/plots/Intestine_Expression_Category/L1_GFPplus_vs_GFPminus_barplot.pdf")
## Warning: Removed 3 rows containing missing values (geom_point).
L1_category_bar <- single_cat_bar(res_L1GFPplus_vs_L1GFPminus_ashr, L1_intestine_gene_cats)
L1_category_bar
```



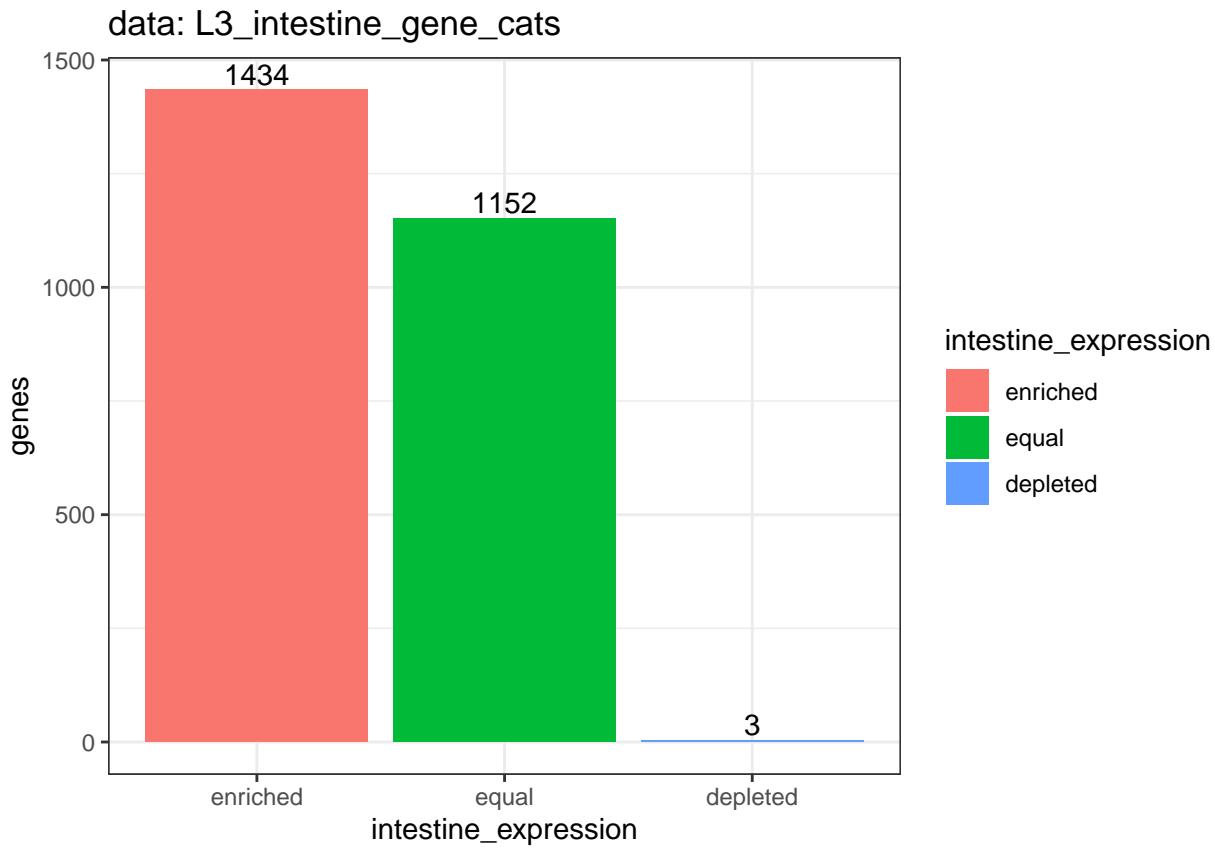
```
ggsave(plot = L1_category_bar, file = "../03_output/plots/Intestine_Expression_Category/L1_GFPplus_vs_GFPminus_barplot.pdf")
L3_shrunk_MA <- single_MA_plot(res_L3GFPplus_vs_L3GFPminus_ashr, L3_intestine_gene_cats)
L3_shrunk_MA

## Warning: Removed 3 rows containing missing values (geom_point).
```

data: res\_L3GFPplus\_vs\_L3GFPminus\_ashr



```
ggsave(plot = L3_shrunk_MA, file = "../03_output/plots/Intestine_Expression_Category/L3_GFPplus_vs_GFPminus_ashr.pdf")
## Warning: Removed 3 rows containing missing values (geom_point).
L3_category_bar <- single_cat_bar(res_L3GFPplus_vs_L3GFPminus_ashr, L3_intestine_gene_cats)
L3_category_bar
```



```
ggsave(plot = L3_category_bar, file = "../03_output/plots/Intestine_Expression_Category/L3_GFPplus_vs_GF...
```

## Alternative Hypothesis output

```
thresh <- 1
sig <- 0.01

res_embryoGFP_alHyp_greater <- results(dds, contrast = c("group", "embryoGFPplus", "embryoGFPminus"), lfcThreshold=thresh, sigLevel=sig)
write_csv(x = res_to_df(res_embryoGFP_alHyp_greater), file = "../03_output/res_embryoGFP_alHyp_greater.csv")

res_L1GFP_alHyp_greater <- results(dds, contrast = c("group", "L1GFPplus", "L1GFPminus"), lfcThreshold=thresh, sigLevel=sig)
write_csv(x = res_to_df(res_L1GFP_alHyp_greater), file = "../03_output/res_L1GFP_alHyp_greater.csv")

res_L3GFP_alHyp_greater <- results(dds, contrast = c("group", "L3GFPplus", "L3GFPminus"), lfcThreshold=thresh, sigLevel=sig)
write_csv(x = res_to_df(res_L3GFP_alHyp_greater), file = "../03_output/res_L3GFP_alHyp_greater.csv")
```

## Session info

```
sessionInfo()

## R version 4.1.0 (2021-05-18)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.7
##
## Matrix products: default
```

```

## BLAS: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      parallel  stats4    stats     graphics  grDevices utils
## [8] datasets  methods   base
##
## other attached packages:
## [1] ComplexHeatmap_2.8.0      InterMineR_1.14.1
## [3] ashr_2.2-54              apeglm_1.14.0
## [5] forcats_0.5.1            stringr_1.4.0
## [7] dplyr_1.0.8               purrr_0.3.4
## [9] readr_2.1.2               tidyrr_1.2.0
## [11] tibble_3.1.6              ggplot2_3.3.5
## [13] tidyverse_1.3.1            pheatmap_1.0.12
## [15] RColorBrewer_1.1-3        corrplot_0.92
## [17] DESeq2_1.32.0             SummarizedExperiment_1.22.0
## [19] Biobase_2.52.0            MatrixGenerics_1.4.3
## [21] matrixStats_0.61.0        GenomicRanges_1.44.0
## [23] GenomeInfoDb_1.28.4       IRanges_2.26.0
## [25] S4Vectors_0.30.2          BiocGenerics_0.38.0
##
## loaded via a namespace (and not attached):
## [1] colorspace_2.0-3           rjson_0.2.21          ellipsis_0.3.2
## [4] circlize_0.4.14            XVector_0.32.0         GlobalOptions_0.1.2
## [7] fs_1.5.2                  clue_0.3-60            rstudioapi_0.13
## [10] farver_2.1.0              bit64_4.0.5            AnnotationDbi_1.54.1
## [13] fansi_1.0.3               mvtnorm_1.1-3          lubridate_1.8.0
## [16] sqldf_0.4-11              xml2_1.3.3             codetools_0.2-18
## [19] splines_4.1.0              doParallel_1.0.17       cachem_1.0.6
## [22] geneplotter_1.70.0         knitr_1.38             jsonlite_1.8.0
## [25] Cairo_1.5-15              broom_0.8.0            annotate_1.70.0
## [28] cluster_2.1.3              dbplyr_2.1.1           png_0.1-7
## [31] compiler_4.1.0             httr_1.4.2             backports_1.4.1
## [34] assertthat_0.2.1            Matrix_1.4-1           fastmap_1.1.0
## [37] cli_3.2.0                 htmltools_0.5.2         tools_4.1.0
## [40] igraph_1.3.0              coda_0.19-4            gtable_0.3.0
## [43] glue_1.6.2                GenomeInfoDbData_1.2.6 Rcpp_1.0.8.3
## [46] bbmle_1.0.24              cellranger_1.1.0        RJSONIO_1.3-1.6
## [49] vctrs_0.4.0                Biostrings_2.60.2       iterators_1.0.14
## [52] xfun_0.30                 proto_1.0.0            rvest_1.0.2
## [55] irlba_2.3.5               lifecycle_1.0.1         XML_3.99-0.9
## [58] zlibbioc_1.38.0            MASS_7.3-56             scales_1.2.0
## [61] vroom_1.5.7               hms_1.1.1              yaml_2.3.5
## [64] memoise_2.0.1              emdbook_1.3.12         bdsmatrix_1.3-4
## [67] stringi_1.7.6              RSQLite_2.2.12          SQUAREM_2021.1
## [70] highr_0.9                  genefilter_1.74.1       foreach_1.5.2
## [73] BiocParallel_1.26.2        shape_1.4.6             truncnorm_1.0-8
## [76] chron_2.3-56              rlang_1.0.2             pkgconfig_2.0.3
## [79] bitops_1.0-7               evaluate_0.15          lattice_0.20-45
## [82] invgamma_1.1                labeling_0.4.2          bit_4.0.4

```

```
## [85] tidyselect_1.1.2      plyr_1.8.7          magrittr_2.0.3
## [88] R6_2.5.1              generics_0.1.2       DelayedArray_0.18.0
## [91] DBI_1.1.2              gsubfn_0.7           pillar_1.7.0
## [94] haven_2.4.3             withr_2.5.0          survival_3.3-1
## [97] KEGGREST_1.32.0        RCurl_1.98-1.6       mixsqp_0.3-43
## [100] modelr_0.1.8           crayon_1.5.1         utf8_1.2.2
## [103] tzdb_0.3.0              rmarkdown_2.13        GetoptLong_1.0.5
## [106] locfit_1.5-9.5          readxl_1.4.0          blob_1.2.3
## [109] reprex_2.0.1             digest_0.6.29        xtable_1.8-4
## [112] numDeriv_2016.8-1.1     munsell_0.5.0
```