# elt2_RNAseq

Robert Williams

3/9/2022

Load packages

Load count matrix

```
elt2D_counts <- read_csv(file = "../01_input/Table_S1_Raw_Read_Counts_wt_elt2D.csv") %>% column_to_rowna
```

```
## Rows: 16708 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): WBGeneID
## dbl (8): wt_sorted_1, wt_sorted_2, wt_sorted_3, wt_sorted_4, elt2D_sorted_1,...
##
## 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(elt2D_counts)
```

```
##                wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## WBGene00000001         532         462         458         525            546
## WBGene00000002         192         165         185         195            169
## WBGene00000003         577         425         649         694            371
## WBGene00000004        2111        1794        2131        1999           1158
## WBGene00000005          11           8          13           6              9
## WBGene00000007          71          82          69          92             19
##                elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4
## WBGene00000001            919            575            661
## WBGene00000002            226            157            147
## WBGene00000003            557            405            429
## WBGene00000004           1832           1233           1288
## WBGene00000005             11              8             10
## WBGene00000007             36             15             18
```

```
coldata <- data.frame(condition = c(rep("wt", 4), rep("elt2D", 4)))
coldata$condition <- factor(coldata$condition, levels = c("wt","elt2D"))
rownames(coldata) <- colnames(elt2D_counts)
all(rownames(coldata) == colnames(elt2D_counts))
```

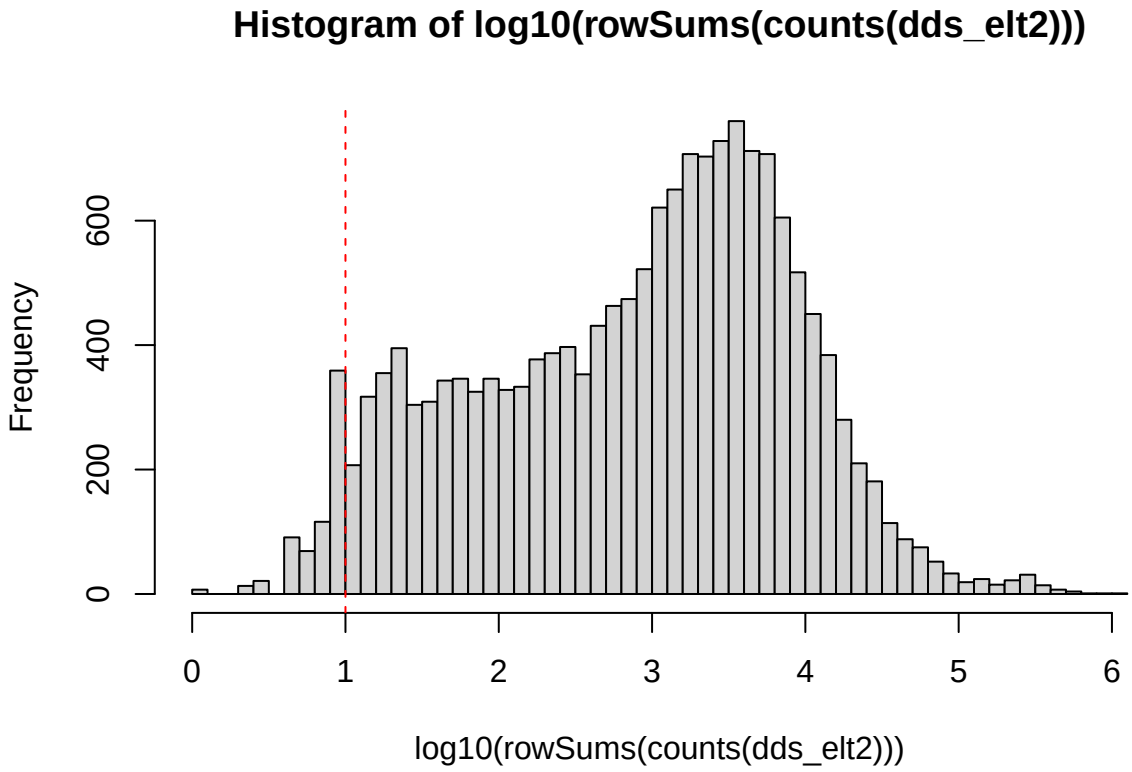
```
## [1] TRUE
```

```
dds_elt2 <- DESeqDataSetFromMatrix(countData = elt2D_counts,
                              colData = coldata,
                              design = ~ condition)
```

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

Visualize read count distribution

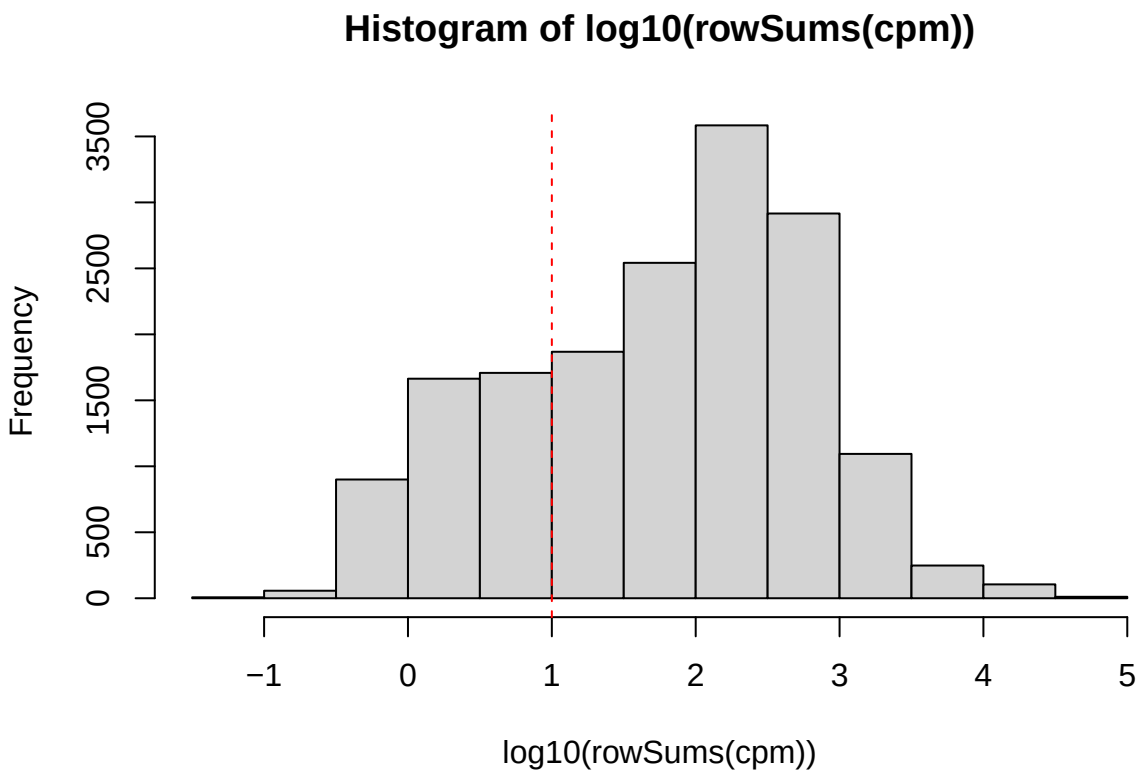
```
raw_count_threshold <- 10
hist(log10(rowSums(counts(dds_elt2))), breaks = 50)
abline(v = log10(raw_count_threshold), col = "red", lty = 2)
```




```
# Filter
```

genes with sum counts per million >= 10 across all samples

```
cpm <- apply(counts(dds_elt2),2, function(x) (x/sum(x))*1000000)
hist(log10(rowSums(cpm)))
abline(v = log10(raw_count_threshold), col = "red", lty = 2)
```




Filter genes with low read counts

```
keep <- rowSums(cpm) >= raw_count_threshold
dds_elt2 <- dds_elt2[keep,]
dds_elt2
```

```
## class: DESeqDataSet
## dim: 12368 8
## metadata(1): version
## assays(1): counts
## rownames(12368): WBGene00000001 WBGene00000002 ... WBGene00235114
##   WBGene00077643
## rowData names(0):
## colnames(8): wt_sorted_1 wt_sorted_2 ... elt2D_sorted_3 elt2D_sorted_4
## colData names(1): condition
```

Perform Differential Expression

```
dds_elt2 <- DESeq(dds_elt2)
```

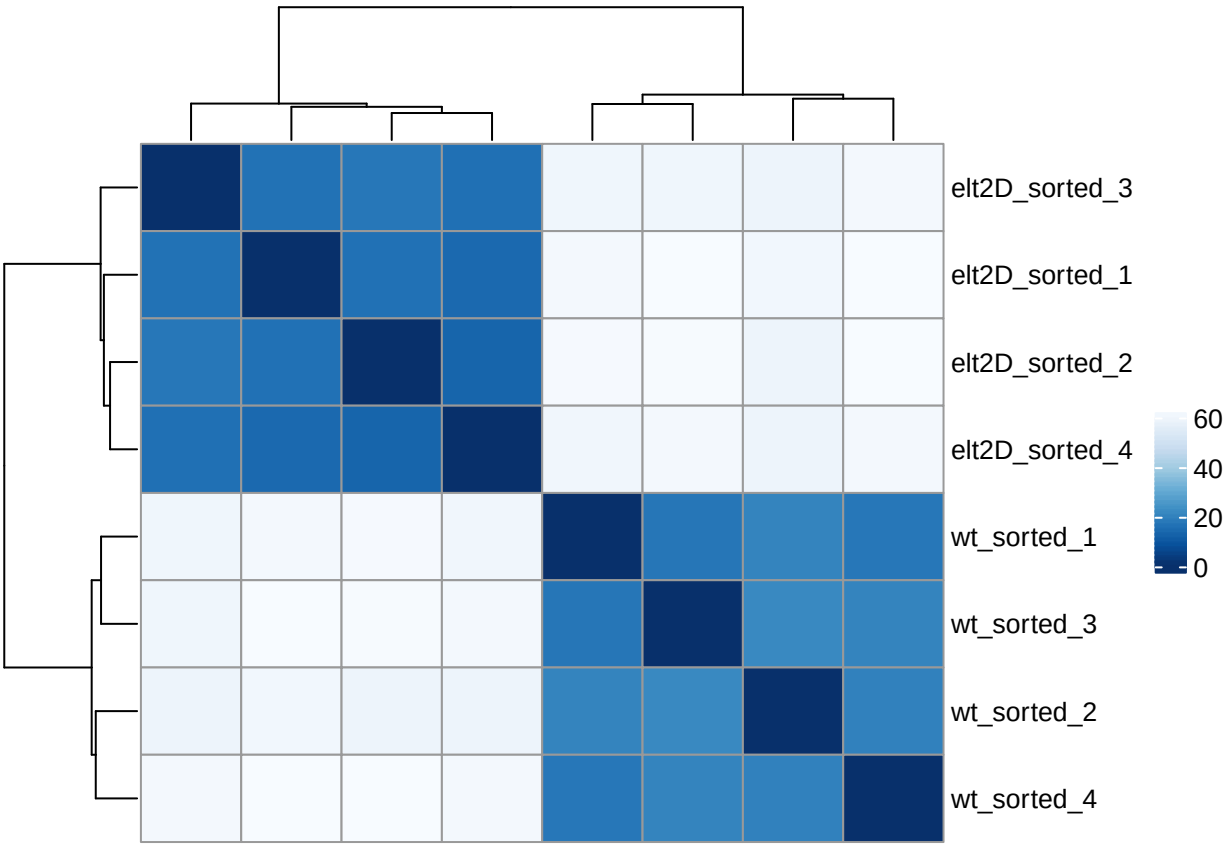
```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

```
resultsNames(dds_elt2)
```

```
## [1] "Intercept"             "condition_elt2D_vs_wt"
```

## Sample-to-sample distance matrix

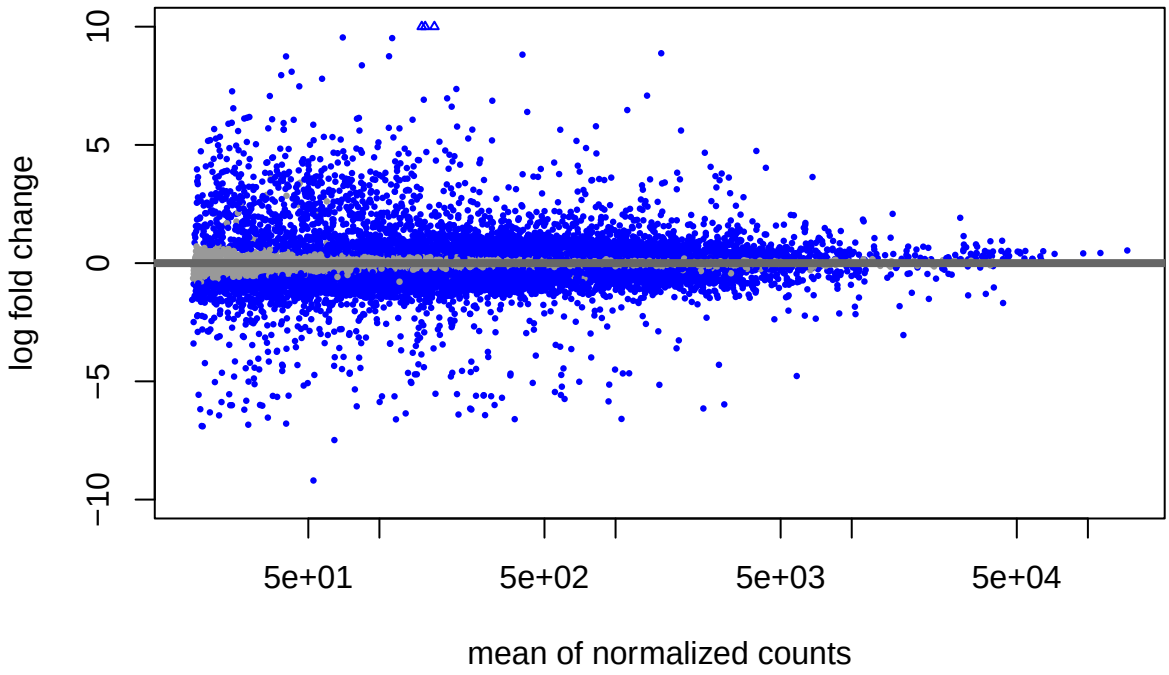
```
vsd <- vst(dds_elt2, blind = FALSE)
sampleDists <- dist(t(assay(vsd)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- colnames(vsd)
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         col = colors)
```




# Differential expression

```
res_elt2D_v_wt <- results(dds_elt2)
```

```
plotMA(dds_elt2, ylim = c(-10,10))
```




Output data table

```
# write_csv(res_to_df(res_elt2D_v_wt), file = "../03_output/res_elt2D_v_wt.csv")
```

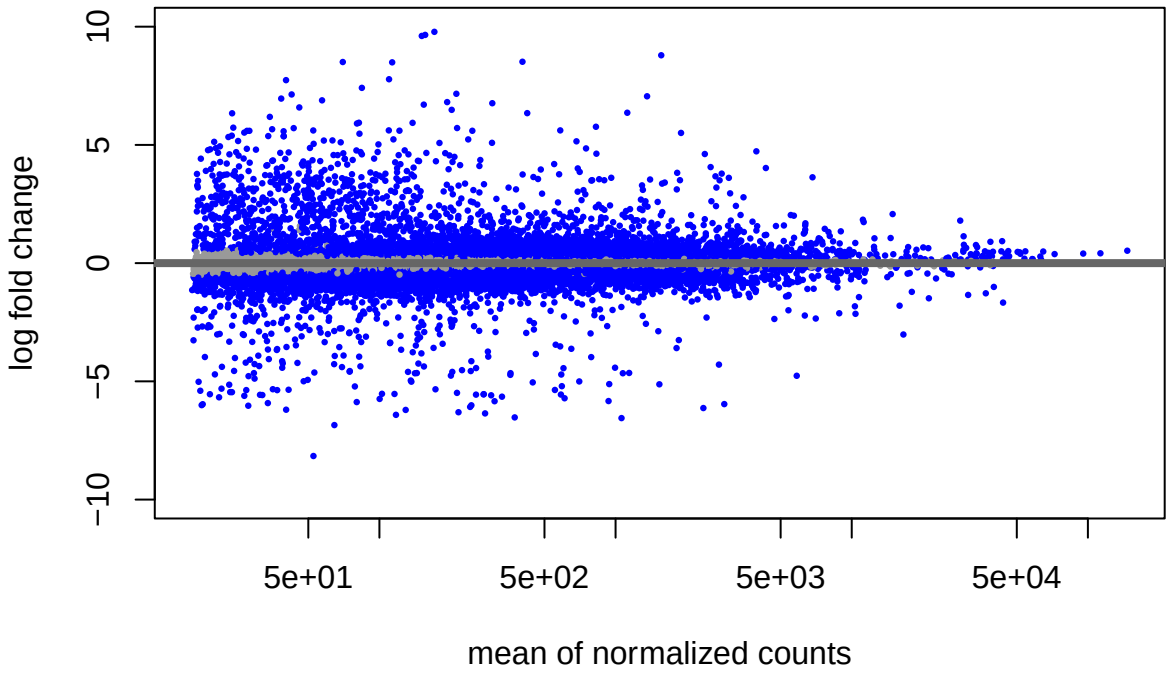
## Log2Shrinkage for gene ranking

```
res_elt2D_v_wt_ashr <- lfcShrink(dds_elt2, coef = "condition_elt2D_vs_wt", 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
```

```
# write_csv(file = "../03_output/res_elt2D_v_wt_ashr_shrunk.csv", x = res_to_df(res_elt2D_v_wt_ashr))
```

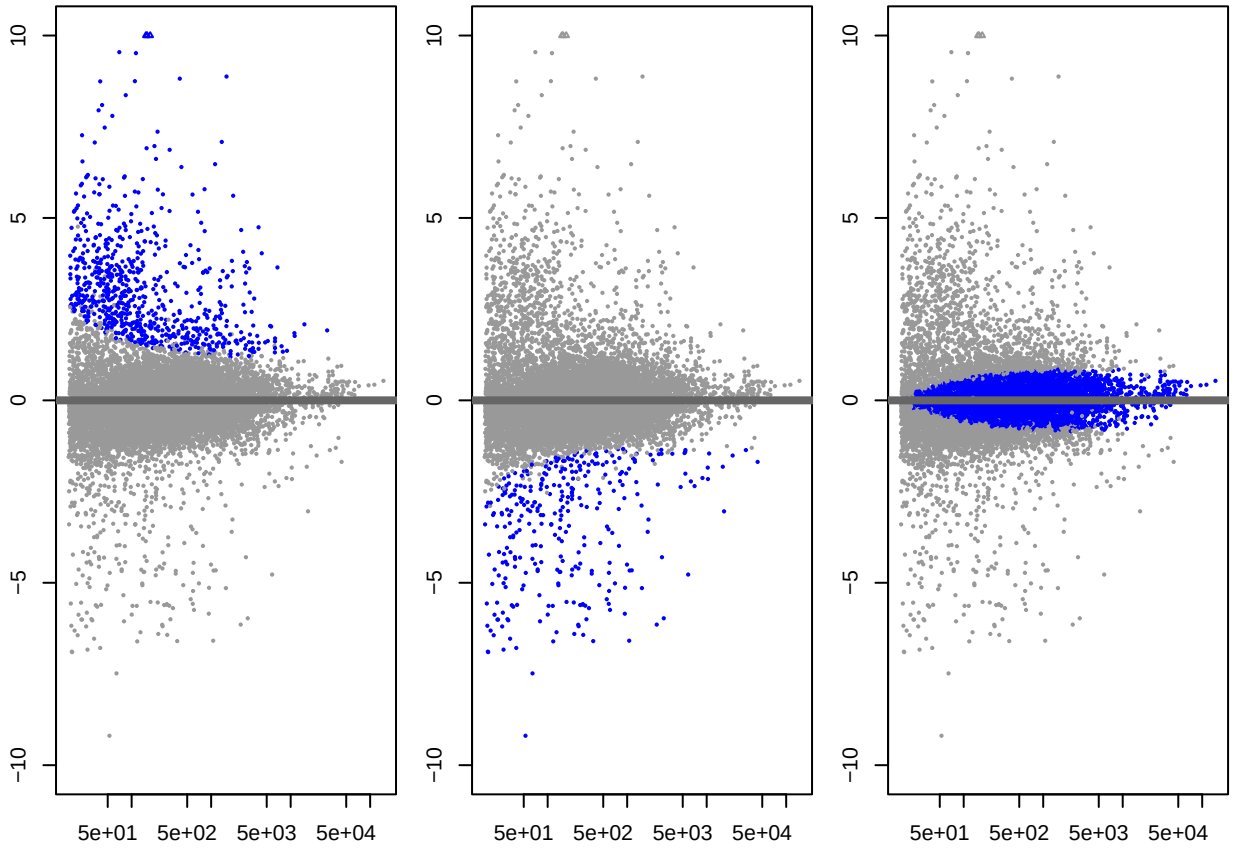
```
plotMA(res_elt2D_v_wt_ashr, ylim = c(-10,10))
```




## Alternative hypothesis testing

```
thresh <- 1
sig <- 0.01
res_altHyp_greater <- results(dds_elt2, altHypothesis = "greater", lfcThreshold = thresh, alpha = sig)
res_altHyp_less <- results(dds_elt2, altHypothesis = "less", lfcThreshold = thresh, alpha = sig)
res_altHyp_lessAbs <- results(dds_elt2, altHypothesis = "lessAbs", lfcThreshold = thresh, alpha = sig)
```

```
par(mfrow=c(1,3),mar=c(2,2,1,1))
ylim <- c(-10,10)
plotMA(res_altHyp_greater, ylim = ylim)
plotMA(res_altHyp_less, ylim = ylim)
plotMA(res_altHyp_lessAbs, ylim = ylim)
```




```
elt2D_greater <- as.data.frame(res_altHyp_greater) %>%
  rownames_to_column(var = "WBGeneID") %>%
  filter(padj < 0.01) %>% mutate(altHyp = "greater", description = "up_ELT2_minus")

elt2D_less <- as.data.frame(res_altHyp_less) %>%
  rownames_to_column(var = "WBGeneID") %>%
  filter(padj < 0.01) %>% mutate(altHyp = "less", description = "down_ELT2_minus")

elt2D_lessAbs <- as.data.frame(res_altHyp_lessAbs) %>%
  rownames_to_column(var = "WBGeneID") %>%
  filter(padj < 0.01) %>% mutate(altHyp = "lessAbs", description = "unchanged_ELT2_minus")

elt2_regulated_genes <- bind_rows(elt2D_greater, elt2D_less, elt2D_lessAbs) %>% select(WBGeneID, altHyp
elt2_regulated_genes$description <- factor(elt2_regulated_genes$description, levels = c("up_ELT2_minus"
head(elt2_regulated_genes)
```

```
##         WBGeneID  altHyp   description
## 1 WBGene00000022 greater up_ELT2_minus
## 2 WBGene00000067 greater up_ELT2_minus
## 3 WBGene00000219 greater up_ELT2_minus
## 4 WBGene00000397 greater up_ELT2_minus
## 5 WBGene00000465 greater up_ELT2_minus
## 6 WBGene00000473 greater up_ELT2_minus
```

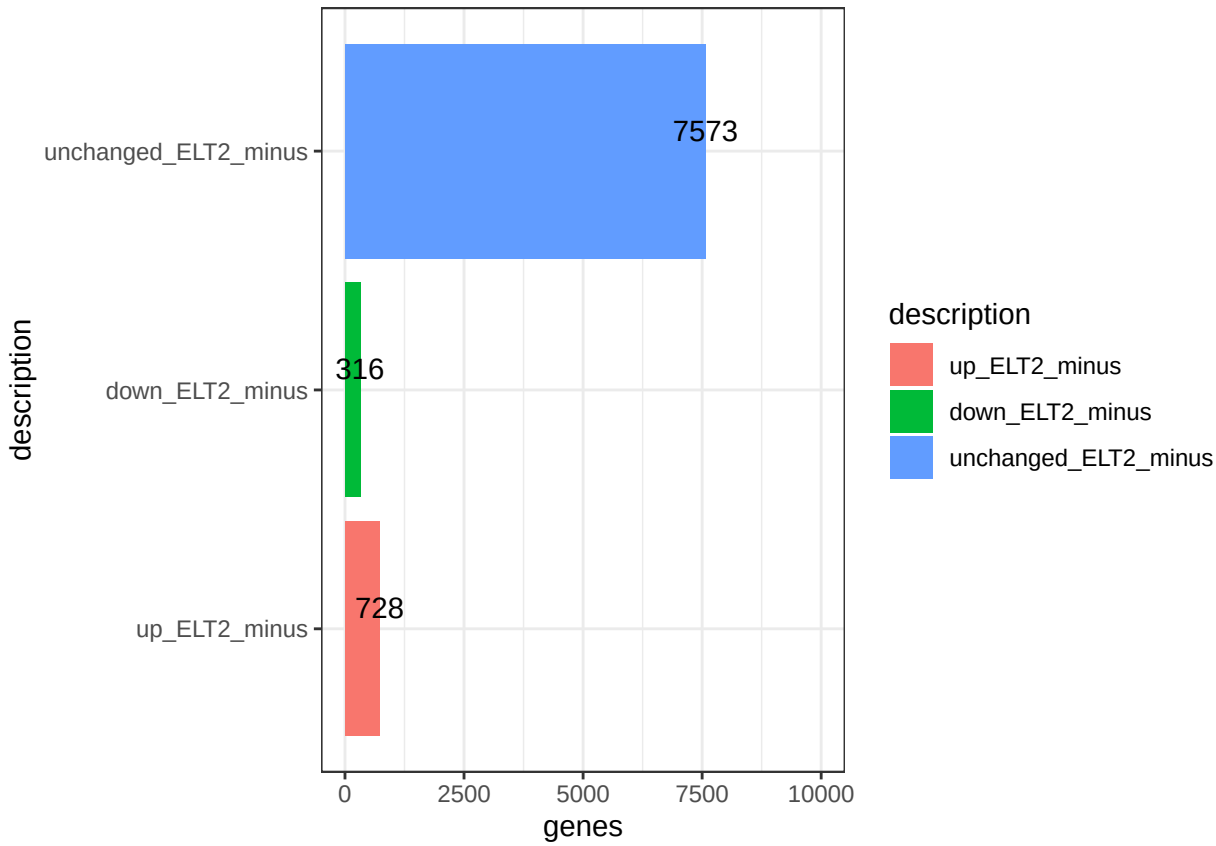
Add gene names

```
paramart <- biomaRt::useMart("parasite_mart", dataset = "wbps_gene", host = "https://parasite.wormbase.
```

```
elt2_regulated_genes <- biomaRt::getBM(
  mart = paramart,
  filter = c("wbps_gene_id"),
  value = elt2_regulated_genes$WBGeneID,
  attributes = c("wbps_gene_id","wormbase_gseq", "wikigene_name")
) %>% right_join(elt2_regulated_genes, by = c("wbps_gene_id" = "WBGeneID"))

elt2_regulated_genes<- elt2_regulated_genes %>% rename(WBGeneID = "wbps_gene_id")

# write_csv(elt2_regulated_genes, file = "../03_output/elt2_regulated_gene_sets.csv")
```

Counts of genes in each category

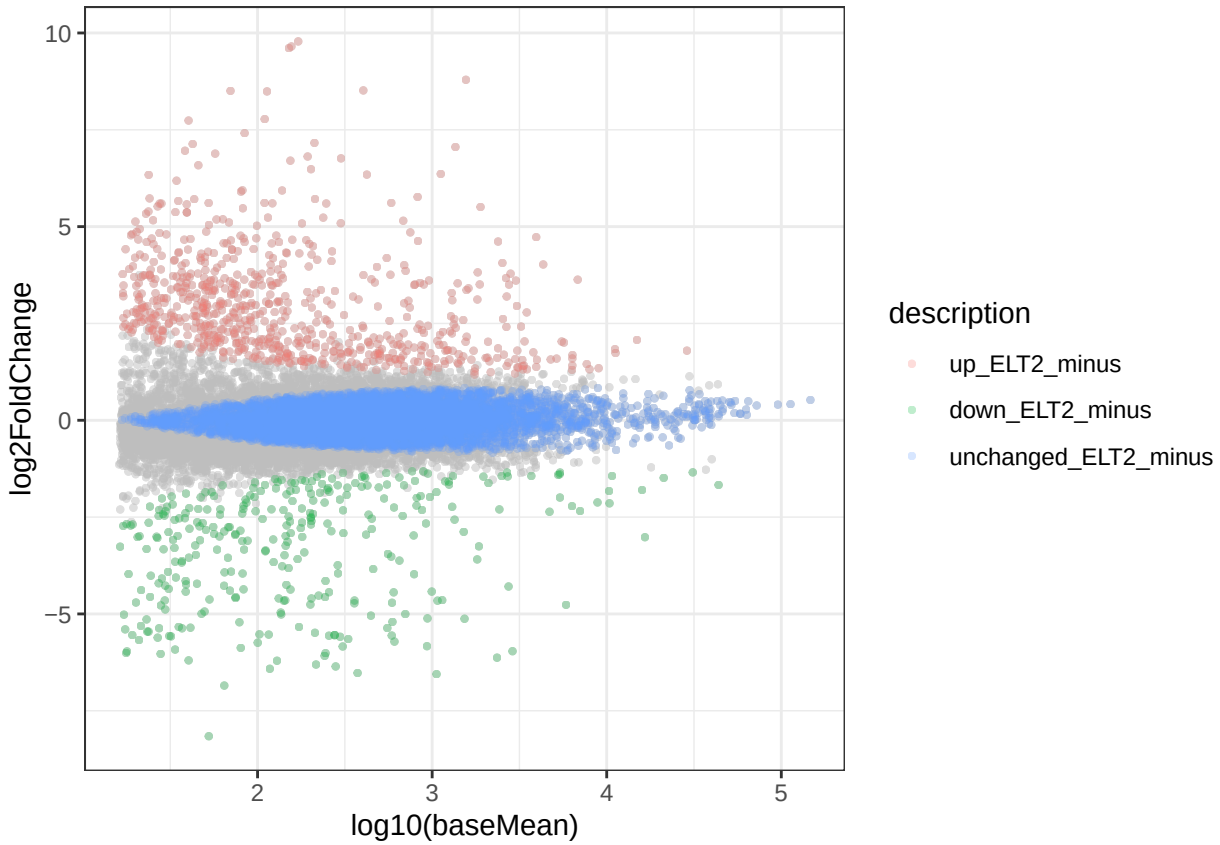
```
elt2_regulated_gene_counts <- elt2_regulated_genes %>% group_by(description) %>% summarise(genes = n()) %>%
  ggplot(aes(x = description, y = genes, fill = description, label = genes)) +
  geom_bar(stat = "identity") +
  geom_text(vjust = -0.5) +
  ylim(c(0, 10000)) +
  coord_flip() +
  theme_bw()

elt2_regulated_gene_counts
```




```
# ggsave(elt2_regulated_gene_counts, filename = "../03_output/elt2_regulated_gene_counts_plot.pdf", wid
```

## MA plot

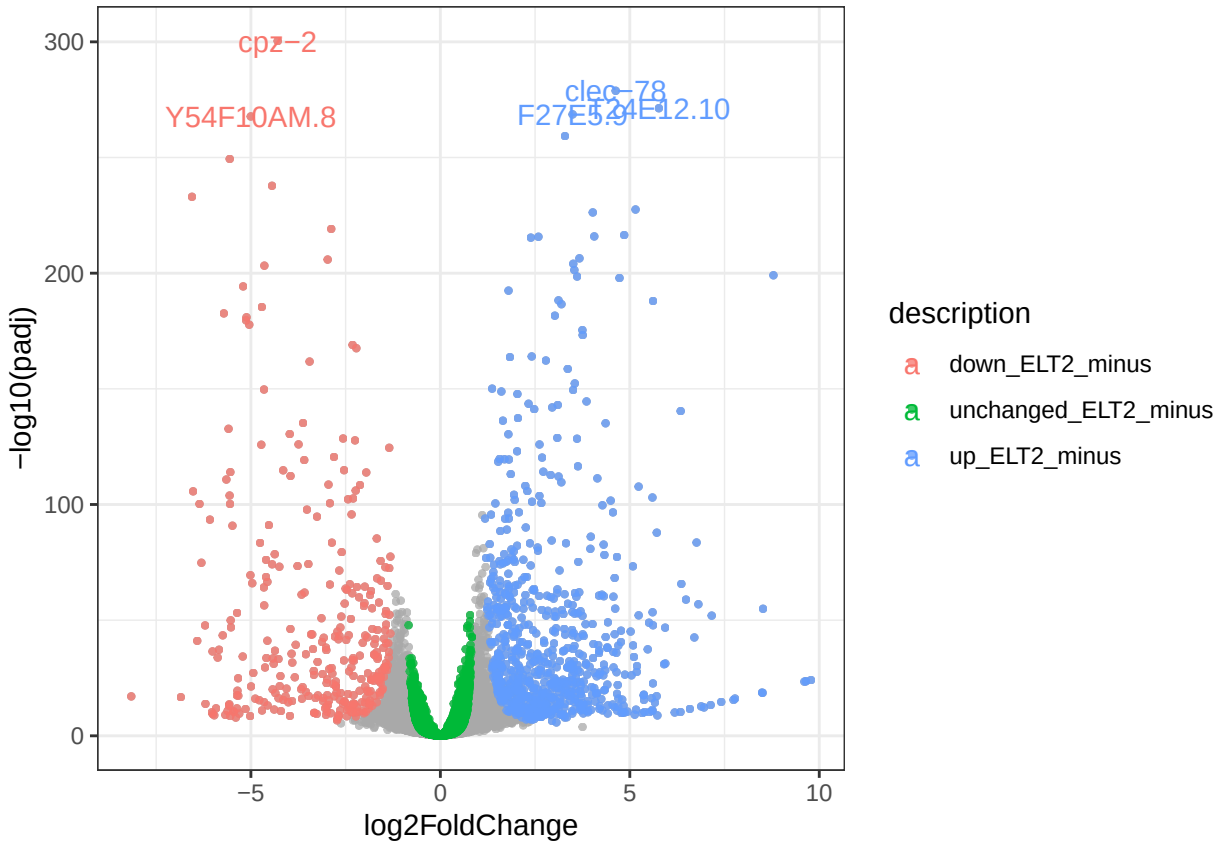
```
res_elt2D_v_wt_ashr_altHyp <- as.data.frame(res_elt2D_v_wt_ashr) %>% rownames_to_column(var = "WBGeneID

elt2D_vs_wt_MA <- ggplot(res_elt2D_v_wt_ashr_altHyp %>% filter(!is.na(description)), aes(x = log10(basel
  geom_point(data = res_elt2D_v_wt_ashr_altHyp, shape = 20, alpha = 0.5, stroke = 0, size = 2, color =
  geom_point(shape = 20, alpha = 0.25, stroke = 0, size = 2) +
  theme_bw()

elt2D_vs_wt_MA
```




```
ggsave(filename = "../03_output/elt2D_vs_wt_MA_plot.pdf",elt2D_vs_wt_MA, width = 5, height = 3, useDingb
```

## Volcano plot

```
elt2D_vs_wt_volcano <- ggplot(res_elt2D_v_wt_ashr_altHyp %>% filter(padj !=0, !is.na(description)),
       aes(x = log2FoldChange, y = -log10(padj), color = description)) +
  geom_point(data = res_elt2D_v_wt_ashr_altHyp %>% filter(padj !=0) %>% dplyr::select(-description),shap
  geom_text(data = res_elt2D_v_wt_ashr_altHyp %>% filter(padj !=0, !is.na(description)) %>% slice_max(-l
  geom_point(shape = 20, alpha = 0.75, stroke = 0, size = 2) +
  theme_bw()

elt2D_vs_wt_volcano
```




```
ggsave(filename = "../03_output/elt2D_vs_wt_volcano_plot.pdf", elt2D_vs_wt_volcano, width = 5, height =
```

## 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                tidyr_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] readxl_1.4.0           backports_1.4.1        circlize_0.4.14
##   [4] BiocFileCache_2.0.0    plyr_1.8.7             igraph_1.3.0
##   [7] splines_4.1.0          BiocParallel_1.26.2    digest_0.6.29
##  [10] invgamma_1.1           foreach_1.5.2          htmltools_0.5.2
##  [13] SQUAREM_2021.1         fansi_1.0.3            magrittr_2.0.3
##  [16] memoise_2.0.1          cluster_2.1.3          doParallel_1.0.17
##  [19] tzdb_0.3.0             Biostrings_2.60.2      annotate_1.70.0
##  [22] modelr_0.1.8           vroom_1.5.7            bdsmatrix_1.3-4
##  [25] prettyunits_1.1.1      colorspace_2.0-3       rappdirs_0.3.3
##  [28] blob_1.2.3             rvest_1.0.2            haven_2.4.3
##  [31] xfun_0.30              crayon_1.5.1           RCurl_1.98-1.6
##  [34] jsonlite_1.8.0         genefilter_1.74.1      survival_3.3-1
##  [37] iterators_1.0.14       glue_1.6.2             gtable_0.3.0
##  [40] zlibbioc_1.38.0        XVector_0.32.0         GetoptLong_1.0.5
##  [43] DelayedArray_0.18.0    shape_1.4.6            scales_1.2.0
##  [46] mvtnorm_1.1-3          DBI_1.1.2              Rcpp_1.0.8.3
##  [49] xtable_1.8-4           progress_1.2.2         emdbook_1.3.12
##  [52] clue_0.3-60            bit_4.0.4              sqldf_0.4-11
##  [55] truncnorm_1.0-8        httr_1.4.2             ellipsis_0.3.2
##  [58] farver_2.1.0           pkgconfig_2.0.3        XML_3.99-0.9
##  [61] dbplyr_2.1.1           locfit_1.5-9.5         utf8_1.2.2
##  [64] RJSONIO_1.3-1.6        labeling_0.4.2         tidyselect_1.1.2
##  [67] rlang_1.0.2            AnnotationDbi_1.54.1   munsell_0.5.0
##  [70] cellranger_1.1.0       tools_4.1.0            cachem_1.0.6
##  [73] cli_3.2.0              gsubfn_0.7             generics_0.1.2
##  [76] RSQLite_2.2.12         broom_0.8.0            evaluate_0.15
##  [79] fastmap_1.1.0          yaml_2.3.5             knitr_1.38
##  [82] bit64_4.0.5            fs_1.5.2               KEGGREST_1.32.0
##  [85] xml2_1.3.3             biomaRt_2.48.3         compiler_4.1.0
##  [88] rstudioapi_0.13        filelock_1.0.2         curl_4.3.2
##  [91] png_0.1-7              reprex_2.0.1           geneplotter_1.70.0
##  [94] stringi_1.7.6          highr_0.9              lattice_0.20-45
##  [97] Matrix_1.4-1           vctrs_0.4.0            pillar_1.7.0
## [100] lifecycle_1.0.1        GlobalOptions_0.1.2    bitops_1.0-7
## [103] irlba_2.3.5            R6_2.5.1               codetools_0.2-18
## [106] MASS_7.3-56            assertthat_0.2.1       chron_2.3-56
## [109] proto_1.0.0            rjson_0.2.21           withr_2.5.0
## [112] GenomeInfoDbData_1.2.6 hms_1.1.1              coda_0.19-4
## [115] rmarkdown_2.13         Cairo_1.5-15           mixsqp_0.3-43
## [118] bbmle_1.0.24           numDeriv_2016.8-1.1    lubridate_1.8.0
```