elt2_target_analysis

Robert Williams

3/9/2022

load packages

```
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

load data files

```
elt2_regulated_genes <- read_csv(file = "../../03_elt2_RNAseq/03_output/elt2_regulated_gene_sets.csv")

embryo_chip <- read.table(file = "../../04_promoters/03_output/LE.promoters.hilo.tsv") %>% rownames_to_
L1_chip <- read.table(file = "../../04_promoters/03_output/L1.promoters.hilo.tsv") %>% rownames_to_colu
L3_chip <- read.table(file = "../../04_promoters/03_output/L3.promoters.hilo.tsv") %>% rownames_to_colu

embryo_intestine_gene_categories <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/inte
L1_intestine_gene_categories <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/intestin
L3_intestine_gene_categories <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/intestin

transcript_type <- read_csv(file = "../01_input/biomaRt_elegans_transcript_biotype.csv")
colnames(transcript_type) <- c("WBGeneID", "genome_id", "gene_name", "biotype")
```

Join the data

```
rna_chip_df <- function(chip_df, category_df){
  chip_df %>% mutate(promoter_status = case_when(!is.na(IDR_mean) ~ "bound",
                                                 is.na(IDR_mean) ~ "not_bound")) %>%
  full_join(category_df, by = "WBGeneID") %>%
  full_join(elt2_regulated_genes %>% select(-wormbase_gseq, -wikigene_name), by = "WBGeneID") %>%
  select(WBGeneID, promoter_status, intestine_expression, elt2_ko = description) %>%
  mutate(intestine_expression = case_when(intestine_expression == "enriched" ~ "enriched",
                                          intestine_expression != "enriched" ~ "not_enriched",
                                          TRUE ~ "not_enriched")) %>%
  rows_update(tibble(WBGeneID = "WBGene00001250", promoter_status = "bound"), by = "WBGeneID") %>%
    replace_na(list(promoter_status = "not_bound"))
}

embryo_rna_chip <- rna_chip_df(embryo_chip, embryo_intestine_gene_categories)
L1_rna_chip <- rna_chip_df(L1_chip, L1_intestine_gene_categories)
L3_rna_chip <- rna_chip_df(L3_chip, L3_intestine_gene_categories)

all_stages_chip <- data.frame(L1_rna_chip, stage = "L1") %>% bind_rows(data.frame(embryo_rna_chip, stag
all_stages_chip$stage <- factor(all_stages_chip$stage, levels = c("embryo", "L1", "L3"))
```

```
all_stages_chip <-
  all_stages_chip %>% mutate(
    elt2_ko = case_when(
      elt2_ko == "up_ELT2_minus" ~ "repressed",
      elt2_ko == "down_ELT2_minus" ~ "activated",
      elt2_ko == "unchanged_ELT2_minus" ~ "independent"
    ),
    elt2_ko = fct_relevel(elt2_ko, c("activated", "repressed", "independent"))
  ) %>%
   filter(WBGeneID %in% (
     transcript_type %>% filter(biotype == "protein_coding") %>% distinct(WBGeneID) %>% pull(WBGeneID)
   ))


head(all_stages_chip)
```

```
##         WBGeneID promoter_status intestine_expression     elt2_ko stage
## 1 WBGene00011364           bound             enriched   repressed    L1
## 2 WBGene00016544           bound             enriched independent    L1
## 3 WBGene00008246           bound             enriched        <NA>    L1
## 4 WBGene00001940       not_bound             enriched        <NA>    L1
## 5 WBGene00007404       not_bound             enriched   activated    L1
## 6 WBGene00009854       not_bound             enriched        <NA>    L1
```

```
write_csv(all_stages_chip, file = "../03_output/all_stages_chip.csv")
```

Question: How many intestine enriched genes have ELT-2 binding?

```
all_stages_chip %>% group_by(stage, promoter_status) %>% summarise(pop = n()) %>% pivot_wider(names_fro
```

```
## `summarise()` has grouped output by 'stage'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 3 x 3
## # Groups:   stage [3]
##   stage  pop_bound_success pop_bound_fail
##   <fct>              <int>          <int>
## 1 embryo              2630          17365
## 2 L1                  3241          16754
## 3 L3                  5044          14951
```

```
bound_expressed_hyper <- all_stages_chip %>% filter(intestine_expression == "enriched") %>%
  group_by(stage, intestine_expression, promoter_status) %>%
  summarise(observed = n()) %>%
  ungroup() %>%
  group_by(stage, intestine_expression) %>%
  mutate(intestine_totals = sum(observed)) %>%
  ungroup() %>%
  group_by(stage, promoter_status) %>%
  mutate(promoter_totals = sum(observed)) %>%
  ungroup() %>%
  left_join(all_stages_chip %>%
              group_by(stage, promoter_status) %>%
              summarise(pop = n()) %>%
```

```
              pivot_wider(names_from = promoter_status, values_from = pop) %>%
              rename(pop_bound_success = bound, pop_bound_fail = not_bound),
            by = "stage") %>%
  rowwise() %>%
  mutate(dhyper = dhyper(x = observed, k = intestine_totals , m = pop_bound_success, n = pop_bound_fail
         expected = intestine_totals*pop_bound_success/(pop_bound_success+pop_bound_fail)) %>%
  pivot_longer(cols = c(observed, expected), names_to = "gene_type", values_to = "genes")%>%
  ungroup() %>%
  mutate(padj = p.adjust(dhyper, method = "BH")) %>%
  mutate(star = case_when(
    gene_type == "observed" & padj > 0.01 ~ "~",
    gene_type == "observed" & padj < 1*10^-10 ~ "***",
    gene_type == "observed" & padj < 1*10^-5 ~ "**",
    gene_type == "observed" & padj < 0.01 ~ "*"
                          ))
```

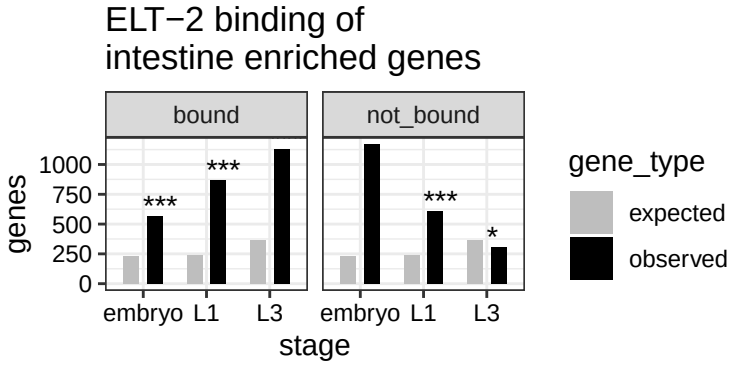
```
## `summarise()` has grouped output by 'stage', 'intestine_expression'. You can
## override using the `.groups` argument.
## `summarise()` has grouped output by 'stage'. You can override using the
## `.groups` argument.
```

```
bound_expressed_hyper
```

```
## # A tibble: 12 x 12
##    stage  intestine_expression promoter_status intestine_totals promoter_totals
##    <fct>  <chr>                <chr>                       <int>           <int>
##  1 embryo enriched             bound                        1736             567
##  2 embryo enriched             bound                        1736             567
##  3 embryo enriched             not_bound                    1736            1169
##  4 embryo enriched             not_bound                    1736            1169
##  5 L1     enriched             bound                        1477             869
##  6 L1     enriched             bound                        1477             869
##  7 L1     enriched             not_bound                    1477             608
##  8 L1     enriched             not_bound                    1477             608
##  9 L3     enriched             bound                        1434            1126
## 10 L3     enriched             bound                        1434            1126
## 11 L3     enriched             not_bound                    1434             308
## 12 L3     enriched             not_bound                    1434             308
## # ... with 7 more variables: pop_bound_success <int>, pop_bound_fail <int>,
## #   dhyper <dbl>, gene_type <chr>, genes <dbl>, padj <dbl>, star <chr>
```

```
bound_expressed_hyper_plot <- bound_expressed_hyper %>%
  ggplot(aes(x = stage, y = genes, fill = gene_type, label = star)) +
  geom_bar(position = position_dodge(width=0.75), stat = "identity", width = 0.5) +
  geom_text(vjust = 0, hjust = 0) +
  scale_fill_manual(values = c("grey", "black")) +
  # ylim(c(0, max(bound_expressed_hyper$genes)+150)) +
  scale_y_continuous(breaks = seq(0,3000, by = 250)) +
  theme_bw() +
  facet_wrap(~promoter_status) +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black")) +
  ggtitle("ELT-2 binding of\nintestine enriched genes")
bound_expressed_hyper_plot
```

```
## Warning: Removed 6 rows containing missing values (geom_text).
```




```
# ggsave(bound_expressed_hyper_plot, file = "../03_output/Gene_Set_Overlaps/bound_expressed_gene_counts_
```

Interpretation: More embryo enriched genes are not bound, bound genes increase over developmental time.

Question: How many ELT-2 targets are differentially expressed

```
bound_regulated_hyper <- all_stages_chip %>%
  drop_na(elt2_ko) %>%
  filter(promoter_status == "bound") %>%
  group_by(stage, elt2_ko, promoter_status) %>%
  summarise(observed = n()) %>%
  ungroup() %>%
  group_by(stage, elt2_ko) %>%
  mutate(elt2_ko_totals = sum(observed)) %>%
  ungroup() %>%
  group_by(stage, promoter_status) %>%
  mutate(promoter_totals = sum(observed)) %>%
  left_join(all_stages_chip %>%
              drop_na(elt2_ko) %>%
              group_by(stage, elt2_ko) %>%
              summarise(pop_regulated_success = n()) %>%
              mutate(pop_regulated_fail = 11461),
            by = c("stage", "elt2_ko")) %>%
  # mutate(pop_bound_success = 2239, pop_bound_fail = 6285) %>%
  ungroup() %>%
  rowwise() %>%
  mutate(dhyper = dhyper(x = observed, k = elt2_ko_totals , m = pop_regulated_success, n = pop_regulate
         expected = elt2_ko_totals*pop_regulated_success/(pop_regulated_success+pop_regulated_fail)) %>%
  pivot_longer(cols = c(observed, expected), names_to = "gene_type", values_to = "genes")%>%
  ungroup() %>%
  mutate(padj = p.adjust(dhyper, method = "BH")) %>%
  mutate(star = case_when(
    gene_type == "observed" & padj > 0.01 ~ "~",
    gene_type == "observed" & padj < 1*10^-10 ~ "***",
    gene_type == "observed" & padj < 1*10^-5 ~ "**",
    gene_type == "observed" & padj < 0.01 ~ "*"
                          ))
```

```
## `summarise()` has grouped output by 'stage', 'elt2_ko'. You can override using
## the `.groups` argument.
```

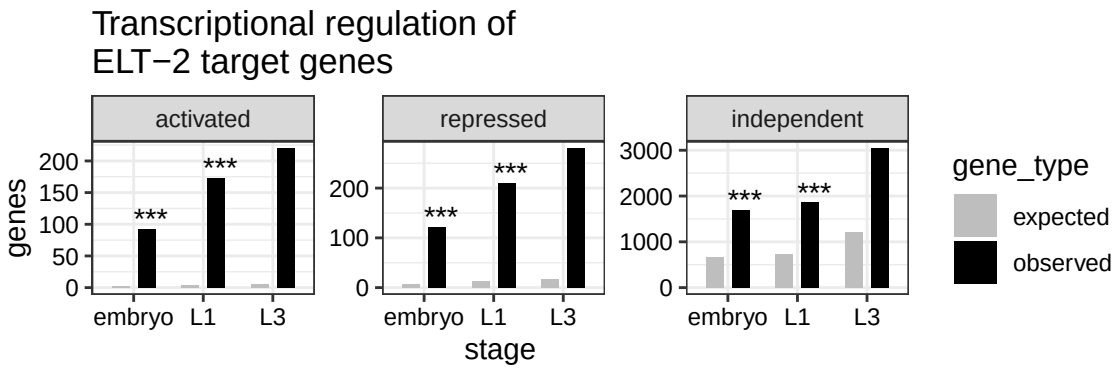
```
## `summarise()` has grouped output by 'stage'. You can override using the
## `.groups` argument.
```

```
bound_regulated_hyper
```

```
## # A tibble: 18 x 12
##    stage elt2_ko promoter_status elt2_ko_totals promoter_totals pop_regulated_s~
##    <fct> <fct>   <chr>                   <int>           <int>            <int>
##  1 embr~ activa~ bound                      92            1901              310
##  2 embr~ activa~ bound                      92            1901              310
##  3 embr~ repres~ bound                     122            1901              716
##  4 embr~ repres~ bound                     122            1901              716
##  5 embr~ indepe~ bound                    1687            1901             7509
##  6 embr~ indepe~ bound                    1687            1901             7509
##  7 L1    activa~ bound                     173            2241              310
##  8 L1    activa~ bound                     173            2241              310
##  9 L1    repres~ bound                     211            2241              716
## 10 L1    repres~ bound                     211            2241              716
## 11 L1    indepe~ bound                    1857            2241             7509
## 12 L1    indepe~ bound                    1857            2241             7509
## 13 L3    activa~ bound                     220            3543              310
## 14 L3    activa~ bound                     220            3543              310
## 15 L3    repres~ bound                     280            3543              716
## 16 L3    repres~ bound                     280            3543              716
## 17 L3    indepe~ bound                    3043            3543             7509
## 18 L3    indepe~ bound                    3043            3543             7509
## # ... with 6 more variables: pop_regulated_fail <dbl>, dhyper <dbl>,
## #   gene_type <chr>, genes <dbl>, padj <dbl>, star <chr>
```

```
bound_regulated_hyper_plot <- bound_regulated_hyper %>%
  ggplot(aes(x = stage, y = genes, fill = gene_type, label = star)) +
  geom_bar(position = position_dodge(width=0.75), stat = "identity", width = 0.5) +
  geom_text(vjust = 0, hjust = 0) +
  scale_fill_manual(values = c("grey", "black")) +
  # scale_y_continuous(breaks = seq(0, 4000, by = 500)) +
  theme_bw() +
  facet_wrap(elt2_ko~., scales = "free") +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black")) +
  ggtitle("Transcriptional regulation of\nELT-2 target genes")

bound_regulated_hyper_plot
```

```
## Warning: Removed 9 rows containing missing values (geom_text).
```




```
# ggsave(bound_regulated_hyper_plot, file = "../03_output/Gene_Set_Overlaps/bound_only_regulated_gene_c
```

Interpretation: Most genes do not respond to ELT-2 deletion, either bound or not bound. A subset of bound genes are up, and a subset are down - consider as direct regulated targets. A subset of genes not bound by ELT-2 also respond to ELT-2 deletion - consider as indirect targets. Higher transcript abundance in the absence of ELT-2 suggests repression or negative regulation. The large number negatively regulated bound genes is surprising, as ELT-2 is considered a transcriptional activator.

Question: Are bound and differentially expressed genes intestine enriched?

```
bound_intestine_regulated_genes <- all_stages_chip %>%
  filter(promoter_status == "bound", elt2_ko != "independent") %>%
  drop_na(elt2_ko) %>%
  group_by(stage, elt2_ko, intestine_expression) %>%
  summarise(observed = n()) %>%
  ungroup() %>%
  group_by(stage, elt2_ko) %>%
  mutate(elt2_ko_totals = sum(observed)) %>%
  ungroup() %>%
  group_by(stage, intestine_expression) %>%
  mutate(intestine_totals = sum(observed)) %>%
  ungroup() %>%
  # mutate(pop_bound_success = 541, pop_bound_fail = 1698) %>%
  left_join(all_stages_chip %>%
              group_by(stage, promoter_status) %>%
              summarise(pop = n()) %>%
              pivot_wider(names_from = promoter_status, values_from = pop) %>%
              rename(pop_bound_success = bound, pop_bound_fail = not_bound),
            by = "stage") %>%
  rowwise() %>%
  mutate(dhyper = dhyper(x = observed, k = elt2_ko_totals , m = pop_bound_success, n = pop_bound_fail),
         expected = elt2_ko_totals*pop_bound_success/(pop_bound_success+pop_bound_fail)) %>%
  pivot_longer(cols = c(observed, expected), names_to = "gene_type", values_to = "genes")%>%
  ungroup() %>%
  mutate(padj = p.adjust(dhyper, method = "BH")) %>%
  mutate(star = case_when(
    gene_type == "observed" & padj > 0.01 ~ "~",
    gene_type == "observed" & padj < 1*10^-10 ~ "***",
    gene_type == "observed" & padj < 1*10^-5 ~ "**",
    gene_type == "observed" & padj < 0.01 ~ "*"
```

```
                          ))
```

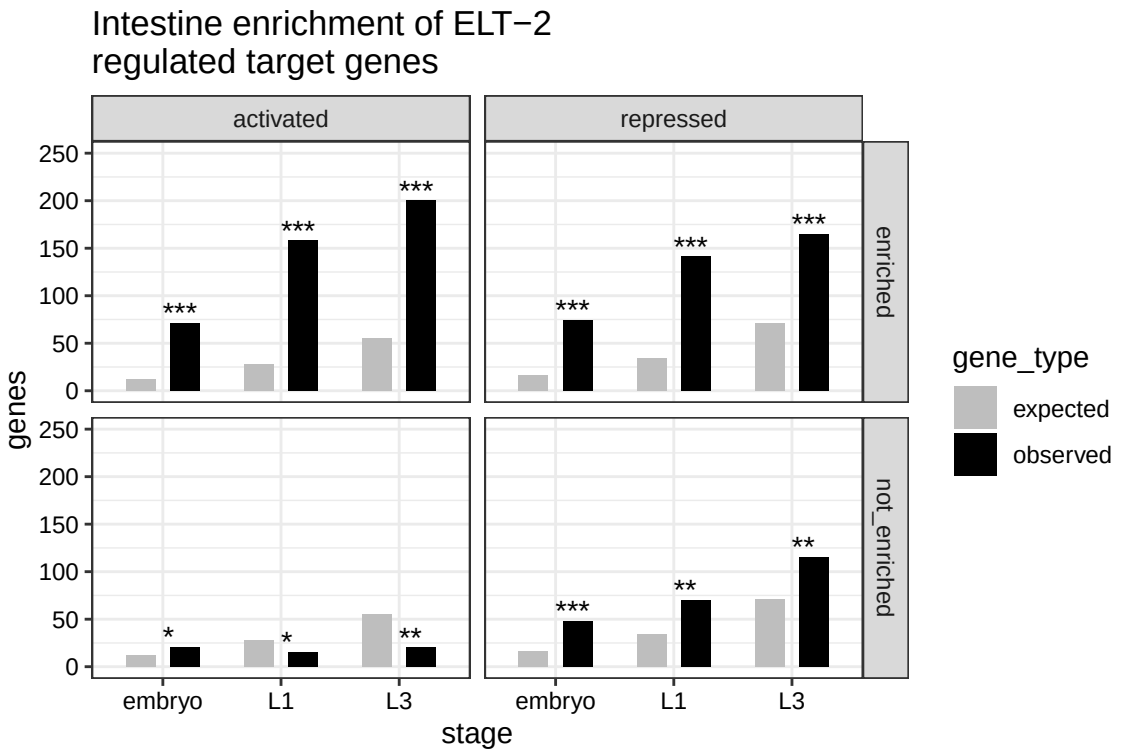
```
## `summarise()` has grouped output by 'stage', 'elt2_ko'. You can override using
## the `.groups` argument.
## `summarise()` has grouped output by 'stage'. You can override using the
## `.groups` argument.
```

```
bound_intestine_regulated_genes
```

```
## # A tibble: 24 x 12
##    stage  elt2_ko   intestine_expression elt2_ko_totals intestine_totals
##    <fct>  <fct>     <chr>                         <int>            <int>
##  1 embryo activated enriched                        92              145
##  2 embryo activated enriched                        92              145
##  3 embryo activated not_enriched                    92               69
##  4 embryo activated not_enriched                    92               69
##  5 embryo repressed enriched                       122              145
##  6 embryo repressed enriched                       122              145
##  7 embryo repressed not_enriched                   122               69
##  8 embryo repressed not_enriched                   122               69
##  9 L1     activated enriched                       173              299
## 10 L1     activated enriched                       173              299
## # ... with 14 more rows, and 7 more variables: pop_bound_success <int>,
## #   pop_bound_fail <int>, dhyper <dbl>, gene_type <chr>, genes <dbl>,
## #   padj <dbl>, star <chr>
```

```
bound_intestine_regulated_genes_plot <- bound_intestine_regulated_genes %>%
  ggplot(aes(x = stage, y = genes, fill = gene_type, label = star)) +
  geom_bar(position = position_dodge(width=0.75), stat = "identity", width = 0.5) +
  geom_text(vjust = 0, hjust = 0) +
  scale_fill_manual(values = c("grey", "black")) +
  theme_bw() +
  ylim(c(0,max(bound_intestine_regulated_genes$genes)+50)) +
  facet_grid(intestine_expression~elt2_ko) +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black")) +
  ggtitle("Intestine enrichment of ELT-2\nregulated target genes")

bound_intestine_regulated_genes_plot
```

```
## Warning: Removed 12 rows containing missing values (geom_text).
```




```
# ggsave(bound_intestine_regulated_genes_plot, file = "../03_output/Gene_Set_Overlaps/bound_intestine_r
```

Interpretation: Repressed target genes are representative of both intestine enriched and not intestine enriched genes. Activated target genes are primarily intestine enriched. Targets independent of ELT-2 regulation are primarily not intestine enriched.

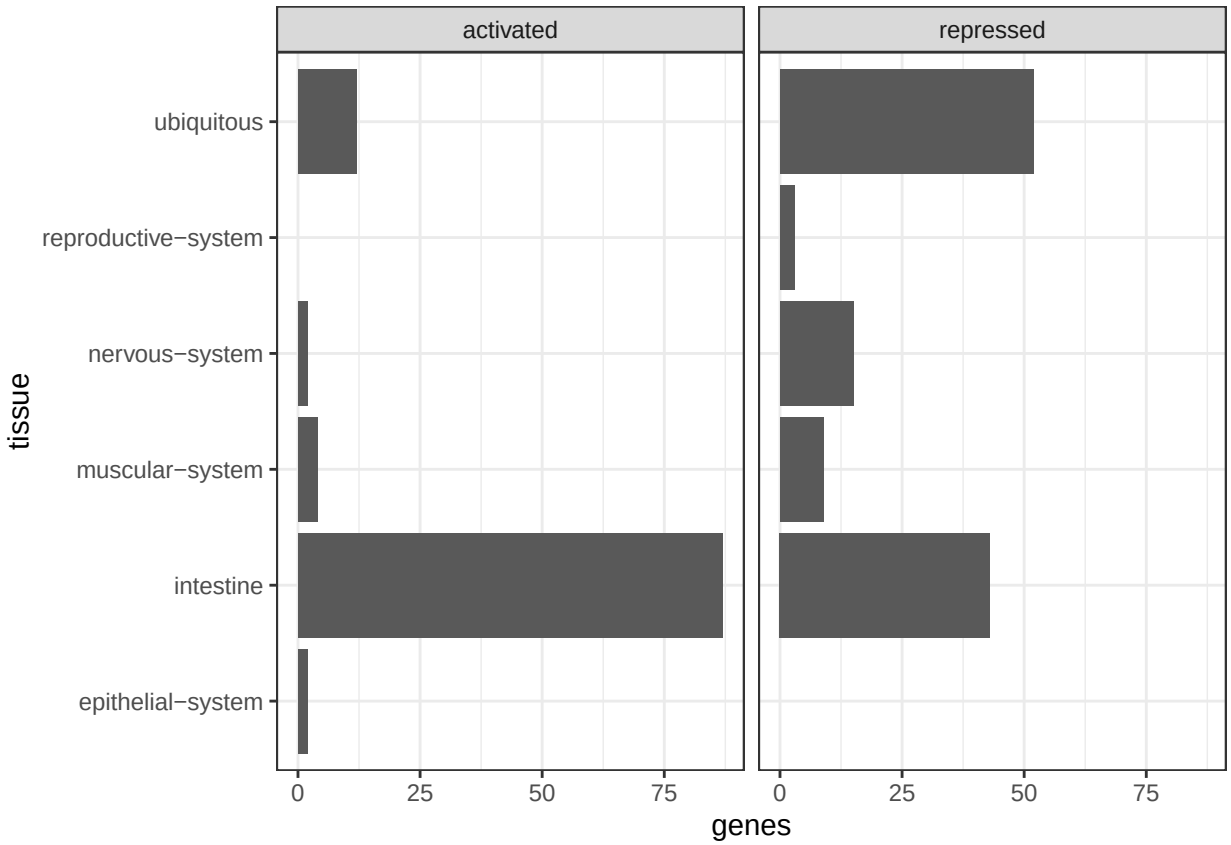
Question: what tissues are ELT-2 regulated genes expressed in?

```
tissue_specific_genes <- read_csv(file = "../../01_tissue_specific_genes/03_output/tissue_specific_genes
```

```
## Rows: 6873 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): WBGeneID, Sequence.name, tissue
##
## 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.
```

```
all_stages_chip %>% filter(promoter_status == "bound", elt2_ko != "independent") %>% inner_join(tissue_s
  group_by(tissue, elt2_ko) %>% summarise(genes = n()) %>%
  ggplot(aes(x = tissue, y = genes)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_bw() +
  facet_grid(.~elt2_ko)
```

```
## `summarise()` has grouped output by 'tissue'. You can override using the
## `.groups` argument.
```



Interpretation: ELT-2 repressed targets are expressed in both intestine and reproductive, nervous, and muscular systems. ELT-2 activated targets are primarily intestine genes with some muscle and epithelial genes.

Calculate hypergeometric statistc

```
# K: number of balls drawn, number of genes that are bound and regulated
# X: number of bound and regulated genes that are associated with a given tissue term
# M: total number of genes associated with a given tissue term
# N: total number of genes associated with any tissue term minus M
bound_regulated_tissue_hyper <- all_stages_chip %>% left_join(tissue_specific_genes, by = "WBGeneID") %
  filter(elt2_ko != "independent", promoter_status == "bound") %>%
  group_by(promoter_status, elt2_ko, tissue) %>%
  summarise(bound_tissue_regulated_total = n()) %>%
  ungroup() %>%
  bind_rows(tribble(~promoter_status, ~elt2_ko, ~tissue, ~bound_tissue_regulated_total,
                    "bound", "activated", "reproductive-system", 0,
                    "bound", "repressed", "epithelial-system", 0)) %>%
  left_join(all_stages_chip %>%
              filter(elt2_ko != "independent", promoter_status == "bound") %>%
              group_by(elt2_ko) %>%
              summarise(bound_reg_total = n()),
            by = c("elt2_ko")) %>%
  left_join(tissue_specific_genes %>%
              group_by(tissue) %>%
              summarise(tissue_success = n()) %>%
```

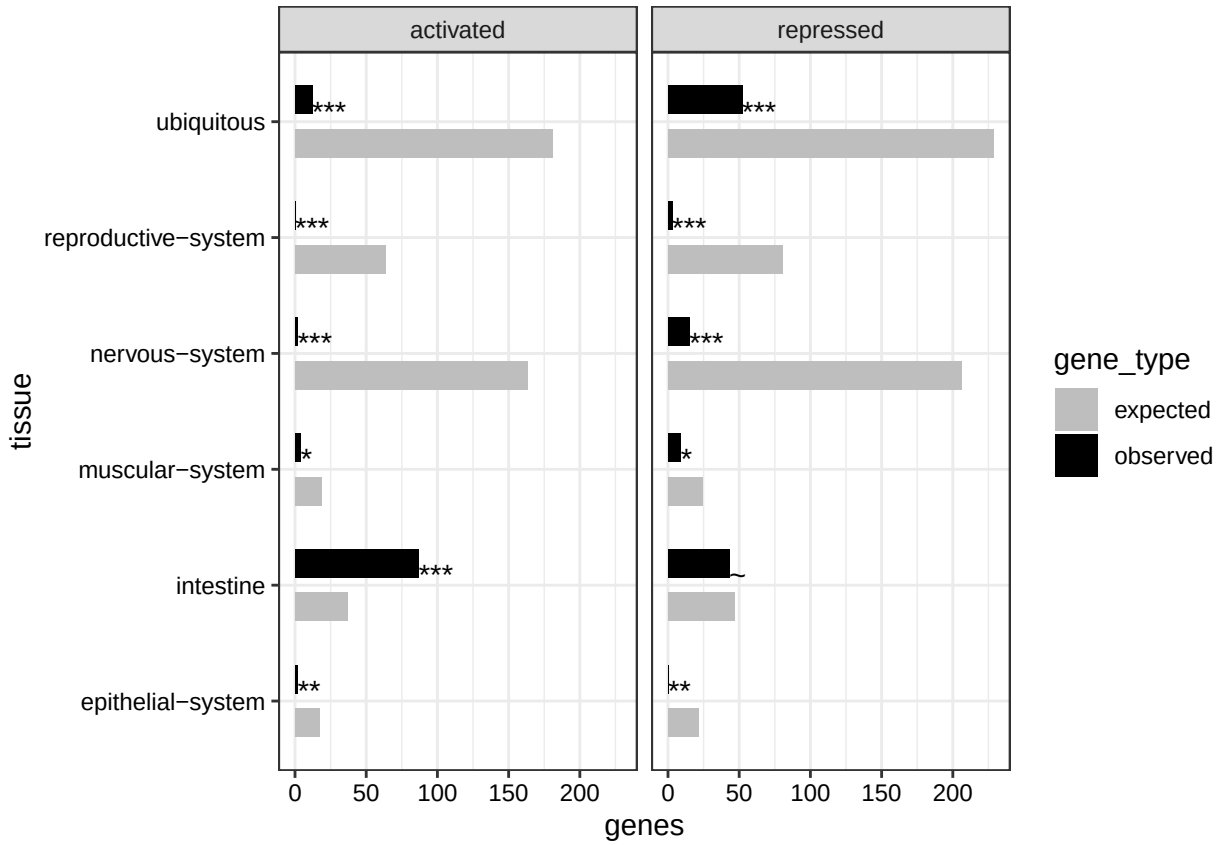
```
              rowwise() %>%
              mutate(tissue_total = nrow(tissue_specific_genes), tissue_fail = tissue_total - tissue_su
            by = "tissue") %>%
  drop_na(tissue) %>%
  mutate(dhyper = dhyper(x = bound_tissue_regulated_total, k = bound_reg_total , m = tissue_success, n =
         expected = bound_reg_total*tissue_success/(tissue_success+tissue_fail)) %>%
  mutate(padj = p.adjust(dhyper, method = "BH"))
```

```
## `summarise()` has grouped output by 'promoter_status', 'elt2_ko'. You can
## override using the `.groups` argument.
```

```
bound_regulated_tissue_hyper <- bound_regulated_tissue_hyper %>%
  rename(observed = bound_tissue_regulated_total) %>%
  pivot_longer(cols = c(observed, expected), names_to = "gene_type", values_to = "genes") %>%
  mutate(star = case_when(
    gene_type == "observed" & padj > 0.01 ~ "~",
    gene_type == "observed" & padj < 1*10^-10 ~ "***",
    gene_type == "observed" & padj < 1*10^-5 ~ "**",
    gene_type == "observed" & padj < 0.01 ~ "*"
                          ))
```

```
bound_regulated_tissue_hyper_plot <- bound_regulated_tissue_hyper %>%
ggplot(aes(x = tissue, y = genes, fill = gene_type, label = star)) +
  geom_bar(position = position_dodge(width=0.75), stat = "identity", width = 0.5) +
  scale_fill_manual(values = c("grey", "black")) +
  geom_text(hjust = 0, vjust = 0)+
  facet_grid(~elt2_ko) +
  theme_bw() +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black")) +
  coord_flip()
bound_regulated_tissue_hyper_plot
```

```
## Warning: Removed 12 rows containing missing values (geom_text).
```

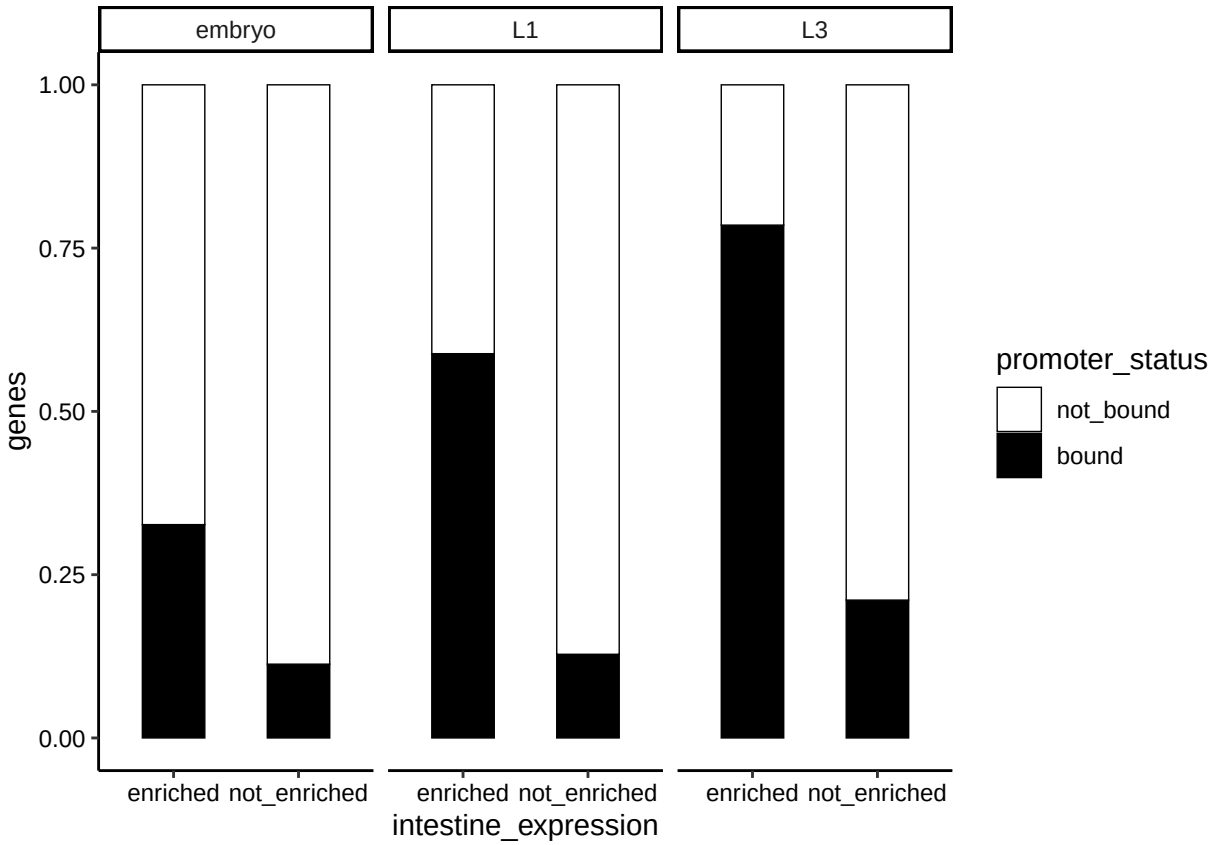



```
# ggsave(bound_regulated_tissue_hyper_plot, file = "../03_output/Gene_Set_Overlaps/bound_regulated_tiss
```

Stacked bar plots

Question: What is the fraction of ELT-2 bound enriched genes?

```
fraction_bound_stack <- all_stages_chip %>%
  group_by(stage, intestine_expression, promoter_status) %>%
  summarise(genes = n(), .groups = "keep") %>%
  mutate(promoter_status = factor(promoter_status, levels = c("not_bound", "bound"))) %>%
  ggplot(aes(x = intestine_expression, y = genes, fill = promoter_status)) +
  geom_bar(stat = "identity", position = "fill", color = "black", width = 0.5, size = 0.25) +
  scale_fill_manual(values = c("white", "black")) +
  theme_classic() +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black")) +
  facet_wrap(~stage)

fraction_bound_stack
```




```
# ggsave(fraction_bound_stack, file = "../03_output/Gene_Set_Overlaps/fraction_bound_stack.pdf", width =
```

```
q1_contingency_df <-
  all_stages_chip %>% group_by(stage, intestine_expression, promoter_status) %>% summarise(genes = n())
  ungroup()  %>%
  pivot_wider(values_from = genes, names_from = promoter_status)
```

```
## `summarise()` has grouped output by 'stage', 'intestine_expression'. You can
## override using the `.groups` argument.
```

```
q1_contingency_df
```

```
## # A tibble: 6 x 4
##   stage  intestine_expression bound not_bound
##   <fct>  <chr>                <int>     <int>
## 1 embryo enriched               567      1169
## 2 embryo not_enriched          2063     16196
## 3 L1     enriched               869       608
## 4 L1     not_enriched          2372     16146
## 5 L3     enriched              1126       308
## 6 L3     not_enriched          3918     14643
```

```
q1_pval_df <- data.frame()
for(i in unique(q1_contingency_df$stage)){
  # print(i)
  chi = chisq.test(q1_contingency_df %>% filter(stage == i) %>% select(-stage, -intestine_expression))
  rownames(chi$observed) <- c("enriched", "not_enriched")
  rownames(chi$expected) <- c("enriched", "not_enriched")
  q1_pval <- chi$p.value
```

```
  q1_expected <- chi$expected
  print(c(i, q1_pval))
  q1_pval_df <-q1_pval_df %>% bind_rows(data.frame(stage = i,
                                    chi.pval = q1_pval,
                                    expected.bound.enriched = chi$expected['enriched','bound'],
                                    expected.not_bound.enriched = chi$expected['enriched',
                                                                               'not_bound'],
                                    expected.bound.not_enriched = chi$expected['not_enriched',
                                                                               'bound'],
                                    expected.not_bound.not_enriched =
                                      chi$expected['not_enriched',
                                                   'not_bound']))
}
```

```
## [1] "embryo"                "2.40101360103356e-139"
## [1] "L1" "0"
## [1] "L3" "0"
```

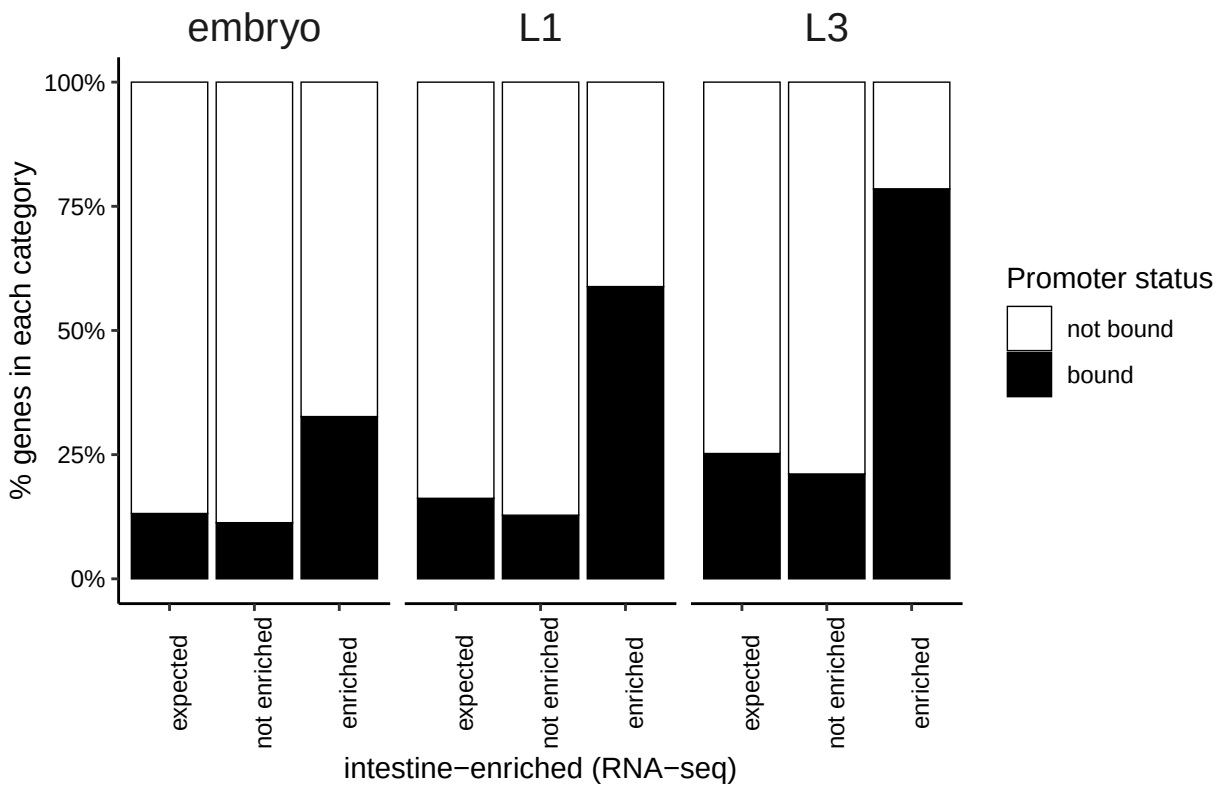
```
q1_table <- q1_contingency_df %>%
  rowwise() %>%
  mutate(total = bound + not_bound,
         percent_bound = (100*bound)/(bound+not_bound),
         percent_not_bound = (100*not_bound)/(bound+not_bound)) %>%
  left_join(q1_pval_df %>% mutate(chi.padj = p.adjust(chi.pval, method = "bonferroni")), by = "stage") %
                          expected.bound.enriched,
                          expected.bound.not_enriched),
           not_bound.expected = ifelse(intestine_expression == "enriched",
                          expected.not_bound.enriched,
                          expected.not_bound.not_enriched)) %>% select(-starts_with("expected"))  %>%
  mutate(fraction.bound.expected = bound.expected / (bound+not_bound),
         fraction.not_bound.expected = not_bound.expected / (bound+not_bound),
         )

q1_table
```

```
## # A tibble: 6 x 13
## # Rowwise:
##   stage  intestine_express~ bound not_bound total percent_bound percent_not_bou~
##   <chr>  <chr>              <int>     <int> <int>         <dbl>            <dbl>
## 1 embryo enriched             567      1169  1736          32.7             67.3
## 2 embryo not_enriched        2063     16196 18259          11.3             88.7
## 3 L1     enriched             869       608  1477          58.8             41.2
## 4 L1     not_enriched        2372     16146 18518          12.8             87.2
## 5 L3     enriched            1126       308  1434          78.5             21.5
## 6 L3     not_enriched        3918     14643 18561          21.1             78.9
## # ... with 6 more variables: chi.pval <dbl>, chi.padj <dbl>,
## #   bound.expected <dbl>, not_bound.expected <dbl>,
## #   fraction.bound.expected <dbl>, fraction.not_bound.expected <dbl>
```

```
like_robs_stacked_data = q1_table %>% pivot_longer(c("bound","not_bound"), names_to = "promoter_status"

# transfer the expected columns to their own rows
expected_rows = q1_table %>%
  select(stage, intestine_expression, bound.expected, not_bound.expected) %>%
  filter(intestine_expression == "enriched") %>%
```

```
  dplyr::rename(bound=bound.expected,
                not_bound=not_bound.expected) %>%

  pivot_longer(cols=c("bound","not_bound"),
               values_to = "genes",
               names_to = "promoter_status") %>%
  mutate(intestine_expression = "expected")

like_robs_stacked_data %<>% rbind(expected_rows) %>%
mutate(promoter_status = factor(promoter_status, levels = c("not_bound", "bound"))) %>%
  mutate(intestine_expression = factor(intestine_expression, levels = c("expected",
                                                                        "not_enriched",
                                                                        "enriched"))) %>%
  mutate(obs.exp = ifelse(intestine_expression=="expected", "expected", "observed"))

p1 = ggplot(like_robs_stacked_data, aes(x = intestine_expression,
                                    y = genes,
                                    fill = promoter_status
                                    #,alpha=obs.exp # will override to .5 for expected, else 1
                                    )
       ) +
  geom_bar(stat = "identity",
           position = "fill",
           color = "black",
           width = 0.9,
           size = 0.25) +
  scale_fill_manual(values = c("white", "black"), labels=c("not bound", "bound")) +
  #scale_alpha_manual(values = c(.5,1)) +
  facet_wrap(~stage) +
theme_classic() +
  theme(axis.text.x=element_text(colour="black", angle=90),
        axis.text.y=element_text(colour="black"),
        strip.background = element_blank(),
        strip.text = element_text(size=15))  +
  guides(
    fill=guide_legend(title="Promoter status"),
    alpha="none") +
  ggtitle("Figure 4C") +
  xlab("intestine-enriched (RNA-seq)") +
  ylab("% genes in each category") +
  scale_y_continuous(labels = scales::percent_format(scale = 100)) +
  scale_x_discrete(labels = c("expected", "not enriched", "enriched"))


p1
```

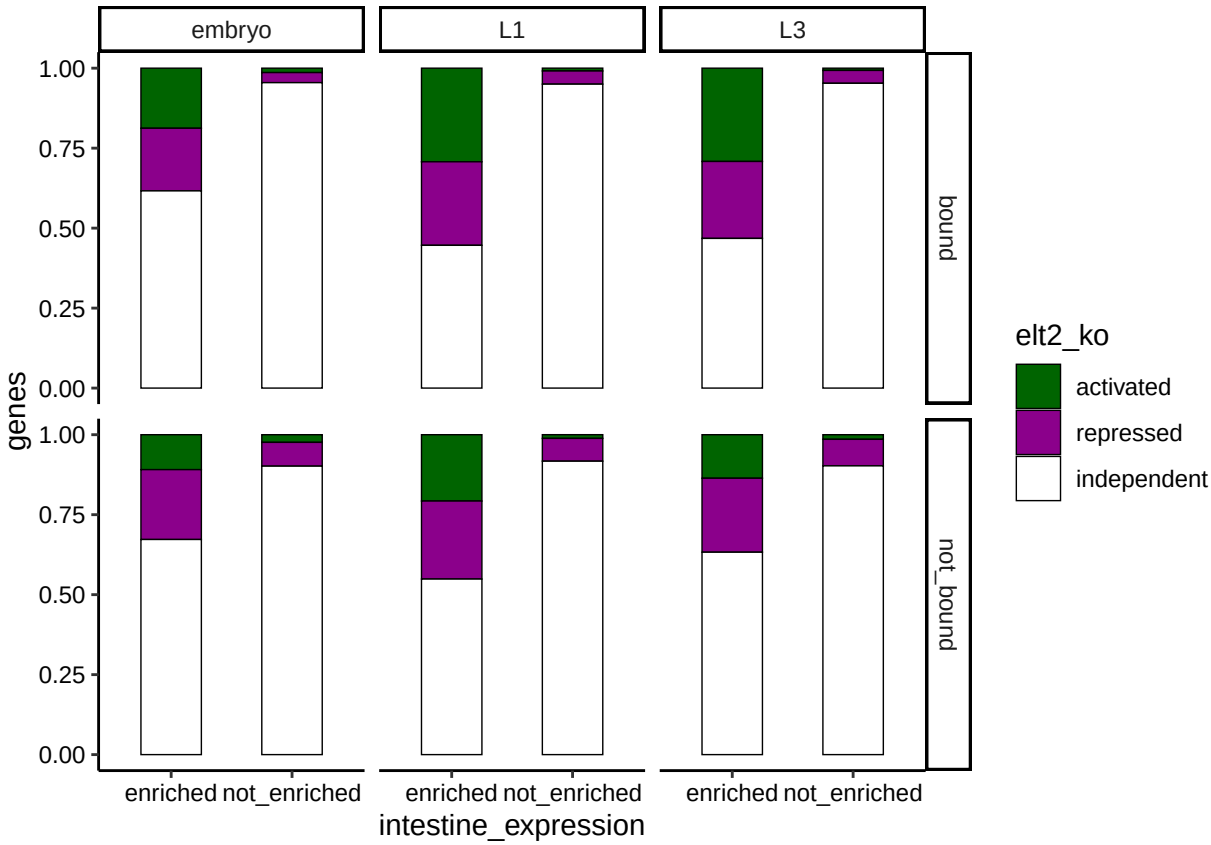
Figure 4C




```
# pdf(file=file.path(david_dir_plots,"fig4_C_fraction_bound_stack.pdf"),height=5,width=7, family="Arial
# p1
# dev.off()
```

Question: What is the fraction of transcriptionally dependent ELT-2 targets?

```
fraction_regulated_targets <- all_stages_chip %>%
  group_by(stage, intestine_expression, promoter_status, elt2_ko) %>%
  summarise(genes = n(), .groups = "keep") %>%
  drop_na(elt2_ko) %>%
  ggplot(aes(x = intestine_expression, y = genes, fill = elt2_ko)) +
  geom_bar(stat = "identity", position = "fill", color = "black", width = 0.5, size = 0.25) +
  scale_fill_manual(values = c("darkgreen", "darkmagenta", "white")) +
  facet_grid(promoter_status~stage) +
  theme_classic() +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black"))
fraction_regulated_targets
```




```
# ggsave(fraction_regulated_targets, file = "../03_output/Gene_Set_Overlaps/fraction_regulated_targets_
```

```
q2_contingency_df <-
  all_stages_chip %>% group_by(stage, intestine_expression, promoter_status, elt2_ko) %>% summarise(gene
  ungroup()  %>%
  pivot_wider(values_from = genes, names_from = elt2_ko)
```

```
## `summarise()` has grouped output by 'stage', 'intestine_expression',
## 'promoter_status'. You can override using the `.groups` argument.
```

```
q2_pval_df <- data.frame()
for(i in unique(q2_contingency_df$stage)){
  # print(i)
  for(j in c("bound", "not_bound")){
    q2_pval <- chisq.test(q2_contingency_df %>% filter(stage == i, promoter_status == j) %>% select(-(st
    print(c(i, j,q2_pval))
    q2_pval_df <-q2_pval_df %>% bind_rows(data.frame(stage = i, promoter_status = j, chi.pval = q2_pval)
  }
}
```

```
## [1] "embryo"              "bound"               "5.20609352504466e-78"
## [1] "embryo"              "not_bound"           "4.46875763938592e-70"
## [1] "L1"                   "bound"                "1.46622850640383e-166"
## [1] "L1"                   "not_bound"            "7.22871665191411e-155"
## [1] "L3"                   "bound"                "3.42975115374516e-251"
```

```
## Warning in chisq.test(q2_contingency_df %>% filter(stage == i, promoter_status
## == : Chi-squared approximation may be incorrect
```

```
## [1] "L3"                    "not_bound"            "1.82664157622379e-36"
```

```
q2_pval_df
```

```
##    stage promoter_status      chi.pval
## 1 embryo           bound  5.206094e-78
## 2 embryo       not_bound  4.468758e-70
## 3     L1           bound 1.466229e-166
## 4     L1       not_bound 7.228717e-155
## 5     L3           bound 3.429751e-251
## 6     L3       not_bound  1.826642e-36
```

```
allchisq.list=lapply(split(q2_contingency_df, q2_contingency_df$stage),
       function(X){
         lapply(
           split(X, X$promoter_status), function(Z){
              chi.obj = chisq.test(Z %>% select(-(stage:promoter_status)))
              observed = data.frame(chi.obj$observed,
                                    stage=Z$stage,
                                    promoter_status=Z$promoter_status,
                                    intestine_expression=Z$intestine_expression,
                                    value_type="observed")
              expected = data.frame(chi.obj$expected,
                                    stage=Z$stage,
                                    promoter_status=Z$promoter_status,
                                    intestine_expression=Z$intestine_expression,
                                    value_type="expected")
              ep = rev(cumsum(rev(chi.obj$expected))/sum(chi.obj$expected))
              ep.df <- data.frame(activated=ep[1],repressed=ep[2],independent=ep[3])
              expected_proportion =
                data.frame(ep.df,
                           stage=Z$stage,
                           promoter_status=Z$promoter_status,
                           intestine_expression=Z$intestine_expression,
                           value_type="expected_proportion")

              bind_rows(observed, expected,expected_proportion)
           }
         )
})
```

```
## Warning in chisq.test(Z %>% select(-(stage:promoter_status))): Chi-squared
## approximation may be incorrect
```

```
allchisq.df = allchisq.list$embryo %>% bind_rows() %>%
  rbind(allchisq.list$L1 %>% bind_rows()) %>%
  rbind(allchisq.list$L3 %>% bind_rows()) %>% pivot_longer(cols=c("activated","repressed","independent"

a = allchisq.df %>% filter(value_type == "observed")
b = allchisq.df %>% filter(value_type == "expected" & intestine_expression == "enriched") %>% mutate(in
c = rbind(a,b)
table(c$intestine_expression, c$value_type)
```

```
##
##                expected observed
##   enriched            0       18
```

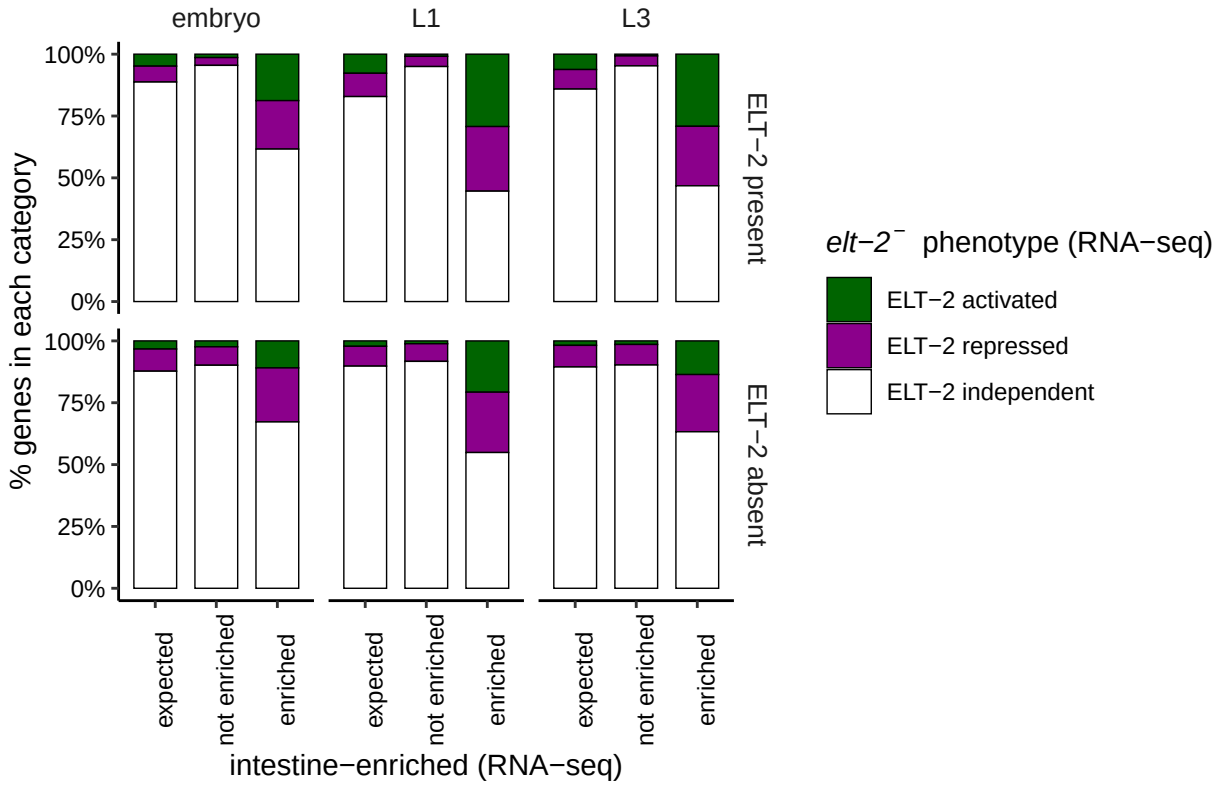
```
##   expected           18        0
##   not_enriched        0       18
```

```
allchisq.df %<>%
  # make an "expected" class in intestine_expression (already a combination of intestine_expression and
  mutate(intestine_expression = ifelse(
    value_type == 'expected' & intestine_expression == 'enriched',
    'expected',
    intestine_expression)) %>%
  # drop the superfluous data
  filter(intestine_expression != "not_enriched" & value_type != "expected") #%>% select(-value_type)

table(allchisq.df$intestine_expression, allchisq.df$value_type)
```

```
##
##            expected_proportion observed
##   enriched                  18       18
```

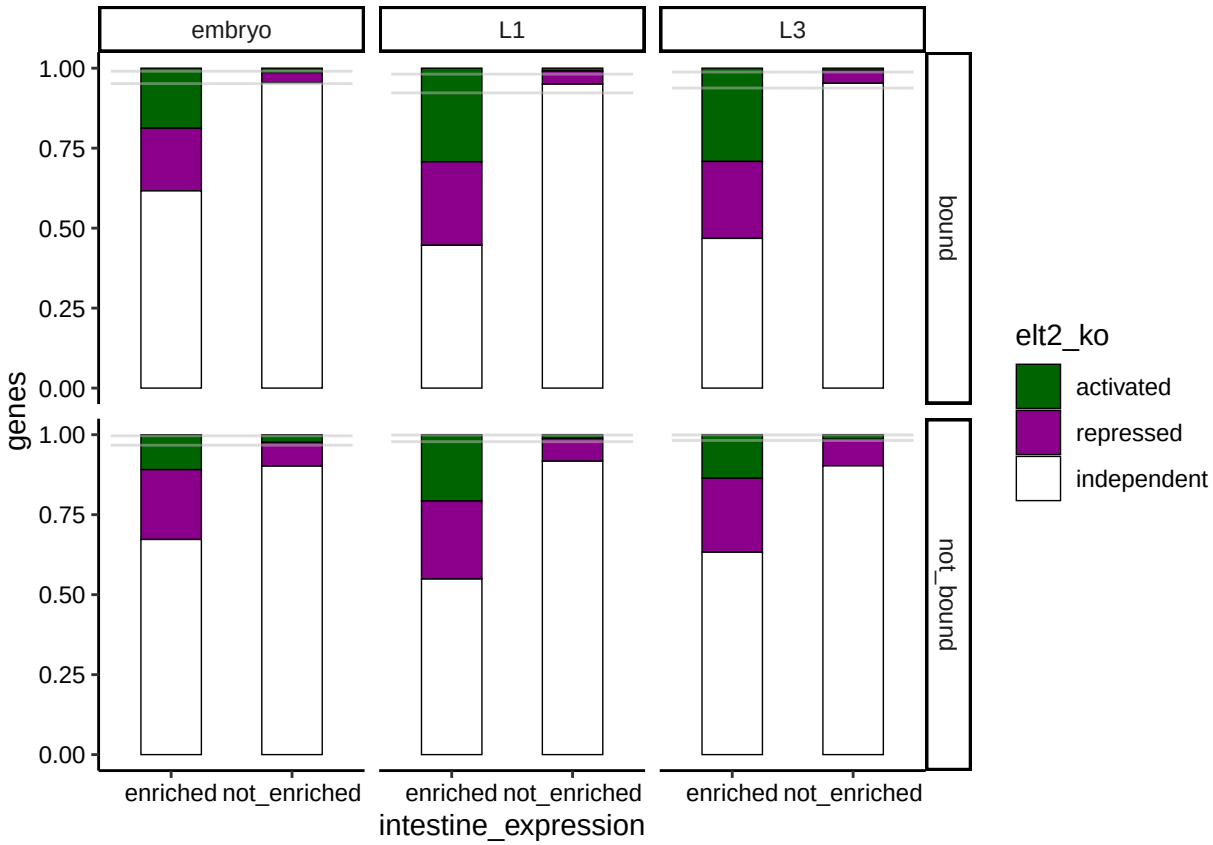
```
allchisq.df$elt2_ko = factor(allchisq.df$elt2_ko, levels=c("activated","repressed","independent"))

c$elt2_ko = factor(c$elt2_ko, levels=c("activated","repressed","independent"))
c$intestine_expression = factor(c$intestine_expression,
                                levels=c("expected","not_enriched","enriched"))


p2 = ggplot(c, aes(x = intestine_expression,
                   y = genes,
                #    alpha = intestine_expression,
                   fill = elt2_ko)) +
  geom_bar(stat = "identity",
           position = "fill",
           color = "black",
           width = .7,
           size = 0.25) +
  scale_fill_manual(values = c("darkgreen", "darkmagenta", "white"), labels=c("ELT-2 activated",
                                                                               "ELT-2 repressed",
                                                                               "ELT-2 independent")) +
  #scale_alpha_manual(values = c(.5,1,1)) +
  facet_grid(promoter_status~stage, labeller = labeller(.rows = c("not_bound" = "ELT-2 absent",
         "bound" = "ELT-2 present"))) +
  theme_classic() +
  theme(axis.text.x=element_text(colour="black", angle=90),
        axis.text.y=element_text(colour="black"),
        strip.background = element_blank(),
        strip.text = element_text(size=10)) +
  guides(alpha="none",
         fill=guide_legend(title=expression(paste(italic("elt-2")^"-", "  phenotype (RNA-seq)")))) +
  ggtitle("Figure 4D") +
  xlab("intestine-enriched (RNA-seq)") +
  ylab("% genes in each category") +
  scale_y_continuous(labels = scales::percent_format(scale = 100))+
  scale_x_discrete(labels = c("expected", "not enriched", "enriched"))

p2 # ggplot output variable
```

Figure 4D




```
# pdf(file=file.path(david_dir_plots,"fig4_D_fraction_bound_stack_KO.pdf"),height=5,width=7, family="Ar
# p2
# dev.off()
```

```
fraction_regulated_targets <- all_stages_chip %>%
  group_by(stage, intestine_expression, promoter_status, elt2_ko) %>%
  summarise(genes = n(), .groups = "keep") %>%
  drop_na(elt2_ko) %>%

  ggplot(aes(x = intestine_expression, y = genes, fill = elt2_ko)) +
  geom_bar(stat = "identity", position = "fill", color = "black", width = 0.5, size = 0.25) +
  scale_fill_manual(values = c("darkgreen", "darkmagenta", "white")) +

  facet_grid(promoter_status~stage) +
  #coord_cartesian(ylim=c(.5,1)) +
  theme_classic() +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black")) +
  geom_segment(allchisq.df %>% filter(genes != 1 & value_type == "expected_proportion"),
               inherit.aes = FALSE,
               mapping=aes(x=.5,xend=2.5, y=genes, yend=genes),
               color="grey",
               alpha=.5)


fraction_regulated_targets
```



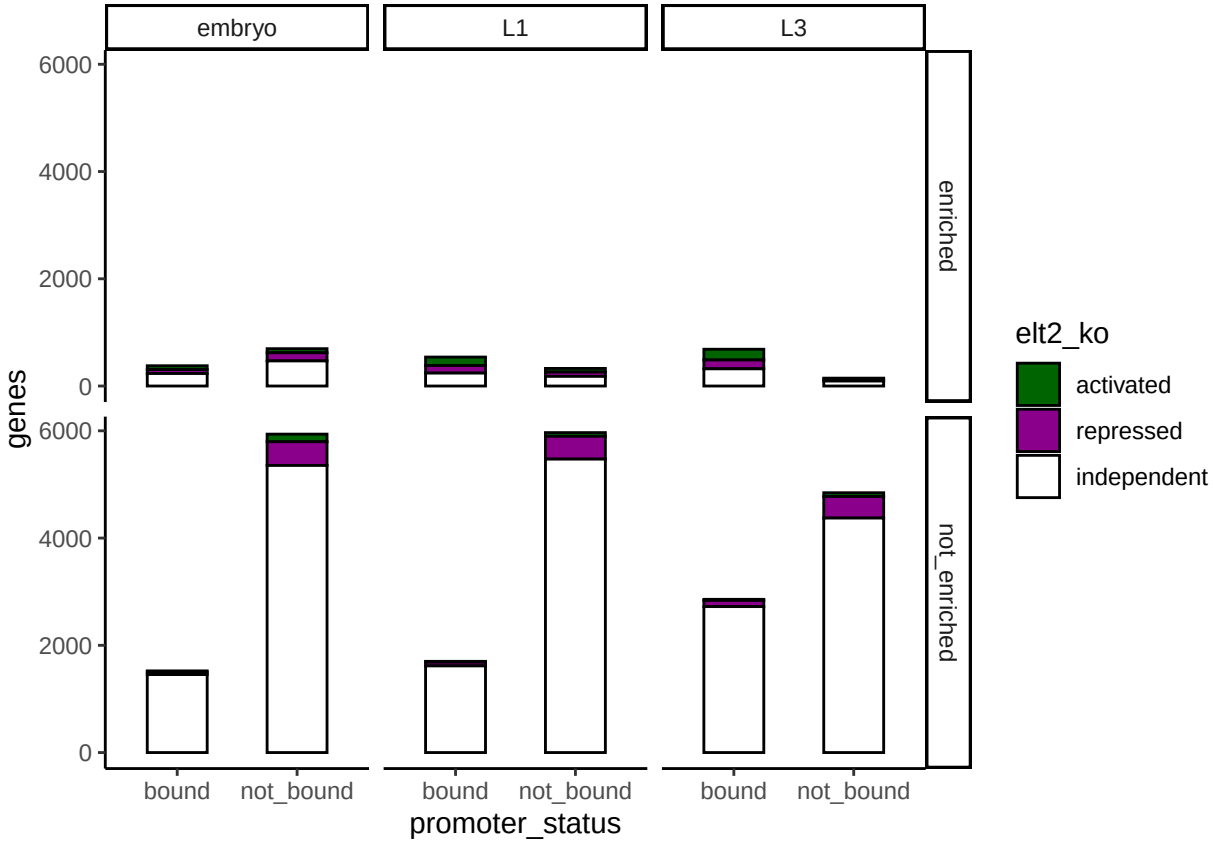

```
q2_table <- q2_contingency_df %>%
  rowwise() %>%
  mutate(total = activated+repressed+independent,
    percent_activated = (100*activated)/(activated+repressed+independent),
         percent_repressed = (100*repressed)/(activated+repressed+independent),
         percent_independent = (100*independent)/(activated+repressed+independent)) %>%
  left_join(q2_pval_df %>% mutate(chi.padj = p.adjust(chi.pval, method = "bonferroni")), by = c("stage"

q2_table
```

```
## # A tibble: 12 x 12
## # Rowwise:
##    stage  intestine_expre~ promoter_status activated repressed independent total
##    <chr>  <chr>            <chr>                <int>     <int>       <int> <int>
##  1 embryo enriched         bound                   71        74         233   378
##  2 embryo enriched         not_bound               76       152         468   696
##  3 embryo not_enriched     bound                   21        48        1454  1523
##  4 embryo not_enriched     not_bound              142       442        5354  5938
##  5 L1     enriched         bound                  158       141         241   540
##  6 L1     enriched         not_bound               68        80         180   328
##  7 L1     not_enriched     bound                   15        70        1616  1701
##  8 L1     not_enriched     not_bound               69       425        5472  5966
##  9 L3     enriched         bound                  200       165         321   686
## 10 L3     enriched         not_bound               20        34          93   147
## 11 L3     not_enriched     bound                   20       115        2722  2857
## 12 L3     not_enriched     not_bound               70       402        4373  4845
## # ... with 5 more variables: percent_activated <dbl>, percent_repressed <dbl>,
```

```
## #   percent_independent <dbl>, chi.pval <dbl>, chi.padj <dbl>
```

```
all_stages_chip %>% group_by(stage, intestine_expression, promoter_status, elt2_ko) %>% summarise(genes
  ggplot(aes(x = promoter_status, y = genes, fill = elt2_ko, width = 0.5)) +
  geom_bar(stat = "identity", position = "stack", color = "black") +
  scale_fill_manual(values = c("darkgreen", "darkmagenta", "white")) +
  facet_grid(intestine_expression~stage) +
  theme_classic()
```

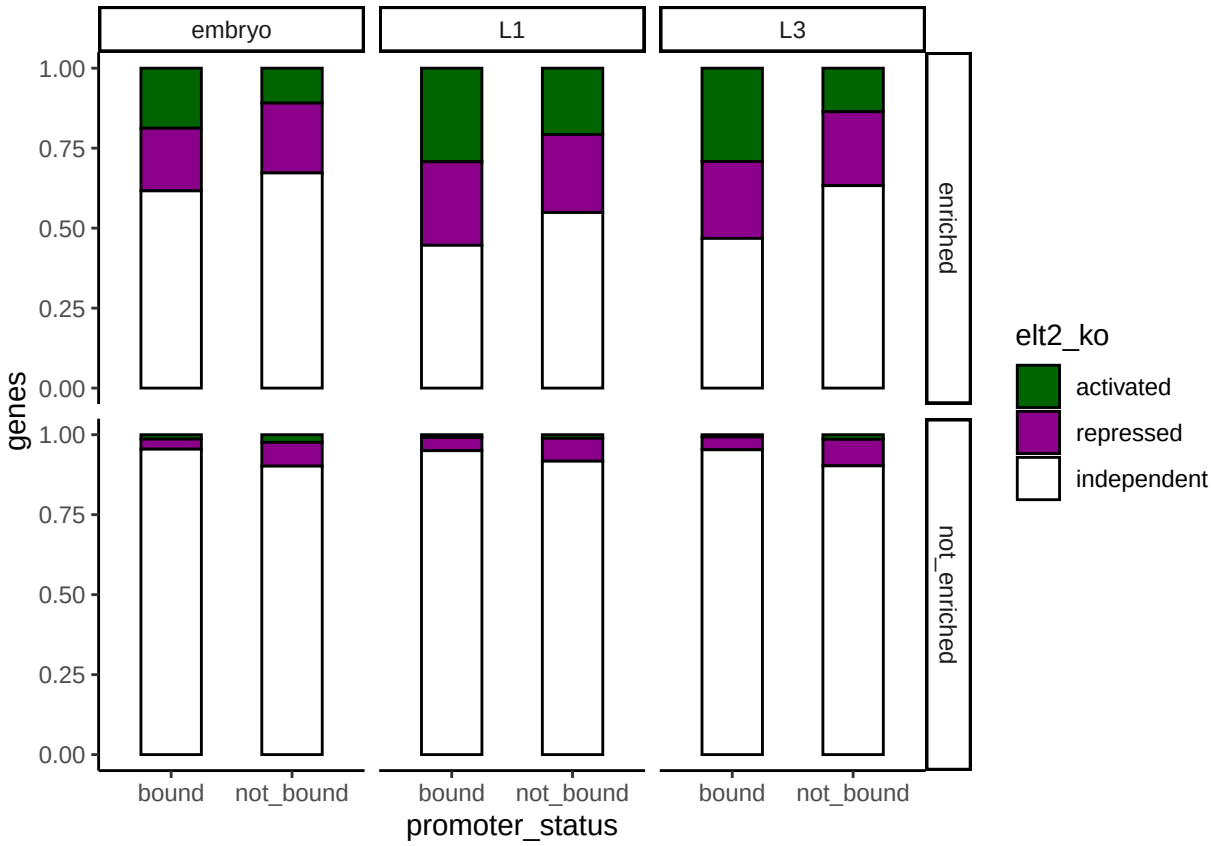
```
## `summarise()` has grouped output by 'stage', 'intestine_expression',
## 'promoter_status'. You can override using the `.groups` argument.
```




```
# +
 # scale_y_log10()
```

```
all_stages_chip %>% group_by(stage, intestine_expression, promoter_status, elt2_ko) %>% summarise(genes
  ggplot(aes(x = promoter_status, y = genes, fill = elt2_ko, width = 0.5)) +
  geom_bar(stat = "identity", position = "fill", color = "black") +
  scale_fill_manual(values = c("darkgreen", "darkmagenta", "white")) +
  facet_grid(intestine_expression~stage) +
  theme_classic()
```

```
## `summarise()` has grouped output by 'stage', 'intestine_expression',
## 'promoter_status'. You can override using the `.groups` argument.
```




```
chisq.df <- data.frame()
for(i in unique(q2_contingency_df$stage)){
  # print(i)
  for(j in c("enriched", "not_enriched")){
    pval <- chisq.test(q2_contingency_df %>% filter(stage == i, intestine_expression == j) %>% select(-
    chisq.df <- bind_rows(chisq.df, data.frame(stage = i, intestine_expression = j, pval = pval))
  }
}

chisq.df  %>% mutate(padj = p.adjust(pval, method = "bonferroni")) %>% left_join(q2_contingency_df, by =
```

```
##     stage intestine_expression         pval         padj promoter_status
## 1  embryo             enriched 1.628674e-03 9.772043e-03           bound
## 2  embryo             enriched 1.628674e-03 9.772043e-03       not_bound
## 3  embryo         not_enriched 3.722337e-10 2.233402e-09           bound
## 4  embryo         not_enriched 3.722337e-10 2.233402e-09       not_bound
## 5      L1             enriched 5.641504e-03 3.384902e-02           bound
## 6      L1             enriched 5.641504e-03 3.384902e-02       not_bound
## 7      L1         not_enriched 2.738950e-05 1.643370e-04           bound
## 8      L1         not_enriched 2.738950e-05 1.643370e-04       not_bound
## 9      L3             enriched 1.409186e-04 8.455115e-04           bound
## 10     L3             enriched 1.409186e-04 8.455115e-04       not_bound
## 11     L3         not_enriched 2.957413e-14 1.774448e-13           bound
## 12     L3         not_enriched 2.957413e-14 1.774448e-13       not_bound
##    activated repressed independent  sig
## 1         71        74         233 TRUE
## 2         76       152         468 TRUE
```

```
## 3         21        48        1454 TRUE
## 4        142       442        5354 TRUE
## 5        158       141         241 TRUE
## 6         68        80         180 TRUE
## 7         15        70        1616 TRUE
## 8         69       425        5472 TRUE
## 9        200       165         321 TRUE
## 10        20        34          93 TRUE
## 11        20       115        2722 TRUE
## 12        70       402        4373 TRUE
```

```
chisq.test(q2_contingency_df %>% filter(stage == "embryo", intestine_expression == "not_enriched") %>% s
```

```
##
##  Pearson's Chi-squared test
##
## data:  q2_contingency_df %>% filter(stage == "embryo", intestine_expression ==     "not_enriched") %
## X-squared = 43.423, df = 2, p-value = 3.722e-10
```

```
all_stages_chip %>% group_by(stage) %>% summarise(n())
```

```
## # A tibble: 3 x 2
##   stage  `n()`
##   <fct>  <int>
## 1 embryo 19995
## 2 L1     19995
## 3 L3     19995
```

```
all_stages_chip %>% group_by(stage, intestine_expression) %>% summarise(n())
```

```
## `summarise()` has grouped output by 'stage'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 6 x 3
## # Groups:   stage [3]
##   stage  intestine_expression `n()`
##   <fct>  <chr>                <int>
## 1 embryo enriched              1736
## 2 embryo not_enriched         18259
## 3 L1     enriched              1477
## 4 L1     not_enriched         18518
## 5 L3     enriched              1434
## 6 L3     not_enriched         18561
```

```
all_stages_chip %>% group_by(stage, intestine_expression, promoter_status) %>% summarise(n())
```

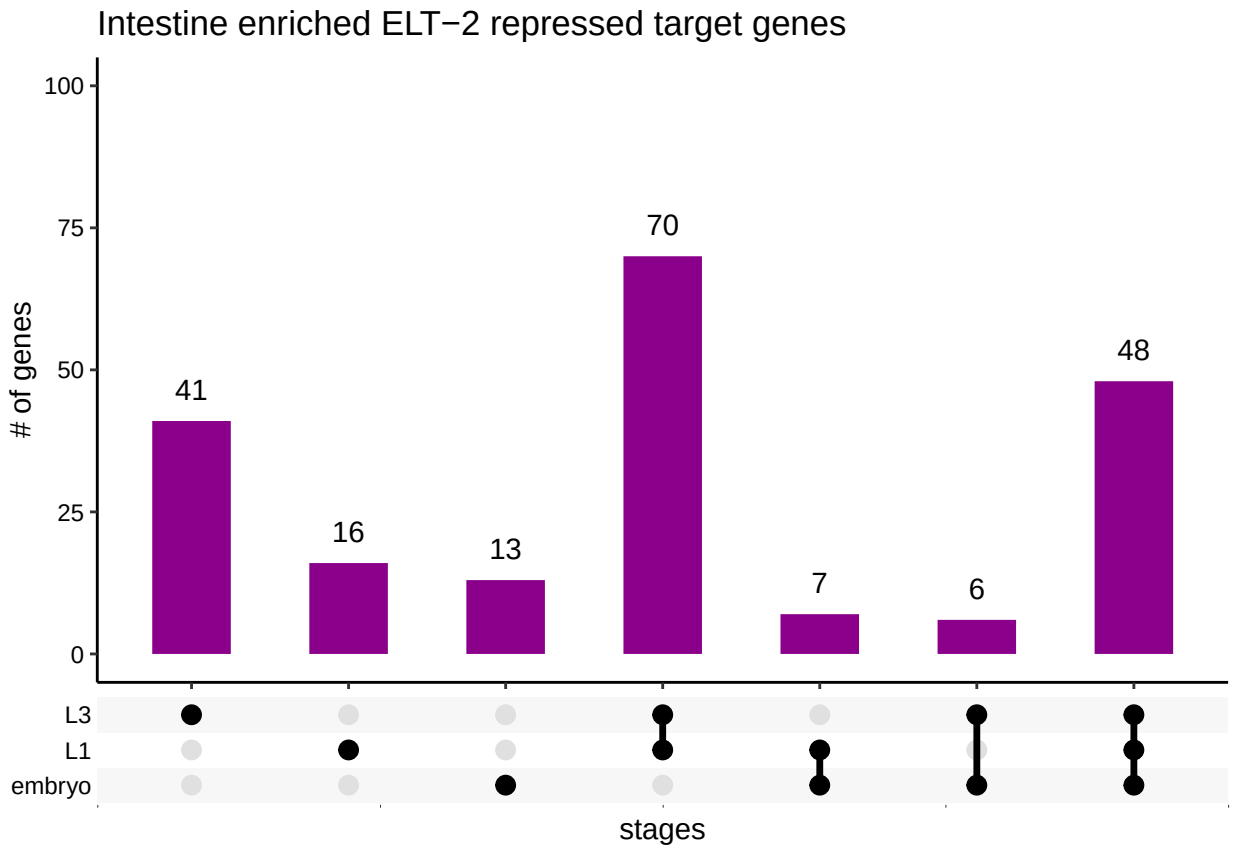
```
## `summarise()` has grouped output by 'stage', 'intestine_expression'. You can
## override using the `.groups` argument.
```

```
## # A tibble: 12 x 4
## # Groups:   stage, intestine_expression [6]
##    stage  intestine_expression promoter_status `n()`
##    <fct>  <chr>                <chr>           <int>
##  1 embryo enriched             bound             567
##  2 embryo enriched             not_bound        1169
##  3 embryo not_enriched         bound            2063
##  4 embryo not_enriched         not_bound       16196
##  5 L1     enriched             bound             869
```

```
##  6 L1     enriched             not_bound         608
##  7 L1     not_enriched         bound            2372
##  8 L1     not_enriched         not_bound       16146
##  9 L3     enriched             bound            1126
## 10 L3     enriched             not_bound         308
## 11 L3     not_enriched         bound            3918
## 12 L3     not_enriched         not_bound       14643
```

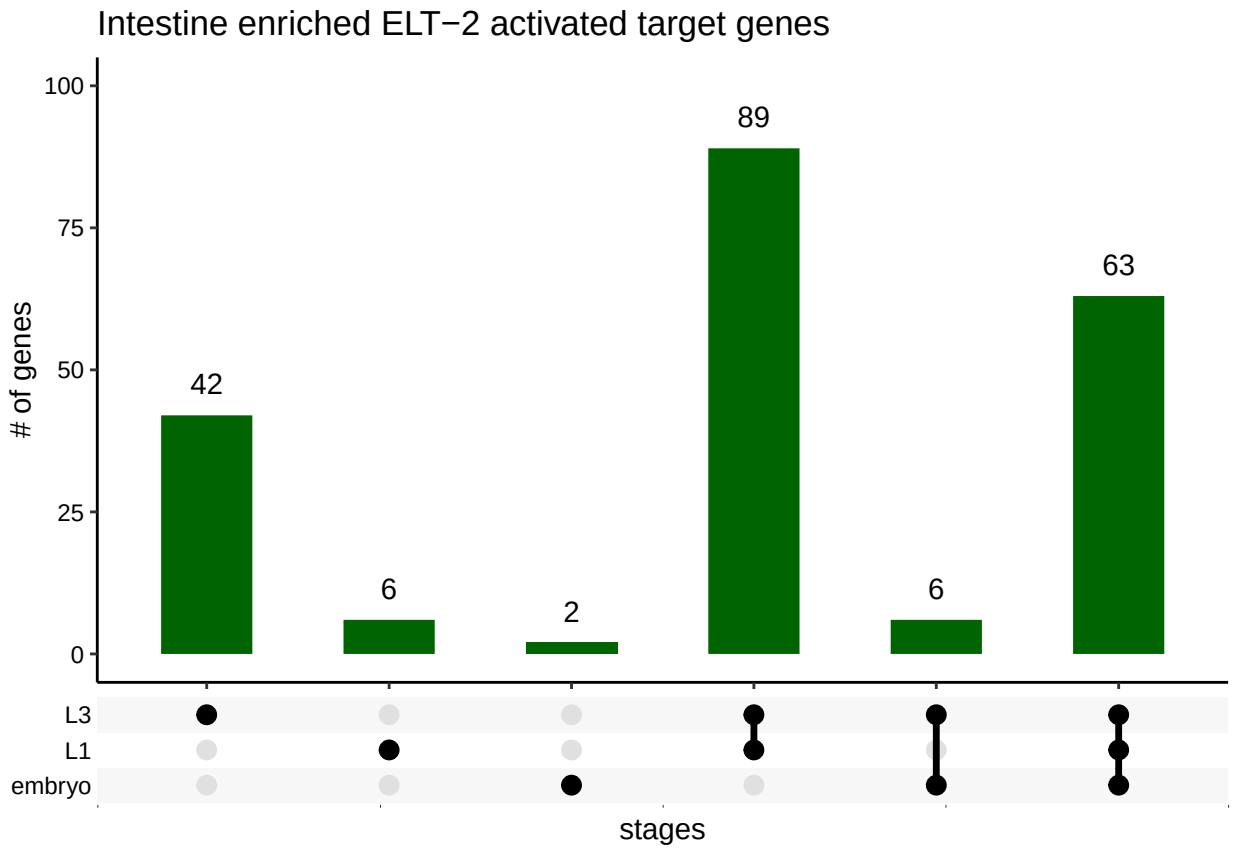
Upset plot: How are intestine enriched ELT-2 regulated targets shared or distinct between stages

```
library("ggupset")

upset_enriched_bound_repressed <-all_stages_chip %>% filter(
  intestine_expression == "enriched",
  promoter_status == "bound",
  elt2_ko %in% c("repressed")
) %>%
  mutate(stage = fct_rev(stage)) %>%
  group_by(WBGeneID) %>%
  summarise(stages = list(stage)) %>%
  ggplot(aes(x = stages)) +
  geom_bar(width = 0.5, fill = "darkmagenta") +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -1) +
  scale_x_upset(order_by = "degree") +
  scale_y_continuous(lim = c(0, 100), name = "# of genes") +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black")) +
  ggtitle("Intestine enriched ELT-2 repressed target genes")
upset_enriched_bound_repressed
```




```
# ggsave(upset_enriched_bound_repressed, filename = "../03_output/Upset/upset_enriched_bound_repressed.
```

```
upset_enriched_bound_activated <- all_stages_chip %>% filter(
  intestine_expression == "enriched",
  promoter_status == "bound",
  elt2_ko %in% c("activated")
) %>%
  mutate(stage = fct_rev(stage)) %>%
  group_by(WBGeneID) %>%
  summarise(stages = list(stage)) %>%
  ggplot(aes(x = stages)) +
  geom_bar(width = 0.5, fill = "darkgreen", ) +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -1) +
  scale_x_upset(order_by = "degree") +
  scale_y_continuous(lim = c(0, 100), name = "# of genes") +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black")) +
  ggtitle("Intestine enriched ELT-2 activated target genes")
upset_enriched_bound_activated
```




```
# ggsave(upset_enriched_bound_activated, filename = "../03_output/Upset/upset_enriched_bound_activated.
```

```
all_stages_chip %>% filter(
  intestine_expression == "enriched",
  promoter_status == "bound",
  elt2_ko %in% c("repressed")
) %>%group_by(WBGeneID) %>%
  summarise(stages = list(stage)) %>%
  mutate(stage_collapsed = sapply(stages, function(x) paste0(sort(x), collapse = "-"))) %>%
  filter(stage_collapsed == "embryo")
```

```
## # A tibble: 13 x 3
##    WBGeneID       stages    stage_collapsed
##    <chr>          <list>    <chr>
##  1 WBGene00000522 <fct [1]> embryo
##  2 WBGene00001247 <fct [1]> embryo
##  3 WBGene00006627 <fct [1]> embryo
##  4 WBGene00006628 <fct [1]> embryo
##  5 WBGene00008102 <fct [1]> embryo
##  6 WBGene00009102 <fct [1]> embryo
##  7 WBGene00011399 <fct [1]> embryo
##  8 WBGene00011427 <fct [1]> embryo
##  9 WBGene00011831 <fct [1]> embryo
## 10 WBGene00012182 <fct [1]> embryo
## 11 WBGene00017197 <fct [1]> embryo
## 12 WBGene00018207 <fct [1]> embryo
## 13 WBGene00021535 <fct [1]> embryo
```

Unique ELT-2 regulated genes

```
all_stages_chip %>% filter(
  # intestine_expression == "enriched",
  promoter_status == "bound",
  elt2_ko %in% c("activated")
)  %>% distinct(WBGeneID) %>% nrow()
```

```
## [1] 223
```

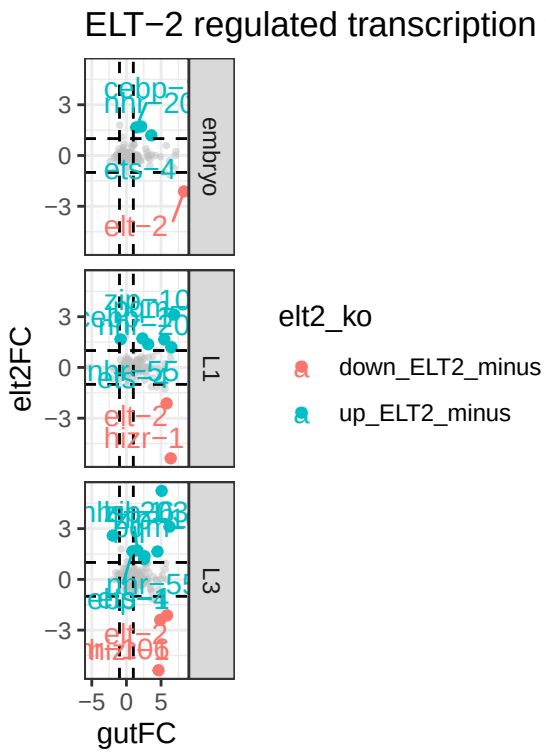
Transcription factor ELT-2 targets

Load data

```
# TF list
wtf3 <- read_csv("../../02_emb_L1_L3_intestine_RNAseq/01_input/TF3-0_namesonly.csv") %>% filter(!grepl(
GO_TFs <- read_delim("../01_input/elegans_genes_direct_and_inferred_for_GO_0001067_transcription_region_
# embryo
res_embryoGFPplus_vs_embryoGFPminus_ashr_shrunk <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq_
res_elt2D_v_wt_ashr_shrunk <- read_csv("../../03_elt2_RNAseq/03_output/res_elt2D_v_wt_ashr_shrunk.csv")
embryo_intestine_gene_categories<- read_csv("../../02_emb_L1_L3_intestine_RNAseq/03_output/intestine_ge
embryo_rna_chip_FC <- embryo_rna_chip %>%
  select(-intestine_expression) %>%
  left_join(embryo_intestine_gene_categories, by = "WBGeneID") %>%
  left_join(res_embryoGFPplus_vs_embryoGFPminus_ashr_shrunk %>% select(gutFC = log2FoldChange, WBGeneID)
  left_join(res_elt2D_v_wt_ashr_shrunk %>% dplyr::select(elt2FC = log2FoldChange, WBGeneID), by = "WBGe
  replace_na(list(gutFC = 0, elt2FC = 0))
# L1
res_L1GFPplus_vs_L1GFPminus_ashr_shrunk <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_outpu
res_elt2D_v_wt_ashr_shrunk <- read_csv("../../03_elt2_RNAseq/03_output/res_elt2D_v_wt_ashr_shrunk.csv")
L1_intestine_gene_categories<- read_csv("../../02_emb_L1_L3_intestine_RNAseq/03_output/intestine_gene_ca
L1_rna_chip_FC <- L1_rna_chip %>%
  select(-intestine_expression) %>%
  left_join(L1_intestine_gene_categories, by = "WBGeneID") %>%
  left_join(res_L1GFPplus_vs_L1GFPminus_ashr_shrunk %>% select(gutFC = log2FoldChange, WBGeneID), by =
  left_join(res_elt2D_v_wt_ashr_shrunk %>% select(elt2FC = log2FoldChange, WBGeneID), by = "WBGeneID") %
  replace_na(list(gutFC = 0, elt2FC = 0))
# L3
res_L3GFPplus_vs_L3GFPminus_ashr_shrunk <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_outpu
res_elt2D_v_wt_ashr_shrunk <- read_csv("../../03_elt2_RNAseq/03_output/res_elt2D_v_wt_ashr_shrunk.csv")
L3_intestine_gene_categories<- read_csv("../../02_emb_L1_L3_intestine_RNAseq/03_output/intestine_gene_ca
L3_rna_chip_FC <- L3_rna_chip %>%
  select(-intestine_expression) %>%
  left_join(L3_intestine_gene_categories, by = "WBGeneID") %>%
  left_join(res_L3GFPplus_vs_L3GFPminus_ashr_shrunk %>% select(gutFC = log2FoldChange, WBGeneID), by =
  left_join(res_elt2D_v_wt_ashr_shrunk %>% select(elt2FC = log2FoldChange, WBGeneID), by = "WBGeneID") %
  replace_na(list(gutFC = 0, elt2FC = 0))
```

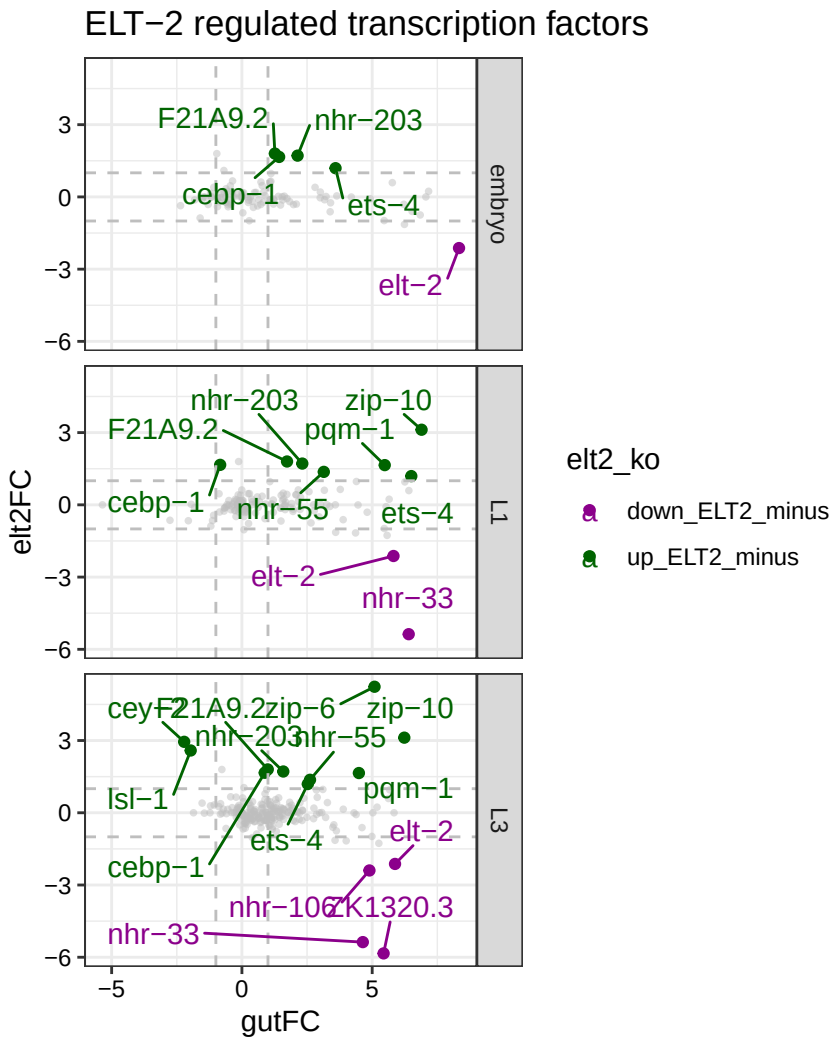
```
all_stage_rna_chip_FC <- embryo_rna_chip_FC %>% mutate(stage = "embryo") %>%
  bind_rows(L1_rna_chip_FC %>% mutate(stage = "L1")) %>%
  bind_rows(L3_rna_chip_FC %>% mutate(stage = "L3")) %>%
  filter(gutFC > -20)

all_stage_rna_chip_FC %>%
  filter(promoter_status == "bound", elt2_ko %in% c("up_ELT2_minus", "down_ELT2_minus")) %>%
```

```
  inner_join(GO_TFs, by = "WBGeneID") %>%
  ggplot(aes(x = gutFC, y = elt2FC, label = gene_name, color = elt2_ko)) +
  geom_point(data = all_stage_rna_chip_FC %>%
               filter(WBGeneID %in% GO_TFs$WBGeneID,
                      promoter_status == "bound") %>%
               select(gutFC, elt2FC, stage), color = "grey", size = 1, alpha = 0.5, shape = 16, aes(lab
  geom_point() +
  geom_hline(yintercept = c(-1,1), linetype = 2) +
  geom_vline(xintercept = c(-1,1), linetype = 2) +
  ggrepel::geom_text_repel(box.padding = 0.5, max.overlaps = 100, force = 5) +
  facet_grid(stage~.) +
  ggtitle("ELT-2 regulated transcription factors") +
  theme_bw()
```



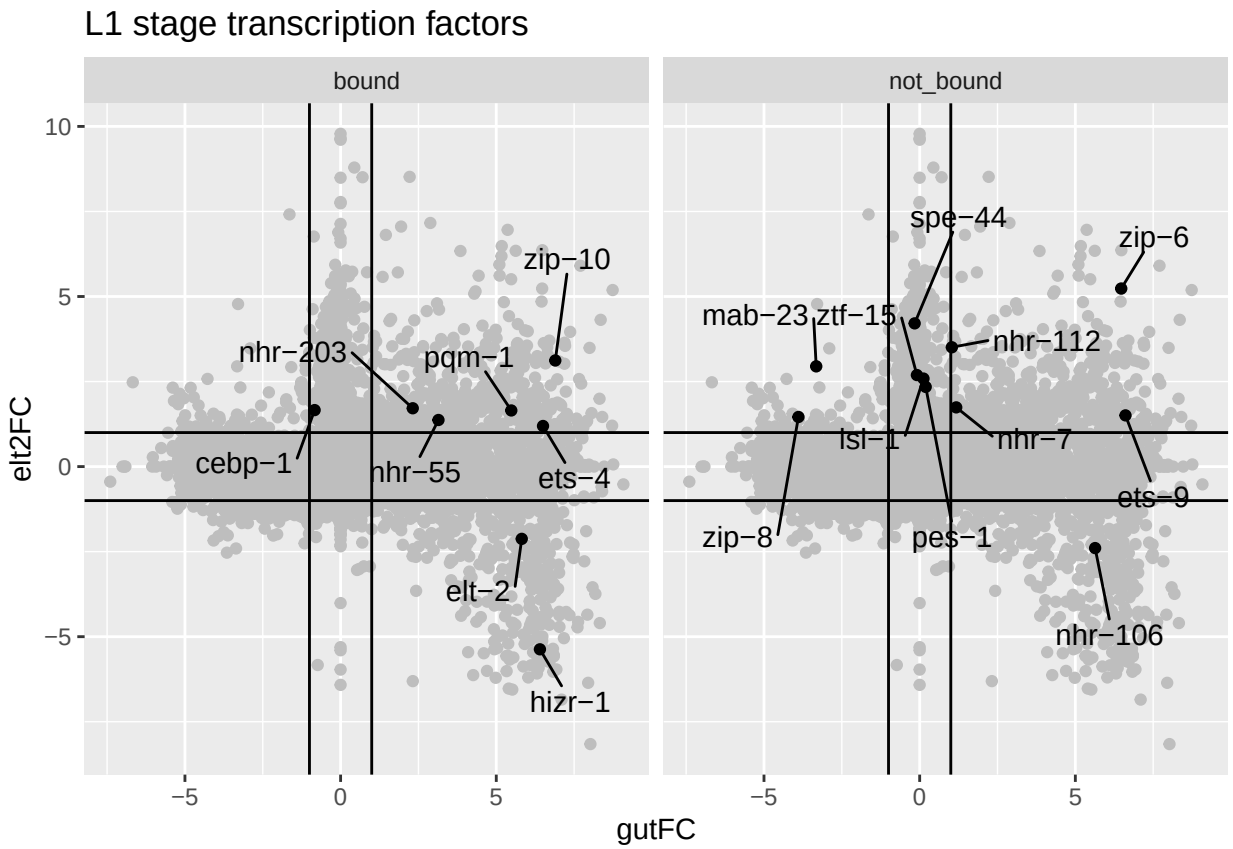

```
elt2_target_TFs_plot <- all_stage_rna_chip_FC %>%
  filter(promoter_status == "bound", elt2_ko %in% c("up_ELT2_minus", "down_ELT2_minus")) %>%
  inner_join(wtf3, by = "WBGeneID") %>%
  ggplot(aes(x = gutFC, y = elt2FC, label = Public_name, color = elt2_ko)) +
  geom_point(data = all_stage_rna_chip_FC %>%
               filter(WBGeneID %in% GO_TFs$WBGeneID,
                      promoter_status == "bound") %>%
               select(gutFC, elt2FC, stage), color = "grey", size = 1, alpha = 0.5, shape = 16, aes(lab
  geom_point() +
  geom_hline(yintercept = c(-1,1), linetype = 2, color = "grey") +
  geom_vline(xintercept = c(-1,1), linetype = 2,color = "grey") +
  ggrepel::geom_text_repel(box.padding = 0.6, max.overlaps = 100, force = 4) +
  scale_color_manual(values = c("darkmagenta", "darkgreen")) +
  facet_grid(stage~.) +
  ggtitle("ELT-2 regulated transcription factors") +
  theme_bw() +
```

```
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black"))

elt2_target_TFs_plot
```



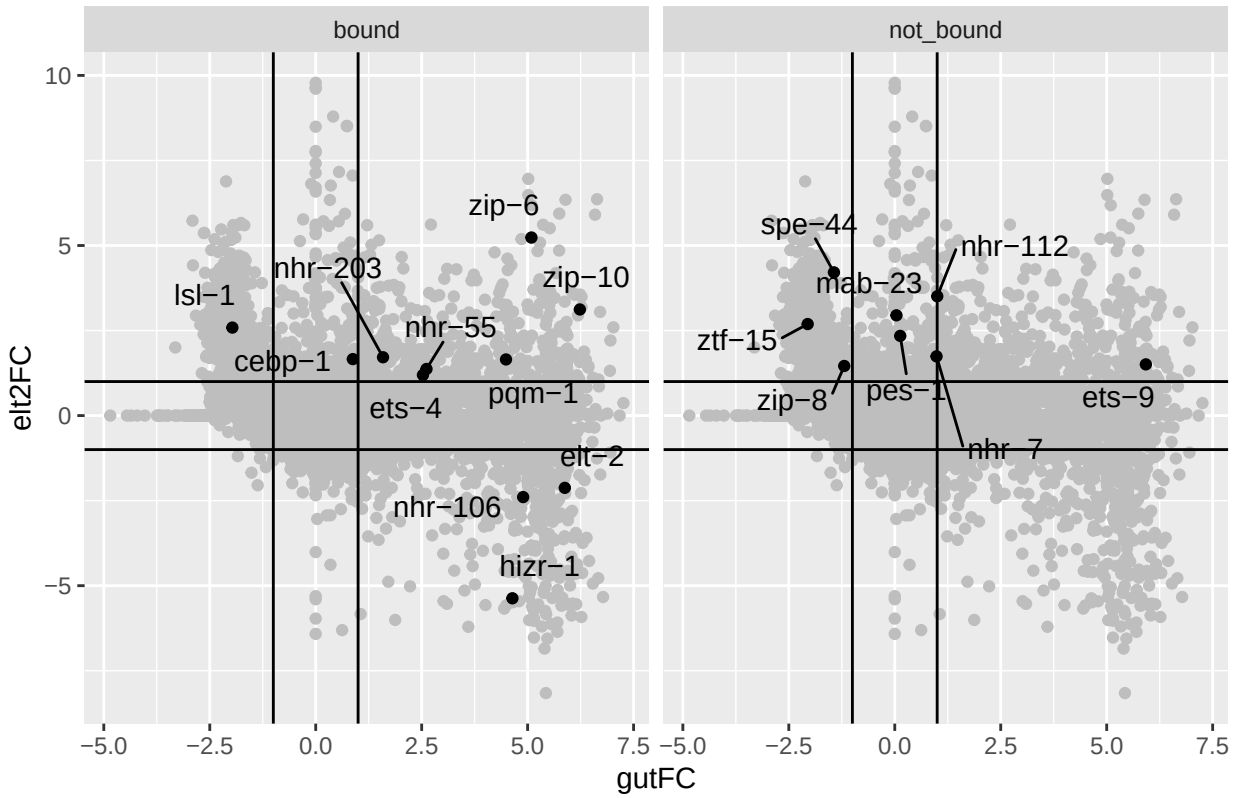

```
ggsave(filename = "../03_output/elt2_target_TFs_plot.pdf", width = 4.5, height = 5.5)
```

```
L1_rna_chip_FC %>%
  filter(elt2_ko %in% c("up_ELT2_minus", "down_ELT2_minus")) %>%
  inner_join(GO_TFs, by = "WBGeneID") %>%
  ggplot(aes(x = gutFC, y = elt2FC, label = gene_name)) +
  # ggplot(aes(x = gutFC, y = elt2FC)) +
  geom_point(data = L1_rna_chip_FC %>% select(gutFC, elt2FC) %>% filter(gutFC > -20), color = "grey", a
  geom_point() +
  geom_hline(yintercept = c(-1,1)) +
  geom_vline(xintercept = c(-1,1)) +
  ggrepel::geom_text_repel(box.padding = 1, max.overlaps = 100) +
  facet_grid(~promoter_status) +
  ggtitle("L1 stage transcription factors")
```




```
# L1_rna_chip_FC %>% group_by(elt2_ko, intestine_expression, promoter_status) %>% summarise(n())
```

```
L3_rna_chip_FC %>%
  filter(elt2_ko %in% c("up_ELT2_minus", "down_ELT2_minus")) %>%
  inner_join(GO_TFs, by = "WBGeneID") %>%
  ggplot(aes(x = gutFC, y = elt2FC, label = gene_name)) +
  # ggplot(aes(x = gutFC, y = elt2FC)) +
  geom_point(data = L3_rna_chip_FC %>% select(gutFC, elt2FC) %>% filter(gutFC > -20), color = "grey", a
  geom_point() +
  geom_hline(yintercept = c(-1,1)) +
  geom_vline(xintercept = c(-1,1)) +
  ggrepel::geom_text_repel(box.padding = 0.5, max.overlaps = 100) +
  facet_grid(~promoter_status) +
  ggtitle("L3 stage transcription factors")
```




```
# L3_rna_chip_FC %>% group_by(elt2_ko, intestine_expression, promoter_status) %>% summarise(n())
```

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] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggupset_0.3.0   forcats_0.5.1   stringr_1.4.0   dplyr_1.0.8
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
```

```
##  [1] Rcpp_1.0.8.3     ggrepel_0.9.1    lubridate_1.8.0  assertthat_0.2.1
##  [5] digest_0.6.29    utf8_1.2.2       R6_2.5.1         cellranger_1.1.0
##  [9] backports_1.4.1  reprex_2.0.1     evaluate_0.15    httr_1.4.2
## [13] highr_0.9        pillar_1.7.0     rlang_1.0.2      readxl_1.4.0
## [17] rstudioapi_0.13  rmarkdown_2.13   labeling_0.4.2   bit_4.0.4
## [21] munsell_0.5.0    broom_0.8.0      compiler_4.1.0   modelr_0.1.8
## [25] xfun_0.30        pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.2
## [29] fansi_1.0.3      crayon_1.5.1     tzdb_0.3.0       dbplyr_2.1.1
## [33] withr_2.5.0      grid_4.1.0       jsonlite_1.8.0   gtable_0.3.0
## [37] lifecycle_1.0.1  DBI_1.1.2        magrittr_2.0.3   scales_1.2.0
## [41] cli_3.2.0        stringi_1.7.6    vroom_1.5.7      farver_2.1.0
## [45] fs_1.5.2         xml2_1.3.3       ellipsis_0.3.2   generics_0.1.2
## [49] vctrs_0.4.0      tools_4.1.0      bit64_4.0.5      glue_1.6.2
## [53] hms_1.1.1        parallel_4.1.0   fastmap_1.1.0    yaml_2.3.5
## [57] colorspace_2.0-3 rvest_1.0.2      knitr_1.38       haven_2.4.3
```