intestine_enriched_genes

Rtpw

3/28/2022

Install packages

```
# BiocManager::install("topGO")
```

Load packages

```
library("topGO")
```

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

```
## Loading required package: parallel
```

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

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
```

```
## Loading required package: graph
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Loading required package: GO.db
```

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

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

```
## Loading required package: IRanges

## Loading required package: S4Vectors

## Warning: package 'S4Vectors' was built under R version 4.1.1

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##

## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve

##
## groupGOTerms:    GOBPTerm, GOMFTerm, GOCCTerm environments built.

##
## Attaching package: 'topGO'

## The following object is masked from 'package:IRanges':
##
##     members
library("tidyverse")

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## 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

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x stringr::boundary() masks graph::boundary()
## x dplyr::collapse()   masks IRanges::collapse()
## x dplyr::combine()    masks Biobase::combine(), BiocGenerics::combine()
## x dplyr::desc()       masks IRanges::desc()
## x tidyr::expand()     masks S4Vectors::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::first()      masks S4Vectors::first()
## x dplyr::lag()        masks stats::lag()
## x ggplot2::Position() masks BiocGenerics::Position(), base::Position()
## x purrr::reduce()     masks IRanges::reduce()
## x dplyr::rename()     masks S4Vectors::rename()
## x dplyr::select()     masks AnnotationDbi::select()
## x dplyr::slice()      masks IRanges::slice()
```

Tissue-specific marker genes analysis

Purpose: evaluate the contamination/enrichment of GFP+ intestine cells by visualizing the log2FoldChange of known tissue-specific genes

```
curated_tissue_genes <- read_csv(file = "../../01_tissue_specific_genes/01_input/Curated_Tissue_Specifi
  mutate(tissue = fct_relevel(tissue, c("intestine", "germline", "neuron", "muscle", "hypodermis"))) %>%
  mutate(gene_name = fct_rev(fct_reorder(gene_name, as.numeric(tissue))))
```

```
## Rows: 15 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): WBGeneID, gene_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.
```

```
res_embryoGFPplus_vs_embryoGFPminus_ashr_df <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_
```

```
## 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.
```

```
res_L1GFPplus_vs_L1GFPminus_ashr_df <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/pa
```

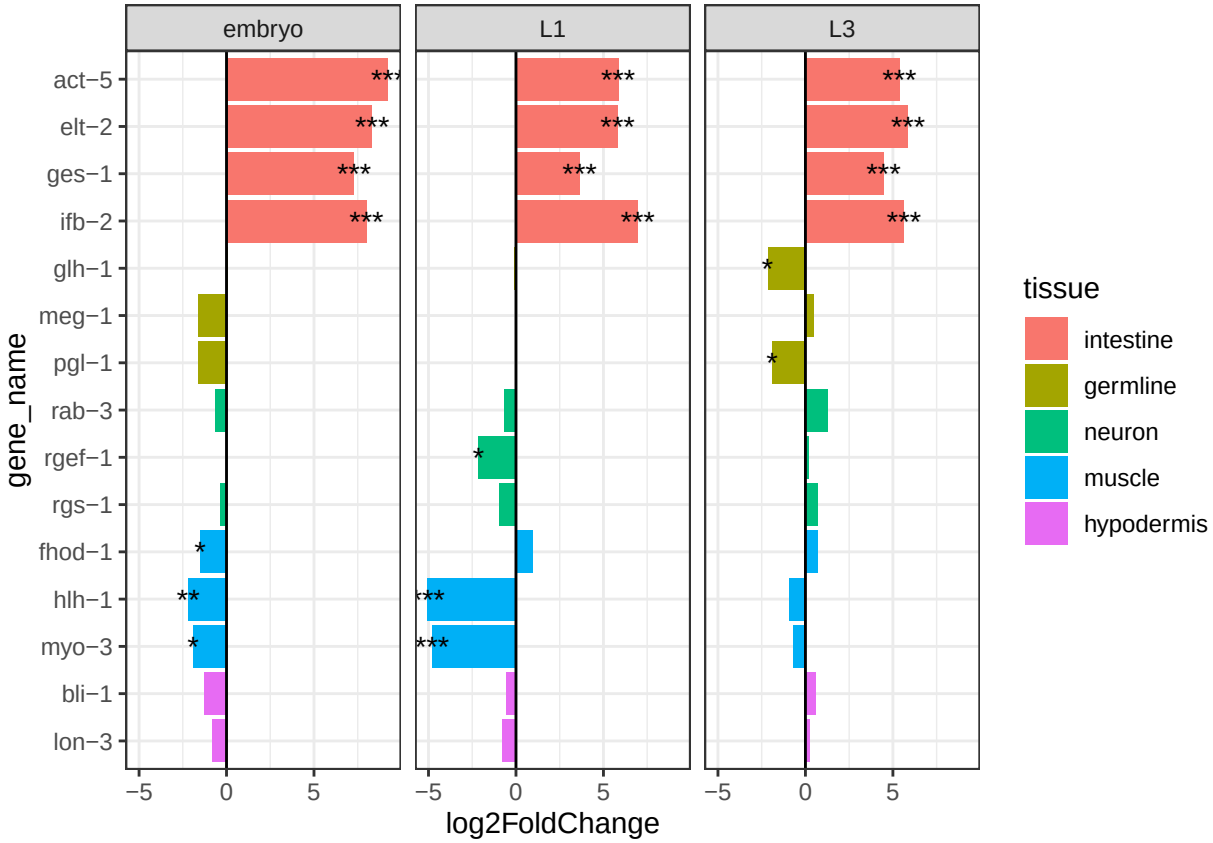
```
## Rows: 15627 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): WBGeneID
## dbl (5): baseMean, log2FoldChange, lfcSE, pvalue, padj
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify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

```
res_L3GFPplus_vs_L3GFPminus_ashr_df <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/pa
```

```
## 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.
```

```
curated_gene_foldchange <- data.frame(res_embryoGFPplus_vs_embryoGFPminus_ashr_df, stage = "embryo") %>%
  bind_rows(data.frame(res_L1GFPplus_vs_L1GFPminus_ashr_df, stage = "L1")) %>%
  bind_rows(data.frame(res_L3GFPplus_vs_L3GFPminus_ashr_df, stage = "L3")) %>%
  right_join(curated_tissue_genes, by = "WBGeneID") %>%
  mutate(star = case_when(
    padj > 0.01 ~ " ",
    padj < 1*10^-10 ~ "***",
```

```
    padj < 1*10^-5 ~ "**",
    padj < 0.01 ~ "*"
                          )) %>%
  ggplot(aes(x = gene_name, y = log2FoldChange, fill = tissue, label = star)) +
  geom_bar(stat = "identity") +
  # geom_hline(yintercept = 1, color = "black", linetype = 2) +
  geom_hline(yintercept = 0, color = "black") +
  facet_wrap(~stage) +
  geom_text() +
  coord_flip()+
  theme_bw()
curated_gene_foldchange
```




```
# ggsave(filename = "../03_output/Curated_Gene_Intestine_FoldChange.pdf", plot = curated_gene_foldchang
```

Intestine Enriched Gene Ontology

Save C. elegans gene ontology table

```
source("../../04_promoters/02_scripts/GOfxns.R")
# paramart <- biomaRt::useMart("parasite_mart", dataset = "wbps_gene", host = "https://parasite.wormbas
# WORMGO <- C_elegans_query(paramart)
#
# saveRDS(WORMGO, file = "../01_input/WORMGO.rds")
# WORMGO<- readRDS(file = "../01_input/WORMGO.rds")
```

```
WORMGO_nonObsolete <- read_tsv(file = "../01_input/Celegans_GOterms_NonObsolete_220330.tsv")
```

```
## Rows: 130120 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): Gene.symbol, Gene.goAnnotation.qualifier, Gene.goAnnotation.ontolog...
##
## 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.
```

```
WORMGO <- WORMGO_nonObsolete %>% select(wbps_gene_id = "Gene.primaryIdentifier",
                                external_gene_id = "Gene.symbol",
                                go_accession = "Gene.goAnnotation.ontologyTerm.identifier"
                                )
```

topGO helper functions

```
mkGOtissue = function(altHyp.df, WORMGO) {
  library(topGO)
  # create a named vector of p-values from DESEQ2 alternative hypothesis method
  allGenes <- altHyp.df %>% drop_na(padj) %>% pull(padj)
  names(allGenes) <- altHyp.df %>% drop_na(padj) %>% pull(WBGeneID)
  # make simple function to determine if a gene has significant p-value or not
  topDiffGenes <- function(geneVec){return(geneVec < 0.01)}
  # assign GO terms to each gene for topGO
  geneID2GO = geneID2GO(WORMGO)
  # set up topGOdata object for each ontology type
  BP.go = new("topGOdata", ontology='BP',
              allGenes = allGenes,
              geneSel = topDiffGenes,
              nodeSize = 10,
              annot = topGO::annFUN.gene2GO,
              gene2GO = geneID2GO)
  MF.go = new("topGOdata", ontology='MF',
              allGenes = allGenes,
              geneSel = topDiffGenes,
              nodeSize = 10,
              annot = topGO::annFUN.gene2GO,
              gene2GO = geneID2GO)
  CC.go = new("topGOdata", ontology='CC',
              allGenes = allGenes,
              geneSel = topDiffGenes,
              nodeSize = 10,
              annot = topGO::annFUN.gene2GO,
              gene2GO = geneID2GO)
  list(BP=BP.go,CC=CC.go,MF=MF.go)
}

GOSummaryTissue<- function(GOdata, topNodes = 200) {
  library(topGO)
  library(dplyr)
  resultFisher <- topGO::runTest(GOdata, algorithm = "weight01", statistic = "fisher")
  resultKS <- topGO::runTest(GOdata, algorithm = "weight01", statistic = "ks")
```

```
  resultFisherParentchild <- topGO::runTest(GOdata, algorithm = "parentchild", statistic = "fisher")
  tab <- topGO::GenTable(
    object=GOdata,
    ks.pval = resultKS,
    fisher.pval = resultFisher,
    fisher.PC.pval = resultFisherParentchild,
    orderBy="fisher.pval",
    topNodes = topNodes
  )
  # not sure where the conversion to char is happening. convert back
  # replace ">1e-30" character with number
  tab <- suppressMessages(
    tab %>% mutate(ks.pval = as.numeric(ks.pval), ks.pval = replace_na(ks.pval, 1e-30),
                   fisher.pval = as.numeric(fisher.pval), fisher.pval = replace_na(fisher.pval, 1e-30)
                   )
  )
  return(tab)
}

runGOtissue = function(altHyp.df, WORMGO, topNodes = 200)
{
  go = mkGOtissue(altHyp.df, WORMGO)
  go$BP.result = GOSummaryTissue(go$BP, topNodes)
  go$MF.result = GOSummaryTissue(go$MF, topNodes)
  go$CC.result = GOSummaryTissue(go$CC, topNodes)
  go
}
```

Embryo intestine GO terms

```
res_embryoGFP_alHyp_greater <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/res_embry
embryo_intestine_GO <- runGOtissue(res_embryoGFP_alHyp_greater, WORMGO)
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

L1 intestine GO terms

```
res_L1GFP_alHyp_greater<- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/res_L1GFP_alHy
L1_intestine_GO <- runGOtissue(res_L1GFP_alHyp_greater, WORMGO)
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

L3 intestine analysis

```
res_L3GFP_alHyp_greater<- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/res_L3GFP_alHyp
L3_intestine_GO <- runGOtissue(res_L3GFP_alHyp_greater, WORMGO)
```

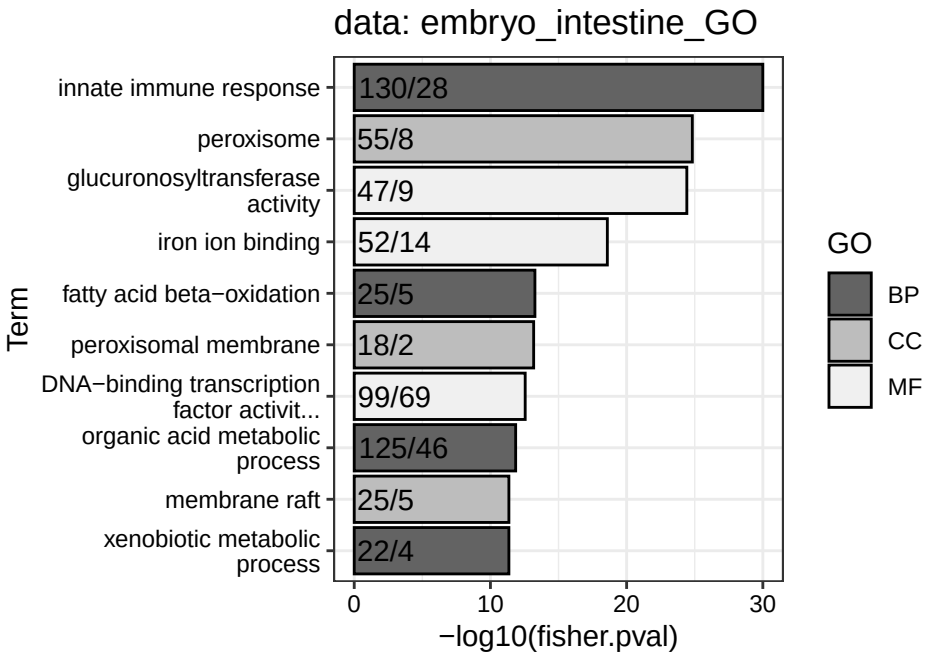
```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

topGO plotting function

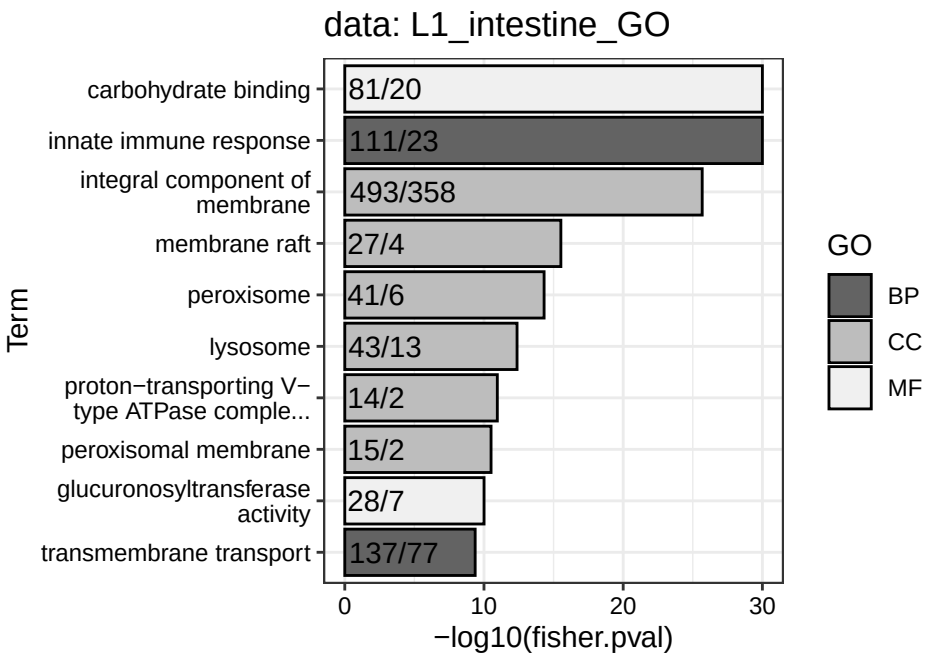
```
fisherGOplot <- function(in.df){
in.df$BP.result %>% mutate(GO = "BP") %>% bind_rows(in.df$MF.result %>% mutate(GO = "MF")) %>% bind_rows
  filter(Significant > Expected) %>%
  mutate(gene_count = paste0(Significant,"/", round(Expected))) %>%
  slice_min(fisher.pval, n = 10) %>%
  mutate(Term = fct_rev(fct_reorder(Term, fisher.pval))) %>%
ggplot(aes(x = Term, y = -log10(fisher.pval), fill = GO, label = gene_count)) +
  geom_bar(stat = "identity", color = "black") +
  geom_text(hjust = -0.05, aes(y = 0))+
  scale_fill_brewer(palette = "Greys", direction = -1) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 25)) +
  coord_flip() +
  theme_bw() +
  theme(axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black")) +
  ggtitle(paste("data:", deparse(substitute(in.df)), sep = " "))
}
```

```
embryo_intestine_GO_plot <- fisherGOplot(embryo_intestine_GO)
embryo_intestine_GO_plot
```



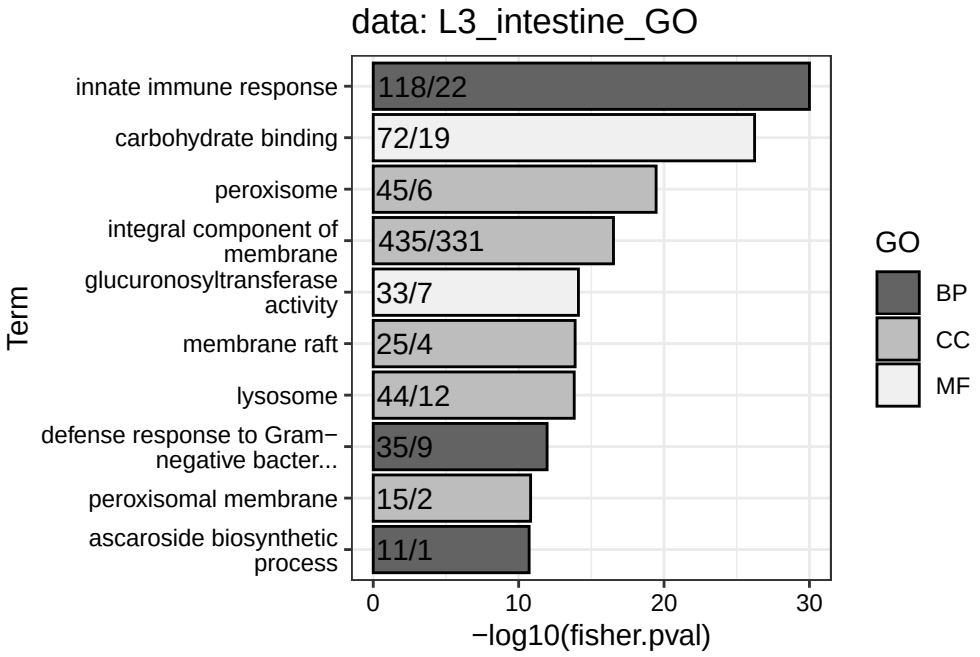
```
ggsave(plot = embryo_intestine_GO_plot, filename = "../03_output/GO_plots/embryo_intestine_GO_plot.pdf"
```

```
L1_intestine_GO_plot <- fisherGOplot(L1_intestine_GO)
L1_intestine_GO_plot
```




```
ggsave(plot = L1_intestine_GO_plot, filename = "../03_output/GO_plots/L1_intestine_GO_plot.pdf", width =
```

```
L3_intestine_GO_plot <- fisherGOplot(L3_intestine_GO)
L3_intestine_GO_plot
```



```
ggsave(plot = L3_intestine_GO_plot, filename = "../03_output/GO_plots/L3_intestine_GO_plot.pdf", width =
```

Intestine enriched per stage upset

```
library(ggupset)
embryo_intestine_gene_categories <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/intes
```

```
## 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.
```

```
L1_intestine_gene_categories <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/intestin
```

```
## Rows: 3361 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.
```

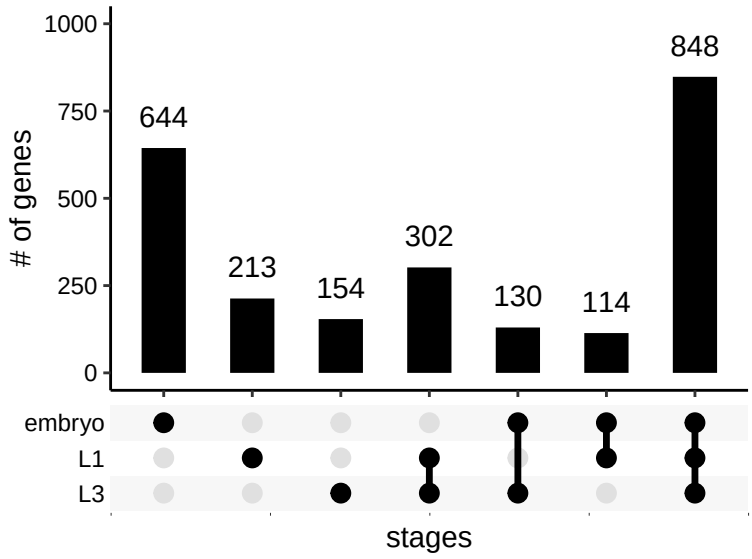
```
L3_intestine_gene_categories <- read_csv(file = "../../02_emb_L1_L3_intestine_RNAseq/03_output/intestin
```

```
## Rows: 2589 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.
```

```
all_stages_enriched <- embryo_intestine_gene_categories %>%
  mutate(stage = "embryo") %>%
  bind_rows(L1_intestine_gene_categories %>% mutate(stage = "L1")) %>%
  bind_rows(L3_intestine_gene_categories %>% mutate(stage = "L3")) %>%
  filter(intestine_expression == "enriched") %>%
  group_by(WBGeneID) %>%
  summarise(stages = list(stage))

intestine_upset <- all_stages_enriched %>%
  ggplot(aes(x = stages)) +
  geom_bar(width = 0.5, fill = "black") +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -1) +
  scale_x_upset(order_by = "degree") +
  scale_y_continuous(lim = c(0, 1000), name = "# of genes") +
  theme_classic() +
  theme(axis.text.x = element_text(colour = "black"),
        axis.text.y = element_text(colour = "black"))

intestine_upset
```




```
ggsave(intestine_upset, file = "../03_output/Intestine_Enriched_UpSet_Plot.pdf", width = 4, height = 3)
```

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] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ggupset_0.3.0        forcats_0.5.1        stringr_1.4.0
##  [4] dplyr_1.0.8          purrr_0.3.4          readr_2.1.2
##  [7] tidyr_1.2.0          tibble_3.1.6         ggplot2_3.3.5
## [10] tidyverse_1.3.1      topGO_2.44.0         SparseM_1.81
## [13] GO.db_3.13.0         AnnotationDbi_1.54.1 IRanges_2.26.0
## [16] S4Vectors_0.30.2     Biobase_2.52.0       graph_1.70.0
## [19] BiocGenerics_0.38.0
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7           matrixStats_0.61.0     fs_1.5.2
##  [4] lubridate_1.8.0        bit64_4.0.5            RColorBrewer_1.1-3
##  [7] httr_1.4.2             GenomeInfoDb_1.28.4    tools_4.1.0
## [10] backports_1.4.1        utf8_1.2.2             R6_2.5.1
```

```
## [13] DBI_1.1.2             colorspace_2.0-3      withr_2.5.0
## [16] tidyselect_1.1.2      bit_4.0.4             compiler_4.1.0
## [19] cli_3.2.0             rvest_1.0.2           xml2_1.3.3
## [22] labeling_0.4.2        scales_1.2.0          digest_0.6.29
## [25] rmarkdown_2.13        XVector_0.32.0        pkgconfig_2.0.3
## [28] htmltools_0.5.2       highr_0.9             dbplyr_2.1.1
## [31] fastmap_1.1.0         rlang_1.0.2           readxl_1.4.0
## [34] rstudioapi_0.13       RSQLite_2.2.12        farver_2.1.0
## [37] generics_0.1.2        jsonlite_1.8.0        vroom_1.5.7
## [40] RCurl_1.98-1.6        magrittr_2.0.3        GenomeInfoDbData_1.2.6
## [43] Rcpp_1.0.8.3          munsell_0.5.0         fansi_1.0.3
## [46] lifecycle_1.0.1       stringi_1.7.6         yaml_2.3.5
## [49] zlibbioc_1.38.0       grid_4.1.0            blob_1.2.3
## [52] crayon_1.5.1          lattice_0.20-45       Biostrings_2.60.2
## [55] haven_2.4.3           hms_1.1.1             KEGGREST_1.32.0
## [58] knitr_1.38            pillar_1.7.0          reprex_2.0.1
## [61] glue_1.6.2            evaluate_0.15         modelr_0.1.8
## [64] png_0.1-7             vctrs_0.4.0           tzdb_0.3.0
## [67] cellranger_1.1.0      gtable_0.3.0          assertthat_0.2.1
## [70] cachem_1.0.6          xfun_0.30             broom_0.8.0
## [73] memoise_2.0.1         ellipsis_0.3.2
```