GO_Term_Enrichment_Analysis

daniellembecker

9/20/2021

Molecular Underpinnings Chronic Nutrient Enrichment Project

RNAseq Gene Ontology (GO) Enrichment Analysis

Previous steps include RNAseq workflow in Bioinformatics>RNAseq>RNAseq workflow and Differential Gene Expression statistical analysis in RAnalysis>Scripts>RNAseq_Differential_Gene_Expression to make DEG statistical data sheet

This code is using the subsetted pver data. This code is also using the GO terms generated by Diamond (B2G), Swiss Prot (B2G), trembl (B2G), and InterProScan here in Dec. 2021. The workflow to make the functional annotation was here and how is was compiled is here

Import the data files

Structural annotation GFF3 file

```
## [1] 27439
```

Identifying treatment and all expressed pver genes (poverA = 0.90,10)

```
#treatment information
treatmentinfo <- read.csv("RAnalysis/Data/RNA-seq/metadata.RNAseq.csv", header = TRUE, sep = ",")
str(treatmentinfo)
```

```
## 'data.frame':    32 obs. of  4 variables:
##  $ fragment.ID: chr  "PV_1" "PV_10" "PV_11" "PV_12" ...
##  $ treatment  : chr  "enriched" "enriched" "enriched" "enriched" ...
##  $ block      : int  1 3 3 3 4 4 4 4 5 5 ...
##  $ sample_id  : chr  "E1" "E10" "E11" "E12" ...
```

```
head(treatmentinfo)
```

```
##   fragment.ID treatment block sample_id
## 1        PV_1  enriched     1        E1
## 2       PV_10  enriched     3       E10
## 3       PV_11  enriched     3       E11
## 4       PV_12  enriched     3       E12
## 5       PV_13  enriched     4       E13
## 6       PV_14  enriched     4       E14
```

```
#DEG significant results
DEG.res <- read.csv("RAnalysis/Output/RNA-seq/DEG/DEGSeq2.sig.results.csv")[,-1]
nrow(DEG.res)
```

```
## [1] 213
```

```
DEG.res$gene_id <-  gsub("_gene","",DEG.res$gene_id) #remove extra characters

colnames(DEG.res)[7] <-"gene_id" # make colnames a true column called gene_id
head(DEG.res)
```

```
##    baseMean log2FoldChange     lfcSE      stat       pvalue         padj
## 1  675.1574      0.2772978 0.0568262  4.879753 1.062190e-06 0.0010534948
## 2  400.7925     -0.6928043 0.1990790 -3.480046 5.013268e-04 0.0445544250
## 3  265.7893     -0.6440066 0.1583020 -4.068215 4.737462e-05 0.0121456901
## 4  231.3199     -1.4468803 0.3323853 -4.353021 1.342742e-05 0.0075180117
## 5 1974.0516     -2.0726526 0.4060384 -5.104572 3.315433e-07 0.0005568933
## 6  284.4700      0.6295127 0.1767905  3.560784 3.697497e-04 0.0376405169
##       gene_id
## 1   Pver_g130
## 2  Pver_g7878
## 3  Pver_g1842
## 4 Pver_g23161
## 5 Pver_g26469
## 6 Pver_g25888
```

```
#make upreg and downreg data frames
DOWNREG  <- DEG.res %>% filter(log2FoldChange < 0)
UPREG  <- DEG.res %>% filter(log2FoldChange > 0)
```

Set ID and gene length vectors, and make a binary matrix indicating which genes are differentially expressed. These are used as input to nullp, which for calculates a Probability Weighting Function for each set of DEGs.

#Build GOSEQ vector #GOseq requires a vector of all genes, all differentially expressed genes, and gene lengths

```
#Make ID and length vectors for all genes, UPREG and DOWNREG

#all genes
DEG <- GFF3[GFF3$gene_id %in% DEG.res$gene_id, ]
dim(DEG) # 213 x 10
```

```
## [1] 213  10
```

```
DEG_names <- as.vector(DEG$gene_id)
gene_vector <- as.integer(GFF3$gene_id%in%DEG_names)
names(gene_vector) <- GFF3$gene_id

#UPREG
DEG.up <- GFF3[GFF3$gene_id %in% UPREG$gene_id, ]
dim(DEG.up) # 66 x 10
```

```
## [1] 66 10
```

```
DEG_names.up <- as.vector(DEG.up$gene_id)
gene_vector.up <- as.integer(GFF3$gene_id%in%DEG_names.up)
names(gene_vector.up) <- GFF3$gene_id


#DOWNREG
DEG.down <- GFF3[GFF3$gene_id %in% DOWNREG$gene_id, ]
dim(DEG.down) # 147 x 10
```

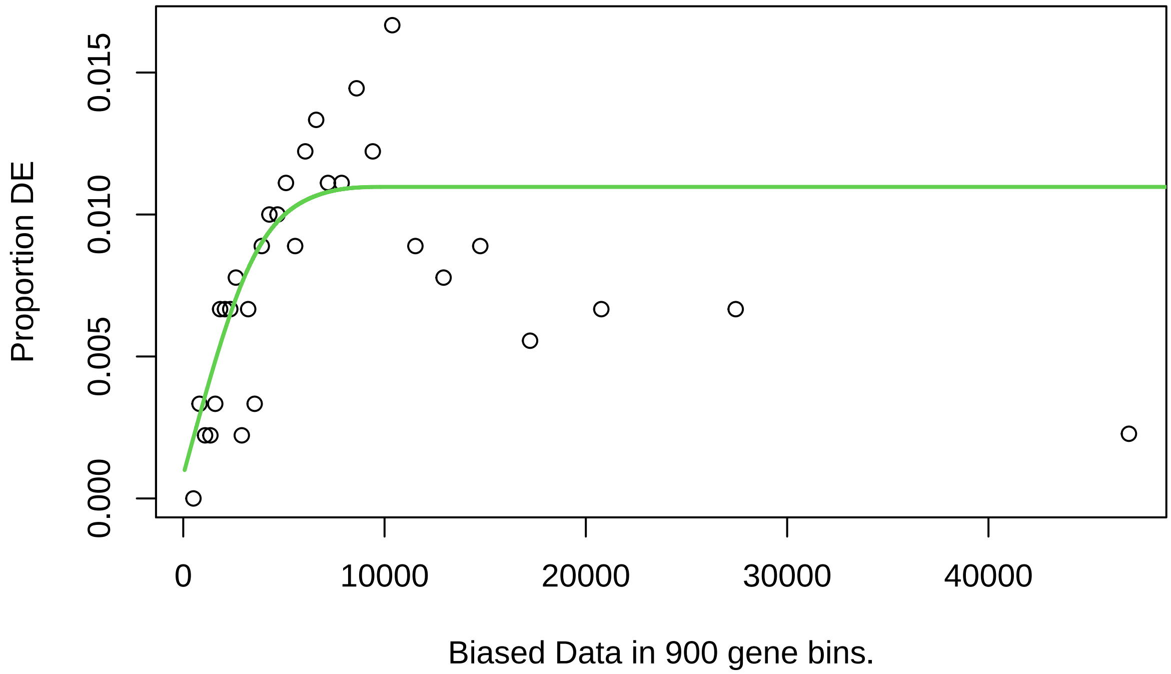
```
## [1] 147  10
```

```
DEG_names.down <- as.vector(DEG.down$gene_id)
gene_vector.down <- as.integer(GFF3$gene_id%in%DEG_names.down)
names(gene_vector.down) <- GFF3$gene_id

# Make ID vector
IDvector <- GFF3$gene_id

# Make length vector
lengthVector <- GFF3$length
```

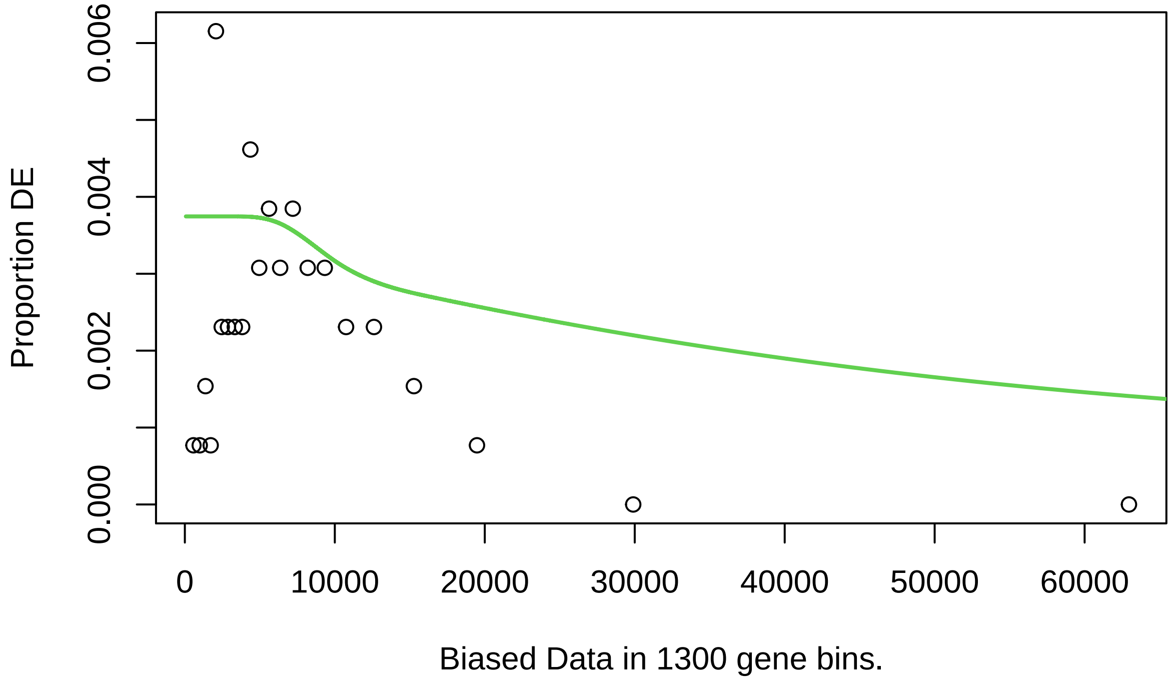
#Calculate Probability Weighting Function

```
pwf<-nullp(gene_vector, ID.vector, bias.data=lengthVector) #weight vector by length of gene
```

```
## Warning in pcls(G): initial point very close to some inequality constraints
```

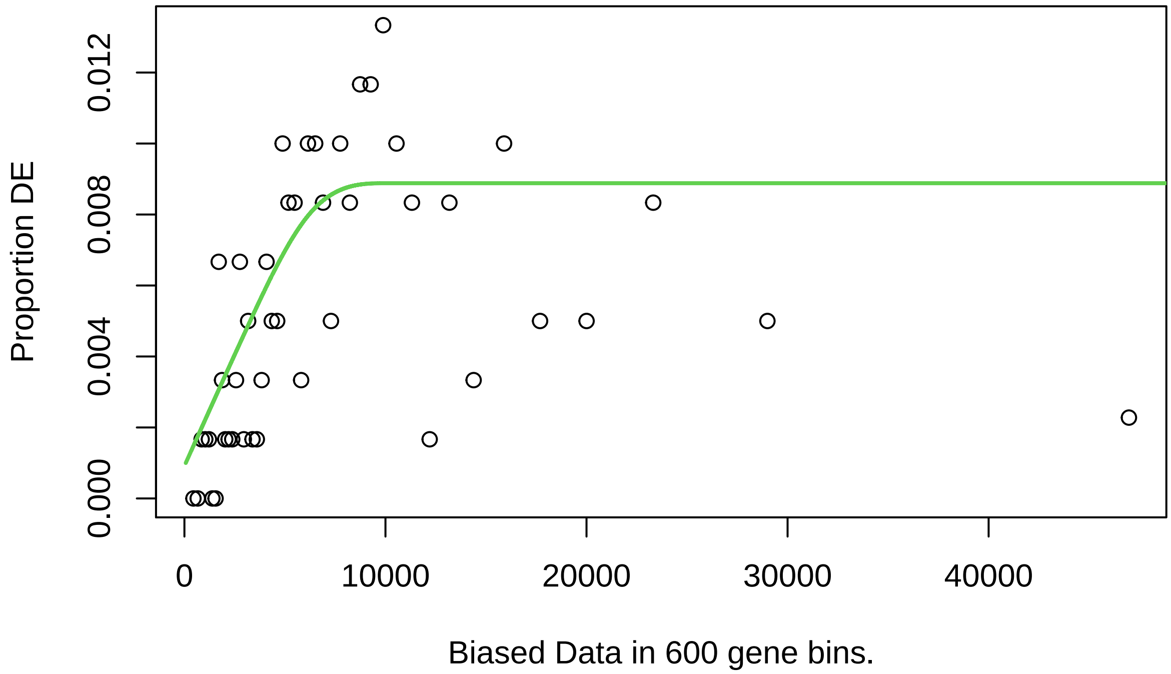


```
pwf.up<-nullp(gene_vector.up, ID.vector, bias.data=lengthVector) #weight vector by length of gene for u
```



```
pwf.down<-nullp(gene_vector.down, ID.vector, bias.data=lengthVector) #weight vector by length of gene f
```

```
## Warning in pcls(G): initial point very close to some inequality constraints
```



GOSeq w/ my annotations from Dec. 2021 (GO terms generated by Diamond (B2G), Swiss Prot (B2G), trembl (B2G), and IPS)

Load GO terms

```
annot_GO <- read.csv("Functional_Annotation/Final_Annotations/pver_GOterms_interprot_swissprot_blast_tre

# GO terms already split into single GO term per row/sequence
colnames(annot_GO) <- c("gene_id", "GO.ID")
annot_GO[annot_GO == "NA"] <- NA
annot_GO$GO.ID<- as.character(annot_GO$GO.ID)
annot_GO$GO.ID <- replace_na(annot_GO$GO.ID, "unknown")
annot_GO$GO.ID <- as.factor(annot_GO$GO.ID)
annot_GO$gene_id <- as.factor(annot_GO$gene_id)
annot_GO <- unique(annot_GO)
annot_GO$gene_id <-  gsub("\\..*","",annot_GO$gene_id) #remove extra characters after the first .


dim(annot_GO) # 282963 x 2
```

```
## [1] 282963      2
```

```
length(unique(annot_GO$GO.ID)) # 17023 unique GO.IDs
```

```
## [1] 17023
```

```
head(annot_GO)
```

```
##   gene_id      GO.ID
## 1 Pver_g1 GO:0002376
## 2 Pver_g1 GO:0005488
## 3 Pver_g1 GO:0005576
## 4 Pver_g1 GO:0009986
## 5 Pver_g1 GO:0012505
## 6 Pver_g1 GO:0031225
```

Perform GOseq

Find enriched GO terms, "selection-unbiased testing for category enrichment amongst differentially expressed (DE) genes for RNA-seq data"

```
### Perform GOseq for all, upreg and down reg
## Perform GOseq
# Find enriched GO terms, "selection-unbiased testing for category enrichment amongst differentially ex

GO.wall<-goseq(pwf, ID_vector, gene2cat=annot_GO, method="Wallenius", use_genes_without_cat=TRUE)
```

```
## Using manually entered categories.
```

```
## Calculating the p-values...
```

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

```
GO.wall.up<-goseq(pwf.up, ID_vector, gene2cat=annot_GO, method="Wallenius", use_genes_without_cat=TRUE)
```

```
## Using manually entered categories.
```

```
## Calculating the p-values...
```

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

```
GO.wall.down<-goseq(pwf.down, ID_vector, gene2cat=annot_GO, method="Wallenius", use_genes_without_cat=T
```

```
## Using manually entered categories.
```

```
## Calculating the p-values...
```

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

```
# Using manually entered categories.
# Calculating the p-values...
# 'select()' returned 1:1 mapping between keys and columns
write.csv(GO.wall, file = "RAnalysis/Output/RNA-seq/GOSeq/pver_GO_ALL.csv")
write.csv(GO.wall.up, file = "RAnalysis/Output/RNA-seq/GOSeq/pver_GO_UPREG.csv")
write.csv(GO.wall.down, file = "RAnalysis/Output/RNA-seq/GOSeq/pver_GO_DOWNREG.csv")


# Find significantly enriched GO terms in all genes
enriched.GO.05 <-GO.wall$category[GO.wall$over_represented_pvalue<.05]
enriched.GO.05 <-data.frame(enriched.GO.05)
colnames(enriched.GO.05) <- c("category")
enriched.GO.05 <- merge(enriched.GO.05, GO.wall, by="category")
enriched.GO.05 <- enriched.GO.05[order(-enriched.GO.05$numDEInCat),]
enriched.GO.05$term <- as.factor(enriched.GO.05$term)
head(enriched.GO.05)
```

```
##       category over_represented_pvalue under_represented_pvalue numDEInCat
## 174 GO:0042802             0.016414584                0.9915369         20
## 98  GO:0016491             0.004917154                0.9983442         11
## 39  GO:0005789             0.029886807                0.9889690          8
## 4   GO:0000978             0.011772891                0.9966816          7
## 43  GO:0005938             0.001567590                0.9996960          7
## 54  GO:0007166             0.015193609                0.9955033          7
##     numInCat
## 174     1448
## 98       534
## 39       440
## 4        301
## 43       195
## 54       281
##                                                                         term
## 174                                                  identical protein binding
## 98                                                   oxidoreductase activity
## 39                                            endoplasmic reticulum membrane
## 4   RNA polymerase II cis-regulatory region sequence-specific DNA binding
## 43                                                               cell cortex
## 54                                    cell surface receptor signaling pathway
##     ontology
## 174       MF
## 98        MF
## 39        CC
## 4         MF
## 43        CC
## 54        BP
```

```
# Subset enriched GO terms by ontology (BP, CC, MF) and save as csv.
MF <- subset(enriched.GO.05, ontology=="MF")
MF <- MF[order(-MF$numDEInCat),]
CC <- subset(enriched.GO.05, ontology=="CC")
CC <- CC[order(-CC$numDEInCat),]
BP <- subset(enriched.GO.05, ontology=="BP")
BP <- BP[order(-BP$numDEInCat),]
write.csv(MF, file = "RAnalysis/Output/RNA-seq/GOSeq/pver_MF_Sig_Enriched_GO.05.csv")
write.csv(CC, file = "RAnalysis/Output/RNA-seq/GOSeq/pver_CC_Sig_Enriched_GO.05.csv")
write.csv(BP, file = "RAnalysis/Output/RNA-seq/GOSeq/pver_BP_Sig_Enriched_GO.05.csv")
write.csv(enriched.GO.05, file = "RAnalysis/Output/RNA-seq/GOSeq/pver_Sig_Enriched_GO.05_ALL.csv")

nrow(enriched.GO.05) # 308
```

```
## [1] 308
```

```
nrow(filter(enriched.GO.05, ontology=="BP")) #number sig BP terms 211
```

```
## [1] 211
```

```
nrow(filter(enriched.GO.05, ontology=="MF")) #number sig MF terms 70
```

```
## [1] 70
```

```
nrow(filter(enriched.GO.05, ontology=="CC")) #number sig CC terms 27
```

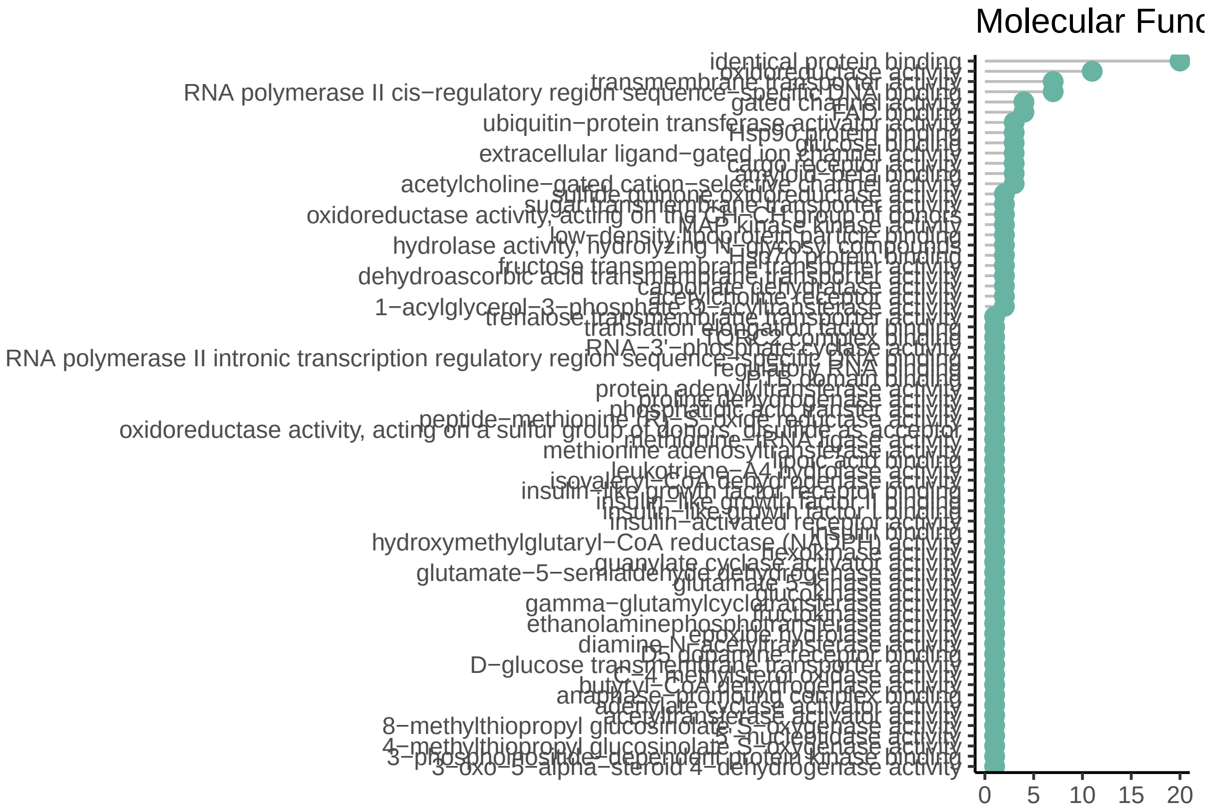
```
## [1] 27
```

Merge DEG gene ids and GO info

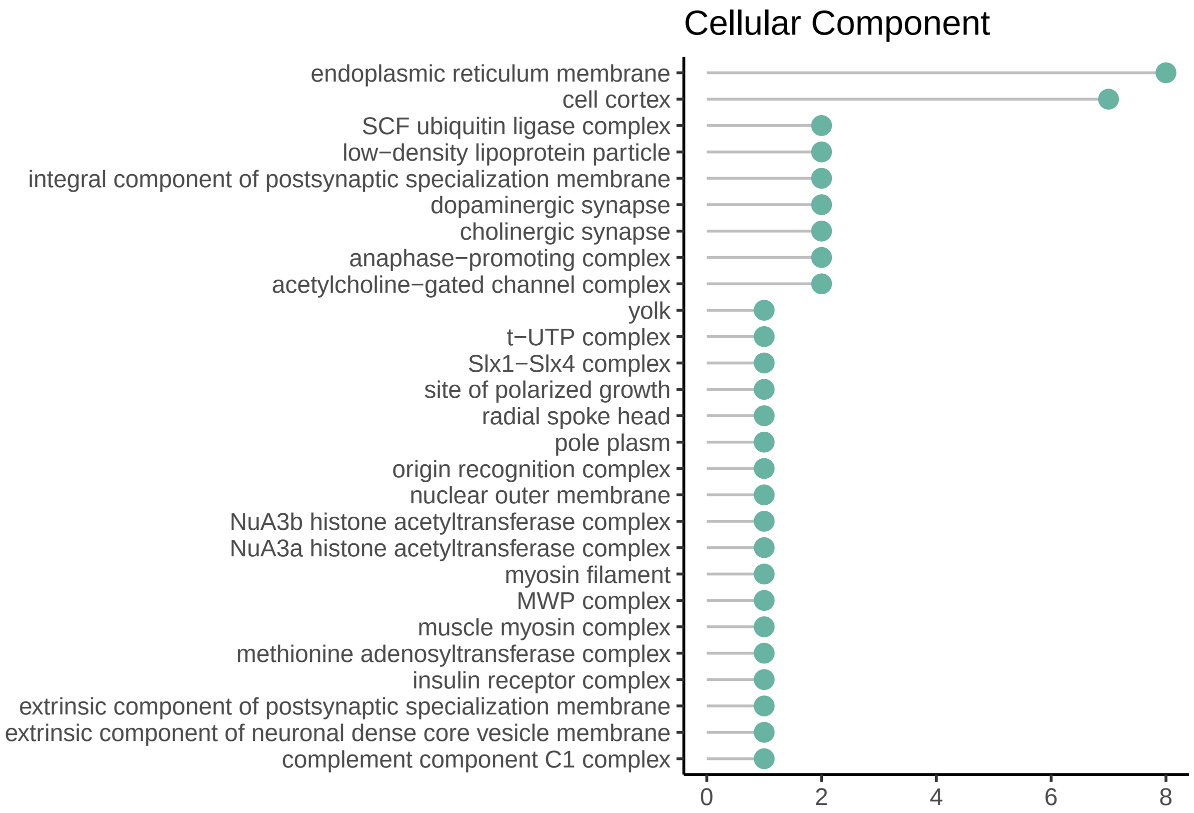
```
# Merge gene ids and sig enriched GO terms
colnames(annot_GO) <- c("gene_id", "category")
merge <- merge(enriched.GO.05, annot_GO, by.x = "category") # contains gene ids and GO info
#merge <- unique(merge)
# Merge DEG with merge
merge_again <- merge(merge, DEG, by = "gene_id") # contains GO info and gene counts for DEGs
```

Plot terms by ontology numbers

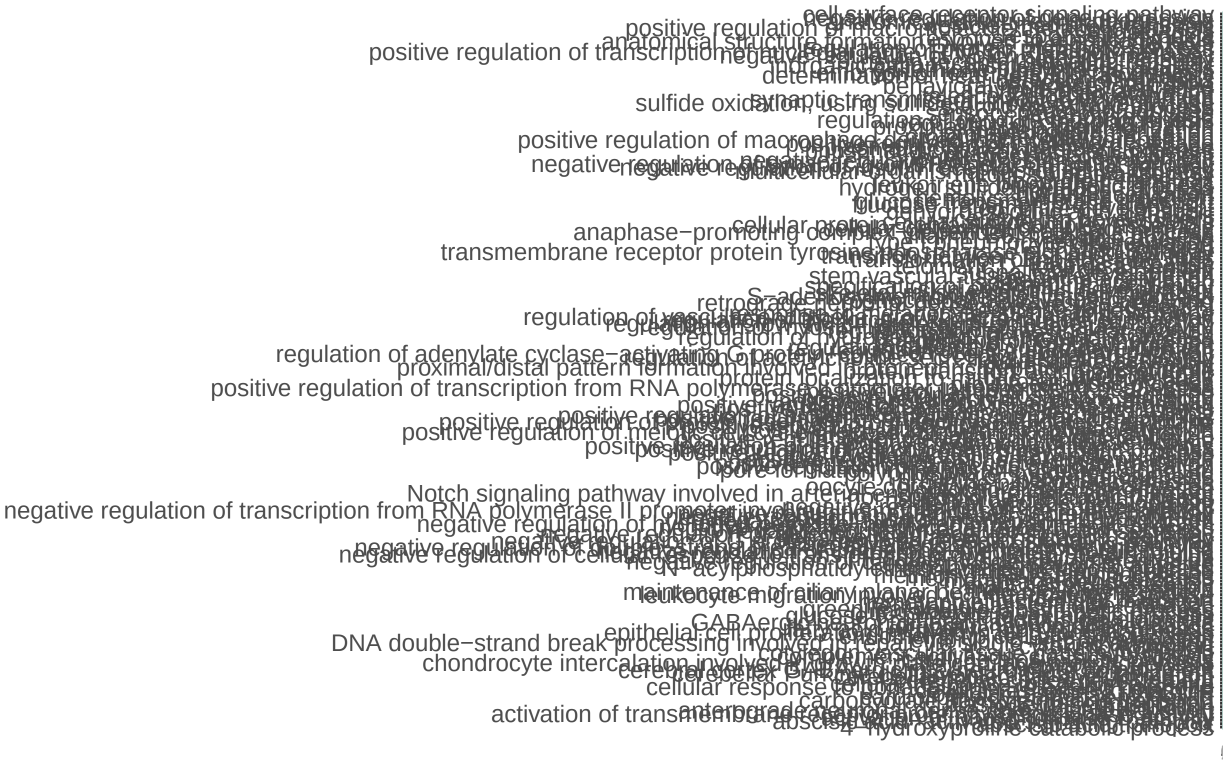
```
MFplot <- MF %>% mutate(term = fct_reorder(term, numDEInCat)) %>%
  ggplot( aes(x=term, y=numDEInCat) ) +
  geom_segment( aes(x=term ,xend=term, y=0, yend=numDEInCat), color="grey") +
  geom_point(size=3, color="#69b3a2") +
  coord_flip() +
  theme(
    panel.grid.minor.y = element_blank(),
    panel.grid.major.y = element_blank(),
    legend.position="none"
  ) +
  xlab("") +
  ylab("") +
  ggtitle("Molecular Function") + #add a main title
  theme(plot.title = element_text(face = 'bold',
                                  size = 12,
                                  hjust = 0)) +
  theme_bw() + #Set background color
  theme(panel.border = element_blank(), # Set border
        panel.grid.major = element_blank(), #Set major gridlines
        panel.grid.minor = element_blank(), #Set minor gridlines
        axis.line = element_line(colour = "black"), #Set axes color
        plot.background=element_blank())#Set the plot background
MFplot
```



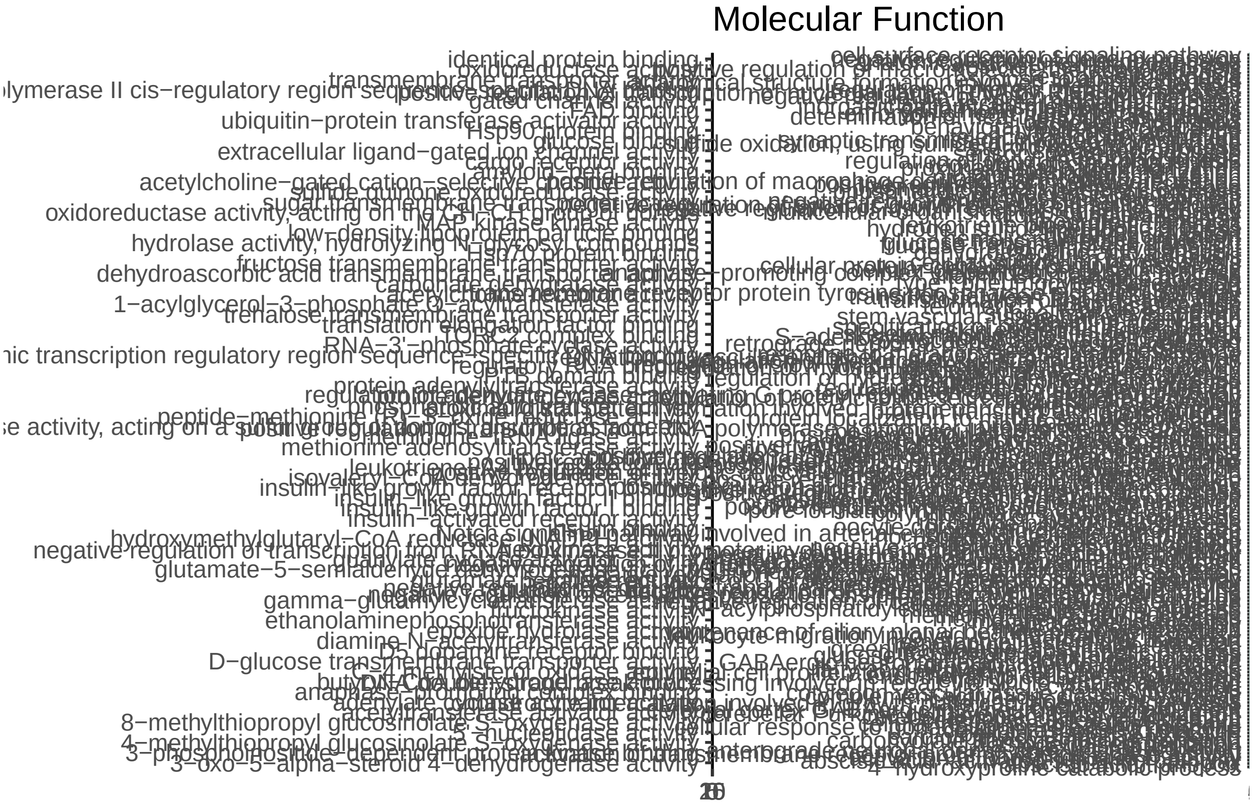
```
CCplot <- CC %>% mutate(term = fct_reorder(term, numDEInCat)) %>%
  ggplot( aes(x=term, y=numDEInCat) ) +
  geom_segment( aes(x=term ,xend=term, y=0, yend=numDEInCat), color="grey") +
  geom_point(size=3, color="#69b3a2") +
  coord_flip() +
  theme(
    panel.grid.minor.y = element_blank(),
    panel.grid.major.y = element_blank(),
    legend.position="none"
  ) +
  xlab("") +
  ylab("") +
  ggtitle("Cellular Component") + #add a main title
  theme(plot.title = element_text(face = 'bold',
                                  size = 12,
                                  hjust = 0)) +
  theme_bw() + #Set background color
  theme(panel.border = element_blank(), # Set border
        panel.grid.major = element_blank(), #Set major gridlines
        panel.grid.minor = element_blank(), #Set minor gridlines
        axis.line = element_line(colour = "black"), #Set axes color
        plot.background=element_blank())#Set the plot background
CCplot
```



```
BPplot <- BP %>% mutate(term = fct_reorder(term, numDEInCat)) %>%
  ggplot( aes(x=term, y=numDEInCat) ) +
  geom_segment( aes(x=term ,xend=term, y=0, yend=numDEInCat), color="grey") +
  geom_point(size=3, color="#69b3a2") +
  coord_flip() +
  theme(
    panel.grid.minor.y = element_blank(),
    panel.grid.major.y = element_blank(),
    legend.position="none") +
  xlab("") +
  ylab("") +
  ggtitle("Biological Process") + #add a main title
  theme(plot.title = element_text(face = 'bold',
                                  size = 12,
                                  hjust = 0)) +
  theme_bw() + #Set background color
  theme(panel.border = element_blank(), # Set border
        panel.grid.major = element_blank(), #Set major gridlines
        panel.grid.minor = element_blank(), #Set minor gridlines
        axis.line = element_line(colour = "black"), #Set axes color
        plot.background=element_blank())#Set the plot background
BPplot # lots of BP, hard to see. Could subset df and plot by BP dfs
```



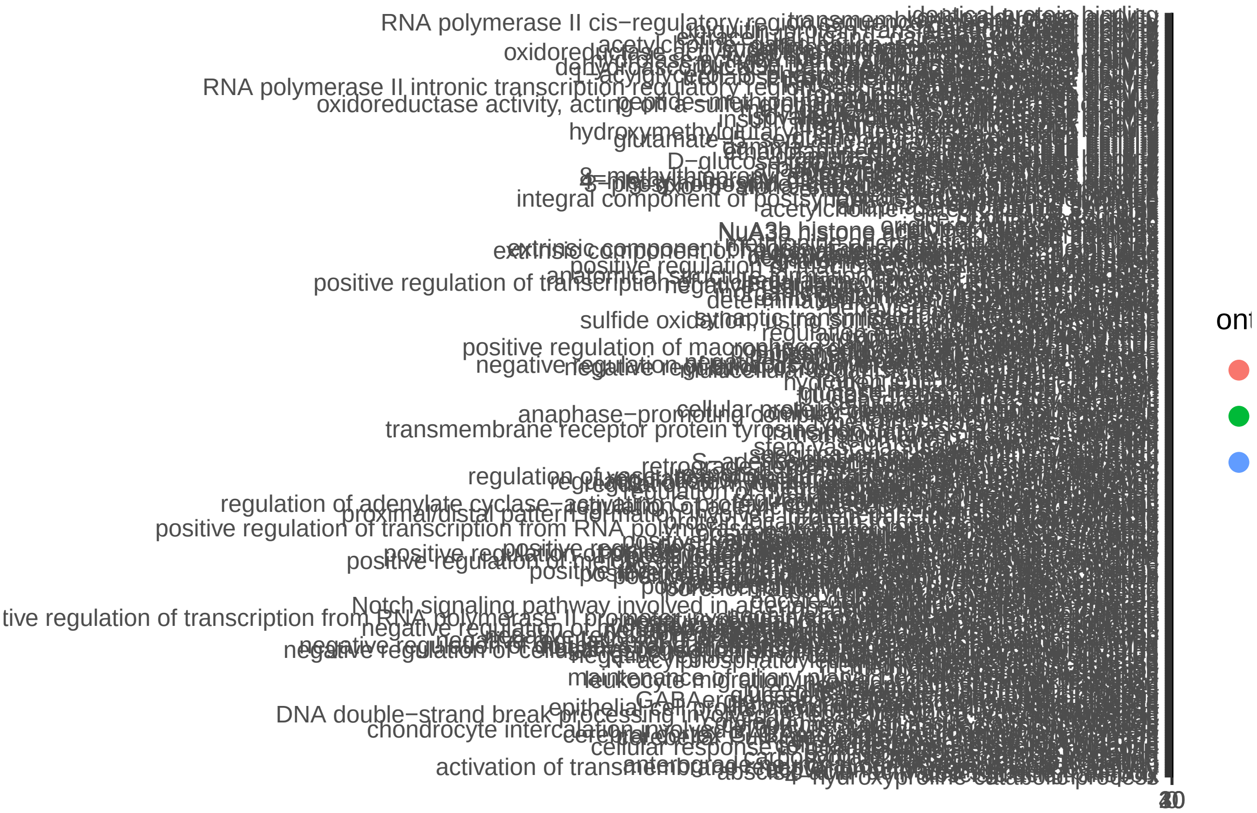
```
# Save MF, CC, and BP plot in grid layout
GOplot <- grid.arrange(MFplot, BPplot, ncol=3, clip="off")
```



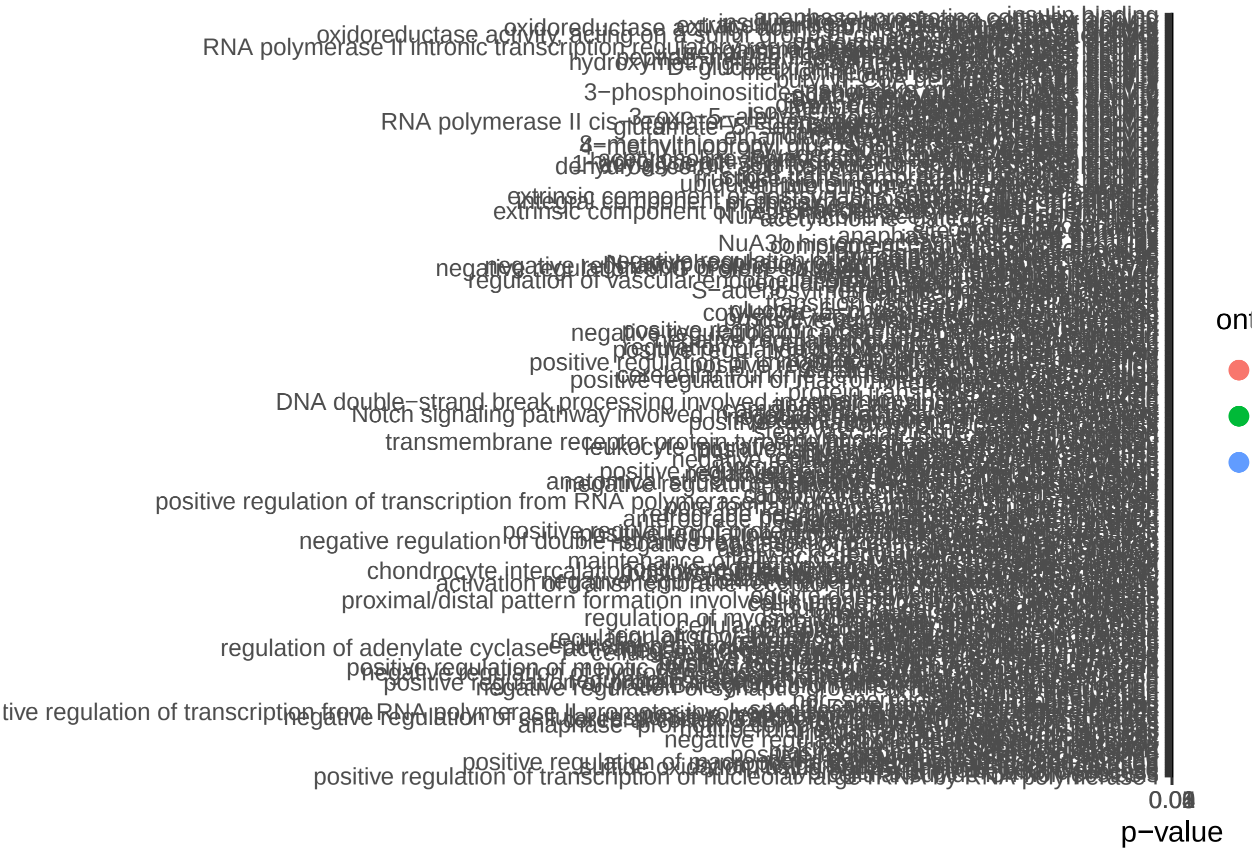
```
ggsave("RAnalysis/Output/Final_Figures/pver_GOplot_05_FullAnnot.pdf", GOplot, width = 21, height = 21,
```

Combining MF, CC, and BP into one plot and order by pvalue

```
GOplot2 <- enriched.GO.05 %>% drop_na(ontology) %>% mutate(term = fct_reorder(term, numDEInCat)) %>%
  mutate(term = fct_reorder(term, ontology)) %>%
  ggplot( aes(x=term, y=numDEInCat) ) +
  geom_segment( aes(x=term ,xend=term, y=0, yend=numDEInCat), color="grey") +
  geom_text(aes(label = over_represented_pvalue), hjust = -1, vjust = 0, size = 2) +
  geom_point(size=3, aes(colour = ontology)) +
  coord_flip() +
  ylim(0,45) +
  theme(
    panel.grid.minor.y = element_blank(),
    panel.grid.major.y = element_blank(),
    legend.position="bottom"
  ) +
  xlab("") +
  ylab("") +
  theme_bw() + #Set background color
  theme(panel.border = element_blank(), # Set border
        panel.grid.major = element_blank(), #Set major gridlines
        panel.grid.minor = element_blank(), #Set minor gridlines
        axis.line = element_line(colour = "black"), #Set axes color
        plot.background=element_blank()) #Set the plot background #set title attributes
GOplot2
```



```r
ggsave("RAnalysis/Output/Final_Figures/pver_GOplot2_05.pdf", GOplot2, width = 28, height = 28, units = 

# Combining into one plot and order by pvalue
GOplot_pvalue <- enriched.GO.05 %>% drop_na(ontology) %>% mutate(term = fct_reorder(term, over_represent
  mutate(term = fct_reorder(term, ontology)) %>%
  ggplot( aes(x=term, y=over_represented_pvalue) ) +
  geom_segment( aes(x=term ,xend=term, y=0, yend=over_represented_pvalue), color="grey") +
  geom_text(aes(label = numDEInCat), hjust = -1, vjust = 0.5, size = 3) +
  geom_point(size=3, aes(colour = ontology)) +
  coord_flip() +
  ylim(0,0.05) +
  theme(
    panel.grid.minor.y = element_blank(),
    panel.grid.major.y = element_blank(),
    legend.position="bottom"
  ) +
  xlab("") +
  ylab("p-value") +
  theme_bw() + #Set background color
  theme(panel.border = element_blank(), # Set border
        panel.grid.major = element_blank(), #Set major gridlines
        panel.grid.minor = element_blank(), #Set minor gridlines
        axis.line = element_line(colour = "black"), #Set axes color
        plot.background=element_blank()) #Set the plot background #set title attributes
GOplot_pvalue
```



```
ggsave("RAnalysis/Output/Final_Figures/pver_GOplot_pvalue_05_FullAnnot.pdf", GOplot_pvalue, width = 28,
```

Find GOslim terms

Run GOslim to get broader categories

```
#load in generic GO database, has all of the upper level categories for GO terms. Ex: regulation of cel
slim <- getOBOCollection("http://current.geneontology.org/ontology/subsets/goslim_generic.obo") #get GO

## filtering all of BP (do MF and CC seperately)
BP_GO <- enriched.GO.05 %>%
  filter(ontology=="BP")
BPGO_collection <- GOCollection(BP_GO$category) #Make library of query terms
slims_bp <- data.frame(goSlim(BPGO_collection, slim, "BP")) #Find common parent terms to slim down our
slims_bp$category <- row.names(slims_bp) #save rownames as category

## filtering all of MF
MF_GO <- enriched.GO.05 %>%
  filter(ontology=="MF")
MFGO_collection <- GOCollection(MF_GO$category) #Make library of query terms
slims_mf <- data.frame(goSlim(MFGO_collection, slim, "MF")) #Find common parent terms to slim down our
slims_mf$category <- row.names(slims_mf) #save rownames as category

## filtering all of CC
```

```
CC_GO <- enriched.GO.05 %>%
  filter(ontology=="CC")
CCGO_collection <- GOCollection(CC_GO$category) #Make library of query terms
slims_cc <- data.frame(goSlim(CCGO_collection, slim, "CC")) #Find common parent terms to slim down our
slims_cc$category <- row.names(slims_cc) #save rownames as category
```

Get mapped terms, using functions from Sam White's Biostars post.

```
#custom function from Sam White's, gets mapped ids for all of your query terms
#Write function mappedIds to get the query terms that mapped to the slim categories
mappedIds <-
  function(df, collection, OFFSPRING) #the command to run requires a dataframe of slim terms, like slim
  {
    map <- as.list(OFFSPRING[rownames(df)]) # Subset GOcollection offspring by the rownames of your dat
    mapped <- lapply(map, intersect, ids(collection)) #Find the terms that intersect between the subset
    df[["go_terms"]] <- vapply(unname(mapped), paste, collapse = ";", character(1L)) #Add column "go_te
    df #show resulting dataframe
  }
#Run function for MF and BP terms
BPslim <- mappedIds(slims_bp, BPGO_collection, GOBPOFFSPRING)
MFslim <- mappedIds(slims_mf, MFGO_collection, GOMFOFFSPRING)
CCslim <- mappedIds(slims_cc, CCGO_collection, GOCCOFFSPRING)
```

Remove duplicate matches, keeping the broader umbrella term

```
#filtering out duplicates

#BP
BPslim <- filter(BPslim, Count>0 & Term!="biological_process") #filter out empty slims and term "biolog
BPsplitted <- strsplit(as.character(BPslim$go_terms), ";") #split into multiple GO ids
BPslimX <- data.frame(Term = rep.int(BPslim$Term, sapply(BPsplitted, length)), go_term = unlist(BPsplit
BPslimX <- merge(BPslimX, BPslim[,c(1,3:4)], by="Term") #Add back counts, term, and category info
BPslimX <- unique(setDT(BPslimX)[order(go_term, -Count)], by = "go_term") #remove duplicate offspring t
BPslim <- data.frame(slim_term=BPslimX$Term, slim_cat=BPslimX$category, category=BPslimX$go_term) #rena
head(BPslim)
```

```
##                                slim_term   slim_cat   category
## 1                  immune system process GO:0002376 GO:0002523
## 2       anatomical structure development GO:0048856 GO:0003143
## 3       anatomical structure development GO:0048856 GO:0003428
## 4 cellular amino acid metabolic proce... GO:0006520 GO:0006431
## 5      sulfur compound metabolic process GO:0006790 GO:0006556
## 6 cellular amino acid metabolic proce... GO:0006520 GO:0006562
```

```
#MF
MFslim <- filter(MFslim, Count>0 & Term!="molecular_function") #filter out empty slims and term "molecu
MFsplitted <- strsplit(as.character(MFslim$go_terms), ";") #split into multiple GO ids
MFslimX <- data.frame(Term = rep.int(MFslim$Term, sapply(MFsplitted, length)), go_term = unlist(MFsplit
MFslimX <- merge(MFslimX, MFslim[,c(1,3:4)], by="Term")  #Add back counts, term, and category info
MFslimX <- unique(setDT(MFslimX)[order(go_term, -Count)], by = "go_term")  #remove duplicate offspring
MFslim <- data.frame(slim_term=MFslimX$Term, slim_cat=MFslimX$category, category=MFslimX$go_term) #rena
head(MFslim)
```

```
##            slim_term   slim_cat   category
## 1 catalytic activity GO:0003824 GO:0000254
## 2        DNA binding GO:0003677 GO:0000978
## 3        DNA binding GO:0003677 GO:0001162
## 4 catalytic activity GO:0003824 GO:0003839
## 5 catalytic activity GO:0003824 GO:0003841
## 6 catalytic activity GO:0003824 GO:0003865
```

```
#CC
CCslim <- filter(CCslim, Count>0 & Term!="cellular_component") #filter out empty slims and term "molecu
CCsplitted <- strsplit(as.character(CCslim$go_terms), ";") #split into multiple GO ids
CCslimX <- data.frame(Term = rep.int(CCslim$Term, sapply(CCsplitted, length)), go_term = unlist(CCsplit
CCslimX <- merge(CCslimX, CCslim[,c(1,3:4)], by="Term")  #Add back counts, term, and category info
CCslimX <- unique(setDT(CCslimX)[order(go_term, -Count)], by = "go_term")  #remove duplicate offspring
CCslim <- data.frame(slim_term=CCslimX$Term, slim_cat=CCslimX$category, category=CCslimX$go_term) #rena
head(CCslim)
```

```
##              slim_term   slim_cat   category
## 1            organelle GO:0043226 GO:0000808
## 2            organelle GO:0043226 GO:0001535
## 3 extracellular region GO:0005576 GO:0005602
## 4            organelle GO:0043226 GO:0005640
## 5            organelle GO:0043226 GO:0005680
## 6            organelle GO:0043226 GO:0005789
```

Save slim info with GO enrichment info for heatmap dataframes.

```
GO.BP <- right_join(BPslim, filter(enriched.GO.05, ontology=="BP"), by="category") #add back GO enrichm
GO.MF <- right_join(MFslim, filter(enriched.GO.05, ontology=="MF"), by="category") #add back GO enrichm
GO.CC <- right_join(CCslim, filter(enriched.GO.05, ontology=="CC"), by="category") #add back GO enrichm
```

Make heatmap

```
BPplot <- GO.BP %>% mutate(term = fct_reorder(term, -over_represented_pvalue)) %>%
    ggplot(aes(x = ontology, y = term)) +
    geom_tile(aes(fill=over_represented_pvalue, width = 1)) +
      scale_y_discrete(position = "right") +
      facet_grid(slim_term~ ., scales = "free_y", labeller = label_wrap_gen(width = 10, multi_line = TRU
      theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
      strip.text.y.left = element_text(angle=0, size = 20, face = "bold"),
      strip.text.x = element_text(size = 30, face = "bold"),
      axis.title = element_blank(),
      axis.text.x = element_text(size = 30),
      axis.text = element_text(size = 14), legend.title = element_text(size = 20), legend.text =
      element_text(size = 20))
```

```
MFplot <- GO.MF %>% mutate(term = fct_reorder(term, -over_represented_pvalue)) %>%
    ggplot(aes(x = ontology, y = term)) +
    geom_tile(aes(fill=over_represented_pvalue, width = 1)) +
      scale_y_discrete(position = "right") +
      facet_grid(slim_term~ ., scales = "free_y", labeller = label_wrap_gen(width = 10, multi_line = TRU
      theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
      strip.text.y.left = element_text(angle=0, size = 20, face = "bold"),
      strip.text.x = element_text(size = 30, face = "bold"),
      axis.title = element_blank(),
      axis.text.x = element_text(size = 30),
      axis.text = element_text(size = 14), legend.title = element_text(size = 20), legend.text =
      element_text(size = 20))
fig5 <- BPplot + MFplot

ggsave("RAnalysis/Output/Final_Figures/GO_enrichment_DEG_heatmap.pdf", fig5, width = 30, height = 40, un
```

Make supplemental table summarizing GO enrichment

```
GO.enrichment.dat <- bind_rows(GO.BP, GO.MF, GO.CC)
head(GO.enrichment.dat)
```

```
##                                slim_term   slim_cat   category
## 1                 immune system process GO:0002376 GO:0002523
## 2       anatomical structure development GO:0048856 GO:0003143
## 3       anatomical structure development GO:0048856 GO:0003428
## 4 cellular amino acid metabolic proce... GO:0006520 GO:0006431
## 5      sulfur compound metabolic process GO:0006790 GO:0006556
## 6 cellular amino acid metabolic proce... GO:0006520 GO:0006562
##   over_represented_pvalue under_represented_pvalue numDEInCat numInCat
## 1              0.02907688                0.9997158          1        3
## 2              0.01370737                0.9984245          3       60
## 3              0.01904549                0.9999089          1        2
## 4              0.03171970                0.9996194          1        4
## 5              0.04583636                0.9991478          1        5
## 6              0.02938333                0.9997097          1        3
##                                                                  term
## 1                    leukocyte migration involved in inflammatory response
## 2                                      embryonic heart tube morphogenesis
## 3 chondrocyte intercalation involved in growth plate cartilage morphogenesis
## 4                                         methionyl-tRNA aminoacylation
## 5                                  S-adenosylmethionine biosynthetic process
## 6                                             proline catabolic process
##   ontology
## 1       BP
## 2       BP
## 3       BP
## 4       BP
## 5       BP
## 6       BP
```

```
#Make dataframe of GO results for clustering and heatmap.
#add gene_IDs. To get gene_IDs we will merge with the GO.terms DF.
GOgenes <- data.frame(gene_id=annot_GO$gene_id, category=annot_GO$category)
GOgenes$gene_id <- as.character(GOgenes$gene_id) #make gene ID a character so we can collapse our many

GO.enrichment.summary  <- left_join(GO.enrichment.dat, GOgenes, by="category" ) #join the DFs

GO.enrichment.summary <- GO.enrichment.summary %>% #collapse and have gene IDs for a particular term in
  group_by(slim_term, slim_cat, category, over_represented_pvalue, under_represented_pvalue, numDEInCat
  summarise(genes = toString(gene_id)) %>% #rename collapsed gene_ID column "gene"
  ungroup()
```

```
## `summarise()` has grouped output by 'slim_term', 'slim_cat', 'category', 'over_represented_pvalue',
```

```
write.csv(GO.enrichment.summary, 'RAnalysis/Output/RNA-seq/GOSeq/GO.enrichment.summary.table.csv')
```

Exploring unique go_slim terms

```
#number of unique go slim terms per category
length(unique(GO.enrichment.summary$slim_term)) #47 unique go terms
```

```
## [1] 43
```

```
#view slim terms per ontology category
## filtering go slim category of BP
BP_slim <- GO.enrichment.summary %>%
  filter(ontology=="BP")
view(BP_slim)
#number of BP go slim terms
length(unique(BP_slim$slim_term)) #33
```

```
## [1] 33
```

```
## filtering go slim category of MF
MF_slim <- GO.enrichment.summary %>%
  filter(ontology=="MF")
view(MF_slim)
#number of BP go slim terms
length(unique(MF_slim$slim_term)) #12
```

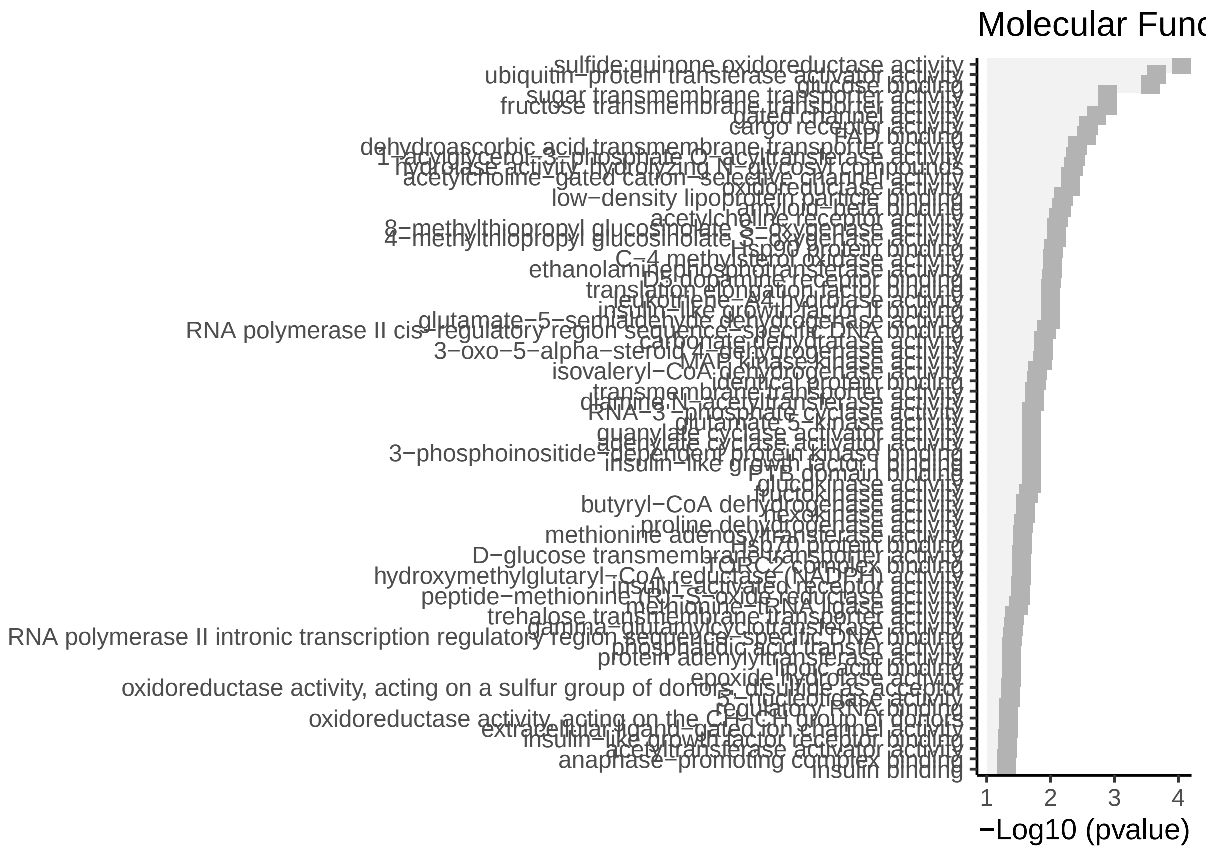
```
## [1] 8
```

```
## filtering go slim category of CC
CC_slim <- GO.enrichment.summary %>%
  filter(ontology=="CC")
view(CC_slim)

#number of CC go slim terms
length(unique(CC_slim$slim_term)) #4
```

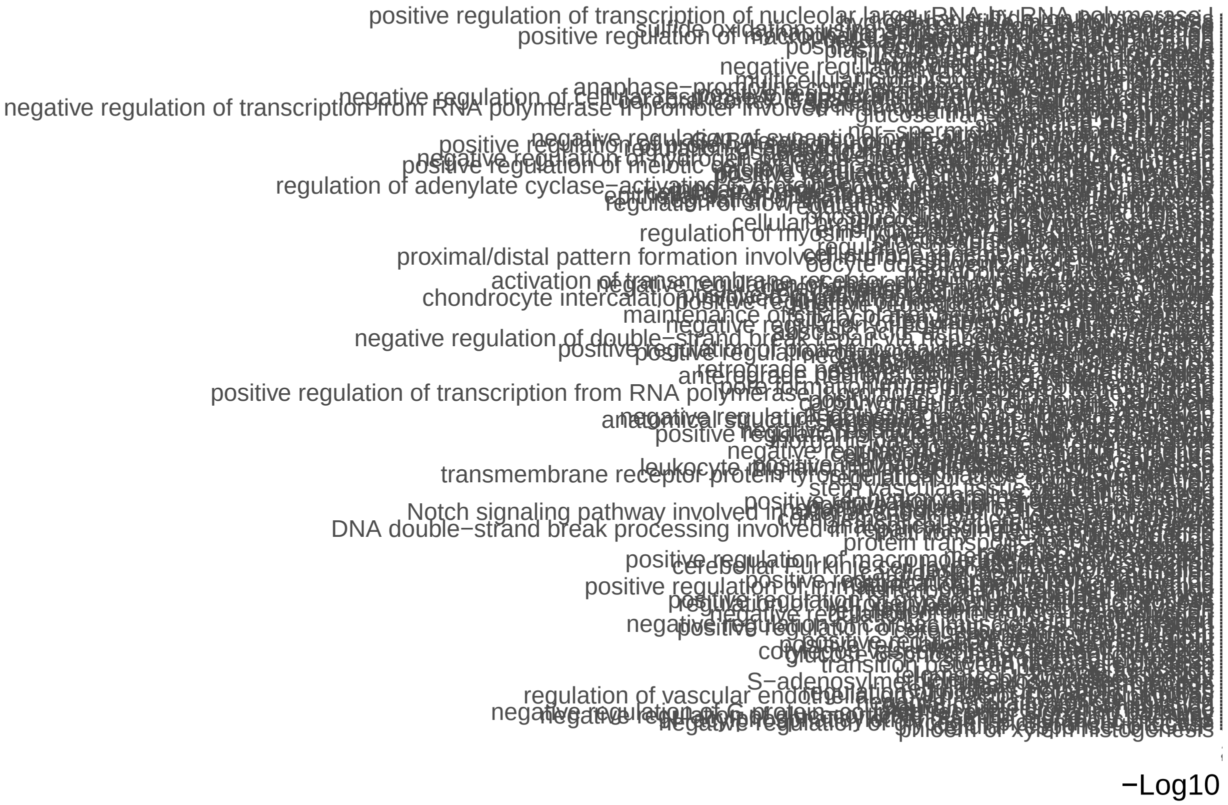
```
## [1] 4
```

Make terms by p value -log10 plots

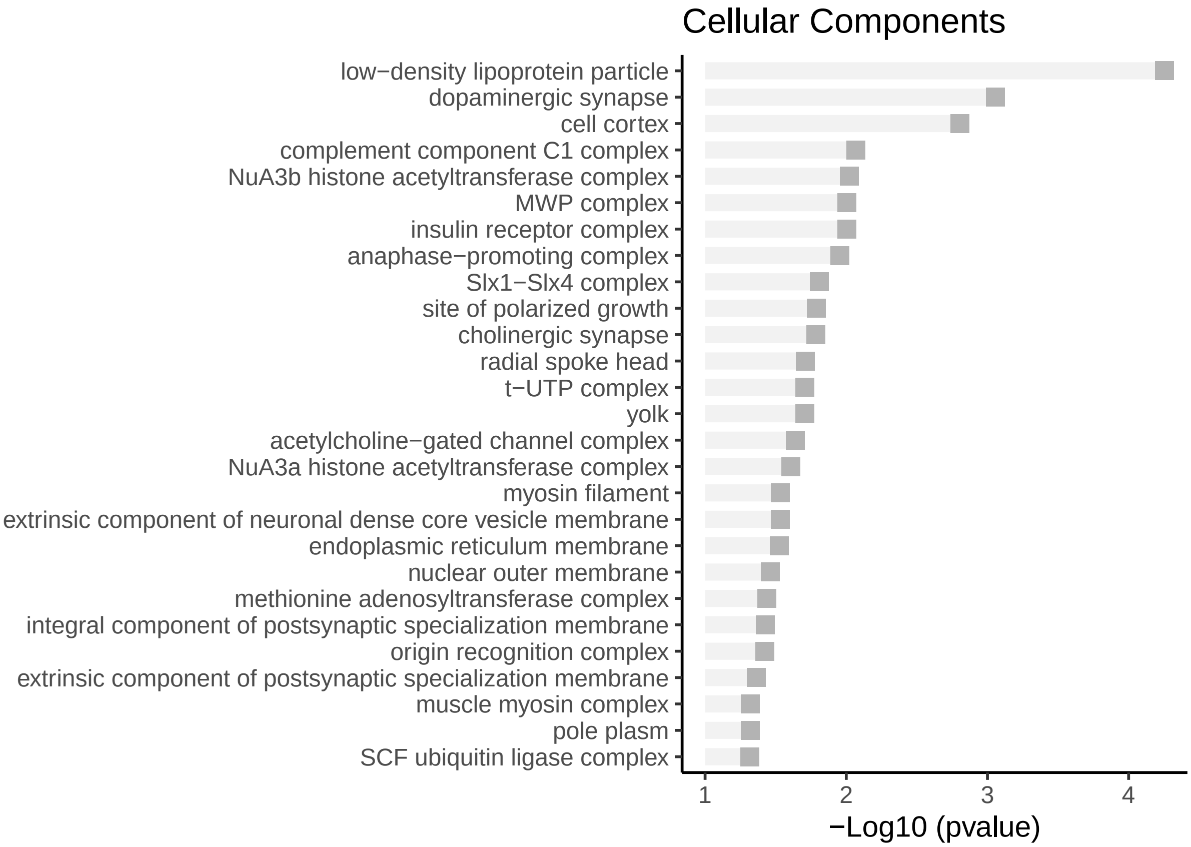
```
MF_plot_GO <- GO.MF  %>%  mutate(term = fct_reorder(term, -log10(over_represented_pvalue))) %>%
                  #mutate(term = fct_reorder(term, Day)) %>%
                  ggplot(aes(x=term, y=-log10(over_represented_pvalue))) +
                  geom_segment( aes(x=term ,xend=term, y=1, yend=-log10(over_represented_pvalue)), colo
                  geom_point(size=3, shape = 15, colour = "grey70") +
                  coord_flip() +
                  theme(
                    panel.grid.minor.y = element_blank(),
                    panel.grid.major.y = element_blank(),
                    legend.position="bottom"
                  ) +
                  xlab("") +
                  ylab("-Log10 (pvalue)") +
                  ggtitle("Molecular Function") +
                  theme_bw() + #Set background color
  #facet_wrap(~slim_term) +
  theme(axis.text.y = element_text(hjust = 1))   +
                  theme(panel.border = element_blank(), # Set border
                        panel.grid.major = element_blank(), #Set major gridlines
                        panel.grid.minor = element_blank(), #Set minor gridlines
                        axis.line = element_line(colour = "black"), #Set axes color
                        plot.background=element_blank())  #Set the plot background #set title attribute

MF_plot_GO
```



```
ggsave("RAnalysis/Output/Final_Figures/GO.plot.MF.terms.pdf", MF_plot_GO , width = 10, height = 10, uni

BP_plot_GO <- GO.BP  %>%  mutate(term = fct_reorder(term, -log10(over_represented_pvalue))) %>%
                  #mutate(term = fct_reorder(term, Day)) %>%
                  ggplot(aes(x=term, y=-log10(over_represented_pvalue))) +
                  geom_segment( aes(x=term ,xend=term, y=1, yend=-log10(over_represented_pvalue)), colo
                  geom_point(size=3, shape = 15, colour = "grey70") +
                  coord_flip() +
                  theme(
                    panel.grid.minor.y = element_blank(),
                    panel.grid.major.y = element_blank(),
                    legend.position="bottom"
                  ) +
                  xlab("") +
                  ylab("-Log10 (pvalue)") +
                  ggtitle("Biological Processes") +
                  theme_bw() + #Set background color
  #facet_wrap(~slim_term) +
  theme(axis.text.y = element_text(hjust = 1))   +
                  theme(panel.border = element_blank(), # Set border
                        panel.grid.major = element_blank(), #Set major gridlines
                        panel.grid.minor = element_blank(), #Set minor gridlines
                        axis.line = element_line(colour = "black"), #Set axes color
                        plot.background=element_blank())  #Set the plot background #set title attribute

BP_plot_GO
```



```
ggsave("RAnalysis/Output/Final_Figures/GO.plot.BP.terms.pdf", BP_plot_GO , width = 10, height = 30, uni

CC_plot_GO <- GO.CC  %>%  mutate(term = fct_reorder(term, -log10(over_represented_pvalue))) %>%
                  #mutate(term = fct_reorder(term, Day)) %>%
                  ggplot(aes(x=term, y=-log10(over_represented_pvalue))) +
                  geom_segment( aes(x=term ,xend=term, y=1, yend=-log10(over_represented_pvalue)), colo
                  geom_point(size=3, shape = 15, colour = "grey70") +
                  coord_flip() +
                  theme(
                    panel.grid.minor.y = element_blank(),
                    panel.grid.major.y = element_blank(),
                    legend.position="bottom"
                  ) +
                  xlab("") +
                  ylab("-Log10 (pvalue)") +
                  ggtitle("Cellular Components") +
                  theme_bw() + #Set background color
  #facet_wrap(~slim_term) +
  theme(axis.text.y = element_text(hjust = 1))   +
                  theme(panel.border = element_blank(), # Set border
                        panel.grid.major = element_blank(), #Set major gridlines
                        panel.grid.minor = element_blank(), #Set minor gridlines
                        axis.line = element_line(colour = "black"), #Set axes color
                        plot.background=element_blank())  #Set the plot background #set title attribute

CC_plot_GO
```



```
ggsave("RAnalysis/Output/Final_Figures/GO.plot.CC.terms.pdf", CC_plot_GO , width = 10, height = 10, uni
```