

Flattened circadian glucocorticoid oscillations cause obesity due to increased lipid turnover and lipid uptake

Tholen S., Kovary K.M., Rabiee A., Bielczyk-Maczyńska E., Yang W., and Teruel M.N.

Updated: 2020-02-06 00:47:21

Introduction

This is a summary of the RNA-seq analysis between the BAT and WAT tissues that were treated with cort vs. control pellets. I decided to include all of the code in this markdown document in case any of you have questions about how things were done or if you'd like to ask me specifics about any part of the analysis it might help direct your question. It might look like just a bunch of code at first, but there are plenty of plots and some interpretation as well.

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
library(tidyverse)
library(sleuth)
library(readxl)
library(pheatmap)
library(RColorBrewer)
library(kableExtra)
library(eulerr)
library(ggrepel)
library(clusterProfiler)
library(org.Mm.eg.db)
library(ggrepel)
```

Sample Metadata Import

This code takes in the experimental details (treatment, lane, tissue, etc.) which will be used to both import and model the data.

```
# Set up metadata table for WAT data
# In previous analysis (see WAT_Analysis.Rmd) samples 8,9,43 (mice 211, 223, 229)
# were determined to be outliers and will be excluded

wat_sample_list = read_xlsx(
  '/Users/kylekovary/Box Sync/kkovary/R Projects/RNaseq/Stefan RNA-seq WAT analysis/Stefan Tholen_Samples_for_RNaseq_WAT.xlsx'
) %>%
  dplyr::mutate(fat = 'WAT', lane = paste0(lane, '_WAT')) %>%
  dplyr::rename(sample_name = `Sample Name`) %>%
  unite('sample_name', c(sample_name, fat)) %>%
  dplyr::select(lane, mouse, sample_name)

path = list.dirs(
  '/Users/kylekovary/Box Sync/kkovary/R Projects/RNaseq/Stefan RNA-seq WAT analysis/kallisto_output'
)[c(1, 2)]
path = path %>% as_tibble() %>%
  separate(
    col = value,
    into = c('a', 'mouse'),
    sep = 'ST',
    remove = F
  ) %>% dplyr::select(value, mouse) %>%
  separate(
    col = mouse,
    into = c('b', 'mouse'),
```

```

sep = '-',
remove = T
) %>% dplyr::select(value, mouse)

s2c_wat = bind_cols(wat_sample_list, path[match(wat_sample_list$mouse, path$mouse), 1]) %>%
dplyr::rename(path = value) %>%
separate(
  sample_name,
  into = c('condition', 'delivery', 'day', 'replicate', 'depot'),
  remove = F
) %>%
unite('treatment', c('condition', 'delivery'), remove = F) %>%
dplyr::rename(sample = sample_name) %>%
dplyr::filter(!mouse %in% c(211, 223, 229)) #%>%
#unite('Treatment',c('Pellet','Delivery'), remove = F)

# Set up BAT metadata table
bat_sample_list = read_xlsx(
  '/Users/kylekovary/Box Sync/kkovary/R Projects/RNAseq/Stefan RNA-seq BAT analysis/Stefan Tholen_Samples_for_RNAseq_BAT.xlsx'
) %>%
mutate(fat = 'BAT', lane = paste0(lane, '_BAT')) %>%
dplyr::rename(sample_name = 'Sample Name') %>%
unite('sample_name', c(sample_name, fat)) %>%
dplyr::select(lane, mouse, sample_name)

path = list.dirs(
  '/Users/kylekovary/Box Sync/kkovary/R Projects/RNAseq/Stefan RNA-seq BAT analysis/kallisto_output/'
)[-c(1, 2)]
path = path %>% as_tibble() %>%
tidyrr::separate(
  col = value,
  into = c('a', 'mouse'),
  sep = 'ST-',
  remove = F
) %>%
dplyr::select(value, mouse)

s2c_bat = bind_cols(bat_sample_list, path[match(bat_sample_list$mouse, path$mouse), 1]) %>%
dplyr::rename(path = value) %>%
separate(
  sample_name,
  into = c('condition', 'delivery', 'day', 'replicate', 'depot'),
  remove = F
) %>%
unite('treatment', c('condition', 'delivery'), remove = F) %>%
dplyr::rename(sample = sample_name)

s2c <- rbind(s2c_wat, s2c_bat)

s2c_wat <- s2c %>% dplyr::filter(depot == 'WAT', treatment %in% c('Cort_pellet','Placebo_pellet'))
s2c_bat<- s2c %>% dplyr::filter(depot == 'BAT', treatment %in% c('Cort_pellet','Placebo_pellet'))

```

Gene ID import

This function reads in a bunch of different accession numbers (ensembl, gene name, transcript name, etc.) which is useful for going between accession numbers required for some analysis as well as making plots with human readable gene names.

```

# Retrieve GeneNames
library(biomaRt)
mart <- biomaRt::useMart(biomart = "ENSEMBL_MART_ENSEMBL",
                          dataset = "mmusculus_gene_ensembl",
                          host = "dec2015.archive.ensembl.org")

# host = "ensembl.org")
t2g <- biomaRt::getBM(
  attributes = c(
    "ensembl_transcript_id",
    "transcript_version",
    "ensembl_gene_id",
    "external_gene_name",
    "description",
    "transcript_biotype",
    'external_transcript_name'
  ),
  mart = mart
)
```

```

)
t2g <- dplyr::rename(t2g,
  target_id = ensembl_transcript_id,
  ens_gene = ensembl_gene_id,
  ext_gene = external_gene_name)

```

Import data for WAT and BAT cort pellet vs placebo pellet conditions

These functions read in the Kallisto files, normalize the data, and store it in a structure that is called a “sleuth object”. They take a long time to run so I don’t actually run them here and instead load in saved sleuth objects in the next step.

```

# WAT cort pellet vs placebo pellet
so_wat <- sleuth_prep(
  s2c_wat,
  target_mapping = t2g,
  aggregation_column = 'ext_gene',
  extra_bootstrap_summary = TRUE
)

# BAT cort pellet vs placebo pellet
so_bat <- sleuth_prep(
  s2c_bat,
  target_mapping = t2g,
  aggregation_column = 'ext_gene',
  extra_bootstrap_summary = TRUE
)

sleuth_save(so_wat, '~/Documents/GitHub/ShinyApps/InVivoRNAseq/data/so_wat')
sleuth_save(so_bat, '~/Documents/GitHub/ShinyApps/InVivoRNAseq/data/so_bat')

```

Fit differential expression models for WAT and BAT cort pellet vs placebo pellet conditions

This is the meat of the analysis and is based on what I showed in lab meeting this week. The steps are roughly:

1. Fit likelihood ratio test using a full model (treatment and lane) and compare that to a reduced model (lane only).
 - this generates a q-value (FDR adjusted p-value) that is used down stream for significance testing
2. Fit a Wald test comparing Cort vs Placebo pellet data. I am not actually using this for downstream analysis, but I was initially interested in it since it creates a beta value that is comparable to a fold change value and wanted it around in case it was useful. It turned out to correlate strongly with fold change (as it should) though the dynamic range was lower and tended to scrunch things up.
3. Calculate a fold change value between Cort and Placebo for downstream analysis
4. Combine all of these values together into a single data structure

This is done for both WAT and BAT, as well as for transcript level analysis and gene level analysis.

```

#####
# Import sleuth objects #
#####

so_wat <- sleuth_load('~/Documents/GitHub/ShinyApps/InVivoRNAseq/data/so_wat')
so_bat <- sleuth_load('~/Documents/GitHub/ShinyApps/InVivoRNAseq/data/so_bat')

#####
# WAT Transcript level comparison #
#####

# LRT w/ lane
full_design <- model.matrix(formula(~ treatment + lane),
  data = s2c_wat,
  contrasts.arg = list(treatment = c(1,0)))

```

```

so_wat <- sleuth_fit(so_wat,
                      formula = full_design,
                      gene_mode = TRUE,
                      fit_name = "full")
so_wat <- sleuth_fit(so_wat,
                      formula = ~ s2c_wat$lane,
                      gene_mode = TRUE,
                      fit_name = "reduced")

so_wat <- sleuth_lrt(so_wat, "reduced", "full")
lrt_results <- sleuth_results(so_wat,
                               'reduced:full',
                               test_type = 'lrt',
                               pval_aggregate = FALSE) %>%
  dplyr::rename(lrt_qval = qval)
#table(lrt_results[, "lrt_qval"] < 0.05)

# Wald Test
#models(so_wat)

so_wat <- sleuth_wt(so_wat, which_beta = 'treatment1')
wt_results <- sleuth_results(so_wat,
                               'treatment1',
                               test_type = 'wt',
                               pval_aggregate = FALSE) %>%
  dplyr::rename(wt_qval = qval)
#table(wt_results[, "wt_qval"] < 0.05)

# Fold change / adjusted pval
so_wat_ttest <- kallisto_table(so_wat) %>% dplyr::group_by(target_id) %>%
  dplyr::summarise(fc = mean(est_counts[treatment == 'Cort_pellet'], na.rm = T) /
    mean(est_counts[treatment == 'Placebo_pellet'], na.rm = T))

# Join 3 methods
joined_wat_results <- dplyr::full_join(
  dplyr::select(lrt_results,
                target_id,
                ens_gene,
                ext_gene,
                external_transcript_name,
                lrt_qval),
  dplyr::select(so_wat_ttest,
                target_id,
                fc),
  by = "target_id"
) %>%
  dplyr::full_join(
    dplyr::select(wt_results,
                  target_id,
                  b,
                  wt_qval),
    by = 'target_id'
) %>% dplyr::as_tibble() %>%
  dplyr::select(target_id, ens_gene, ext_gene, external_transcript_name, fc, b, everything()) %>%
  dplyr::mutate(Direction = ifelse(lrt_qval < 0.05 & log2(fc) > 0,
                                    'Up',
                                    ifelse(lrt_qval < 0.05 & log2(fc) < 0,
                                          'Down',
                                          'Unchanged')))

#####
# WAT Gene Level Aggregation #
#####

# LRT w/ lane

so_wat <- sleuth_fit(so_wat,
                      formula = full_design,
                      gene_mode = TRUE,
                      fit_name = "full")
so_wat <- sleuth_fit(so_wat,
                      formula = ~ s2c_wat$lane,
                      gene_mode = TRUE,
                      fit_name = "reduced")

so_wat <- sleuth_lrt(so_wat, "reduced", "full")
lrt_results <- sleuth_results(so_wat,
                               'reduced:full',
                               test_type = 'lrt',
                               pval_aggregate = TRUE) %>%

```

```

dplyr::rename(lrt_qval = qval)
#table(lrt_results[, "lrt_qval"] < 0.05)

# Wald Test
#models(so_wat)

so_wat <- sleuth_wt(so_wat, which_beta = 'treatment1')
wt_results <- sleuth_results(so_wat,
  'treatment1',
  test_type = 'wt',
  pval_aggregate = TRUE) %>%
dplyr::rename(wt_qval = qval)
#table(wt_results[, "wt_qval"] < 0.05)

# Fold change / adjusted pval
so_wat_ttest <- kallisto_table(so_wat) %>%
dplyr::mutate(target_id = so_wat$target_mapping[match(target_id, so_wat$target_mapping$target_id), 'ext_gene']) %>%
dplyr::group_by(target_id) %>%
dplyr::summarise(fc = mean(est_counts[treatment == 'Cort_pellet'], na.rm = T) /
mean(est_counts[treatment == 'Placebo_pellet'], na.rm = T))

# Join 3 methods
joined_wat_gene_results <- dplyr::full_join(
  dplyr::select(lrt_results,
    target_id,
    lrt_qval),
  dplyr::select(so_wat_ttest,
    target_id,
    fc),
  by = "target_id")
) %>%
dplyr::full_join(
  dplyr::select(wt_results,
    target_id,
    wt_qval),
  by = 'target_id'
) %>% as_tibble() %>% filter(!duplicated(target_id)) %>%
dplyr::select(target_id, fc, everything()) %>%
dplyr::mutate(Direction = ifelse(lrt_qval < 0.05 & log2(fc) > 0,
  'Up',
  ifelse(lrt_qval < 0.05 & log2(fc) < 0,
  'Down',
  'Unchanged')))

#####
# BAT Transcript level comparison #
#####

# LRT w/ lane
full_design <- model.matrix(formula(~ treatment + lane),
  data = s2c_bat,
  contrasts.arg = list(treatment = c(1,0)))
so_bat <- sleuth_fit(so_bat,
  formula = full_design,
  #gene_mode = TRUE,
  fit_name = "full")
so_bat <- sleuth_fit(so_bat,
  formula = ~ s2c_bat$lane,
  #gene_mode = TRUE,
  fit_name = "reduced")

so_bat <- sleuth_lrt(so_bat, "reduced", "full")
lrt_results <- sleuth_results(so_bat,
  'reduced:full',
  test_type = 'lrt',
  pval_aggregate = FALSE) %>%
dplyr::rename(lrt_qval = qval)
#table(lrt_results[, "lrt_qval"] < 0.05)

# Wald Test
#models(so_bat)

so_bat <- sleuth_wt(so_bat, which_beta = 'treatment1')
wt_results <- sleuth_results(so_bat,
  'treatment1',
  test_type = 'wt',
  pval_aggregate = FALSE) %>%
dplyr::rename(wt_qval = qval)
#table(wt_results[, "wt_qval"] < 0.05)

```

```

# Fold change / adjusted pval
so_bat_ttest <- kallisto_table(so_bat) %>%
  dplyr::group_by(target_id) %>%
  dplyr::summarise(fc = mean(est_counts[treatment == 'Cort_pellet'], na.rm = T) /
    mean(est_counts[treatment == 'Placebo_pellet'], na.rm = T))

# Join 3 methods
joined_bat_results <- dplyr::full_join(
  dplyr::select(lrt_results,
    target_id,
    ens_gene,
    ext_gene,
    external_transcript_name,
    lrt_qval),
  dplyr::select(so_bat_ttest,
    target_id,
    fc),
  by = "target_id")
) %>%
  dplyr::full_join(
    dplyr::select(wt_results,
      target_id,
      b,
      wt_qval),
    by = 'target_id')
) %>% dplyr::as_tibble() %>%
dplyr::select(target_id, ens_gene, ext_gene, external_transcript_name, fc, b, everything()) %>%
dplyr::mutate(Direction = ifelse(lrt_qval < 0.05 & log2(fc) > 0,
  'Up',
  ifelse(lrt_qval < 0.05 & log2(fc) < 0,
    'Down',
    'Unchanged')))

#####
## BAT Gene Level Aggrgation #
#####
# LRT w/ lane

so_bat <- sleuth_fit(so_bat,
  formula = full_design,
  gene_mode = TRUE,
  fit_name = "full")
so_bat <- sleuth_fit(so_bat,
  formula = ~ s2c_bat$lane,
  gene_mode = TRUE,
  fit_name = "reduced")

so_bat <- sleuth_lrt(so_bat, "reduced", "full")
lrt_results <- sleuth_results(so_bat,
  'reduced:full',
  test_type = 'lrt',
  pval_aggregate = TRUE) %>%
  dplyr::rename(lrt_qval = qval)
#table(lrt_results[, "lrt_qval"] < 0.05)

# Wald Test
#models(so_bat)

so_bat <- sleuth_wt(so_bat, which_beta = 'treatment1')
wt_results <- sleuth_results(so_bat,
  'treatment1',
  test_type = 'wt',
  pval_aggregate = TRUE) %>%
  dplyr::rename(wt_qval = qval)
#table(wt_results[, "wt_qval"] < 0.05)

# Fold change / adjusted pval
so_bat_ttest <- kallisto_table(so_bat) %>%
  dplyr::mutate(target_id = so_bat$target_mapping[match(target_id, so_bat$target_mapping$target_id), 'ext_gene']) %>%
  dplyr::group_by(target_id) %>%
  summarise(fc = mean(est_counts[treatment == 'Cort_pellet'], na.rm = T) /
    mean(est_counts[treatment == 'Placebo_pellet'], na.rm = T))

# Join 3 methods
joined_bat_gene_results <- dplyr::full_join(
  dplyr::select(lrt_results,
    target_id,
    lrt_qval),
  dplyr::select(so_bat_ttest,
    target_id,
    ext_gene))

```

```

            fc),
  by = "target_id"
) %>%
dplyr::full_join(
  dplyr::select(wt_results,
    target_id,
    wt_qval),
  by = 'target_id'
) %>% as_tibble() %>% filter(!duplicated(target_id)) %>%
dplyr::select(target_id, fc, everything()) %>%
mutate(Direction = ifelse(lrt_qval < 0.05 & log2(fc) > 0,
  'Up',
  ifelse(lrt_qval < 0.05 & log2(fc) < 0,
  'Down',
  'Unchanged')))
```

Volcano Plots

WAT Transcript level

```

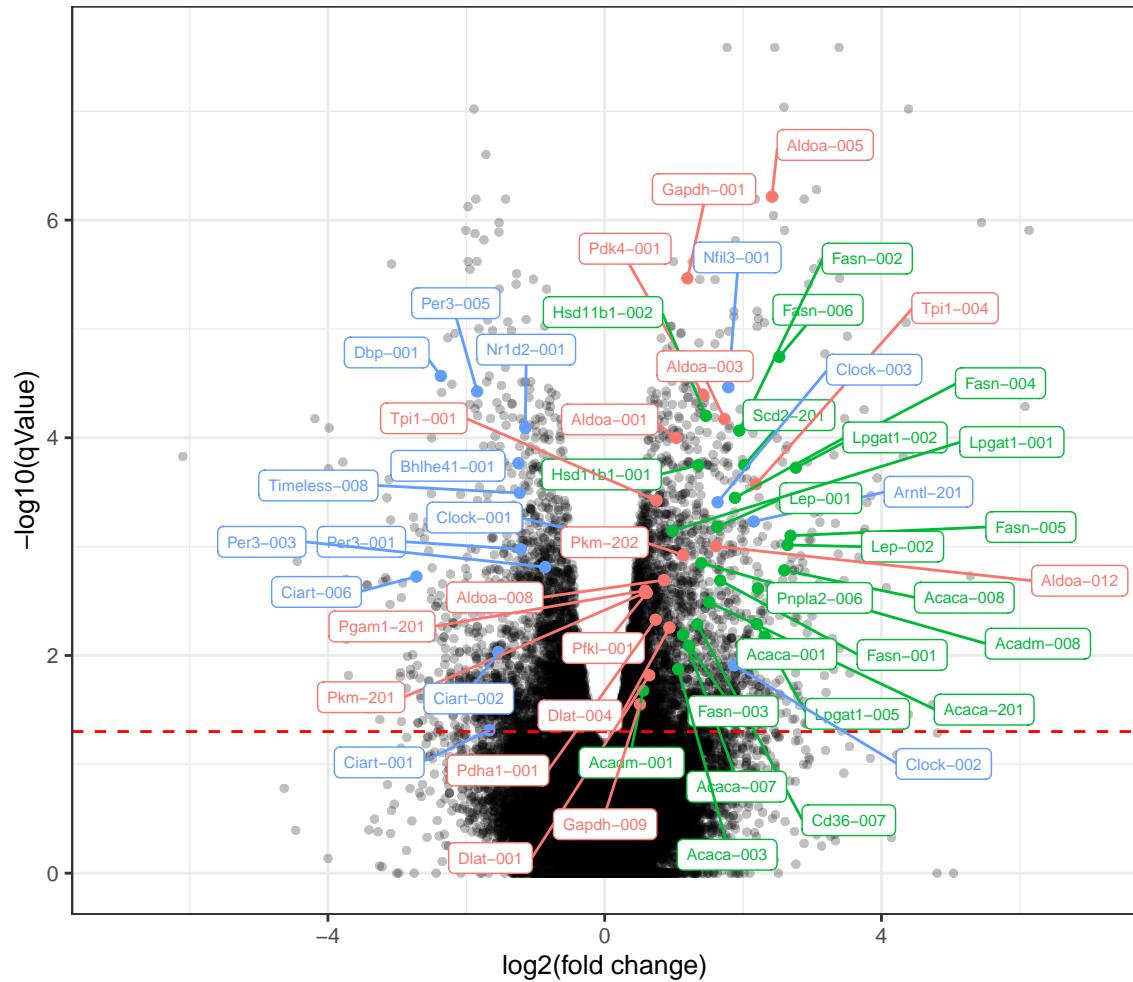
#####
# WAT Transcript level #
#####
gly_enz <- c('Pfkl', 'Aldoa', 'Tpi1', 'Gapdh', 'Pgami', 'Pkm', 'Pdh1', 'Dlat', 'Pdk4')
lipid_met <- c('Acaca', 'Fasn', 'Scd1', 'Scd2', 'Lpgat1', 'Acadm', 'Pnpla2', 'Lep', 'Hsdi1b1', 'Cd36')
circ_rhy <- c('Clock', 'Arntl', 'Per3', 'Ccart', 'Timeless', 'Nfil3', 'Nr1d2', 'Bhlhe41', 'Dbp')

joined_wat_results <-
joined_wat_results %>% mutate(Category = ifelse(
  ext_gene %in% gly_enz,
  'Glycolytic enzymes and regulators',
  ifelse(
    ext_gene %in% lipid_met,
    'Lipid metabolism',
    ifelse(ext_gene %in% circ_rhy,
      'Circadian rhythm',
      NA)
  )
)) %>%
mutate(Category = factor(Category,
  levels = c('Glycolytic enzymes and regulators',
  'Lipid metabolism',
  'Circadian rhythm')))

ggplot(filter(joined_wat_results, !is.na(Direction)),
  aes(x = log2(fc),
  y = -log10(lrt_qval))) +
geom_point(alpha = 0.25,
  stroke = 0,
  data = filter(joined_wat_results,
    !is.na(Direction),
    is.na(Category))) +
geom_point(alpha = 1,
  size = 2,
  stroke = 0,
  data = filter(joined_wat_results,
    lrt_qval < 0.05,
    !is.na(Category)),
  aes(color = Category)) +
geom_hline(yintercept = -log10(0.05),
  color = 'red',
  linetype = 'dashed') +
geom_label_repel(data = filter(joined_wat_results,
  !is.na(Category),
  lrt_qval < 0.05),
  aes(label = external_transcript_name,
  color = Category),
  size = 2,
  force = 20) +
xlim(-7,7) +
#scale_color_manual(values = c('#4393c3', '#878787', '#d6604d')) +
xlab('log2(fold change)') +
ylab('-log10(qValue)') +
ggtitle('WAT Cort Pellet vs Placebo Pellet Transcripts') +
theme_bw() +
```

```
theme(text = element_text(size=10),
      legend.position="bottom")
```

WAT Cort Pellet vs Placebo Pellet Transcripts



Volcano plot of differentially expressed transcripts between cort vs placebo pellet WAT. Genes from Stefan's figure 6D are highlighted here.

WAT Gene level

```
#####
# WAT Gene level #
#####

gly_enz <- c('Pfk1', 'Aldoa', 'Tpi1', 'Gapdh', 'Pgamt', 'Pkm', 'Pdhai', 'Dlat', 'Pdk4')
lipid_met <- c('Acaca', 'Fasn', 'Scd1', 'Scd2', 'Lpgat1', 'Acadm', 'Pnpla2', 'Lep', 'Hsd11b1', 'Cd36')
circ_rhy <- c('Clock', 'Arntl', 'Per3', 'Ciart', 'Timeless', 'Nfil3', 'Nr1d2', 'Bhlhe41', 'Dbp')

selected_genes = tibble(genes = c(gly_enz, lipid_met, circ_rhy),
                       order = 1:28)

joined_wat_gene_results <-
  joined_wat_gene_results %>%
  mutate(Category = ifelse(
```

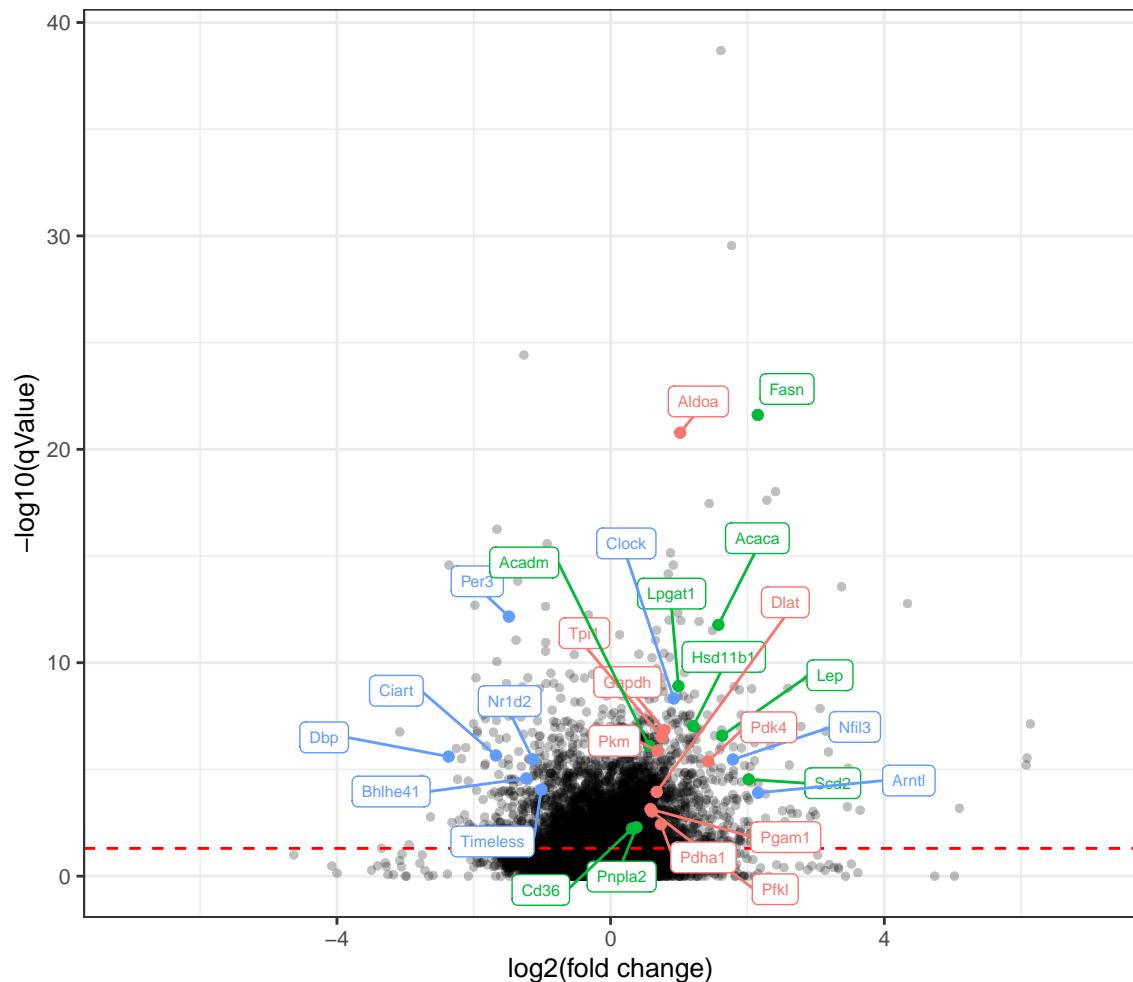
```

target_id %in% gly_enz,
'Glycolytic enzymes and regulators',
ifelse(
  target_id %in% lipid_met,
  'Lipid metabolism',
  ifelse(target_id %in% circ_rhy,
    'Circadian rhythm',
    NA)
  )) %>%
mutate(Category = factor(Category,
  levels = c('Glycolytic enzymes and regulators',
            'Lipid metabolism',
            'Circadian rhythm')))

joined_wat_gene_results %>% filter(!is.na(Direction)) %>%
ggplot(., 
  aes(x = log2(fc),
      y = -log10(lrt_qval))) +
  geom_point(alpha = 0.25,
             stroke = 0,
             data = filter(joined_wat_gene_results,
                           !is.na(Direction),
                           is.na(Category))) +
  geom_point(alpha = 1,
             size = 2,
             stroke = 0,
             data = filter(joined_wat_gene_results,
                           lrt_qval < 0.05,
                           is.na(Category)),
             aes(color = Category)) +
  geom_hline(yintercept = -log10(0.05),
             color = 'red',
             linetype = 'dashed') +
  geom_label_repel(data = filter(joined_wat_gene_results,
                                 !is.na(Category),
                                 lrt_qval < 0.05),
                   aes(label = target_id,
                       color = Category),
                   size = 2,
                   force = 20) +
  xlim(-7,7) +
  #scale_color_manual(values = c('#4393c3', '#878787', '#d6604d')) +
  xlab('log2(fold change)') +
  ylab('-log10(qValue)') +
  ggtitle('WAT Cort Pellet vs Placebo Pellet - Gene level') +
  theme_bw() +
  theme(text = element_text(size=10),
        legend.position="bottom")

```

WAT Cort Pellet vs Placebo Pellet – Gene level



Category a Glycolytic enzymes and regulators a Lipid metabolism a Circadian rhythm

Volcano plot of deferentially expressed genes between cort vs placebo pellet WAT. Genes from Stefan's figure 6D are highlighted here.

BAT Transcript level

```
#####
# BAT Transcript level #
#####

bat_selected <- c('Atp7a', 'Cacna1a', 'Cacna1c', 'Cacna1s', 'Cacng6',
'Kcna5', 'Kcnab3', 'Kcncl', 'Kcnj15', 'Kcnq1', 'Scn5a',
'Slc8a3', 'Slc9a2', 'Slc9a7', 'Slc39a14')

selected_genes = tibble(genes = c(gly_enz, lipid_met, circ_rhy),
order = 1:28)

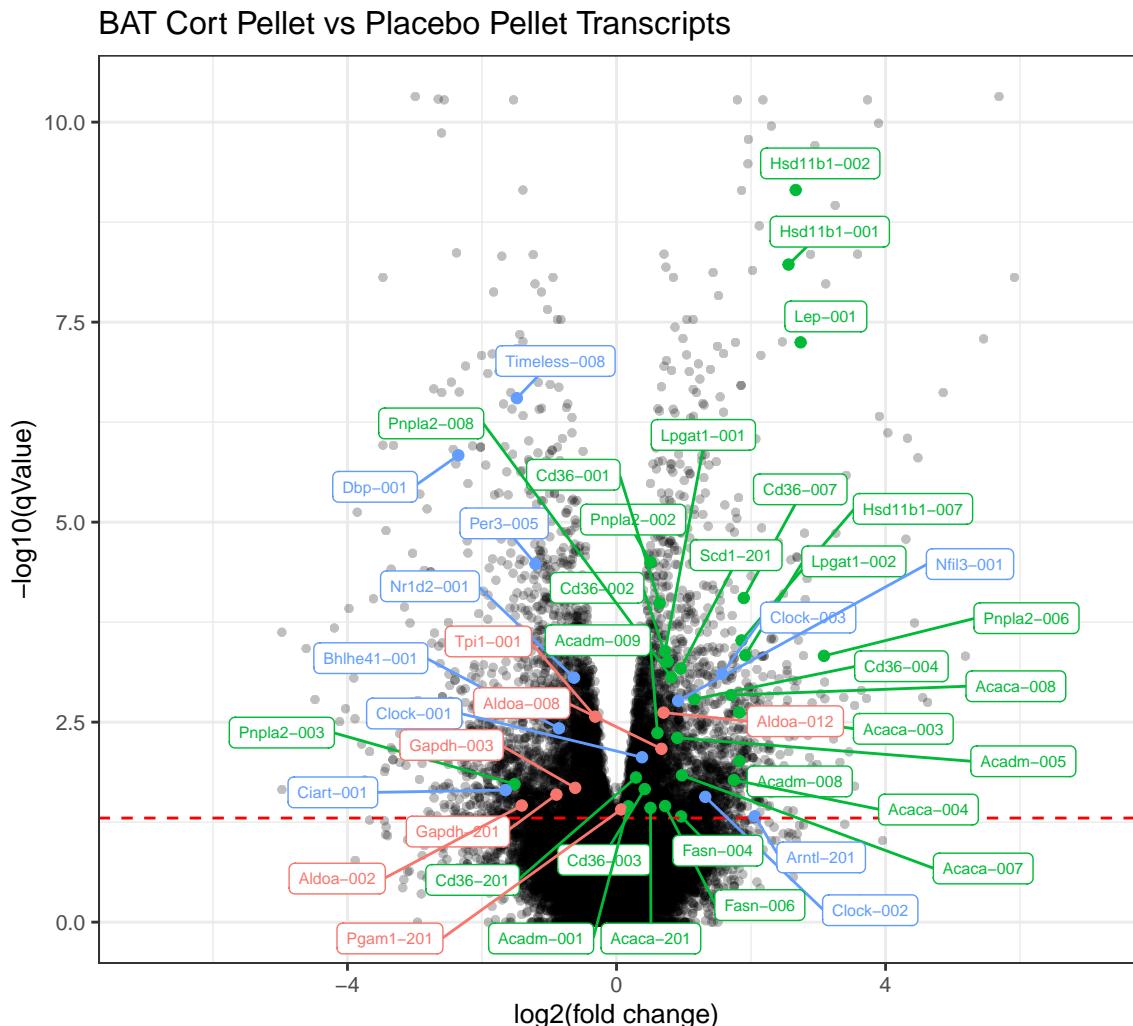
joined_bat_results <-
joined_bat_results %>% mutate(Category = ifelse(
ext_gene %in% gly_enz,
'Glycolytic enzymes and regulators',
ifelse(
ext_gene %in% lipid_met,
'Lipid metabolism',
```

```

  ifelse(ext_gene %in% circ_rhy,
        'Circadian rhythm',
        NA)
    )
  )) %>%
mutate(Category = factor(Category,
                          levels = c('Glycolytic enzymes and regulators',
                                    'Lipid metabolism',
                                    'Circadian rhythm')))

ggplot(filter(joined_bat_results, !is.na(Direction)),
       aes(x = log2(fc),
           y = -log10(lrt_qval))) +
  geom_point(alpha = 0.25,
             stroke = 0,
             data = filter(joined_bat_results,
                           !is.na(Direction),
                           is.na(Category))) +
  geom_point(alpha = 1,
             size = 2,
             stroke = 0,
             data = filter(joined_bat_results,
                           lrt_qval < 0.05,
                           !is.na(Category)),
             aes(color = Category)) +
  geom_hline(yintercept = -log10(0.05),
             color = 'red',
             linetype = 'dashed') +
  geom_label_repel(data = filter(joined_bat_results,
                                 !is.na(Category),
                                 lrt_qval < 0.05),
                   aes(label = external_transcript_name,
                       color = Category),
                   size = 2,
                   force = 20) +
  xlim(-7,7) +
  #scale_color_manual(values = c('#4393c3', '#878787', '#d6604d')) +
  xlab('log2(fold change)') +
  ylab('-log10(qValue)') +
  ggtitle('BAT Cort Pellet vs Placebo Pellet Transcripts') +
  theme_bw() +
  theme(text = element_text(size=10),
        legend.position="bottom")

```



Category a Glycolytic enzymes and regulators a Lipid metabolism a Circadian rhythm

Volcano plot of differentially expressed transcripts between cort vs placebo pellet BAT. Genes from Stefan's figure 6D are highlighted here.

BAT Gene Level

```

#####
# BAT Gene level #
#####

gly_enz <- c('Pfk1', 'Aldoa', 'Tp11', 'Gapdh', 'Pgami', 'Pkm', 'Pdhai', 'Dlat', 'Pdk4')
lipid_met <- c('Acaca', 'Fasn', 'Scd1', 'Scd2', 'Lpgat1', 'Acadm', 'Pnpla2', 'Lep', 'Hsdi1ib1', 'Cd36')
circ_rhy <- c('Clock', 'Arntl', 'Per3', 'Ciart', 'Timeless', 'Nfil3', 'Nr1d2', 'Blhhe41', 'Dbp')

selected_genes = tibble(genes = c(gly_enz, lipid_met, circ_rhy),
                        order = 1:28)

joined_bat_gene_results <-
  joined_bat_gene_results %>%
  mutate(Category = ifelse(
    target_id %in% gly_enz,
    'Glycolytic enzymes and regulators',
    ifelse(
      target_id %in% lipid_met,

```

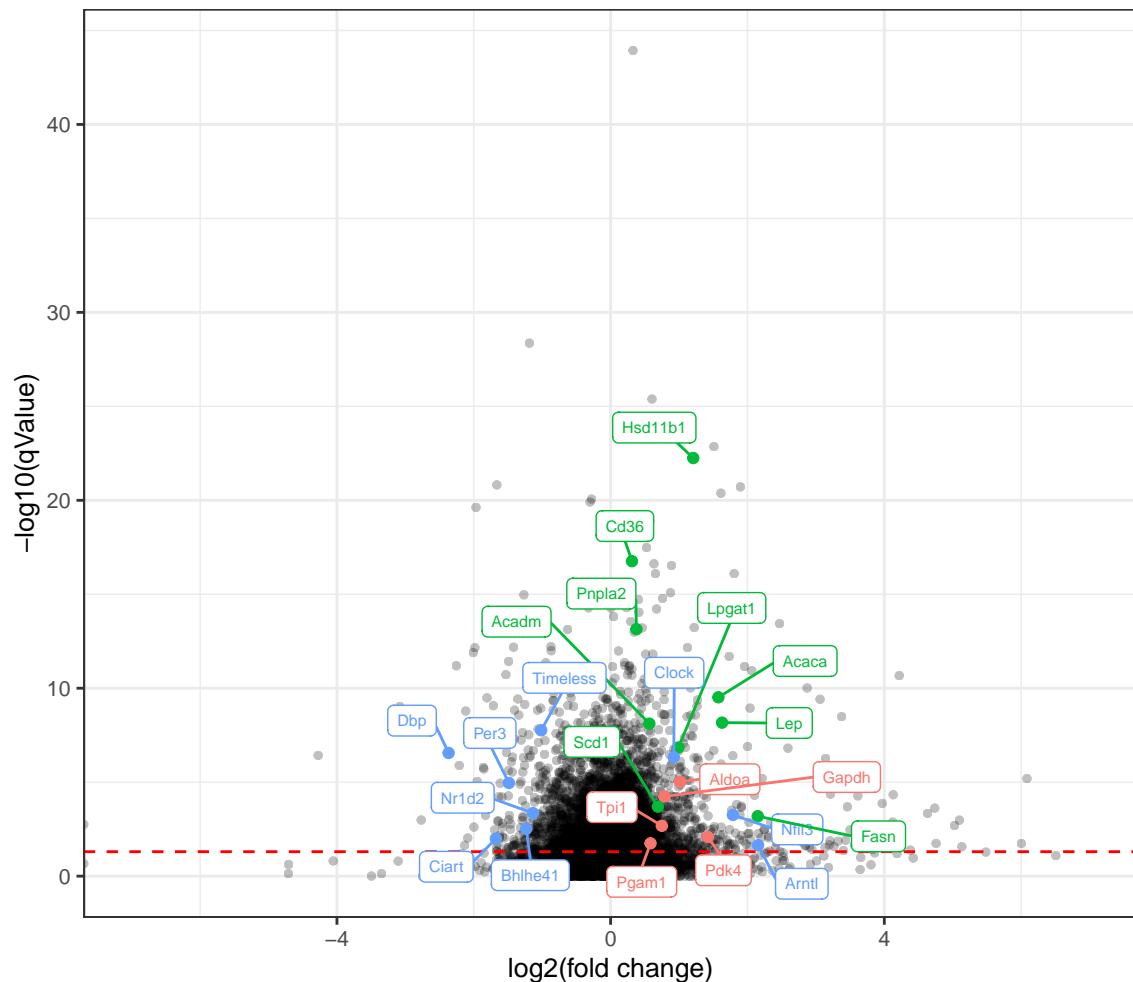
```

'Lipid metabolism',
ifelse(target_id %in% circ_rhy,
      'Circadian rhythm',
      NA)
))) %>%
mutate(Category = factor(Category,
                          levels = c('Glycolytic enzymes and regulators',
                                    'Lipid metabolism',
                                    'Circadian rhythm')))

joined_bat_gene_results %>% filter(!is.na(Direction)) %>%
ggplot(., 
       aes(x = log2(fc),
           y = -log10(lrt_qval))) +
geom_point(alpha = 0.25,
           stroke = 0,
           data = filter(joined_bat_gene_results,
                         !is.na(Direction),
                         is.na(Category))) +
geom_point(alpha = 1,
           size = 2,
           stroke = 0,
           data = filter(joined_bat_gene_results,
                         lrt_qval < 0.05,
                         !is.na(Category)),
           aes(color = Category)) +
geom_hline(yintercept = -log10(0.05),
            color = 'red',
            linetype = 'dashed') +
geom_label_repel(data = filter(joined_bat_gene_results,
                               !is.na(Category),
                               lrt_qval < 0.05),
                 aes(label = target_id,
                     color = Category),
                 size = 2,
                 force = 20) +
xlim(-7,7) +
#scale_color_manual(values = c('#4393c3', '#878787', '#d6604d')) +
xlab('log2(fold change)') +
ylab('-log10(qValue)') +
ggtitle('BAT Cort Pellet vs Placebo Pellet - Gene level') +
theme_bw() +
theme(text = element_text(size=10),
      legend.position="bottom")

```

BAT Cort Pellet vs Placebo Pellet – Gene level



Category a Glycolytic enzymes and regulators a Lipid metabolism a Circadian rhythm

Volcano plot of differentially expressed transcripts between cort vs placebo pellet BAT. Genes from Stefan's figure 6D are highlighted here.

In all of these plots, the genes that Stefan highlighted in figure 6D are shown on the plot. We can include other options, or include this later and color genes based on interesting GO terms. Lots of options here. The genes that were highlighted in the BAT analysis in the manuscript are no longer significant except for one, but I don't think this should be much of an issue since these genes are not expanded upon in the manuscript as far as I can tell. There are 4 volcano plots here two for WAT and two for BAT, with transcript level differential expression analysis and combined gene level analysis

Bar plots of select genes (Figure 6D)

WAT transcript level

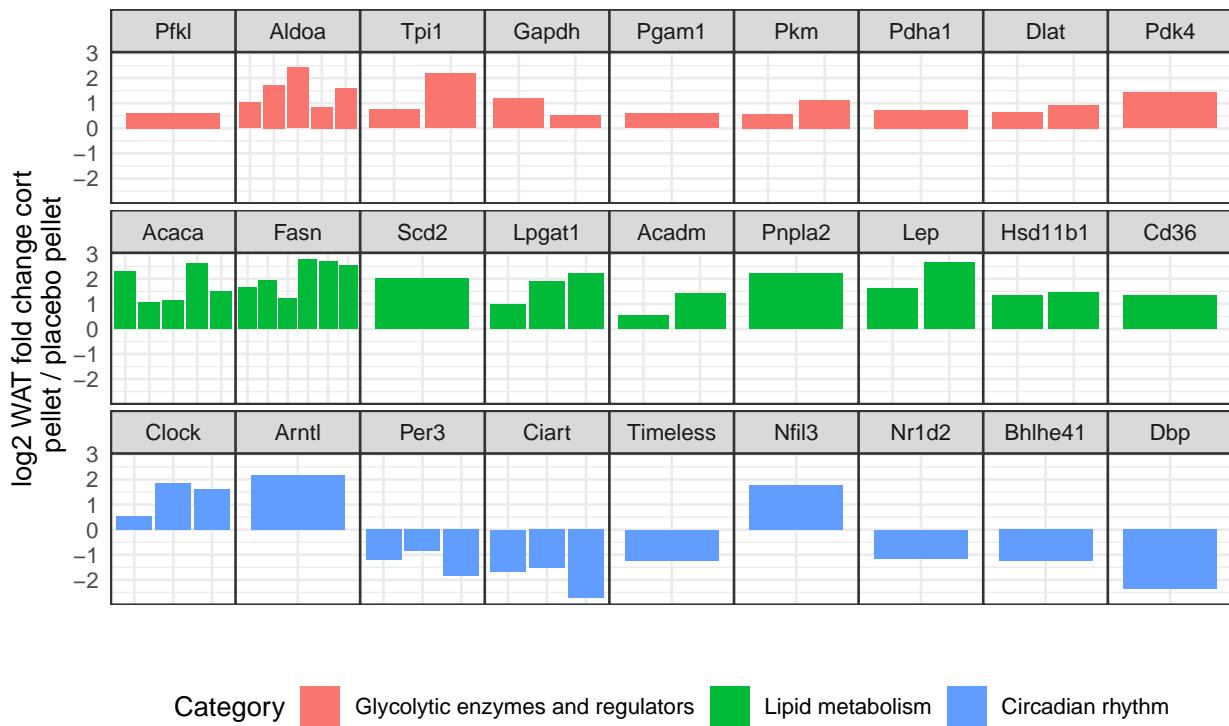
```
#####
# WAT Transcripts #
#####
```

```

joined_wat_results %>% filter(!is.na(Category), lrt_qval < 0.05) %>%
  mutate(ext_gene = factor(ext_gene, levels = c(gly_enz,lipid_met,circ_rhy)),
    Category = factor(Category,
      levels = c('Glycolytic enzymes and regulators',
        'Lipid metabolism',
        'Circadian rhythm'))) %>%
  ggplot(., aes(x = external_transcript_name, y = log2(fc), fill = Category)) +
  geom_bar(stat = 'identity') +
  facet_wrap(~ext_gene,
    scales = 'free_x',
    nrow = 3) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
    axis.ticks = element_blank(),
    panel.spacing = unit(0, "lines"),
    legend.position="bottom") +
  ggtitle('Selected gene abundances - transcript level') +
  xlab('') + ylab('log2 WAT fold change cort\n pellet / placebo pellet')

```

Selected gene abundances – transcript level



Category Glycolytic enzymes and regulators Lipid metabolism Circadian rhythm

WAT gene level

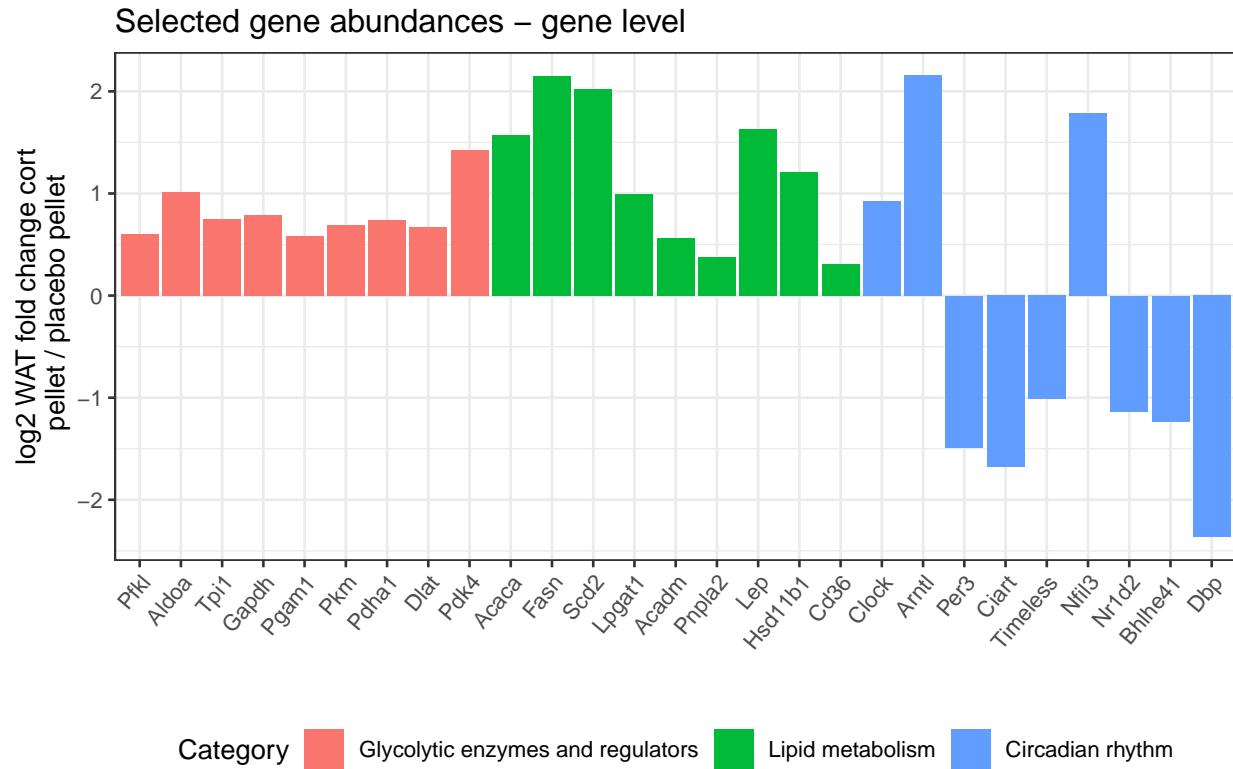
```

#####
# WAT Genes #
#####

joined_wat_gene_results %>% filter(!is.na(Category), lrt_qval < 0.05) %>%
  mutate(target_id = factor(target_id, levels = c(gly_enz,lipid_met,circ_rhy)),
    Category = factor(Category,
      levels = c('Glycolytic enzymes and regulators',
        'Lipid metabolism',
        'Circadian rhythm'))) %>%
  ggplot(., aes(x = target_id, y = log2(fc), fill = Category)) +
  geom_bar(stat = 'identity') +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 50, hjust = 1),
    panel.spacing = unit(0, "lines"),
    legend.position="bottom") +
  ggtitle('WAT gene level')

```

```
ggtitle('Selected gene abundances - gene level') +
  xlab('') + ylab('log2 WAT fold change cort\npellet / placebo pellet')
```

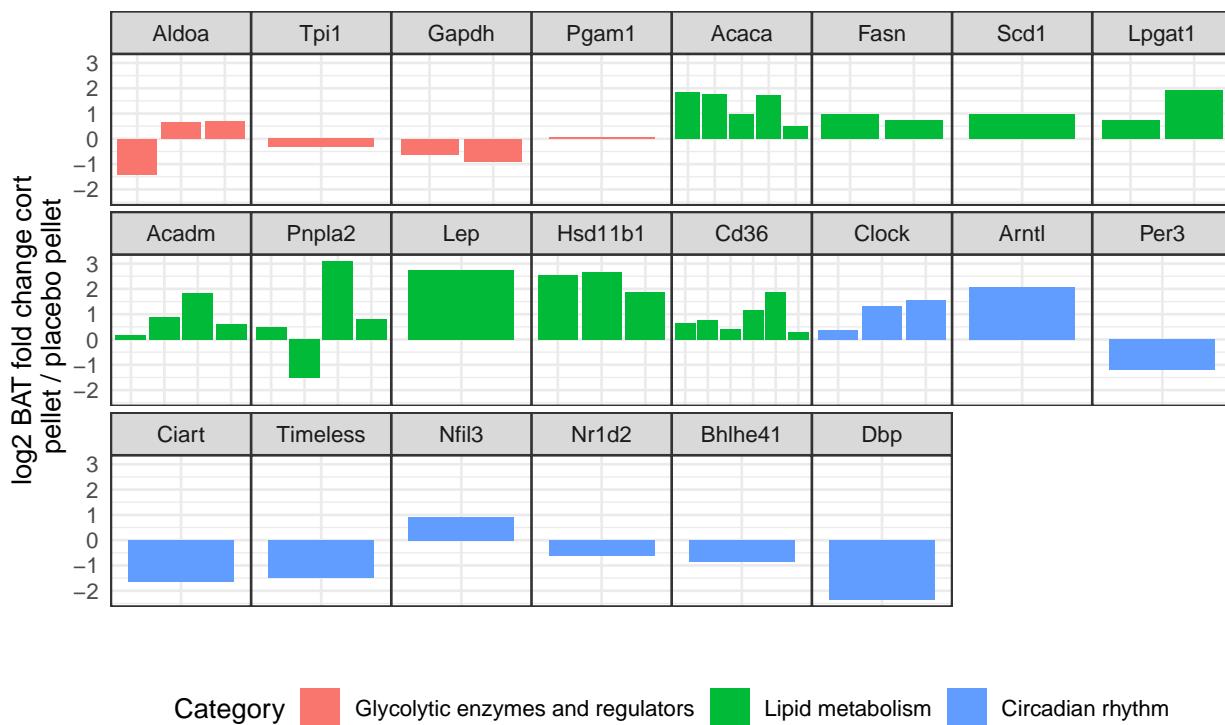


BAT transcript level

```
#####
# BAT Transcript #
#####

joined_bat_results %>% filter(!is.na(Category), lrt_qval < 0.05) %>%
  mutate(ext_gene = factor(ext_gene, levels = c(gly_enz, lipid_met, circ_rhy)),
        Category = factor(Category,
                           levels = c('Glycolytic enzymes and regulators',
                                     'Lipid metabolism',
                                     'Circadian rhythm'))) %>%
  ggplot(., aes(x = external_transcript_name, y = log2(fc), fill = Category)) +
  geom_bar(stat = 'identity') +
  facet_wrap(~ext_gene,
             scales = 'free_x',
             nrow = 3) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        panel.spacing = unit(0, "lines"),
        legend.position="bottom") +
  ggtitle('Selected gene abundances - transcript level') +
  xlab('') + ylab('log2 BAT fold change cort\npellet / placebo pellet')
```

Selected gene abundances – transcript level

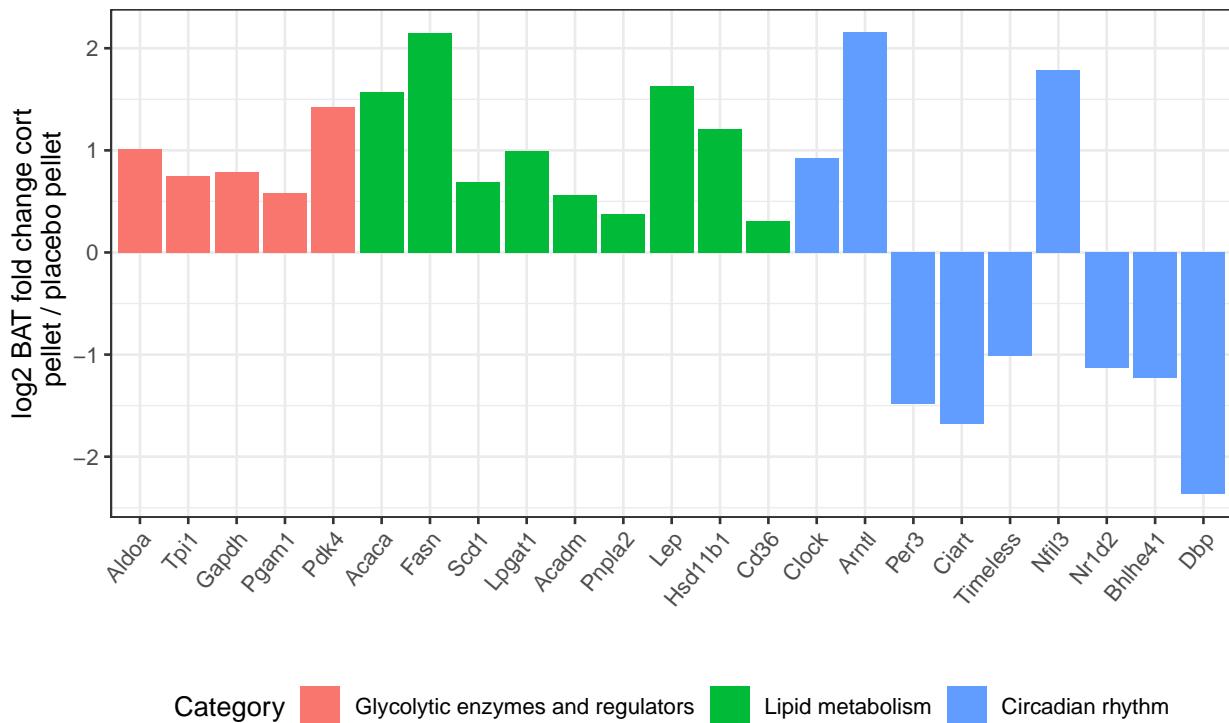


BAT gene level

```
#####
# BAT Gene #
#####

joined_bat_gene_results %>% filter(!is.na(Category), lrt_val < 0.05) %>%
  mutate(target_id = factor(target_id, levels = c(gly_enz,lipid_met,circ_rhy)),
         Category = factor(Category,
                           levels = c('Glycolytic enzymes and regulators',
                                     'Lipid metabolism',
                                     'Circadian rhythm')))) %>%
  ggplot(., aes(x = target_id, y = log2(fc), fill = Category)) +
  geom_bar(stat = 'identity') +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 50, hjust = 1),
        panel.spacing = unit(0, "lines"),
        legend.position="bottom") +
  ggtitle('Selected gene abundances - gene level') +
  xlab('') + ylab('log2 BAT fold change cort\npellet / placebo pellet')
```

Selected gene abundances – gene level



Category Glycolytic enzymes and regulators Lipid metabolism Circadian rhythm

Here the genes that Stefan highlighted in his WAT plot are shown. The WAT genes remain the same, and I've shown both transcript abundances as well as gene level expression. As stated in the last section, the genes that were highlighted in the BAT analysis in the manuscript are no longer significant except for one, so I just included the genes from the WAT figure. We can also pull out interesting GO terms that are talked about later, potentially mitochondrial genes for the BAT analysis.

GO Term analysis

WAT GO Term analysis

```
#####
# WAT GO Term analysis #
#####

gene.df <- clusterProfiler::bitr(dplyr::filter(joined_wat_results, lrt_qval < 0.05)$ext_gene,
                                    fromType = 'SYMBOL',
                                    toType = 'ENTREZID',
                                    OrgDb = org.Mm.eg.db)

universe <- clusterProfiler::bitr(joined_wat_results$ext_gene,
                                    fromType = 'SYMBOL',
                                    toType = 'ENTREZID',
                                    OrgDb = org.Mm.eg.db)

goDF <- clusterProfiler::enrichGO(gene = gene.df$ENTREZID,
                                    universe = universe$ENTREZID,
                                    ont = 'ALL',
                                    OrgDb = org.Mm.eg.db)

subsetFC <- joined_wat_results %>% dplyr::rename(GeneName = ext_gene, FC = fc) %>%
  mutate(FC = log2(FC))

# zScore function
zScore <- function(x, subsetFC, universe){
```

```

genes <- x %>% strsplit('/')[-1] %>% unlist()
convert <- dplyr::filter(universe, ENTREZID %in% genes)$SYMBOL
convert <- convert[!duplicated(convert)]

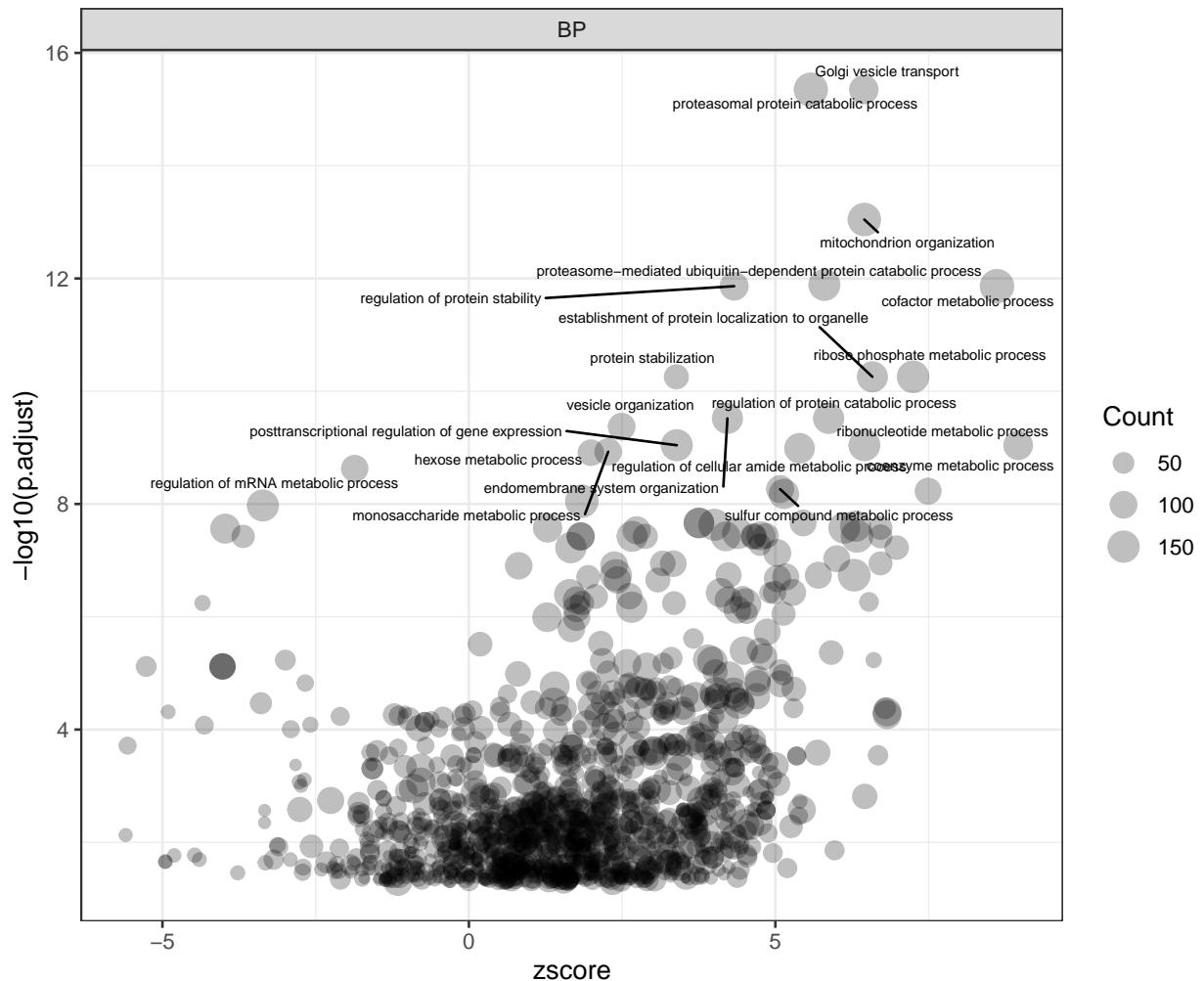
FC <- dplyr::filter(subsetFC, GeneName %in% convert)$FC
return((sum(FC > 0, na.rm = T) - sum(FC < 0, na.rm = T)) / sqrt(length(FC)))
}

goDF <- goDF %>% as_tibble() %>% group_by(ONTOLOGY) %>% mutate(rank = rank(p.adjust)) %>%
  rowwise() %>% dplyr::mutate(zscore = zScore(geneID, subsetFC, universe))

ggplot(filter(goDF, ONTOLOGY == 'BP'),
       aes(x = zscore,
           y = -log10(p.adjust),
           size = Count)) +
  geom_point(alpha= 0.25) +
  facet_wrap(~ONTOLOGY,
             scales = 'free',
             nrow = 1) +
  geom_text_repel(data = filter(goDF, rank <= 20,
                                ONTOLOGY == 'BP'),
                  aes(label = Description),
                  colour = 'black',
                  size = 2,
                  force = 10) +
  theme_bw() + ggtitle('GO Term Analysis: WAT Cort Pellet vs Placebo Pellet')

```

GO Term Analysis: WAT Cort Pellet vs Placebo Pellet



BAT GO Term analysis

```
#####
# BAT GO Term analysis #
#####

gene.df <- clusterProfiler::bitr(dplyr::filter(joined_bat_results, lrt_qval < 0.05)$ext_gene,
                                    fromType = 'SYMBOL',
                                    toType = 'ENTREZID',
                                    OrgDb = org.Mm.eg.db)

universe <- clusterProfiler::bitr(joined_bat_results$ext_gene,
                                    fromType = 'SYMBOL',
                                    toType = 'ENTREZID',
                                    OrgDb = org.Mm.eg.db)

goDF <- clusterProfiler::enrichGO(gene = gene.df$ENTREZID,
                                    universe = universe$ENTREZID,
                                    ont = 'ALL',
                                    OrgDb = org.Mm.eg.db)

subsetFC <- joined_bat_results %>% dplyr::rename(GeneName = ext_gene, FC = fc) %>%
  mutate(FC = log2(FC))

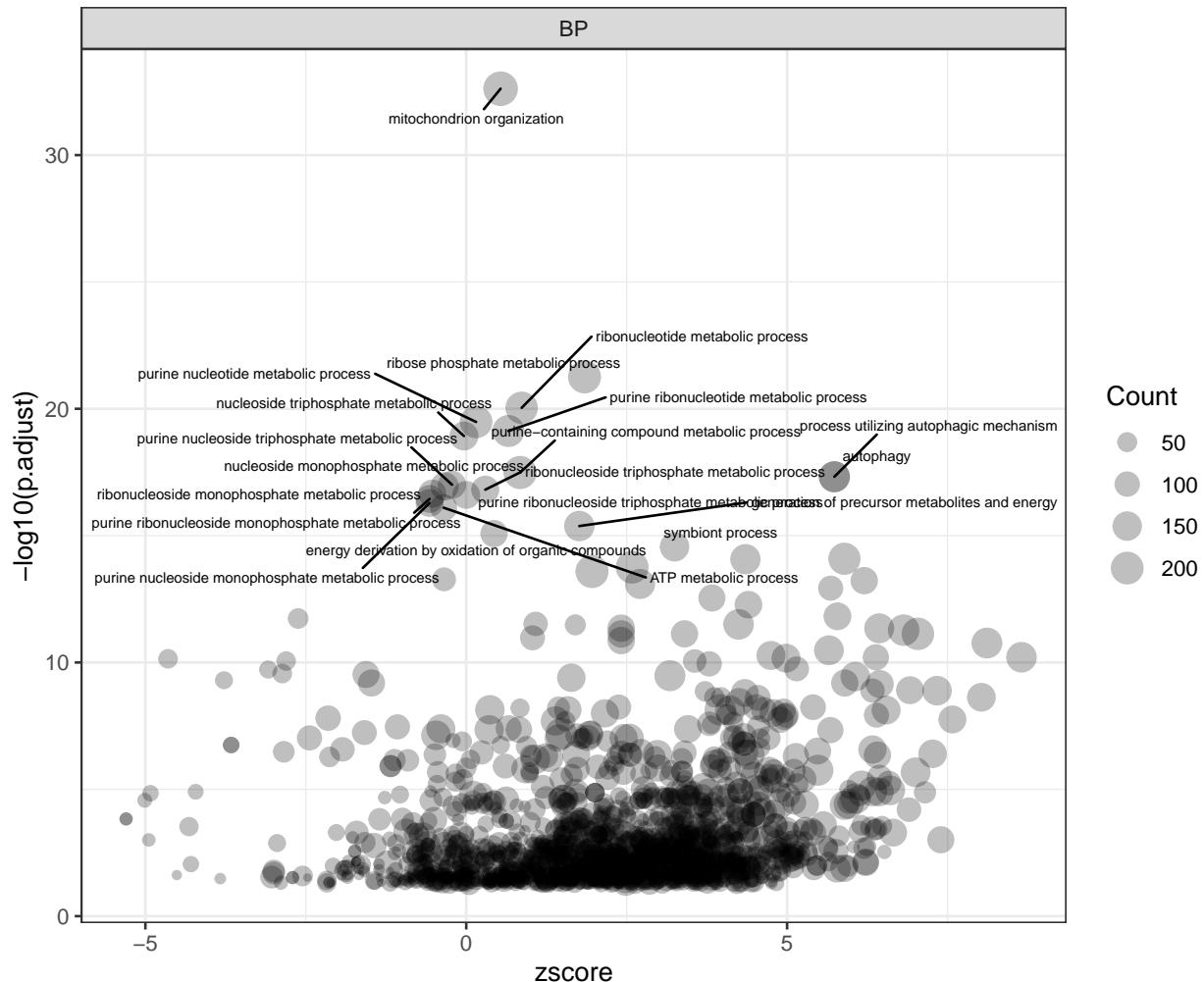
# zScore function
zScore <- function(x, subsetFC, universe){
  genes <- x %>% strsplit('/') %>% unlist()
  convert <- dplyr::filter(universe, ENTREZID %in% genes)$SYMBOL
  convert <- convert[!duplicated(convert)]

  FC <- dplyr::filter(subsetFC, GeneName %in% convert)$FC
  return((sum(FC > 0, na.rm = T) - sum(FC < 0, na.rm = T)) / sqrt(length(FC)))
}

goDF <- goDF %>% as_tibble() %>% group_by(ONTOLOGY) %>% mutate(rank = rank(p.adjust)) %>%
  rowwise() %>% dplyr::mutate(zscore = zScore(geneID, subsetFC, universe))

ggplot(filter(goDF, ONTOLOGY == 'BP'),
       aes(x = zscore,
           y = -log10(p.adjust),
           size = Count)) +
  geom_point(alpha= 0.25) +
  facet_wrap(~ONTOLOGY,
             scales = 'free',
             nrow = 1) +
  geom_text_repel(data = filter(goDF, rank <= 20,
                                ONTOLOGY == 'BP'),
                  aes(label = Description),
                  colour = 'black',
                  size = 2,
                  force = 10) +
  theme_bw() + ggtitle('GO Term Analysis: BAT Cort Pellet vs Placebo Pellet')
```

GO Term Analysis: BAT Cort Pellet vs Placebo Pellet



Not much for me to say here, we can take a closer look at this and decide how best to represent it. I can also provide a table in case you guys want to comb through these to see if you find something that you'd like to highlight in the manuscript. I think the potentially more interesting GO Term analysis comes when comparing BAT and WAT.

Overlap between WAT and BAT

Overlap between the up regulated genes between WAT and BAT

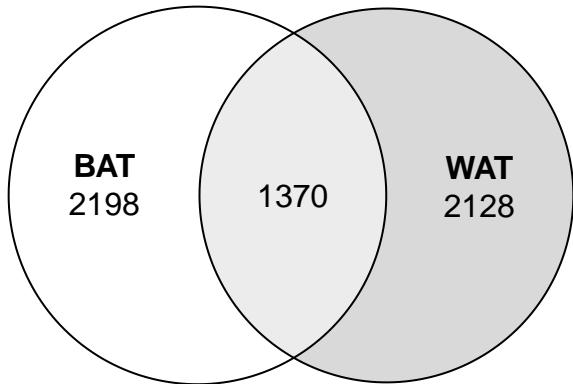
```
#####
# Up regulated transcripts #
#####

all_combined <- full_join(joined_bat_results, joined_wat_results, by = 'target_id')

up = cbind(all_combined$lrt_qval.x < 0.05 & all_combined$fc.x > 1,
           all_combined$lrt_qval.y < 0.05 & all_combined$fc.y > 1)
colnames(up) = c('BAT', 'WAT')
up[is.na(up)] = FALSE

plot(euler(up, shape = "ellipse"), quantities = TRUE, main = 'Up regulated transcripts')
```

Up regulated transcripts



Overlap between the down regulated genes between WAT and BAT

```
#####
# Down regulated transcripts #
#####

down = cbind(all_combined$lrt_qval.x < 0.05 & all_combined$fc.x < 1,
             all_combined$lrt_qval.y < 0.05 & all_combined$fc.y < 1)
colnames(down) = c('BAT','WAT')
down[is.na(down)] = FALSE

plot(euler(down, shape = "ellipse"), quantities = TRUE, main = 'Down regulated transcripts')
```

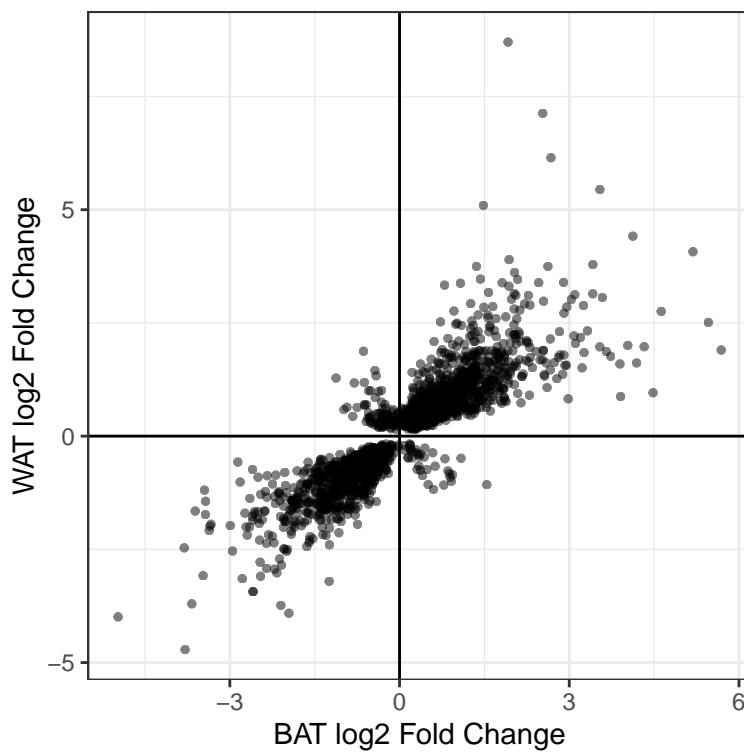
Down regulated transcripts



Pairwise plots between differentially expressed WAT and BAT tissues under Cort pellet treatment

```
#####
# Pairwise Plots #
#####

all_combined %>% mutate(comb_hit = lrt_qval.x < 0.05 & lrt_qval.y < 0.05) %>%
  filter(comb_hit) %>%
  ggplot(., aes(x= log2(fc.x),
    y = log2(fc.y))) +
  geom_point(alpha= 0.5, stroke = 0) + geom_hline(yintercept = 0) + geom_vline(xintercept = 0) +
  theme_bw() + xlab('BAT log2 Fold Change') + ylab('WAT log2 Fold Change')
```



This pairwise plot shows the significant agreement between the up and down regulated genes between WAT and BAT. I noticed an interesting group of negatively correlated genes, so I thought it would be interesting to do GO term analysis on the genes that are positively correlated vs the ones that negatively correlated.

```
#####
# GO Term enrichment for on diagonal overlap #
#####

on_diag <- all_combined %>% mutate(comb_hit = lrt_qval.x < 0.05 &
  lrt_qval.y < 0.05 &
  log2(fc.x)/log2(fc.y) > 0) %>%
  filter(comb_hit)

gene.df <- clusterProfiler::bitr(on_diag$ext_gene.x,
  fromType = 'SYMBOL',
  toType = 'ENTREZID',
  OrgDb = org.Mm.eg.db)

universe <- clusterProfiler::bitr(all_combined$ext_gene.x,
  fromType = 'SYMBOL',
  toType = 'ENTREZID',
  OrgDb = org.Mm.eg.db)

goDF <- clusterProfiler::enrichGO(gene = gene.df$ENTREZID,
  universe = universe$ENTREZID,
  ont = 'ALL',
```

```

OrgDb = org.Mm,eg.db

# zScore function for differential expression between two groups
zScoreDiag <- function(x, subsetFC, universe){
  genes <- x %>% strsplit('/')[-1] %>% unlist()
  convert <- dplyr::filter(universe, ENTREZID %in% genes)$SYMBOL
  convert <- convert[!duplicated(convert)]

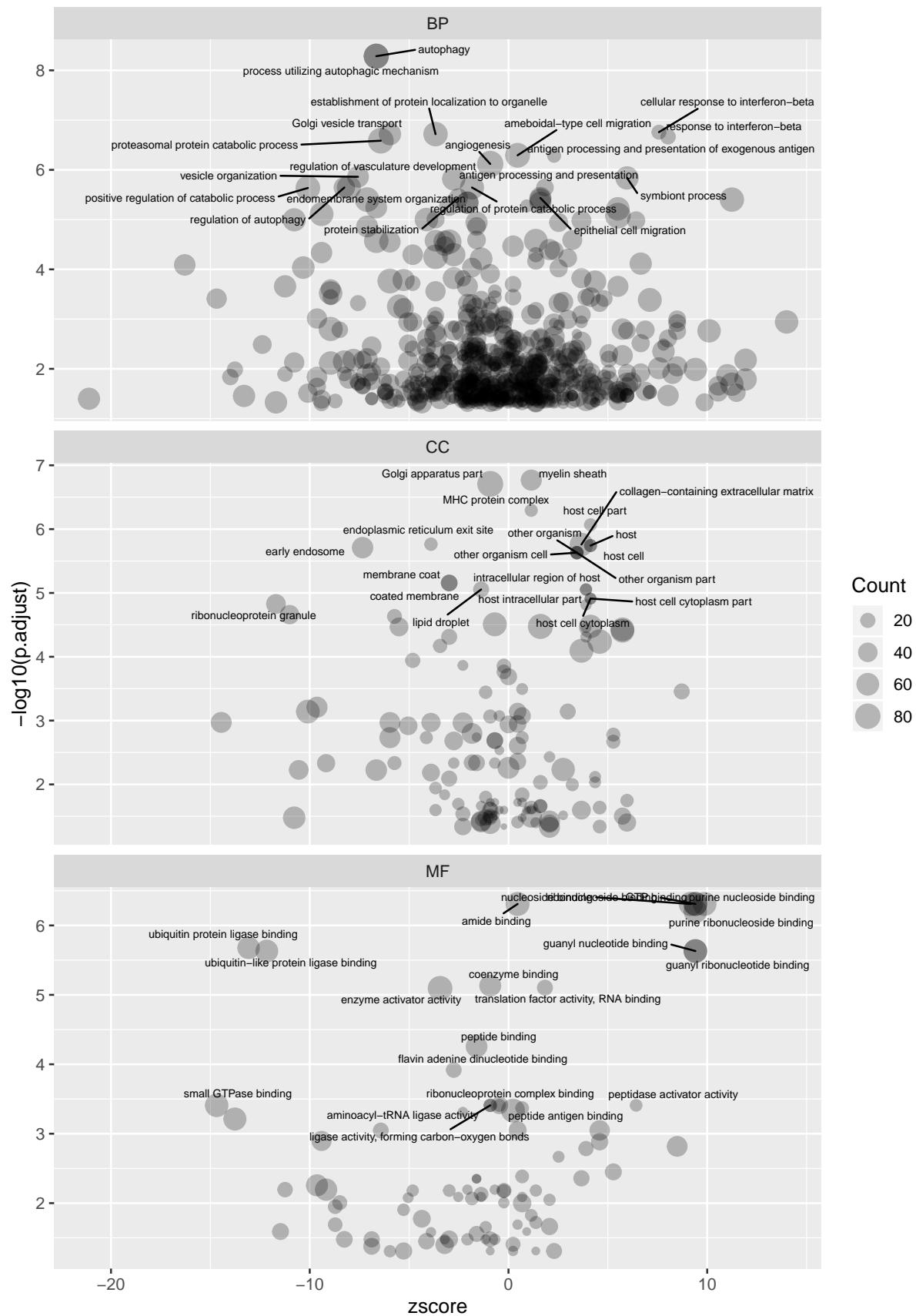
  subsetFC <- subsetFC %>% dplyr::filter(GeneName %in% convert)
  #return(subsetFC)
  return((sum(subsetFC$fc.x < subsetFC$fc.y, na.rm = T) - sum(subsetFC$fc.x > subsetFC$fc.y, na.rm = T)) / sqrt(ncol(subsetFC)))
}

subsetFC <- all_combined %>% dplyr::rename(GeneName = ext_gene.x)

goDF <- goDF %>% as_tibble() %>% group_by(ONTOLOGY) %>% mutate(rank = rank(p.adjust)) %>%
  rowwise() %>% dplyr::mutate(zscore = zScoreDiag(geneID, subsetFC, universe))

ggplot(goDF, aes(x = zscore, y = -log10(p.adjust))) + geom_point(aes(size = Count), alpha = 0.25) +
  facet_wrap(~ONTOLOGY, ncol = 1, scales = 'free_y') +
  geom_text_repel(data = filter(goDF, rank <= 20),
                  aes(label = Description),
                  colour = 'black',
                  size = 2,
                  force = 10)

```



This is a GO term analysis of the genes that are positively correlated between WAT and BAT. Here the zscore is a bit different than before and is meant to capture the differential expression between WAT and BAT instead of Cort vs Control. The ratio between Cort and Control values from each tissue are compared to each other, so a positive zscore would indicate higher relative expression in the WAT Cort/Placebo condition than the BAT Cort/Placebo condition.

```
#####
# GO Term enrichment for off diagonal overlap #
#####

off_diag <- all_combined %>% mutate(comb_hit = lrt_qval.x < 0.05 &
  lrt_qval.y < 0.05 &
  log2(fc.x)/log2(fc.y) < 0) %>%
  filter(comb_hit)

gene.df <- clusterProfiler::bitr(off_diag$ext_gene.x,
  fromType = 'SYMBOL',
  toType = 'ENTREZID',
  OrgDb = org.Mm.eg.db)

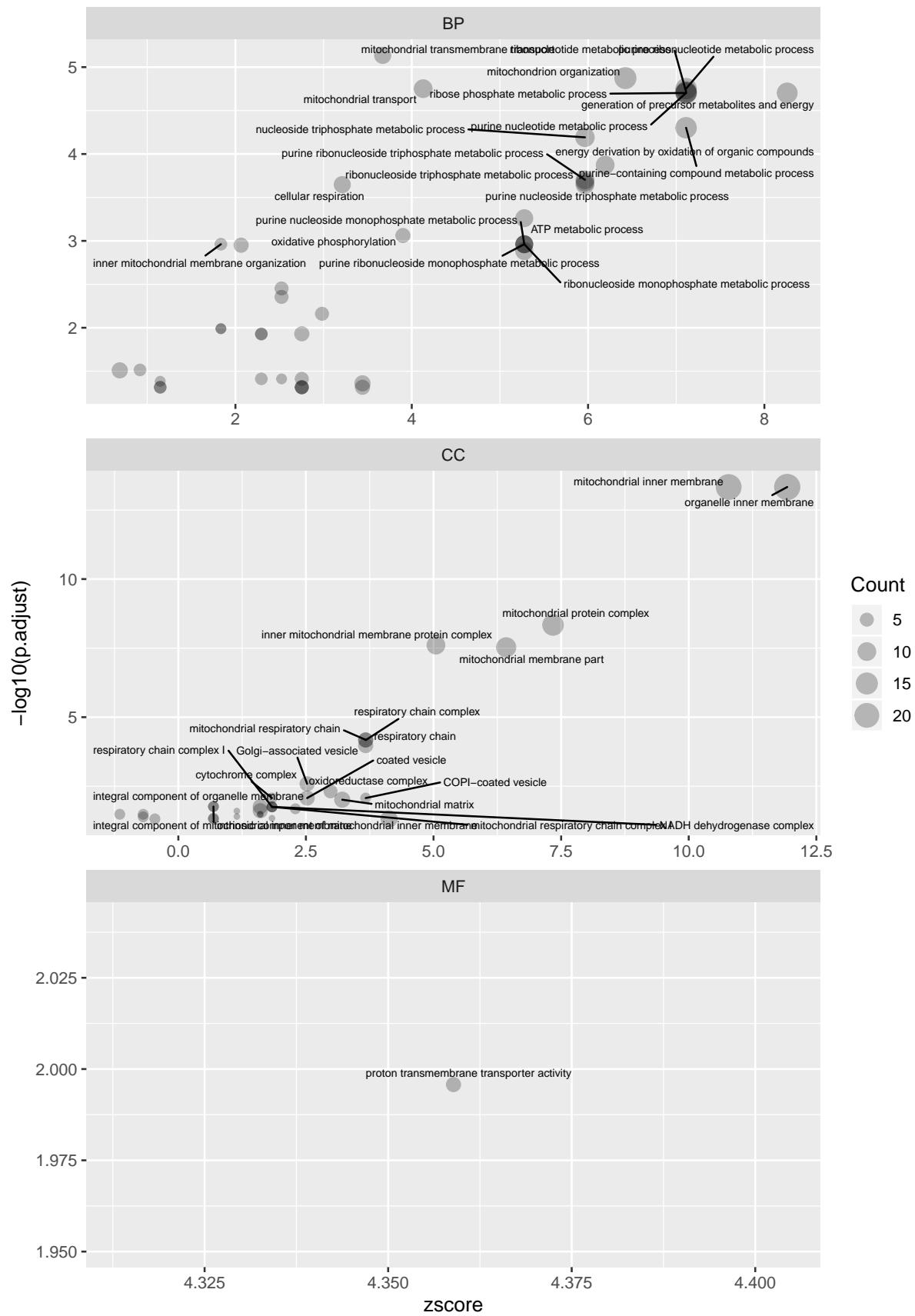
universe <- clusterProfiler::bitr(all_combined$ext_gene.x,
  fromType = 'SYMBOL',
  toType = 'ENTREZID',
  OrgDb = org.Mm.eg.db)

goDF <- clusterProfiler::enrichGO(gene = gene.df$ENTREZID,
  universe = universe$ENTREZID,
  ont = 'ALL',
  OrgDb = org.Mm.eg.db)

subsetFC <- all_combined %>% dplyr::rename(GeneName = ext_gene.x)

goDF <- goDF %>% as_tibble() %>% group_by(ONTOLOGY) %>% mutate(rank = rank(p.adjust)) %>%
  rowwise() %>% dplyr::mutate(zscore = zScoreDiag(geneID, subsetFC, universe))

ggplot(goDF, aes(x = zscore, y = -log10(p.adjust))) + geom_point(aes(size = Count), alpha = 0.25) +
  facet_wrap(~ONTOLOGY, ncol = 1, scales = 'free') +
  geom_text_repel(data = filter(goDF, rank <= 20),
    aes(label = Description),
    colour = 'black',
    size = 2,
    force = 10)
```



This is a GO term analysis of the genes that are negatively correlated between WAT and BAT. Here the zscore described above is used again. There is a clear trend that mitochondrial genes are being effected in WAT and BAT differently by cort treatment. It would appear that in WAT these genes are increased in abundance and in BAT the are decreased in abundance.

TF Motifs WAT up vs down regulated genes

```
#####
# WAT Analysis #
#####

# Export data for oPOSSUM analysis
joined_wat_results %>% filter(lrt_qval < 0.05, fc < 1) %>%
  filter(!duplicated(ens_gene)) %>%
  write_csv('~/Downloads/down_regulated_wat.csv')

joined_wat_results %>% filter(lrt_qval < 0.05, fc > 1) %>%
  filter(!duplicated(ens_gene)) %>%
  write_csv('~/Downloads/up_regulated_wat.csv')

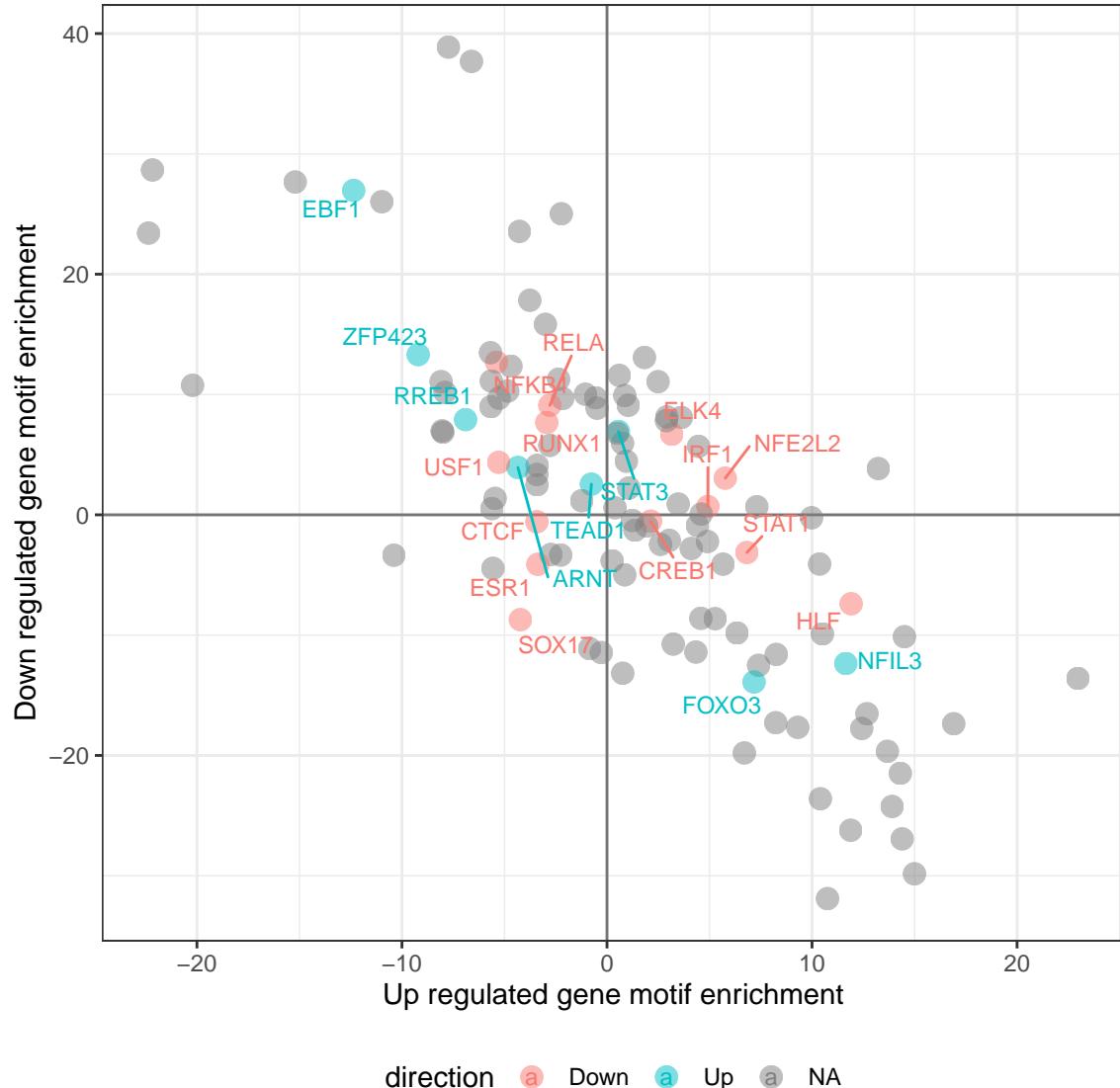
# Load in oPOSSUM results
tf_up <- read_xlsx '~/Documents/GitHub/ShinyApps/InVivoRNAseq/tf_motifs_up_regulated_wat.xlsx' %>% dplyr::select(TF, `Z-score`) %>%
  dplyr::rename(zscore_up = `Z-score`)

tf_down <- read_xlsx '~/Documents/GitHub/ShinyApps/InVivoRNAseq/tf_motifs_down_regulated_wat.xlsx' %>% dplyr::select(TF, `Z-score`) %>%
  dplyr::rename(zscore_down = `Z-score`)

tf_all <- full_join(tf_up, tf_down, by = 'TF') %>% mutate(TF = toupper(TF))

tf_all <- joined_wat_results %>% dplyr::select(ext_gene, fc, lrt_qval) %>%
  dplyr::rename(TF = ext_gene) %>%
  mutate(direction = ifelse(lrt_qval < 0.05 & fc > 1,
    'Up',
    ifelse(lrt_qval < 0.05 & fc < 1,
      'Down',
      NA)),
    TF = toupper(TF)) %>%
  filter(!is.na(direction)) %>%
  right_join(tf_all, 'TF') %>% filter(!duplicated(TF))

ggplot(tf_all, aes(x = zscore_up, y = zscore_down, color = direction)) +
  geom_vline(xintercept = 0, alpha = 0.5) + geom_hline(yintercept = 0, alpha = 0.5) +
  geom_point(size = 4, alpha = 0.5, stroke = 0) +
  geom_text_repel(data = filter(tf_all, !is.na(direction)),
    aes(label = TF, color = direction), size = 3, force = 10) +
  theme_bw() + theme(legend.position = 'bottom') +
  xlab('Up regulated gene motif enrichment') + ylab('Down regulated gene motif enrichment')
```



This is still a work in progress but I think could be interesting to show for the overall gene expression of BAT and WAT, and possibly for some interesting GO terms (like maybe the mitochondrial ones). This analysis takes a bit of time to explain but here goes... I used an analysis tool called oPOSSUM (<http://opossum.cisreg.ca/oPOSSUM3/>) that looks for over-representation of regulatory motifs around a given set of genes.

Here I submitted two lists of genes to oPOSSUM: the WAT genes that were increased in Cort treatment and the genes that were decreased in Cort treatment. oPOSSUM outputs a table where transcription factors that are over or under represented in the gene list are shown with a z-score that is calculated based on the abundance of these motifs in the submitted gene list vs the whole genome.

I then used the z-scores that were generated and created a pairwise plot between them, where each point is a transcription factor and high z-scores show an over representation for the motifs for that transcription factor and low z-scores show an under representation of the motifs for that transcription factor. Interestingly there is a strong negative correlation shown here, which can be interpreted as: 1) Motifs that are enriched in the up-regulated genes are depleted in the down regulated genes 2) Motifs that are enriched in the down-regulated genes are depleted in the up regulated genes 3) This makes sense if there was a change in the overall gene regulatory program in the cells.

I also colored the TFs red if they were down regulated themselves in the data set, blue if they were down regulated, and grey if they were either not significant or were not measured. It's nice to see blue (up-regulated) TFs in the lower right quadrant, suggesting that the up-regulation of FOXO3 and NFIL3 could explain some of the differential expression of genes after adding Cort.

One thing that puzzled me at first was the up regulation of two TFs in the upper left quadrant, since EBF1 and ZFP423 are typically associated with increased expression of their targets and this quadrant represents the motifs for the down regulated genes. I found though that EBF1 and ZFP423 together act as transcriptional repressors, and their increased abundance together could be repressing genes in WAT after Cort treatment.

```
#####
# BAT Analysis #
#####

## # Export data for oPOSSUM analysis
# joined_bat_results %>% filter(lrt_qval < 0.05, fc < 1) %>%
#   filter(!duplicated(ens_gene)) %>%
#   write_csv('~/Downloads/down_regulated_bat.csv')
#
# joined_bat_results %>% filter(lrt_qval < 0.05, fc > 1) %>%
#   filter(!duplicated(ens_gene)) %>%
#   write_csv('~/Downloads/up_regulated_bat.csv')
#
## # Load in oPOSSUM results
# tf_up <- read_xlsx '~/Documents/GitHub/ShinyApps/InVivoRNaseq/tf_motifs_up_regulated_bat.xlsx' %>%
#   dplyr::select(TF, `Z-score`) %>%
#   dplyr::rename(zscore_up = `Z-score`)
#
# tf_down <- read_xlsx '~/Documents/GitHub/ShinyApps/InVivoRNaseq/tf_motifs_down_regulated_bat.xlsx' %>%
#   dplyr::select(TF, `Z-score`) %>%
#   dplyr::rename(zscore_down = `Z-score`)
#
# tf_all <- full_join(tf_up, tf_down, by = 'TF') %>% mutate(TF = toupper(TF))
#
# tf_all <- joined_wat_results %>% dplyr::select(ext_gene, fc, lrt_qval) %>%
#   dplyr::rename(TF = ext_gene) %>%
#   mutate(direction = ifelse(lrt_qval < 0.05 & fc > 1,
#     'Up',
#     ifelse(lrt_qval < 0.05 & fc < 1,
#       'Down',
#       NA)),
#     TF = toupper(TF)) %>%
#   filter(!is.na(direction)) %>%
#   right_join(tf_all, 'TF') %>% filter(!duplicated(TF))
#
# ggplot(tf_all, aes(x = zscore_up, y = zscore_down, color = direction)) +
#   geom_vline(xintercept = 0, alpha = 0.5) + geom_hline(yintercept = 0, alpha = 0.5) +
#   geom_point(size = 4, alpha = 0.5, stroke = 0) +
#   geom_text_repel(data = filter(tf_all, zscore_up > 10, zscore_down < -5),
#     aes(label = TF, color = direction), size = 3, force = 10) +
#   geom_text_repel(data = filter(tf_all, zscore_down > 10, zscore_up < -5),
#     aes(label = TF, color = direction), size = 3, force = 10) +
#   theme_bw()
```

Conclusions and next steps

- I still need to do the oPOSSUM analysis for the BAT tissue, and would also like to use another program called Pscan that does the same thing to see how well they agree with one another.
- I think it would also be important to show (likely in the supplement) volcano plots of deferentially expressed genes (or lack thereof) between other conditions such as:
 - cort injection vs placebo injection
 - placebo pellet vs placebo injection
 - cort injection vs placebo pellet
- This can either be done separately or all together, using the likelihood ratio test model on all of those data sets at once. From my previous analysis I think the number of hits should be few to none.

- Let me know what other plots or analysis you think we need and which ones you would like to include for sure and I will finalize those plots and put them into the figures.