Analysis notebook for A Quantitative Systems Pharmacology (QSP) Approach to Repurposing Drugs for Non-Alcoholic Fatty Liver Disease (NAFLD)

Introduction

This code is the analysis portion of the paper. Intermediate data files are stored throughout the notebook as commented out `RDS` files. These are typically saved when creating the intermediate data is both/or time consuming or used at branch points within the analysis. Some of the code chunks rely on functions from my personal packages which are not on `CRAN` or `BioConductor` installed from github.

Installing my packages

```
require(devtools)

# My package(s)
my_pkgs <- c("lefutils", "basicOmics")

if(any(my_pkgs %in% row.names(installed.packages()))){
 devtools::install_github(paste0("lefeverde/", my_pkgs))
}
```

Pre-processing pipeline

Creating gene metadata

```
require(tidyverse)
require(biomaRt)
gene_dir <- 'data-raw/DiStefano_raw_data/kallisto/'

# Getting ENS ids from 1st tsv
ens_trans <-
  paste0(gene_dir,list.files(gene_dir)[[1]]) %>%
  read_delim(., "\t") %>%
  pull(target_id) %>%
  str_split(., "\\.") %>%
  map(1) %>%
  unlist


attributes <- c("ensembl_gene_id", "ensembl_transcript_id")
```

```
ensembl = useEnsembl(biomart="ensembl", dataset="hsapiens_gene_ensembl", version=94)

new_tx2gene <-  getBM(attributes, "ensembl_transcript_id", ens_trans, ensembl)


attributes <- c("ensembl_gene_id",
                "external_gene_name",
                "gene_biotype",
                "description")

ens_genes <- unique(new_tx2gene$ensembl_gene_id)
new_gene_annots <- getBM(attributes, "ensembl_gene_id", ens_genes, ensembl)
```

Identifying mis-labeled samples

```
library(matrixStats)
library(edgeR)
library(reshape2)
library(ggplot2)

sex_linked_genes <- c('ENSG00000131002', 'ENSG00000183878', 'ENSG00000067048')

txi_n192 <- readRDS('DiStefano_txi_and_p_192.rds')
p192 <- txi_n192[[1]]
txi_n192 <- txi_n192[[2]]
cnts <- txi_n192$counts
cnts_per_million <- cnts[rowMedians(cnts) > 1,] %>%
  data.frame(.) %>%
  cpm(., log = FALSE)

plot_data <-
  data.frame(cnts_per_million[sex_linked_genes[1],],
             sex=p192$sex,
             sample_id=p192$dcl_patient_id) %>%
  melt(.) %>%
  dplyr::select( -one_of('variable')) %>%
  arrange(., value)
```

Sex is incorrect for some samples

Since this gene is Y-linked, I expect the males to have much higher expression than the females. I think the non-zero expression in the females can be attributed to multi-mapping reads. Kallisto does mapping a little differently, and so these are not discarded by default like with STAR and HISAT2.

```
female_cutoff <-  plot_data %>%
  dplyr::filter(., sex =='Female') %>%
  arrange(., value) %>%
  pull() %>%
  quantile(., probs = .95)

male_cutoff <-  plot_data %>%
```

```
  dplyr::filter(., sex =='Male') %>%
  arrange(., value) %>%
  pull() %>%
  quantile(., probs = .05)

suspicious_samples <- list(
  dplyr::filter(plot_data[plot_data$sex == 'Male',], value < male_cutoff),
  dplyr::filter(plot_data[plot_data$sex == 'Female',], value > female_cutoff)
) %>%
  lapply(., function(x){
    x$sample_id
  }) %>%
  do.call(c, .)
```

Creating new subsetted txi

```
suspicious_samples <-
  c("DLDR_0189","DLDR_0031","DLDR_0022","DLDR_0017","DLDR_0131", "DLDR_0130","DLDR_0140","DLDR_0122","DL

load('data-raw/tx2gene.RData')
p192 <- readRDS('data-raw/DiStefano_raw_data/pdata.rds')
p <- p192 %>%
  dplyr::filter(., !dcl_patient_id %in% suspicious_samples)
row.names(p) <- p$dcl_patient_id

files <-
  paste0(p$file_name, '.gz') %>%
  paste0('data-raw/DiStefano_raw_data/kallisto/', .)
names(files) <- row.names(p)
txi <- tximport(files, type = 'kallisto', tx2gene = tx2gene, ignoreTxVersion = T, countsFromAbundance =
#saveRDS(list(p, txi), file='DiStefano_txi_and_p_182.rds')
```

DEG analysis

Required packages and setup

```
require(lefutils)
require(basicOmics)
require(edgeR)
require(limma)
require(sva)
```

LIMMA-VOOM w/quantile normalization and SVA

Disease group

```
load('data-raw/gene_annots.RData')
txi_list <- readRDS('DiStefano_txi_and_p_182.rds')
```

```
p <- txi_list[[1]]
txi <- txi_list[[2]]

group_cont <-  c('STEATOSIS-NORMAL', 'Lob-NORMAL', 'Lob-STEATOSIS', 'Fibrosis-NORMAL', 'Fibrosis-STEATOS

mod <- basicOmics::better_model_matrix(~0 + group, data=p)
mod0 <- model.matrix(~1, data=p )
dge <- edgeR::DGEList(counts =  txi$counts,
                      samples = p,
                      genes = gene_annots[row.names(txi$counts),])

dge <- dge[edgeR::filterByExpr(dge, group = dge$samples$diagnosis),]
dge <- dge[(rowSums(dge$counts > 1) >= .75*dim(dge)[2]),]
dge <- dge[!is.na(dge$genes$entrezgene),]

v <- voom(dge, mod, normalize.method = 'quantile')
svobj <- sva(v$E, mod=mod, mod0=mod0)
# saveRDS(list(v, svobj), 'DiStefano_group_n182_voom_qn_and_svobj.rds')
modSV <- cbind(mod, svobj$sv)
fit <- make_cont_fit(v, group_cont, modSV)
# saveRDS(fit, 'DiStefano_group_n182_qn_sva_fit.rds')

res <- get_limma_results(fit, group_cont)
# saveRDS(res, file = 'DiStefano_group_n182_qn_sva_res.rds')
res$coefficient <- gsub('-', '_vs_', res$coefficient)

sigsub <- res[res$adj.P.Val < .05,]
```

Disease stage

```
library(sva)
load('data-raw/gene_annots.RData')
txi_list <- readRDS('DiStefano_txi_and_p_182.rds')
p <- txi_list[[1]]
txi <- txi_list[[2]]
diagnosis_levels <- levels(p$diagnosis)
# Comparing diagnosis progression
group_cont <-
  paste0(diagnosis_levels[-1],
         '-',
         diagnosis_levels[1:(length(diagnosis_levels)- 1)])
# Comparing everything to normal
group_cont <-
  c(group_cont,
    paste0(diagnosis_levels[-1], '-', 'NORMAL'))
# Comparing fib to lob 1
group_cont <-
  c(group_cont,
    paste0(diagnosis_levels[6:8], '-', "Lob_Inflam_1"))
# Comparing to fib to steatosis
group_cont <-
  c(group_cont,
```

```
    paste0(diagnosis_levels[6:8], '-', "STEATOSIS_2"),
    paste0(diagnosis_levels[6:8], '-', "STEATOSIS_3"))


mod <- basicOmics::better_model_matrix(~0 + diagnosis, data=p)
mod0 <- model.matrix(~1, data=p )
dge <- edgeR::DGEList(counts =  txi$counts,
                      samples = p,
                      genes = gene_annots[row.names(txi$counts),])

dge <- dge[edgeR::filterByExpr(dge, group = dge$samples$diagnosis),]
dge <- dge[(rowSums(dge$counts > 1) >= .75*dim(dge)[2]),]
dge <- dge[!is.na(dge$genes$entrezgene),]

v <- voom(dge, mod, normalize.method = 'quantile')
svobj <- sva(v$E, mod=mod, mod0=mod0)
# saveRDS(list(v, svobj), 'DiStefano_diagnosis_n182_voom_qn_and_svobj.rds')
modSV <- cbind(mod, svobj$sv)
fit <- make_cont_fit(v, group_cont, modSV)
# saveRDS(fit, 'DiStefano_diagnosis_n182_qn_sva_fit.rds')
res <- get_limma_results(fit, unique(group_cont))
# saveRDS(res, file = 'DiStefano_diagnosis_n182_qn_sva_res.rds')
sigsub <- res[res$adj.P.Val < .05,]
```

GSEA pathway analysis

Required packages and setup

```
require(lefutils)
require(basicOmics)
require(clusterProfiler)
```

GSEA with genes ranked by moderated t-statistic

Disease group

```
n_perm <- 10000

gsea_list <- kegg_gsea_by_coef(read_rds('DiStefano_group_n182_qn_sva_res.rds'), nPerm = n_perm)
gsea_df <- lapply(gsea_list, data.frame) %>%
  rbind_named_df_list(., 'coefficient') %>%
  filter(p.adjust < .05)

fn <-  paste0('group_kegg_gsea_nperm', n_perm, '_res.rds')
# saveRDS(gsea_df, fn)
```

Disease stage

```
n_perm <- 10000

gsea_list <- kegg_gsea_by_coef(read_rds('DiStefano_diagnosis_n182_qn_sva_res.rds'), nPerm = n_perm)
gsea_df <- lapply(gsea_list, data.frame) %>%
  rbind_named_df_list(., 'coefficient') %>%
  filter(p.adjust < .05)

fn <-  paste0('diagnosis_kegg_gsea_nperm', n_perm, '_res.rds')
saveRDS(gsea_df, fn)
```

GSVA pathway cluster analysis

Loading KEGG metadata

```
## KEGG Hierchy
pth_data <-
  read_rds('../reference_files/msigdb_v7.0/msigdb_v7.0_all_tibble.rds') %>%
  filter(SUB_CATEGORY_CODE == "CP:KEGG") %>%
  dplyr::select(STANDARD_NAME,EXACT_SOURCE) %>%
  setNames(., c('name','id'))

hier_kegg <-
  read_rds('data-raw/kegg_pathway_hierarchy.rds') %>%
  left_join(pth_data, .) %>%
  arrange(category_level_1)

cp_kegg <-
  '../reference_files/msigdb_v7.0/c2_kegg_entrez_msigdb_v7.0.gmt' %>%
  read.gmt(.) %>%
  filter(ont %in% hier_kegg$name) %>%
  split(., .$ont) %>%
  lapply(., function(x){
    x$gene
  })


kegg_annots <-
  hier_kegg$category_level_1 %>% set_names(., hier_kegg$name)
```

Loading and pre-processing gene expression data

```
load('data-raw/gene_annots.RData')

v <-  read_rds('DiStefano_diagnosis_n182_voom_qn_and_svobj.rds')[[1]]
svobj <- read_rds('DiStefano_diagnosis_n182_voom_qn_and_svobj.rds')[[2]]
p <-  v$targets
```

```
e_rm <-
  removeBatchEffect(v, covariates = svobj$sv, design = v$design)
e_rm <-
  uniquefy_by_variance(e_rm, gene_annots, 'entrezgene')
row.names(e_rm) <- gene_annots[row.names(e_rm),]$entrezgene



v_e <- uniquefy_by_variance(v$E, gene_annots, 'entrezgene')
row.names(v_e) <- gene_annots[row.names(v_e),]$entrezgene
```

Clustering the GSVA data

```
gsva_res_sva <-
  gsva(e_rm, cp_kegg, method='gsva',  min.sz=1, max.sz=100000, parallel.sz=3)


tmp_data <- gsva_res_sva[names(kegg_annots),row.names(p)]

# Scaling row wise
tmp_data <-
  t(scale(t(tmp_data)))
tmp_data <- as.data.frame(tmp_data, check.names=FALSE)

col_clust <-
  as.dist(1 - cor(tmp_data, method = 'pearson')) %>%
  hclust(., "ward.D2")
```

Creating new groups based on clustering

Apparently the default `cutree` will sort the clusters by number of items in each cluster. This is different than `ComplexHeatmap`. The `dendextend` version has an option to turn this "feature" off. To do this, set the confusingly named argument `order_clusters_as_data` to `FALSE`.

```
coefs_from_clusters <-
  dendextend::cutree(col_clust, 3, order_clusters_as_data=FALSE) %>%
  enframe %>%
  as.data.frame %>%
  setNames(., c('sample_id', "cluster_idx"))
coefs_from_clusters$cluster_idx <- paste0('c', coefs_from_clusters$cluster_idx)
coefs_from_clusters$cluster_idx <-
  factor(coefs_from_clusters$cluster_idx, levels = unique(coefs_from_clusters$cluster_idx))
row.names(coefs_from_clusters) <- coefs_from_clusters$sample_id
coefs_from_clusters$sample_id <- NULL
lvls <- levels(coefs_from_clusters$cluster_idx)

group_cont <-
  lapply(seq_along(lvls), function(i){
    x <- lvls[i]
    y <-
      paste(lvls[-i], collapse = '+') %>%
```

```
      # paste(lvls, collapse = '+') %>%
      paste0("(", ., ")/", length(lvls[-i]))
      # paste0("(", ., ")/", length(lvls))
    paste0(x, '-', y)
  }) %>% unlist
```

Creating gene signatures (GS)

DRP analysis

```
group_cont <- c("c2-c1", "c3-c1", "c3-c2")

mod <-
  better_model_matrix(~0 + cluster_idx, data = coefs_from_clusters)

cont_mod <-
  makeContrasts(contrasts = group_cont, levels=data.frame(mod, check.names = FALSE))

fit <-
  t(scale(t(gsva_res_sva))) %>%
  lmFit(., data.frame(mod)) # Row scaling

fit <- fit %>%
  contrasts.fit(., cont_mod) %>%
  eBayes

res <- get_limma_results(fit, group_cont)
names(res)[2] <- 'name'
res <-
  right_join(pth_data, res) %>%
  right_join(hier_kegg, .)
res <- cbind(coefficient=res$coefficient, res[,-7])
# saveRDS(res, 'lv_on_gsva_msig_kegg_pathways_from_row_scaled_clusters.rds')
```

DEG analysis

```
fit <-
  lmFit(v, data.frame(mod)) %>%
  contrasts.fit(., cont_mod) %>%
  eBayes

res <- get_limma_results(fit, group_cont)
# saveRDS(res, "gene_res_from_clusters.rds")
```

Creating GS

```
cols_to_keep <- c("coefficient", "external_gene_name", "entrezgene", "logFC", "adj.P.Val")
gene_res <- read_rds('gene_res_from_clusters.rds') %>%
```

```
  dplyr::select(cols_to_keep) %>%
  distinct
gene_res$entrezgene <- as.character(gene_res$entrezgene)

sig_pth_res <- read_rds("lv_on_gsva_msig_kegg_pathways_from_row_scaled_clusters.rds") %>%
  filter(adj.P.Val < .05) %>%
  dplyr::select(one_of("coefficient","id","logFC", "adj.P.Val")) %>%
  distinct %>%
  right_join( hier_kegg, .)

sig_gene_res <- gene_res %>%
  filter(adj.P.Val < .05)

pth_gene_res <-
  inner_join(kegg_df, gene_res)
coefficient <- pth_gene_res$coefficient
pth_gene_res <- pth_gene_res[,-3]
pth_gene_res <- cbind(coefficient, pth_gene_res)
colnames(pth_gene_res)[5:6] <-  paste0("gene_", colnames(pth_gene_res)[5:6])

degs_out_pth <- left_join(fens_pathways, pth_gene_res)
names(degs_out_pth)[1] <- "Clusters"

degs_out_pth <- degs_out_pth %>% filter(gene_adj.P.Val < .05)
# write_csv(degs_out_pth, "degs_from_selected_pathways_20200122.csv")
```