# Metabolite Fold Enrichment Summary

*Kaitlin Marquis*

*2/5/2019*

## Contents

```r
#load required packages
library(readxl)
library(Hmisc)
library(dplyr)
library(stringr)
library(ggplot2)
library(ggpubr)
library(scales)
library(reshape2)
library(data.table)
library(WriteXLS)
```

## Data Description

I compiled the fold enrichment data for the Yeoman, McMillan, Srinivasan, and Vitali metabolomics datasets for metabolites whose concentrations were determined to change signficantly between BV positive and BV negative cohorts.

```r
#load data
data <- read_xlsx('Fold Enrichment Summary.xlsx', col_names = TRUE)

#capitalize all metabolites
data$Metabolite <- capitalize(data$Metabolite)

#change fold enrichment from character to number
data$`Fold Enrichment (BV+/-)` <- as.numeric(data$`Fold Enrichment (BV+/-)`)

#filter out unknown samples
data <- data %>%
  filter(!str_detect(Metabolite, 'Unknown'))

#manually fix identical metabolites that are named differently between the data sets

##fix GHB
data[which(data$Metabolite == '4-hydroxybutyrate (GHB)'), 'Metabolite'] <- 'GHB'
```

```
data[which(data$Metabolite == '4-Hydroxybutanoic acid'), 'Metabolite'] <- 'GHB'

##fix 2-hydroxyisovalerate
data[which(data$Metabolite == '2-Hydroxyisovalerate'), 'Metabolite'] <- '2-hydroxyisovalerate'
data[which(data$Metabolite == 'Alpha-hydroxyisovalerate'), 'Metabolite'] <- '2-hydroxyisovalerate'

##fix glutamate
data[which(data$Metabolite == 'Glutamic acid (Glutamate)'), 'Metabolite'] <- 'Glutamate'

##fix GABA
data[which(data$Metabolite == 'Gamma-aminobutyrate (GABA)'), 'Metabolite'] <- 'GABA'

##fix 1,2-propanediol
data[which(data$Metabolite == '1,2-Propanediol'), 'Metabolite'] <- '1,2-propanediol'

##fix alpha-ketoglutarate
data[which(data$Metabolite == 'A_ketoglutarate'), 'Metabolite'] <- 'alpha-ketoglutarate'
data[which(data$Metabolite == 'A-Ketoglutaric acid'), 'Metabolite'] <- 'alpha-ketoglutarate'

##fix alpha-hydroxyisocaproate
data[which(data$Metabolite == '2-hydroxyisocaproate'), 'Metabolite']  <- 'alpha-hydroxyisocaproate'
data[which(data$Metabolite == 'Alpha-hydroxyisocaproate'), 'Metabolite']  <- 'alpha-hydroxyisocaproate'

##fix glycerol 3-phosphate
data[which(data$Metabolite == 'G3P'), 'Metabolite']  <- 'glycerol 3-phosphate'
data[which(data$Metabolite == 'Glycerol 3-phosphate (G3P)'), 'Metabolite']  <- 'glycerol 3-phosphate'
data[which(data$Metabolite == 'Glycerol-3-P'), 'Metabolite']  <- 'glycerol 3-phosphate'

##fix glycerol 2-phosphate
data[which(data$Metabolite == 'Glycerol 2-phosphate'), 'Metabolite']  <- 'glycerol 2-phosphate'
data[which(data$Metabolite == 'Glycerol-2-P'), 'Metabolite']  <- 'glycerol 2-phosphate'

##fix Glucose-6-phosphate (G6P)
data[which(data$Metabolite == 'Glucose-6-phosphate (G6P)'), 'Metabolite']  <- 'glucose-6-phosphate'
data[which(data$Metabolite == 'Glucose-6-P'), 'Metabolite']  <- 'glucose-6-phosphate'

##fix Aspartate
data[which(data$Metabolite == 'Aspartic acid (Aspartate)'), 'Metabolite'] <- 'Aspartate'

##fix Gluconic acid
data[which(data$Metabolite == 'Gluconate'), 'Metabolite'] <- 'Gluconic acid'

##fix N-acetyl-lysine
data[which(data$Metabolite == 'N-Acetyl-Lysine'), 'Metabolite'] <- 'N-acetyl-lysine'
data[which(data$Metabolite == 'N6-acetyllysine'), 'Metabolite'] <- 'N-acetyl-lysine'

##fix N-acetyl-putrescine
data[which(data$Metabolite == 'N-acetylputrescine'), 'Metabolite'] <- 'N-acetyl-putrescine'

##fix N-acetyl-serine
data[which(data$Metabolite == 'N-Acetyl-serine'), 'Metabolite'] <- 'N-acetyl-serine'
data[which(data$Metabolite == 'N-acetylserine'), 'Metabolite'] <- 'N-acetyl-serine'
```

```
##fix N-acetyl-glucosamine
data[which(data$Metabolite == 'N-acetylglucosamine'), 'Metabolite'] <- 'N-acetyl-glucosamine'
data[which(data$Metabolite == 'N-Acetyl glucosamine'), 'Metabolite'] <- 'N-acetyl-glucosamine'

##fix Myo-inositol
data[which(data$Metabolite == 'Inositol, myo-'), 'Metabolite'] <- 'Myo-inositol'

##fix Scyllo-inositol
data[which(data$Metabolite == 'Inositol, scyllo-'), 'Metabolite'] <- 'Scyllo-inositol'

##fix Glyceric acid
data[which(data$Metabolite == 'Glyceric_acid'), 'Metabolite'] <- 'Glyceric acid'

##fix Succinate
data[which(data$Metabolite == 'Succinic acid (Succinate)'), 'Metabolite'] <- 'Succinate'

##fix Lactate
data[which(data$Metabolite == 'Lactic acid (Lactate)'), 'Metabolite'] <- 'Lactate'

##fix Acetate
data[which(data$Metabolite == 'Acetic acid (Acetate)'), 'Metabolite'] <- 'Acetate'

##fix Phenyllactate
data[which(data$Metabolite == 'Phenyllactate (PLA)'), 'Metabolite'] <- 'Phenyllactate'

##fix Dehydroascorbate
data[which(data$Metabolite == 'Dehydroascorbic acid'), 'Metabolite'] <- 'Dehydroascorbate'

##fix 2-hydroxyisoglutarate
data[which(data$Metabolite == '2-Hydroxyglutaric acid'), 'Metabolite'] <- '2-hydroxyisoglutarate'

##fix 2-aminoadipate
data[which(data$Metabolite == '2-Aminoadipate'), 'Metabolite'] <- '2-aminoadipate'
data[which(data$Metabolite == '2-Aminoadipic acid'), 'Metabolite'] <- '2-aminoadipate'

##fix 3-methyl-2-oxovalerate
data[which(data$Metabolite == '3-Methyl-2-oxovalerate'), 'Metabolite'] <- '3-methyl-2-oxovalerate'

##fix Aminomalonate
data[which(data$Metabolite == 'Aminomalonic acid'), 'Metabolite'] <- 'Aminomalonate'

##fix Azelate
data[which(data$Metabolite == 'Azelate (nonanedioate)'), 'Metabolite'] <- 'Azelate'
data[which(data$Metabolite == 'Azelaic acid'), 'Metabolite'] <- 'Azelate'

##fix beta-alanine
data[which(data$Metabolite == 'B-Alanine'), 'Metabolite'] <- 'Beta-alanine'

##fix citrate
data[which(data$Metabolite == 'Citric acid'), 'Metabolite'] <- 'Citrate'

##fix formate
data[which(data$Metabolite == 'Formic acid'), 'Metabolite'] <- 'Formate'
```

```r
##fix Fructose-6-phosphate
data[which(data$Metabolite == 'Fructose-6-P'), 'Metabolite']  <- 'fructose-6-phosphate'
data[which(data$Metabolite == 'Fructose-6-phosphate'), 'Metabolite']  <- 'fructose-6-phosphate'

##fix fumarate
data[which(data$Metabolite == 'Fumaric acid (Fumarate)'), 'Metabolite'] <- 'Fumarate'

##fix malate
data[which(data$Metabolite == 'Malic acid'), 'Metabolite'] <- 'Malate'

##fix Mannose-6-phosphate
data[which(data$Metabolite == 'Mannose-6-P'), 'Metabolite']  <- 'mannose-6-phosphate'
data[which(data$Metabolite == 'Mannose-6-phosphate'), 'Metabolite']  <- 'mannose-6-phosphate'

##fix Nicotinamide adenine dinucleotide (NAD+)
data[which(data$Metabolite == 'NAD'), 'Metabolite']  <- 'Nicotinamide adenine dinucleotide (NAD+)'

##fix pyruvate
data[which(data$Metabolite == 'Pyruvic acid'), 'Metabolite']  <- 'Pyruvate'

##fix Erythronic acid
data[which(data$Metabolite == 'Erythronate*'), 'Metabolite']  <- 'Erythronic acid'

##fix N-acetyl-glutamate
data[which(data$Metabolite == 'N-acetylglutamate'), 'Metabolite'] <- 'N-acetyl-glutamate'
data[which(data$Metabolite == 'N-Acetylglutamic acid'), 'Metabolite'] <- 'N-acetyl-glutamate'

##fix 2-hydroxyisovalerate
data[which(data$Metabolite == '3-Methyl-2-hydroxybutanoic acid'), 'Metabolite'] <- '2-hydroxyisovalerat

data$Direction <- ifelse(data$`Fold Enrichment (BV+/-)` > 1, 'Increase', 'Decrease')
data$rescale <- scales::rescale(data$`Fold Enrichment (BV+/-)`, c(0,1))
data[which(data$rescale == Inf), 'rescale'] <- 1
```

## Create heatmap

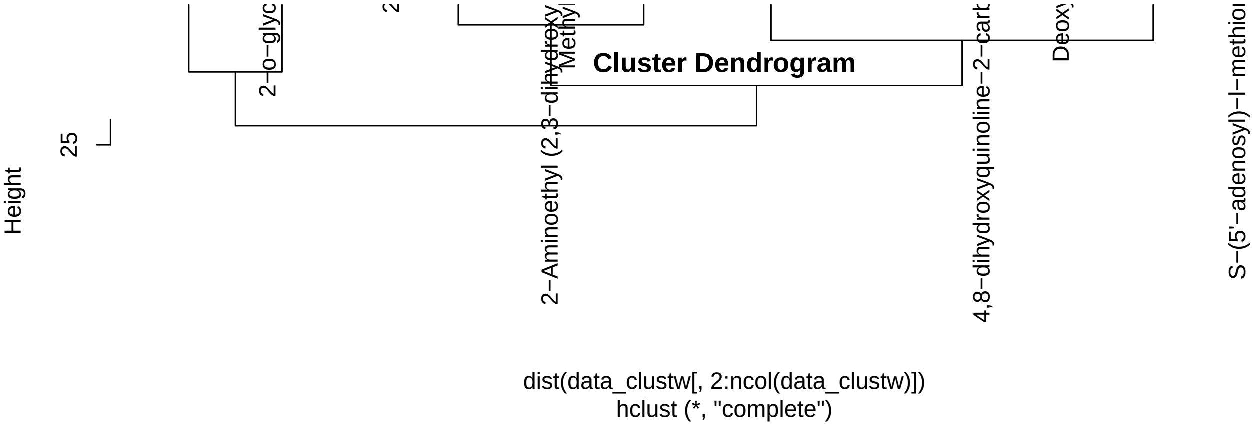
## Updated Heatmap

```r
#load data
data_lit <- read_xlsx('Metabolite Microbiome Metadata.xlsx', sheet = 'Metabolites from Literature')
data_screen <- read_xlsx('Metabolite Microbiome Metadata.xlsx', sheet = 'Metabolites from Screen')

#select relevant columns for plotting
lit <- data_lit %>%
  select(MetaboliteName, Study, `Fold Enrichment (BV+/-)`) %>%
  mutate(Direction = ifelse(as.numeric(`Fold Enrichment (BV+/-)`) > 1, 'Increase', 'Decrease')) %>%
  filter(!str_detect(MetaboliteName, 'unknown'))

screen <- data_screen %>%
  select(MetaboliteName, Source, `POC for Inf A`) %>%
  mutate(Direction = ifelse(`POC for Inf A` > 100, 'Increase', 'Decrease'))
```

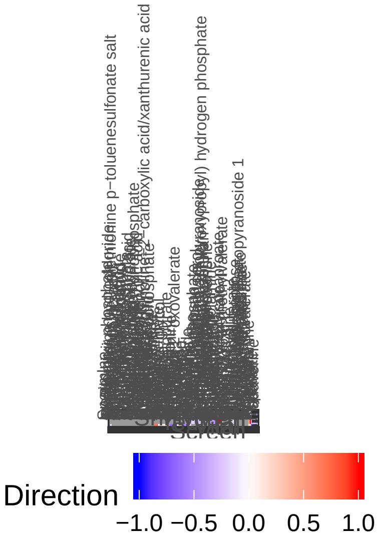
```r
screen$`POC for Inf A` <- sapply(screen$`POC for Inf A`, function(x) {x/100}) #change from percent to f

#change colnames to artificially join
colnames(lit) <- c('Metabolite', 'Source', 'FC', 'Direction')
colnames(screen) <- c('Metabolite', 'Source', 'FC', 'Direction')
screen$Source <- rep('Screen', nrow(screen)) #overwrite screen values to be the same so they are plotte

#clean up dataframes
lit$Metabolite <- capitalize(as.character(lit$Metabolite))
lit$Metabolite <- gsub('Alpha-hydroxyisovalerate', '2-hydroxyisovalerate', lit$Metabolite)
lit$Metabolite <- gsub('Nad+', 'NAD', lit$Metabolite)

screen$Metabolite <- capitalize(as.character(screen$Metabolite))
screen$Metabolite <- gsub('Alpha-hydroxyisovalerate', '2-hydroxyisovalerate', screen$Metabolite)
screen$Metabolite <- gsub('Nad+', 'NAD', screen$Metabolite)

#rbind data
data_merge <- rbind.data.frame(lit, screen)


#filter Srinivasan data
data_merge_noSrini <- data_merge %>%
  filter(Source != 'Srinivasan')

#Srinivasan filtered data in one or two
data_merge_Srinionly <- data_merge %>%
  filter(Source == 'Srinivasan')

data_mergefilter <- data_merge_Srinionly %>%
  filter(Metabolite %in% data_merge_noSrini$Metabolite)

#dataset with filtered Srinivasan
data_mergefilt <- rbind.data.frame(data_mergefilter, data_merge_noSrini) %>%
  arrange(FC)

#rescale by literature vs screen
merge_lit <- data_mergefilt %>%
  dplyr::filter(Source != 'Screen') %>%
  mutate(Direction = ifelse(as.numeric(FC) >1, 'Increase', 'Decrease'))

rescale_FUN <- function (x, direction) {
  subset_data <- subset(x, x$Direction == direction)
  subset_data$rescale <- rescale(as.numeric(subset_data$FC), to = c(.1,1)) #change rescale 0 to .1
  return(subset_data)
}

merge_litI <- rescale_FUN(merge_lit, 'Increase')
merge_litD <- rescale_FUN(merge_lit, 'Decrease')

merge_lit <- rbind.data.frame(merge_litD, merge_litI)

merge_screen <- data_mergefilt %>%
  filter(Source == 'Screen') %>%
  mutate(Direction = ifelse(as.numeric(FC) >1, 'Increase', 'Decrease'))

rescale_screen_inhibitors <- function (x, direction) {
  subset_data <- subset(x, x$Direction == direction)
  subset_data$rescale <- rescale(as.numeric(subset_data$FC), to = c(1,.1)) #change rescale 0 to .1
  return(subset_data)
}

merge_screenI <- rescale_FUN(merge_screen, 'Increase')
merge_screenD <- rescale_screen_inhibitors(merge_screen, 'Decrease')

merge_screen <- rbind.data.frame(merge_screenD, merge_screenI)

merge <- rbind.data.frame(merge_lit, merge_screen)
merge[which(merge$rescale == Inf), 'rescale'] <- 1

#cast merge
cast_merge <- dcast(setDT(merge),Metabolite ~ Source, value.var = c('rescale','Direction'))
melt_merge <- reshape(cast_merge, varying = 2:ncol(cast_merge), sep = '_', direction = 'long')
colnames(melt_merge) <- c('Metabolite', 'Source', 'rescale', 'Direction', 'ID')

#arrange order
melt_merge$Source <- ordered(melt_merge$Source, levels = c('McMillan', 'Srinivasan', 'Yeoman', 'Vitali'

#change rescale values to be positive or negative
melt_merge$value <- ifelse(melt_merge$Direction == 'Decrease', melt_merge$rescale*-1, melt_merge$rescal
```

## Sorted and subsetted heatmap

```
#cluster metabolites to arrange heatmap

#first make rescale values negative to allow for clustering
data_clust <- merge %>%
  mutate(rescale_clust = ifelse(merge$Direction == 'Decrease', merge$rescale*-1, merge$rescale)) %>%
  select(Metabolite, Source, rescale_clust)

#spread data to wide format
data_clustw <- dcast(setDT(data_clust),Metabolite ~ Source, value.var = 'rescale_clust')

#change NAs to arbitrary value of 10 because data was rescaled to 0
data_clustw <- as.data.frame(sapply(data_clustw, function (x) ifelse(is.na(x) == TRUE, 10, x)))

#change data to numeric
data_clustw <- cbind.data.frame(data_clustw[,1], sapply(data_clustw[,-1], function (x) {as.numeric(as.c
colnames(data_clustw) <- c('Metabolite', 'McMillan', 'Screen', 'Srinivasan', 'Vitali', 'Yeoman')
rownames(data_clustw) <- data_clustw$Metabolite

#cluster
hc <- hclust(dist(data_clustw[,2:ncol(data_clustw)]))
plot(hc)
```



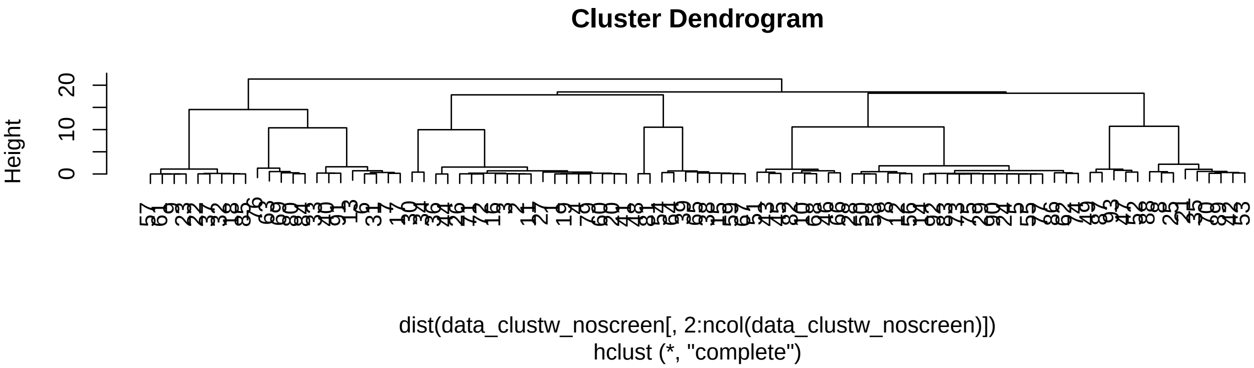
```
#obtain cluster order
hc_order <- hc$order

#reorder metabolites by hc_order
data_clustw_ordered <- data_clustw[hc_order, 'Metabolite']

#factor melt_merge dataset by hc_order
melt_merge$Metabolite <- factor(melt_merge$Metabolite, levels = data_clustw_ordered)

plot_ordered <- ggplot(melt_merge, aes(x = Metabolite, y = Source, fill = Direction)) +
  geom_tile(aes(fill = value)) +
  theme_bw() +
  theme(panel.background = element_rect(fill = NA, color = 'black'),
        panel.grid = element_blank()) +
  scale_fill_gradient2(low = 'blue', mid = 'white', high = 'red', na.value = 'gray60') +
  coord_fixed(ratio = 4) +
  scale_x_discrete(position = 'top') +
  theme(legend.position = 'bottom', axis.text.x = element_text(angle = 90, size = 6, hjust = 0, vjust =
  labs(x = NULL, y = NULL)

plot_ordered
```



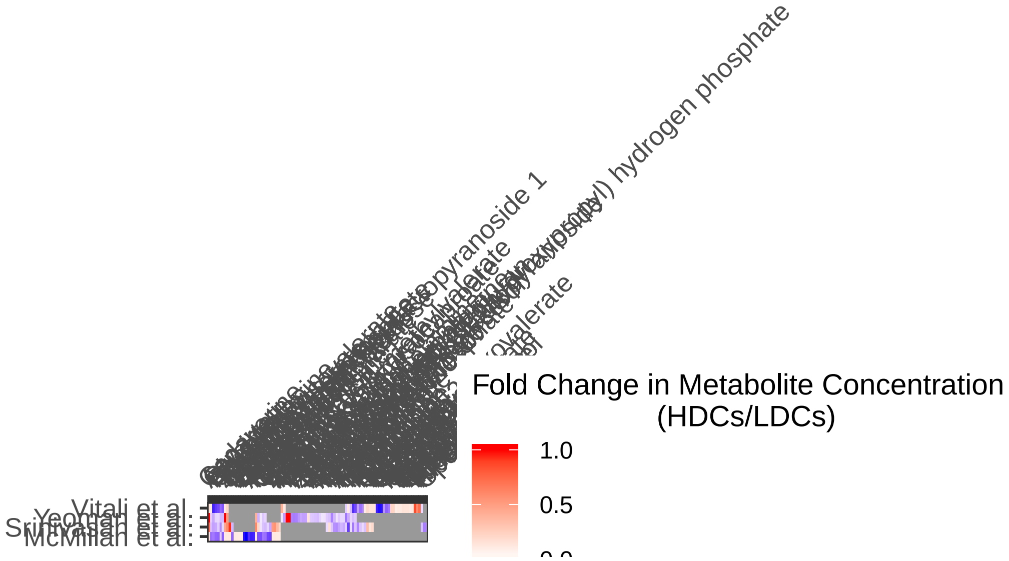
```
ggsave(filename = 'Ordered plot.pdf')
```

```
## Saving 10 x 3 in image
```

```
#filter out screen data
data_clust_noscreen <- data_clust %>%
  filter(Source == 'McMillan' | Source == 'Yeoman' | Source == 'Vitali' | Source == 'Srinivasan')

#spread data to wide format
data_clustw_noscreen <- dcast(setDT(data_clust_noscreen), Metabolite ~ Source, value.var = 'rescale_clu

#change NAs to arbitrary value of 10 because data was rescaled to 0
data_clustw_noscreen <- as.data.frame(sapply(data_clustw_noscreen, function (x) ifelse(is.na(x) == TRUE

#write information to excel file to add metabolite class for relevant molecules
WriteXLS(data_clustw_noscreen, ExcelFileName = 'Significant_lit_metabolites.xlsx')

#cluster
hc_noscreen <- hclust(dist(data_clustw_noscreen[,2:ncol(data_clustw_noscreen)]))
plot(hc_noscreen)
```



```
#obtain cluster order
hc_order_noscreen <- hc_noscreen$order

#reorder metabolites by hc_order
data_clustw_ordered_noscreen <- data_clustw_noscreen[hc_order_noscreen, 'Metabolite']

#select metabolites from melt merge in data_clustw_ordered_noscreen
metabolites_noscreen <- data_clustw_noscreen$Metabolite

melt_merge_noscreen <- melt_merge %>%
  filter(Source == 'McMillan' | Source == 'Yeoman' | Source == 'Vitali' | Source == 'Srinivasan') %>%
  filter(as.character(Metabolite) %in% metabolites_noscreen)

#plot
noscreen_plot <- ggplot(melt_merge_noscreen, aes(x = Metabolite, y = Source, fill = Direction)) +
    geom_tile(aes(fill = value)) +
  theme_bw() +
  theme(panel.background = element_rect(fill = NA, color = 'black'),
      panel.grid = element_blank()) +
  scale_fill_gradient2(low = 'blue', mid = 'white', high = 'red', na.value = 'gray60') +
  coord_fixed(ratio = 4) +
  scale_x_discrete(position = 'top') +
  theme(axis.text.x = element_text(angle = 45, size = 9.5, hjust = 0, vjust = 0),
        axis.text.y = element_text(size = 9.5),
        legend.position = 'right',
        legend.direction = 'vertical',
        legend.title.align = .5,
        legend.text.align = .5) +
  labs(x = NULL, y = NULL, fill = 'Fold Change in Metabolite Concentration \n (HDCs/LDCs)') +
  scale_y_discrete(labels = c('McMillan et al.', 'Srinivasan et al.', 'Yeoman et al.', 'Vitali et al.')


noscreen_plot
```

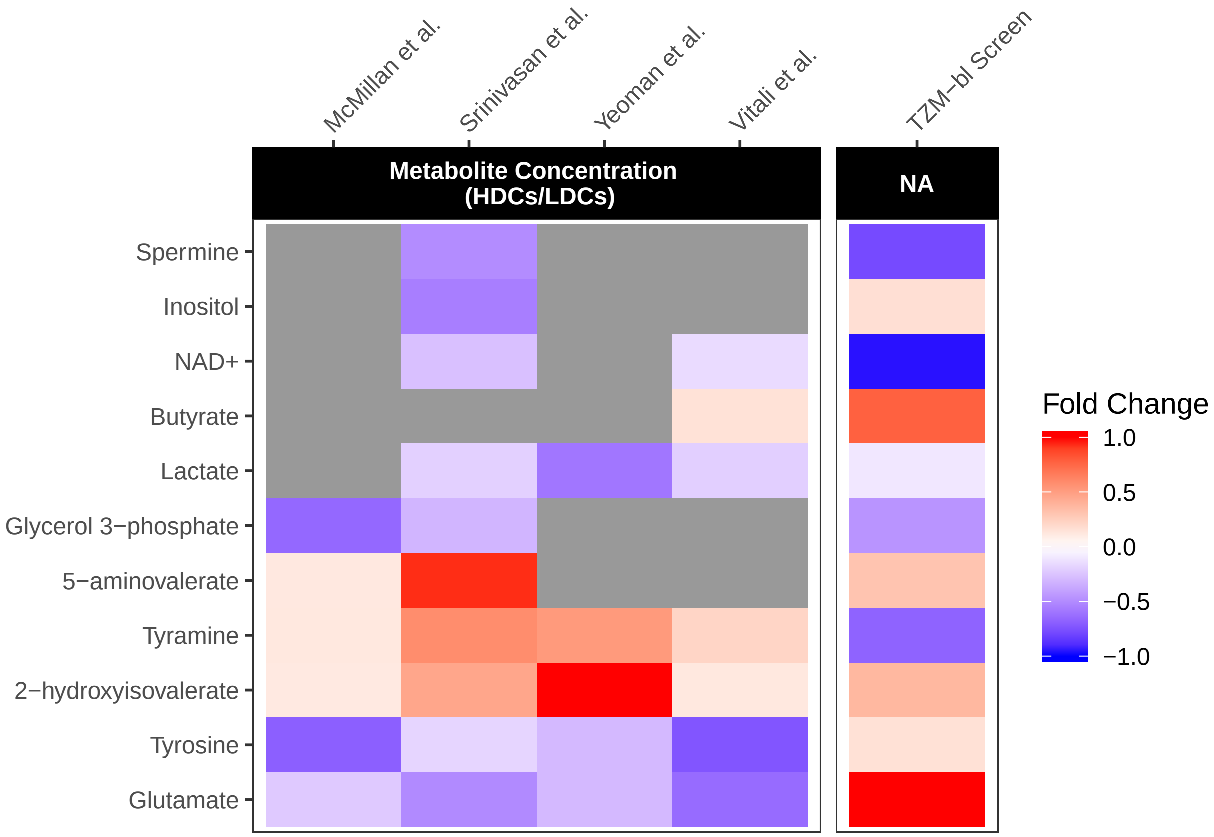


```
ggsave(filename = 'Metabolomics_Data_only.pdf')
```

```
## Saving 10 x 3 in image
```

## Cross reference metabolomics data with screen data

```
#filter metabolite dataset to only show metabolites that overlapped in the screen and literature
subset_metabolites <- melt_merge %>% filter(Metabolite %in% screen$Metabolite) %>% filter(Metabolite %i

colnames(subset_metabolites) <- c('Metabolite', 'Source', 'Relative Fold Change', 'Direction', 'ID', 'va

#add groups to data
subset_metabolites$Group <- ifelse(subset_metabolites$Source == 'Screen', 'HIV Replication ', 'Metaboli
subset_metabolites$Group <- factor(subset_metabolites$Group, levels = c('Metabolite Concentration \n (HI

#filter plot to only show metabolites that overlapped the screen and the literates
plot_subset <- ggplot(subset_metabolites, aes(x = Source, y = Metabolite, fill = Direction)) +
  facet_grid(~Group, scales = 'free', shrink = F, space = 'free_x') +
  geom_tile(aes(fill = value)) +
  theme_bw() +
  theme(panel.background = element_rect(fill = NA, color = 'black'),
        panel.grid = element_blank(),
        panel.grid.major = element_line(colour = "black")) +
    scale_fill_gradient2(low = 'blue', mid = 'white', high = 'red', na.value = 'gray60', limits = c(-1,
    #coord_fixed(ratio = .8) +
    scale_x_discrete(labels = c('McMillan' = 'McMillan et al.', 'Srinivasan' = 'Srinivasan et al.', 'Ye
    theme(legend.position = 'right',
          axis.text.x = element_text(angle = 45, hjust = 0, vjust = 0),
          strip.background = element_rect(color = 'black', fill = 'black'),
          strip.text.x = element_text(color = 'white', face = 'bold')) +
    labs(x = NULL, y = NULL, fill = 'Fold Change')

plot_subset
```



```
ggsave(filename = 'Subset plot.pdf')
```

```
## Saving 6.5 x 4.5 in image
```