Broliden_5325

21 October, 2020

Contents

#Load libraries and other scripts

```
library(igraph)
library(rafalib)
library(sva)
library(batchelor)
source("~/repos/niceRplots/R/plotting_functions.R")
source("~/repos/niceRplots/R/add_fig_label.R")
source("~/repos/niceRplots/R/helper_functions.R")
```

#Defining some variables for the analysis

```
PATH <- "~/Desktop/NBIS/SMS_Projects/broliden_5325/" #Path to the data

#color palettes
pal <- RColorBrewer::brewer.pal(8,"Set1") #color pallete for plots
heat_pal <- c("#000000", colorRampPalette(c("#000000","grey5","grey30","orange3","yellow","yellow","whi
cor_pal <- colorRampPalette(c("navy","white","firebrick"))(90)


#Graph construction
fct <- .5 # FC threshold for differential expression
pvt <- 0.01 # Pvalue threshold for differential expression
min_pct <- .1 # minimun level of detected bacteria in each sample group
```

Loading data and metadata

```
# Load metadata
metadata <- read.csv(paste0(PATH,"/data/Clinical_visit_2_3_updatesept28.csv"))
rownames(metadata) <- as.character(metadata$ID)

# Load datasets
dataset_names <- c("Tissue_RNAseq_V3_normalized",
                   "ASV_tissue_V3_normalized_batch_corrected",
                   "ASV_CVL_V3_normalized_batch_corrected",
                   "ASV_CVL_V2_normalized_batch_corrected",
                   "ASV_CVL_V2_normalized_NOT_batch_corrected")

#import datasets and return matrix with taxa-level resolution
```

```
datasets <- lapply(dataset_names, function(x){
  x <- read.csv( paste0(PATH,"results/",x,".csv"),row.names = 1)
  return(as.matrix(x)) })
names(datasets) <- dataset_names


#fill all samples with 0s to make plotting easier
datasets_all_samples <- lapply(datasets, function(x) {
  temp <- matrix(0,nrow = nrow(x),
                 ncol = nrow(metadata),
                 dimnames = list(rownames(x),rownames(metadata)))
  temp[rownames(x),colnames(x)] <- x
  return(temp)
})
lapply(datasets_all_samples,dim)
```

Comparing CVL2 samples before and after batch correction

```
mypar(1,2,mar=c(2,15,1,1))

for( dataset in c("ASV_CVL_V2_normalized_NOT_batch_corrected",
            "ASV_CVL_V2_normalized_batch_corrected")){

  # Computing differential expression across batches
  all_microbiome <- cbind(datasets_all_samples[[dataset]])
  datasets <- read.csv(paste0(PATH,"/results/batches_CVL2.csv"))[,2]

  all_microbiome <- all_microbiome[,as.character(read.csv(paste0(PATH,"/results/batches_CVL2.csv"))[,1])]

  NN <- min(table(datasets))


  res <- data.frame( matrix(0,nrow = 1,ncol = 6) )
  for(i in levels(datasets)){
    for(j in rownames(all_microbiome) ){

      set.seed(1)
      a <- c(sample(all_microbiome[j,datasets == i],NN))
      set.seed(1)
      b <- sample(all_microbiome[j,datasets != i],NN)

      perc1 <- sum(all_microbiome[j,datasets == i]>0) / sum(datasets == i)
      perc2 <- sum(all_microbiome[j,datasets != i]>0) / sum(datasets != i)

      temp <- wilcox.test(x=a, y=b,exact = F)

      fc <- log2( (mean(a)+1e-3) / (mean(b)+1e-3) )

      res <- rbind(res, setNames(c(j,i,fc,perc1,perc2,unlist(temp)[2] ),
                                 c("bacteria","cluster","fc","perc.1","perc.2","pvalue")) )
      colnames(res) <- c("bacteria","cluster","fc","perc.1","perc.2","pvalue")
    }
```

```
  }


  res <- res[-1,]
  res$pvalue <- as.numeric(res$pvalue)
  res$perc.1 <- as.numeric(res$perc.1)
  res$perc.2 <- as.numeric(res$perc.2)
  res$perc.diff <- res$perc.1 - res$perc.2

  res$fc <- as.numeric(res$fc)
  res$FDR <- p.adjust(res$pvalue)
  res <- res[order(res$pvalue),]
  res <- res[res$fc > fct,]
  res <- res[abs(res$pvalue) < 0.05,]
  res <- res[ (res$perc.1 > min_pct) | (res$perc.2 > min_pct) ,]
  dim(res)

  ord <- factor(sapply(unique(as.character(res$bacteria)),function(x){getcluster(all_microbiome, x, data

  plot_dots( all_microbiome , names(sort(ord)) , clustering = datasets, main = dataset,show_grid = T,cex
             cex.row = .8,cex.col = .8,srt = 0)
}
```