#!/usr/bin/env Rscript

suppressPackageStartupMessages(library("optparse"))

suppressPackageStartupMessages(library("stats"))

option\_list <- list(

make\_option(c("-f", "--infile"),action = "store",type = "character", help = "The Input file"),

make\_option(c("-p", "--group"),action = "store",type = "character",default = "Unbound", help = "The group name"),

make\_option(c("-t", "--outtitle"),action = "store",type = "character",default = "Terms", help = "The title of outimage;default = Term Plot"),

make\_option(c("-n", "--outname"),action = "store",type = "character",default = "term", help = "The name of outimage;default = term.pdf")

)

opt <- parse\_args(OptionParser(option\_list = option\_list))

# opt$infile <- "test.txt"

# opt$infile <- "/users/chengc/dev2016/graphicwork/Macaque/3\_pca/All\_sample\_LncRNA\_exp\_RPKM\_annot\_gencode.xls.lncRNA"

# opt$group <- "/users/chengc/dev2016/graphicwork/Macaque/3\_pca/Sample\_group.txt"

library("ggplot2")

# library("factoextra")

# library("FactoMineR")

# library("stats")

library("ggthemes")

theme\_paper <- theme(

# panel.border = element\_rect(fill = NA,colour = "black"),

# panel.grid.major = element\_line(colour = "grey88",

# size = 0.2),

# panel.grid.minor = element\_line(colour = "grey95",

# size = 0.5),

# axis.text.x= element\_text(vjust = 1,hjust = 1, angle = 45),

# legend.position = "top",

# legend.direction = "horizontal",

panel.grid.major.x = element\_line(colour = "grey70",

size = 0.2),

legend.title = element\_text(size = 14),

legend.text = element\_text(size = 14),

axis.title = element\_text(size = 14, face = "bold"),

plot.title = element\_text(size = 14,face = "bold"),

axis.text = element\_text(size = 14, colour = "black"))

data = read.table(opt$infile, header=T, com='', quote='',sep="\t", check.names=F)

data <- data[order(data[,6]),]

rlen <- length(rownames(data))

if (rlen == 0){

print("null data,exit")

q()

}

# data <- subset(primary\_data,Module==opt$group)

data$lp <- -log(data[,6]+0.00000000001,10)

if (rlen>10){

data <- data[1:10,]

}

# print(data)

rlen <- length(rownames(data))

clen <- length(colnames(data))

maxlp <- max(data$lp)

#M1

# data$PT = factor(data$Term,levels = c("Terpenoid backbone biosynthesis","Sulfur relay system","Nicotinate and nicotinamide metabolism","Purine metabolism","DNA replication","Protein export","Glycerophospholipid metabolism","Bacterial secretion system","Homologous recombination","Pyrimidine metabolism","Mismatch repair","Peptidoglycan biosynthesis","Aminoacyl-tRNA biosynthesis","Ribosome"))

# #M2

# data$PT = factor(data$Term,levels = c("Pentose phosphate pathway","Microbial metabolism in diverse environments","Alanine, aspartate and glutamate metabolism","Glycerolipid metabolism","Lysine biosynthesis","2-Oxocarboxylic acid metabolism","Nitrogen metabolism","Histidine metabolism","ABC transporters"))

# #M3

# data$PT = factor(data$Term,levels = c("Microbial metabolism in diverse environments","Fatty acid biosynthesis","Citrate cycle (TCA cycle)","Biosynthesis of secondary metabolites","Pyruvate metabolism","Metabolic pathways","Amino sugar and nucleotide sugar metabolism","Starch and sucrose metabolism","Propanoate metabolism","Pentose and glucuronate interconversions","Ascorbate and aldarate metabolism","Phosphotransferase system (PTS)","Galactose metabolism"))

# #M4

# data$PT = factor(data$Term,levels = c("Alanine, aspartate and glutamate metabolism","Glycine, serine and threonine metabolism","Microbial metabolism in diverse environments","RNA degradation","Purine metabolism","Glycolysis / Gluconeogenesis","Butanoate metabolism"))

# #M1\_HMP

# data$PT = factor(data$Term,levels = c("Arginine and proline metabolism","Degradation of aromatic compounds","Purine metabolism","Riboflavin metabolism","ABC transporters","Bacterial chemotaxis","Phosphonate and phosphinate metabolism","Bacterial secretion system","Pentose and glucuronate interconversions","Flagellar assembly"))

#Unbound

# data$PT = factor(data$Term,levels = c("RNA degradation","Nicotinate and nicotinamide metabolism","Biosynthesis of amino acids","Homologous recombination","Aminoacyl-tRNA biosynthesis","Pyrimidine metabolism","Amino sugar and nucleotide sugar metabolism","Oxidative phosphorylation","Biosynthesis of secondary metabolites","Metabolic pathways"))

#unUnbound

for(i in 1:rlen){

data[i,clen+1] <- paste(strwrap(data[i,1],width = 50), collapse = "\n")

maxlen <- 90

if (nchar(data[i,clen+1]) > maxlen ){

while (substr(data[i,clen+1], maxlen, maxlen)!=" " && substr(data[i,clen+1], maxlen, maxlen)!="\n" && nchar(data[i,clen+1]) > maxlen){

maxlen<- maxlen+1

}

data[i,clen+1] <- strtrim(data[i,clen+1],maxlen-1)

data[i,clen+1] <- paste(data[i,clen+1],"..")

}

}

data[,clen+1] = factor(data[,clen+1],levels = rev(data[,clen+1]))

ggplot(data, aes(x=data[,clen+1], y=lp)) +

#

# geom\_bar(stat="identity",fill=rgb(212,93,24, maxColorValue = 255),width=0.5) + theme\_minimal() +

# geom\_bar(aes(fill=data$PValue),stat="identity",width=0.5) + theme\_minimal() + scale\_fill\_gradient(low="#198752",high = "#c0f3da")+

#Unbound

geom\_bar(aes(fill=data[,7]),stat="identity",width=0.5) + scale\_fill\_gradient(low="#dd4a41",high = "#e98882")+

#unUnbound

# geom\_bar(aes(fill=data$CP),stat="identity",width=0.5) + scale\_fill\_gradient(low="#2f8f90",high = "#5bc8ca")+

#M2

# geom\_bar(aes(fill=data$CP),stat="identity",width=0.5) + theme\_minimal() + scale\_fill\_gradient(low="#00578b",high = "#5994cd")+

# #M3

# geom\_bar(aes(fill=data$CP),stat="identity",width=0.5) + theme\_minimal() + scale\_fill\_gradient(low="#a53f2a",high = "#f0ccc4")+

# #M4

# geom\_bar(aes(fill=data$CP),stat="identity",width=0.5) + theme\_minimal() + scale\_fill\_gradient(low="#d8bd00",high = "#ffee73")+

# #M1 HMP

# geom\_bar(aes(fill=data$CP),stat="identity",width=0.5) + theme\_minimal() + scale\_fill\_gradient(low="#00e0e1",high = "#a3f4f4")+

coord\_flip() + theme\_minimal()+theme\_paper+

geom\_text(aes(x=data[,clen+1],y=lp + 1.6 \* sign(lp),label=paste(data[,4],"/", data[,5],sep = "")),hjust=0.5, size=6.5,

color="black")+

ylim(0,maxlp+3)+

labs(x="Pathway/Term",

y="-log10 Pvalue",

title=opt$outname,

fill="Corrected Pvalue")

ggsave(paste(opt$outname,".pdf",sep = ""), width = 270, height = 130, units = "mm")

ggsave(paste(opt$outname,".png",sep = ""), width = 270, height = 130, units = "mm", dpi=600)

# ggsave(paste(opt$outname,".80.png",sep = ""), width = 270, height = 130, units = "mm", dpi=80)

# ggsave(paste(opt$outname,".300.png",sep = ""), width = 270, height = 130, units = "mm", dpi=300)

# ggsave(paste(opt$outname,".600.png",sep = ""), width = 270, height = 130, units = "mm", dpi=600)