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", help = "The group file"),

make\_option(c("-e", "--ellipsetype"),action = "store",type = "character", default = "convex",help = "ellipsetype: t,norm,euclid,convex"),

make\_option(c("-c","--column"),action = "store",type = "integer",default = 2,

help = "the column number for pca group"),

make\_option(c("-t", "--text"),action = "store\_true",type = "logical",default = FALSE, help = "annotate for each point, [default]"),

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

)

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

# opt$infile <- "/users/chengc/dev2016/graphicwork/Macaque/3\_pca/All\_sample\_merged\_RPKM\_edSymbol.xls.mRNA"

# 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("factoextra")

library("FactoMineR")

library("stats")

library("ggthemes")

NO\_REUSE = F

# opt$file = "Sample\_correlation.dat"

# opt$file = "test.txt"

# # get the filename to use later

# # try to reuse earlier-loaded data if possible

# print('Reading matrix file.')

primary\_data = read.table(opt$infile, header=T, com='', sep="\t", row.names=1, check.names=F)

data.class <- read.table(opt$group,header=T,row.names=1,sep='\t')

primary\_data <- primary\_data[rowSums(primary\_data) != 0,]

primary\_data <- primary\_data[rowMeans(primary\_data) != max(primary\_data),]

# head(primary\_data)

data\_matrix <- as.matrix(primary\_data)

data\_matrix <- t(data\_matrix)

data\_matrix1 <- data\_matrix[,colSums(data\_matrix[,1:ncol(data\_matrix)]) > 0]

data\_matrix1 <- data\_matrix1[order(row.names(data\_matrix1)),]

data.class <- data.class[order(row.names(data.class)),]

# data.class

# dd <- merge(data\_matrix1[,c(1:10)],data.class,by=0,all.x=TRUE)

data.pca <- prcomp(data\_matrix1, scale = TRUE)

data.pca$x

cat(c('sample\t'),file=paste(opt$outname,".PCA.xls",sep=''))

write.table(data.pca$x, file=paste(opt$outname,".PCA.xls",sep=''), quote=F, append=T, sep='\t')

pc <- predict(data.pca)

et <- opt$ellipsetype

a <- as.numeric(opt$column)

a <- a - 1

data.class[,a]

# data.pca <- PCA(data\_matrix1, graph = FALSE)

# f <- fviz\_pca\_ind(data.pca, label="none", habillage=data.class[,a],addEllipses=TRUE, ellipse.level=0.95,ellipse.type="t",pointsize=4)

# "t", "norm", "euclid","convex"

p <- fviz\_pca\_ind(data.pca, label="none", habillage=data.class[,a],addEllipses=TRUE, ellipse.level=0.95,ellipse.type=et,pointsize=2)

if (opt$text){

p <- p +

geom\_text(aes(label=row.names(data.pca$x)),hjust=0, vjust=0)

}

p <- p +

# scale\_shape\_manual(values = c(20,18,20,18,20,18,20,18,20)) +

scale\_color\_brewer(palette="Set1", direction=-1) +

scale\_fill\_brewer(palette="Set1", direction=-1) +

theme\_bw() +

theme(

title = element\_text(size = 12),

# legend.position = c(0.78, 0.7),

legend.title = element\_text(size = 12),

legend.text = element\_text(size = 12),

legend.key.height = unit(0.7, "cm"),

legend.key.width = unit(0.7, "cm"),

axis.text = element\_text(size = 12),

axis.title = element\_text(size = 12))

ggsave(paste(opt$outname,"\_0.5.pdf",sep = ""), width = 150, height = 100, units = "mm")

# pdf(file = paste(opt$outname,"\_PCA.pdf",sep=''))

# # tiff("PCA\_plot\_time.tif",2400,2400,compression ='lzw',res=300)

# print(p)

# dev.off()

# f <- fviz\_pca\_biplot(data.pca,

# habillage = data.class[,1], addEllipses = TRUE,

# col.var = "red", alpha.var ="cos2",

# label = "var") +

# scale\_color\_brewer(palette="Dark2")+

# theme\_minimal()

# f