MICROARRAY GSE100924 study

Dani Ibañez

19/3/2021

library(knitr)  
knitr::opts\_chunk$set(echo = TRUE, message = FALSE, warning = FALSE,   
 comment = NA, prompt = TRUE, tidy = FALSE,   
 fig.width = 7, fig.height = 7, fig\_caption = TRUE,  
 cache=FALSE)  
#Sys.setlocale("LC\_TIME", "C")

#{r setting directories} #setwd(".") #dir.create("data") #dir.create("results") #

> targets <- read.csv2("c:/MASTER/GITLOCAL/MICROARRAY2/data/targets.csv", header = TRUE, sep = ";")   
> knitr::kable(targets, booktabs = TRUE, caption = 'Content of the targets file used for the current analysis')

Content of the targets file used for the current analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FileName | Group | Genotype | Temperature | ShortName |
| GSM2696488\_WT\_RT\_1.CEL | WT.RT | WT | RT | WT.RT.1 |
| GSM2696489\_WT\_RT\_2.CEL | WT.RT | WT | RT | WT.RT.2 |
| GSM2696490\_WT\_RT\_3.CEL | WT.RT | WT | RT | WT.RT.3 |
| GSM2696491\_KO\_RT\_1.CEL | KO.RT | KO | RT | KO.RT.1 |
| GSM2696492\_KO\_RT\_2.CEL | KO.RT | KO | RT | KO.RT.2 |
| GSM2696493\_KO\_RT\_3.CEL | KO.RT | KO | RT | KO.RT.3 |
| GSM2696494\_WT\_Cold\_1.CEL | WT.COLD | WT | COLD | WT.COLD.1 |
| GSM2696495\_WT\_Cold\_2.CEL | WT.COLD | WT | COLD | WT.COLD.2 |
| GSM2696496\_WT\_Cold\_3.CEL | WT.COLD | WT | COLD | WT.COLD.3 |
| GSM2696497\_KO\_Cold\_1.CEL | KO.COLD | KO | COLD | KO.COLD.1 |
| GSM2696498\_KO\_Cold\_2.CEL | KO.COLD | KO | COLD | KO.COLD.2 |
| GSM2696499\_KO\_Cold\_3.CEL | KO.COLD | KO | COLD | KO.COLD.3 |

> if (!requireNamespace("BiocManager", quietly = TRUE))  
+ install.packages("BiocManager")  
> BiocManager::install()  
>   
> if(!(require(knitr))) install.packages("knitr")  
> if(!(require(colorspace))) install.packages("colorspace")  
> if(!(require(gplots))) install.packages("gplots")  
> if(!(require(ggplot2))) install.packages("ggplot2")  
> if(!(require(ggrepel))) install.packages("ggrepel")  
> if(!(require(htmlTable))) install.packages("htmlTable")  
> if(!(require(prettydoc))) install.packages("prettydoc")  
> if(!(require(devtools))) install.packages("devtools")  
> if(!(require(BiocManager))) install.packages("BiocManager")  
>   
> #install.packages("Rtools")  
> #install.packages("knitr")  
> #install.packages("colorspace")  
> #install.packages("gplots")  
> #install.packages("ggplot2")  
> #install.packages("ggrepel")  
> #install.packages("htmlTable")  
> #install.packages("prettydoc")  
> #install.packages("devtools")  
> #install.packages("BiocManager")  
> #BiocManager::install("oligo")  
> #BiocManager::install("pd.mogene.2.1.st")  
> #BiocManager::install("arrayQualityMetrics")  
> #BiocManager::install("pvca")  
>   
> # NOT NEEDED UNTIL ANALYSES ARE PERFORMED  
> # BiocManager::install("limma")  
> # BiocManager::install("genefilter")  
> # BiocManager::install("mogene21sttranscriptcluster.db")  
> # BiocManager::install("annotate")  
> # BiocManager::install("org.Mm.eg.db")  
> # BiocManager::install("ReactomePA")  
> # BiocManager::install("reactome.db")

> library(oligo)  
> celFiles <- list.celfiles("c:/MASTER/GITLOCAL/MICROARRAY2/data", full.names = TRUE)  
> library(Biobase)  
> my.targets <-read.AnnotatedDataFrame(file.path("c:/MASTER/GITLOCAL/MICROARRAY2/data","targets.csv"),   
+ header = TRUE, row.names = 1,   
+ sep=";")   
> brut\_Data <- read.celfiles(celFiles, phenoData = my.targets)

Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696488\_WT\_RT\_1.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696489\_WT\_RT\_2.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696490\_WT\_RT\_3.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696491\_KO\_RT\_1.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696492\_KO\_RT\_2.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696493\_KO\_RT\_3.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696494\_WT\_Cold\_1.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696495\_WT\_Cold\_2.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696496\_WT\_Cold\_3.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696497\_KO\_Cold\_1.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696498\_KO\_Cold\_2.CEL  
Reading in : c:/MASTER/GITLOCAL/MICROARRAY2/data/GSM2696499\_KO\_Cold\_3.CEL

> class(brut\_Data)

[1] "GeneFeatureSet"  
attr(,"package")  
[1] "oligoClasses"

> my.targets@data$ShortName->rownames(pData(brut\_Data))  
>   
> colnames(brut\_Data) <-rownames(pData(brut\_Data))   
>   
> head(brut\_Data)

GeneFeatureSet (storageMode: lockedEnvironment)  
assayData: 6 features, 12 samples   
 element names: exprs   
protocolData  
 rowNames: WT.RT.1 WT.RT.2 ... KO.COLD.3 (12 total)  
 varLabels: exprs dates  
 varMetadata: labelDescription channel  
phenoData  
 rowNames: WT.RT.1 WT.RT.2 ... KO.COLD.3 (12 total)  
 varLabels: Group Genotype Temperature ShortName  
 varMetadata: labelDescription channel  
featureData: none  
experimentData: use 'experimentData(object)'  
Annotation: pd.mogene.2.1.st

> #library(arrayQualityMetrics)  
> #arrayQualityMetrics(brut\_Data, force=TRUE)

> library(ggplot2)  
> library(ggrepel)  
> plotPCA3 <- function (datos, labels, factor, title, scale,colores, size = 1.5, glineas = 0.25) {  
+ data <- prcomp(t(datos),scale=scale)  
+ # plot adjustments  
+ dataDf <- data.frame(data$x)  
+ Group <- factor  
+ loads <- round(data$sdev^2/sum(data$sdev^2)\*100,1)  
+ # main plot  
+ p1 <- ggplot(dataDf,aes(x=PC1, y=PC2)) +  
+ theme\_classic() +  
+ geom\_hline(yintercept = 0, color = "gray70") +  
+ geom\_vline(xintercept = 0, color = "gray70") +  
+ geom\_point(aes(color = Group), alpha = 0.55, size = 3) +  
+ coord\_cartesian(xlim = c(min(data$x[,1])-5,max(data$x[,1])+5)) +  
+ scale\_fill\_discrete(name = "Group")  
+ # avoiding labels superposition  
+ p1 + geom\_text\_repel(aes(y = PC2 + 0.25, label = labels),segment.size = 0.25, size = size) +   
+ labs(x = c(paste("PC1",loads[1],"%")),y=c(paste("PC2",loads[2],"%"))) +   
+ ggtitle(paste("Principal Component Analysis for: ",title,sep=" "))+   
+ theme(plot.title = element\_text(hjust = 0.5)) +  
+ scale\_color\_manual(values=colores)  
+ }