Ejercicio para la plaza de soporte a la investigación en el proyecto 'Proyecto de atención del niño con nevus congénito y melanoma infantil' (Ref. Conv-FU-17/2022)

Nerea Carrón Rodas

14th February 2022

Instalación de las librerías necesarias.

El primer paso sería descargar todos los paquetes que son necesarios para realizar el ejercicio, si se da el caso que no estén ya descargados.

```
# Instalar el paquete BiocManager si no está instalado. Es necesario para
#   instalar paquetes de Bioconductor.
if (!require("BiocManager", quietly = TRUE)){
    install.packages("BiocManager")
}

# Paquete necesario para cargar el listado de los archivos CEL
if (!require("oligo", quietly = TRUE)){
  BiocManager::install("oligo")
}

# Paquetes necesarios para generar el archivo con los datos fenotípicos
#   a partir del fichero Series Matrix File
if (!require("Biobase", quietly = TRUE)){
  BiocManager::install("Biobase")
}

if (!require("GEOquery", quietly = TRUE)){
  BiocManager::install("GEOquery")
}

# Paquete necesario para unificar toda la información de los archivos .CEL
#   y del fenotipo en un único objeto AffyBatch.
if (!require("affy", quietly = TRUE)){
  BiocManager::install("affy")
}

# Paquetes para realizar el dendograma
if (!require("ggplot2", quietly = TRUE)){
  install.packages("ggplot2")
}

if (!require("ggdendro", quietly = TRUE)){
```

```
  install.packages("ggdendro")
}

# Paquetes para realizar el análisis de componentes principales
if (!require("ggfortify", quietly = TRUE)){
  install.packages("ggfortify")
}
```

```
# Cargamos librerias necesarias
library(oligo)
library(Biobase)
library(affy)
library(ggplot2)
library(ggrepel)
library(GEOquery)
library(ggdendro)
library(stats)
library(ggfortify)
```

1. Establecer directorio de trabajo y directorio donde se guardarán los resultados

```
directory <- "~/Documentos/BusquedaTrabajo/Ref.Conv-FU-17_2022"
knitr::opts_knit$set(root.dir = directory)
out_folder <- paste(directory,"out", sep="/")
```

2. Descargar los ficheros CEL

Los descargo desde la web y utilizando la terminal de Linux los descomprimo.

```
cd ~/Descargas
mv GSE23117_RAW.tar ~/Documentos/BusquedaTrabajo/Ref.Conv-FU-17_2022/data
cd ~/Documentos/BusquedaTrabajo/Ref.Conv-FU-17_2022/data
tar xvf GSE23117_RAW.tar
mkdir CelFiles
mv *.CEL.gz CelFiles/
cd CelFiles/
gunzip *
ls
```

3. Cargar el listado de los archivos CEL.

Para ello utilizo la función list.celfiles del paquete oligo

```
celFiles <- list.celfiles(paste(directory,"data/CelFiles", sep="/"), full.names=T)
```

4. Generar el archivo con los datos fenotípicos a partir del fichero Series Matrix File

4.1. Descargar el fichero Series Matrix File

Lo descargo y descomprimo utilizando la terminal de Linux.

```
cd ~/Documentos/BusquedaTrabajo/Ref.Conv-FU-17_2022/data
wget https://ftp.ncbi.nlm.nih.gov/geo/series/GSE23nnn/GSE23117/matrix/GSE23117_series_matrix.txt.gz
gunzip GSE23117_series_matrix.txt.gz
ls
```

4.2. Generar archivo

Para ello utilizo dos paquetes diferentes de Bioconductor. GEOquery para poder leer de forma eficiente Series Matrix File utilizando la función getGEO y Biobase para poder extraer la información correspondiente a los metadatos que describen las muestras del experimento, para ello utilizo la función pData.

```
gsm <- getGEO(filename=file.path(file.path(directory, 'data'), 'GSE23117_series_matrix.txt'))
```

```
## Rows: 54675 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr  (1): ID_REF
## dbl (15): GSM569471, GSM569472, GSM569473, GSM569474, GSM569475, GSM569476, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## File stored at:
##
## /tmp/RtmpgmxqQO/GPL570.soft
```

```
# Metadatos que describen las muestras del experimento
phenotype <- pData(gsm)
```

Guardo los datos del objeto phenotype en un fichero txt.

```
write.table(phenotype, file=file.path(out_folder, 'GSE23117_phenotype.txt'),
            quote = F, row.names = F, sep = "\t")
```

5. Unificar la información de los archivos .CEL y del fenotipo en un único objeto AffyBatch.

Para ello utilizo la función read.affybatch del paquete affy.

```
data <- read.affybatch(filenames = celFiles, phenoData = phenoData(gsm))
```

6. Normalizar con el método RMA (robust multi-array average).

Para normalizar con el método RMA utilizo la función rma del paquete affy.

```
data_norm <- rma(data)
```

```
## Warning: replacing previous import 'AnnotationDbi::tail' by 'utils::tail' when
## loading 'hgu133plus2cdf'
```

```
## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head' when
## loading 'hgu133plus2cdf'
```

```
##
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

7. Realizar el dendograma mostrando como se clasifican los enfermos y los controles.

Para poder mostrar correctamente las muestras de los controles y los pacientes y dentro de estos que nivel de la enfermedad padecen (disease status), creo un objeto llamado dist_stat donde se asocia el ID del paciente (GEO accession) con el 'disease status'.

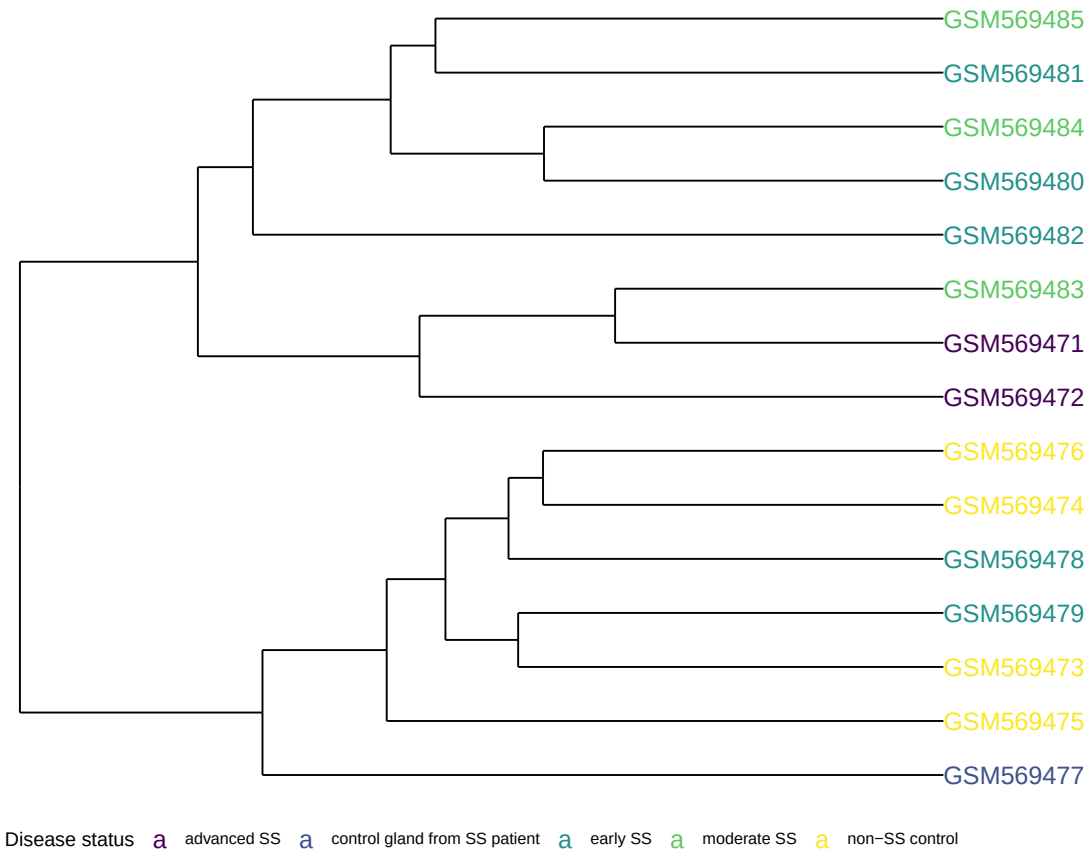
```
dis_stat <- data.frame("label" = phenotype$geo_accession,
                       "disease status" =phenotype$`disease status:ch1`)
```

Para generar el dendrograma utilizo diferentes funciones del paquete ggdendro.

```
# Preparación de los datos para poder representarlos con ggplot2
expression_norm <- t(exprs(data_norm))
hc <- hclust(dist(expression_norm))
dendr <- dendro_data(hc)
dendr[['labels']] <- merge(dendr[['labels']], dis_stat, by='label')

# Dendrograma
ggplot() +
  geom_segment(data=segment(dendr),
               aes(x=x, y=y, xend=xend, yend=yend)) +
  theme_void()+
  geom_text(data=label(dendr),
            aes(x, y, label=label, hjust=0, color=dis_stat$disease.status),
            size=5) +
  scale_color_viridis_d()+
  coord_flip() +
  scale_y_reverse(expand=c(0.2, 0)) +
  labs(title = "Dendrogram",
       color = "Disease status") +
  theme(legend.position = "bottom")
```

Dendrogram



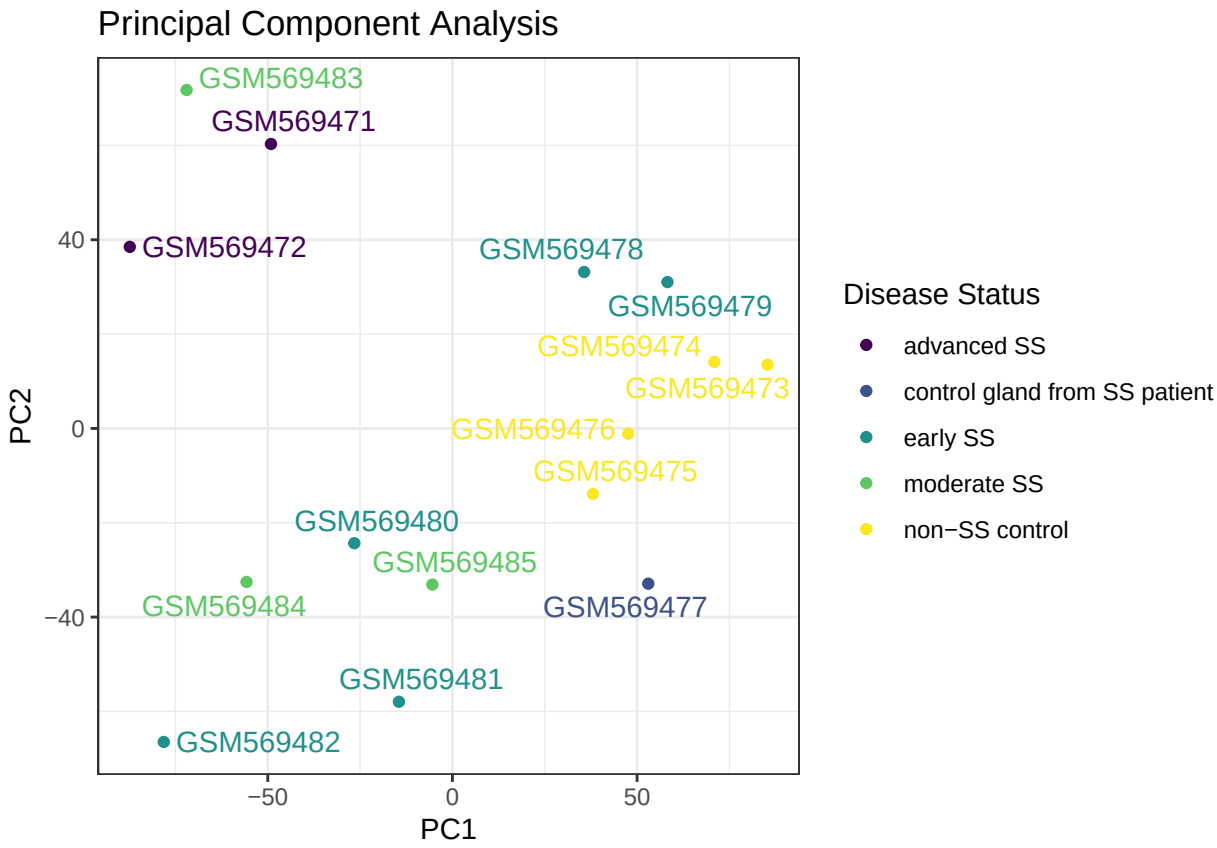

```
ggsave(filename = file.path(out_folder, "GSE23117_dendrograma.png"),
       last_plot(), width = 9, height = 7)
```

8. Realizar un análisis de componentes principales

Para hacer el análisis de componentes principales (PCA), he utilizado la función prcomp del paquete stats.

```
# PCA
pcaResult <- prcomp(expression_norm)

# Plot
pcaResult_df <- as.data.frame(pcaResult$x)
ggplot(data=pcaResult_df, mapping=aes(x=PC1,
                                      y = PC2,
                                      colour = dis_stat$disease.status,
                                      label = dis_stat$label))+
  geom_point()+
  theme_bw()+
  scale_color_viridis_d()+
  geom_text_repel(show.legend = F)+
  labs(title="Principal Component Analysis",
       colour="Disease Status")
```




```
ggsave(filename = file.path(out_folder, "GSE23117_pca.png"), last_plot(), width = 9, height = 7)
```