Exercise Sjogren’s syndrome

Aina Montalban

February 2022

Contents

Introduction

This is a RMarkdown document with the code for the exercise of gene expression profiles of samples of minor salivary glands of patients with *Sjogren’s syndrome* (SS).

The data for this analysis can be found in “https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE23117”. As we can see, the GEO accession number is **GSE23117**. Hence, the main **objectives** of this exercise are:

- Normalizing the data
- Plotting the data:
  - Using a dendogram
  - Using Principal Component Analysis (PCA)

Establishing a working directory

```r
workingDir <- getwd()
data_dir <- file.path(workingDir, "data")
results_dir <- file.path(workingDir, "results")
```

Loading packages

To carry out this exercise, we need to load several packages:

Bioconductor packages

- **oligo**: package to read CEL files
- **Biobase**: package to generate the phenotype data and obtain the expression matrix
- **affy**: package to unify phenotype and CEL data and perform normalization of the data
- **hgu133plus2cdf**: package to have the Affymetrix Affymetrix HG-U133_Plus_2 Array annotation data used in the analysis
- **GEOquery**: package to read the series matrix file

Others

- **ggplot2**: package to build plots
- **ggrepel**: package to repel overlapping text labels in a plot
- **ggdendro**: package to plot dendograms

```r
# load packages
library(oligo)
library(Biobase)
library(gplots)
library(ggplot2)
library(ggrepel)
library(affy)
library(GEOquery)
library(hgu133plus2cdf)
library(ggdendro)
```

Loading data

To continue with our analysis, we will import the **.CEL files** and create an annotated dataframe of our **phenotypic data**. We have two options to do that: (1) download the data directly from the GEO database or (2) use the package *GEOquery* and download the data using a function. In this case, we will choose the first option, that is downloading the raw data directly from the database. The files are found in the *data/* folder.

CEL files

To read the .CEL files, we need to specify the path of the files:

```
CELfiles_fullName <- list.celfiles(file.path(data_dir), full.names=TRUE) # save the full path
```

Once we have obtain the list of the names of the .CEL files, we can read them using the function *read.celfiles()* from the *oligo* package:

```
raw_data <- read.celfiles(CELfiles_fullName)
```

We change the sample names to obtain the names without the file extension .CEL

```
name_sample <- sampleNames(raw_data)
pData(raw_data)$name_sample <- name_sample
sampleNames <- sub(".*_", "", name_sample)
sampleNames <- sub(".CEL.gz$", "", name_sample)
sampleNames(raw_data) <- sampleNames
```

Phenotypic data

The next step is to obtain a dataframe with the phenotypic data. To do that, we are going to read the Series Matrix File.

```
gse <- getGEO(filename = file.path(data_dir, "GSE23117_series_matrix.txt.gz"),
              GSEMatrix = FALSE)
```

From this object, we can retrieve the phenotype data:

```
pheno_data <- pData(phenoData(gse))
```

We can store only some information, such as:

- The *geo_accession* number of each sample
- A *group* variable with 2 levels, with 0 (control) and 1 (SS)
- A *shortName* variable that indicates the stage of the disease

```
targets <- pheno_data[, c("geo_accession", "disease status:ch1")]
targets$group <- c(rep(1, 2), rep(0,4), rep(1,9))
targets$shortName <- c(rep("advanced_SS", 2), rep("control", 4),
                       rep("control_SS", 1), rep("early_SS", 5),
                       rep("moderate_SS", 3))
```

We can write a CSV file with the phenotype data:

```
# write a CSV file
write.csv(targets, file.path(data_dir, "targets.csv"), row.names = FALSE)
```

We build an AnnotatedDataFrame to later create an *AffyBatch* object:

```
columnDesc <-  data.frame(labelDescription= c("geo_accession",
                                              "disease status:ch1", "group",
                                              "shortName"))
myAnnotDF <- new("AnnotatedDataFrame", data=targets, varMetadata= columnDesc)
```

Unifying CEL files and phenotypic data

To normalize the data using the *affy* package, we need to create an **AffyBatch** object.

```
affy_object <- read.affybatch(filenames = CELfiles_fullName,
                              phenoData = myAnnotDF)
show(affy_object)
```

```
## AffyBatch object
## size of arrays=1164x1164 features (27 kb)
## cdf=HG-U133_Plus_2 (54675 affyids)
## number of samples=15
## number of genes=54675
## annotation=hgu133plus2
## notes=
```

Normalizing the data

Next, we will normalize the expression matrix with the function **rma** from the **affy** package. The main steps of this function are:

- Background correcting
- Normalizing
- Calculating expression

```
exp_norm <- affy::rma(affy_object)
```

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

We can select some rows of the normalized expression matrix:

```
head(exprs(exp_norm))[,1:5]
```

```
##             GSM569471 GSM569472 GSM569473 GSM569474 GSM569475
## 1007_s_at    9.851975  9.727738 10.888743 10.402887 10.491031
## 1053_at      6.341534  6.257264  6.720002  6.456307  6.091974
## 117_at       5.906726  5.678346  5.717558  5.629298  5.660079
## 121_at       7.339844  7.559815  7.433192  7.521091  7.327734
## 1255_g_at    3.470013  3.495447  3.679210  3.571284  3.579521
## 1294_at      8.418723  7.943172  8.005757  7.892548  8.087714
```

Plotting normalized data

Dendrogram

We will perform clustering to identify if the samples form groups. Different approaches can yield to different grouping. However, we would expect that the control samples form a group and the SS samples form another group.

We need to decide:

- A distance measure (i.e., how similar are the samples)
- A cluster algorithm (i.e., how to group the samples)

For instance, we can choose the euclidean distance and the average linkage agglomeration.

```
dist_eucl_clust <- hclust(dist(t(exprs(exp_norm))),
                                method="average")
dend <- as.dendrogram(dist_eucl_clust)
```

We rename the sample names with the short name:

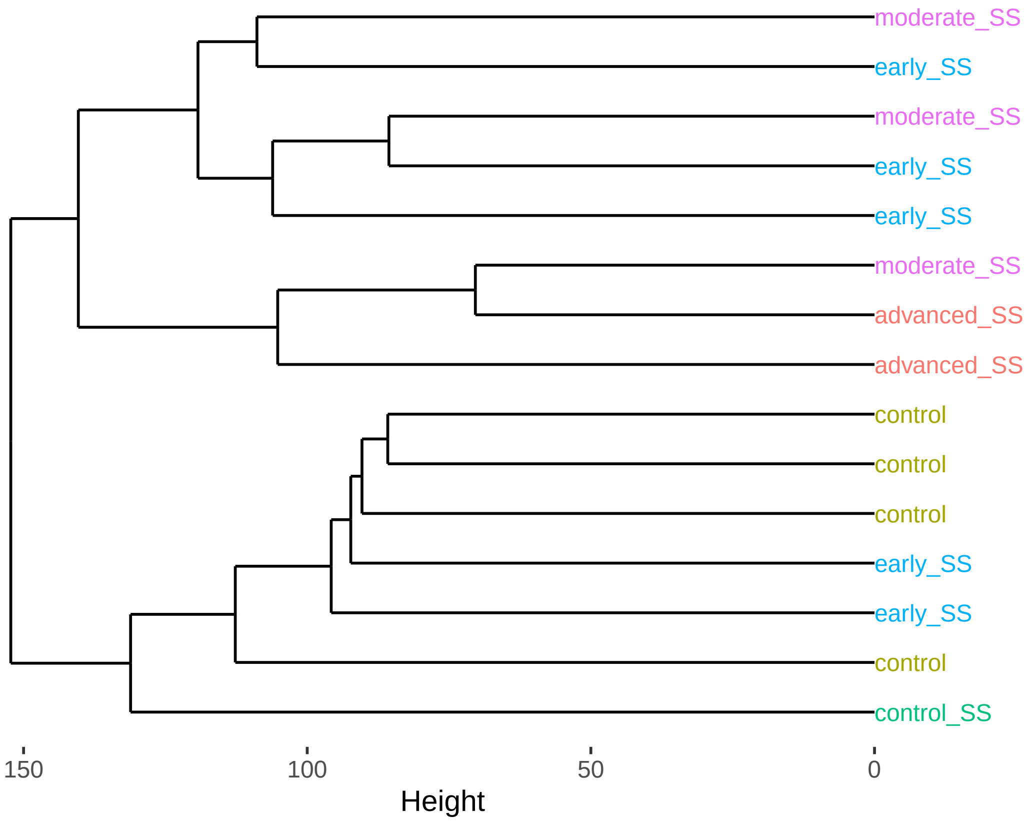
```
dend_data <- dendro_data(dend, type = "rectangle")
lbs <- label(dend_data)$label
lst <- c()
for (ele in lbs){
    gName <- targets[targets$geo_accession==ele,"shortName"]
    lst <- c(lst, gName)
}
dend_data$labels[, "label"] <- lst
```

And finally, we plot the tree:

```
p1 <- ggplot(segment(dend_data)) +
      geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
      coord_flip() +  scale_y_reverse(expand = c(0.2, 0.2))

p1 <- p1 + theme(axis.line.y=element_blank(), axis.ticks.y=element_blank(),
     axis.text.y=element_blank(), axis.title.y=element_blank(),
     panel.background=element_rect(fill="white")) + labs(y="Height")

p1 <- p1 + geom_text(data = label(dend_data),
              aes(x = x, y = y, label = label, color=label),
              size = 3, vjust=0.5, hjust=0, show.legend = FALSE)
p1
```



In the dendrogram above, we can see clearly two groups. One group is formed by all the control samples, two early SS and one control SS. This makes sense as the early SS samples might be more similar to the control samples than an advanced sample. On the other hand, the other cluster is formed by the moderate and advanced SS samples as well as some early samples.

Principal component analysis

For the PCA, we define a variable with the transposed expression matrix (i.e., we need the genes as columns and samples as rows):

```
X <- t(exprs(exp_norm))
head(X)[1:4, 1:5]
```

```
##             1007_s_at  1053_at   117_at   121_at 1255_g_at
## GSM569471  9.851975 6.341534 5.906726 7.339844  3.470013
## GSM569472  9.727738 6.257264 5.678346 7.559815  3.495447
## GSM569473 10.888743 6.720002 5.717558 7.433192  3.679210
## GSM569474 10.402887 6.456307 5.629298 7.521091  3.571284
```

We compute the PCA using the *prcomp* function:

```
pca <- prcomp(X, scale. = FALSE)
Groups <- as.factor(targets$group)
shortName <- targets$shortName
```

We calculate the loads and create a dataframe:

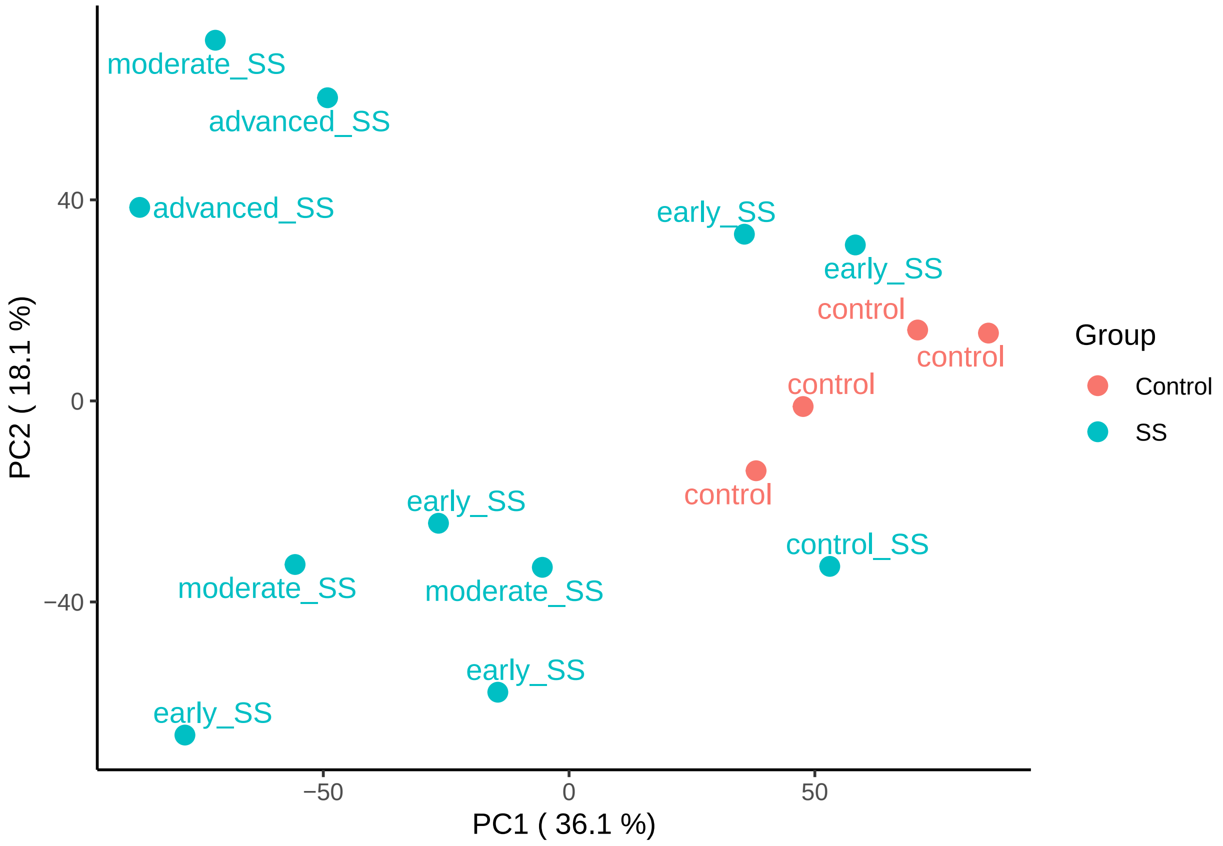
```
loads <- round(pca$sdev^2/sum(pca$sdev^2)*100,1)
pca_df <- data.frame(pca$x)
```

Plot the PCA:

```
p2 <- ggplot(pca_df, aes(x=PC1, y=PC2, color=Groups, label=shortName)) +
       geom_point(aes(color=Groups), size=3) +
        scale_colour_discrete(name="Group", labels=c("Control", "SS"))

p2 <- p2 + theme_classic() + labs(x=c(paste("PC1 (",loads[1],"%)")),
                                  y=c(paste("PC2 (",loads[2],"%)")))

p2 <- p2 + geom_text_repel(show.legend = FALSE)
p2
```



As we have seen in the dendrogram, the control samples are closer as well as the early and control SS samples, whereas the advanced and moderate SS samples are farther from the control samples.