Anàlisi de Components Principals

Paula Solé Vallés

2024-12-03

Obtenció de les dades

```
# Set working directory
setwd('/Volumes/ftp/Paula Sole/processed_data')

# Upload files
PA_log <- read.table('PA_dataframe_log.txt', sep="\t")
AR_log <- read.table('AR_dataframe_log.txt', sep="\t")

Luk_dataframe <- read.table('Luk_dataframe.txt', sep="\t")
Not_dataframe <- read.table('Not_dataframe.txt', sep="\t")
```

PCA segons teixit

PA

Treballem amb les dades *normalitzades* de PA tant de Lukullus com Notabillis i dels 3 tipus d'anàlisi (paràmetres fisiològics, transcriptòmica i metabolòmica). En els conjunts de dades amb que treballem les variables es disposen en columnes i les mostres, en files.

Modifiquem el dataframe de manera adequada per a realitzar el PCA.

```
# Save group names
groups_PA <- row.names(PA_log)
print(groups_PA)
```

```
##  [1] "LUK_C_PA"   "LUK_C_PA.1" "LUK_C_PA.2" "LUK_S_PA"   "LUK_S_PA.1"
##  [6] "LUK_S_PA.2" "NOT_C_PA"   "NOT_C_PA.1" "NOT_C_PA.2" "NOT_S_PA"
## [11] "NOT_S_PA.1" "NOT_S_PA.2"
```

```
# Create uniform names for the samples
groups_PA <- gsub("(LUK|NOT)_(C|S).*", "\\1_\\2", groups_PA)
print(groups_PA)
```

```
##  [1] "LUK_C" "LUK_C" "LUK_C" "LUK_S" "LUK_S" "LUK_S" "NOT_C" "NOT_C"
"NOT_C"
## [10] "NOT_S" "NOT_S" "NOT_S"
```

```
# Convert the dataframe to numeric values
PA_log <- apply(PA_log, 2, as.numeric)
row.names(PA_log) <- groups_PA
```

```r
# Delete columns with NA
columns_NA <- colSums(is.na(PA_log)) > 0
index_columns_NA <- which(columns_NA)
print(index_columns_NA)
```

```
## LWP
##   3
```

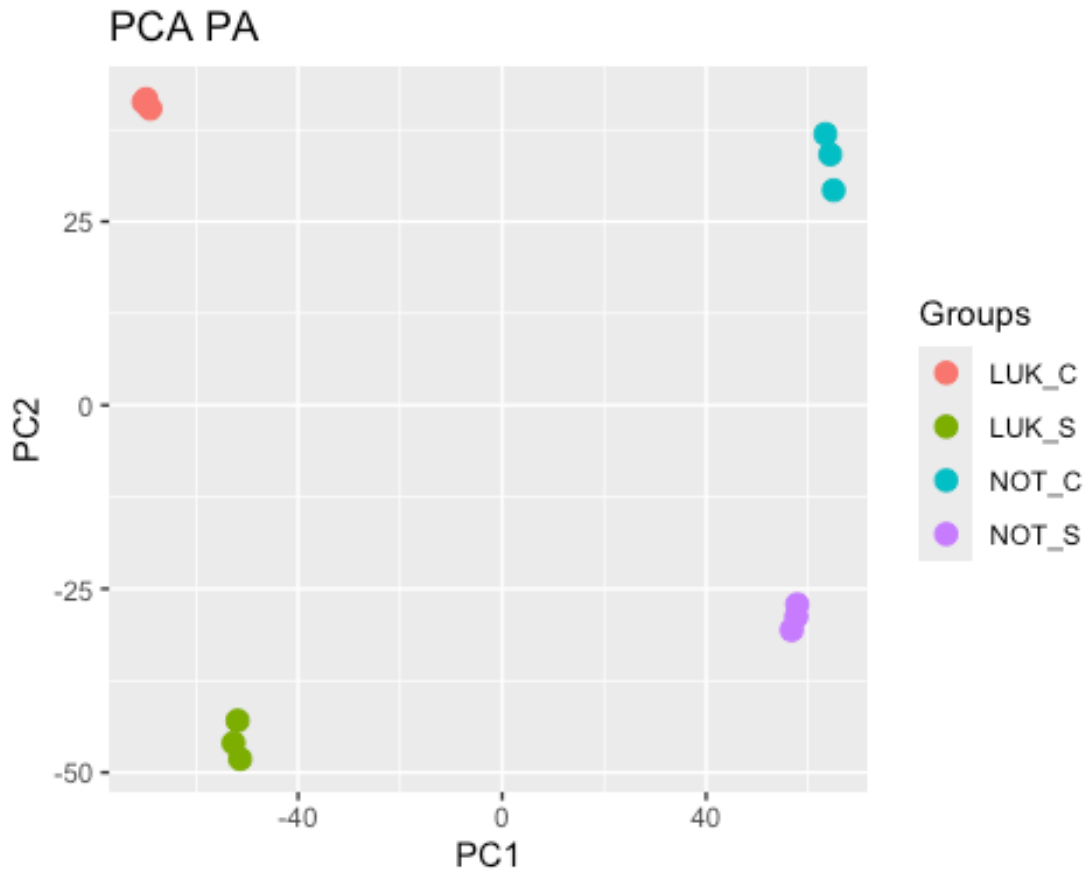
```r
PA_log <- PA_log[, -index_columns_NA]
```

Realitzem el PCA i visualitzem el resultat.

```r
# Filter the constant columns
PA_log <- PA_log[, apply(PA_log, 2, var) != 0]

# Perform the PCA
PCA_PA <- prcomp(PA_log, center = TRUE, scale. = TRUE)

# Summary of the result
summary(PCA_PA)
```

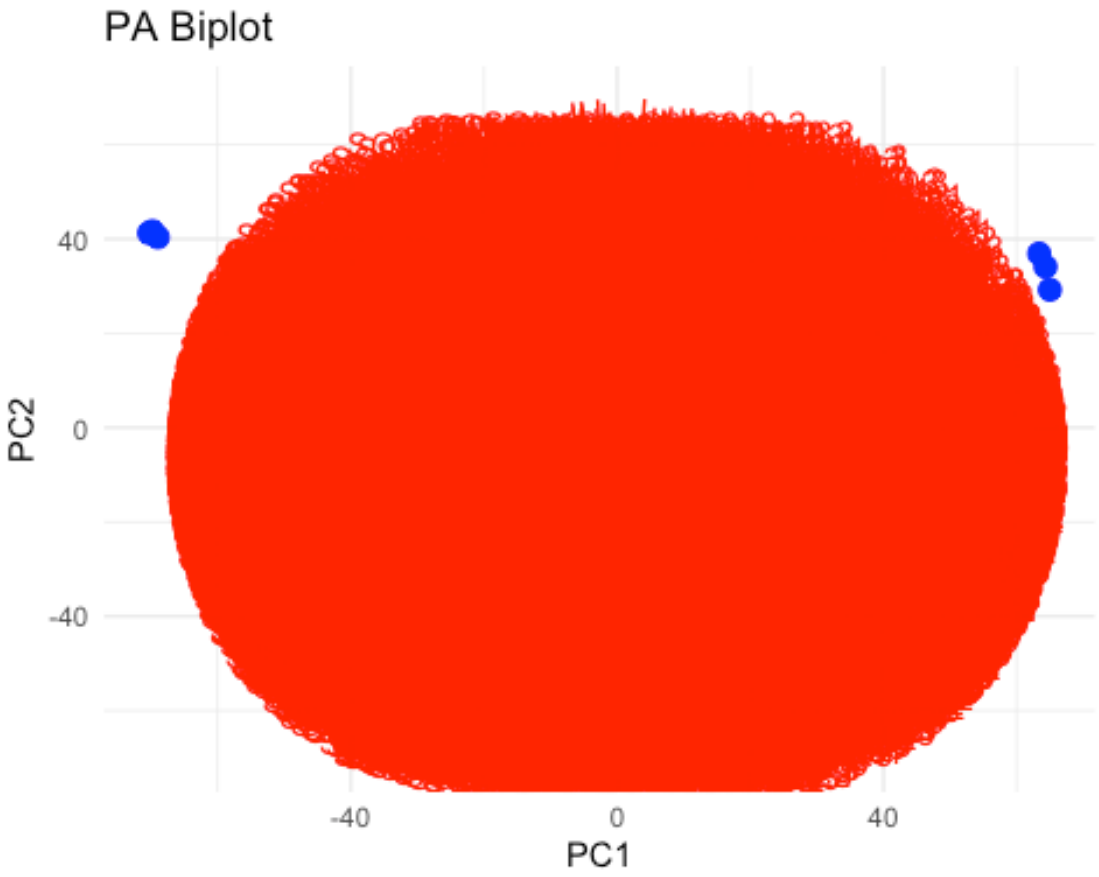
```
## Importance of components:
##                             PC1     PC2     PC3      PC4      PC5      PC6
## Standard deviation     63.9249 39.5781 30.8178 18.60848 18.22630 17.30631
## Proportion of Variance  0.4623  0.1772  0.1074  0.03917  0.03758  0.03388
## Cumulative Proportion   0.4623  0.6395  0.7469  0.78607  0.82365  0.85753
##                             PC7      PC8      PC9     PC10    PC11
PC12
## Standard deviation     16.56592 16.27832 15.73961 15.45741 15.2769 3.796e-
13
## Proportion of Variance  0.03104  0.02998  0.02802  0.02703  0.0264
0.000e+00
## Cumulative Proportion   0.88857  0.91855  0.94657  0.97360  1.0000
1.000e+00
```

```r
# Dataframe with the PCA result and the groups variable
PCA_PA_df <- data.frame(PC1 = PCA_PA$x[,1], PC2 = PCA_PA$x[,2], Grupo =
groups_PA)

# Graphic
ggplot(PCA_PA_df, aes(x = PC1, y = PC2, color = groups_PA)) +
  geom_point(size = 3) +
  labs(title = "PCA PA", x = "PC1", y = "PC2", color = "Groups")
```



Realitzem d'un biplot:

```
scores <- as.data.frame(PCA_PA$x)
loadings <- as.data.frame(PCA_PA$rotation)

# Scale the loadings so that they are displayed correctly in the biplot
scale_factor <- max(abs(scores$PC1), abs(scores$PC2)) /
max(abs(loadings$PC1), abs(loadings$PC2))
loadings <- loadings * scale_factor

# Create the biplot with ggplot2
ggplot() +
  geom_point(data = scores, aes(x = PC1, y = PC2), color = "blue", size = 3)
+  # Samples
  geom_segment(data = loadings, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow = arrow(length = unit(0., "cm")), color = "red") +  #
Variables arrows
   geom_text(data = loadings, aes(x = PC1, y = PC2, label =
rownames(loadings)),
          color = "red", vjust = 1.5) +  # variables Labels
  labs(title = "PA Biplot", x = "PC1", y = "PC2") +
  theme_minimal()
```



AR

Treballem amb les dades *normalitzades* de AR tant de Lukullus com Notabillis i dels 3 tipus d'anàlisi (paràmetres fisiològics, transcriptòmica i metabolòmica). En els conjunts de dades amb que treballem les variables es disposen en columnes i les mostres, en files.

Modifiquem el dataframe de manera adequada per a realitzar el PCA.

```
# Save group names
groups_AR <- row.names(AR_log)
print(groups_AR)
```

```
##  [1] "LUK_C_AR"   "LUK_C_AR.1" "LUK_C_AR.2" "LUK_S_AR"   "LUK_S_AR.1"
##  [6] "LUK_S_AR.2" "NOT_C_AR"   "NOT_C_AR.1" "NOT_C_AR.2" "NOT_S_AR"
## [11] "NOT_S_AR.1" "NOT_S_AR.2"
```

```
groups_AR <- gsub("(LUK|NOT)_(C|S).*", "\\1_\\2", groups_AR)
print(groups_AR)
```

```
##  [1] "LUK_C" "LUK_C" "LUK_C" "LUK_S" "LUK_S" "LUK_S" "NOT_C" "NOT_C"
"NOT_C"
## [10] "NOT_S" "NOT_S" "NOT_S"
```

```r
# Convert the dataframe to numeric values
AR_log <- apply(AR_log, 2, as.numeric)
row.names(AR_log) <- groups_AR

# Delete columns with NA
columns_NA <- colSums(is.na(AR_log)) > 0
index_columns_NA <- which(columns_NA)
print(index_columns_NA)
```

```
## LWP
##   3
```

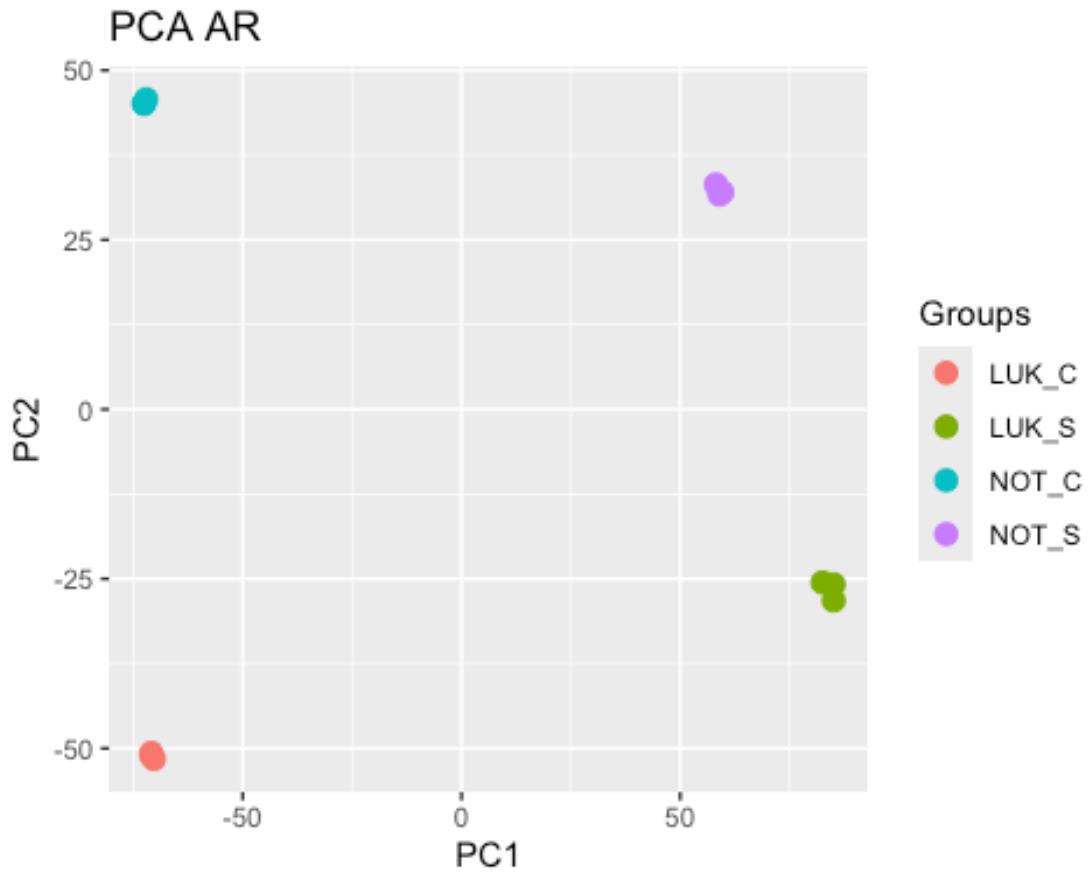
```r
AR_log <- AR_log[, -index_columns_NA]
```

Realitzem el PCA i visualitzem el resultat.

```r
# Filter the constant columns
AR_log <- AR_log[, apply(AR_log, 2, var) != 0]

# Perform the PCA
PCA_AR <- prcomp(AR_log, center = TRUE, scale. = TRUE)

# Summary of the result
summary(PCA_AR)
```

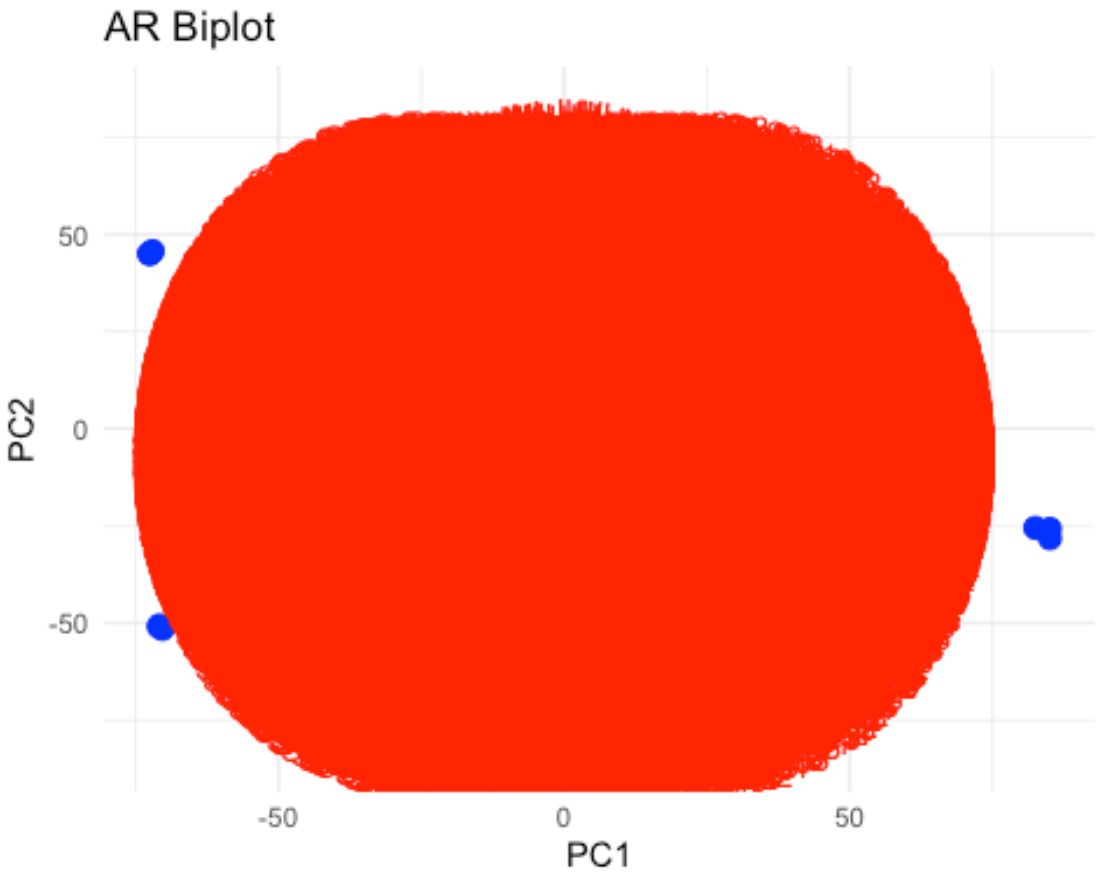
```
## Importance of components:
##                             PC1     PC2     PC3      PC4      PC5      PC6
## Standard deviation     75.3222 41.8160 31.1531 12.59951 12.01336 11.82610
## Proportion of Variance  0.6028  0.1858  0.1031  0.01687  0.01533  0.01486
## Cumulative Proportion   0.6028  0.7886  0.8917  0.90855  0.92388  0.93874
##                             PC7      PC8     PC9     PC10     PC11
PC12
## Standard deviation     11.74753 10.95203 10.4946 10.27259 10.14528 1.065e-
13
## Proportion of Variance  0.01466  0.01274  0.0117  0.01121  0.01094
0.000e+00
## Cumulative Proportion   0.95341  0.96615  0.9778  0.98906  1.00000
1.000e+00
```

```r
# Dataframe with the PCA result and the groups variable
PCA_AR_df <- data.frame(PC1 = PCA_AR$x[,1], PC2 = PCA_AR$x[,2], Grupo =
groups_AR)

# Graphic
ggplot(PCA_AR_df, aes(x = PC1, y = PC2, color = groups_AR)) +
  geom_point(size = 3) +
  labs(title = "PCA AR", x = "PC1", y = "PC2", color = "Groups")
```



Realitzem d'un biplot:

```
scores <- as.data.frame(PCA_AR$x)
loadings <- as.data.frame(PCA_AR$rotation)

# Scale the loadings so that they are displayed correctly in the biplot
scale_factor <- max(abs(scores$PC1), abs(scores$PC2)) /
max(abs(loadings$PC1), abs(loadings$PC2))
loadings <- loadings * scale_factor

# Create the biplot with ggplot2
ggplot() +
  geom_point(data = scores, aes(x = PC1, y = PC2), color = "blue", size = 3)
+  # Samples
  geom_segment(data = loadings, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow = arrow(length = unit(0., "cm")), color = "red") +  #
Variables arrows
   geom_text(data = loadings, aes(x = PC1, y = PC2, label =
rownames(loadings)),
          color = "red", vjust = 1.5) +  # variables Labels
  labs(title = "AR Biplot", x = "PC1", y = "PC2") +
  theme_minimal()
```



Segons genotip

Lukullus

Treballem amb les dades de *Lukullus* tant de PA com AR i dels 3 tipus d'anàlisi (paràmetres fisiològics, transcriptòmica i metabolòmica). En els conjunts de dades amb que treballem les variables es disposen en columnes i les mostres, en files.

Modifiquem el dataframe de manera adequada per a realitzar el PCA.

```
# Save group names
groups_LUK <- row.names(Luk_dataframe)
print(groups_LUK)
```

```
##  [1] "LUK_C_AR"   "LUK_C_AR.1" "LUK_C_AR.2" "LUK_S_AR"   "LUK_S_AR.1"
##  [6] "LUK_S_AR.2" "LUK_C_PA"   "LUK_C_PA.1" "LUK_C_PA.2" "LUK_S_PA"
## [11] "LUK_S_PA.1" "LUK_S_PA.2"
```

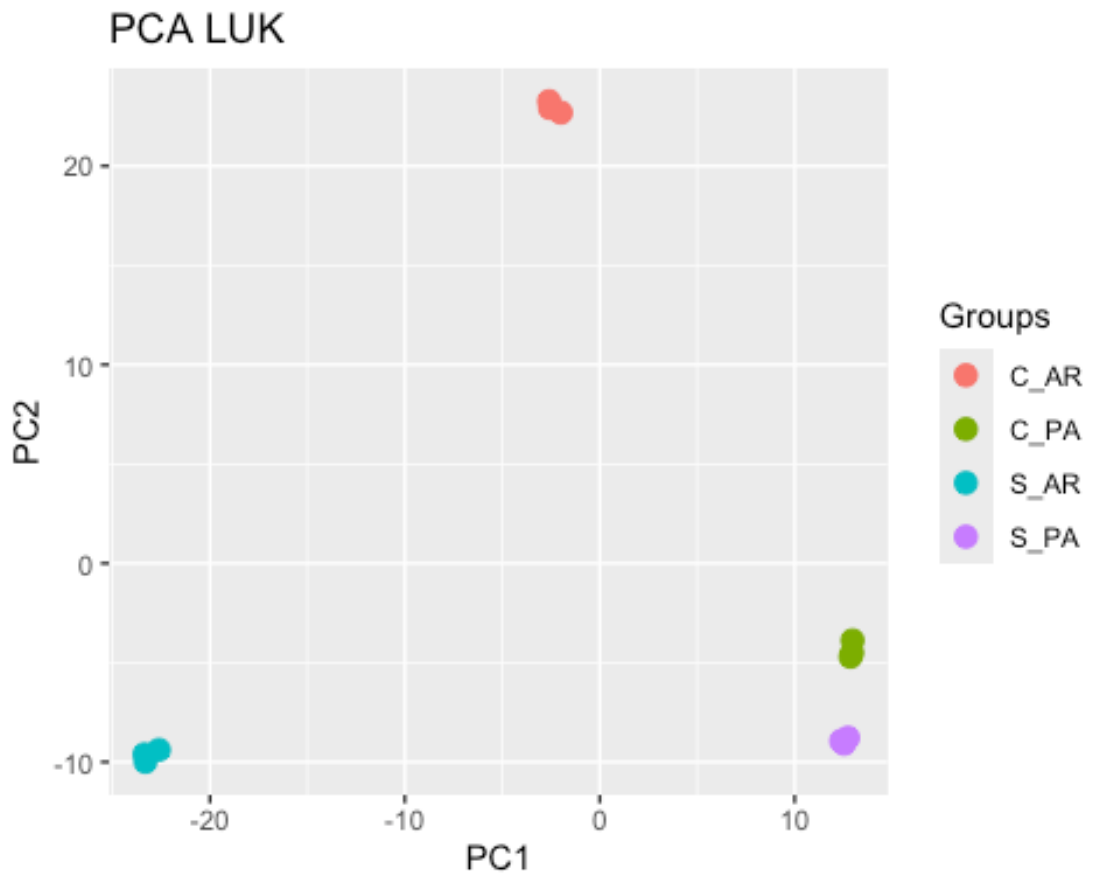
```
groups_LUK <- gsub(".*(C|S)_(AR|PA).*", "\\1_\\2", groups_LUK)
print(groups_LUK)
```

```
##  [1] "C_AR" "C_AR" "C_AR" "S_AR" "S_AR" "S_AR" "C_PA" "C_PA" "C_PA" "S_PA"
## [11] "S_PA" "S_PA"

# Convert the dataframe to numeric values
Luk_dataframe <- apply(Luk_dataframe, 2, as.numeric)
row.names(Luk_dataframe) <- groups_LUK

# Delete columns with NA
columns_NA <- colSums(is.na(Luk_dataframe)) > 0
index_columns_NA <- which(columns_NA)
print(index_columns_NA)

## LWP
##   3

Luk_dataframe <- Luk_dataframe[, -index_columns_NA]
```

Realitzem el PCA i visualitzem el resultat.

```
# Filter the constant columns
Luk_dataframe <- Luk_dataframe[, apply(Luk_dataframe, 2, var) != 0]
# Scale the data
Luk_df_scaled <- scale(Luk_dataframe)

# Perform the PCA
PCA_LUK <- prcomp(Luk_df_scaled, center = TRUE, scale. = TRUE)

# Summary of the result
summary(PCA_LUK)

## Importance of components:
##                           PC1     PC2    PC3     PC4     PC5     PC6
PC7
## Standard deviation     15.3490 14.0068 6.2195 2.71482 2.44256 2.32447
2.19263
## Proportion of Variance  0.4628  0.3854 0.0760 0.01448 0.01172 0.01062
0.00945
## Cumulative Proportion   0.4628  0.8483 0.9243 0.93877 0.95049 0.96111
0.97055
##                           PC8     PC9    PC10    PC11      PC12
## Standard deviation     2.09385 2.02818 1.96519 1.62146 7.988e-15
## Proportion of Variance 0.00861 0.00808 0.00759 0.00517 0.000e+00
## Cumulative Proportion  0.97917 0.98725 0.99483 1.00000 1.000e+00

# Dataframe with the PCA result and the groups variable
PCA_LUK_df <- data.frame(PC1 = PCA_LUK$x[,1], PC2 = PCA_LUK$x[,2], Grupo =
groups_LUK)

# Graphic
ggplot(PCA_LUK_df, aes(x = PC1, y = PC2, color = groups_LUK)) +
```

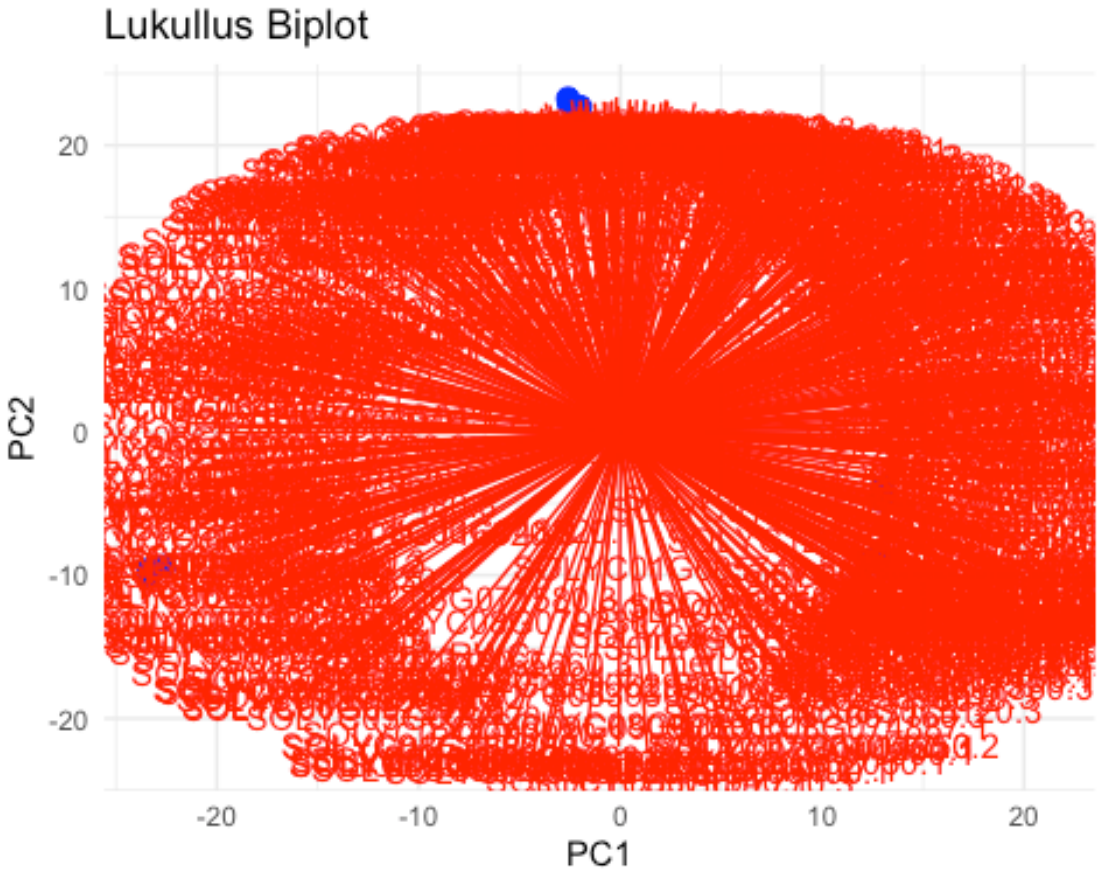
```
  geom_point(size = 3) +
  labs(title = "PCA LUK", x = "PC1", y = "PC2", color = "Groups")
```



Realització d'un biplot:

```
scores <- as.data.frame(PCA_LUK$x)
loadings <- as.data.frame(PCA_LUK$rotation)

# Scale the loadings so that they are displayed correctly in the biplot
scale_factor <- max(abs(scores$PC1), abs(scores$PC2)) /
max(abs(loadings$PC1), abs(loadings$PC2))
loadings <- loadings * scale_factor

# Create the biplot with ggplot2
ggplot() +
  geom_point(data = scores, aes(x = PC1, y = PC2), color = "blue", size = 3)
+  # Samples
  geom_segment(data = loadings, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow = arrow(length = unit(0., "cm")), color = "red") +  #
Variables arrows
   geom_text(data = loadings, aes(x = PC1, y = PC2, label =
rownames(loadings)),
          color = "red", vjust = 1.5) +  # variables Labels
```

```
  labs(title = "Lukullus Biplot", x = "PC1", y = "PC2") +
  theme_minimal()
```



Notabilis

Treballem amb les dades de *notabilis* tant de PA com AR i dels 3 tipus d'anàlisi (paràmetres fisiològics, transcriptòmica i metabolòmica). En els conjunts de dades amb que treballem les variables es disposen en columnes i les mostres, en files.

Modifiquem el dataframe de manera adequada per a realitzar el PCA.

```
# Save group names
groups_NOT <- row.names(Not_dataframe)
print(groups_NOT)
```

```
##  [1] "NOT_C_AR"   "NOT_C_AR.1" "NOT_C_AR.2" "NOT_S_AR"   "NOT_S_AR.1"
##  [6] "NOT_S_AR.2" "NOT_C_PA"   "NOT_C_PA.1" "NOT_C_PA.2" "NOT_S_PA"
## [11] "NOT_S_PA.1" "NOT_S_PA.2"
```

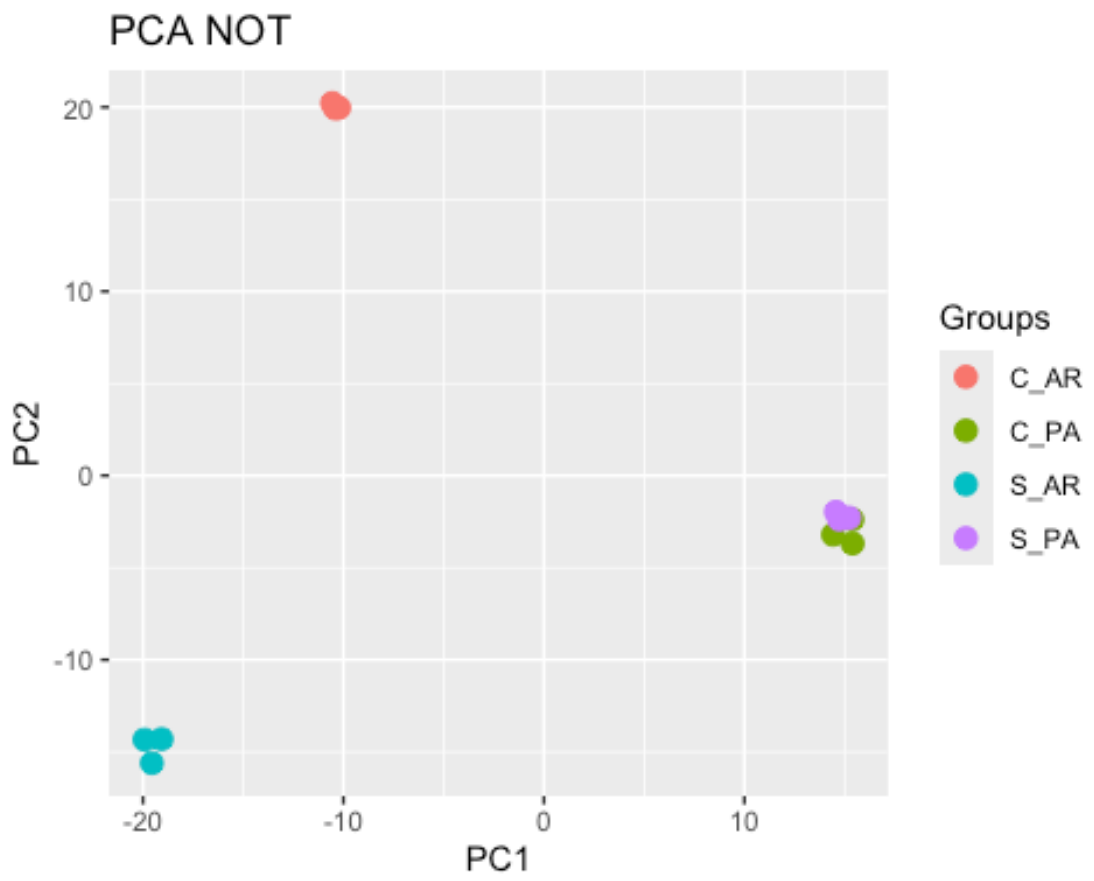
```
groups_NOT <- gsub(".*(C|S)_(AR|PA).*", "\\1_\\2", groups_NOT)
print(groups_NOT)
```

```
##  [1] "C_AR" "C_AR" "C_AR" "S_AR" "S_AR" "S_AR" "C_PA" "C_PA" "C_PA" "S_PA"
## [11] "S_PA" "S_PA"
```

```r
# Convert the dataframe to numeric values
Not_dataframe <- apply(Not_dataframe, 2, as.numeric)
row.names (Not_dataframe) <- groups_NOT

# Delete columns with NA
columns_NA <- colSums(is.na(Not_dataframe)) > 0
index_columns_NA <- which(columns_NA)
print(index_columns_NA)

## LWP
##   3

Not_dataframe <- Not_dataframe[, -index_columns_NA]
```

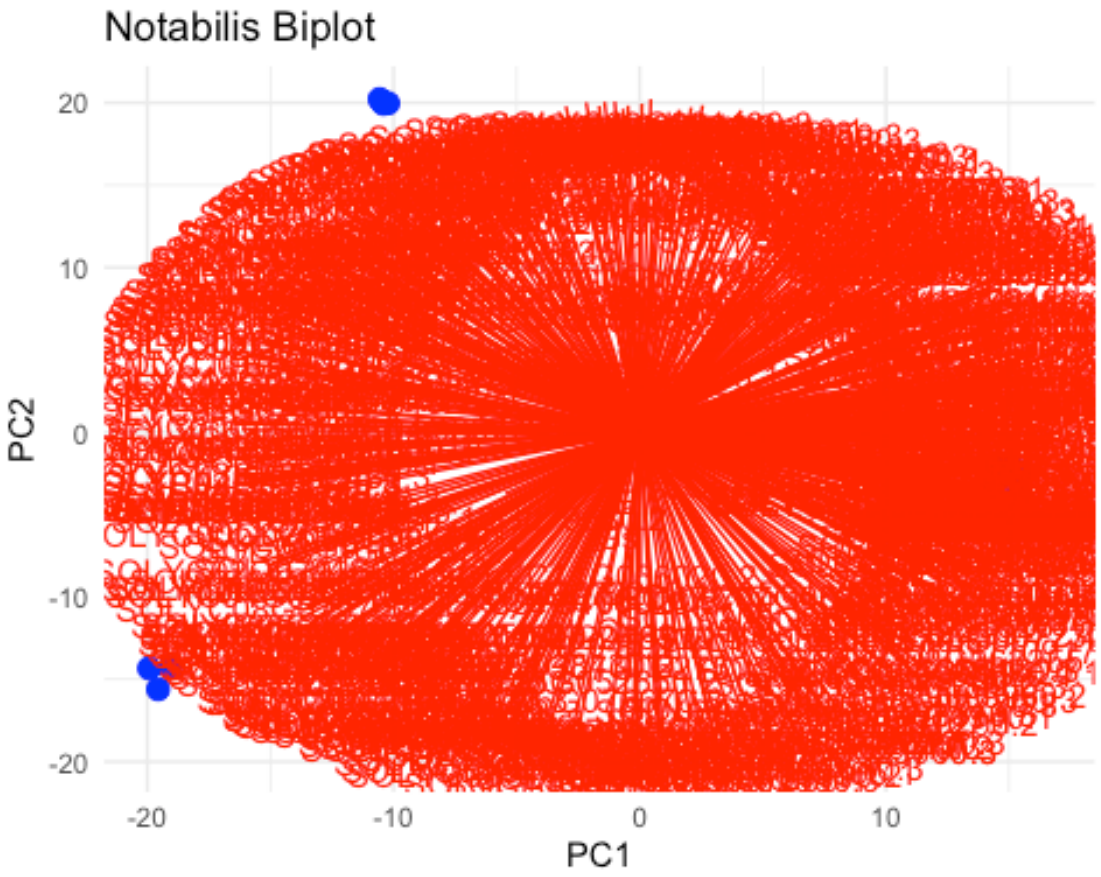
Realitzem el PCA i visualitzem el resultat.

```r
# Filter the constant columns
Not_dataframe <- Not_dataframe[, apply(Not_dataframe, 2, var) != 0]
# Scale the data
Not_df_scaled <- scale(Not_dataframe)

# Perform the PCA
PCA_NOT <- prcomp(Not_df_scaled, center = TRUE, scale. = TRUE)

# Summary of the result
summary(PCA_NOT)

## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
PC7
## Standard deviation     15.9855 13.1539 6.31365 3.22866 2.94899 2.82707
2.53858
## Proportion of Variance  0.4895  0.3315 0.07636 0.01997 0.01666 0.01531
0.01235
## Cumulative Proportion   0.4895  0.8210 0.89737 0.91734 0.93400 0.94931
0.96165
##                           PC8     PC9    PC10    PC11      PC12
## Standard deviation     2.42461 2.37821 2.18796 1.92234 6.441e-15
## Proportion of Variance 0.01126 0.01084 0.00917 0.00708 0.000e+00
## Cumulative Proportion  0.97291 0.98375 0.99292 1.00000 1.000e+00

# Dataframe with the PCA result and the groups variable
PCA_NOT_df <- data.frame(PC1 = PCA_NOT$x[,1], PC2 = PCA_NOT$x[,2], Grupo =
groups_NOT)

# Graphic
ggplot(PCA_NOT_df, aes(x = PC1, y = PC2, color = groups_NOT)) +
  geom_point(size = 3) +
  labs(title = "PCA NOT", x = "PC1", y = "PC2", color = "Groups")
```



Realització d'un biplot:

```
scores <- as.data.frame(PCA_NOT$x)
loadings <- as.data.frame(PCA_NOT$rotation)

# Scale the loadings so that they are displayed correctly in the biplot
scale_factor <- max(abs(scores$PC1), abs(scores$PC2)) /
max(abs(loadings$PC1), abs(loadings$PC2))
loadings <- loadings * scale_factor

# Create the biplot with ggplot2
ggplot() +
  geom_point(data = scores, aes(x = PC1, y = PC2), color = "blue", size = 3)
+  # Samples
  geom_segment(data = loadings, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow = arrow(length = unit(0., "cm")), color = "red") +  #
Variables arrows
   geom_text(data = loadings, aes(x = PC1, y = PC2, label =
rownames(loadings)),
          color = "red", vjust = 1.5) +  # variables Labels
  labs(title = "Notabilis Biplot", x = "PC1", y = "PC2") +
  theme_minimal()
```



Exportació dels resultats

Guardem els gràfics en format png:

```
png(file="~/Desktop/PCA/PCA2_plots.png", width=600, height=200)

par(mfrow=c(2,2))

plot1<- ggplot(PCA_PA_df, aes(x = PC1, y = PC2, color = groups_PA)) +
  geom_point(size = 3) +
  labs(title = "PCA PA", x = "PC1", y = "PC2", color = "Groups")

plot2 <- ggplot(PCA_AR_df, aes(x = PC1, y = PC2, color = groups_AR)) +
  geom_point(size = 3) +
  labs(title = "PCA AR", x = "PC1", y = "PC2", color = "Groups")

plot3 <- ggplot(PCA_LUK_df, aes(x = PC1, y = PC2, color = groups_LUK)) +
  geom_point(size = 3) +
  labs(title = "PCA LUK", x = "PC1", y = "PC2", color = "Groups")

plot4 <- ggplot(PCA_NOT_df, aes(x = PC1, y = PC2, color = groups_NOT)) +
```

```
  geom_point(size = 3) +
  labs(title = "PCA NOT", x = "PC1", y = "PC2", color = "Groups")

grid.arrange(plot1, plot2, plot3, plot4, ncol = 2, nrow = 2)

dev.off()

## quartz_off_screen
##                 2
```