PAC1

Joan Mora Peralta

2024-10-25

```
system("git clone https://github.com/nutrimetabolomics/metaboData.git")
```

Aquí hem descarregat l'estat actual del repositori Git.

```
assayData <- read.csv("metaboData/Datasets/2018-MetabotypingPaper/DataValues_S013.csv", header=TRUE, sep
phenoData <- read.csv("metaboData/Datasets/2018-MetabotypingPaper/DataInfo_S013.csv", header=TRUE, sep =
experiment <- SummarizedExperiment(assays = list(assayData = assayData), colData = phenoData)
```

Aquí hem accedit a la carpeta descarregada i transformat els arxius .csv en dues variables que passen a formar part del nou objecte 'experiment' de la classe SummarizedExperiment.

```
experiment
```

```
## class: SummarizedExperiment
## dim: 39 695
## metadata(0):
## assays(1): assayData
## rownames(39): 1 2 ... 38 39
## rowData names(0):
## colnames(695): SUBJECTS SURGERY ... SM.C24.0_T5 SM.C24.1_T5
## colData names(3): VarName varTpe Description
```

```
colData(experiment)
```

```
## DataFrame with 695 rows and 3 columns
##                  VarName    varTpe Description
##                 <factor>  <factor>    <factor>
## SUBJECTS       SUBJECTS integer      dataDesc
## SURGERY        SURGERY  character    dataDesc
## AGE            AGE      integer      dataDesc
## GENDER         GENDER   character    dataDesc
## Group          Group    integer      dataDesc
## ...                ...       ...         ...
## SM.C18.0_T5 SM.C18.0_T5   numeric    dataDesc
## SM.C18.1_T5 SM.C18.1_T5   numeric    dataDesc
## SM.C20.2_T5 SM.C20.2_T5   numeric    dataDesc
## SM.C24.0_T5 SM.C24.0_T5   numeric    dataDesc
## SM.C24.1_T5 SM.C24.1_T5   numeric    dataDesc
```

```r
rowData(experiment)
```

```
## DataFrame with 39 rows and 0 columns
```

En aquest primer anàlisi veiem que es tracta d'un dataset d'e l'experiment d'un experiment amb dimensions: 39x695 (observacions i variables). Els subjectes estudiats estàn numertars i les variables son de diferent naturalesa ("SURGERY" , "SM.C24.0_T5" ...). Per tant, les primeres variables s'espera que siguin més categòriques i més generals i després signifiqin valors en diferents registres metabolòmics. Això concorda amb la descripció de l'experiment escrit en el catàleg del repositori, que diu : Data used in the paper "Metabotypes of response to bariatric surgery independent of the magnitude of weight loss". Les descripcions no estàn ben anotades, ja que en totes hi tenim el mateix valor "dataDesc".

També veiem com rowData no presenta més columnes d'informació, ja que no hem carregat cap anotació al instanciar l'objecte SummarizedExperiment.

```r
rowData(experiment) <- data.frame(anotacions_ = c(rep("info de fila", 39)))
```

Ja que no tenim anotacions de les files, les posem de manera artificial a mode d'exemple.

```r
assay(experiment)[,1:2] <- apply(assay(experiment)[,1:2], 2, as.factor)
assay(experiment)[,4:9] <- apply(assay(experiment)[,4:9], 2, as.factor)
```

Transformem el tipus de certes columnes a factors, ja que d'altra manera R les interpretarà com a numèriques.

```r
na_total <- sum(is.na(assay(experiment)))
na_prop <- na_total / prod(dim(experiment))
cat("Proporció de NA en el dataset: ",round(na_prop*100, 0),"%")
```

```
## Proporció de NA en el dataset:  13 %
```

```r
na_cols <- as.matrix(apply(assay(experiment), 2, function(x) sum(is.na(x))>0))
na_cols_total <- sum(na_cols)
na_cols_prop <- na_cols_total / dim(assay(experiment))[2]
cat("Proporció columnes amb NA en el dataset: ",round(na_cols_prop*100, 0),"%")
```

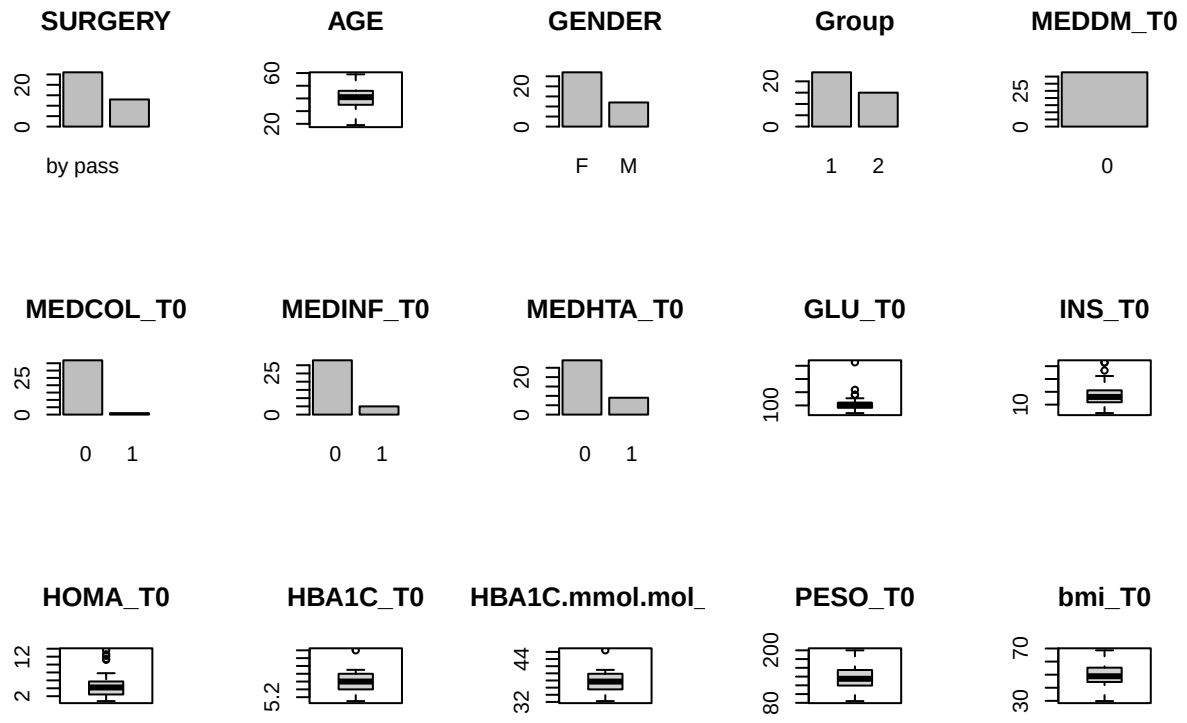
```
## Proporció columnes amb NA en el dataset:  78 %
```

```r
na_fila <- as.matrix(apply(assay(experiment), 1, function(x) sum(is.na(x))>0))
na_fila_total <- sum(na_fila)
na_fila_prop <- na_fila_total / dim(assay(experiment))[1]
cat("Proporció files amb NA en el dataset: ",round(na_fila_prop*100, 0),"%")
```

```
## Proporció files amb NA en el dataset:  100 %
```

Veiem que hi ha molta presència de valors nuls, i que aquests no estan concentrats en cap regió, sino que totes les observacions i gran part de les columnes en tenen. En un cas pràctic crec que no es pot eliminar cap columna s'hauria d'imputar valors que afectin mínimament a la resta de valors com ara la mitjana en columnes quantitatives o medianes en factors. Però primer s'hauria d'interpretar molt bé què significa cada una.

```r
tipus <- sapply(assay(experiment), class)

getPlot <- function(i) {
    if (tipus[i] == "integer" || tipus[i] == "numeric") {
        boxplot(as.numeric(assay(experiment)[, i]), main = colnames(assay(experiment))[i])
    } else {
        barplot(table(assay(experiment)[, i]), main = colnames(assay(experiment))[i])
    }
}
par(mfrow = c(3, 5))
invisible(lapply(2:16, getPlot))
```




```r
invisible(dev.off())
```

Aquí es pot comprovar que hi ha una important disparitat entre els dos tipus de cirurgies i els dos sexes, que les edats son bastant properes al voltant dels 40 anys. Els nivells de glucosa, insulina i altres metabolits presenten diferents distribucions, en especial TAS_T0 que clarament no segueix una distribució normal, el què significa que no podrem, de moment, fer cap estudi paramètric. MEDDM_T0 només presenta un sol valor, el què significa que la taula tampoc és, de moment, de rang màxim i una regressió tindrà infinites solucions possibles.

```r
metadata(experiment) <- list(
    titol = "PAC1",
    aula = 3,
    autor = "Joan Mora"
```

```
)
save(experiment, file = "contenidor.Rda")
```

Afegim més informació a la part de metadades de l'experiment i el guardem en format binari.

```
# Funció per corregir valors NA d'una columna amb la mitjana dels altres valors o 0 si tots son NA.
imput <- function(x) {
    nas <- is.na(x)
    mitjana <- mean(x, na.rm = TRUE)
    if(is.na(mitjana)) {
        mitjana <- 0
    }
    x[nas] <- mitjana
    return(x)
}

# Corregim totes les columnes i comprovem que no en quedi cap amb valors NA
matriu <- assay(experiment)[,10:ncol(experiment)]
matriu <- apply(matriu, 2, imput)
sum(is.na(matriu))
```

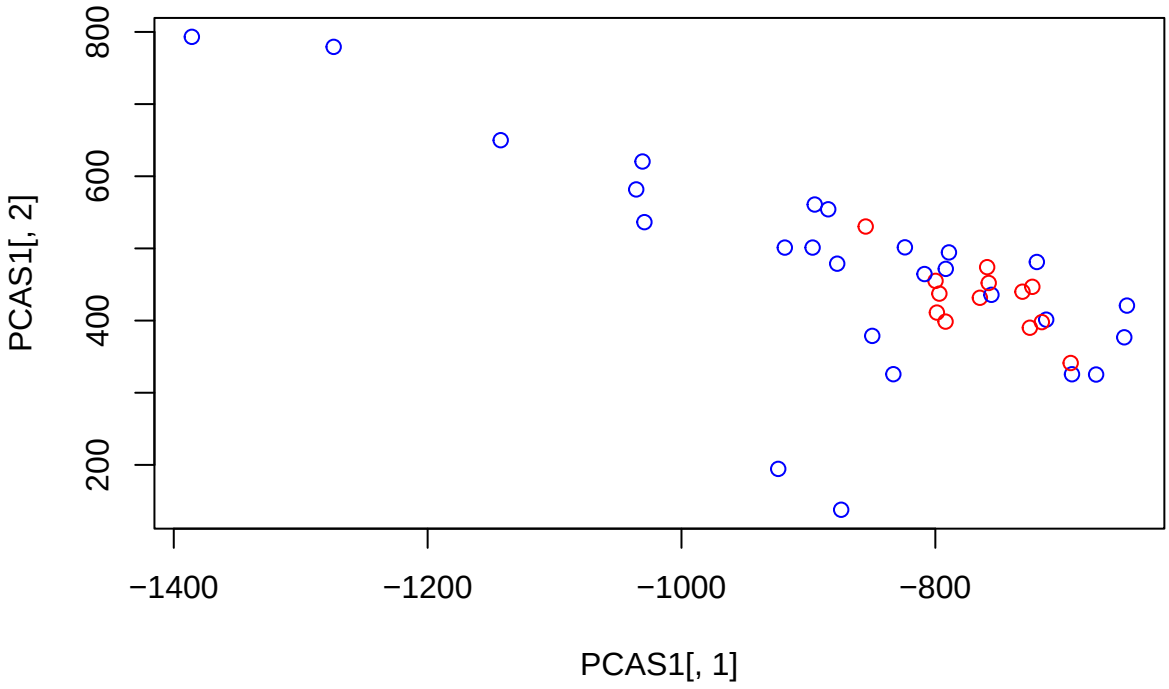
```
## [1] 0
```

```
# Identifiquem i filtrem les columnes que només tenen un valor únic. Perquè si no, al escalar, haurem d
constant_cols <- apply(matriu, 2, function(x) length(unique(x)) == 1)
which(constant_cols)
```

```
##  MEDDM_T2 MEDCOL_T2         X
##       169       170       248
```

```
matriu_filtered <- matriu[, !constant_cols]

# Al ser medicions amb comportaments diferents (sd no son comparables), optem per escalar els valors.
escalat <- scale(matriu_filtered, center = TRUE, scale = TRUE)

# Matri de covariancia ajustada
n <- dim(matriu)[1]
S <- cov(escalat) * (n - 1) / n

# Matriu de correlació
R <- cor(escalat)

# Valors propis
EIG <- eigen(S)

# Obtenir components principals
eigenVecs1 <- EIG$vectors
PCAS1 <- matriu_filtered %*% eigenVecs1

# Visualització 2D de les dues principals components
colors <- c("blue", "red")
surgery_types <- assay(experiment)$SURGERY
```

```r
color_vector <- ifelse(surgery_types == "by pass", colors[1], colors[2])
plot(PCAS1[, 1], PCAS1[, 2], col = color_vector)
```




```r
# Veure quin valor relatiou té cada component principal de forma relativa
vars1 <- EIG$values / sum(EIG$values)
relatius <- round(vars1, 3)
relatius[1:5]
```

```
## [1] 0.138 0.123 0.101 0.080 0.049
```

Finalment hem fet un intent d'analitzar les components principals seguint un exemple dels apunts, on es veu finalment que les dues principals components encaixen en un comportament aparentment definit i que podria ser analitzat a continuació amb una regressió lineal. To i això, els percentatges son molt petits i poc concentrats en els primers components, llavors, per molt que es pugui trobar una relació, aquesta pot no tenir un efecte predictiu important.

Enllaç al repositori: MORA-Peralta-Joan-PEC1 (https://github.com/Yuan1985/MORA-Peralta-Joan-PEC1)