PEC 1 – Análisis de datos ómicos

Pedro Suárez Urquiza

2024-10-28

Contenido

Introducción

En esta prueba de evaluación continua estudiaremos el proceso de análisis de datos ómicos, en específico el análisis de un conjunto de datos de metabolómica. El proceso irá desde la descarga de datos, la creación del objeto SummarizedExperiment hasta el análisis exploratorio de datos en el que comprobaremos brevemente la estructura de nuestro dataset y haremos una breve exploración multivariante.

Objetivos

El objetivo principal del trabajo es el análisis exploratorio de un conjunto de datos de metabolómica. El informe pretende reflejar todo el proceso del análisis, desde la descarga de datos desde un repositorio github, la creación del contenedor SummarizedExperiment, de Bioconductor en R, el análisis exploratorio de los datos y la creación de un repositorio propio donde esté el código y otros objetos pedidos por el enunciado del ejercicio.

Materiales y métodos

Los datos se descargaron del repositorio github proporcionado por el ejercicio: https://github.com/nutrimetabolomics/metaboData/ . Se clonó el repositorio el repositorio desde la terminal de R mediante el comando git clone https://github.com/nutrimetabolomics/metaboData.git. Se seleccionó el primer dataset del repositorio "2018-MetabotypingPaper". Este dataset recoge los metabolitos en sangre de pacientes sometidos a cirugía bariátrica, en diferentes momentos, antes de la cirugía, al mes, a los 3 meses y a los 6 meses.

Para el manejo del conjunto de datos se hará uso de la librería SummarizedExperiment, del paquete Bioconductor. Con esta librería podemos crear un objeto parecido a expressionSet, que contiene datos sobre las variables del experimento, información sobre el experimento en si y la matriz de datos de mediciones. La ventaja de SummarizedExperiment contra expresiónSet es que puede contener varias matrices de datos dentro del objeto (siempre que tengan las mismas variables). Esto tiene la ventaja de permitir tener la matriz de datos sin procesar, con distintos escalados o en diferentes momentos del experimento.

Debido al elevado número de variables de los datos, el análisis exploratorio de los datos se llevará a cabo principalmente mediante análisis multivariante, mediante análisis de componentes principales y agrupamiento jerárquico. Si aparecen grupos dentro del conjunto de datos se intentarán relacionar con la fuente de variabilidad del estudio. Los datos faltantes del dataset fueron imputados.

Resultados

Carga de los datos

El dataset elegido es el del artículo "Metabotypes of response to bariatric surgery independent of the magnitude of weight loss". En el que se miden los metabolitos

en sangre de pacientes con obesidad morbida sometidos a cirugia bariatrica. Con estos podríamos saber que cambios metabólicos experimentan pacientes con este tipo de cirugia.

```
datos <- read.csv("metaboData/Datasets/2018-MetabotypingPaper/DataValues_S013.csv")
datos_variables <- read.csv("metaboData/Datasets/2018-MetabotypingPaper/DataInfo_S013.csv")

colnames(datos)
```

```
##   [1] "X.1"               "SUBJECTS"          "SURGERY"
##   [4] "AGE"               "GENDER"            "Group"
##   [7] "MEDDM_T0"          "MEDCOL_T0"         "MEDINF_T0"
##  [10] "MEDHTA_T0"         "GLU_T0"            "INS_T0"
##  [13] "HOMA_T0"           "HBA1C_T0"          "HBA1C.mmol.mol_T0"
##  [16] "PESO_T0"           "bmi_T0"            "CC_T0"
##  [19] "CINT_T0"           "CAD_T0"            "TAD_T0"
##  [22] "TAS_T0"            "TG_T0"             "COL_T0"
##  [25] "LDL_T0"            "HDL_T0"            "VLDL_T0"
.

.

.

## [688] "SM..OH..C22.2_T5"  "SM..OH..C24.1_T5"  "SM.C16.0_T5"
## [691] "SM.C16.1_T5"       "SM.C18.0_T5"       "SM.C18.1_T5"
## [694] "SM.C20.2_T5"       "SM.C24.0_T5"       "SM.C24.1_T5"
```

```
colnames(datos_variables)
```

```
## [1] "X"           "VarName"     "varTpe"      "Description"
```

Explorando el archivo DataValues, veo que contiene tanto los metadatos de los pacientes como las mediciones, por otro lado DataInfo contiene información sobre las variables del estudio. Por ello, es necesario primero separar los metadatos de las mediciones del archivo “valores”

Primero separamos los metadatos de los pacientes y los transformamos en el tipo “DataFrame”, que es el reconocido por SummarizedExperiment

```
metadatos <- datos[,c("SUBJECTS", "SURGERY", "AGE", "GENDER", "Group")]
metadatos_df <- as(metadatos, "DataFrame")

datos <- subset(datos, select = -c(SUBJECTS, SURGERY, AGE, GENDER, Group,X))
```

Las mediciones tienen bastante valores NA, para que no den problema después podemos realizar una imputación por kNN

```
library("VIM")

datos_imputados <- kNN(datos, k = 7, imp_var = FALSE)
```

Además, las mediciones están separadas en las tomas en varios momentos, T0, T2, T4 y T5, en función del mes en el que se tomo la muestra, antes de la cirugia, al mes, a los 3 o a los 6. Por ello, aprovechando que Summarized experiment puede contener más de una matriz de mediciones, creo que es mejor separarlas. Dentro del bucle selecciono las columnas para cada T, para cada matriz, le quito del nombre el tiempo (ya se sabe por la matriz en la que está contenida) y la transformo en la matriz transpuesta (Se requieren las mediciones en las filas y las observaciones en las columnas).

```
for (t in c("T0","T2","T4","T5")){
  mediciones_t <- datos_imputados[,grepl(t,colnames(datos))]
  print(dim(mediciones_t))
  colnames(mediciones_t) <- gsub(t,"",colnames(mediciones_t))
  assign(paste0("mediciones_", t), t(as.matrix(mediciones_t)))
}
```

```
## [1]  39 172
## [1]  39 173
## [1]  39 172
## [1]  39 172
```

El T2 tiene una fila de más, y todas deberían tener las mismas, la buscamos y la eliminamos.

Seleccionaremos uno de los tiempos, T0, por ejemplo y luego le quitaremos el nombre, para quedarnos con las mismas variables que en los archivos de mediciones. Después lo transformamos en un archivo DataFrame también.

```
datos_variables <- datos_variables[(grepl("T0",datos_variables$VarName
)),]
datos_variables$VarName <- gsub("T0","",datos_variables$VarName)
datos_variables_df <- as(datos_variables,"DataFrame")
```

Ahora podemos crear el contenedor SummarizedExperiment con los datos de las mediciones, los metadatos y la información sobre las variables.

```
se <- SummarizedExperiment(
  assays = list(Tprev = mediciones_T0,
                T1 = mediciones_T2,
                T3 = mediciones_T4,
                T6 = mediciones_T5),
  colData = metadatos_df,
  rowData = datos_variables_df)
```

Análisis exploratorio

Ahora que tenemos el contenedor, podemos pasar a explorarlo.

Podemos acceder al metadata del contenedor con colData()

```
head(colData(se))
```

```
## DataFrame with 6 rows and 5 columns
##    SUBJECTS     SURGERY       AGE      GENDER     Group
```

```
##   <integer> <character> <integer> <character> <integer>
## 1         1     by pass        27           F         1
## 2         2     by pass        19           F         2
## 3         3     by pass        42           F         1
## 4         4     by pass        37           F         2
## 5         5     tubular        42           F         1
## 6         6     by pass        24           F         2
```

```
dim(colData(se))
```

```
## [1] 39  5
```

Vemos que tenemos 39 pacientes, y que las variables clínicas recogidas son el sexo, el tipo de cirugia y el grupo.

La información de las variables está almacenada dentro de rowData().

```
head(rowData(se))
```

```
## DataFrame with 6 rows and 4 columns
##                    X     VarName      varTpe Description
##          <character> <character> <character> <character>
## MEDDM_     MEDDM_T0      MEDDM_     integer    dataDesc
## MEDCOL_   MEDCOL_T0     MEDCOL_     integer    dataDesc
## MEDINF_   MEDINF_T0     MEDINF_     integer    dataDesc
## MEDHTA_   MEDHTA_T0     MEDHTA_     integer    dataDesc
## GLU_         GLU_T0        GLU_     integer    dataDesc
## INS_         INS_T0        INS_     numeric    dataDesc
```

La ventaja de SummarizedExperiment vs Expression sets es que SummarizedExperiment puede almacenar más de una matriz de mediciones, podemos ver todas con assays().

```
assays(se)
```

```
## List of length 4
## names(4): Tprev T1 T3 T6
```

Podríamos acceder a los datos de un ensayo en concreto

```
head(assay(se,"Tprev"),2)
```

```
##         [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,
12] [,13]
## MEDDM_     0    0    0    0    0    0    0    0    0     0     0
0     0
## MEDCOL_    0    0    0    0    0    0    0    0    0     0     0
0     0
##         [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
[,24] [,25]
## MEDDM_      0     0     0     0     0     0     0     0     0     0
0     0
## MEDCOL_     0     0     0     0     0     0     0     0     0     0
0     0
##         [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35]
[,36] [,37]
```

```
## MEDDM_      0     0     0     0     0     0     0     0     0     0
0     0
## MEDCOL_     0     0     1     0     0     0     0     0     0     0
0     0
##         [,38] [,39]
## MEDDM_      0     0
## MEDCOL_     0     0
```

Los valores son recogidos en distintos tiempos, antes de la cirugia y a los meses de esta, por lo que puede ser más interesante ver la evolución. Podemos añadir una nueva matriz (en este caso la evolución entre antes de la cirugia y el ultimo mes del ensayo) al contenedor así:

```
assays(se)[["Tprev_T6"]] <- assay(se,"T6")-assay(se,"Tprev")
```

Al ser tantísimas variables, podríamos hacer un análisis de componentes principales para ver la fuente de variabilidad de los datos:

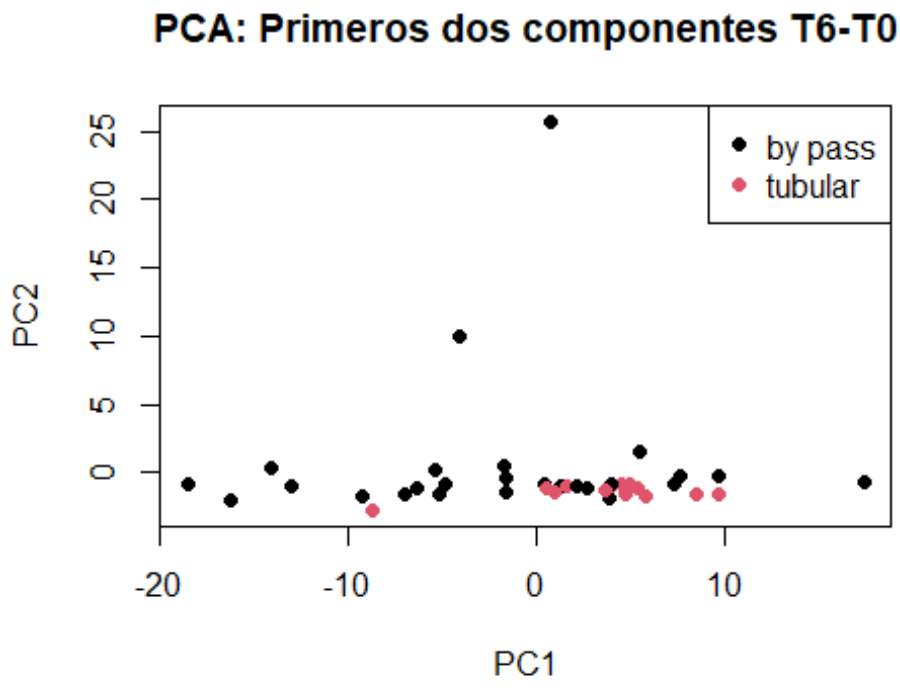
```
pca_T6_Tprev <- prcomp(t(assay(se,"Tprev_T6")),scale. = TRUE)

summary(pca_T6_Tprev)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC
6     PC7
## Standard deviation     7.6983 4.6297 3.20152 2.96095 2.62511 2.4720
3 2.16185
## Proportion of Variance 0.3446 0.1246 0.05959 0.05097 0.04007 0.0355
3 0.02717
## Cumulative Proportion  0.3446 0.4692 0.52876 0.57973 0.61980 0.6553
3 0.68250
##                           PC8     PC9    PC10    PC11   PC12    PC
13    PC14
## Standard deviation     2.13452 2.05553 1.96663 1.89272 1.7837 1.773
23 1.68230
## Proportion of Variance 0.02649 0.02457 0.02249 0.02083 0.0185 0.018
28 0.01645
## Cumulative Proportion  0.70899 0.73355 0.75604 0.77687 0.7954 0.813
65 0.83010
##                          PC15    PC16   PC17    PC18    PC19   PC20
PC21
## Standard deviation     1.6432 1.53265 1.4955 1.45868 1.36582 1.3177
1.28173
## Proportion of Variance 0.0157 0.01366 0.0130 0.01237 0.01085 0.0101
0.00955
## Cumulative Proportion  0.8458 0.85946 0.8725 0.88483 0.89568 0.9058
0.91532
##                           PC22    PC23    PC24    PC25    PC26    P
C27    PC28
## Standard deviation     1.24493 1.20511 1.16877 1.13074 1.06709 0.97
720 0.95988
## Proportion of Variance 0.00901 0.00844 0.00794 0.00743 0.00662 0.00
555 0.00536
## Cumulative Proportion  0.92433 0.93278 0.94072 0.94815 0.95477 0.96
```

```
032 0.96568
##                            PC29    PC30    PC31    PC32    PC33    P
C34    PC35
## Standard deviation     0.92188 0.88793 0.84665 0.83123 0.76580 0.75
178 0.73304
## Proportion of Variance 0.00494 0.00458 0.00417 0.00402 0.00341 0.00
329 0.00312
## Cumulative Proportion  0.97062 0.97521 0.97937 0.98339 0.98680 0.99
009 0.99321
##                            PC36    PC37    PC38      PC39
## Standard deviation     0.67113 0.60879 0.58899 3.419e-15
## Proportion of Variance 0.00262 0.00215 0.00202 0.000e+00
## Cumulative Proportion  0.99583 0.99798 1.00000 1.000e+00
```

Podemos observar que el primer y segundo componente explican el 34,4 y 12,4% de la variabilidad respectivamente, necesitando 20 componentes para explicar el 90% de la variabilidad de la muestra.

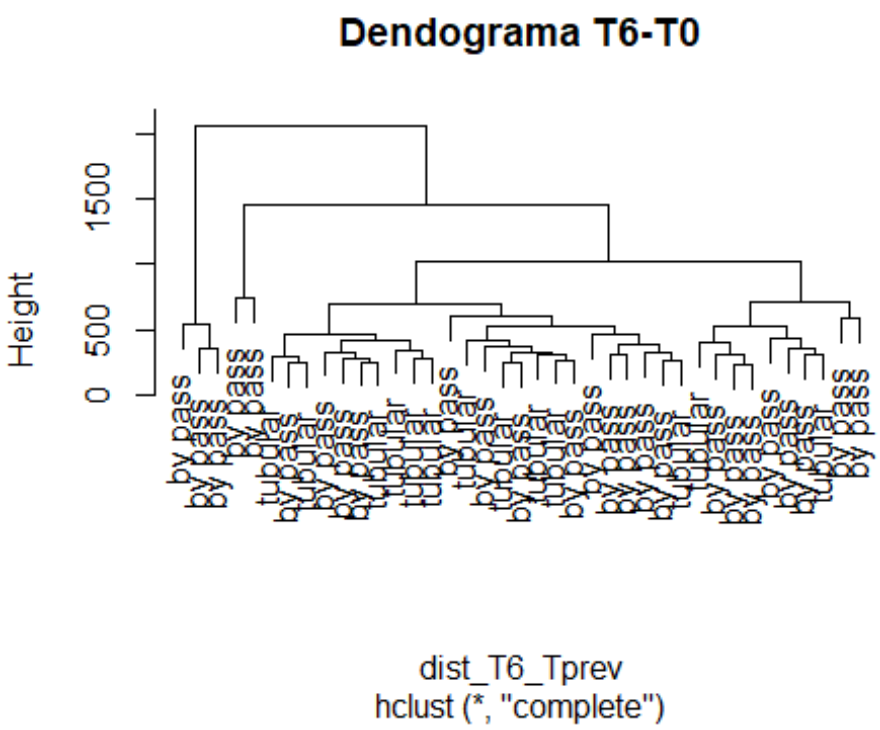
```
plot(pca_T6_Tprev$x[, 1:2], col = factor(colData(se)$SURGERY), pch = 1
9,
     main = "PCA: Primeros dos componentes T6-T0", xlab = "PC1", ylab
= "PC2")
legend("topright", legend = levels(factor(colData(se)$SURGERY)),
       col = 1:length(levels(factor(colData(se)$SURGERY))), pch = 19)
```



La mayor parte de la variabilidad está explicada por el primer componente principal, en el eje del segundo componente podemos observar dos outliers.

Otra forma de hacer un análisis exploratorio de los datos es mediante un clustering de las observaciones.

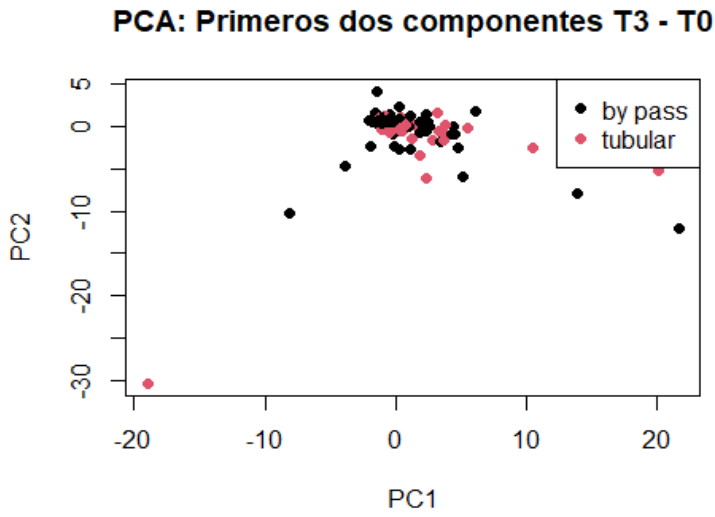
```
dist_T6_Tprev <- dist(t(assay(se,"Tprev_T6")))
clust_T6_Tprev <- hclust(dist_T6_Tprev)
plot(clust_T6_Tprev,labels = factor(colData(se)$SURGERY), main = "Dend
ograma T6-T0")
```

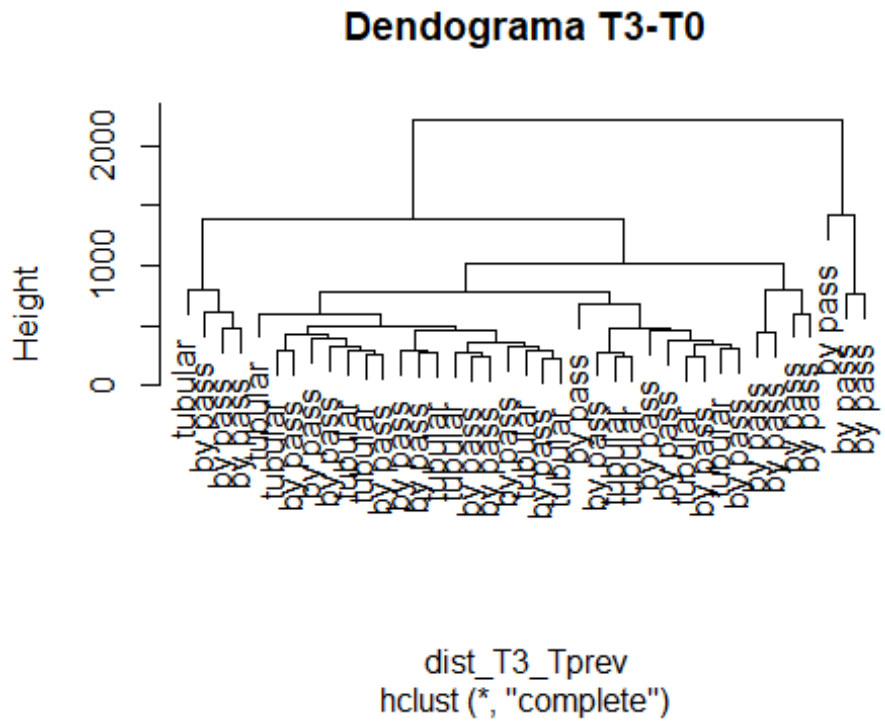


Podemos hacer estas exploraciones para las distintas mediciones. Ocultaré el código en este caso para no hacer el informe muy largo.

T3_T0

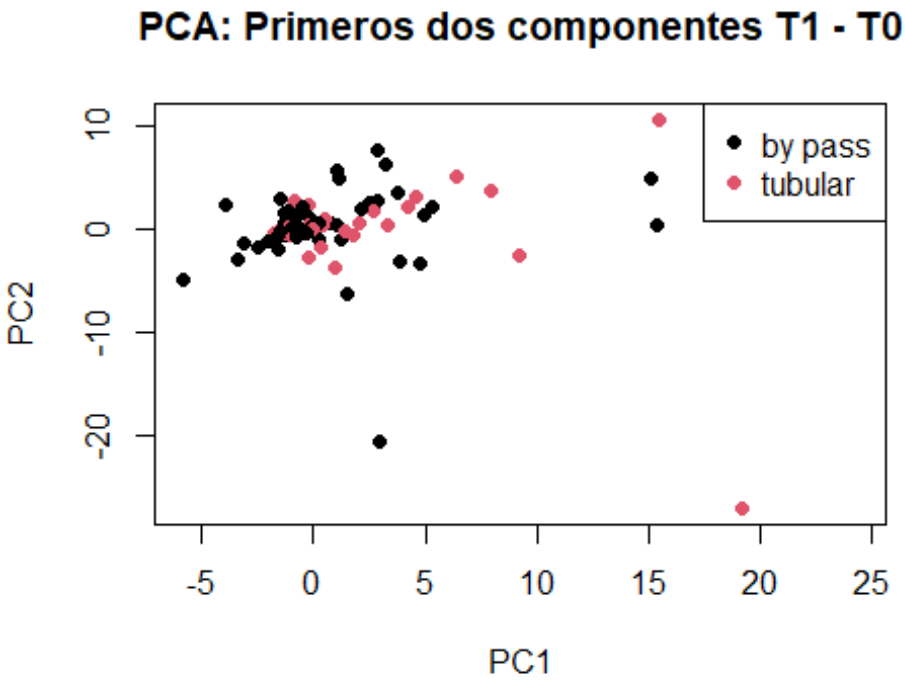
```
## Importance of components:
##                            PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     3.4760 3.0059 2.0760 1.50754 1.27330 1.24479
1.05814
## Proportion of Variance 0.3098 0.2317 0.1105 0.05827 0.04157 0.03973
0.02871
## Cumulative Proportion  0.3098 0.5415 0.6520 0.71027 0.75184 0.79157
0.82028
```

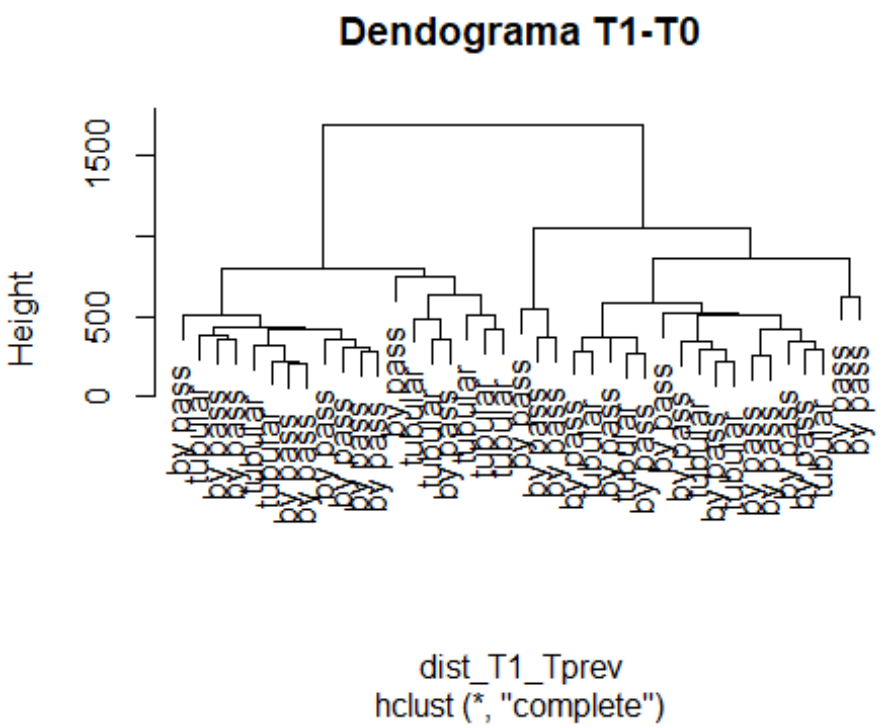




T1_T0

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     3.732 3.2137 1.59772 1.51018 1.21106 1.07848
1.00781
## Proportion of Variance 0.357 0.2648 0.06545 0.05848 0.03761 0.02982
0.02604
## Cumulative Proportion  0.357 0.6219 0.68732 0.74580 0.78340 0.81323
0.83927
```





Parece que la cirugia no explica el mayor porcentaje de la variabilidad de los datos en las distintas mediciones.

Creación del archivo rda y md.

Para guardar el objeto contenedor con los datos y metadatos en formato binario usamos la función save

```
save(se,file = "SummarizedExperiment_PSU.rda")
```

Para crear el archivo markdown con los metadatos primero extraemos los metadatos de las muestras y las variables:

```
metadatos_muestras <- colData(se)

metadatos_variables <- rowData(se)
```

Ahora creamos el archivo markdown

```
metadatos <- file("metadatos_dataset.md")

writeLines(c("# Metadatos del Dataset",
             "## Metadatos de las Muestras",
             "```",
             capture.output(print(metadatos_muestras)),  # Imprimir lo
s metadatos de las muestras
             "```",
             "## Metadatos de las Variables",
             "```",
             capture.output(print(metadatos_variables)),  # Imprimir l
os metadatos de las variables
             "```"), metadatos)

close(metadatos)
```

Creación del repositorio propio

Para la creación de mi repositorio lo hice desde la página de inicio de github, previa creación de cuenta. Eligiendo el nombre del repositorio y eligiendo la opción de público. Para subir mis archivos a este repositorio desde R, cree mi proyecto basado en este repositorio eligiendo las siguientes opciones desde R:

New Project → Version Control → Git → Copiamos link y elegimos nombre del proyecto.

Después, para subir los archivos al repositorio, solo es necesario pulsar commit desde el panel de R y después darle a push.

Discusión

Con el análisis exploratorio hemos visto que tenemos un dataset conformado por las mediciones en distintos momentos de tiempo de pacientes sometidos a cirugía bariátrica. Análizando la evolución de los metabolitos en las distintas tomas comprobamos que la fuente de variación de los datos en estas tomas parece no estar relacionada con el tipo de cirugía realizada en los pacientes. Por otra parte, el dataset presentaba bastantes datos NA, por lo que fue necesaria una imputación previa.

Enlace al repositorio Github

https://github.com/pedrosurqui/Suarez-Urquiza-Pedro-PEC1.git