

# PEC1\_Análisis de datos ómicos

José Ángel Mercado

2024-11-01

## Contents

1. Abstract . . . . .	1
2. Objetivos del estudio . . . . .	1
3. Material y métodos . . . . .	2
4. Resultados . . . . .	2
4.1 Carga de los datos en R . . . . .	2
4.2 Creación del contenedor SummarizedExperiment . . . . .	3
4.3 Exportación del contenedor en formato .RData . . . . .	4
4.4 Depuración de los datos . . . . .	4
4.5 Exploración multivariante de los datos . . . . .	4
4.6 Creación de un repositorio en GitHub . . . . .	10
5. Limitaciones y conclusiones del estudio . . . . .	10
6. Información de la sesión . . . . .	11

## 1. Abstract

El cáncer gástrico es el quinto cáncer más común y el tercero más mortal. Normalmente el diagnóstico del cáncer gástrico es tardío, por lo que es necesario encontrar nuevos métodos de detección. Chan et al. (2016) analizaron la urina de individuos sanos, pacientes con cáncer benigno y pacientes con cáncer gástrico utilizando espectroscopía de resonancia magnética nuclear para cuantificar un gran número de metabolitos. En esta PEC, dichos datos se orgnaizaron en un objeto de clase SummarizedExperiment, se depuraron siguiendo criterios de calidad y se realizó un análisis multivariante de los datos (agrupamiento jerárquico aglomerativo de las muestras y análisis de componentes principales (PCA)). El argupamiento jerárquico no detectó la presencia de ningún clúster que estuviera enriquecido en muestras de alguno de los tres grupos. En cambio, el PCA reveló una diferencia en el perfil metabolómico global de los pacientes con cáncer gástrico respecto de los individuos sanos y de los pacientes con cáncer benigno. Estos resultados son prometedores en el ámbito clínico puesto que la cuantificación de metabolitos podría ayudar al diagnóstico precoz del cáncer gástrico.

## 2. Objetivos del estudio

El objetivo de este ejercicio es determinar si existe un perfil metabolómico global que permita distinguir a los pacientes con cáncer gástrico de los pacientes con cáncer benigno y de los individuos sanos.

### 3. Material y métodos

Para este ejercicio he elegido el conjunto de datos publicado por Chan et al. (2016) y utilizado en el tutorial de CIMBC titulado “Basic Metabolomics Data Analysis Workflow”. Estos datos están disponible en el repositorio de GitHub que se nos proporcionó con datos de ejemplo para trabajar en esta PEC.

Los datos están contenidos en un archivo Excel con dos hojas de datos. En la primera, aparece el nivel medido de distintos metabolitos en una serie de muestras. Las muestras se agrupan en 4 categorías que se definen en la columna “Class”: QC (muestras para realizar el control de calidad del experimento), GC (Gastric Cancer), BN (“Benign Tumor”) y HC (“Healthy controls”). En la segunda hoja del Excel, encontramos el nombre de los metabolitos que se han medido en este estudio, así como algunos datos sobre la calidad de las medidas de dichos metabolitos y el porcentaje de muestras en el que no se pudo medir dichos metabolitos.

Dichos datos se cargaron en R en un contenedor de tipo SummarizedExperiment. A continuación, se filtraron siguiendo dos criterios de calidad, de forma que solo se continuó el análisis con los metabolitos con un QC-RSD inferior al 20% y un porcentaje de valores nulos (NA) inferior al 10%. Por último, se realizó un agrupamiento jerárquico aglomerativo de las muestras y un análisis de componentes principales.

### 4. Resultados

#### 4.1 Carga de los datos en R

```
# Para leer el archivo Excel utilizaré el paquete openxlsx
library(openxlsx)

# Cargamos cada hoja del archivo Excel en un dataframe de R
data = read.xlsx("GastricCancer_NMR.xlsx", sheet = "Data")
peak = read.xlsx("GastricCancer_NMR.xlsx", sheet = "Peak")

# Exploramos las primeras observaciones de ambas tablas
data[1:5, 1:6]
```

```
##   Idx SampleID SampleType Class    M1    M2
## 1   1 sample_1         QC    QC  90.1  491.6
## 2   2 sample_2        Sample    GC  43.0  525.7
## 3   3 sample_3        Sample    BN 214.3 10703.2
## 4   4 sample_4        Sample    HE  31.6   59.7
## 5   5 sample_5        Sample    GC  81.9  258.7
```

```
head(peak)
```

```
##   Idx Name                                Label Perc_missing   QC_RSD
## 1   1   M1      1_3-Dimethylurate          11.4285714 32.208005
## 2   2   M2  1_6-Anhydro- -D-glucose        0.7142857 31.178028
## 3   3   M3    1_7-Dimethylxanthine         5.0000000 34.990605
## 4   4   M4    1-Methylnicotinamide         8.5714286 12.804201
## 5   5   M5      2-Amino adipate            1.4285714  9.372664
## 6   6   M6      2-Aminobutyrate            5.0000000 46.977149
```

```
dim(data)
```

```
## [1] 140 153
```

```
dim(peak)
```

```
## [1] 149 5
```

Tenemos un total de 140 muestras y 149 metabolitos cuantificados.

## 4.2 Creación del contenedor SummarizedExperiment

El contenedor SummarizedExperiment se puede dividir en cuatro partes, la matriz con los datos, la información sobre las filas, la información sobre las columnas y los metadatos del experimento. En nuestro caso, la matriz contendrá la cuantificación de cada metabolito en las distintas muestras. Cada fila corresponderá a un metabolito y cada columna corresponderá a una muestra, es decir, los datos estarán traspuestos respecto a como están contenidos en el archivo Excel. La información de las filas será básicamente el nombre completo de cada metabolito. La información de las columnas hará referencia a las muestras y a qué grupo pertenece cada muestra (en nuestro caso, HC, BN, GC). Por último, el contenedor SummarizedExperiment permite añadir metadatos, es decir, información en formato de texto libre sobre el experimento (autores, artículo en el que se ha publicado, etc.).

Primero crearemos cada una de las cuatro partes por separado, y luego crearemos el contenedor con la función SummarizedExperiment().

```
# Cargamos el paquete SummarizedExperiment de Bioconductor
library(SummarizedExperiment)

# Creamos la matriz de datos
# Para ello, seleccionamos solo la cuantificación de cada metabolito en cada muestra
matriz.datos = as.matrix(data[, 5:153])

# Añadimos los nombres de las filas y columnas
rownames(matriz.datos) = data$SampleID
colnames(matriz.datos) = peak$Name

# Trasponemos dicha matriz
matriz.datos = t(matriz.datos)

# Creamos un dataframe que contendrá la información de las filas (metabolitos)
info.filas = data.frame(peak[, 2:5])

# Creamos un dataframe que contendrá la información de las columnas (muestras)
info.columnas = data.frame(data[2:4])

# Creamos un objeto que contendrá otra información sobre el experimento (metadata)
# He decidido que contenga la referencia del artículo del que proceden los datos
metadatos = "Chan, A. W., Mercier, P., Schiller, D., Bailey, R., Robbins, S., Eurich, D. T., Sawyer, M."

se = SummarizedExperiment(assays = list(metabolite_cuantification=matriz.datos),
                           colData = info.columnas,
                           rowData = info.filas,
                           metadata = list(publication_reference=metadatos))

se
```

```
## class: SummarizedExperiment
## dim: 149 140
## metadata(1): publication_reference
## assays(1): metabolite_cuantification
## rownames(149): M1 M2 ... M148 M149
## rowData names(4): Name Label Perc_missing QC_RSD
## colnames(140): sample_1 sample_2 ... sample_139 sample_140
## colData names(3): SampleID SampleType Class
```

### 4.3 Exportación del contenedor en formato .RData

En el enunciado de la PEC se pide que se incluya el contenedor SummarizedExperiment en formato .RData en el repositorio de GitHub. El código de R para exportar el objeto “se” a formato .RData se detalla a continuación:

```
save(se, file = "Metabolic Data in a SummarizedExperiment container.RData")
```

Además, el enunciado de la PEC pide que los datos originales se carguen en formato texto en el repositorio de GitHub. Yo he decidido subir cada hoja del archivo Excel original en un fichero de texto. Para obtener dichos ficheros de texto, el código de R empleado fue:

```
write.table(data, file = "Original Metabolic Data.txt")
write.table(peak, file = "Original Peak Data.txt")
```

### 4.4 Depuración de los datos

Una vez tenemos los datos en el formato SummarizedExperiment, lo primero es depurar los datos. La idea es eliminar los metabolitos que no se hayan cuantificado correctamente. Los criterios de calidad que he aplicado son los mismos que aparecen en el tutorial de CIMBC. De forma que vamos a seleccionar los metabolitos que tengan un QC-RSD (una medida de calidad) inferior al 20% y también los metabolitos que tengan menos de un 10% de valores nulos (NA). En el objeto “se”, los metabolitos ocupan las filas, por lo que seleccionaremos únicamente las filas que cumplan ambos criterios.

```
se_depurado = se[rowData(se)$Perc_missing < 10 & rowData(se)$QC_RSD < 20, ]
se_depurado
```

```
## class: SummarizedExperiment
## dim: 52 140
## metadata(1): publication_reference
## assays(1): metabolite_cuantification
## rownames(52): M4 M5 ... M148 M149
## rowData names(4): Name Label Perc_missing QC_RSD
## colnames(140): sample_1 sample_2 ... sample_139 sample_140
## colData names(3): SampleID SampleType Class
```

Después del filtrado, el número de metabolitos se ha reducido de 149 a 52.

### 4.5 Exploración multivariante de los datos

Como una primera exploración global de los datos, realicé un agrupamiento jerárquico aglomerativo de las muestras. Para ello, se escalaron los datos de todas las variables, se calculó la matriz de distancias y, finalmente se realizó el agrupamiento jerárquico aglomerativo y se representó en forma de dendograma.

```

# Primero se escalan los datos
datos.escalados = scale(t(assays(se_depurado)[[1]]))

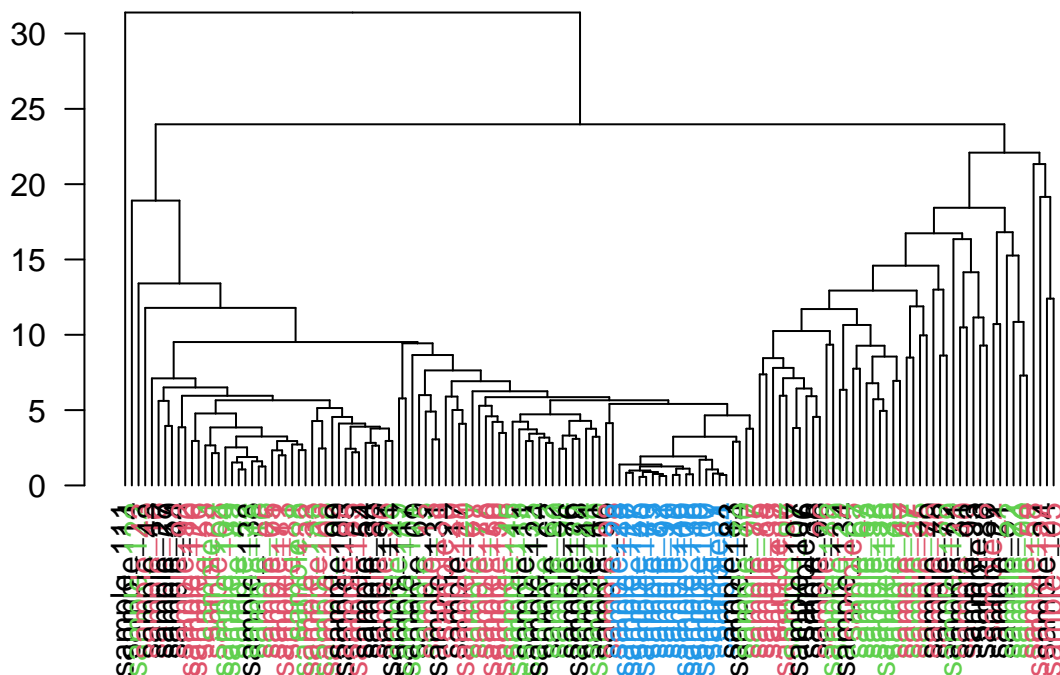
# Cálculo de la matriz de distancia
matriz.distancia = dist(datos.escalados)
dendrograma = as.dendrogram(hclust(matriz.distancia))

# Es interesante conocer si las muestras se agrupan en función de si provienen de un
# paciente o de un individuo sano. Para ello, vamos a colorear el nombre de cada muestra
# El color 1 (negro) será para BN (cáncer benigno)
# El color 2 (rosa) será para GC (cáncer gástrico)
# El color 3 (verde) será para HE (individuos sanos)
# El color 4 (azul) será para QC (control de calidad)
colores = as.factor(se_depurado$Class)

# Para que funcione correctamente, hay que ordenar los colores para que sigan el mismo
# orden que en el dendrograma
# La función labels_colors que asigna el color a las etiquetas del dendrograma está
# disponible en el paquete dendextend
library(dendextend)
labels_colors(dendrograma) = as.numeric(colores[order.dendrogram(dendrograma)])

# Finalmente visualizamos el dendrograma
plot(dendrograma, las = 1)

```



Las muestras del grupo QC (control de calidad) forman claramente un clúster separado del resto. Las

muestras de los otros 3 grupos no parecen formar clusters independientes, sino que aparecen mezcladas entre sí.

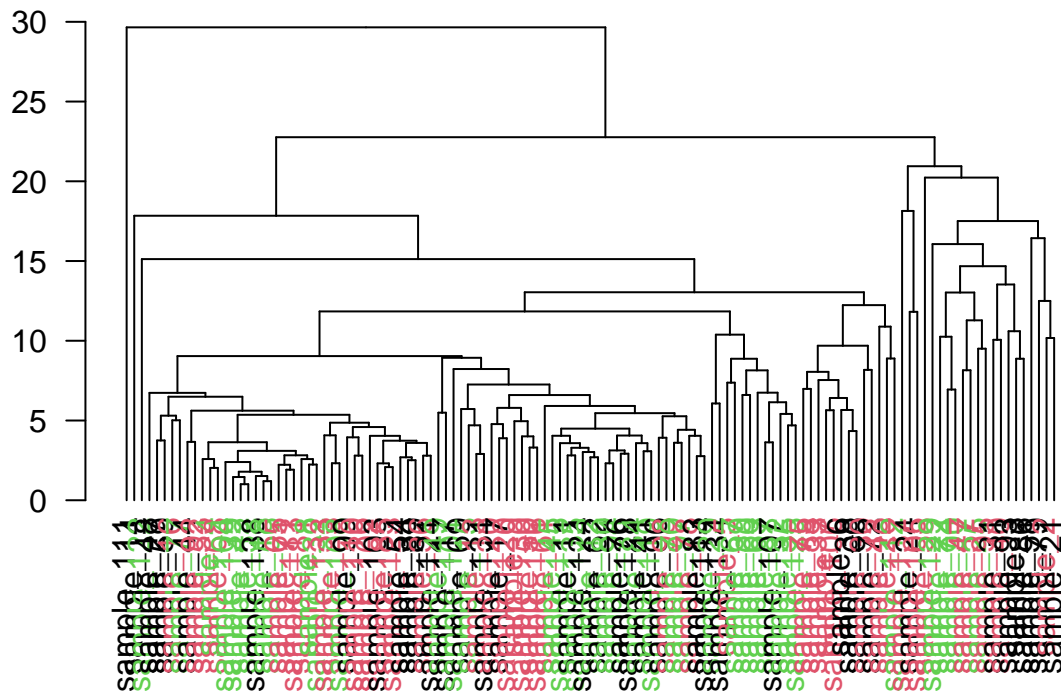
Como el objetivo de nuestro análisis es averiguar si hay un perfil metabolómico global que diferencie a los pacientes con cáncer gástrico (benigno o no) de los individuos sanos, vamos a repetir el agrupamiento jerárquico eliminando primero las muestras etiquetadas como QC (control de calidad).

```
se_depurado = se_depurado[ , se_depurado$Class=="BN" |  
                             se_depurado$Class=="GC" |  
                             se_depurado$Class=="HE"]  
se_depurado
```

```
## class: SummarizedExperiment  
## dim: 52 123  
## metadata(1): publication_reference  
## assays(1): metabolite_cuantification  
## rownames(52): M4 M5 ... M148 M149  
## rowData names(4): Name Label Perc_missing QC_RSD  
## colnames(123): sample_2 sample_3 ... sample_138 sample_139  
## colData names(3): SampleID SampleType Class
```

El número de muestras ha descendido de 140 a 123.

```
# Repetimos el agrupamiento jerárquico aglomerativo  
# Se escalan los datos  
datos.escalados = scale(t(assays(se_depurado)[[1]]))  
  
# Cálculo de la matriz de distancia  
matriz.distancia = dist(datos.escalados)  
dendograma = as.dendrogram(hclust(matriz.distancia))  
  
# En esta ocasión, solo tendremos los 3 primeros colores  
colores = as.factor(se_depurado$Class)  
  
# Ordenamos los colores y los asignamos a las etiquetas del dendograma  
labels_colors(dendograma) = as.numeric(colores[order.dendrogram(dendograma)])  
  
# Finalmente visualizamos el dendograma  
plot(dendograma, las = 1)
```



De nuevo, ninguno de los clústers que se forman están formados exclusivamente por muestras de uno de los grupos. Al contrario, parece que las muestras de los 3 grupos se encuentran mezcladas. Por otro lado, podemos observar como la muestra 111, es la más diferente del resto.

Como otra aproximación distinta para la exploración multivariante de los datos, se realizó un análisis de componentes principales (PCA).

```
# Matriz con los datos
# Seleccionamos solo las muestras que no contengan ningún NA entre las medidas
casos.completos = complete.cases(t(assays(se_depurado)[[1]]))
x = t(assays(se_depurado)[[1]])[casos.completos, ]

# De nuevo vamos a colorear las etiquetas en función del grupo al que pertenezca la muestra
colores = as.factor(se_depurado$Class)[casos.completos]

# Para realizar el PCA, utilicé el paquete FactoMineR
library(FactoMineR)
pca = PCA(x, scale.unit = TRUE, graph = FALSE)
summary.PCA(pca)
```

```
##
## Call:
## PCA(X = x, scale.unit = TRUE, graph = FALSE)
##
##
## Eigenvalues
```

```

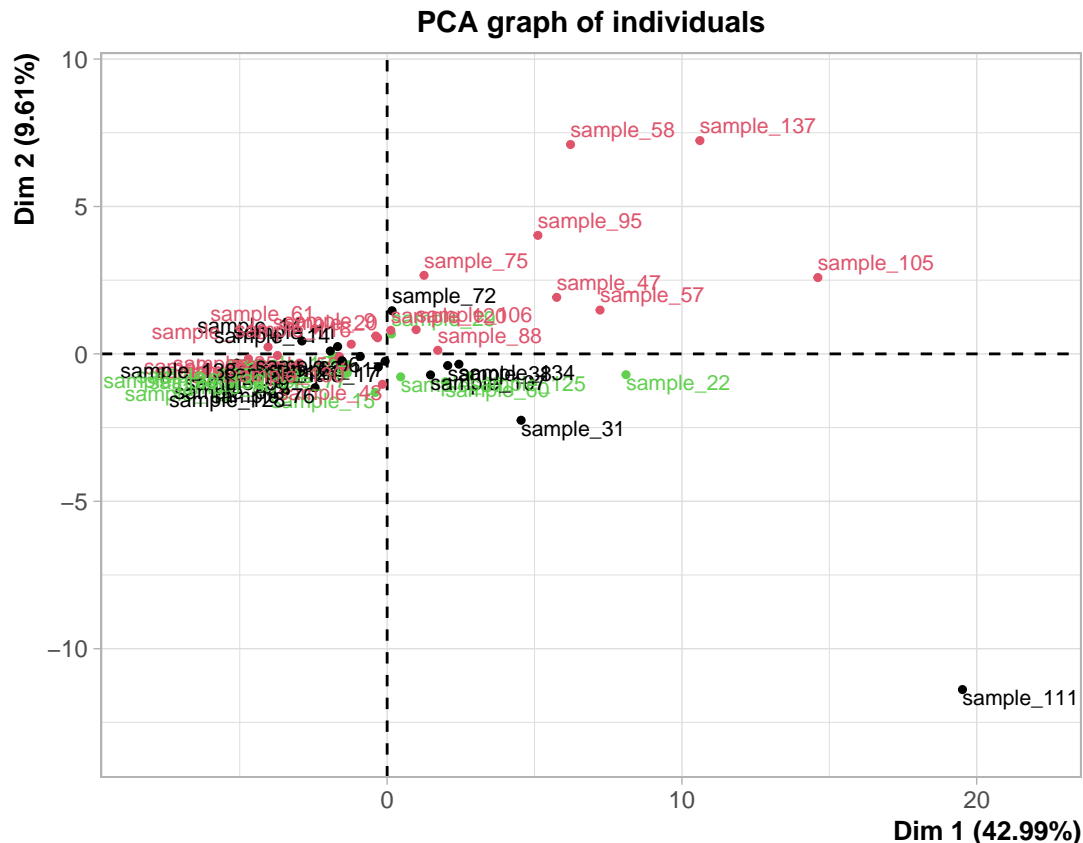
##          Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance      22.353    5.000    3.200    2.828    2.367    2.009    1.695
## % of var.     42.987    9.615    6.154    5.438    4.551    3.863    3.260
## Cumulative % of var. 42.987  52.601  58.755  64.194  68.745  72.608  75.868
##          Dim.8   Dim.9   Dim.10  Dim.11  Dim.12  Dim.13  Dim.14
## Variance      1.467    1.366    1.212    1.057    0.931    0.828    0.669
## % of var.     2.821    2.628    2.331    2.033    1.790    1.592    1.286
## Cumulative % of var. 78.689  81.317  83.648  85.681  87.471  89.063  90.349
##          Dim.15  Dim.16  Dim.17  Dim.18  Dim.19  Dim.20  Dim.21
## Variance      0.595    0.530    0.455    0.430    0.355    0.324    0.307
## % of var.     1.145    1.019    0.876    0.826    0.683    0.624    0.591
## Cumulative % of var. 91.494  92.513  93.388  94.215  94.898  95.522  96.113
##          Dim.22  Dim.23  Dim.24  Dim.25  Dim.26  Dim.27  Dim.28
## Variance      0.246    0.213    0.195    0.172    0.163    0.135    0.123
## % of var.     0.473    0.409    0.374    0.331    0.313    0.260    0.237
## Cumulative % of var. 96.586  96.995  97.370  97.700  98.013  98.274  98.510
##          Dim.29  Dim.30  Dim.31  Dim.32  Dim.33  Dim.34  Dim.35
## Variance      0.102    0.096    0.089    0.078    0.056    0.053    0.046
## % of var.     0.196    0.184    0.171    0.149    0.107    0.103    0.089
## Cumulative % of var. 98.706  98.890  99.061  99.210  99.317  99.420  99.510
##          Dim.36  Dim.37  Dim.38  Dim.39  Dim.40  Dim.41  Dim.42
## Variance      0.044    0.040    0.034    0.028    0.024    0.018    0.014
## % of var.     0.085    0.078    0.065    0.054    0.046    0.035    0.027
## Cumulative % of var. 99.595  99.672  99.737  99.791  99.837  99.872  99.899
##          Dim.43  Dim.44  Dim.45  Dim.46  Dim.47  Dim.48  Dim.49
## Variance      0.012    0.010    0.009    0.007    0.005    0.004    0.003
## % of var.     0.024    0.019    0.018    0.013    0.010    0.008    0.005
## Cumulative % of var. 99.922  99.941  99.959  99.972  99.982  99.990  99.995
##          Dim.50  Dim.51  Dim.52
## Variance      0.002    0.001    0.000
## % of var.     0.003    0.001    0.000
## Cumulative % of var. 99.998 100.000 100.000
##
## Individuals (the 10 first)
##          Dist   Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3
## sample_8 | 5.276 | -4.738 1.702 0.806 | -0.350 0.041 0.004 | 1.087
## sample_9 | 4.107 | -0.384 0.011 0.009 | 0.607 0.125 0.022 | 0.533
## sample_11 | 3.401 | -1.678 0.214 0.244 | 0.248 0.021 0.005 | 0.295
## sample_12 | 2.994 | -1.856 0.261 0.384 | -0.068 0.002 0.001 | 0.268
## sample_14 | 10.037 | -2.889 0.633 0.083 | 0.438 0.065 0.002 | 0.445
## sample_15 | 3.851 | -0.404 0.012 0.011 | -1.309 0.581 0.116 | 0.353
## sample_17 | 3.111 | -0.298 0.007 0.009 | -0.445 0.067 0.020 | -0.284
## sample_18 | 5.721 | -5.355 2.174 0.876 | -1.064 0.383 0.035 | 0.488
## sample_20 | 4.392 | -0.323 0.008 0.005 | 0.547 0.101 0.016 | -0.837
## sample_22 | 11.907 | 8.104 4.980 0.463 | -0.706 0.169 0.004 | 3.257
##          ctr   cos2
## sample_8 0.626 0.042 |
## sample_9 0.150 0.017 |
## sample_11 0.046 0.008 |
## sample_12 0.038 0.008 |
## sample_14 0.105 0.002 |
## sample_15 0.066 0.008 |
## sample_17 0.043 0.008 |
## sample_18 0.126 0.007 |

```



```
## sample_20 0.371 0.036 |
## sample_22 5.618 0.075 |
##
## Variables (the 10 first)
##          Dim.1    ctr    cos2    Dim.2    ctr    cos2    Dim.3    ctr    cos2
## M4          | 0.510  1.165 0.260 | -0.410  3.367 0.168 | -0.102  0.325 0.010
## M5          | 0.826  3.055 0.683 | -0.112  0.252 0.013 | -0.017  0.009 0.000
## M7          | 0.451  0.912 0.204 |  0.631  7.954 0.398 |  0.234  1.707 0.055
## M8          | 0.779  2.715 0.607 | -0.476  4.522 0.226 |  0.278  2.415 0.077
## M11         | 0.587  1.542 0.345 | -0.498  4.970 0.248 |  0.141  0.618 0.020
## M14         | 0.677  2.053 0.459 | -0.171  0.584 0.029 |  0.344  3.706 0.119
## M15         | 0.730  2.386 0.533 |  0.012  0.003 0.000 |  0.290  2.629 0.084
## M25         | 0.717  2.298 0.514 |  0.453  4.100 0.205 |  0.180  1.009 0.032
## M26         | 0.437  0.855 0.191 |  0.508  5.158 0.258 |  0.517  8.346 0.267
## M31         | 0.829  3.074 0.687 | -0.256  1.309 0.065 |  0.036  0.041 0.001
##
## M4          |
## M5          |
## M7          |
## M8          |
## M11         |
## M14         |
## M15         |
## M25         |
## M26         |
## M31         |
```

```
plot.PCA(pca, cex = 0.7, habillage = "ind", col.hab = as.numeric(colores))
```



Al igual que ocurría con el agrupamiento jerárquico aglomerativo, en el gráfico del PCA se observa como la muestra 111 es la más diferente del resto. Por otro lado, se observa que la componente 2 del PCA, aunque solo explica el 9.61% de la varianza, parece separar las muestras entre individuos sanos o con cáncer benigno (colores verde y negro, respectivamente) de los individuos con cáncer gástrico (color rosa).

## 4.6 Creación de un repositorio en GitHub

He creado un repositorio en GitHub en el que he incluido el informe del trabajo realizado en formato HTML, el archivo RMarkdown con el que se generó el informe, el contenedor SummarizedExperiment en formato RData, los datos originales en formato Excel y txt y un archivo con los metadatos en formato md.

El link de dicho repositorio es: <https://github.com/ja-mercado/Mercado-Hornos-JoseAngel-PEC1>

## 5. Limitaciones y conclusiones del estudio

La conclusión de este estudio es que, basándonos en el resultado del PCA, los pacientes con cáncer gástrico tienen un perfil metabólico global diferente del de los individuos sanos, mientras que los pacientes con cáncer benigno tienen un perfil metabólico no distinguible de los individuos sanos.

La principal limitación de este estudio es que solo se ha realizado un análisis global, teniendo en cuenta todas las variables simultáneamente. Desde el punto de visto clínico, lo ideal sería identificar un único metabolito (o un grupo muy reducido) que permitiera distinguir a los pacientes de los individuos sanos, ya que es más barato y menos laborioso cuantificar un único metabolito que realizar un perfil metabolómico de cada posible paciente. Además, al tener en cuenta todas las variables de forma global, podríamos estar enmascarando grandes diferencias entre los grupos que solo se dan a nivel de uno o pocos metabolitos. Por último, para

el PCA, solo se tuvieron en cuenta las muestras en las que se pudieron cuantificar todos los metabolitos, es decir, aquellas muestras que no presentaban un valor nulo (NA) en ninguna de las variables.

## 6. Información de la sesión

```
sessionInfo()
```

```
## R version 4.3.3 (2024-02-29 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.utf8  LC_CTYPE=Spanish_Spain.utf8
## [3] LC_MONETARY=Spanish_Spain.utf8 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.utf8
##
## time zone: Europe/Madrid
## tzcode source: internal
##
## attached base packages:
## [1] stats4      stats      graphics  grDevices  utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] FactoMineR_2.11             dendextend_1.18.1
## [3] SummarizedExperiment_1.32.0 Biobase_2.62.0
## [5] GenomicRanges_1.54.1       GenomeInfoDb_1.38.8
## [7] IRanges_2.36.0             S4Vectors_0.40.2
## [9] BiocGenerics_0.48.1         MatrixGenerics_1.14.0
## [11] matrixStats_1.4.1          openxlsx_4.2.7.1
##
## loaded via a namespace (and not attached):
## [1] gtable_0.3.5                xfun_0.49                ggplot2_3.5.1
## [4] htmlwidgets_1.6.4          ggrepel_0.9.6            lattice_0.22-5
## [7] vctrs_0.6.5                 tools_4.3.3              bitops_1.0-7
## [10] generics_0.1.3             tibble_3.2.1             fansi_1.0.6
## [13] cluster_2.1.6              highr_0.11               pkgconfig_2.0.3
## [16] Matrix_1.6-5               scatterplot3d_0.3-44     lifecycle_1.0.4
## [19] GenomeInfoDbData_1.2.11    farver_2.1.2             compiler_4.3.3
## [22] munsell_0.5.1              leaps_3.2                htmltools_0.5.8.1
## [25] Rcurl_1.98-1.14            yaml_2.3.8               pillar_1.9.0
## [28] crayon_1.5.2               MASS_7.3-60.0.1          DT_0.33
## [31] flashClust_1.01-2          DelayedArray_0.28.0      viridis_0.6.5
## [34] abind_1.4-5                tidyselect_1.2.1         zip_2.3.1
## [37] digest_0.6.35              mvtnorm_1.3-1            stringi_1.8.4
## [40] dplyr_1.1.4                labeling_0.4.3           fastmap_1.2.0
## [43] grid_4.3.3                 colorspace_2.1-0         cli_3.6.2
## [46] SparseArray_1.2.4          magrittr_2.0.3           S4Arrays_1.2.1
## [49] utf8_1.2.4                 withr_3.0.0              scales_1.3.0
```

## [52]	estimability_1.5.1	rmarkdown_2.27	XVector_0.42.0
## [55]	emmeans_1.10.5	gridExtra_2.3	evaluate_0.23
## [58]	knitr_1.45	viridisLite_0.4.2	rlang_1.1.3
## [61]	Rcpp_1.0.12	xtable_1.8-4	glue_1.7.0
## [64]	rstudioapi_0.17.1	R6_2.5.1	zlibbioc_1.48.2
## [67]	multcompView_0.1-10		