PEC1 Análisis de datos ómicos

Alan Padilla Campoy

2024-11-05

Repositorio de GitHub: https://github.com/apadillacamuoc/Padilla-Campoy-Alan-PEC1.git

El documento que escogí es el documento **GastricCancer_NMR.xlsx** de la carpeta **2023-CIMCBTutorial** que son datos de cancer gastrico usados en un tutorial de analisis de datos metabolomicos.

Lo primero que debemos hacer es leer el archivo. El archivo Excel tiene dos hojas, la primera llamada *Data* y contiene lo siguiente:

- **Idx:** número de identificación de la muestra
- **SampleID:** código de identificación de la muestra
- **SampleType:** es el tipo general de muestra, QC es un control del estudio y Sample es una muestra obtenida de un paciente.
- **Class:** es el tipo de muestra. QC (control del estudio), GC (muestra de paciente con cáncer), BN (muestra de paciente con tumor benigno) y HE (muestra de paciente sano).
- **Metabolitos M1, M2, ..., M149:** las concentraciones de los metabolitos en las muestras de orina de los pacientes.

La segunda hoja llamada *Peak* incluye la información relevante de cada metabolito.

- **Idx:** número de identificación del metabolito
- **Name:** nombre que el metabolito tiene en la hoja *Data*
- **Label:** nombre quimico del metabolito
- **Perc_missing:** que porcentaje de muestras no contienen mediciones para este metabolito
- **QC_RSD:** puntuación de calidad que representa la variación en las mediciones de este metabolito en todas las muestras

```
library(readxl)
library(SummarizedExperiment)

data_sheet <- read_excel("GastricCancer_NMR.xlsx", sheet = "Data")
peak_sheet <- read_excel("GastricCancer_NMR.xlsx", sheet = "Peak")
```

Posteriormente creamos el elemento SummarizedExperiment, utilizando la libreria descargada de Bioconductor. Primero extraemos la matriz de datos. Creamos *dataframes* de los metadatos de las filas y columnas.

```
data <- as.matrix(data_sheet[, 5:ncol(data_sheet)])

row_data <- data_sheet[, c("Idx", "SampleID", "SampleType", "Class")]
col_data <- peak_sheet[, c("Idx", "Name", "Label", "Perc_missing",
"QC_RSD")]

rownames(data) <- row_data$SampleID
colnames(data) <- col_data$Name
```

Posteriormente podemos utilizar esa información de las filas y columnas para crear el objeto SummarizedExperiment.

```
se_object <- SummarizedExperiment(
  assays = list(counts = data),
  rowData = row_data,
  colData = col_data
)

se_object

## class: SummarizedExperiment
## dim: 140 149
## metadata(0):
## assays(1): counts
## rownames(140): sample_1 sample_2 ... sample_139 sample_140
## rowData names(4): Idx SampleID SampleType Class
## colnames(149): M1 M2 ... M148 M149
## colData names(5): Idx Name Label Perc_missing QC_RSD
```

Para poder acceder a la información con facilidad podriamos generar un archivo binario .Rda, que contenga los datos y metadatos.

```
save(se_object, file = "GastricCancer_NMR_data.Rda")
```

Para generar un archivo de texto con la matriz de los datos podemos hacerlo de la siguiente manera:

```
data_matrix <- assay(se_object, "counts")
row_metadata <- rowData(se_object)
col_metadata <- colData(se_object)

write.table(data_matrix, file = "datos.txt", sep = "\t", row.names =
TRUE, col.names = TRUE, quote = FALSE)
```

Para generar un archivo .md con los metadatos podemos hacerlo así:

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

writeLines(c(
  "# Metadatos del Dataset",
  "",
```

```
  "## Información General",
  paste("Número de muestras:", dim(se_object)[1]),
  paste("Número de metabolitos:", dim(se_object)[2]),
  "",
  "## Resumen del Objeto",
  "```",
  capture.output(print(se_object)),
  "```",
  "",
  "## Metadatos de las Muestras",
  "```",
  capture.output(head(rowData(se_object))),
  "```",
  "",
  "## Metadatos de los Metabolitos",
  "```",
  capture.output(head(colData(se_object))),
  "```"
), fileConn)

close(fileConn)
```

Exploracion general de los datos

Para explorar los datos podemos utilizar el objeto SummarizedExperiment que acabamos de crear. Primero extraemos las concentraciones de metabolitos, las clases de los datos y lo combinamos en un *dataframe*. También nos aseguramos que las clases sean únicas.

```
metabolite_data <- assay(se_object)
class_data <- rowData(se_object)$Class
data_combined <- as.data.frame(metabolite_data)
data_combined$Class <- class_data
unique_classes <- unique(class_data)
```

Podemos comparar la estadística descriptiva para ver que características generales tienen los datos.

```
summary_data <- data.frame(
    Mean = colMeans(assay(se_object), na.rm = TRUE),
    SD = apply(assay(se_object), 2, sd, na.rm = TRUE),
    Min = apply(assay(se_object), 2, min, na.rm = TRUE),
    Max = apply(assay(se_object), 2, max, na.rm = TRUE)
)
summary_data
```

```
##             Mean           SD   Min      Max
## M1    101.070968   123.613783   0.4    909.9
## M2    641.997842  2397.535634   3.1  26195.8
## M3    146.366917   131.850171   0.1    862.5
```

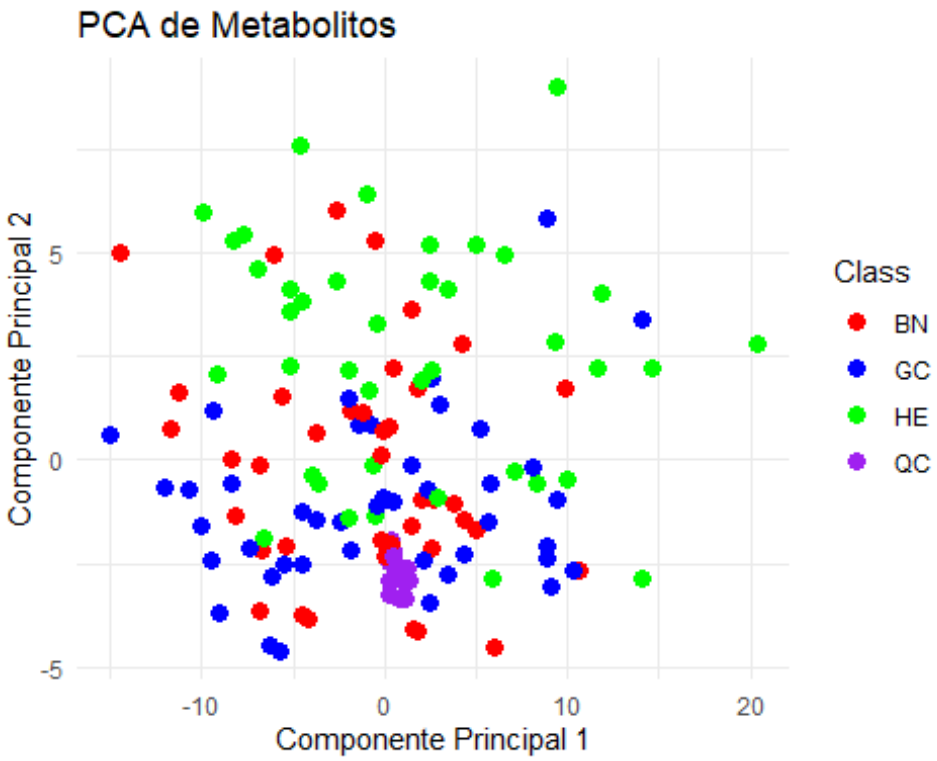
```
## M4     43.833594    39.051947   0.1    242.5
## M5    231.107971   337.542135   1.3   2503.0
## M6     41.633835    48.400778   0.2    339.4
## M7     74.119853    97.379986   4.6    492.6
## M8     67.809286    61.131399   9.3    525.0
## M9     64.099115    78.150663   0.7    612.1
## M10   124.809302   205.715425   0.1   2026.8
## M11   192.530827   366.126168   0.3   2676.3
## M12    69.127857    67.770383   1.7    576.2
## M13   376.102857  1175.674366  17.2  10712.7
## M14    68.748905    63.046778   0.2    437.6
## M15    57.157143    41.037884   7.9    212.3
## M16   125.033077   118.421972   3.7    665.9
## M17    39.161765    41.794732   0.4    187.6
## M18   210.986232   207.146351   2.6   1236.5
## M19    65.745312    49.997808   0.4    217.7
## M20    62.467669    61.506142   0.5    341.9
## M21    66.405208   119.958531   0.1    997.2
## M22    58.258929    81.022176   0.3    713.5
## M23   487.505755   371.763417  65.8   2499.1
## M24   113.278400   100.123299   0.3    446.4
## M25    24.225000    20.712659   0.4    171.8
## M26    28.312879    36.019499   2.0    374.6
## M27    63.185366   116.019643   0.4   1062.5
## M28   217.136975  1407.732325   0.1  14787.1
## M29   269.003650   662.242177   4.8   4719.0
## M30    92.890580   178.688171   2.2   1156.8
## M31    70.592806   164.529466   0.2   1336.6
## M32   174.485714   158.055160   0.6    874.2
## M33   351.807143   216.233798  26.8   1127.6
## M34   147.765000   207.299769   0.1   1605.4
## M35   497.602521  1142.680502   0.1   6596.8
## M36    45.003150    53.001008   0.1    305.9
## M37    85.502857   113.592119   4.3   1026.5
## M38   261.782222   297.515615   9.8   1632.5
## M39    60.212950    87.415025   0.5    913.9
## M40    50.865942    36.291537   0.8    204.3
## M41    84.334746   126.283004   0.1    746.0
## M42   150.777778   116.078149  12.5    810.1
## M43    18.500000    20.303835   4.1    191.7
## M44    67.313971    70.342671   0.3    454.1
## M45  2927.834286  3067.015483  49.9  16673.9
## M46   310.181818   452.698669   1.5   3749.4
## M47   324.726190   352.361974   0.1   1771.3
## M48  9989.246429  6409.779386 988.0  33766.6
## M49   322.085000   236.321259   9.6   1547.1
## M50   353.630435  1481.296362   0.2  17082.2
## M51   466.445385   566.370600   2.1   5732.7
## M52   184.894286   291.663679   9.6   3337.3
## M53   796.820000   912.501386  40.2   6413.9
```

```
## M54     9.919643    14.291967   0.2     82.9
## M55   507.694074  2286.810950   0.4  19704.1
## M56   103.464885    76.869529   2.9    367.9
## M57   680.399275  1598.774326   0.1  16077.3
## M58   579.325899   787.859717  15.6   6371.7
## M59   523.912687  1056.451672  32.8  10448.2
## M60  1910.025581 14293.948084   0.2 160844.7
## M61   727.572662   723.636325   1.7   4920.9
## M62   420.415672   361.756216   0.3   1897.9
## M63   867.850000  1717.830773   3.7  15297.4
## M64   294.537121   475.561297   0.1   5168.5
## M65   533.420000   470.359355  25.6   2432.1
## M66  1745.720714  2193.160792  34.6  16544.5
## M67   163.229630   140.549850   2.5    686.9
## M68   211.005147   217.114020   0.1   1639.0
## M69    72.978676    71.663638   0.1    546.5
## M70    80.413043   107.936982   1.7    598.3
## M71    47.874286    53.697440   4.4    490.4
## M72   139.659091   130.709727   0.3    629.0
## M73    98.432857    64.872548  10.1    366.3
## M74    28.170677    30.352175   0.1    171.0
## M75   165.529286   163.699520  21.6   1225.3
## M76   491.087591   533.048493   0.1   3230.2
## M77   374.370370   298.291067   0.1   1751.5
## M78    25.318182    31.270607   0.1    241.6
## M79   120.472727   391.117521   0.4   3504.4
## M80   676.307914  2499.161985   0.4  27945.1
## M81   684.441429   649.408098   1.2   3849.4
## M82    60.814815   142.964031   0.1   1249.7
## M83   601.870290  1200.734256   0.1   8918.0
## M84   110.000000   131.132886   0.1    813.0
## M85    49.265441    49.840569   0.1    376.0
## M86    94.947857   111.204549   4.7    959.3
## M87    38.300000    35.821601   2.5    213.7
## M88   187.600719   136.312047   0.5    798.8
## M89   710.082143   783.637680  47.4   6317.1
## M90   108.087143   130.038949   2.8    748.7
## M91    71.000000    54.450781   4.6    337.5
## M92    24.986325    29.539471   0.1    169.4
## M93    80.090000    65.765358   5.6    505.6
## M94   431.653957   483.411465  12.9   2624.2
## M95    57.425962    52.660276   0.7    305.5
## M96    10.374167     5.450503   0.1     32.6
## M97    13.296000     9.054614   0.7     82.0
## M98    76.033577   128.051499   0.6   1257.3
## M99    73.641176    92.346760   0.8    737.2
## M100  372.451563   757.248668   0.1   5188.2
## M101   43.211594    50.494757   0.1    428.8
## M102  144.626984   115.508403   0.1    636.2
## M103  249.864179   268.486833   2.6   1482.8
```

```
## M104  322.674820   251.915731   0.2   1579.2
## M105  173.642647   323.266905   0.1   2182.2
## M106   72.784173    75.079148   5.3    598.7
## M107  356.076429   311.605044   0.4   2073.8
## M108   56.526812   125.749143   0.5   1397.2
## M109  121.052518   184.487365   0.1   1045.7
## M110   81.154198    60.371135   2.3    359.1
## M111  126.861940   595.453329   0.2   6373.7
## M112  253.036429   232.781326  18.8   1579.1
## M113  285.835780   389.108870   0.2   2078.8
## M114  170.485714   181.584703   1.0   1143.4
## M115  199.189781   274.185645   5.8   2134.5
## M116   36.179856    44.104210   0.5    318.7
## M117  283.700000   330.004793   7.9   1613.1
## M118  249.646043   264.252481   4.6   1434.2
## M119  101.496429    73.669542  10.3    526.8
## M120   94.751111   100.699122   0.2    934.9
## M121   18.896460    28.415683   0.5    217.2
## M122   26.612409    25.247077   1.0    202.2
## M123  204.233083   194.638812   4.8   1418.2
## M124  539.325926   509.599256   3.3   3789.7
## M125  263.210000   242.458999  12.4   1619.1
## M126   48.964706    64.435119   1.9    609.4
## M127  118.885455   161.193324   0.1    786.2
## M128  511.052344   824.699401   0.1   5959.6
## M129 1825.710000  1182.979472 133.3   8038.2
## M130   63.730000   127.981835   0.2   1188.7
## M131 1952.384286  1593.305452  76.6   8348.6
## M132  165.654286   283.888698  21.4   3155.4
## M133  104.902857   205.187084   4.1   1894.5
## M134 1254.829710  1486.712268  64.6   8567.8
## M135 3076.526429  6833.187519   0.9  53432.0
## M136  239.513889   596.173050   0.1   5014.9
## M137 1308.520588  1830.938936   0.4  12900.8
## M138  955.661871  1195.593433   2.1   6344.9
## M139  292.305556   825.077903   0.1   5569.5
## M140  248.693333   792.729100   0.1   6960.3
## M141  239.773913   671.306304   3.0   6173.9
## M142   22.079137    35.298734   0.1    282.9
## M143  737.940580  1515.224427   0.2   8413.2
## M144   41.966429   105.764076  17.0   1251.4
## M145  195.785981   737.026741   0.1   5479.2
## M146  116.432836   465.680093   0.1   4791.5
## M147  178.897826   141.864057  12.0    840.2
## M148  329.830435   356.819685   1.0   2560.3
## M149  179.732857    96.985829  22.1    502.5
```

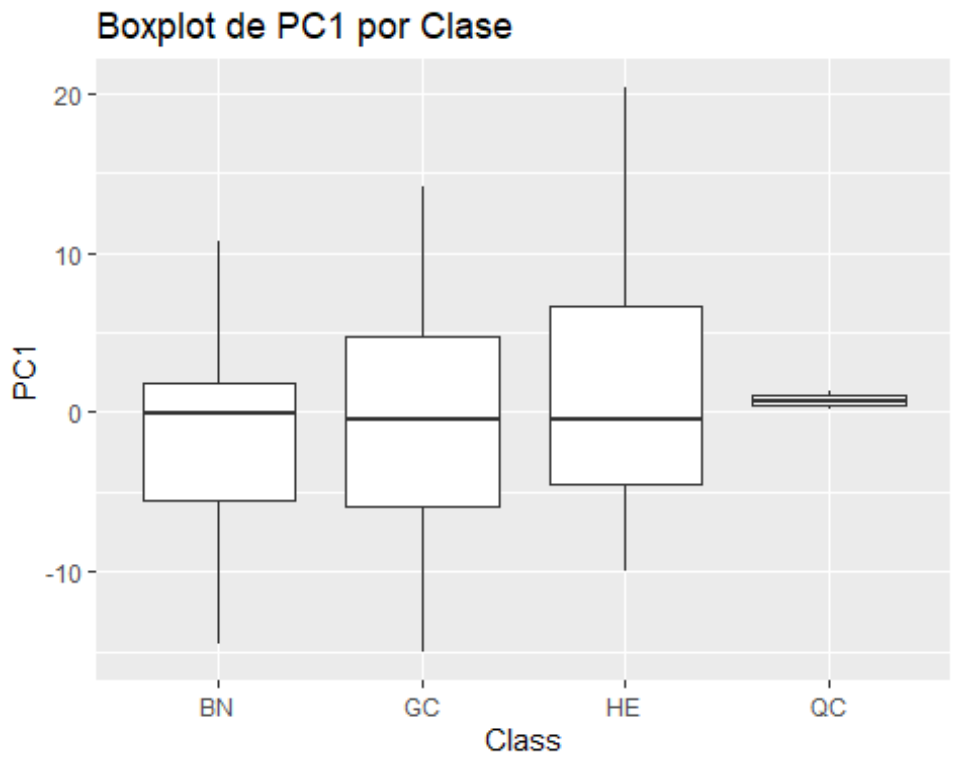
Tambien hacemos un análisis de componentes principales, PCA. Para ver mejor como se agrupan los datos.

```r
library(ggplot2)

data_matrix <- assay(se_object)
data_matrix[is.na(data_matrix)] <- 0
assay(se_object) <- data_matrix
normalized_data <- log1p(assay(se_object))
pca_result <- prcomp(normalized_data, center = TRUE, scale. = TRUE)
pca_data <- as.data.frame(pca_result$x)
pca_data$Class <- rowData(se_object)$Class

ggplot(pca_data, aes(x = PC1, y = PC2, color = Class)) +
    geom_point(size = 3) +
    labs(title = "PCA de Metabolitos", x = "Componente Principal 1", y = "Componente Principal 2") +
    theme_minimal() +
    scale_color_manual(values = c("red", "blue", "green", "purple"))
```



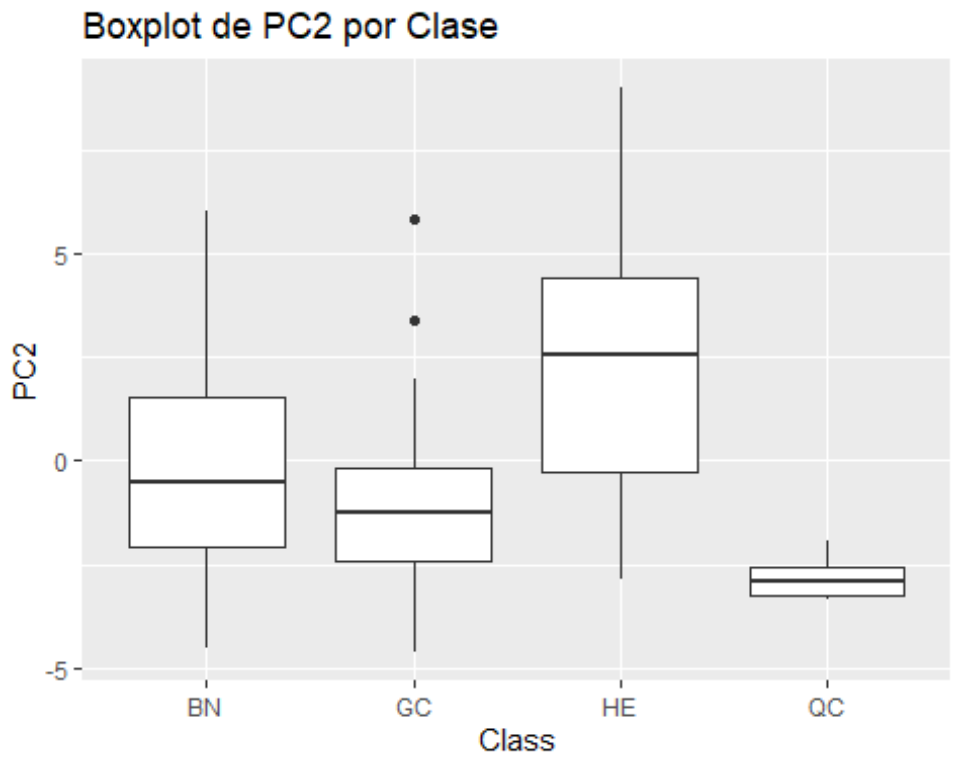
De igual manera podemos generar un diagrama de caja que nos ayuda a visualizar las diferencias entre clases. Hacemos uno para el componente principal 1.

```r
ggplot(data = pca_data, aes(x = Class, y = PC1)) +
    geom_boxplot() +
    labs(title = "Boxplot de PC1 por Clase")
```



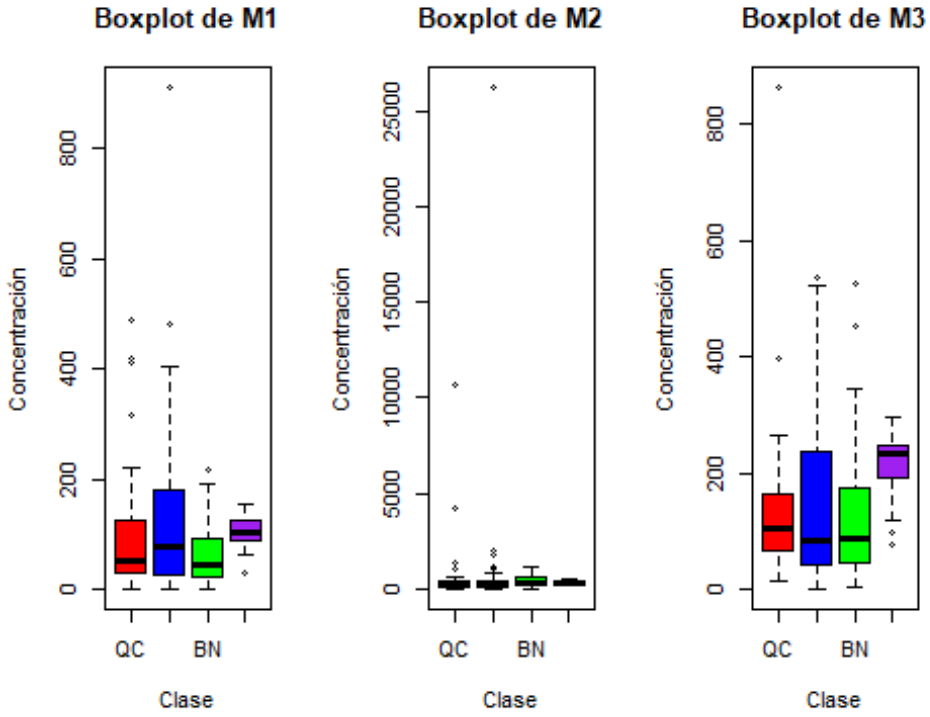
Y posteriormente hacemos uno del componente principal 2. Y esto se puede repetir con todos los PC.

```
ggplot(data = pca_data, aes(x = Class, y = PC2)) +
    geom_boxplot() +
    labs(title = "Boxplot de PC2 por Clase")
```



Podemos realizar tambien una comparación entre metabolitos, aqui estamos comprando el M1, M2 y M3, pero se puede comparar cualquier combinación.

```
selected_metabolites <- c("M1", "M2", "M3")

par(mfrow = c(1, length(selected_metabolites)))
for (metabolite in selected_metabolites) {
  boxplot(data_combined[[metabolite]] ~ data_combined$Class,
          main = paste("Boxplot de", metabolite),
          xlab = "Clase",
          ylab = "Concentración",
          col = c("red", "blue", "green", "purple"),
          names = unique_classes)
}
```


Boxplot de M1
Concentración
0
200
400
600
800
QC
BN
Clase
Boxplot de M2
Concentración
0
5000
10000
15000
20000
25000
QC
BN
Clase
Boxplot de M3
Concentración
0
200
400
600
800
QC
BN
Clase