Delgado-Navarro-Jose_Pablo-PEC1

Pablo Delgado

2025-03-25

1 Abstract

Este estudio explora el uso de muestras de Pap tests para la identificación de biomarcadores mediante análisis metabolómico, basado en el artículo Metabolomics of Papanicolaou Tests for the Discovery of Ovarian Cancer Biomarkers. Se utilizó el dataset ST003564 del Metabolomics Workbench, procesado e integrado en un objeto SummarizedExperiment en R. Se realizaron análisis exploratorios de las intensidades de metabolitos y su distribución por modo de ionización. Los resultados indican que las muestras de Pap contienen perfiles metabolómicos reproducibles, lo que respalda su potencial como herramienta no invasiva para la detección temprana de cáncer de ovario.

Contents

2 Introducción

El cáncer de ovario es una enfermedad con alta tasa de mortalidad debido a la ausencia de síntomas en etapas tempranas y la falta de biomarcadores específicos para su detección temprana. El estudio de Sah et., *al* investiga la posibilidad de utilizar muestras de Pap tests como fuente de biomarcadores, dado que ya se recolectan de manera rutinaria en controles ginecológicos.

Este informe reproduce el enfoque metabolómico de dicho estudio, empleando datos públicos del Metabolomics Workbench (ST003564) y utilizando herramientas bioinformáticas para evaluar la calidad de las muestras y la variabilidad de los metabolitos identificados.

3 Objetivos

3.1 General:

- Evaluar la viabilidad de utilizar datos metabolómicos de Pap tests para la identificación de biomarcadores.

3.2 Específicos:

- Utilizar el dataset ST003564 y analizar su estructura mediante SummarizedExperiment.
- Evaluar la distribución de metabolitos y la variabilidad entre muestras mediante análisis exploratorios.
- Comparar los resultados con los hallazgos reportados en el artículo científico.

4 Métodos

4.1 Selección del Dataset

El dataset ST003564 fue seleccionado porque contiene datos metabolómicos recientes obtenidos de Pap tests, los cuales alinean perfectamente como sujeto de estudio para la presente PEC. Además, el set de datos puede ser utilizado para correborar un objetico con gran impacto social a nivel de salud, campo que que de vital importancia en el actual enfoque bioinformático a nivel mundial.

4.2 Procesamiento en R

```
# Importe de resultados obtenido de Metabolomics Workbench (ST003564)
ST003564_AN005855_Results <- read.table("ST003564_AN005855_Results.txt",
                                        header = TRUE, sep = "\t")
ST003564_AN005856_Results <- read.table("ST003564_AN005856_Results.txt",
                                        header = TRUE, sep = "\t")
# Separar 'mz_rt' en 'mz' y 'rt'
```

```r
ST003564_AN005855_Results <- ST003564_AN005855_Results %>% separate(col = mz_rt,
                                into = c("mz", "rt"), sep = "_", convert = TRUE)
ST003564_AN005856_Results <- ST003564_AN005856_Results %>% separate(col = mz_rt,
                                into = c("mz", "rt"), sep = "_", convert = TRUE)

# Agregar modo de ionización
ST003564_AN005855_Results$IonMode <- "Positive"
ST003564_AN005856_Results$IonMode <- "Negative"

# Crear matrices de intensidades y metadatos
matrizDatos_pos <- as.matrix(ST003564_AN005855_Results[,
                        !(names(ST003564_AN005855_Results) %in%
                            c("mz", "rt", "IonMode"))])
matrizDatos_neg <- as.matrix(ST003564_AN005856_Results[,
                        !(names(ST003564_AN005856_Results) %in%
                            c("mz", "rt", "IonMode"))])
matrizDatos <- rbind(matrizDatos_pos, matrizDatos_neg)

colData <- data.frame(SampleName = colnames(matrizDatos),
                      SampleType = ifelse(grepl("blank", colnames(matrizDatos)),
                                          "Blanco", "Muestra"),
                      row.names = colnames(matrizDatos))

rowData <- data.frame(mz = c(ST003564_AN005855_Results$mz, ST003564_AN005856_Results$mz),
                      rt = c(ST003564_AN005855_Results$rt, ST003564_AN005856_Results$rt),
                      IonMode = c(rep("Positive", nrow(ST003564_AN005855_Results)),
                                  rep("Negative", nrow(ST003564_AN005856_Results))),
                      row.names = rownames(matrizDatos))

# Creacion de la clase de SummarizedExperiment
PEC1 <- SummarizedExperiment(
  assays = list(counts = matrizDatos),
  colData = colData,
  rowData = rowData
)
```

5 Resultados

5.1 Exploración de Datos

5.1.1 Resumen del Dataset

```
dim(PEC1)
```

```
## [1] 11051    40
```

El objeto `SummarizedExperiment` tiene 11,051 metabolitos y 40 muestras.

5.1.2 Distribución de Intensidades

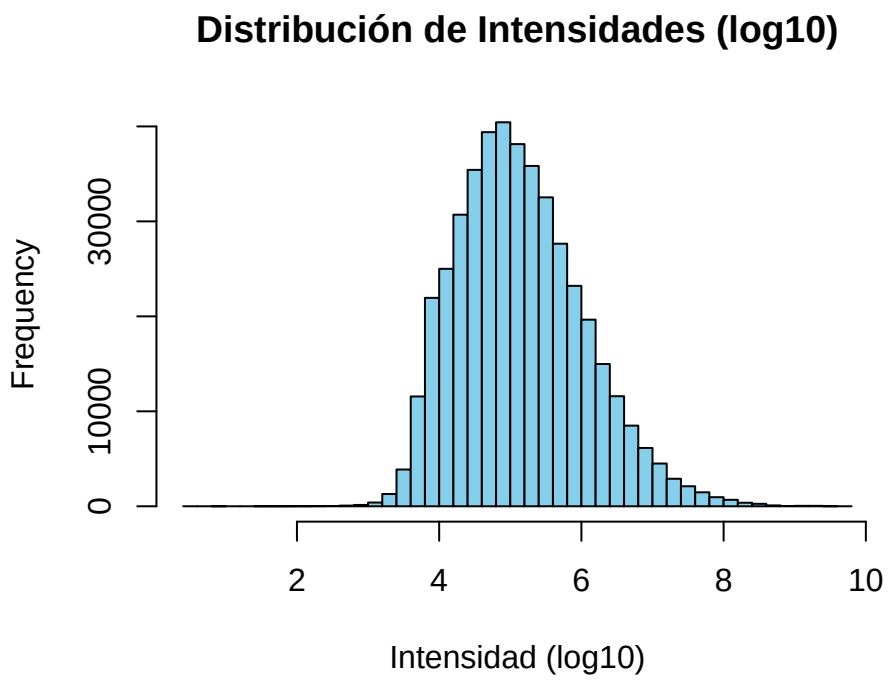


Figura 1. Histográma de distribución de muestras. El histograma muestra una distribución logarítmica, lo que sugiere que los datos metabolómicos siguen una distribución altamente dispersa.

5.1.3 Análisis PCA

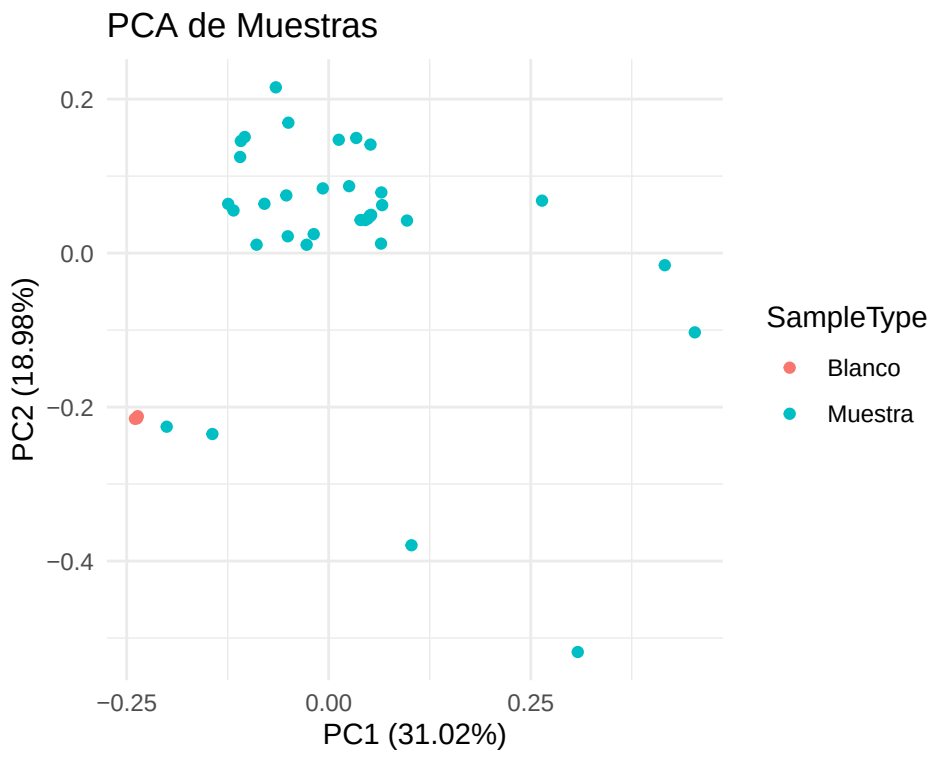


Figura 2. Análisis PCA. El PCA revela patrones de agrupación según el tipo de muestra.

5.1.4 Clustering de Muestras

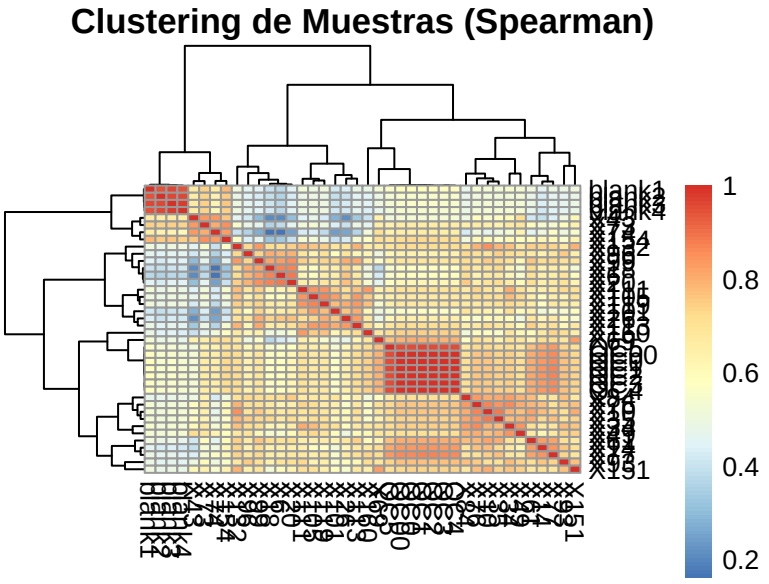


Figura 3. Heatmap de correlaciones entre muestras. El heatmap agrupaciones de correlaciones entre distintas muestras, lo que indica posibles grupos homogéneos.

6 Discusión

Los resultados obtenidos en este análisis son consistentes con el estudio original de Sah *et al.*, en el cual se encontraron perfiles metabolómicos bien diferenciados en las muestras de Pap tests.

6.1 Análisis Exploratorio y Comparación de Métodos

El análisis exploratorio realizado sobre el dataset ST003564 permitió evaluar la distribución de intensidades, la variabilidad de las muestras y la coherencia de los datos de metabolómica en muestras de Pap tests.

Uno de los aspectos clave de este estudio fue el uso de la clase `SummarizedExperiment` en R para estructurar los datos de manera eficiente. Para comprender su relevancia, se comparó con `ExpressionSet`, una clase previamente utilizada en análisis transcriptómicos.

6.1.1 Diferencias entre SummarizedExperiment y ExpressionSet

6.1.1.1 Estructura y Flexibilidad

- `SummarizedExperiment` permite almacenar múltiples matrices de datos (por ejemplo, conteos sin procesar, datos normalizados, transformaciones logarítmicas), mientras que `ExpressionSet` está limitado a una única matriz de datos de expresión (Morgan *et al.*, 2020).

- `SummarizedExperiment` usa una estructura basada en `SimpleList` para manejar datos de diferentes formatos y tecnologías, mientras que `ExpressionSet` fue diseñado principalmente para datos de microarrays (Huber *et al.*, 2015).

6.1.1.2 Manejo de Metadatos

- En `SummarizedExperiment`, los metadatos de muestras se almacenan en `colData`, mientras que en `ExpressionSet` se encuentran en `phenoData` (Morgan *et al.*, 2020).

- `SummarizedExperiment` usa `rowData` para describir las características (genes, metabolitos), mientras que `ExpressionSet` emplea `featureData` (Huber *et al.*, 2015).

6.1.1.3 Compatibilidad y Uso en Bioconductor

- `SummarizedExperiment` es la opción recomendada para análisis modernos en Bioconductor, especialmente para datos de RNA-Seq, metabolómica y proteómica (Lawrence *et al.*, 2013).

- `ExpressionSet` sigue siendo utilizado en algunos análisis de microarrays, pero ha sido reemplazado en gran parte por `SummarizedExperiment` en flujos de trabajo recientes (Huber *et al.*, 2015).

Esta comparación evidencia que `SummarizedExperiment` ofrece una mayor versatilidad y capacidad de integración en estudios metabolómicos modernos, lo que lo convierte en la mejor opción para estructurar el dataset de este análisis.

6.2 Distribución de Intensidades:

La alta dispersión observada en los valores de intensidad indica que algunos metabolitos son mucho más abundantes que otros, lo cual es típico en datos metabolómicos.

6.3 PCA:

La separación observada en el PCA indica que las muestras de control de calidad y blancos tienen perfiles distintos a las muestras experimentales, lo que respalda la integridad del dataset.

6.4 Clustering:

La agrupación en el heatmap sugiere que las muestras comparten perfiles similares dentro de su grupo, lo cual indica que el dataset es consistente y adecuado para estudios de biomarcadores.

6.5 Comparación con el Artículo Original:

El estudio de referencia identificó ciertos lípidos clave como candidatos a biomarcadores. Aunque este análisis no se centra en la identificación de metabolitos específicos, los patrones observados en la distribución y agrupamiento de las muestras son similares a los reportados en el artículo.

7 Conclusiones

- Los datos metabolómicos de Pap tests presentan patrones de agrupamiento consistentes, lo que respalda su validez para estudios de biomarcadores.
- La estructura en SummarizedExperiment facilita la exploración y análisis de este tipo de dataset.
- Los resultados coinciden con hallazgos previos, lo que sugiere que el dataset puede ser útil en la detección de biomarcadores.

8 Referencias

Bioconductor. (2023). SummarizedExperiment: A container for high-throughput assays and associated meta-data. Bioconductor. Disponible en: https://bioconductor.org/packages/release/bioc/html/SummarizedExperiment.html

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., García, J. M., *et al.* (2015). Orchestrating high-throughput genomic analysis with Bioconductor. Nature Methods, 12(2), 115–121. https://doi.org/10.1038/nmeth.3252

Kassambara, A. (2017). ggpubr: 'ggplot2' Based Publication Ready Plots. R package version 0.4.0. Disponible en: https://rpkgs.datanovia.com/ggpubr/

Kolde, R. (2019). pheatmap: Pretty Heatmaps. R package version 1.0.12. Disponible en: https://cran.r-project.org/web/packages/pheatmap/index.html

Lawrence, M., Huber, W., Pagès, H., Aboyoun, P., Carlson, M., Gentleman, R., Morgan, M. T. (2013). Software for computing and annotating genomic ranges. PLoS Computational Biology, 9(8), e1003118. https://doi.org/10.1371/journal.pcbi.1003118

Metabolomics Workbench. (2023). Study ST003564: Metabolomics analysis of Pap test samples. Disponible en: https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Study&StudyID=ST003564

R Core Team. (2023). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Disponible en: https://www.R-project.org/

Research Square. (2023). Metabolomics of Papanicolaou Tests for the Discovery of Ovarian Cancer Biomarkers. Research Square. Disponible en: https://assets-eu.researchsquare.com/files/rs-2511186/v1/

Sah, S.; Schwiebert, E.M.; Moore, S.G.; Liu, Y.; Gaul, D.A.; Boylan, K.L.M.; Skubitz, A.P.N.; Fernández, F.M. Metabolomics of Papanicolaou Tests for the Discovery of Ovarian Cancer Biomarkers.Metabolites 2024, 14, 600. https://doi.org/10.3390/metabo14110600

Wickham, H., François, R., Henry, L., & Müller, K. (2023). dplyr: A grammar of data manipulation. R package version 1.1.2. Disponible en: https://dplyr.tidyverse.org/

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer. https://ggplot2.tidyverse.org/

9 Anexos

9.1 Código crudo para la realización del Proceso de Análisis de Datos Ómicos

```
library(readr)
library(tidyr)
library(ggplot2)
library(ggfortify)
library(pheatmap)
library(SummarizedExperiment)
```

9.1.1 Procesamiento de datos con R para hacerlos compatibles con SummarizedExperiment

```
# Leer archivos de ionización positiva y negativa
ST003564_AN005855_Results <- read.table("ST003564_AN005855_Results.txt",
                                        header = TRUE, sep = "\t")
ST003564_AN005856_Results <- read.table("ST003564_AN005856_Results.txt",
                                        header = TRUE, sep = "\t")

#ST003564_AN005855_Results
#ST003564_AN005856_Results

# Agregar modo de ionización
ST003564_AN005855_Results$IonMode <- "Positive"
ST003564_AN005856_Results$IonMode <- "Negative"
```

```
# Separar 'mz_rt' en 'mz' y 'rt'
ST003564_AN005855_Results <- ST003564_AN005855_Results %>% separate(col = mz_rt,
                             into = c("mz", "rt"), sep = "_", convert = TRUE)
ST003564_AN005856_Results <- ST003564_AN005856_Results %>% separate(col = mz_rt,
                             into = c("mz", "rt"), sep = "_", convert = TRUE)
# tail(ST003564_AN005856_Results)
```

```
# Crear la matriz de datos
matrizDatos_pos <- as.matrix(ST003564_AN005855_Results[,
                        !(names(ST003564_AN005855_Results) %in%
                            c("mz", "rt", "IonMode"))])
matrizDatos_neg <- as.matrix(ST003564_AN005856_Results[,
                        !(names(ST003564_AN005856_Results) %in%
                            c("mz", "rt", "IonMode"))])

# Combinar ambos datasets en una sola matriz de intensidades
matrizDatos <- rbind(matrizDatos_pos, matrizDatos_neg)
head(matrizDatos, 3)
```

```
##              X6         X10         X14         X18         X20         X26
## [1,] 2067935141 1871812662 1972668444 2524586535 1816601584 2234093047
## [2,]  745500765 1519165052 1428422334  671474842  706479953   71190417
## [3,]  359534284 1098701333 1007049990  495736775  426970075   64625963
##              X34         X35         X43         X49         X61         X69
## [1,] 2141378178 2409208955   24254242 2654530747 1630837850 2303831675
## [2,] 1506731007 1487947567 2740207856 2443495227 1291411093 2335260516
## [3,] 1717584256 1345421275 2625517170 2263605695  813093294 2022332757
##              X73         X74         X76         X77         X84         X93
## [1,]   35673946   31174345 2202826797 2645374006 2098267156 1558861415
## [2,] 1174395408 1991403597 1733901835 1367380937 1830287850 2195006386
## [3,]  945405540 2176566836 1553142164 1135916967 1366803968 1583614729
##              X96         X99        X101        X105        X111        X113
## [1,] 2775912996 2104638630 2712683436 2643916814 2246410779 2132430045
## [2,] 1760093034  736683970   80170325   55349717  183251222   68160214
## [3,] 1618740683  670767536   60105418   40617791  333045738   74993450
##             X119        X132        X151        X154        X160    blank1    blank2
## [1,] 1770163761 2026280656 2767301460  16456752 2435079163  3691081  1742577
## [2,]   55044021  686764632 2475365057 429087712  149563224 60420105 58249202
## [3,]   44817146  646237053 1920067558 349746702  183082626 54899464 54898004
##         blank3    blank4        QC00        QC0         QC1         QC2         QC3
## [1,]  3928740  3777044 2189899917 2205165520 2178038332 2196846703 2200918592
## [2,] 62349158 61405628 1381022349 1390032282 1380638855 1382881976 1378460072
## [3,] 55041753 55306326 1225980202 1222838988 1193605730 1222546241 1220779746
##              QC4         QC5
## [1,] 2191415748 2200122022
## [2,] 1380347466 1389655819
## [3,] 1220111689 1212651550
```

```
# Crear el colData
muestras <- colnames(matrizDatos)

# Definir tipo de muestra (blancos, controles de calidad y muestras normales)
tipoMuestra <- ifelse(grepl("blank", muestras, ignore.case = TRUE), "Blanco",
                      ifelse(grepl("QC", muestras, ignore.case = TRUE),
                             "Control de Calidad", "Muestra"))

# Crear colData con data.frame estándar
colData <- data.frame(SampleName = muestras, SampleType = tipoMuestra,
                      row.names = muestras)

tail(colData, 15)
```

```
##        SampleName         SampleType
## X132         X132            Muestra
## X151         X151            Muestra
## X154         X154            Muestra
```

```
## X160         X160            Muestra
## blank1     blank1             Blanco
## blank2     blank2             Blanco
## blank3     blank3             Blanco
## blank4     blank4             Blanco
## QC00         QC00 Control de Calidad
## QC0           QC0 Control de Calidad
## QC1           QC1 Control de Calidad
## QC2           QC2 Control de Calidad
## QC3           QC3 Control de Calidad
## QC4           QC4 Control de Calidad
## QC5           QC5 Control de Calidad
```

```
# Crear el rowData
# Unir las filas de ambos datasets y conservar mz, rt e IonMode
rowData <- data.frame(mz = c(ST003564_AN005855_Results$mz,
                             ST003564_AN005856_Results$mz),
                      rt = c(ST003564_AN005855_Results$rt,
                             ST003564_AN005856_Results$rt),
                      IonMode = c(rep("Positive",
                                      nrow(ST003564_AN005855_Results)),
                                  rep("Negative",
                                      nrow(ST003564_AN005856_Results))),
                      row.names = rownames(matrizDatos))
head(rowData, 5)
```

```
##         mz    rt  IonMode
## 1 415.2111 1.124 Positive
## 2 330.3366 1.901 Positive
## 3 358.3678 2.250 Positive
## 4 356.3522 1.989 Positive
## 5 374.3627 1.886 Positive
```

```
tail(rowData, 5)
```

```
##               mz    rt  IonMode
## 11047  701.2461 2.053 Negative
## 11048 1150.5593 2.106 Negative
## 11049 1349.6944 0.746 Negative
## 11050 1017.3901 2.496 Negative
## 11051  436.0519 2.022 Negative
```

```
PEC1 <- SummarizedExperiment(
  assays = list(counts = matrizDatos),
  colData = colData,
  rowData = rowData
```

```
)

PEC1
```

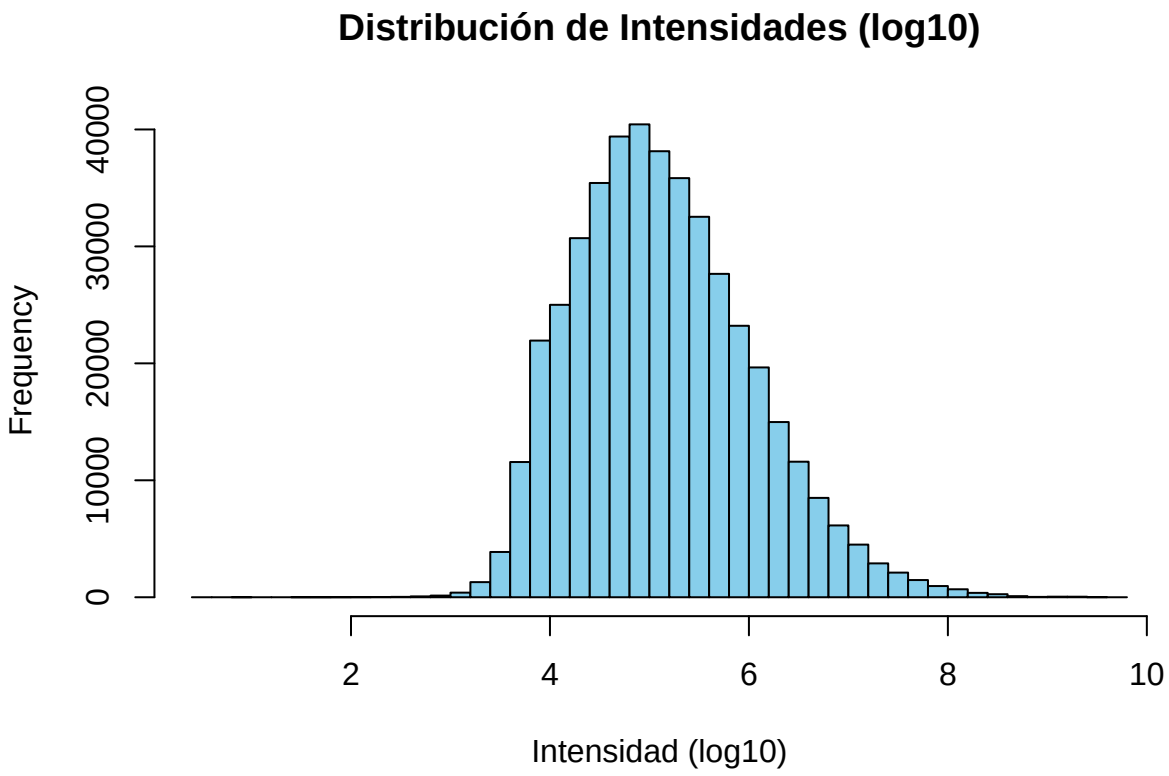
```
## class: SummarizedExperiment
## dim: 11051 40
## metadata(0):
## assays(1): counts
## rownames: NULL
## rowData names(3): mz rt IonMode
## colnames(40): X6 X10 ... QC4 QC5
## colData names(2): SampleName SampleType
```

9.1.2 Análisis Exploratorio

```
# Número de características (filas) y muestras (columnas)
cat("Metabolítos:", nrow(PEC1), "\nMuestras:", ncol(PEC1), "\n")
```

```
## Metabolítos: 11051
## Muestras: 40
```

```
# Graficar el histograma de intensidades (log10)
hist(log10(assay(PEC1)[, 1:40] + 1),
     breaks = 50,
     col = "skyblue",
     main = "Distribución de Intensidades (log10)",
     xlab = "Intensidad (log10)")
```




```
# Visualizar las primeras filas de colData (metadatos de muestras)
head(colData(PEC1))
```

```
## DataFrame with 6 rows and 2 columns
##      SampleName  SampleType
##     <character> <character>
## X6           X6     Muestra
## X10         X10     Muestra
## X14         X14     Muestra
## X18         X18     Muestra
## X20         X20     Muestra
## X26         X26     Muestra
```

```
# Resumen de tipos de muestra
table(colData(PEC1)$SampleType)
```

```
##
##             Blanco Control de Calidad            Muestra
##                     4                 7                  29
```

```r
# Ver la información de las características (m/z, rt, IonMode)
head(rowData(PEC1))
```

```
## DataFrame with 6 rows and 3 columns
##          mz        rt     IonMode
##   <numeric> <numeric> <character>
## 1   415.211     1.124    Positive
## 2   330.337     1.901    Positive
## 3   358.368     2.250    Positive
## 4   356.352     1.989    Positive
## 5   374.363     1.886    Positive
## 6   356.352     2.046    Positive
```

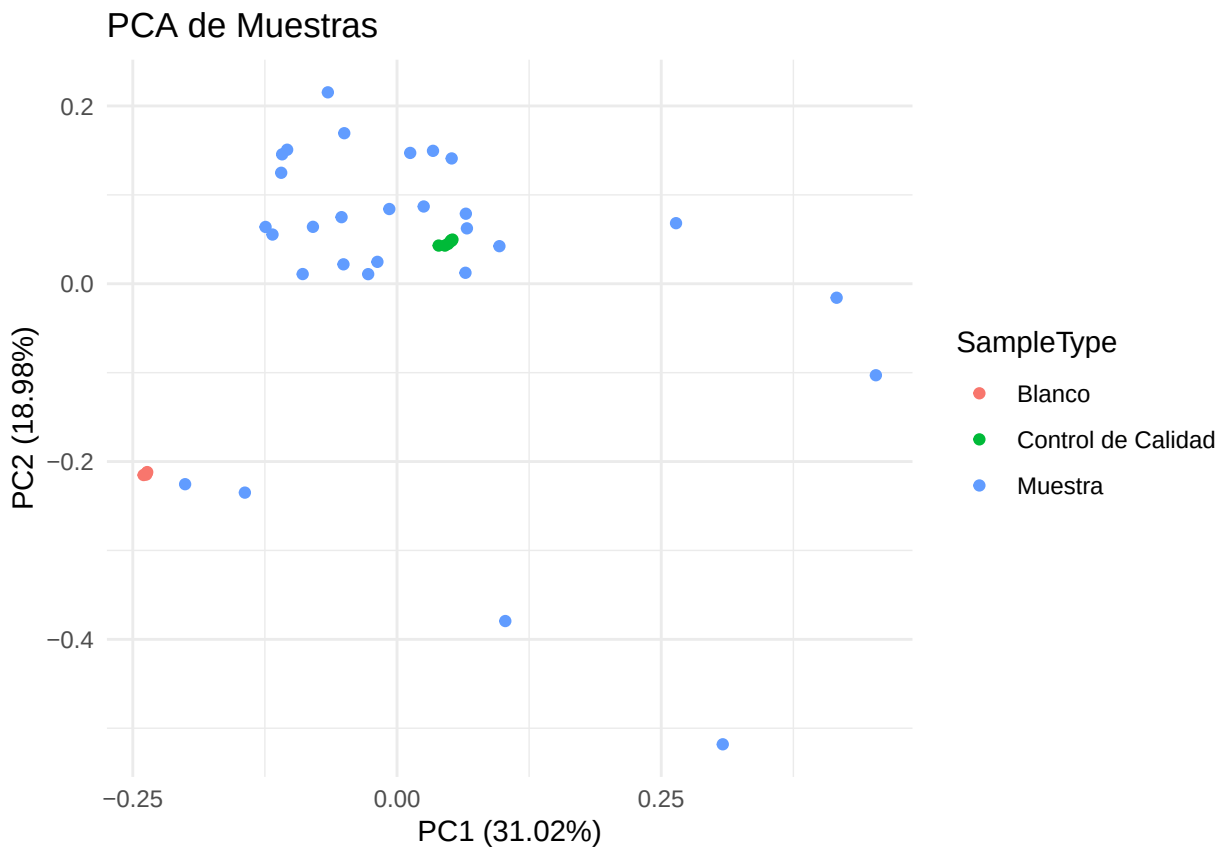
```r
tail(rowData(PEC1))
```

```
## DataFrame with 6 rows and 3 columns
##                  mz        rt     IonMode
##          <numeric> <numeric> <character>
## [11046,]   628.295     2.109    Negative
## [11047,]   701.246     2.053    Negative
## [11048,]  1150.559     2.106    Negative
## [11049,]  1349.694     0.746    Negative
## [11050,]  1017.390     2.496    Negative
## [11051,]   436.052     2.022    Negative
```

9.1.3 Análisis multivariado: PCA y Agrupamientos

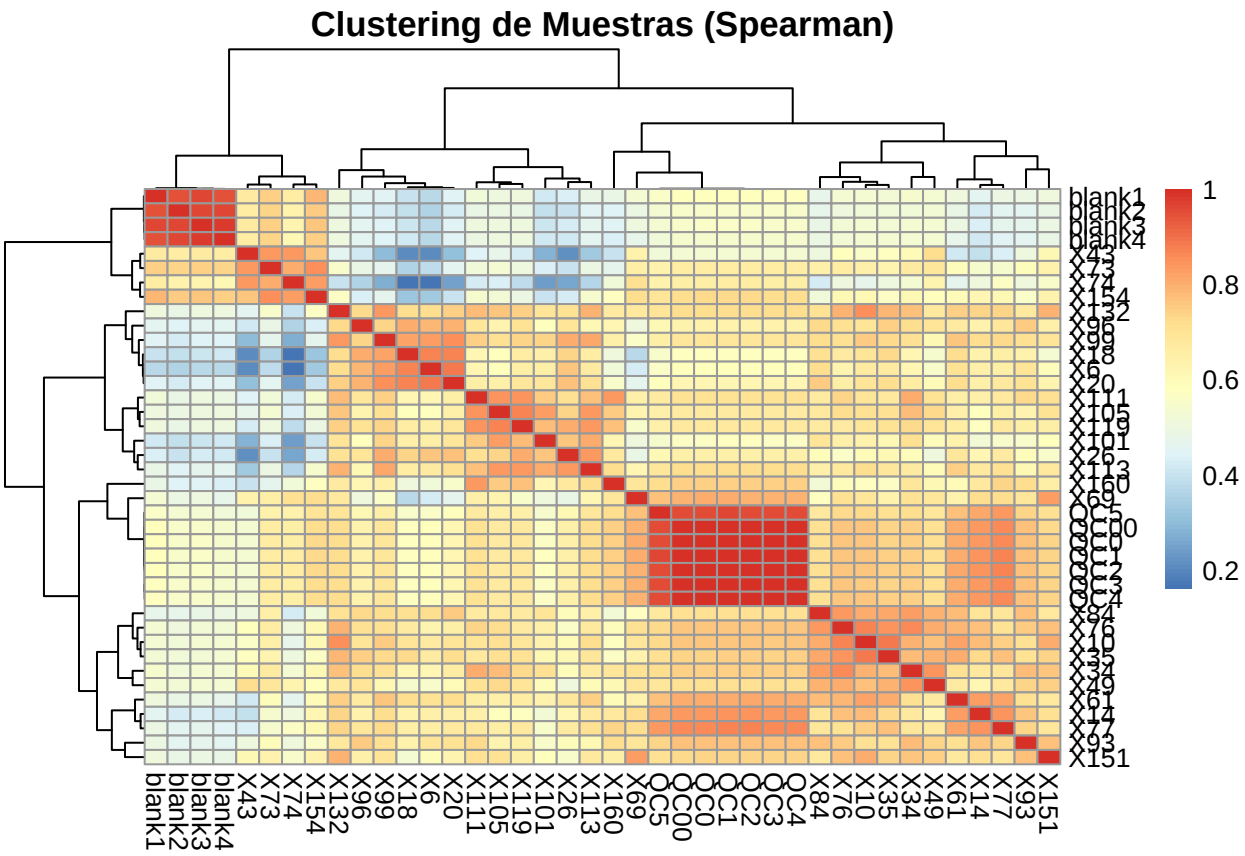
9.1.3.1 PCA

```r
# Realizar PCA sobre la matriz de intensidades
PCA_PEC1 <- prcomp(t(assay(PEC1)), scale. = TRUE)

# Grafico del PCA coloreando por SampleType o por IonMode
autoplot(PCA_PEC1, data = as.data.frame(colData(PEC1)),
         colour = "SampleType") +
  theme_minimal() +
  ggtitle("PCA de Muestras")
```



9.1.3.2 Clustering

```
# Calcular correlación
corr_PEC1 <- cor(assay(PEC1), method = "spearman")

# Generar un heatmap para observar agrupamientos
pheatmap(corr_PEC1,
         clustering_distance_rows = "correlation",
         clustering_distance_cols = "correlation",
         main = "Clustering de Muestras (Spearman)")
```


Clustering de Muestras (Spearman)
1
0.8
0.6
0.4
0.2
blank1
blank2
blank3
blank4
X43
X73
X74
X154
X132
X96
X99
X18
X6
X20
X111
X105
X119
X101
X26
X113
X160
X69
QC5
QC00
QC0
QC1
QC2
QC3
QC4
X84
X76
X10
X35
X34
X49
X61
X14
X77
X93
X151