PEC1_Informe

Ivana Hermanova

2025-04-02

Contents

1 Resumen

El objetivo de esta actividad es explorar un conjunto de datos metabolómicos, crear un objeto *Summarized-Experiment* (**se**) y guardar los datos, el código R y los resultados en un repositorio en GitHub.

Se seleccionó un conjunto de datos que contenía niveles metabolómicos en plasma de pacientes sometidos a biopsia prostática, junto con registros clínicos detallados. A algunos de estos pacientes se les diagnosticó posteriormente cáncer de próstata. La selección del conjunto de datos se basó en la hipótesis de que los niveles plasmáticos de poliaminas podrían funcionar como marcadores pronósticos. En primer lugar, se creó un objeto de clase *SummarizedExperiment* y se realizó un análisis exploratorio. Para abordar la hipótesis, se calcularon los niveles medios de poliaminas en ambos grupos. Aunque la literatura respalda la importancia de las poliaminas en la progresión del cáncer de próstata, en esta cohorte los niveles medios de poliaminas en plasma no mostraron valor pronóstico.

2 Objetivos

El objetivo de esta actividad es seleccionar y descargar un dataset de metabolómica, crear un objeto de clase SummarizedExperiment y llevar a cabo un análisis exploratorio. A continuación, se elaborará un informe que describa el proceso y se creará un repositorio en GitHub.

Para incluir una pregunta biologica, se busca determinar si existen diferencias significativas en las concentraciones de poliaminas en plasma de pacientes sometidos a biopsia prostática, comparando a aquellos que, posteriormente, desarrollan cáncer de próstata con aquellos que no lo hacen. Este estudio podría aportar evidencia sobre el potencial de las poliaminas como biomarcador predictivo en el contexto del cáncer de próstata.

3 Métodos

El dataset se ha descargado de Metabolomics Workbench (https://www.metabolomicsworkbench.org/data/DRCCStudySummary.php?Mode=SetupRawDataDownload&StudyID=ST002498). Para explorar y analizar los archivos se utilizó el entorno R (http://r-project.org), haciendo uso del paquete BiocManager (https://bioconductor.org/install/) y de la librería SummarizedExperiment (SE)(https://bioconductor.org/packages/release/bioc/html/SummarizedExperiment.html). El objeto SE se ha guardado desde R.

Se emplearon diversos comandos en R para realizar un análisis exploratorio que permitió verificar la estructura del objeto SummarizedExperiment (dimensiones, metadatos de las muestras y de los metabolitos) y los datos contenidos en el dataset (número de pacientes, aquellos que han desarrollado cáncer, cáncer metastásico, número de poliaminas medidos). Para evaluar el valor pronóstico de las poliaminas, se calcularon las medias de los niveles de poliaminas en pacientes con diagnóstico de cáncer de próstata en comparación con aquellos sin diagnóstico, utilizando el test no paramétrico de Wilcoxon.

4 Resultados

1.Summarized experiment

El objeto **se** contiene niveles de 1169 metabolitos medidos en 580 muestras. Su único assay, denominado "counts", almacena los niveles de estos metabolitos (s-1-pyrroline-5-carboxylate spermidine, etc), el rowData incorpora 18 variables descriptivas (como CHEMICAL_NAME, CHEM_ID, etc.) que proporcionan información sobre cada metabolito, y el colData contiene 420 variables clínicas y demográficas (por ejemplo, SAMPLE_ID, PreviousPositiveBiopsy, ControlAtypia.RB, ControlAtrophy.RB, etc).

Table 1: Resumen del objeto SummarizedExperiment

Propiedad	Valor
Clase	SummarizedExperiment
Dimensiones	1169 x 580
Número de assays	1
Nombres de assays	counts
Número de variables en rowData	18
Número de variables en colData	420

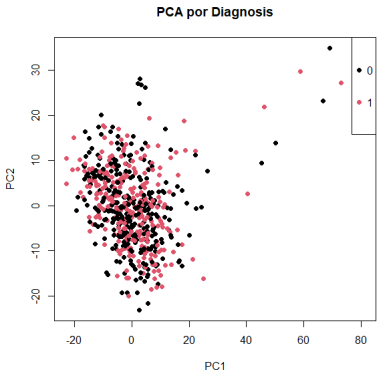
Respuesta para la pregunta de la PEC1: Cuáles son sus principales diferencias con la clase ExpressionSet?

ExpressionSet es una clase más antigua y orientada a microarrays, con una estructura fija que se centra en un único conjunto de datos. **SummarizedExperiment** es más flexible, capaz de manejar múltiples assays, diferentes mediciones, utilizando una estructura de metadatos basada en DataFrame.

2. Analisis exploratorio

2.1 PCA

Para explorar la estructura general y la variabilidad de los datos, se realizó análisis de componentes principales (PCA). El primer gráfico ofrece una visión general de la varianza de los perfiles metabolómicos sin tener en cuenta la clasificación por diagnóstico. Se detectan puntos aislados que podrian considerarse atípicos (outliers). A continuación, se generó un gráfico PCA en el que las muestras individuales se codificaron por colores según el diagnóstico de cáncer de próstata (0 = sin cáncer, 1 = con cáncer). Este segundo gráfico permite evaluar si la variabilidad de los datos está relacionada con el estado diagnóstico. Se observa que las muestras de ambos grupos se solapaan en gran medida, lo que sugiere que los perfiles metabolómicos no difieren significativamente en función del diagnóstico.



2.2 Analisis de la distribucion de pacientes por diagnostico

El dataset contiene datos de 580 pacientes, de los cuales 267 han desarrollado cáncer de próstata (diagnosis=1) y 8 han desarrollado metástasis.

Table 2: **Número de casos con cáncer de próstata**

Cáncer	Pacientes	Metastasis
No	313	0
Si	267	8

2.3 Analisis del valor pronostico de las poliaminas

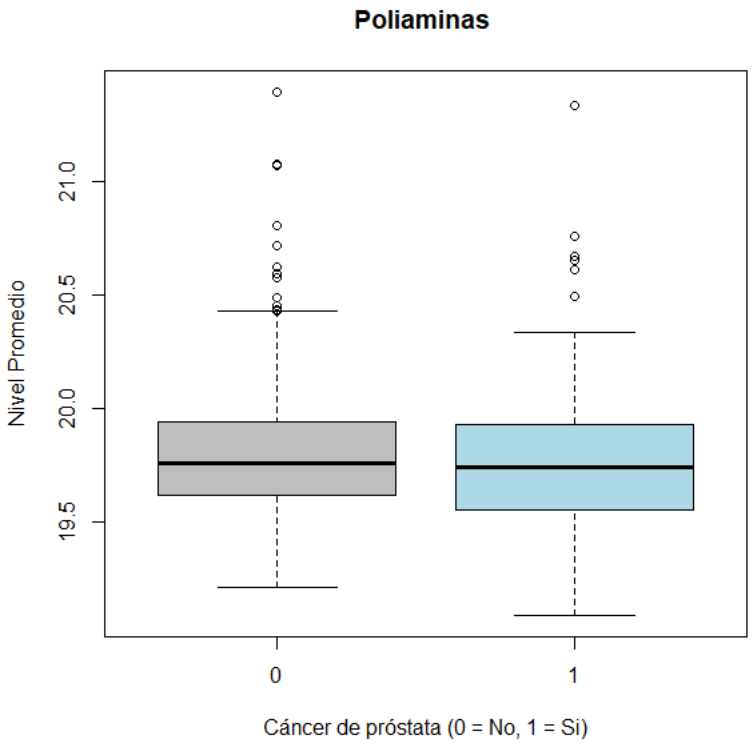
En total, se analizaron 1169 metabolitos, 7 poliaminas (spermidine, N-acetylputrescine, 5-methylthioadenosine (MTA), 4-acetamidobutanoate, acisoga, (N(1) + N(8))-acetylspermidine, N-acetyl-isoputreanine).

Table 3: Nombres de Poliaminas

Poliamina
spermidine
N-acetylputrescine
5-methylthioadenosine (MTA)
4-acetamidobutanoate
acisoga
(N(1) + N(8))-acetylspermidine
N-acetyl-isoputreanine

Se calculó la media de los niveles de poliaminas en grupos de pacientes con y sin diagnóstico de cáncer de

próstata. La media fue de 19.803 en pacientes sin el diagnóstico y de 19.758 en pacientes que han desarrollado cáncer de próstata. El test de Wilcoxon confirmó que no hay diferencia significativa entre ambos grupos. Se concluye que el valor de las poliaminas medidas en plasma no tiene valor pronóstico.



5 Discusión

En este estudio se exploraron los niveles de poliaminas en plasma de pacientes sometidos a biopsia prostática, con el objetivo de evaluar su potencial como biomarcadores pronósticos del cáncer de próstata. Se creó un objeto de clase SummarizedExperiment y, mediante un análisis exploratorio, se verificó la integridad del objeto. Posteriormente, se compararon las medias de los niveles de poliaminas entre pacientes con diagnóstico de cáncer de próstata y aquellos sin diagnóstico, utilizando el test no paramétrico de Wilcoxon. Los resultados mostraron diferencias muy pequeñas (19.803 vs. 19.758) que no alcanzaron significación estadística, lo que sugiere que, en este conjunto de datos, las poliaminas plasmáticas no poseen valor pronóstico.

Es importante considerar que, aunque las poliaminas tienen una relevancia intrínseca en el tejido prostático, sus niveles en plasma pueden no reflejar adecuadamente su actividad local en la próstata. Esta discrepancia podría deberse a la compleja regulación del metabolismo en distintos compartimentos biológicos o a limitaciones en la sensibilidad de la medición plasmática. La mínima diferencia observada podría estar influenciada por la variabilidad intragrupo y por la posible presencia de factores no ajustados, tales como la edad o antecedentes clínicos.

Por otro lado, la literatura sugiere una relación entre las poliaminas y la progresión del cáncer de próstata, lo que plantea la hipótesis de que estos compuestos podrían funcionar como marcadores pronósticos. No obstante, nuestros resultados indican que, al menos en esta cohorte, los niveles plasmáticos de poliaminas no difieren significativamente entre los grupos de diagnóstico.

6 Conclusiones

Se integraron datos metabolómicos, anotación de metabolitos y metadata clínica en un objeto SummarizedExperiment. El análisis exploratorio y el test de Wilcoxon mostraron diferencias mínimas y no significativas en los niveles de poliaminas entre pacientes con y sin cáncer de próstata. En esta cohorte, las poliaminas plasmáticas no resultaron ser un biomarcador pronóstico para el cáncer de próstata.

7 Referencias

1. Conjunto de datos

https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Study&StudyID=ST002498&StudyType=MS&ResultType=1

2. github de la PEC1

https://github.com/Ivanovna66/Hermanova-Ivana-PEC1

3. Poliaminas y cancer

   https://pubmed.ncbi.nlm.nih.gov/40071465/

8 Annexo I-Codigo R

```r
# ------------------------------------------------------------------------------
# Cargamos librerias necesarias
# ------------------------------------------------------------------------------
# Instalamos SummarizedExperiment
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}
if (!require("SummarizedExperiment", character.only = TRUE)) {
  BiocManager::install("SummarizedExperiment")
}
```

```
## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
```

```
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'
```

```
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library(readr) #Para leer lso archivos
library(dplyr)  #Para manipulación de datos
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:Biobase':
##
##     combine

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union

## The following object is masked from 'package:matrixStats':
##
##     count

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(SummarizedExperiment)  #Para trabajar con objetos SummarizedExperiment
library(knitr) #Para las tablas
```

```
## Warning: package 'knitr' was built under R version 4.4.3
```

```
# ------------------------------------------------------------------------------
# 1. Importamos los datos
# ------------------------------------------------------------------------------
preprocessed_data <- read_delim("https://raw.githubusercontent.com/Ivanovna66/Hermanova-Ivana-1/be4c57d
                          delim = "\t",
```

```
                          escape_double = FALSE,
                          trim_ws = TRUE)
```

```
## Rows: 580 Columns: 1170
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr    (1): CLIENT_SAMPLE_ID
## dbl (1169): S-1-pyrroline-5-carboxylate, spermidine, 1-methylnicotinamide, 1...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
metabolite_annotations <- read_delim("https://raw.githubusercontent.com/Ivanovna66/Hermanova-Ivana-1/be
                          delim = "\t",
                          escape_double = FALSE,
                          trim_ws = TRUE)
```

```
## Rows: 1482 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (11): SUPER_PATHWAY, SUB_PATHWAY, TYPE, INCHIKEY, SMILES, CHEMICAL_NAME,...
## dbl  (5): CHEM_ID, LIB_ID, COMP_ID, CHRO_LIB_ENTRY_ID, PATHWAY_SORTORDER
## num  (2): CHEMSPIDER, PUBCHEM
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
clinical_annotations <- read_delim("https://raw.githubusercontent.com/Ivanovna66/Hermanova-Ivana-1/be4c
                          delim = "\t",
                          escape_double = FALSE,
                          trim_ws = TRUE)
```

```
## Rows: 584 Columns: 420
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr  (64): SAMPLE_ID, EthnicityOtherDesc, EmploymentOtherDesc, PreviousCance...
## dbl (333): PreviousPositiveBiopsy, Age, EthnicityLatino, Ethnicity, Employme...
## lgl  (23): RepeatBiopsyDate_2.fu1control, RepeatBiopsyResults_2.fu1control, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# ------------------------------------------------------------------------------
# 2. Preparamos los datos para crear el objeto SummarizedExperiment (SE)
# ------------------------------------------------------------------------------
# 2.1. Creamos la matriz de datos (assay)
# Extraemos las mediciones (todas las columnas excepto CLIENT_SAMPLE_ID) y las convertimos a matriz
assay_data <- preprocessed_data %>% select(-CLIENT_SAMPLE_ID) %>% as.matrix()

# Obtenemos los nombres de los metabolitos (columnas, excepto CLIENT_SAMPLE_ID
metabolite_names <- colnames(preprocessed_data)[-1]

# Transponemos la matriz para que:
# - Las filas sean los metabolitos
# - Las columnas sean las muestras (identificadas por CLIENT_SAMPLE_ID)
```

```r
assay_matrix <- t(assay_data)
rownames(assay_matrix) <- metabolite_names
colnames(assay_matrix) <- preprocessed_data$CLIENT_SAMPLE_ID

# 2.2. Preparamos los metadatos de los features (rowData)
# Creamos un data frame con todos los metabolitos
row_metadata <- metabolite_annotations %>%
  filter(CHEMICAL_NAME %in% metabolite_names) %>%
  arrange(match(CHEMICAL_NAME, metabolite_names)) %>%
  as.data.frame()

# 2.3. Preparamos los metadatos de las muestras (colData)
# Filtramos la metadata clínica para guardar solo las muestras que estan en preprocessed_data
col_metadata <- clinical_annotations %>%
  filter(SAMPLE_ID %in% preprocessed_data$CLIENT_SAMPLE_ID) %>%
  arrange(match(SAMPLE_ID, preprocessed_data$CLIENT_SAMPLE_ID)) %>%
  as.data.frame()
rownames(col_metadata) <- col_metadata$SAMPLE_ID

# 2.4. Comprobamos las dimensiones
dim(assay_matrix)
```

```
## [1] 1169  580
```

```r
dim(row_metadata)
```

```
## [1] 1169   18
```

```r
dim(col_metadata)
```

```
## [1] 580 420
```

```r
# ------------------------------------------------------------------------------
# 3. Creamos el objeto SummarizedExperiment
# ------------------------------------------------------------------------------
se <- SummarizedExperiment(
  assays = list(counts = assay_matrix),
  rowData = row_metadata,
  colData = col_metadata
)

# Revisamos la estructura de SE
print(dim(se))            # Dimensiones: numero de metabolitos  y pacientes
```

```
## [1] 1169  580
```

```r
print(assayNames(se))     # Nombre del assay
```

```
## [1] "counts"
```

```r
head(rownames(se))        # Los primeros nombres de los metabolitos
```

```
## [1] "S-1-pyrroline-5-carboxylate" "spermidine"
## [3] "1-methylnicotinamide"        "12,13-DiHOME"
## [5] "alpha-ketoglutarate"         "kynurenate"
```

```r
head(colnames(se))        # Los primeros ID de las muestras
```

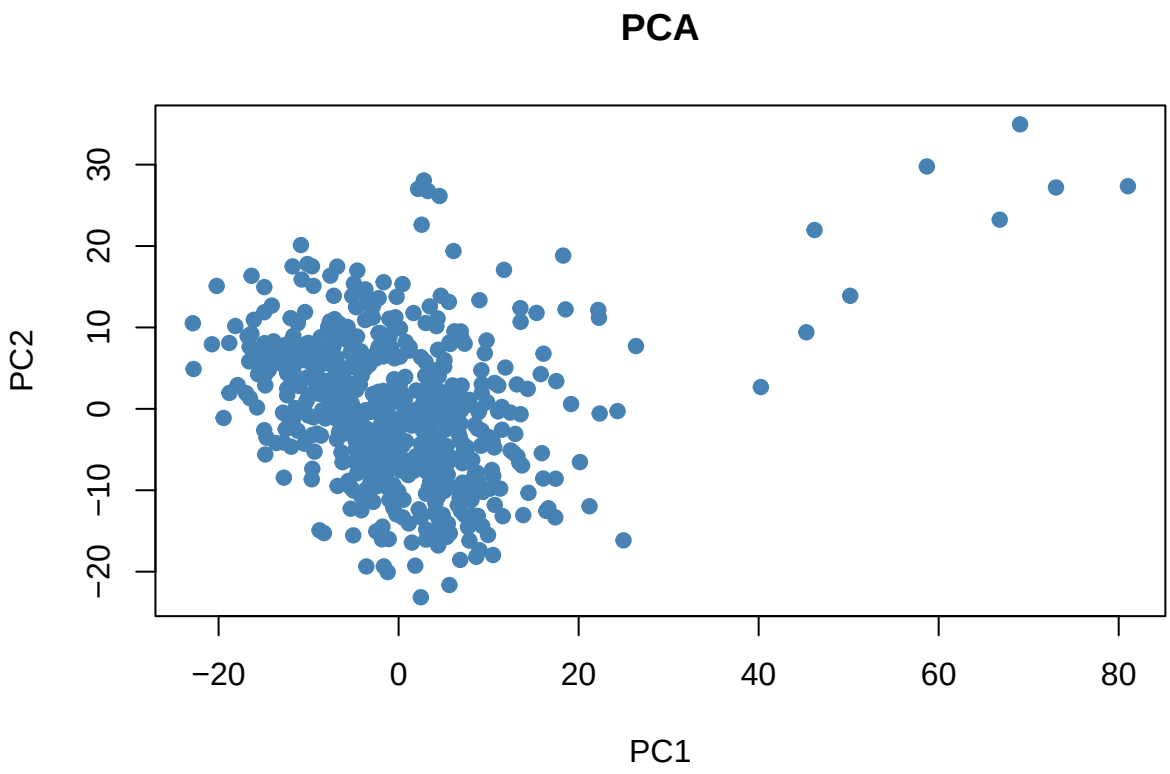
```
## [1] "C00002" "C00003" "C00005" "C00008" "C00009" "C00013"
```

```r
# Guardamos el objeto SE
save(se, file = "se.Rda")

# ------------------------------------------------------------------------------
# 4. Creamos la tabla con datos informativos sobre el objeto SE
# ------------------------------------------------------------------------------
summary_se <- data.frame(
  Propiedad = c("Clase", "Dimensiones", "Numero de assays", "Nombres de assays",
                "Numero de variables en rowData", "Numero de variables en colData"),
  Valor = c(
    class(se)[1],
    paste(dim(se), collapse = " x "),
    length(assayNames(se)),
    paste(assayNames(se), collapse = ", "),
    length(colnames(rowData(se))),
    length(colnames(colData(se)))
  ),
  stringsAsFactors = FALSE
)

# Mostramos la tabla
kable(summary_se, caption = "Resumen del objeto SummarizedExperiment")
```

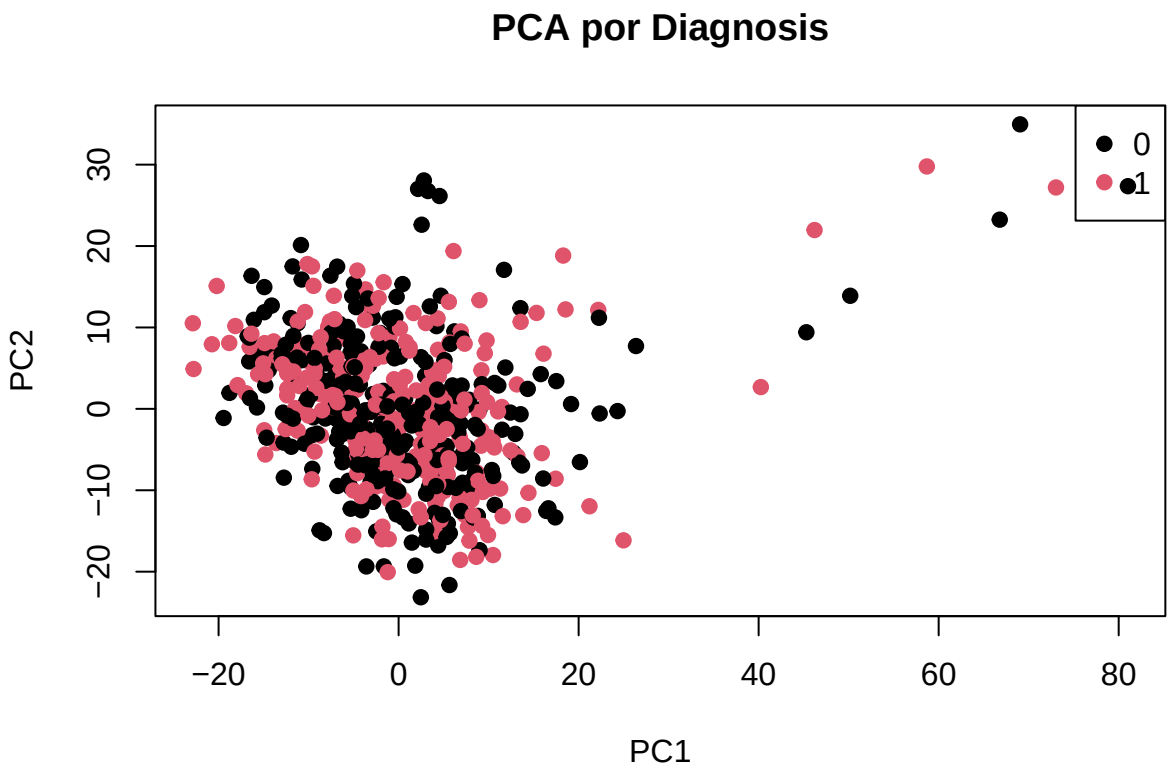
Table 1: Resumen del objeto SummarizedExperiment

Propiedad	Valor
Clase	SummarizedExperiment
Dimensiones	1169 x 580
Numero de assays	1
Nombres de assays	counts
Numero de variables en rowData	18
Numero de variables en colData	420

```r
# ------------------------------------------------------------------------------
# 5. Ejemplos de exploracion
# ------------------------------------------------------------------------------

# 5.1 PCA
# Extraemos la matriz de datos
data_matrix <- assay(se, "counts")

# Realizamos el PCA
pca_result <- prcomp(t(data_matrix), scale. = TRUE)

# Grafica
plot(pca_result$x[, 1], pca_result$x[, 2],
     xlab = "PC1", ylab = "PC2",
     main = "PCA",
     pch = 19, col = "steelblue")
```




```
# PCA por Diagnosis
  groups <- as.factor(colData(se)$Diagnosis)
  plot(pca_result$x[, 1], pca_result$x[, 2],
       xlab = "PC1", ylab = "PC2",
       main = "PCA por Diagnosis",
       pch = 19, col = groups)
  legend("topright", legend = levels(groups),
         col = 1:length(levels(groups)), pch = 19)
```




```
# 5.2 Analisis de la distribucion de pacientes por diagnostico
# Numero de pacientes que desarrollaron cáncer de próstata
diagnosis <- colData(se)$Diagnosis
patients_table <- table(diagnosis)

# Numero de pacientes que desarrollaron cáncer de próstata metastatico
metastases <- with(colData(se), (BoneScanResults.fu1case == 2) | (BoneScanResults.fu2case == 2))

# Numero total de pacientes que desarrollaron cancer de prostata metastatico
total_metastases <- sum(metastases, na.rm = TRUE)
total_metastases
```

```
## [1] 8
```

```
# Creamos un data frame con los resultados
result_table <- data.frame(
  Cancer = c("No", "Si"),
  Pacientes = c(as.numeric(patients_table["0"]), as.numeric(patients_table["1"])),
  Metastasis = c(0, total_metastases)
)
# Mostramos la tabla
kable(result_table, caption = "Numero de casos con cancer de prostata")
```

Table 2: Numero de casos con cancer de prostata

Cancer	Pacientes	Metastasis
No	313	0
Si	267	8

```
# 5.3 Analisis del valor pronostico de las poliaminas
# Filtramos rowData del objeto se para obtener solo las filas con SUB_PATHWAY "Polyamine Metabolism"
polyamine_annotations <- rowData(se) %>%
  as.data.frame() %>%
  filter(SUB_PATHWAY == "Polyamine Metabolism")

# Extraer los nombres de los metabolitos, asumiendo que la columna con los nombres es CHEMICAL_NAME
polyamine_names <- polyamine_annotations$CHEMICAL_NAME

# Crear tabla de poliaminas
polyamine_table <- data.frame(Poliamina = polyamine_names)
kable(polyamine_table, caption = "Nombres de Poliaminas")
```

Table 3: Nombres de Poliaminas

Poliamina
spermidine
N-acetylputrescine
5-methylthioadenosine (MTA)
4-acetamidobutanoate
acisoga
(N(1) + N(8))-acetylspermidine
N-acetyl-isoputreanine

```
# Filtramos el objeto SE para guardar solo los metabolitos de pathway "Polyamine Metabolism"
se_polyamine <- se[ !is.na(rowData(se)$SUB_PATHWAY) &
                      rowData(se)$SUB_PATHWAY == "Polyamine Metabolism", ]

# Calculamos la media de las intensidades en cada muestra para los metabolitos del pathway "Polyamine M
polyamine_means <- colMeans(assay(se_polyamine, "counts"))

# Calculamos las medias de polyamine_means para cada grupo de diagnóstico
group_means <- tapply(polyamine_means, diagnosis, mean, na.rm = TRUE)
print(group_means)
```

```
##        0        1
## 19.80315 19.75886
```

```
# Wilcoxon test para comparar los grupos
wilcox_result <- wilcox.test(polyamine_means ~ diagnosis)
print(wilcox_result)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  polyamine_means by diagnosis
## W = 44836, p-value = 0.1295
```

```
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Boxplot para visualizar el resultado
boxplot(polyamine_means ~ diagnosis,
        main = "Poliaminas ",
        xlab = "Cancer de prostata (0 = No, 1 = Si)",
        ylab = "Nivel Promedio",
        col = c("grey", "lightblue"))
```