Análisis de datos ómicos - PEC 1

Eva Saco Vilas

2024-10-26

Tabla de Contenidos

Abstract

En este documento se lleva a cabo el proceso de selección y descarga de un conjunto de datos de investigación en el campo de la bioinformática. Sobre estos datos se realiza un breve estudio mientras se hace uso de herramientas como Git y *SummarizedExperiment* del paquete BioConductor de R para mantener un control de versiones sobre este trabajo y su reposición en la plataforma Github.

Durante el transcurso del trabajo se muestran tanto un pequeño resumen de los datos del dataset y su distribución, como el proceso de preprocesado para poder encapsularlos en el objeto *SummarizedExperiment.* Adicionalmente, se expone al menos un caso de uso que muestra una ventaja significativa frente a su no utilización.

Objetivos del estudio

Este estudio pretende repasar y afianzar los conocimientos adquiridos durante el Reto 1 de la asignatura de Análisis de Datos Ómicos. Para lograrlo será necesario **Seleccionar y descargar un conjunto de datos de investigación**, **Realizar un breve estudio del conjunto de datos**, **Hacer uso de la clase *SummarizedExperiment* de BioConductor** para cargar el dataset en cuestión y, finalmente, **Crear un repositorio en github y guardar los resultados obtenidos en este**.

Materiales y Métodos

Los materiales utilizados para la realización de esta práctica han sido el repositorio https://github.com/nutrimetabolomics/metaboData/ de Github y más concretamente el conjunto de datos llamado "2018-Phosphoproteomics". Adicionalmente, se han utilizado el lenguaje de programación R, el programa de RStudio para la creación del fichero RMarkdown que genera este documento, el paquete *SummarizedExperiment* de BioConductor, el software Git y la plataforma GitHub.

Desarrollo de la Práctica

Descarga del dataset

Para comenzar a trabajar con el conjunto de datos seleccionado en este caso, 2018-Phosphoproteomics, el primer paso a realizar será descargarlo. Este proceso se ha llevado a cabo clonando el repositorio aportado en el enunciado de la práctica mediante el uso del comando **git clone https://github.com/nutrimetabolomics/metaboData/**. Una vez clonado a nuestro ordenador, se han copiado los contenidos de la carpeta "Datasets/2018-Phosphoproteomics" en otra carpeta que contendrá, más adelante, el respositorio objeto de la entrega de esta práctica.

Creación del contenedor *SummarizedExperiment*

Para la creación del contenedor *SummarizedExperiment* que contenga toda la información del dataset descargado con anterioridad, será necesaria la instalación del paquete Bioconductor correspondiente. Así mismo, se deberá cargar no solamente el paquete indicado sino también el paquete readxl para poder trabajar con ficheros ".xlsx". Esto se debe a que el conjunto de datos objeto de este estudio se encuentra en este formato.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("SummarizedExperiment")
install.packages("readxl")
```

```
## package 'readxl' successfully unpacked and MD5 sums checked
```

```
##
## The downloaded binary packages are in
##  C:\Users\evasa\AppData\Local\Temp\Rtmp4ajv6w\downloaded_packages
```

```
# Cargar los paquetes
library(SummarizedExperiment)
library(readxl)
```

Una vez instalados los paquetes, podremos proceder a la carga literal de los datos en R, utilizando funciones para leer los datos del fichero ".xlsx" como se muestra a continuación:

```
datos_phospho <- read_excel("2018-Phosphoproteomics/TIO2+PTYR-human-MSS+MSIvsPD.xlsx",
                            sheet= "originalData")

metadatos_phospho <- read_excel("2018-Phosphoproteomics/TIO2+PTYR-human-MSS+MSIvsPD.xlsx",
                                sheet= "targets")
```

Una vez cargada la información en el entorno de R, será necesario adecuar el formato de los datos al requerido para su encapsulación en un objeto *SummarizedExperiment*.

En la variable *datos_phospho*, se encuentra básicamente una tabla con diversas columnas de distinta índole. Las más interesantes en este caso son las numéricas, como por ejemplo M1_1_MSS, que si consultamos la tabla de metadatos, correspondería con las medidas de la muestra M1 que pertenecen al grupo MSS.

Por ello, de la tabla cargada anteriormente, la información a cargar en nuestro *SummarizedExperiment* será aquella relacionada con esas columnas. A mayores, también es interesante identificar cada fila de manera inequívoca para un acceso más sencillo. Esto se puede conseguir utilizando otra información que existe en la tabla, por ejemplo, en el documento adjunto al dataset descargado, se utiliza la columna *Accession* con un pequeño preprocesado para hacer las entradas únicas. De esta misma manera, el preprocesado de los datos quedaría como se puede ver a continuación:

```
# Obtenemos solo las columnas de interés
datos_medidas <- as.matrix(datos_phospho[, seq(5,16)])

# Creamos nombres de filas únicos a partir de los datos de la columna Accession
nombres_filas <- make.names(datos_phospho$Accession, unique=TRUE)

# Establecemos los nombres de las filas en la matriz creada
rownames(datos_medidas) <- nombres_filas

# Imprimimos una muestra de la matriz
head(datos_medidas)
```

```
##             M1_1_MSS   M1_2_MSS   M5_1_MSS   M5_2_MSS    T49_1_MSS  T49_2_MSS
## O00560       24.29438  44475.964      0.000   6269.141    1135.8169   21933.90
```

```
## O00560.1      0.00000  43138.904   2102.056  50355.051     248.9275    3239.16
## O00560.2   3412.60332 172143.040  77323.019 307637.429   98442.2773  192982.37
## O15264   220431.17880 145656.887 104287.815  75887.365  773377.4981  481165.54
## O15264.1  18254.77813   8529.755  35955.901  44102.316   57145.1682   34638.01
## O15551   644513.31840 261938.025 187023.484 124867.715 4487443.6920 2572575.27
##             M42_1_PD   M42_2_PD     M43_1_PD    M43_2_PD    M64_1_PD
## O00560         0.000       0.00     772.9056    2136.746    1820.724
## O00560.1    1315.904       0.00       0.0000       0.000       0.000
## O00560.2   24851.344   16547.95    5565.2821       0.000    3264.563
## O15264   1027196.292 1163747.38 4080239.1820 4885818.113 3093786.793
## O15264.1   21231.256   49499.70  666107.0448  379313.615  255792.117
## O15551    535809.187  434645.89   91361.8781   65997.913  243250.439
##              M64_2_PD
## O00560      1727.9098
## O00560.1     892.3565
## O00560.2    5901.9577
## O15264   2759104.5440
## O15264.1  579765.0018
## O15551    206632.6444
```

Una vez preparados los datos en forma de matriz, el siguiente paso será la preparación de los metadatos para poder crear el objeto *SummarizedExperiment* correctamente.

En este proceso, es fundamental asegurar que los nombres de las filas de los metadatos se correspondan con los nombres de las columnas de los datos de la matriz. Afortunadamente, éstos son fácilmente recreables a partir de los datos guardados en la variable *metadatos_phospho*.

```
# Convertimos los datos en el tipo DataFrame para poder crear el SummarizedExperiment
metadatos_preprocesados <- as(metadatos_phospho, "DataFrame")

# Añadimos los rownames fusionando los valores de las columnas Sample...1 y Phenotype
rownames(metadatos_preprocesados) <- paste(metadatos_preprocesados$Sample...1,
                                           metadatos_preprocesados$Phenotype, sep = "_")

# Imprimimos una muestra de los metadatos
head(metadatos_preprocesados)
```

```
## DataFrame with 6 rows and 4 columns
##            Sample...1  Sample...2 Individual   Phenotype
##           <character> <character>  <numeric> <character>
## M1_1_MSS         M1_1          M1          1         MSS
## M1_2_MSS         M1_2          M1          1         MSS
## M5_1_MSS         M5_1          M5          2         MSS
## M5_2_MSS         M5_2          M5          2         MSS
## T49_1_MSS       T49_1         T49          3         MSS
## T49_2_MSS       T49_2         T49          3         MSS
```

Una vez tenemos los datos en una matriz y los metadatos en un DataFrame, podemos proceder a encapsularlos en un objeto *SummarizedExperiment* de la siguiente forma:

```
# Creamos nuestro objeto SummarizedExperiment
experimento_phospho <- SummarizedExperiment(assays = list(counts = datos_medidas),
                                            colData = metadatos_preprocesados)
```

De esta manera, a partir de nuestro objeto *SummarizedExperiment*, podremos proceder a la posterior exploración de datos.

Exploración de los datos

Lo primero que podemos hacer para visualizar un poco los datos contenidos en nuestro experimento es obtener el siguiente resumen:

```
# Imprimimos un resumen del experimento
experimento_phospho
```

```
## class: SummarizedExperiment
## dim: 1438 12
## metadata(0):
## assays(1): counts
## rownames(1438): O00560 O00560.1 ... Q13283.1 Q9NYF8.12
## rowData names(0):
## colnames(12): M1_1_MSS M1_2_MSS ... M64_1_PD M64_2_PD
## colData names(4): Sample...1 Sample...2 Individual Phenotype
```

En este resumen se puede observar que el experimento contiene 1438 filas y 12 columnas. Las columnas, como se vio anteriormente, corresponden a cada una de las muestras en los datos. También se pueden observar otros datos como los nombres de las filas, establecidos anteriormente, y algún apunte sobre los metadatos como la información detallada de las muestras (Phenotype, etc.).

Adicionalmente, se puede realizar un análisis desde un punto de vista más estadístico. Con la función **assay()** sobre el objeto *SummarizedExperiment* se pueden obtener los datos en sí como se ve a continuación:

```
# Obtenemos los datos numéricos del experimento
datos_experimento <- assay(experimento_phospho)

# Realizamos un resumen estadístico de los datos
summary(datos_experimento)
```

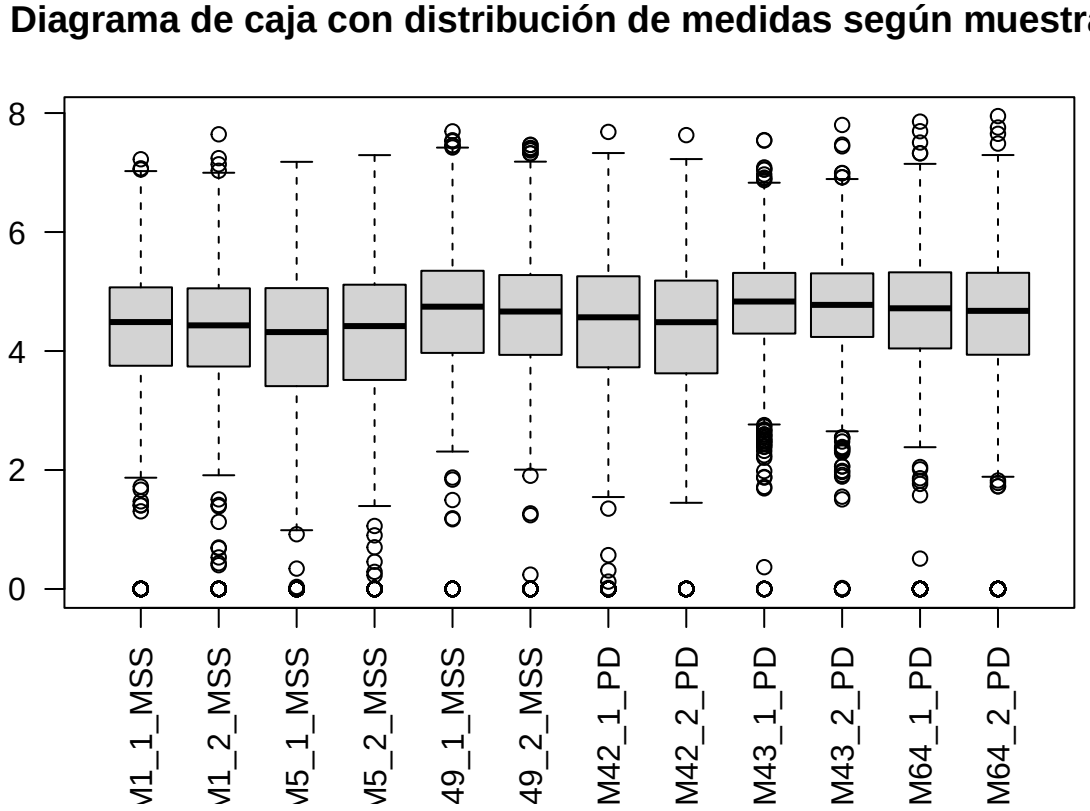
```
##     M1_1_MSS           M1_2_MSS           M5_1_MSS           M5_2_MSS
##  Min.   :       0   Min.   :       0   Min.   :       0   Min.   :       0
##  1st Qu.:    5653   1st Qu.:    5497   1st Qu.:    2573   1st Qu.:    3273
##  Median :   30682   Median :   26980   Median :   20801   Median :   26241
##  Mean   :  229841   Mean   :  253151   Mean   :  232967   Mean   :  261067
##  3rd Qu.:  117373   3rd Qu.:  113004   3rd Qu.:  113958   3rd Qu.:  130132
##  Max.   :16719906   Max.   :43928481   Max.   :15135169   Max.   :19631820
##    T49_1_MSS          T49_2_MSS          M42_1_PD           M42_2_PD
##  Min.   :       0   Min.   :       0   Min.   :       0   Min.   :       0
##  1st Qu.:    9306   1st Qu.:    8611   1st Qu.:    5341   1st Qu.:    4216
##  Median :   55641   Median :   46110   Median :   36854   Median :   30533
##  Mean   :  542449   Mean   :  462616   Mean   :  388424   Mean   :  333587
##  3rd Qu.:  223103   3rd Qu.:  189141   3rd Qu.:  180252   3rd Qu.:  152088
##  Max.   :49218872   Max.   :29240206   Max.   :48177680   Max.   :42558111
##     M43_1_PD           M43_2_PD           M64_1_PD           M64_2_PD
##  Min.   :       0   Min.   :       0   Min.   :       0   Min.   :       0
##  1st Qu.:   19641   1st Qu.:   17299   1st Qu.:   11038   1st Qu.:    8660
##  Median :   67945   Median :   59607   Median :   52249   Median :   47330
```

```
##  Mean   :  349020   Mean   :  358822   Mean   :  470655   Mean   :  484712
##  3rd Qu.:  205471   3rd Qu.:  201924   3rd Qu.:  209896   3rd Qu.:  206036
##  Max.   :35049402   Max.   :63082982   Max.   :71750330   Max.   :88912734
```

Como se puede observar en el resumen anterior, los datos de las muestras se encuentran en un rango de valores muy amplio.

Una manera de comprender mejor los datos del experimento es visualizándolos. En este caso, como se hace también en el documento de estudio de este dataset, se mostrará un diagrama de cajas aplicando una función logarítmica para su mejor visualización:

```
# Imprimimos un diagrama de cajas con los datos
boxplot(log10(datos_experimento+1), las=2,
        main = "Diagrama de caja con distribución de medidas según muestras")
```



Como ya se ha mencionado anteriormente, este tipo de exploración de los datos ya está cubierta en el documento adjunto al dataset descargado. Por ello, con afán de aportar información adicional y explotar las capacidades de la clase *SummarizedExperiment*, a continuación se mostrará información de los datos con un filtrado previo mediante el uso de ésta.

En nuestro experimento se encuentran datos de diversas muestras de 2 fenotipos distintos, MSS y PD. Estas muestras se reparten en columnas mientras que en las filas tenemos distintos fosfopéptidos. Si por ejemplo quisiéramos obtener las medias de las medidas de las distintas muestras para cada fosfopéptido de manera general, sería tan sencillo como utilizar las siguientes instrucciones:

```r
# Obtenemos la media de valores para cada fila
medias_generales <- rowMeans(assay(experimento_phospho))

# Imprimimos una muestra de las medias
head(medias_generales)
```

```
##     O00560   O00560.1   O00560.2     O15264   O15264.1     O15551
##    6691.45    8441.03   75672.65 1567558.22  179194.55  821338.29
```

Sin embargo, si por el contrario quisiéramos calcular las medias solo a partir de muestras con un fenotipo determinado, como por ejemplo el fenotipo PD, se podría hacer de la siguiente manera:

```r
# Obtenemos la media de valores para cada fila
medias_pd <- rowMeans(assay(experimento_phospho[, colData(experimento_phospho)$Phenotype == "PD"]))

# Imprimimos una muestra de las medias
head(medias_pd)
```

```
##       O00560     O00560.1     O00560.2       O15264     O15264.1       O15551
##    1076.3809     368.0435    9355.1828 2834982.0500  325284.7885  262949.6588
```

Este cálculo implica un filtrado que, de no tener un objeto *SummarizedExperiment*, no sería tan sencillo de obtener.

Reposición de los datos en GitHub

Para la reposición de los datos de esta práctica en Github es necesario crear un nuevo repositorio en la plataforma. En este caso, la dirección url de este repositorio será: https://github.com/Eviis22/Saco-Vilas-Eva-PEC1.

Una vez hecho esto, es necesario inicializar un repositorio local en nuestro ordenador, para ello es necesario situarse en la carpeta que contendrá el repositorio y utilizar el programa git desde una línea de comandos de la siguiente manera:

```
$ git init
$ git remote add origin https://github.com/Eviis22/Saco-Vilas-Eva-PEC1.git
$ git branch -M main
```

De esta forma, el repositorio local estará creado y enlazado con el repositorio creado en github. Es probable que también sea necesario establecer el usuario y correo por defecto en git con las opciones:

```
$ git config --global user.name "NombreUsuario"
$ git config --global user.email "CorreoUsuario"
```

Sustituyendo las credenciales por las utilizadas en la cuenta de GitHub.

Con el repositorio ya creado, es necesario añadir los ficheros relacionados con la práctica en la carpeta del repositorio.

Una vez la práctica se haya llevado a cabo y el documento RMarkdown esté completo, es necesario guardar un objeto contenedor con los datos y metadatos en formato binario de la siguiente forma:

```r
save(experimento_phospho, file="contenedor_SummarizedExperiment.Rda")
```

Finalmente, se pueden añadir todos los ficheros al commit actual y subirlos al repositorio remoto en GitHub de la siguiente forma:

```
$ git add .
$ git commit -m "Primera version de la PEC1 de Análisis de Datos Ómicos"
$ git push
```

Discusión, limitaciones y conclusión del estudio

Como se ha visto durante el desarrollo de esta práctica, el objeto *SummarizedExperiment* puede resultar de tanta utilidad como los ExpressionSets. Este permite agrupar la información de un experimento de manera que el análisis de sus datos sea más sencillo.

Respecto al estudio de donde vienen estos datos, este ejercicio ha revisado la información expuesta en él y éste concluye que debido a la dispersión de los datos es posible que sea necesario otro método de normalización en lugar de la aplicación de logaritmos. Una posibilidad sería aplicar raíces cuadradas sobre los datos para su normalización. Durante el desarrollo de la práctica se ha experimentado con distintas opciones para esta función pero ninguna ha dado resultados tan visualmente atractivos como los del logaritmo por lo que se ha descartado su inclusión en el documento a fin de evitar extenderlo innecesariamente.

Con respecto al uso de GitHub, se ha podido observar que su uso es muy intuitivo y parece de gran utilidad. Aunque en este caso no se haya explotado todo su potencial, se deja entrever que para proyectos de este tipo es prácticamente imprescindible.

URL repositorio GitHub: https://github.com/Eviis22/Saco-Vilas-Eva-PEC1