

PEC 1

ARANTXA GARCIA REDON

2025-03-30

En primer lugar, vamos a cargar los paquetes que necesitamos para resolver la PEC y además vamos a cargar los datos en R para poder trabajar con ellos. En mi caso he decidido trabajar con el conjunto de datos `human_cachexia`. He decidido trabajar con este conjunto de datos porque es sencillo pero versátil en cuanto a los análisis que se puede hacer con él. Se trata de datos de los metabolitos de varios pacientes, un grupo control y un grupo caquético. En los metadatos de nos indica que las muestras no son pareadas, es decir, que no se trata de datos de una misma persona que en un momento era normal y después desarrolla la enfermedad, por lo que nos da una guía de los tests estadísticos que podemos emplear y los que no. Nos dice además que todos los datos son numéricos y que no hay datos nulos.

```
# Se carga SummarizedExperiment. EN la cabecera se incluye:  
# - warning=FALSE, message=FALSE: que permite que aunque se visualice el código, los  
# warnings que aparecen al cargar la librería no ensucien el informe.  
# - results='hide' porque la opción head también devuelve un resultado muy largo.  
library(SummarizedExperiment)  
# Se cargan los datos y los metadatos para trabajar con ellos  
datos <- read.csv("human_cachexia.csv")  
metadatos <- readLines("description.md")  
# Se visualizan los datos.  
# De nuevo he empleado  
print(metadatos)  
head(datos)
```

Este conjunto de datos tiene la siguiente estructura: - Las filas representan los diferentes pacientes del estudio. Algunos son caquéticos y otros no lo son.

- En las columnas tenemos los diferentes datos que se recogen de cada paciente. La primera columna se trata de el identificador del paciente, la segunda su condición de salud que puede ser normal o caquético y el resto de las columnas son diferentes metabolitos que se han recogido de cada uno de los pacientes.

1. Creación de un objeto `SummarizedExperiment`

Para la creación del objeto `SummarizedExperiment` (de ahora en adelante, SE), he accedido a la documentación oficial de Bioconductor (<https://www.bioconductor.org/packages/release/bioc/vignettes/SummarizedExperiment/inst/doc/SummarizedExperiment.html#subsetting>).

En primer lugar, mirando el apartado de la anatomía de un objeto SE, lo que se puede observar es la necesidad de transponer los datos dado que en un objeto SE las diferentes muestras (pacientes) están en las columnas y en las filas se deben recoger los features, que en este caso son los metabolitos. Además se admite que se tenga más de un ensayo para unas mismas samples y features y unos metadatos, que se almacenen por separado pero ligados al conjunto de datos principal.

En el código de a continuación se va a crear el objeto SE tras preprocesar los datos para poder llevar a cabo su creación.

```
# Creo una nueva matriz que es la traspuesta de datos, en la cual elimino  
# las dos primeras columnas para quedarme solo con los datos numéricos
```

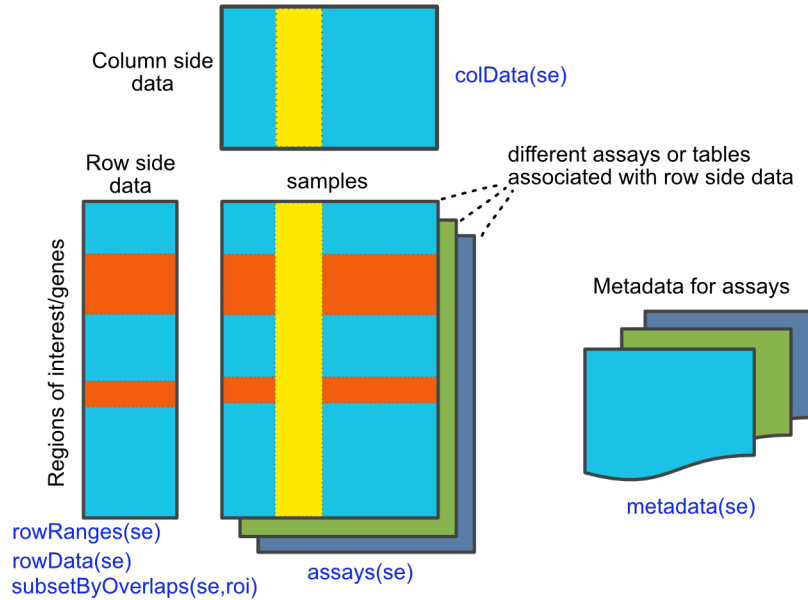


Figure 1: Estructura SE

```
datos_num<-t(datos[, -c(1,2)])
# El coldatos contendrá los valores de la segunda columna de datos.
# Es decir el dato de control o enfermos guardado en un vector
coldatos<-datos[,2]
# Convertimos coldatos en un dataframe que contiene el ID del paciente y su condicion
coldatos<-data.frame(SampleName=datos[,1], Muscle.loss=datos[,2])
# Creamos el objeto SE.
se<-SummarizedExperiment(assays=list(counts=datos_num), colData=coldatos, metadata=metadatos)
```

El objeto SE contiene los siguientes componentes: - `assays = list(counts = datos_num)`: esto contendrá los datos numéricos que hacen referencia a las cantidades de metabolitos de la matriz transpuesta `datos_num`.

- `colData = coldatos`: contiene los metadatos asociados a cada columna (es decir, a las muestras). En este caso, se incluye la información del dataframe `coldatos`, que contiene:
 - `SampleName` (nombre de la muestra)
 - `Muscle.loss` (información sobre la pérdida muscular)
- `metadata = metadatos`: Este argumento nos permite introducir los metadatos en formato texto que hemos importado del archivo `description`.

En la PEC nos pide que comparemos dos tipos de