Ibáñez-Gómez-Adriana-PEC1

Adriana Ibáñez

2025-03-28

0. Preparació i càrrega dels paquets i llibreries

Després d'instalar els paquets necessaris amb la funció install.packages, carreguem les diferents llibreries:

```
library(BiocManager)
```

```
## Warning: package 'BiocManager' was built under R version 4.4.2
```

```
library(SummarizedExperiment)
```

```
## Cargando paquete requerido: MatrixGenerics
```

```
## Warning: package 'MatrixGenerics' was built under R version 4.4.2
```

```
## Cargando paquete requerido: matrixStats
```

```
## Warning: package 'matrixStats' was built under R version 4.4.2
```

```
##
## Adjuntando el paquete: '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
```

```
## Cargando paquete requerido: GenomicRanges
```

```
## Cargando paquete requerido: stats4
```

```
## Cargando paquete requerido: BiocGenerics
```

```
##
## Adjuntando el paquete: '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

## Cargando paquete requerido: S4Vectors

##
## Adjuntando el paquete: 'S4Vectors'

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

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

## Cargando paquete requerido: IRanges

## Warning: package 'IRanges' was built under R version 4.4.2

##
## Adjuntando el paquete: 'IRanges'

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

## Cargando paquete requerido: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 4.4.2

## Cargando paquete requerido: Biobase

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

##
## Adjuntando el paquete: 'Biobase'

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

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

```r
library(readr)
library(dplyr)
```

```
##
## Adjuntando el paquete: '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
```

```r
library(tibble)
```

1. Carregar un dataset de metabolòmica

Carreguem el dataset escollit, en aquest cas MetabotypingPaper. En aquest cas tenim dos arxius excel, un que conté les dades raw i l'altre, que conté la informació de les dades.

```r
# Carregar els fitxers CSV que tenim a la mateixa carpeta que aquest document
data <- read.csv("DataValues_S013.csv", row.names = 1, check.names = FALSE)
data_info <- read.csv("DataInfo_S013.csv", row.names= 1)
data$X <- NULL
```

2. Crear un objecte de classe SumarizedExperiments

Arribats a aquest punt, necessitem afegir la informació que ens proporciona el document AAinformation_S006, com a metadada, ja que es una llista que conté informació necessària dels diferents valors / mesures que conté aquest dataset.

```
Information <- read.csv("AAInformation.txt", row.names = 1)
Information$X <- NULL
```

Un cop tenim tota la informació que ens proporcionen afegida, observem que tenim 5 característiques apart del ID que son característiques dels subjectes i no valors de metabolits, com la resta. Per aprofitar les facilitats de la classe SummarizedExperiment, es vol reorganitzar les dades per que siguin el màxim eficient. Per aixo mateix, afegim com a col_data aquestes característiques propies dels subjectes.

```
col_data <- data[, c("SURGERY", "AGE", "GENDER", "Group")]
assay_data <- t(data[, -c(1:5)])
```

```
# Creem el objecte de classe SummarizedExperiment
se <- SummarizedExperiment(
  assays = list(counts = assay_data), # Assignem les features a la assays
  colData = col_data # Assignem les característiques de cada "Subject" a la colData
)
```

Ara mateix tenim el objecte creat i amb les dades característiques de cada subjecte, hem fet la col_data. Per que tingui més sentit, s'inverteix el dataset per obtenir 39 columnes (subject id) i que en les files trobem els diferents valors dels metabòlits. Així aconseguim separar les dades en diferents taules que es poden superposar per trobar combinacions concretes.

Ara ens dediquem a afegir diferentes dades com el nom de les files:

```
row_names <- rownames(se)
```

Ara, creem una nova columna que es digui timepoint amb les dades de T (T0-5) i s'ha esborrat la informació a la columna de metabolite del temps amb el codi:

```
row_meta <- data.frame(
  feature_id = row_names,
  metabolite = gsub("_(T[0-5]+)$", "", row_names),
  timepoint = gsub(".*_(T[0-5]+)$", "\\1", row_names),
  stringsAsFactors = FALSE
)
```

Ara afegim la taula Informació, amb la informació dels diferents aminoacids, amb el codi:

```
row_meta_full <- merge(
  row_meta,
  Information,
  by.x = "metabolite",
  by.y = "Metabolite.abbreviation",
  all.x = TRUE  # Keep all measured metabolites
)
```

Ara afegim aquesta informació com a rowData al se, amb el codi:

```
rownames(row_meta_full) <- row_meta_full$feature_id
row_meta_full <- row_meta_full[rownames(se), ]
rowData(se) <- row_meta_full
```

Ara afegim el fitxer de data_info com a metadada del se, amb el codi:

```
metadata(se)$data_info <- data_info
```

Diferències SummarizedExperiment amb ExpressionSet

SummarizedExperiment i ExpressionSet son classes diferents. SummarizedExperiment és una classe usada per emmagatzemar matrius rectangulars de resultats experimentals que permet gestionat múltiples matrius de dades (assays) i qualsevol tipus de dades, com més complexes com les obtingudes se seqüenciació massiva, amb un model més estructurat. El maneig de les dades en aquesta classe s'emmagatzemen en un assays object, que permet treballar amb múltiples matrius de dades.

Per altra banda, ExpressionSet és una classe que combina múltiples fonts d'informació en una estructura convenient per representar normalment dades genòmiques, que está pensat per manipular dades provinents de dades d'expressió (assayData) com microarrays, metadades de mostres (phenoData) i dades de característiques (featureData). Es basa en objectes S4 i en emmagatzemar dades d'expressió gènica amb anotacions sobre mostres i característiques. El maneig de les dades s'una amb l'objecte assayData per contenir matrius d'expressió i altres tipus de dades associades.

Per aquestes característiques i degut a que ExpressionSet es va dissenyar inicialment per a dades de microarrays, mentre SummarizedExperiment és mes adequat per a dades de RNAseq i altres tecnologies d'alt rendiment, aquestes classes han tingut diferent suport i evolució. ExpressionSet encara es fa servir però el seu ús ha disminuït amb el temps i SummarizedExperiment és l'estàndard més recent en Bioconductor i s'està expandint per donar suport a nous tipus de dades.

3. Anàlisi exploratòria

Primer volem veure quantes variables hi ha, independentment del temps en el que ha sigut mesurat, per això:

```
se
```

```
## class: SummarizedExperiment
## dim: 689 39
## metadata(1): data_info
## assays(1): counts
## rownames(689): MEDDM_T0 MEDCOL_T0 ... SM.C24.0_T5 SM.C24.1_T5
## rowData names(7): metabolite feature_id ... Platform Data.type
## colnames(39): 1 2 ... 38 39
## colData names(4): SURGERY AGE GENDER Group
```

```
# Creem una taula per veure differents features per timepoint
table_timepoints <- table(rowData(se)$timepoint)
table_timepoints
```

```
##
##  T0  T2  T4  T5
## 172 173 172 172
```

Veiem que es mesuren 172 variables (menys en el T3 que es va mesurar una variable diferent).

Ar

```
library(ggplot2)
library(dplyr)
```

```
save(se, file = "SummarizedExperiment.Rda")
```

Guardar la matriu d'expressió en un fitxer .txt

```
write.table(assay(se), file = "expression_data.txt", sep = "\t", quote = FALSE, col.names = NA)
```

Guardar les metadades (colData) en un fitxer .txt

```
write.table(colData(se), file = "sample_metadata.txt", sep = "\t", quote = FALSE, col.names = NA)
```

Guardar les metadades dels metabolits (rowData)

```
write.table(rowData(se), file = "metabolite_metadata.txt", sep = "\t", quote = FALSE, col.names = NA)
```

Bibliografia

1. Falcon, S., Morgan, M., & Gentleman, R. (2007). An Introduction to Bioconductor's ExpressionSet Class. Bioconductor. Retrieved from https://www.bioconductor.org/packages/release/bioc/vignettes/Biobase/inst/doc/ExpressionSetIntroduction.pdf
2. Bioconductor. (n.d.). SummarizedExperiment: A container for omics experiment data. Retrieved from https://www.bioconductor.org/packages/release/bioc/vignettes/SummarizedExperiment/inst/doc/SummarizedExperiment.html