

# Notas

García Carlos

2025-01-29

## Contents

Bioconductor . . . . .	1
Ejercicio Grupal . . . . .	1
Summarized Experiment . . . . .	2

## Bioconductor

Todos los paquetes de bioconductor tienen una “**vignettes**” que son documentos en donde prueban los paquetes y su documentación, además de que diario lo evalúan sus paquetes en varios sistemas operativos para saber su funcionalidad y que no estén fallando.

En la página de Bioconductor puedes revisar los paquetes que hay actualmente, habiendo 4 tipos:

- Paquetes de software
- De anotación
- De datos experimentales
- De Workflow.

Los paquetes no son exclusivos de un solo tipo, pueden entre conectarse o tener más de una funcionalidad.

Los paquetes se instalan de diferente manera entre Bioconductor y de Cran, para instalar un paquete de bioconductor, debes de usar la función de **BiocManager::install**, con el no solo puede instalar paquete de Bioconductor, si no también de Cran.

Cada paquete es un directorio. En Bioconductor es que la gran mayoría de tus funciones tengan ejemplos de uso que se puedan ejecutar, para asegurar su uso, además que tienen un control de versiones, donde los autores dicen que es lo que se fue actualizando.

A bioconductor lo actualizan dos veces al año, es decir, cada 6 meses en abril y octubre, R se actualiza una vez al año en el mes de abril.

## Ejercicio Grupal

El paquete que elegí se llama “**Curated Data from The Cancer Genome Atlas (TCGA) as MultiAssayExperiment Objects**” el cual es un paquete enfocado en poder acceder a los datos públicos disponibles de la base de datos llamada “**The Cancer Genome Atlas**” (TCGA).

La **TCGA** es un programa de genómica del cáncer de referencia, caracterizó molecularmente más de 20.000 cánceres primarios y comparó muestras normales de 33 tipos de cáncer.

El paquete cuenta con un manual para poder usarlo con buena documentación e información para usar, desde como instalarlo, como citarlo, descargar datasets, entre otras más funciones.

El paquete se encuentra en el top 22 de de paquetes de la sección de “**ExperimentData**”, además de que tienen un **GitHub** especializado para responder preguntas y dudas, además dentro de la misma página de Bioconductor en la que se han realizado preguntas.

## Summarized Experiment

```
## Lets build our first SummarizedExperiment object
library("SummarizedExperiment")
## ?SummarizedExperiment

## De los ejemplos en la ayuda oficial

## Creamos los datos para nuestro objeto de tipo SummarizedExperiment
## para 200 genes a lo largo de 6 muestras
nrows <- 200
ncols <- 6
## Números al azar de cuentas
set.seed(20210223)
counts <- matrix(runif(nrows * ncols, 1, 1e4), nrows)
## Información de nuestros genes
rowRanges <- GRanges(
  rep(c("chr1", "chr2"), c(50, 150)),
  IRanges(floor(runif(200, 1e5, 1e6)), width = 100),
  strand = sample(c("+", "-"), 200, TRUE),
  feature_id = sprintf("ID%03d", 1:200)
)
names(rowRanges) <- paste0("gene_", seq_len(length(rowRanges)))
## Información de nuestras muestras
colData <- DataFrame(
  Treatment = rep(c("ChIP", "Input"), 3),
  row.names = LETTERS[1:6]
)
## Juntamos ahora toda la información en un solo objeto de R
rse <- SummarizedExperiment(
  assays = SimpleList(counts = counts),
  rowRanges = rowRanges,
  colData = colData
)

## Exploremos el objeto resultante
rse

## class: RangedSummarizedExperiment
## dim: 200 6
## metadata(0):
## assays(1): counts
## rownames(200): gene_1 gene_2 ... gene_199 gene_200
## rowData names(1): feature_id
## colnames(6): A B ... E F
## colData names(1): Treatment
```

```
## Número de genes y muestras
dim(rse)
```

```
## [1] 200 6
```

```
## IDs de nuestros genes y muestras
dimnames(rse)
```

```
## [[1]]
## [1] "gene_1" "gene_2" "gene_3" "gene_4" "gene_5" "gene_6"
## [7] "gene_7" "gene_8" "gene_9" "gene_10" "gene_11" "gene_12"
## [13] "gene_13" "gene_14" "gene_15" "gene_16" "gene_17" "gene_18"
## [19] "gene_19" "gene_20" "gene_21" "gene_22" "gene_23" "gene_24"
## [25] "gene_25" "gene_26" "gene_27" "gene_28" "gene_29" "gene_30"
## [31] "gene_31" "gene_32" "gene_33" "gene_34" "gene_35" "gene_36"
## [37] "gene_37" "gene_38" "gene_39" "gene_40" "gene_41" "gene_42"
## [43] "gene_43" "gene_44" "gene_45" "gene_46" "gene_47" "gene_48"
## [49] "gene_49" "gene_50" "gene_51" "gene_52" "gene_53" "gene_54"
## [55] "gene_55" "gene_56" "gene_57" "gene_58" "gene_59" "gene_60"
## [61] "gene_61" "gene_62" "gene_63" "gene_64" "gene_65" "gene_66"
## [67] "gene_67" "gene_68" "gene_69" "gene_70" "gene_71" "gene_72"
## [73] "gene_73" "gene_74" "gene_75" "gene_76" "gene_77" "gene_78"
## [79] "gene_79" "gene_80" "gene_81" "gene_82" "gene_83" "gene_84"
## [85] "gene_85" "gene_86" "gene_87" "gene_88" "gene_89" "gene_90"
## [91] "gene_91" "gene_92" "gene_93" "gene_94" "gene_95" "gene_96"
## [97] "gene_97" "gene_98" "gene_99" "gene_100" "gene_101" "gene_102"
## [103] "gene_103" "gene_104" "gene_105" "gene_106" "gene_107" "gene_108"
## [109] "gene_109" "gene_110" "gene_111" "gene_112" "gene_113" "gene_114"
## [115] "gene_115" "gene_116" "gene_117" "gene_118" "gene_119" "gene_120"
## [121] "gene_121" "gene_122" "gene_123" "gene_124" "gene_125" "gene_126"
## [127] "gene_127" "gene_128" "gene_129" "gene_130" "gene_131" "gene_132"
## [133] "gene_133" "gene_134" "gene_135" "gene_136" "gene_137" "gene_138"
## [139] "gene_139" "gene_140" "gene_141" "gene_142" "gene_143" "gene_144"
## [145] "gene_145" "gene_146" "gene_147" "gene_148" "gene_149" "gene_150"
## [151] "gene_151" "gene_152" "gene_153" "gene_154" "gene_155" "gene_156"
## [157] "gene_157" "gene_158" "gene_159" "gene_160" "gene_161" "gene_162"
## [163] "gene_163" "gene_164" "gene_165" "gene_166" "gene_167" "gene_168"
## [169] "gene_169" "gene_170" "gene_171" "gene_172" "gene_173" "gene_174"
## [175] "gene_175" "gene_176" "gene_177" "gene_178" "gene_179" "gene_180"
## [181] "gene_181" "gene_182" "gene_183" "gene_184" "gene_185" "gene_186"
## [187] "gene_187" "gene_188" "gene_189" "gene_190" "gene_191" "gene_192"
## [193] "gene_193" "gene_194" "gene_195" "gene_196" "gene_197" "gene_198"
## [199] "gene_199" "gene_200"
##
## [[2]]
## [1] "A" "B" "C" "D" "E" "F"
```

```
## Nombres de tablas de cuentas que tenemos (RPKM, CPM, counts, logcounts, etc)
assayNames(rse)
```

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

```
## El inicio de nuestra tabla de cuentas
head(assay(rse))
```

```
##           A           B           C           D           E           F
## gene_1 2577.960 8526.615 2226.3070 3615.8967 1723.8851 3267.954
## gene_2 7793.183 3462.579 478.2716 7688.3839 295.2813 2698.921
## gene_3 9571.769 5280.564 9772.1671 9916.0076 2621.2085 6880.067
## gene_4 4641.969 2784.091 6670.6757 258.5218 2771.8970 8737.586
## gene_5 6436.758 7053.276 9978.8199 3588.1194 1447.9821 7290.890
## gene_6 6845.704 1502.045 4383.5686 9750.8286 3529.3153 2192.060
```

```
## Información de los genes en un objeto de Bioconductor
rowRanges(rse)
```

```
## GRanges object with 200 ranges and 1 metadata column:
##           seqnames      ranges strand | feature_id
##           <Rle>       <IRanges> <Rle> | <character>
## gene_1      chr1 286235-286334      + | ID001
## gene_2      chr1 586770-586869      - | ID002
## gene_3      chr1 577897-577996      + | ID003
## gene_4      chr1 494350-494449      + | ID004
## gene_5      chr1 686692-686791      - | ID005
## ...         ...         ...      ... | ...
## gene_196    chr2 804998-805097      - | ID196
## gene_197    chr2 177462-177561      - | ID197
## gene_198    chr2 649993-650092      - | ID198
## gene_199    chr2 275940-276039      - | ID199
## gene_200    chr2 487418-487517      + | ID200
## -----
## seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

```
## Tabla con información de los genes
rowData(rse) # es idéntico a 'mcols(rowRanges(rse))'
```

```
## DataFrame with 200 rows and 1 column
##           feature_id
##           <character>
## gene_1      ID001
## gene_2      ID002
## gene_3      ID003
## gene_4      ID004
## gene_5      ID005
## ...         ...
## gene_196    ID196
## gene_197    ID197
## gene_198    ID198
## gene_199    ID199
## gene_200    ID200
```

```
## Tabla con información de las muestras
colData(rse)
```

```
## DataFrame with 6 rows and 1 column
##      Treatment
##   <character>
## A      ChIP
## B      Input
## C      ChIP
## D      Input
## E      ChIP
## F      Input
```

```
## Comando 1
rse[1:2, ]
```

```
## class: RangedSummarizedExperiment
## dim: 2 6
## metadata(0):
## assays(1): counts
## rownames(2): gene_1 gene_2
## rowData names(1): feature_id
## colnames(6): A B ... E F
## colData names(1): Treatment
```

```
## Comando 2
rse[, c("A", "D", "F")]
```

```
## class: RangedSummarizedExperiment
## dim: 200 3
## metadata(0):
## assays(1): counts
## rownames(200): gene_1 gene_2 ... gene_199 gene_200
## rowData names(1): feature_id
## colnames(3): A D F
## colData names(1): Treatment
```

```
library("iSEE")
```

```
## Cargando paquete requerido: SingleCellExperiment
```

```
iSEE::iSEE(rse)
```