



RNA-seq

Expositor	Leonardo Collado Torres
Fecha	@Feb 23, 2021 → Feb 26, 2021

0. Generalidades del curso

[Ligas importantes](#)

[Prerrequisitos](#)

1. Creación de mi git en **RStudio**

[Repositorios creados:](#)

2. *Bioconductor*

3. Objetos de Bioconductor paraa datos de expresión

[SummerizedExperiment](#)

[GenomicRanges](#)

[rtracklayer](#)

[iSSE](#)

4. Datos de RNA-seq a través de `recount3`

Diferencia de cálculo `raw_count` y `count`

5. Bases estadísticas

[Regresión lineal](#)

[Modelos estadísticos en R](#)

Paquete `Explorerecount3` `study explorerrecount3` `study explorerrecount3` `study explorerrecount3` `study explorerrecount3` `study explorerModelMatrix`


Manejo de datos de RNA-seq

[Normalización](#)

[Library size normalization](#)

Extras

[R graphics](#)

 [Jeff Leek How to be a modern scientist](#)

[R Themes](#)

[Punto de corte](#)

[Paquetes](#)

[Librería *purr*](#)

[Genefilter](#)


0. Generalidades del curso

Ligas importantes

- Git del curso

lcolladotor/rnaseq_LCG-UNAM_2021

Introductory RNA-seq course for LCG-UNAM Feb 2021 (in Spanish) - lcolladotor/rnaseq_LCG-UNAM_2021


 https://github.com/lcolladotor/rnaseq_LCG-UNAM_2021



- *Book del curso*


Intro RNA-seq LCG-UNAM 2021

Intro RNA-seq LCG-UNAM 2021

 https://lcolladotor.github.io/rnaseq_LCG-UNAM_2021/index.html#course-schedule

- Canal *Slack*


Slack

 <http://comunidadbioinfo.slack.com>

- Servidor *RStudio LCG*

RStudio: Browser Not Supported


Your web browser is not supported by RStudio. RStudio requires one of the following browser versions (or higher): See the RStudio Platform Support page for more information about browser support in RStudio products.

 <http://132.248.220.108:8787/>

- Canal *YouTube* (Leonardo Collado)

Leonardo Collado Torres

Welcome! Here you can find videos from the R/Bioconductor-powered Team Data Science team lead by Leonardo Collado Torres as well as videos from the LIBD


 <https://www.youtube.com/channel/UCxB1GhLzIEOikTL-aDwqFg>

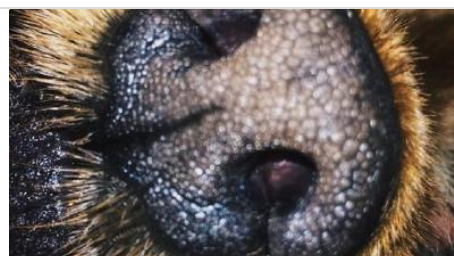


- Proyecto final del curso

paurosales/proyecto_rnaseq_2021

Proyecto final del curso de RNA-seq 2021 impartido por Leonardo Collado Torres para los estudiantes de la Licenciatura en Ciencias Genómicas de la UNAM.

 https://github.com/paurosales/proyecto_rnaseq_2021



Prerrequisitos

Instalación de paquetes en **RStudio**



Durante el curso usé el ambiente de la LCG para no descargarlo en *Windows*

```
## For installing Bioconductor packages
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

## Install required packages
BiocManager::install(
  c(
    "usethis", ## Utilities
    "here",
    "biocthis",
    "postcards",
    "pryr",
```

```

"sessioninfo",

"SummarizedExperiment", ## Main containers / vis
"iSEE",

"ExploreModelMatrix", ## RNA-seq
"limma",
"recount3",

"pheatmap", ## Visualization
"ggplot2",
"patchwork",
"RColorBrewer",

"spatialLIBD" ## Advanced
)
)

```


1. Creación de mi *git* en *RStudio*

Repositorios creados:

- Notas del curso

paurosales/curso_rnaseq_2021

Contribute to paurosales/curso_rnaseq_2021 development by creating an account on GitHub.

 https://github.com/paurosales/curso_rnaseq_2021



- Repositorio personal

Paulina Rosales-Becerra

Born on April 21st, 2001, in Guadalajara, México. Currently studying an undergraduate degree on Genomic Science at Universidad Nacional Autónoma de México (UNAM). Interested in Genomic Medicine and Data Science.

<http://paurosales.github.io>

2. *Bioconductor*

Repositorio abierto y comunitario de paquetes de R para el análisis de datos genómicos. Usa **R como lenguaje base**. Desarrollo colaborativo (importancia de la comunidad científica).



Actualizado dos veces al año (abril y octubre). En abril también se actualiza R.

Ejercicio: Bioconductor

3. Objetos de Bioconductor paraa datos de expresión

SummerizedExperiment

Contención de datos en una arreglo particular, principalmente compuesto por tres tablas:

- colData
- rowRanges
- assay(s)

Se pueden agregar datos extra como datos de los experimentos (metaData/exptData)



metaData es el que se utiliza en código



The diagram illustrates the structure of a SummarizedExperiment object and its associated data matrices. At the top, the R code for creating a SummarizedExperiment object is shown:

```
se <- SummarizedExperiment(
  assays,
  rowData,
  colData,
  exptData
)
```

Below the code, the components are visualized:

- rowData(se)**: A matrix where rows represent features (genes) and columns represent samples. It is labeled "Features (genes)" on the left.
- assays(se)**: A stack of matrices representing different assays. It is labeled "Features (genes)" on the left. A hand icon points to a specific assay matrix.
- colData(se)**: A matrix where rows represent samples and columns represent sample-level data. It is labeled "Samples" on the left.
- exptData(se)**: A matrix where rows represent samples and columns represent experimental data. It is labeled "Samples" on the left.

Arrows indicate the relationships between the components:

- Arrows point from the "Features (genes)" label to both **rowData(se)** and **assays(se)**.
- An arrow points from the "Samples" label to both **colData(se)** and **exptData(se)**.
- Curved arrows point from the **colData(se)** matrix to the **assays(se)** stack, indicating that sample-level data is used to filter or subset the assays.

Below the visualizations, the R code for accessing specific data is shown:

```
rowData(se)$entrezId
assays(se)$count
exptData(se)$projectId
```

¿Cómo se manejan estas tablas en conjunto?

Diagram illustrating the relationship between data structures in a Summarized Experiment:

- Central Structure:** A stack of `assays(se)`.
- Left Structure:** A single `Features (rows)` object, associated with `rowRanges(se)` and `rowData(se)`.
- Right Structure:** A single `Samples (Columns)` object, associated with `colData(se)` and `se[, se$dex == "T"]`.
- Bottom Structure:** `assaySubsetByOverlaps(se, roi)` and `assay[se[, se$dex == "T"]]`.
- Far Right Structure:** A stack of `metadata(se)` objects, associated with `metadata(se)$modelFormula`.

Arrows indicate data flow and relationships between these components.

Resultado del objeto sce notas disponibles en le
repositorio de notas del curso

Permite acceder a regiones del genoma específicas

```
gr <- GRanges(
  seqnames = Rle(c("chr1", "chr2", "chr1", "chr3"), c(1, 3, 2, 4)),
  ranges = IRanges(101:110, end = 111:120, names = head(letters, 10)),
  strand = Rle(strand(c("-", "+", "*", "+", "-")), c(1, 2, 2, 3, 2)),
  score = 1:10,
  GC = seq(1, 0, length=10))
gr
```


```
## GRanges object with 10 ranges and 2 metadata columns:
##      seqnames      ranges strand |      score      GC
##      <Rle> <IRanges>  <Rle> | <integer> <numeric>
## a      chr1    101-111      - |         1 1.000000
## b      chr2    102-112      + |         2 0.888889
## c      chr2    103-113      + |         3 0.777778
## d      chr2    104-114      * |         4 0.666667
## e      chr1    105-115      * |         5 0.555556
## f      chr1    106-116      + |         6 0.444444
## g      chr3    107-117      + |         7 0.333333
## h      chr3    108-118      + |         8 0.222222
## i      chr3    109-119      - |         9 0.111111
## j      chr3    110-120      - |        10 0.000000
## -----
##      seqinfo: 3 sequences from an unspecified genome; no seqlengths
```

¿Cómo se construyen?

rtracklayer

rtracklayer

Extensible framework for interacting with multiple genome browsers (currently UCSC built-in) and manipulating annotation tracks in various formats (currently GFF, BED, bedGraph,

 <http://bioconductor.org/packages/release/bioc/html/rtracklayer.html>



Manejo de distintos formatos que contienen información genómica.

iSEE

Muestra de manera interactiva la información de un objeto `SingleCellExperiment`

```
## Explora el objeto rse de forma interactiva
library("iSEE")
iSEE::iSEE(rse)
```

4. Datos de RNA-seq a través de **recount3**

Explore and download data from the recount3 project

The recount3 package enables access to a large amount of uniformly processed RNA-seq data from human and mouse.

You can download RangedSummarizedExperiment objects at

 <http://research.libd.org/recount3/>



Procesamiento uniforme de datos públicos RNA-seq, permite:

- "Democratización de datos"
- Acceso para cualquier persona que quiera, independientemente de su acceso a clústers o *high-performance*
- Comparación de datos de expresión a partir de el análisis de las mismas herramientas
- Fácil de usar
- Datos de humano y ratón

Fase 1 (2011) **ReCount**

20 muestras

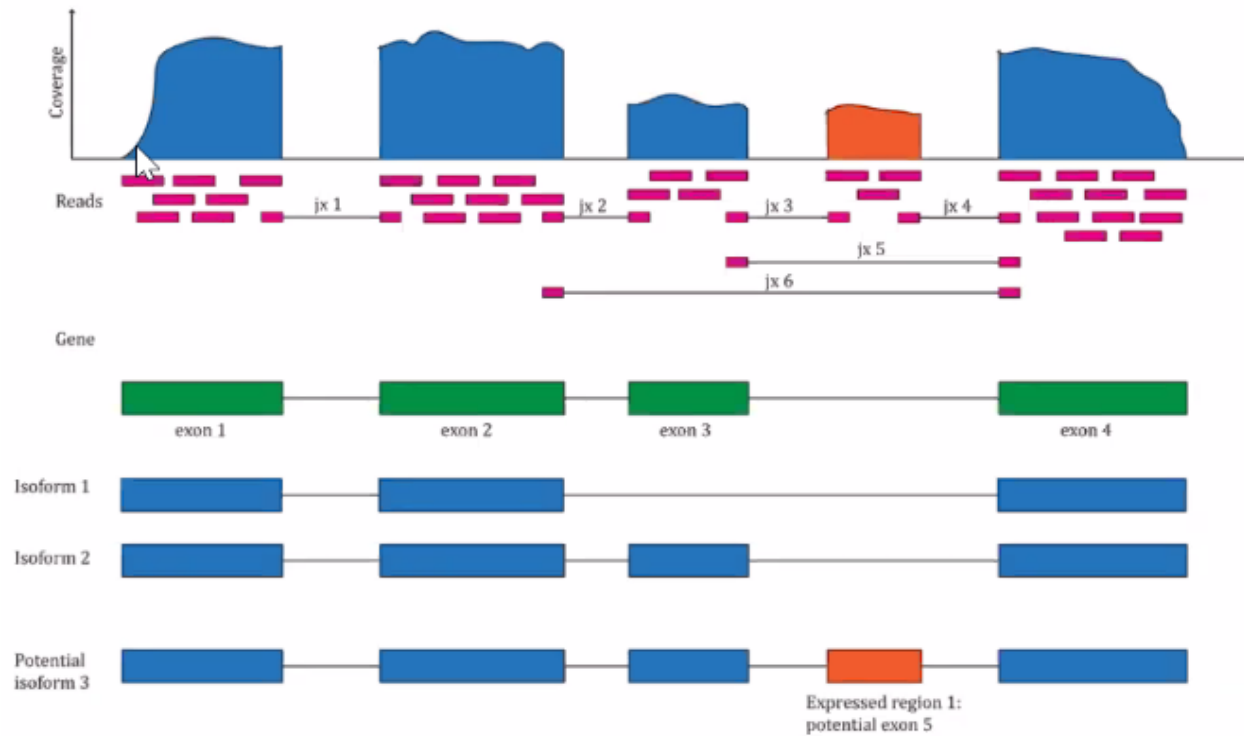
Fase 2 (2017) **recount2**

70 mil muestras

Fase 3 (en aprobación)

recount3

700 mil muestras



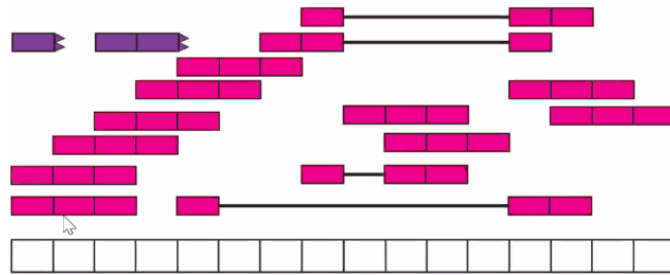
Ejemplo de información disponible por `recount2` y `recount3`

Diferencia de cálculo `raw_count` y `count`

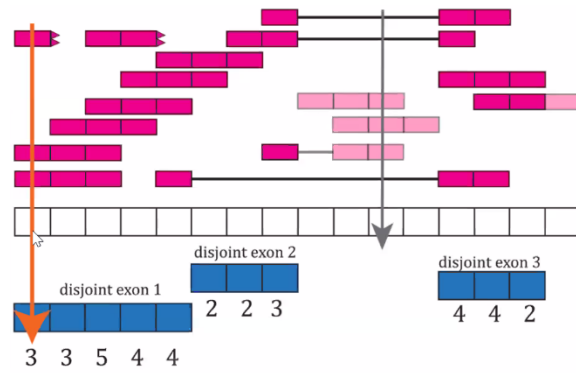
Procedimiento llevado a cabo por `recount`

Comprimen la información y se calcula la covertura de cada base.

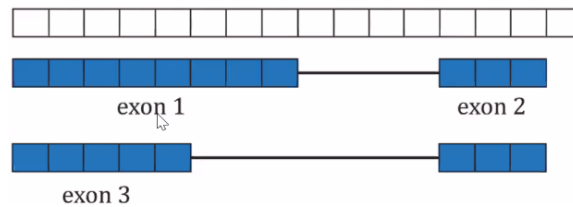




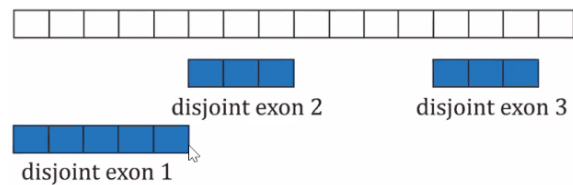
Cuantificar las lecturas para cada base del genoma

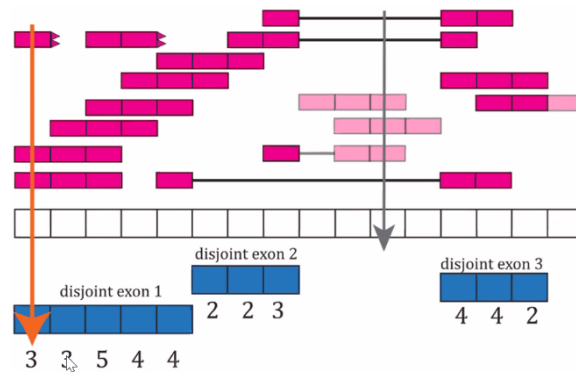


La notación indica cuáles son las coordenadas de los genes en las lecturas

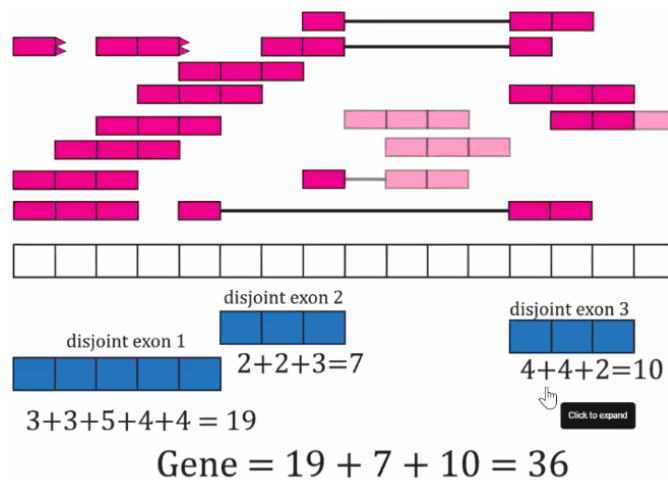


Encuentran las regiones diferentes de exones disjuntos (no superlapantes)





calcular valores para cada segmento exónico



El 36 es el valor de `raw_count` que `recount` calcula. NO el valor de lecturas cobrelapantes que normalmente se calcula como `count`

5. Bases estadísticas

Regresión lineal

$$Y = \beta_0 - \beta_1 X + \epsilon$$

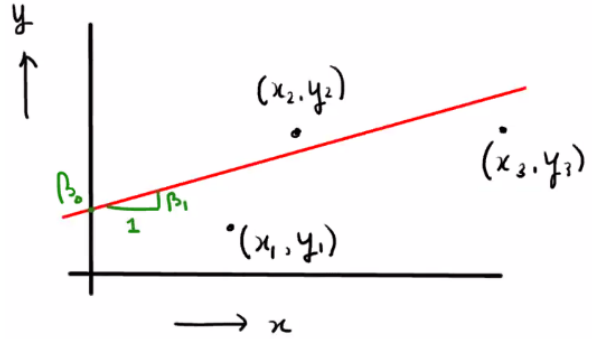
con

$$\epsilon \sim N(0, \sigma^2)$$

$Y \rightarrow$ response variable

$X \rightarrow$ explanatory variable

β_i es el cambio promedio en Y con respecto a X



Modelos estadísticos en R

Con `model.matrix()`

Funciona con variables categóricas y numéricas.

Las variables categóricas las convierte o interpreta como *dummy variables* o variables indicativas (binarias).

```
## ?model.matrix
## model.matrix(log(Y) ~ log(X_1) * log(X_2)) para variables X dependientes
## model.matrix(log(Y) ~ log(X_1) + log(X_2)) para variables X independientes
mat <- with(trees, model.matrix(log(Volume) ~ log(Height) + log(Girth)))
mat
```

```
##      (Intercept) log(Height) log(Girth)
## 1             1    4.248495    2.116256
## 2             1    4.174387    2.151762
## 3             1    4.143135    2.174752
## 4             1    4.276666    2.351375
## 5             1    4.394449    2.370244
```

Columnas:

`Intercept` es el β_0 o nivel basal don X y $Y = 0$

`log(X)` es el β_1 de la variable X

Paquete `Explorerecount3` `study` `explorerrecount3` `study` `explorerrecount3` `study` `explorerrecount3` `study` `explorerModelMatrix`

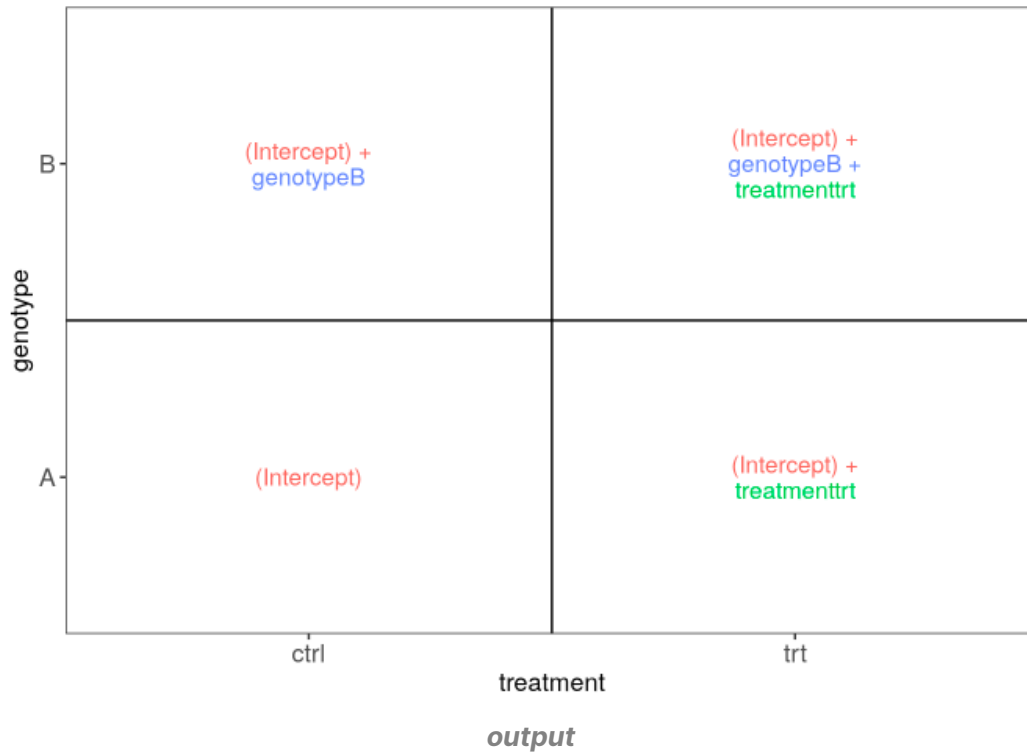
Paquete de *Bioconductor*

```
## Datos de ejemplo
(sampleData <- data.frame(
  genotype = rep(c("A", "B"), each = 4),
  treatment = rep(c("ctrl", "trt"), 4)
))
```

```
##   genotype treatment
## 1      A      ctrl
## 2      A      trt
## 3      A      ctrl
## 4      A      trt
## 5      B      ctrl
## 6      B      trt
## 7      B      ctrl
## 8      B      trt
```

```
## Creemos las imágenes usando ExploreModelMatrix
vd <- ExploreModelMatrix::VisualizeDesign(
  sampleData = sampleData,
  # Misma sintaxis que regresión lineal pero sin Y porque la Y cambia de acuerdo al gen
  designFormula = ~ genotype + treatment,
  textSizeFitted = 4
)

## Veamos las imágenes
cowplot::plot_grid(plotlist = vd$plotlist)
```



No hay genotipo A porque es el de referencia, es decir, por *default* es A a menos de que sea B.

Podemos crear una matriz con *dummy variables*, de acuerdo a variables de referencia generadas automáticamente.

```
mod <- model.matrix(~ genotype + treatment, data = sampleData)
mod
```

```
##   genotypeB treatmenttrt
## 1         0           0
## 2         0           1
## 3         0           0
## 4         0           1
## 5         1           0
## 6         1           1
## 7         1           0
## 8         1           1
```

Para cambiarlas a nivel código podemos usar:

```
## Cambiar la variable de referencia en por comandos
sampleData$genotype
## [1] "A" "A" "A" "A" "B" "B" "B" "B"
factor(sampleData$genotype)
## [1] A A A A B B B B
## Levels: A B
factor(sampleData$genotype, levels = c("B", "A"))
## [1] A A A A B B B B
## Levels: A B
```

```
> sampleData$genotype
[1] "A" "A" "A" "A" "B" "B" "B" "B"
> factor(sampleData$genotype)
[1] A A A A B B B B
Levels: A B
> factor(sampleData$genotype, levels = c("B", "A"))
[1] A A A A B B B B
Levels: B A
```

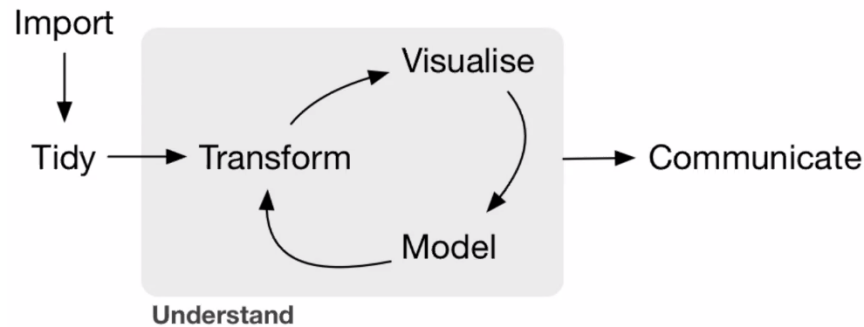
En el ambiente interactivo se puede cambiar la variable de referencia:

```
## Usaremos shiny otra vez
app <- ExploreModelMatrix(
  sampleData = sampleData,
  designFormula = ~ genotype + treatment
)
if (interactive()) shiny::runApp(app)
```



Ejercicios más avanzados en el repositorio de GitHub

Manejo de datos de RNA-seq



Normalización

Library size normalization

Toma en cuenta la suma de los niveles de expresión de todos los genes de todas las muestras y las compara. Normalizando de acuerdo a dicha comparación.

Librería `edgeR`

Tiene su propia clase de objetos, pero es fácil transformar un objeto

`SummarizedExperiment` al objeto utilizado en `edgeR`

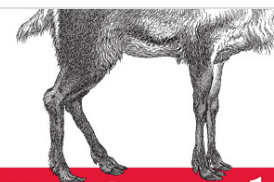
	Condition A	Condition B		A norm lib size	B norm lib size	A / B
Gene1	8	8		0.075471698	0.150943	0.5
Gene2	33	33		0.311320755	0.622642	0.5
Gene3	12	12		0.113207547	0.226415	0.5
Gene4	13	0		0.122641509	0	#DIV/0!
Gene5	6	0		0.056603774	0	#DIV/0!
Gene6	34	0		0.320754717	0	#DIV/0!
Total	106	53				

Extras

R graphics

R Graphics Cookbook, 2nd edition

This cookbook contains more than 150 recipes to help scientists, engineers, programmers, and data analysts generate high-quality graphs quickly-without having to comb through all
<https://r-graphics.org/>



R Graphics

 **Jeff Leek** *How to be a modern scientist*

Jeff Leek


<http://jtleek.com/>

R Themes

```
remotes::install_github(c(  
  "gadenbuie/rstthemes"  
))  
remotes::install_cran("suncalc")  
rstthemes::install_rstthemes(include_base16 = TRUE)
```

Punto de corte


Ejemplo

 https://github.com/LieberInstitute/brainseq_phase2/blob/master/expr_cutoff/pdf/suggested_expr_cutoffs_gene.pdf

Paquetes

Librería *purr*


Programación funcional

 <https://github.com/ComunidadBioInfo/cdsb2019/blob/master/05-fp.pdf>

Genefilter

genefilter

DOI: 10.18129/B9.bioc.genefilter Bioconductor version: Release (3.12) Some basic functions for filtering genes. Author: R. Gentleman, V. Carey, W. Huber, F. Hahne Maintainer:

 <https://www.bioconductor.org/packages/release/bioc/html/genefilter.html>

