

Access the slides and files here:

https://github.com/j-berg/bioinformatics_bootcamp

#5.1

R Studio

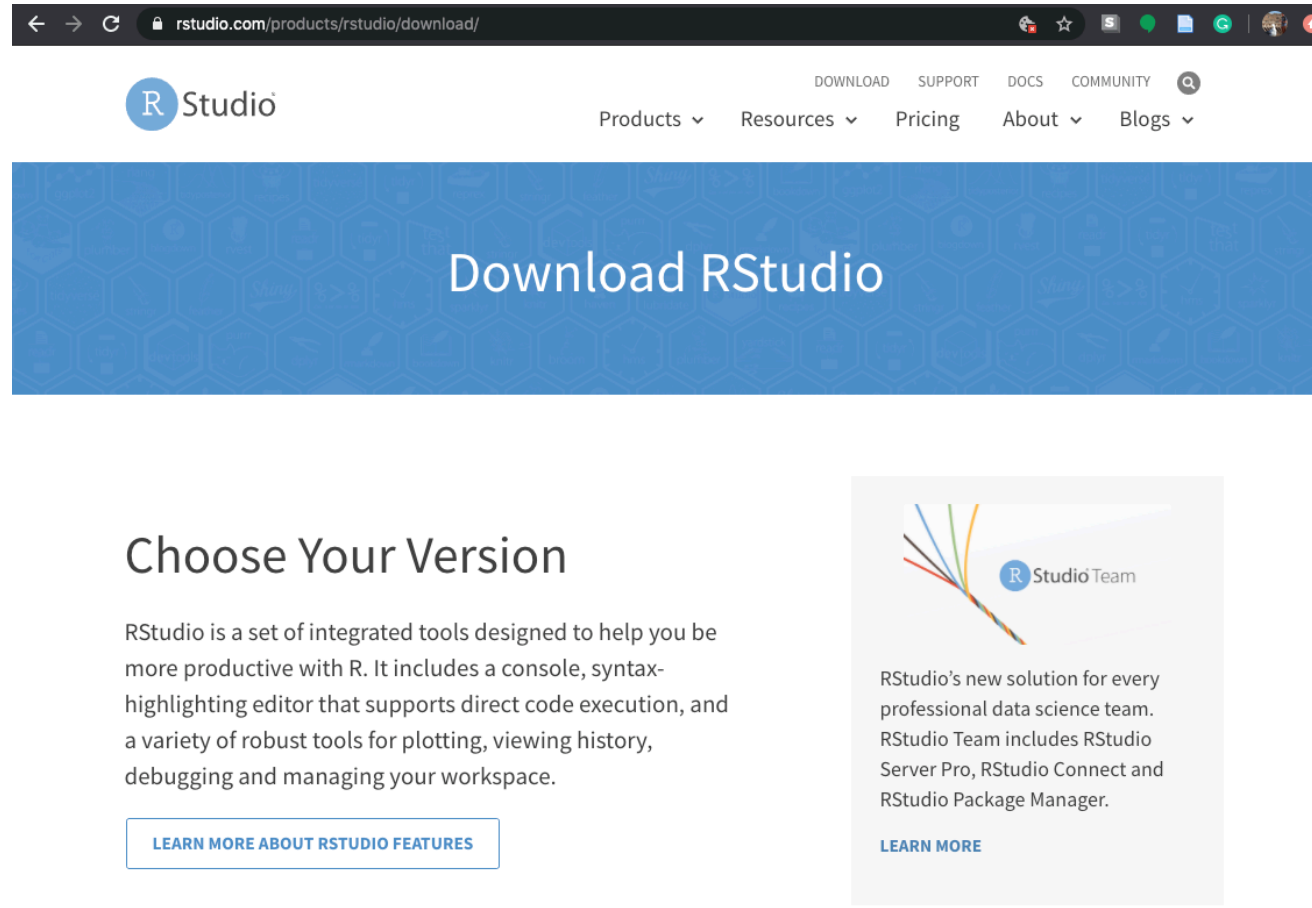
Bioconductor

Libraries & Documentation

Variables, loops, functions, etc.

Outputting analyses

Downloading RStudio



A screenshot of the RStudio website's download page. The browser address bar shows 'rstudio.com/products/rstudio/download/'. The page features a blue header with the RStudio logo and navigation links: Products, Resources, Pricing, About, and Blogs. A large blue banner with a hexagonal pattern contains the text 'Download RStudio'. Below this, the section 'Choose Your Version' describes RStudio as a set of integrated tools for R. To the right, a box for 'RStudio Team' describes a new solution for professional data science teams, including RStudio Server Pro, RStudio Connect, and RStudio Package Manager.

← → ↻ 🔒 rstudio.com/products/rstudio/download/

R Studio

DOWNLOAD SUPPORT DOCS COMMUNITY

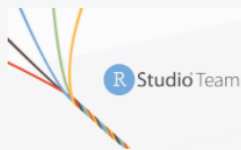
Products ▾ Resources ▾ Pricing About ▾ Blogs ▾

Download RStudio

Choose Your Version

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace.

[LEARN MORE ABOUT RSTUDIO FEATURES](#)



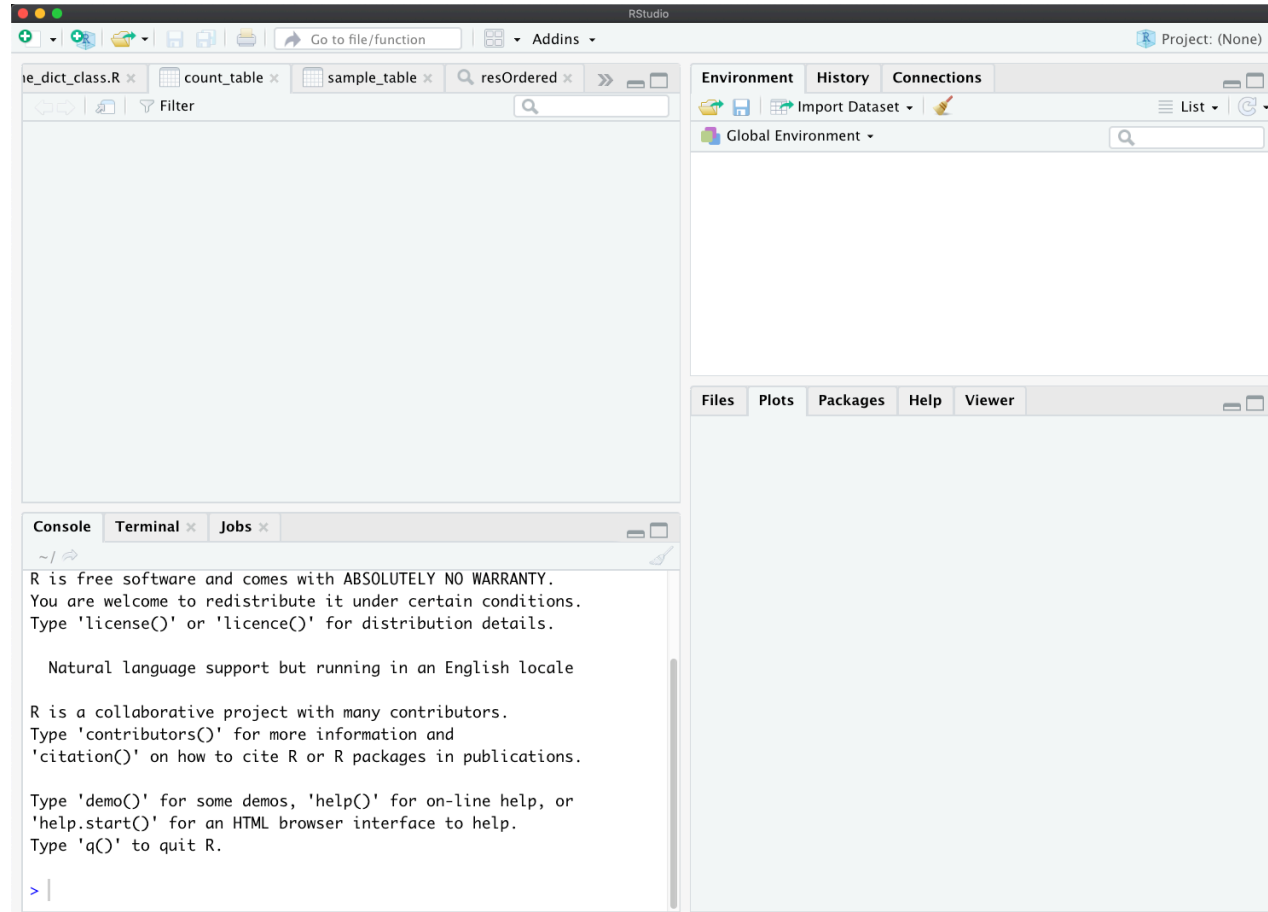
R Studio Team

RStudio's new solution for every professional data science team. RStudio Team includes RStudio Server Pro, RStudio Connect and RStudio Package Manager.

[LEARN MORE](#)

Using RStudio

View data here ->



<- View environment variables here

<- View files, plots, and help menus

Enter commands here ->

Integers

- Numeric types refer to the precision of the value
- Integers: whole numbers

```
> x <- 1
```

```
> type(x)
```

```
> y <- as.integer(x)
```

```
> type(y)
```

```
> type(y)
```

Mixing data types

- Combining datatypes may lead to incompatibilities, errors, so be aware!
- Converting a float or double to an integer will lose precision
 - > x <- 1.234567890123456789
 - > y <- as.integer(x)
 - > type(x)
 - > type(y)
 - > print(x)
 - > print(y)

Strings

- Represents text instead of numbers
- Generally indicated with quotes
- Can force integers and floats/doubles to strings

```
> x <- "Hello world!"
```

```
> print(x)
```

```
> y <- 1.23456789
```

```
> toString(y)
```

```
> paste(x, toString(y), sep="!!!")
```

Arrays

- A sequential, ordered list of values

```
> x <- c(1, 2, 3)
```

```
> y <- c("a", "b", "c")
```

```
> z <- c(1, 2, 3, "d", "e")
```

- Notice this converts your integers to strings

- What happens when we try this?

```
> zz <- array(c(x, y), dim = c(3, 2))
```

- Access something from an array
- What does this do?

```
> zz[2, 1]
```


Sets

Try this:

```
> v <- c(1,1,2,3,4,4,5,6)  
> unique(v)
```

Why does this happen? Let's talk about another similar data type

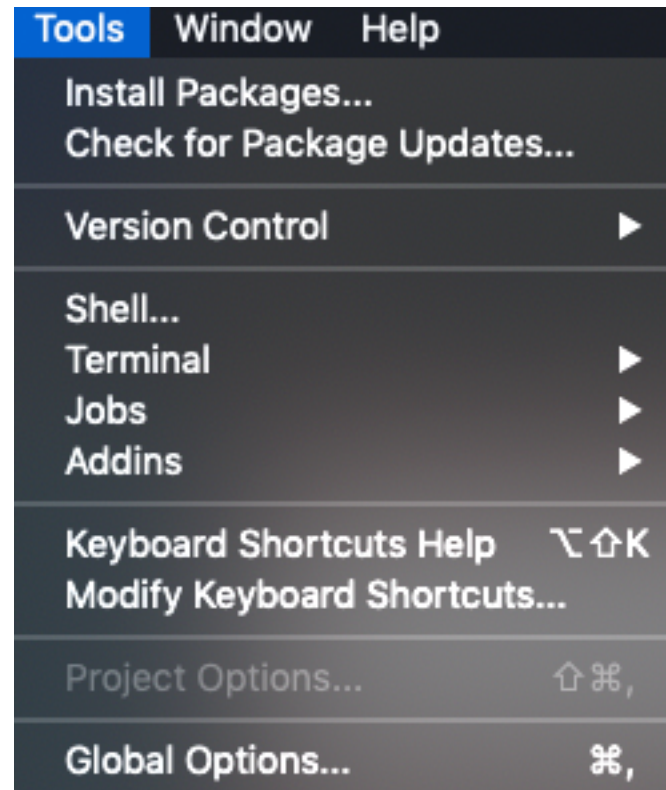
Dictionaries (or hashes)

```
> library(hash)
> h <- hash()
> h[["a"]] <- "Hello"
> h[["b"]] <- "world!"
> h
> h[["a"]]
> h[["a"]] <- "Goodbye"
> h
```

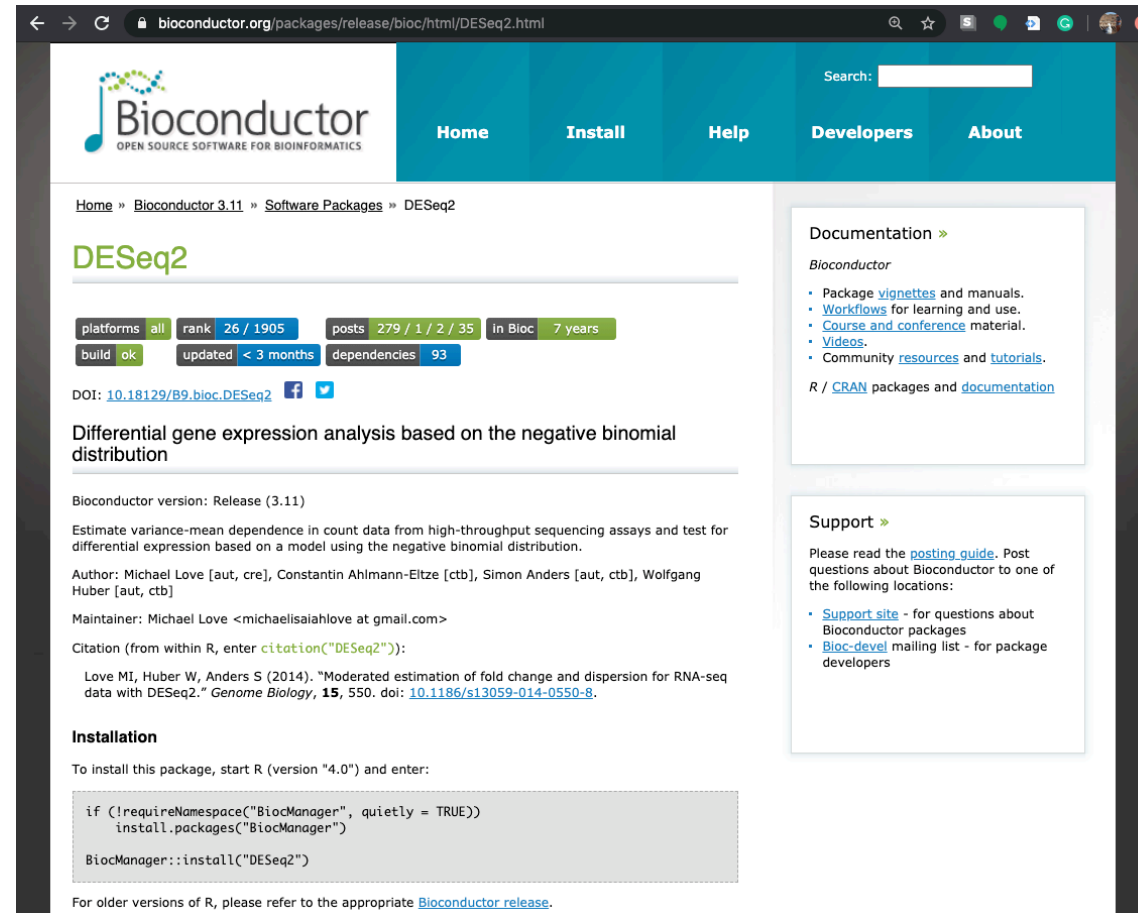
Libraries

- A collection of functions
 - External
 - Internal (we'll talk about this later)
- Download as follows:
 - Basic packages:
 - Tools -> Packages -> enter package name and install
 - Download "data.table"
 - Bioinformatics packages
 - Google "download DESeq2" and click on the Bioconductor link
 - Run the installation code block
 - If it asks you whether you want to update all/some/none, "none" is usually okay

Install general R packages



Install bioinformatics software



The screenshot shows the Bioconductor website for the DESeq2 package. The page includes a navigation bar with links to Home, Install, Help, Developers, and About. The main content area displays the package name 'DESeq2' in green, followed by a summary of its features: platforms (all), rank (26 / 1905), posts (279 / 1 / 2 / 35), in Bioc (7 years), build (ok), updated (< 3 months), and dependencies (93). The DOI is 10.18129/B9.bioc.DESeq2. The description states: 'Differential gene expression analysis based on the negative binomial distribution'. The Bioconductor version is Release (3.11). The author is Michael Love [aut, cre], Constantin Ahlmann-Eltze [ctb], Simon Anders [aut, ctb], and Wolfgang Huber [aut, ctb]. The maintainer is Michael Love <michaelsaiahlove at gmail.com>. The citation is: Love MI, Huber W, Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome Biology*, 15, 550. doi: 10.1186/s13059-014-0550-8. The installation instructions are: 'To install this package, start R (version "4.0") and enter: if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager") BiocManager::install("DESeq2")'. For older versions of R, it refers to the appropriate Bioconductor release. The right sidebar contains sections for Documentation and Support.

bioconductor.org/packages/release/bioc/html/DESeq2.html

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help Developers About

Search:

Home » Bioconductor 3.11 » Software Packages » DESeq2

DESeq2

platforms all rank 26 / 1905 posts 279 / 1 / 2 / 35 in Bioc 7 years
build ok updated < 3 months dependencies 93

DOI: 10.18129/B9.bioc.DESeq2

Differential gene expression analysis based on the negative binomial distribution

Bioconductor version: Release (3.11)

Estimate variance-mean dependence in count data from high-throughput sequencing assays and test for differential expression based on a model using the negative binomial distribution.

Author: Michael Love [aut, cre], Constantin Ahlmann-Eltze [ctb], Simon Anders [aut, ctb], Wolfgang Huber [aut, ctb]

Maintainer: Michael Love <michaelsaiahlove at gmail.com>

Citation (from within R, enter `citation("DESeq2")`):
Love MI, Huber W, Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome Biology*, 15, 550. doi: 10.1186/s13059-014-0550-8.

Installation

To install this package, start R (version "4.0") and enter:

```
if (!requireNamespace("BiocManager", quietly = TRUE))  
  install.packages("BiocManager")  
  
BiocManager::install("DESeq2")
```

For older versions of R, please refer to the appropriate [Bioconductor release](#).

Documentation »

Bioconductor

- Package [vignettes](#) and manuals.
- [Workflows](#) for learning and use.
- [Course and conference](#) material.
- [Videos](#).
- Community [resources](#) and [tutorials](#).

R / [CRAN](#) packages and [documentation](#)

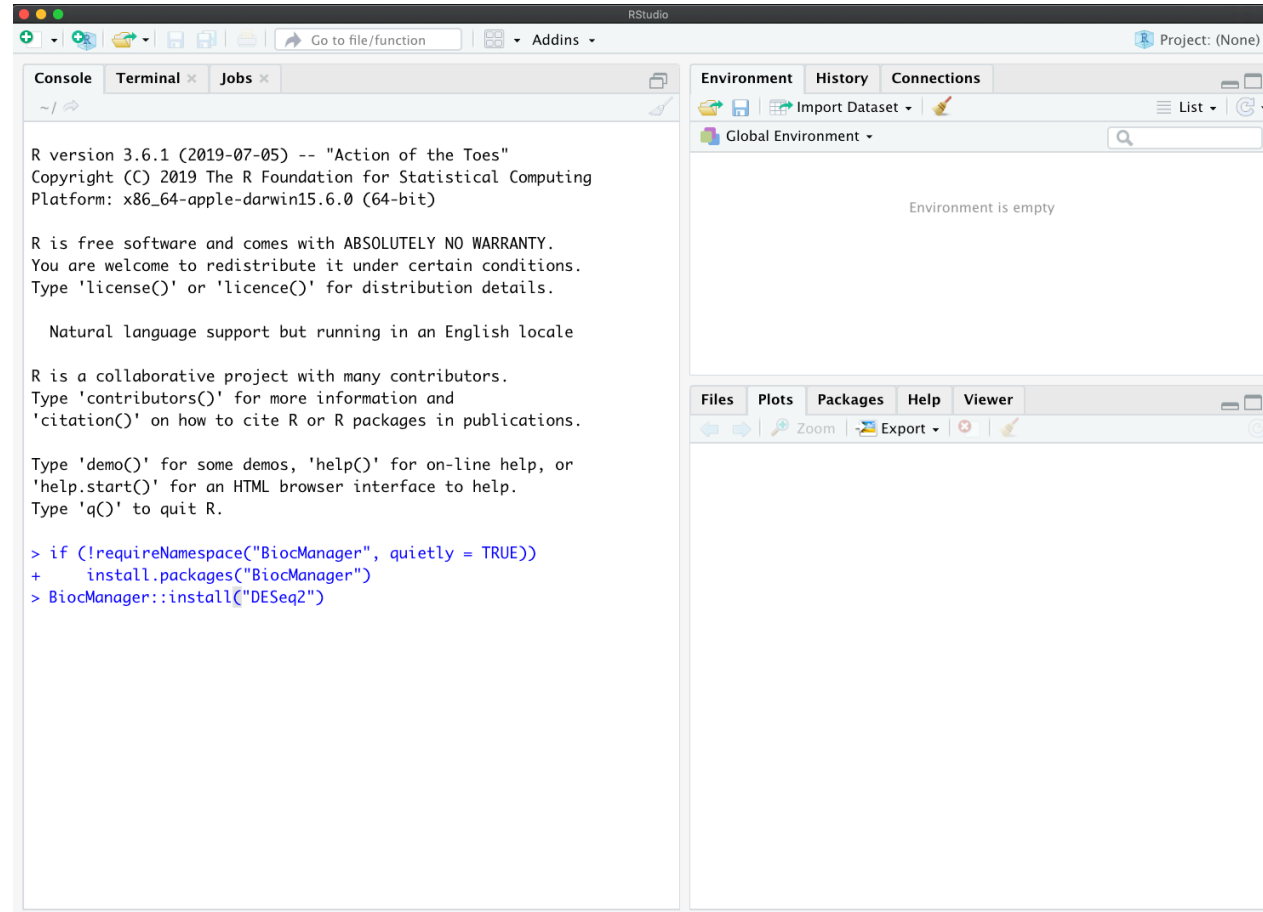
Support »

Please read the [posting guide](#). Post questions about Bioconductor to one of the following locations:

- [Support site](#) - for questions about Bioconductor packages
- [Bioc-devel](#) mailing list - for package developers

Run these commands
to install ->

Installing DESeq2



The screenshot shows the RStudio application window. The top toolbar includes icons for file operations and a search bar. The left sidebar contains tabs for 'Console', 'Terminal', and 'Jobs'. The 'Console' tab is active, displaying the R version 3.6.1 (2019-07-05) and the installation of DESeq2 using BiocManager. The right sidebar contains tabs for 'Environment', 'History', and 'Connections'. The 'Environment' tab is active, showing 'Global Environment' and 'Environment is empty'. The bottom toolbar includes tabs for 'Files', 'Plots', 'Packages', 'Help', and 'Viewer'. The 'Packages' tab is active, showing the installed packages.

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

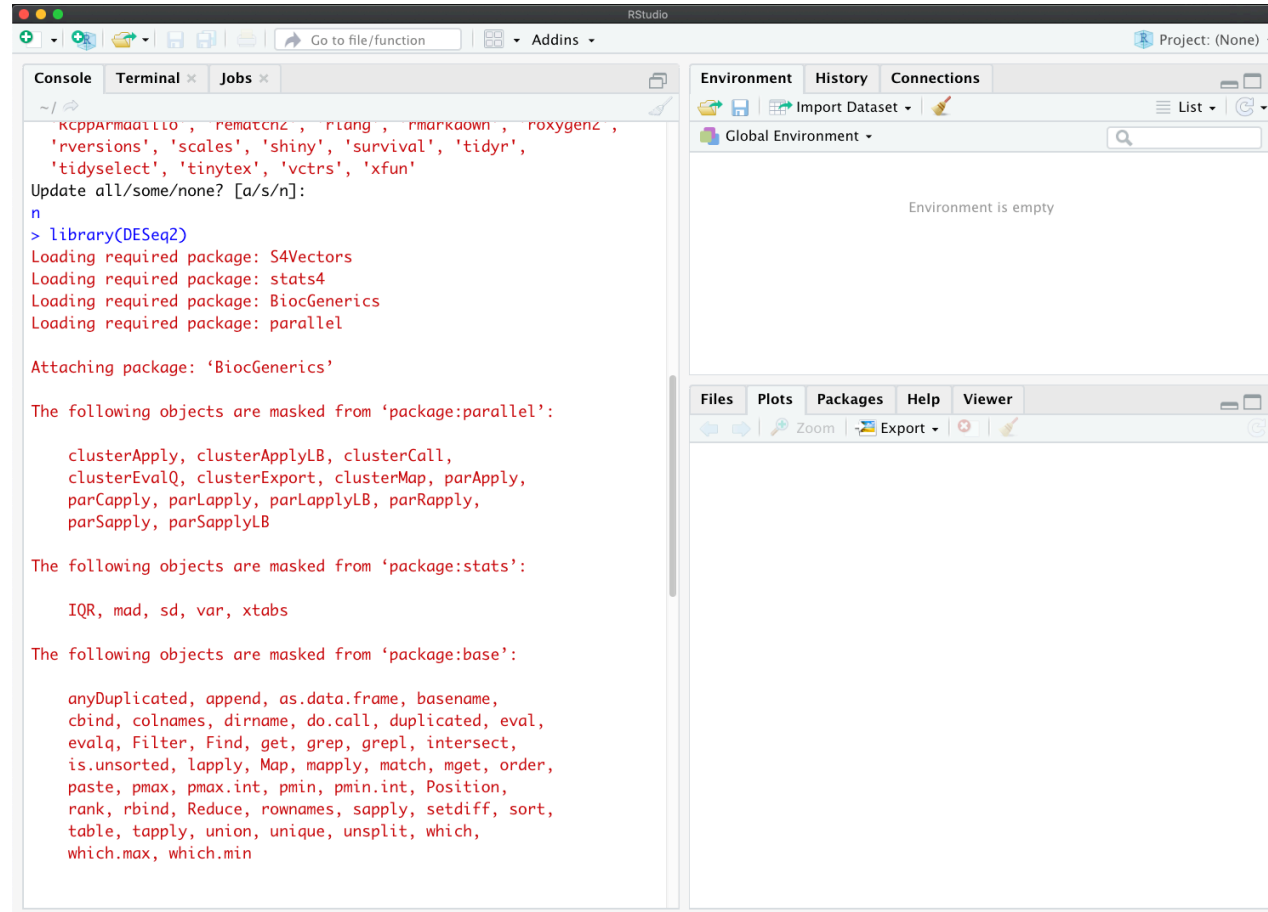
Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> if (!requireNamespace("BiocManager", quietly = TRUE))
+   install.packages("BiocManager")
> BiocManager::install("DESeq2")
```

Installing DESeq2



The screenshot shows the RStudio interface with the following components:

- Console:** Displays the installation process for DESeq2. It starts with a list of installed packages: `rcpparmadillo`, `rematch2`, `riang`, `rmarkdown`, `roxygen2`, `rversions`, `scales`, `shiny`, `survival`, `tidyr`, `tidyselect`, `tinytex`, `vctrs`, and `xfun`. It then prompts for an update, which is declined. The command `library(DESeq2)` is executed, leading to the loading of required packages: `S4Vectors`, `stats4`, `BiocGenerics`, and `parallel`. It then attaches the `BiocGenerics` package and lists masked objects from `package:parallel`, `package:stats`, and `package:base`.
- Environment:** Shows the Global Environment, which is currently empty.
- Files, Plots, Packages, Help, Viewer:** These panels are visible at the bottom of the interface.

```
~/
rcpparmadillo, rematch2, riang, rmarkdown, roxygen2,
'rversions', 'scales', 'shiny', 'survival', 'tidyr',
'tidyselect', 'tinytex', 'vctrs', 'xfun'
Update all/some/none? [a/s/n]:
n
> library(DESeq2)
Loading required package: S4Vectors
Loading required package: stats4
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: 'BiocGenerics'

The following objects are masked from 'package:parallel':

  clusterApply, clusterApplyLB, clusterCall,
  clusterEvalQ, clusterExport, clusterMap, parApply,
  parCapply, parLapply, parLapplyLB, parRapply,
  parSapply, parSapplyLB

The following objects are masked from 'package:stats':

  IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

  anyDuplicated, 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, setdiff, sort,
  table, tapply, union, unique, unsplit, which,
  which.max, which.min
```

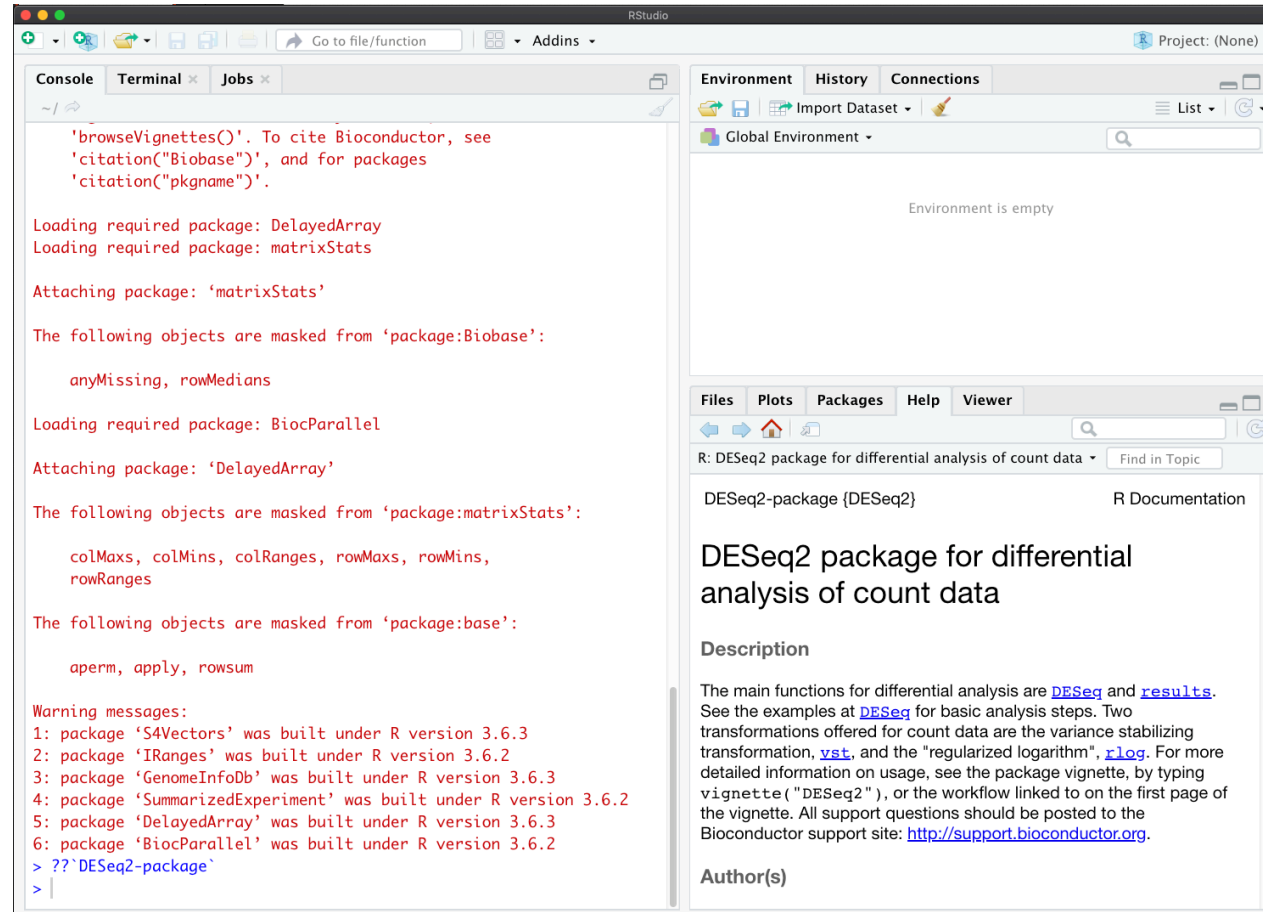
Loading a library

- Need to give the session or script access to the functions from the library
 - > `library(data.table)`
 - > `library(DESeq2)`

Getting help

- StackOverflow
- Library documentation
 - Usually opened with a “?” before the function name (sometimes a “??”)
- Viewing the help menus
 - > ?data.table
 - > ??DESeq2
 - > ?estimateSizeFactors
 - > browseVignettes("DESeq2")
 - -> [DESeq2::counts](#)

Accessing Bioconductor tool help menu

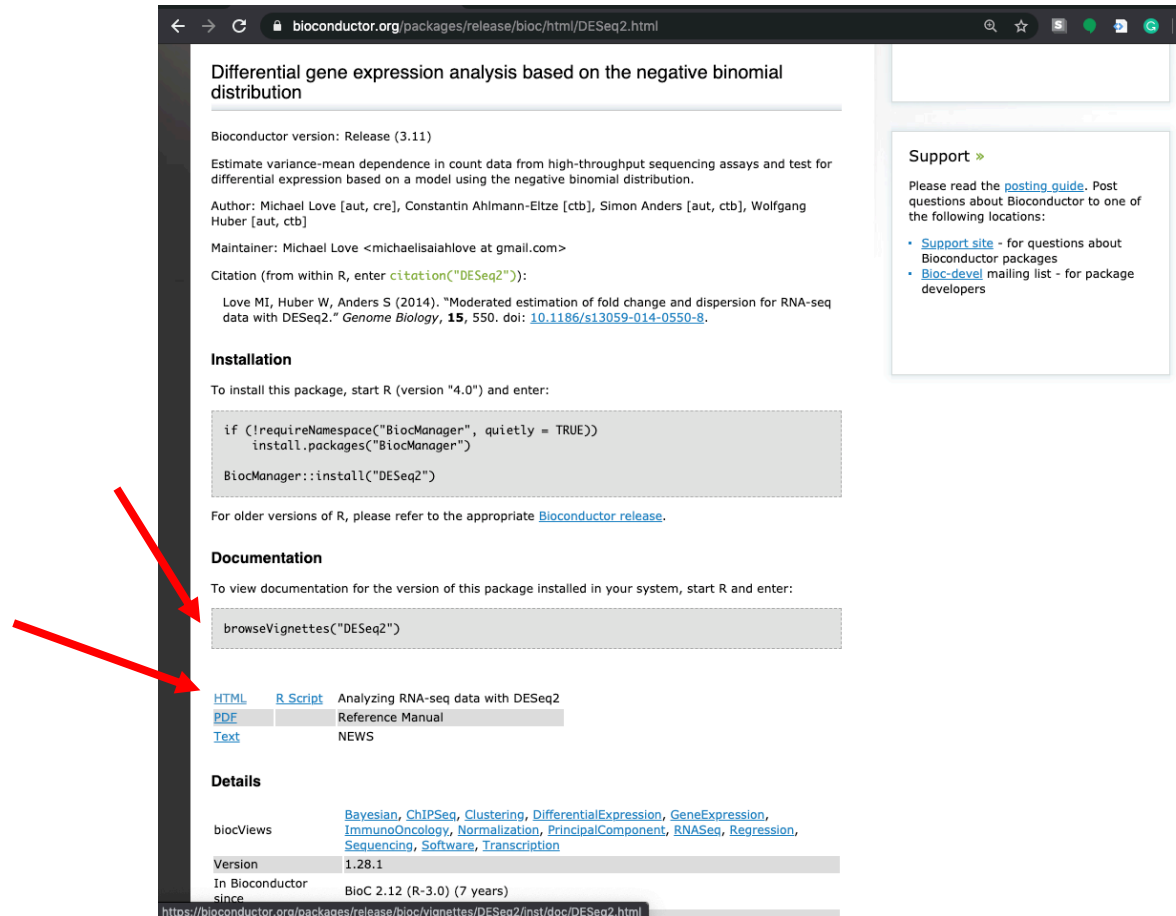


The screenshot displays the RStudio interface. The left pane shows the R console with the following output:

```
'browseVignettes()'. To cite Bioconductor, see  
'citation("Biobase")', and for packages  
'citation("pkgname")'.  
  
Loading required package: DelayedArray  
Loading required package: matrixStats  
  
Attaching package: 'matrixStats'  
  
The following objects are masked from 'package:Biobase':  
  
    anyMissing, rowMedians  
  
Loading required package: BiocParallel  
  
Attaching package: 'DelayedArray'  
  
The following objects are masked from 'package:matrixStats':  
  
    colMaxs, colMins, colRanges, rowMaxs, rowMins,  
    rowRanges  
  
The following objects are masked from 'package:base':  
  
    aperm, apply, rowsum  
  
Warning messages:  
1: package 'S4Vectors' was built under R version 3.6.3  
2: package 'IRanges' was built under R version 3.6.2  
3: package 'GenomeInfoDb' was built under R version 3.6.3  
4: package 'SummarizedExperiment' was built under R version 3.6.2  
5: package 'DelayedArray' was built under R version 3.6.3  
6: package 'BiocParallel' was built under R version 3.6.2  
> ?? 'DESeq2-package'  
>
```

The right pane shows the 'Help' tab with the 'DESeq2 package for differential analysis of count data' documentation. The title is 'DESeq2 package for differential analysis of count data'. The description states: 'The main functions for differential analysis are [DESeq](#) and [results](#). See the examples at [DESeq](#) for basic analysis steps. Two transformations offered for count data are the variance stabilizing transformation, [vst](#), and the "regularized logarithm", [rlog](#). For more detailed information on usage, see the package vignette, by typing `vignette("DESeq2")`, or the workflow linked to on the first page of the vignette. All support questions should be posted to the Bioconductor support site: <http://support.bioconductor.org>.'

Accessing Bioconductor tool help menu



The screenshot shows the Bioconductor website for the DESeq2 package. The page is titled "Differential gene expression analysis based on the negative binomial distribution". It includes sections for Bioconductor version, description, author, maintainer, citation, installation, documentation, and details. Two red arrows point to the "Documentation" section and the "HTML" link in the "Details" section.

bioconductor.org/packages/release/bioc/html/DESeq2.html

Differential gene expression analysis based on the negative binomial distribution

Bioconductor version: Release (3.11)

Estimate variance-mean dependence in count data from high-throughput sequencing assays and test for differential expression based on a model using the negative binomial distribution.

Author: Michael Love [aut, cre], Constantin Ahlmann-Eltze [ctb], Simon Anders [aut, ctb], Wolfgang Huber [aut, ctb]

Maintainer: Michael Love <michaelsaiahlove@gmail.com>

Citation (from within R, enter `citation("DESeq2")`):

Love MI, Huber W, Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome Biology*, **15**, 550. doi: [10.1186/s13059-014-0550-8](https://doi.org/10.1186/s13059-014-0550-8).

Installation

To install this package, start R (version "4.0") and enter:

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

BiocManager::install("DESeq2")
```

For older versions of R, please refer to the appropriate [Bioconductor release](#).

Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("DESeq2")
```

Details

biocViews	Bayesian , ChIPSeq , Clustering , DifferentialExpression , GeneExpression , Immunology , Immunology , Normalization , PrincipalComponent , RNASeq , Regression , Sequencing , Software , Transcription
Version	1.28.1
In Bioconductor since	BioC 2.12 (R-3.0) (7 years)

[HTML](#) [R Script](#) Analyzing RNA-seq data with DESeq2
[PDF](#) Reference Manual
[Text](#) NEWS

<https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html>

if/else conditionals

- Say we want to get the output of a function and depending on that output, perform one of two options
- For example, we only want to run the next step if the output is less than 100. We will need if/else logic, and will need to use operators

```
> y <- function(x)
```

```
> if (y < 100) {do this} else {do this}
```

- Let's break this down and go over some edge cases

> `if (y < 100) {do this} else {do this}`

- Structure
 - The statement being evaluated is in parantheses
 - The task to run if the statement is true is in curly brackets
 - The syntax of how to do this will differ language to language
 - “else” covers all leftover options – this can be left blank

Operators

- >: greater than
- <: less than
- >=: greater than or equal to
- <=: less than or equal to
- ==: equal to
- !=: not equal to
- &: and
- |: or

- What if there are several options you want to encode?

```
> if (y < 100) {  
    do this  
} else if (y > 100 & y <= 200) {  
    do this  
} else if (y > 200 & y <= 300) {  
    do this  
} else {  
    do this  
}
```

For loops

- Used to repeat a task for a defined list of values

```
> A <- c("Hello", "world", "!")  
> for (x in A) {print(x)}
```

- This can be combined with conditions

```
> for (x in A) {  
  if (nchar(x) > 2) {  
    print(x)  
  }  
}
```

- Thought exercise: What will this print out?

While loops

- What if we don't know how many iterations we want to run?

```
> y <- 0      # We initialize the counting variable here
> while (y < 10) {
  print("Hello")
  y <- y + 1
}
> print("Goodbye")
```

- However, this is flexible, if we changed the iterator to only add 0.5, would run 2x loops

```
> addone <- function(x) {  
    x <- x + 1  
    return(x)  
}  
> addone(1)  
> x
```

Variables set or called in function are only available for that function call, will not set or modify outside variables (usually)

What about multiple inputs?

```
> addtwo <- function(x, y) {  
  z <- (x + 1) / y  
  return(z)  
}  
> addtwo(1, 2)
```

If we want to save the output from the function, we need to set a new variable

```
> output <- addtwo(1, 2)  
> print(output)
```

What about multiple outputs?

```
> multitask <- function(x) {  
  if (x > 100) {  
    x <- x / 2  
    y <- "large number"  
  } else {  
    y <- "small number"  
  }  
  output_array <- c(x, y)  
  return(output_array)  
}  
> multitask(1)  
> multitask(101)
```

How about saving multiple outputs?

- R only allows for returning one object from a function, so we have to pass out an array of all the output values

```
> outputs <- multitask(101)
```

```
> outputs[1]
```

```
> outputs[2]
```

Writing a library of your own functions

- Save functions to a file ending in “.r”
- Run:

```
> setwd("~/Desktop/")           # or where ever the hello.r script is
> source("custom.r")
> sayHello()
```

Homework

- Write an R script with the following functions (a separate function for each task):
 1. Print “Hello world”
 2. Print “Goodbye world”
 3. Take a user input and add 5. If the output is over 10, return 10, else return the modified input value
 4. Take a user input and print “100” if input is less than 100, “200” if input is between 100 and 200, “300” if input is between 200 and 300, or “other” for anything else
 5. Print a list of values, each time pre-pended with “value #:” where # is the position in the list of that value being printed
- Run these functions with various inputs that will give all possible outputs and save these commands to a txt file.