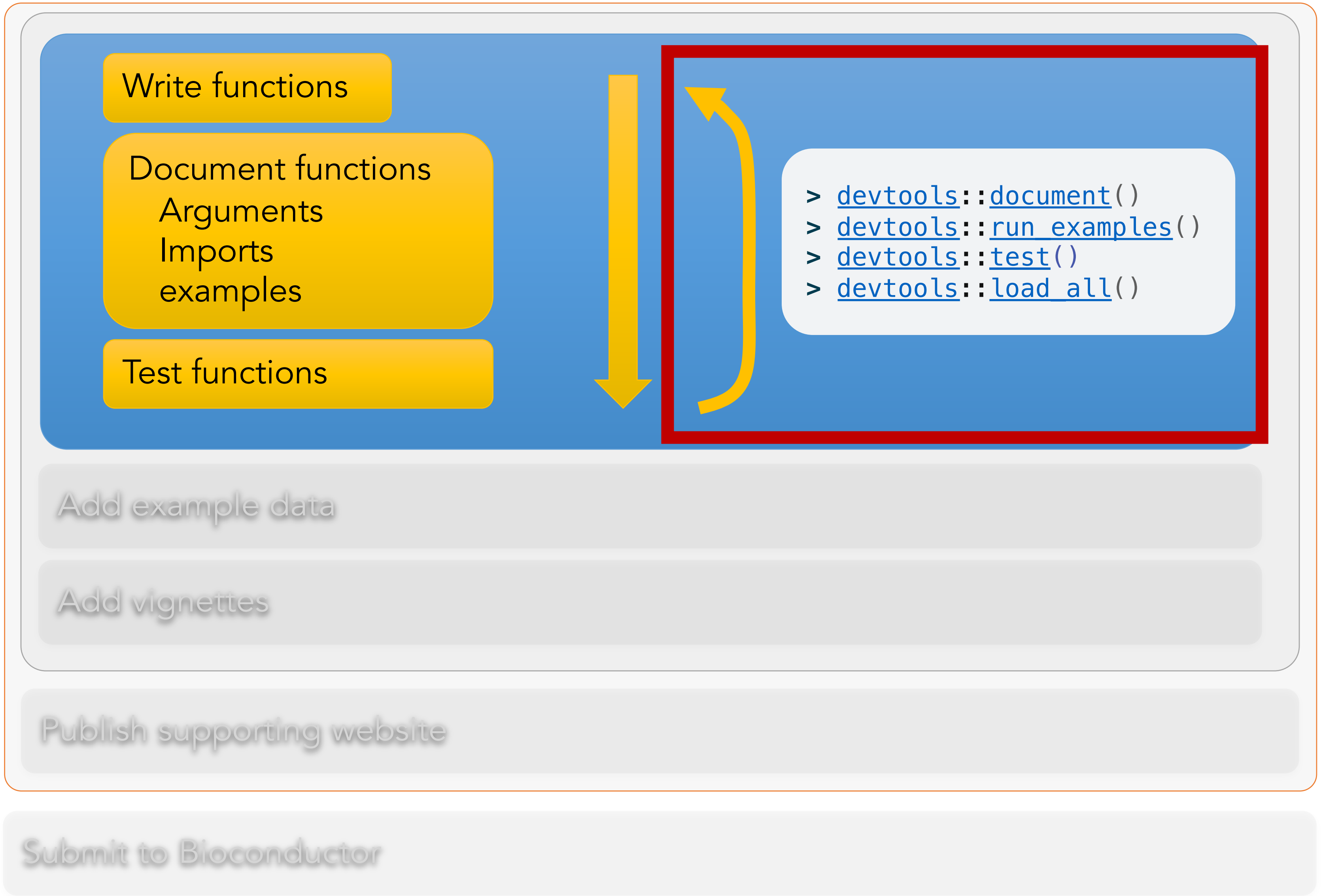
Improving package:
Support for (raw) data, vignettes

Physalia course 2023

Instructor: Jacques Serizay

Standard package content



Write functions

Document functions
- Arguments
- Imports
- examples

Test functions

```
> devtools::document()
> devtools::run_examples()
> devtools::test()
> devtools::load_all()
```

Add example data

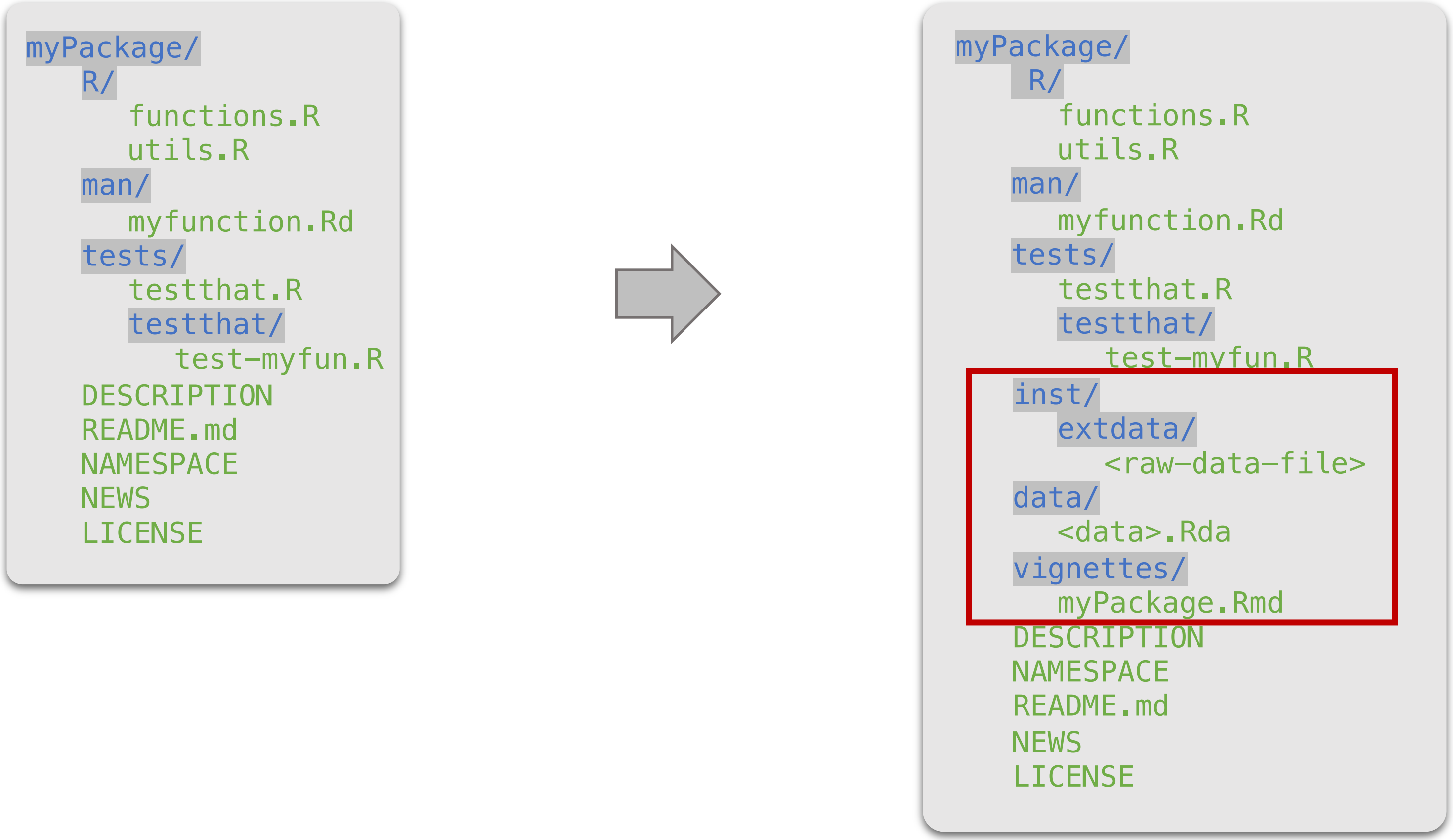
Add vignettes

Publish supporting website

Submit to Bioconductor

```
myPackage/
  R/
    functions.R
    utils.R
  man/
    myfunction.Rd
  tests/
    testthat.R
    testthat/
      test-myfun.R
  DESCRIPTION
  README.md
  NAMESPACE
  NEWS
  LICENSE
```

Standard package content



```
myPackage/
   R/
      functions.R
      utils.R
   man/
      myfunction.Rd
   tests/
      testthat.R
      testthat/
         test-myfun.R
   DESCRIPTION
   README.md
   NAMESPACE
   NEWS
   LICENSE
```

```
myPackage/
   R/
      functions.R
      utils.R
   man/
      myfunction.Rd
   tests/
      testthat.R
      testthat/
         test-myfun.R
   inst/
      extdata/
         <raw-data-file>
   data/
      <data>.Rda
   vignettes/
      myPackage.Rmd
   DESCRIPTION
   NAMESPACE
   README.md
   NEWS
   LICENSE
```

Why providing data in your package?

Data shipped with your package are meant to:

1. Provide a means to run examples and demonstrate package functionalities in vignettes;

2. Directly enable analysis (in “data” packages)

Adding data to package

2 types of ”data”:

- Raw: e.g. genomic files (bed, bigwig, bam, …) or other (tables, text files, …)
- Processed: `.Rda `files, containing R objects to be loaded in memory in R.

Package size limits

Watch out! Your package should be < 5Mb. Genomic files (particularly) can expand in size very quickly.

Be cautious!

Package size limits

Watch out! Your package should be < 5Mb. Genomic files (particularly) can expand in size very quickly. Be cautious!

This is a rather strict requirement for Bioconductor. Under only very few specific circumstances will the BioC core members allow you to exceed this (e.g. developing specific packages coordinated by the BioC team itself).

Package size limits

Watch out! Your package should be < 5Mb. Genomic files (particularly) can expand in size very quickly. Be cautious!

This is a rather strict requirement for Bioconductor. Under only very few specific circumstances will the BioC core members allow you to exceed this (e.g. developing specific packages coordinated by the BioC team itself).

<u>Unrelated but still worth mentioning:</u>

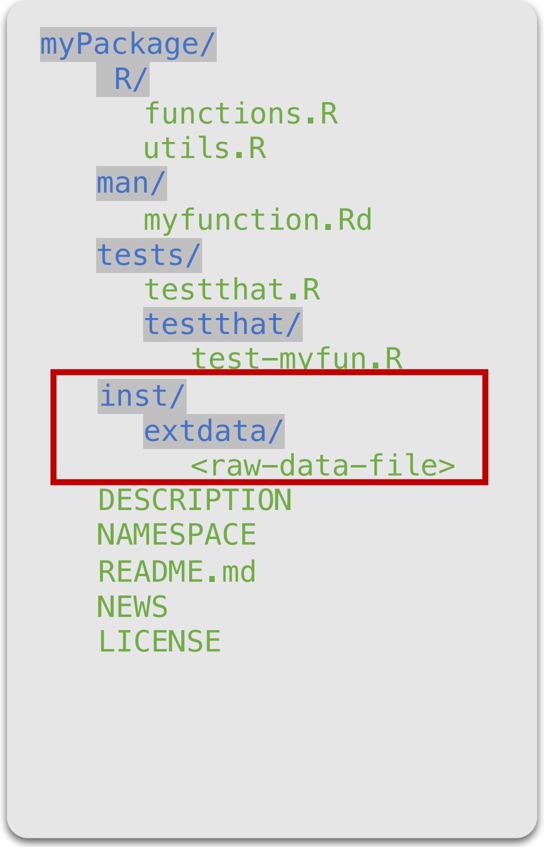
Don’t forget, git never forgets! If you add a dataset and commit/push it to your git repo, it will stay there forever. Even after deleting it, it will still be in your `. git` local folder and in git memory (because you should be able to recover it back, since everything is reversible in git). This usually results in enormous `. git` folders… Watch out for storage space!

Raw data

Raw data can be virtually any file, but it has to be relevant for the package development.

The main reason to include such files is when a key part of a package’s functionality is to act on an external file (e.g. `readr`, `vroom`, …).

Raw data is stored in `inst/extdata`.



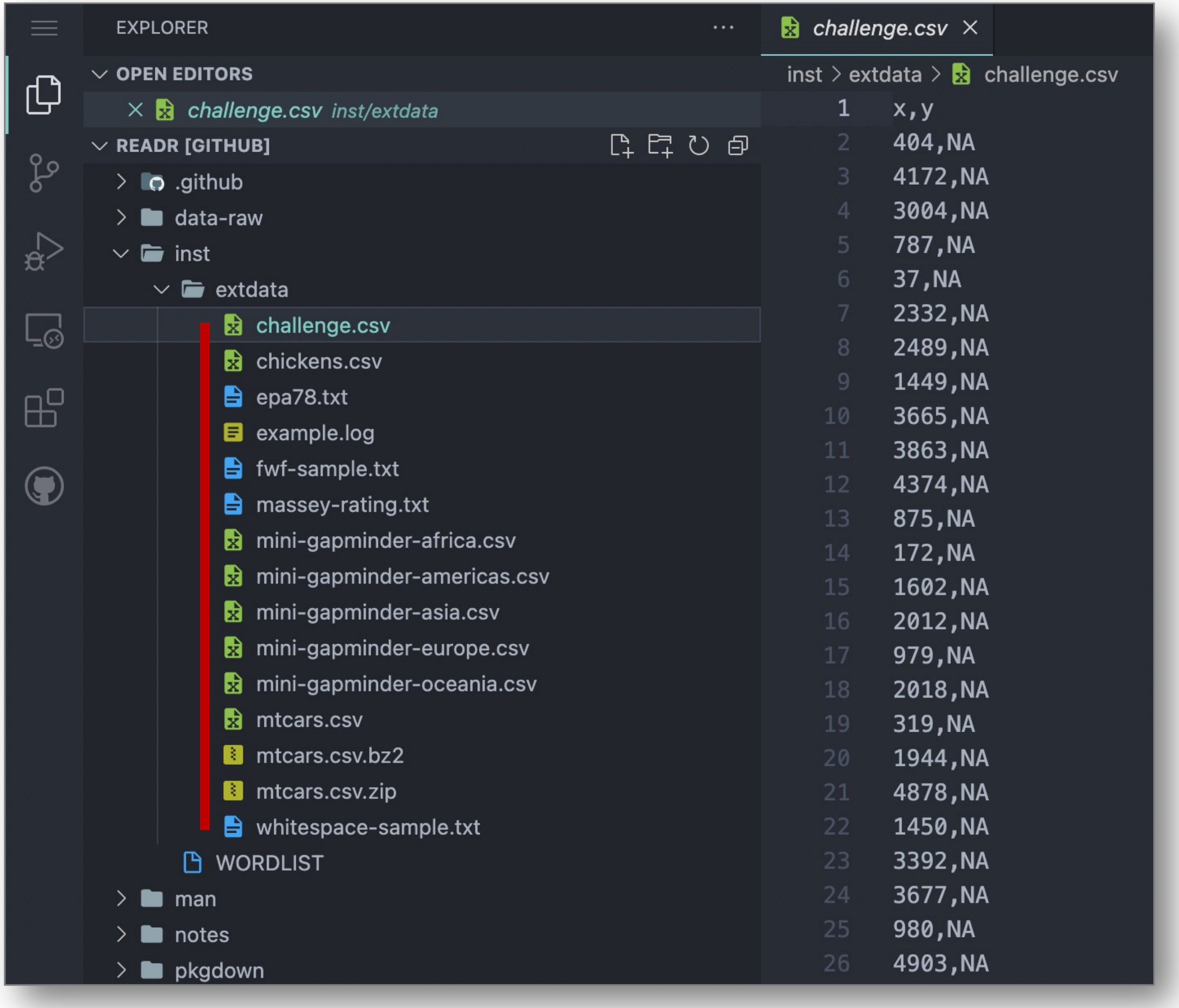
```
myPackage/
   R/
     functions.R
     utils.R
   man/
     myfunction.Rd
   tests/
     testthat.R
     testthat/
       test-myfun.R
   inst/
     extdata/
       <raw-data-file>
   DESCRIPTION
   NAMESPACE
   README.md
   NEWS
   LICENSE
```

Raw data

Raw data can be virtually any file, but it has to be relevant for the package development.

The main reason to include such files is when a key part of a package's functionality is to act on an external file (e.g. `readr`, `vroom`, ...).

Raw data is stored in `inst/extdata`.

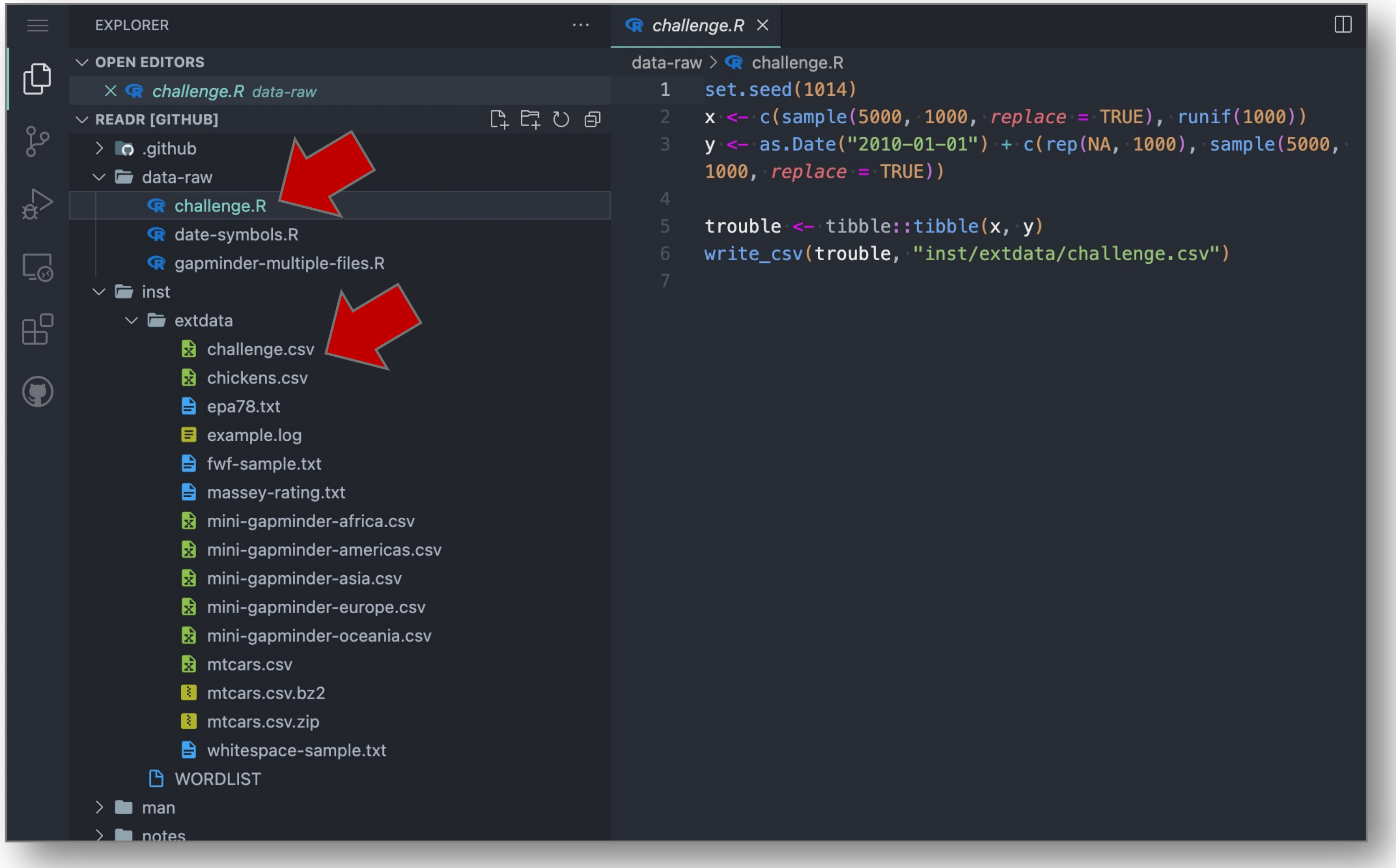


```
x,y
404,NA
4172,NA
3004,NA
787,NA
37,NA
2332,NA
2489,NA
1449,NA
3665,NA
3863,NA
4374,NA
875,NA
172,NA
1602,NA
2012,NA
979,NA
2018,NA
319,NA
1944,NA
4878,NA
1450,NA
3392,NA
3677,NA
980,NA
4903,NA
```

Raw data

It is good practice to document how raw data files were generated.

You can do it by adding a `data-raw` folder to the root of your package and add R files describing how the data was created.



```r
set.seed(1014)
x <- c(sample(5000, 1000, replace = TRUE), runif(1000))
y <- as.Date("2010-01-01") + c(rep(NA, 1000), sample(5000,
1000, replace = TRUE))

trouble <- tibble::tibble(x, y)
write_csv(trouble, "inst/extdata/challenge.csv")

```

Raw data

It is good practice to document how raw data files were generated.

You can do it by adding a `data-raw` folder to the root of your package and add R files describing how the data was created.

Of note, Bioconductor specifically prefers these files to be located in `inst/scripts/`: https://contributions.bioconductor.org/docs.html#doc-inst-script.

Use raw data

Raw data should generally not be read by the end-user.

Instead, it is a way for the package writer to describe how to import files.

Use raw data

Raw data should generally not be read by the end-user.

Instead, it is a way for the package writer to describe how to import files.

For this reason, raw data is generally only a **toy dataset**, a **small subset** of an actual dataset (*e.g.* only a single chromosome out of a whole genome, etc.)

Use raw data

Raw data should generally not be read by the end-user.

Instead, it is a way for the package writer to describe how to import files.

For this reason, raw data is generally only a **toy dataset**, a **small subset** of an actual dataset (*e.g.* only a single chromosome out of a whole genome, etc.).

For this reason as well, if your package deals with already existing file formats (e.g. bed, bam, bigwig, …), many BioC core packages (e.g. GenomicRanges, rtracklayer, Biostrings, ...) **already provide toy datasets!!**

Hosting all these raw data files has a cost (economical and *environmental*). Please, do check whether core packages can provide the type of files you'd need as a toy dataset.

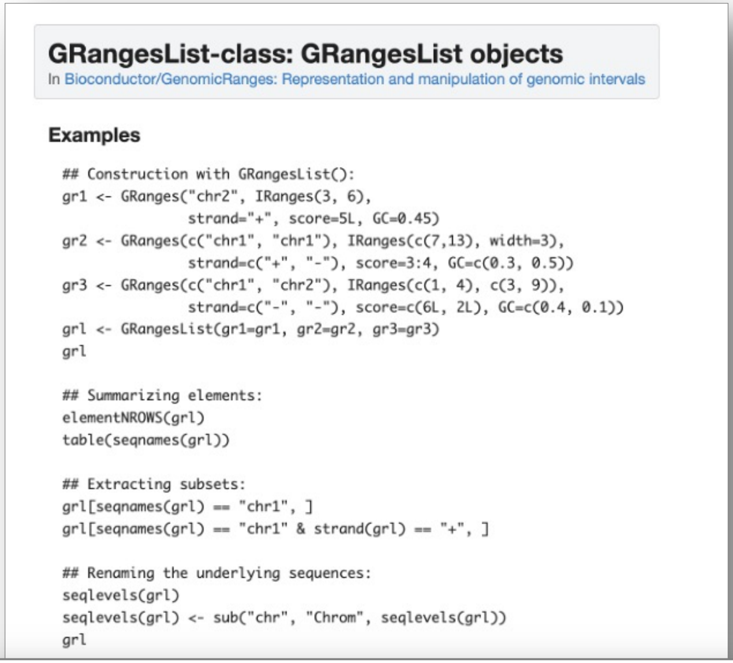
Use raw data

The package writer can have access to their (or other packages’) raw data files using `system.file()`:

```
## List raw data files shipped with GenomicRanges
> system.file('extdata', package = 'GenomicRanges') |> list.files()
[1] "feature_frags.txt”

## Get full path to "feature_frags.txt”
> system.file('extdata', "feature_frags.txt", package = 'GenomicRanges')
[1] "/Users/jacques/Library/R/arm64/4.3/library/GenomicRanges/extdata/feature_frags.txt"
```

Use raw data

The package writer can have access to their (or other packages') raw data files using `system.file()`:

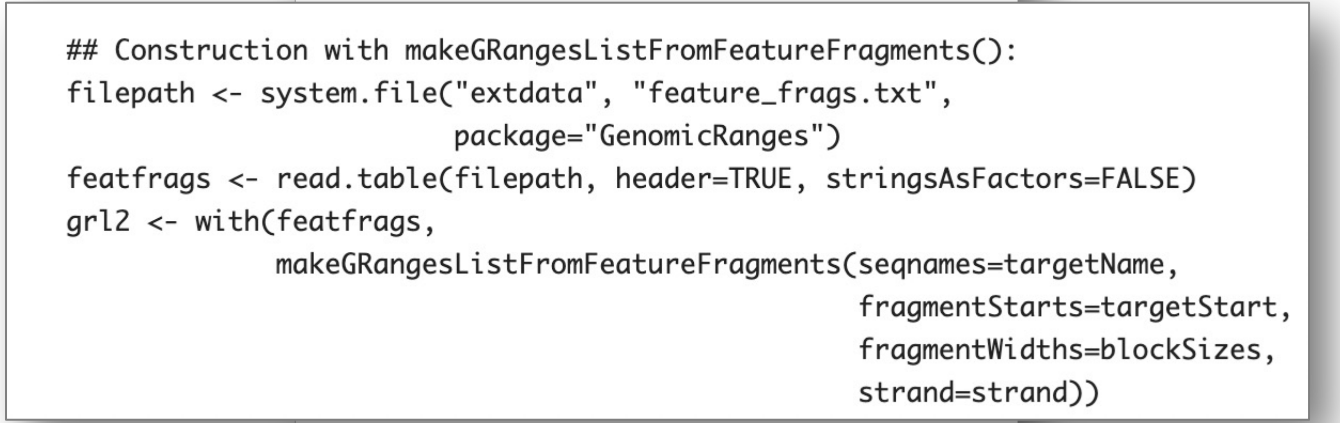


GRangesList-class: GRangesList objects

In Bioconductor/GenomicRanges: Representation and manipulation of genomic intervals

Examples

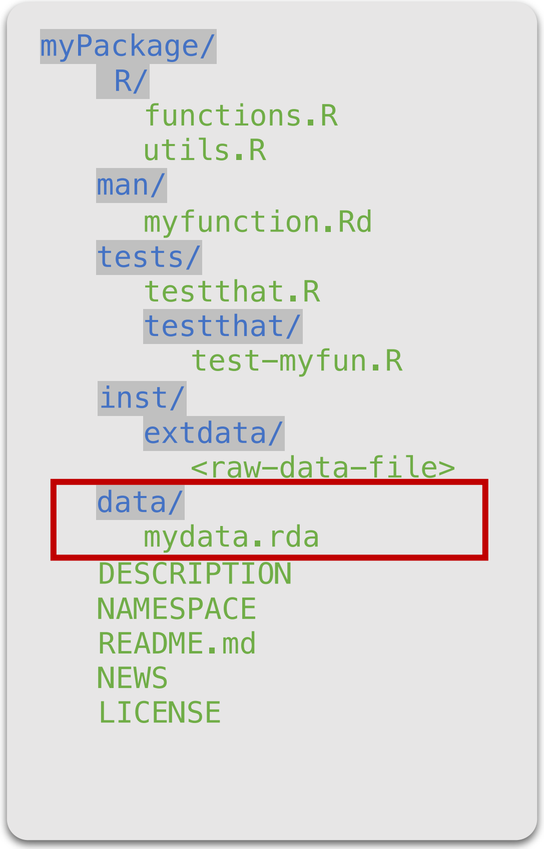
```
## Construction with GRangesList():
gr1 <- GRanges("chr2", IRanges(3, 6),
               strand="+", score=5L, GC=0.45)
gr2 <- GRanges(c("chr1", "chr1"), IRanges(c(7,13), width=3),
               strand=c("+", "-"), score=3:4, GC=c(0.3, 0.5))
gr3 <- GRanges(c("chr1", "chr2"), IRanges(c(1, 4), c(3, 9)),
               strand=c("-", "-"), score=c(6L, 2L), GC=c(0.4, 0.1))
grl <- GRangesList(gr1=gr1, gr2=gr2, gr3=gr3)
grl

## Summarizing elements:
elementNROWS(grl)
table(seqnames(grl))

## Extracting subsets:
grl[seqnames(grl) == "chr1", ]
grl[seqnames(grl) == "chr1" & strand(grl) == "+", ]

## Renaming the underlying sequences:
seqlevels(grl)
seqlevels(grl) <- sub("chr", "Chrom", seqlevels(grl))
grl
```



```
## Construction with makeGRangesListFromFeatureFragments():
filepath <- system.file("extdata", "feature_frags.txt",
                        package="GenomicRanges")
featfrags <- read.table(filepath, header=TRUE, stringsAsFactors=FALSE)
grl2 <- with(featfrags,
             makeGRangesListFromFeatureFragments(seqnames=targetName,
                                                 fragmentStarts=targetStart,
                                                 fragmentWidths=blockSizes,
                                                 strand=strand))
```

Processed data

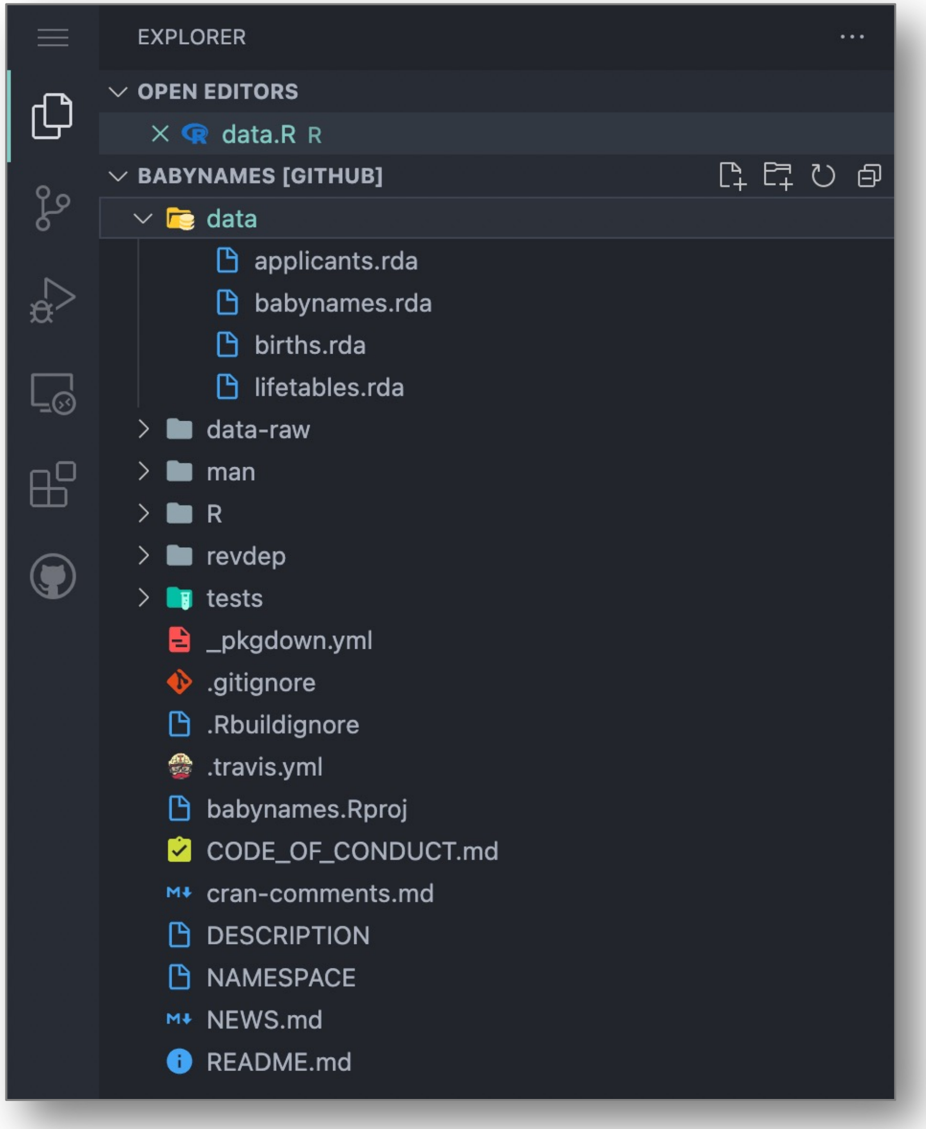
Processed data are stored in the `data/` folder.



```
myPackage/
    R/
      functions.R
      utils.R
    man/
      myfunction.Rd
    tests/
      testthat.R
      testthat/
        test-myfun.R
    inst/
      extdata/
        <raw-data-file>
    data/
      mydata.rda
    DESCRIPTION
    NAMESPACE
    README.md
    NEWS
    LICENSE
```

Processed data

Processed data are stored in the `data/` folder.

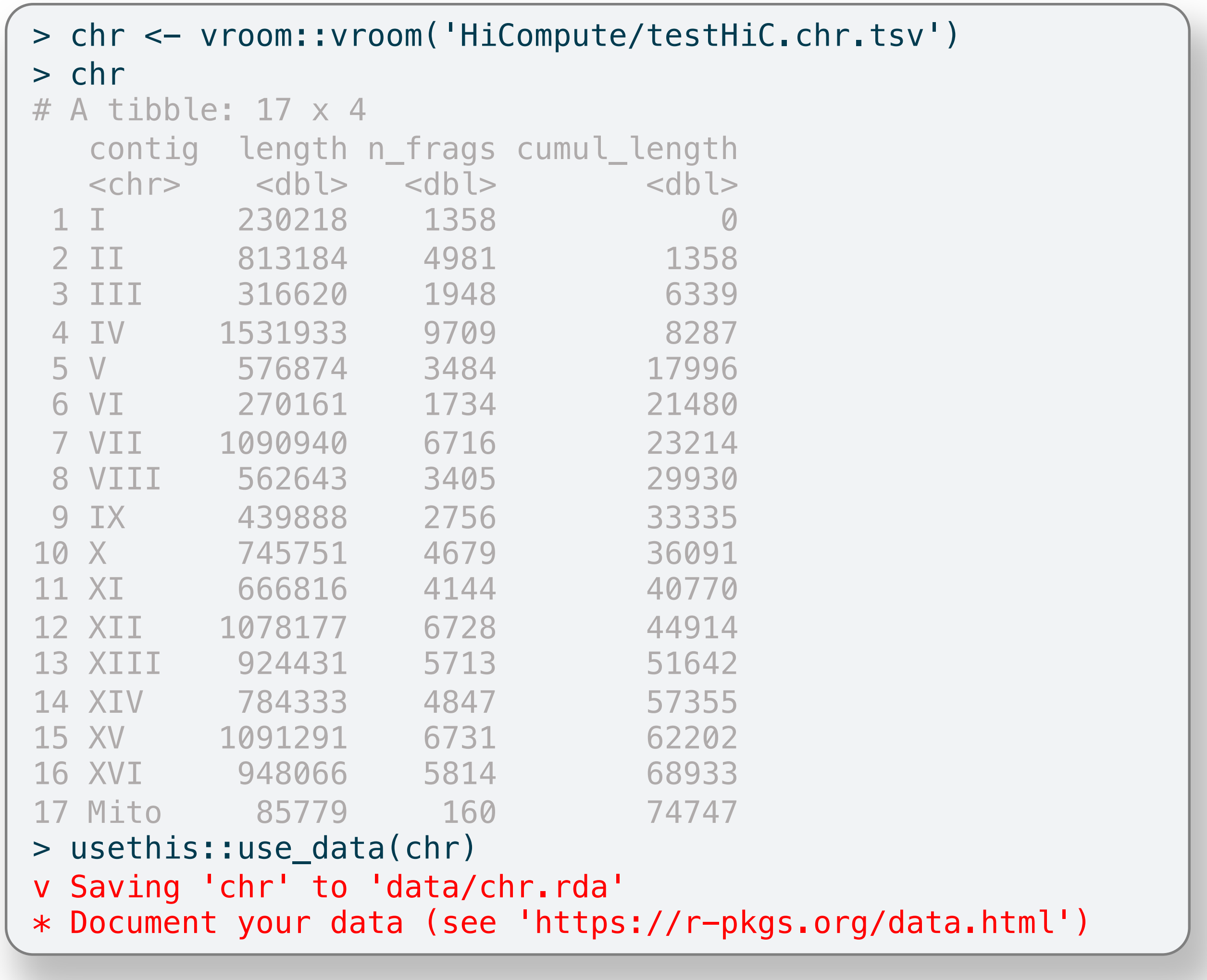


```
EXPLORER                                 ...
OPEN EDITORS
   × data.R R
BABYNAMES [GITHUB]
  data
      applicants.rda
      babynames.rda
      births.rda
      lifetables.rda
  data-raw
  man
  R
  revdep
  tests
  _pkgdown.yml
  .gitignore
  .Rbuildignore
  .travis.yml
  babynames.Rproj
  CODE_OF_CONDUCT.md
  cran-comments.md
  DESCRIPTION
  NAMESPACE
  NEWS.md
  README.md
```

Processed data

Processed data are stored in the `data/` folder.

The best way to provide processed data in your package is through `usethis::use_data()`.



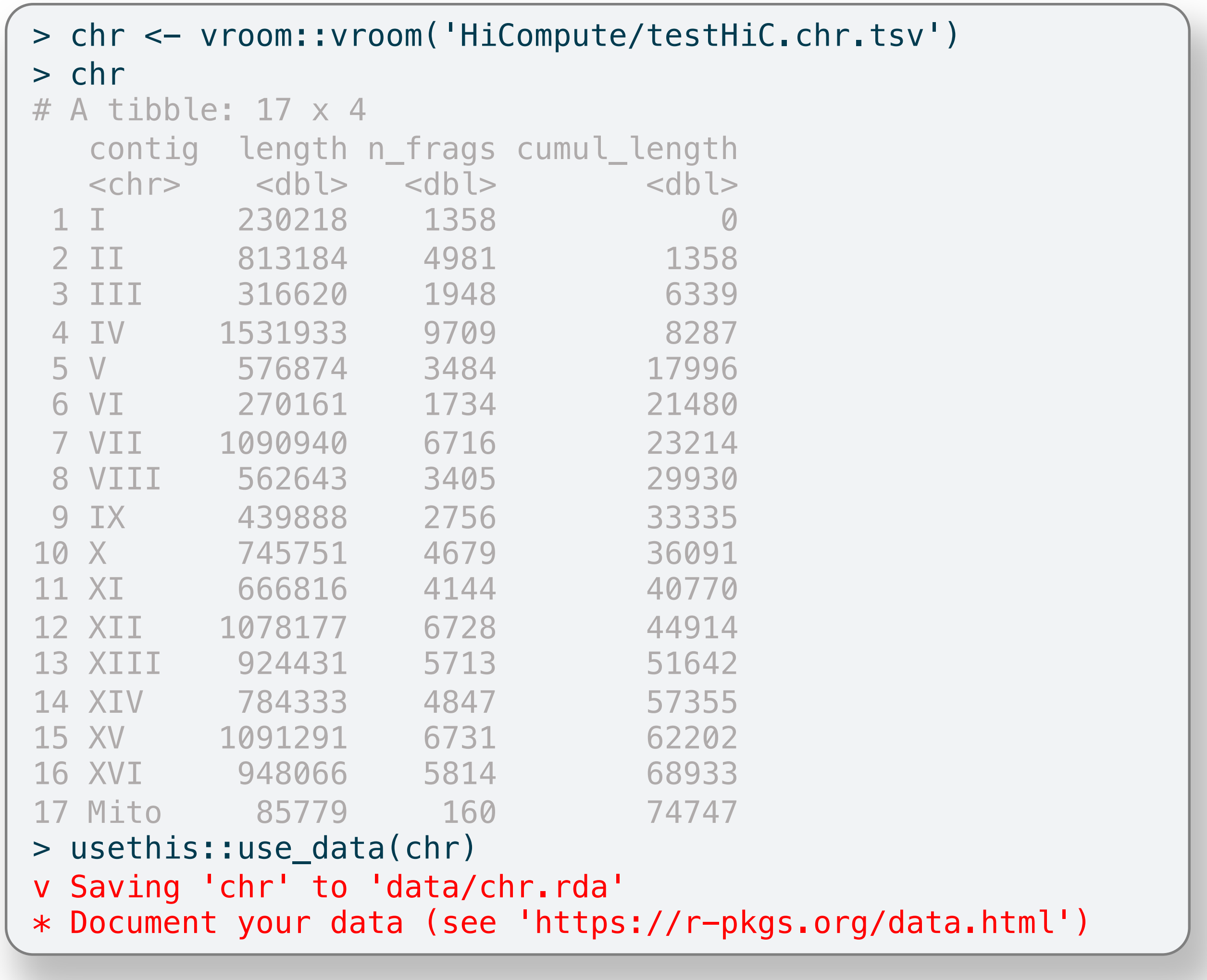
```
> chr <- vroom::vroom('HiCompute/testHiC.chr.tsv')
> chr
# A tibble: 17 x 4
   contig  length n_frags cumul_length
   <chr>    <dbl>   <dbl>        <dbl>
 1 I       230218    1358            0
 2 II      813184    4981         1358
 3 III     316620    1948         6339
 4 IV     1531933    9709         8287
 5 V       576874    3484        17996
 6 VI      270161    1734        21480
 7 VII    1090940    6716        23214
 8 VIII    562643    3405        29930
 9 IX      439888    2756        33335
10 X       745751    4679        36091
11 XI      666816    4144        40770
12 XII    1078177    6728        44914
13 XIII    924431    5713        51642
14 XIV     784333    4847        57355
15 XV     1091291    6731        62202
16 XVI     948066    5814        68933
17 Mito     85779     160        74747
> usethis::use_data(chr)
v Saving 'chr' to 'data/chr.rda'
* Document your data (see 'https://r-pkgs.org/data.html')
```

Processed data

Processed data are stored in the `data/` folder.

The best way to provide processed data in your package is through `usethis::use_data()`.

Avoid at all costs creating the `data/` folder yourself. You should be able to use `usethis::use_data()` instead.

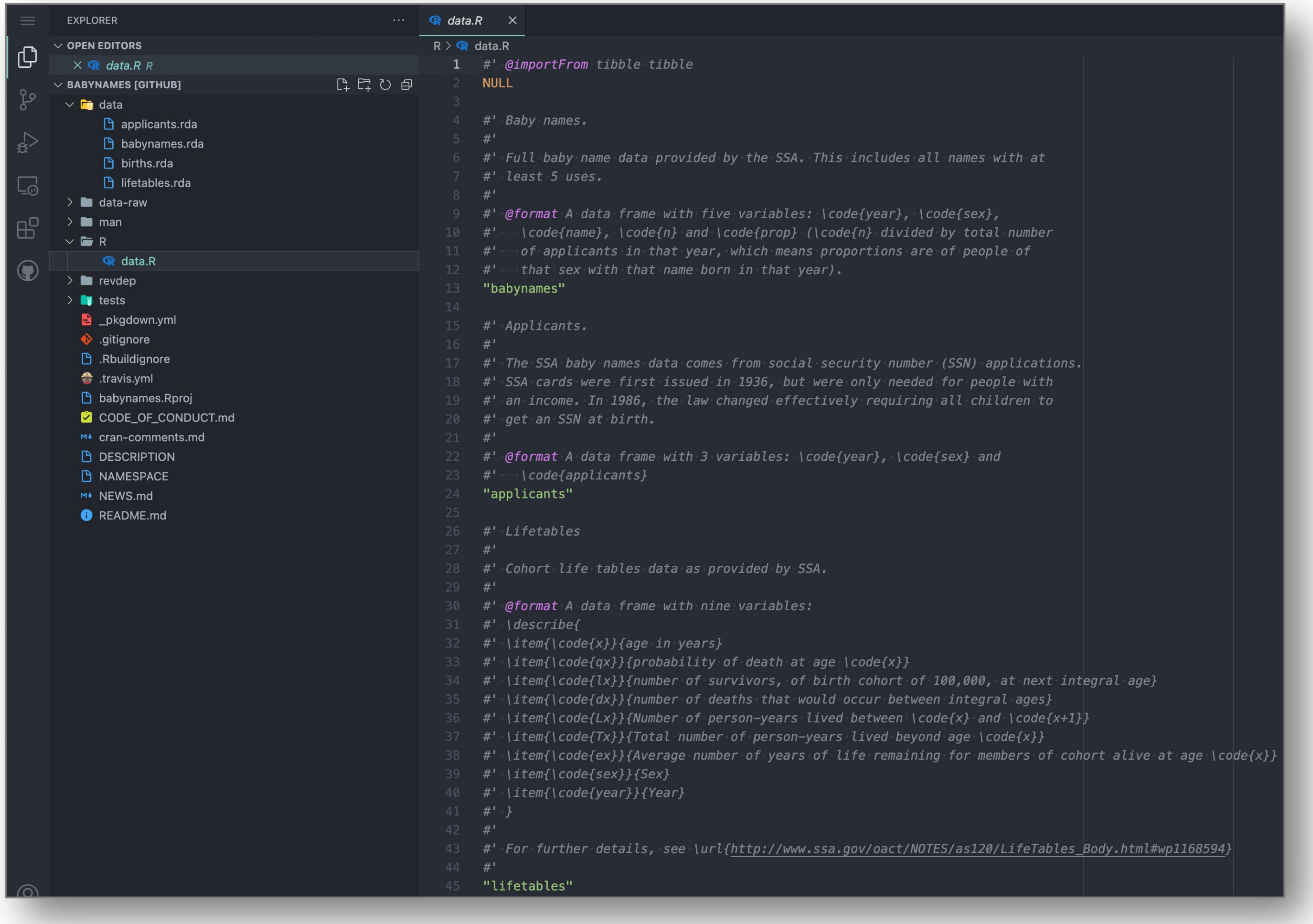


```
> chr <- vroom::vroom('HiCompute/testHiC.chr.tsv')
> chr
# A tibble: 17 x 4
   contig  length n_frags cumul_length
   <chr>    <dbl>   <dbl>        <dbl>
 1 I       230218    1358            0
 2 II      813184    4981         1358
 3 III     316620    1948         6339
 4 IV     1531933    9709         8287
 5 V       576874    3484        17996
 6 VI      270161    1734        21480
 7 VII    1090940    6716        23214
 8 VIII    562643    3405        29930
 9 IX      439888    2756        33335
10 X       745751    4679        36091
11 XI      666816    4144        40770
12 XII    1078177    6728        44914
13 XIII    924431    5713        51642
14 XIV     784333    4847        57355
15 XV     1091291    6731        62202
16 XVI     948066    5814        68933
17 Mito     85779     160        74747
> usethis::use_data(chr)
v Saving 'chr' to 'data/chr.rda'
* Document your data (see 'https://r-pkgs.org/data.html')
```

Processed data

Like anything else in your package, your data should be documented.

The recommended way is to do that in a `R/data.R` file

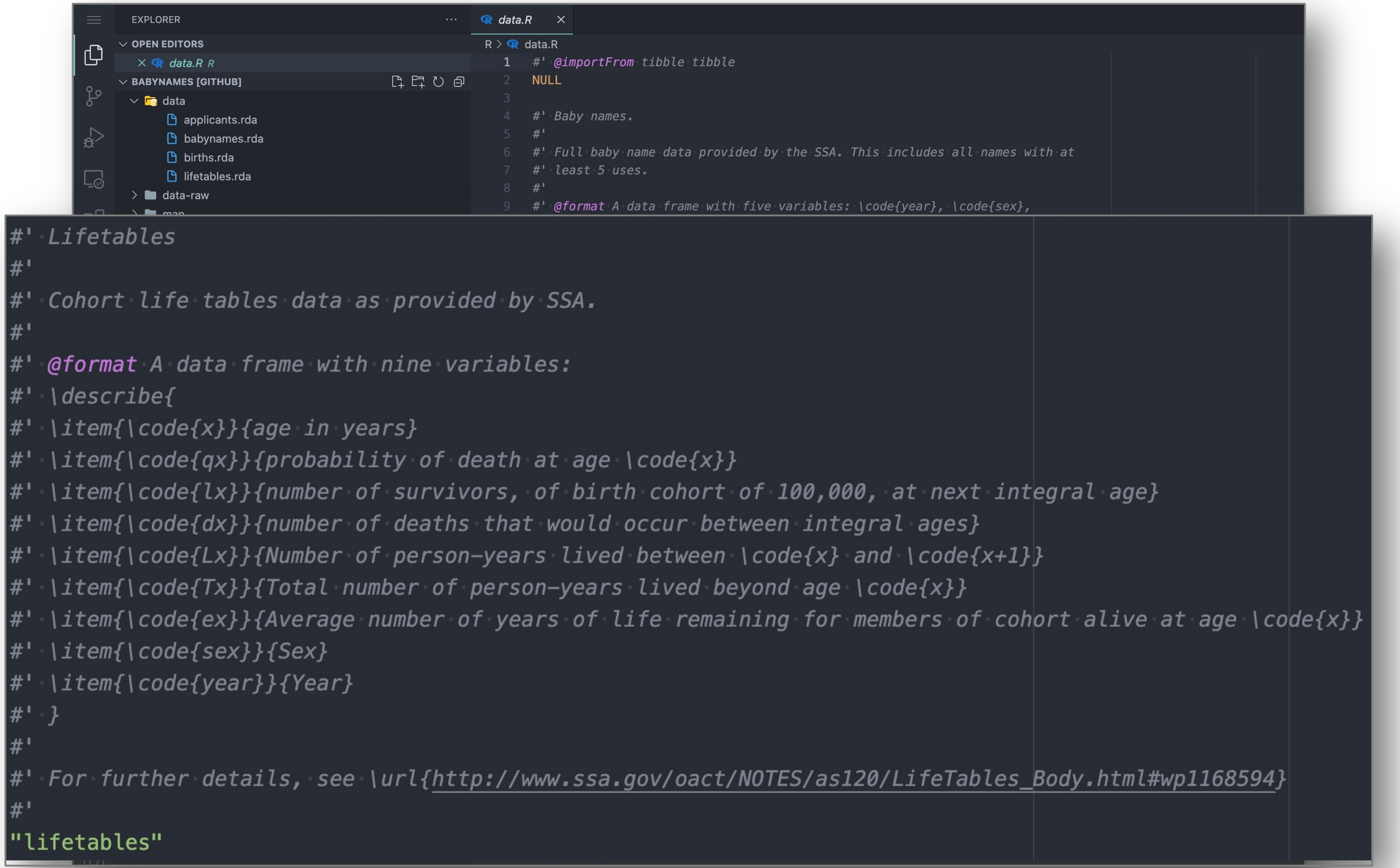


```r
#' @importFrom tibble tibble
NULL

#' Baby names.
#'
#' Full baby name data provided by the SSA. This includes all names with at
#' least 5 uses.
#'
#' @format A data frame with five variables: \code{year}, \code{sex},
#'   \code{name}, \code{n} and \code{prop} (\code{n} divided by total number
#'   of applicants in that year, which means proportions are of people of
#'   that sex with that name born in that year).
"babynames"

#' Applicants.
#'
#' The SSA baby names data comes from social security number (SSN) applications.
#' SSA cards were first issued in 1936, but were only needed for people with
#' an income. In 1986, the law changed effectively requiring all children to
#' get an SSN at birth.
#'
#' @format A data frame with 3 variables: \code{year}, \code{sex} and
#'   \code{applicants}
"applicants"

#' Lifetables
#'
#' Cohort life tables data as provided by SSA.
#'
#' @format A data frame with nine variables:
#' \describe{
#' \item{\code{x}}{age in years}
#' \item{\code{qx}}{probability of death at age \code{x}}
#' \item{\code{lx}}{number of survivors, of birth cohort of 100,000, at next integral age}
#' \item{\code{dx}}{number of deaths that would occur between integral ages}
#' \item{\code{Lx}}{Number of person-years lived between \code{x} and \code{x+1}}
#' \item{\code{Tx}}{Total number of person-years lived beyond age \code{x}}
#' \item{\code{ex}}{Average number of years of life remaining for members of cohort alive at age \code{x}}
#' \item{\code{sex}}{Sex}
#' \item{\code{year}}{Year}
#' }
#'
#' For further details, see \url{http://www.ssa.gov/oact/NOTES/as120/LifeTables_Body.html#wp1168594}
#'
"lifetables"
```

Processed data

Like anything else in your package, your data should be documented.

The recommended way is to do that in a `R/data.R` file

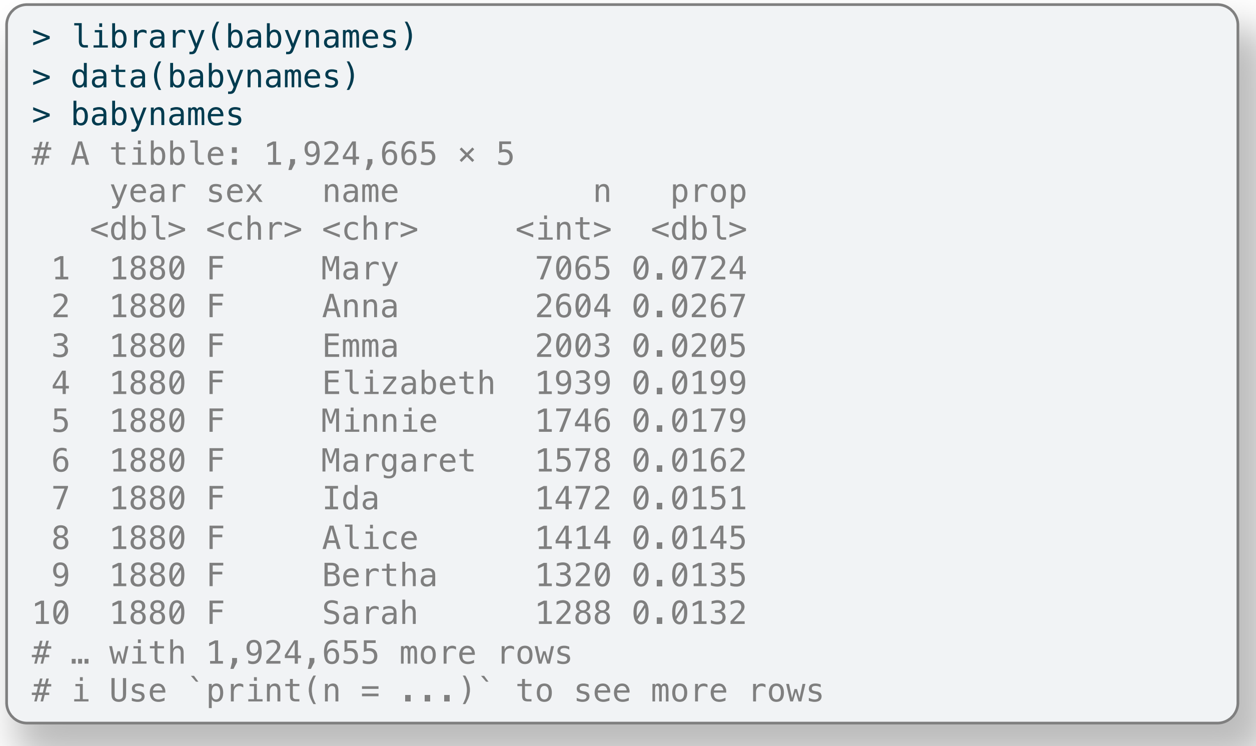


```r
#' @importFrom tibble tibble
NULL

#' Baby names.
#'
#' Full baby name data provided by the SSA. This includes all names with at
#' least 5 uses.
#'
#' @format A data frame with five variables: \code{year}, \code{sex},
```

```r
#' Lifetables
#'
#' Cohort life tables data as provided by SSA.
#'
#' @format A data frame with nine variables:
#' \describe{
#' \item{\code{x}}{age in years}
#' \item{\code{qx}}{probability of death at age \code{x}}
#' \item{\code{lx}}{number of survivors, of birth cohort of 100,000, at next integral age}
#' \item{\code{dx}}{number of deaths that would occur between integral ages}
#' \item{\code{Lx}}{Number of person-years lived between \code{x} and \code{x+1}}
#' \item{\code{Tx}}{Total number of person-years lived beyond age \code{x}}
#' \item{\code{ex}}{Average number of years of life remaining for members of cohort alive at age \code{x}}
#' \item{\code{sex}}{Sex}
#' \item{\code{year}}{Year}
#' }
#'
#' For further details, see \url{http://www.ssa.gov/oact/NOTES/as120/LifeTables_Body.html#wp1168594}
#'
"lifetables"
```

Processed data

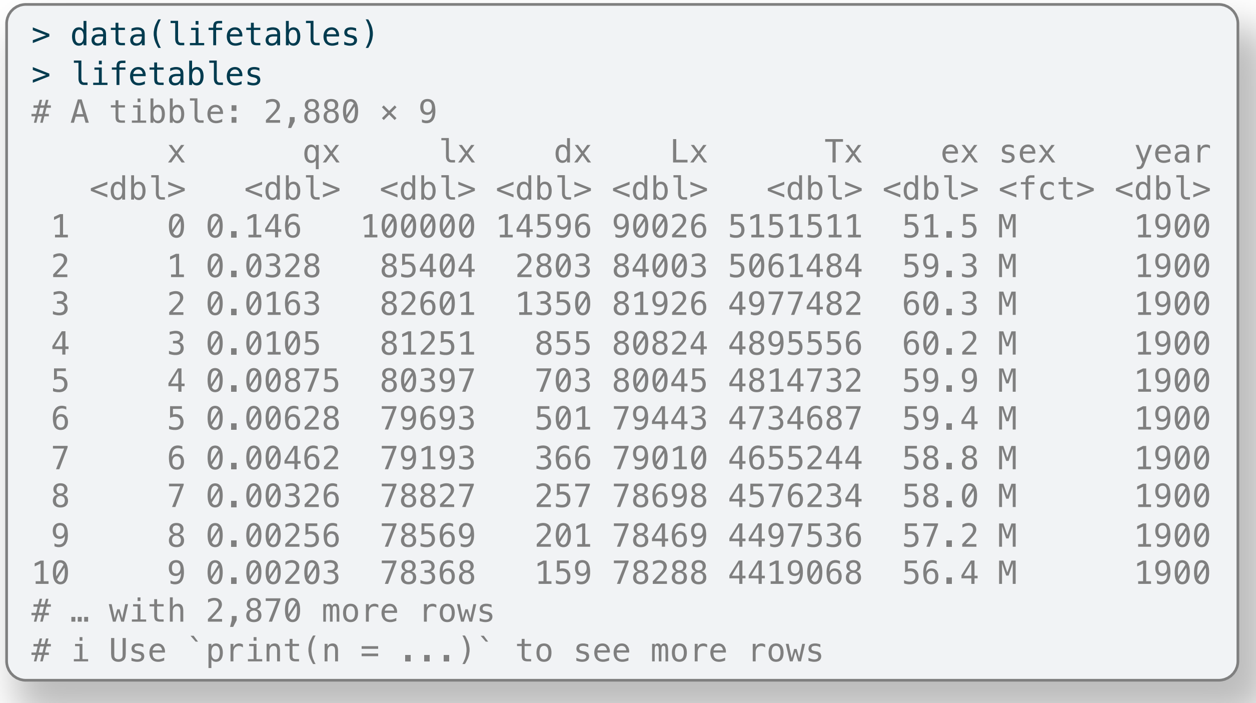
Processed data are made readily available to your package end-users through the `data()` function.



```
> library(babynames)
> data(babynames)
> babynames
# A tibble: 1,924,665 × 5
    year sex   name           n   prop
   <dbl> <chr> <chr>      <int>  <dbl>
 1  1880 F     Mary        7065 0.0724
 2  1880 F     Anna        2604 0.0267
 3  1880 F     Emma        2003 0.0205
 4  1880 F     Elizabeth   1939 0.0199
 5  1880 F     Minnie      1746 0.0179
 6  1880 F     Margaret    1578 0.0162
 7  1880 F     Ida         1472 0.0151
 8  1880 F     Alice       1414 0.0145
 9  1880 F     Bertha      1320 0.0135
10  1880 F     Sarah       1288 0.0132
# … with 1,924,655 more rows
# i Use `print(n = ...)` to see more rows
```

Processed data

Processed data are made readily available to your package end-users through the `data()` function.



```
> data(lifetables)
> lifetables
# A tibble: 2,880 × 9
       x      qx     lx    dx    Lx      Tx    ex sex    year
   <dbl>   <dbl>  <dbl> <dbl> <dbl>   <dbl> <dbl> <fct> <dbl>
 1     0 0.146   100000 14596 90026 5151511  51.5 M      1900
 2     1 0.0328   85404  2803 84003 5061484  59.3 M      1900
 3     2 0.0163   82601  1350 81926 4977482  60.3 M      1900
 4     3 0.0105   81251   855 80824 4895556  60.2 M      1900
 5     4 0.00875  80397   703 80045 4814732  59.9 M      1900
 6     5 0.00628  79693   501 79443 4734687  59.4 M      1900
 7     6 0.00462  79193   366 79010 4655244  58.8 M      1900
 8     7 0.00326  78827   257 78698 4576234  58.0 M      1900
 9     8 0.00256  78569   201 78469 4497536  57.2 M      1900
10     9 0.00203  78368   159 78288 4419068  56.4 M      1900
# … with 2,870 more rows
# i Use `print(n = ...)` to see more rows
```

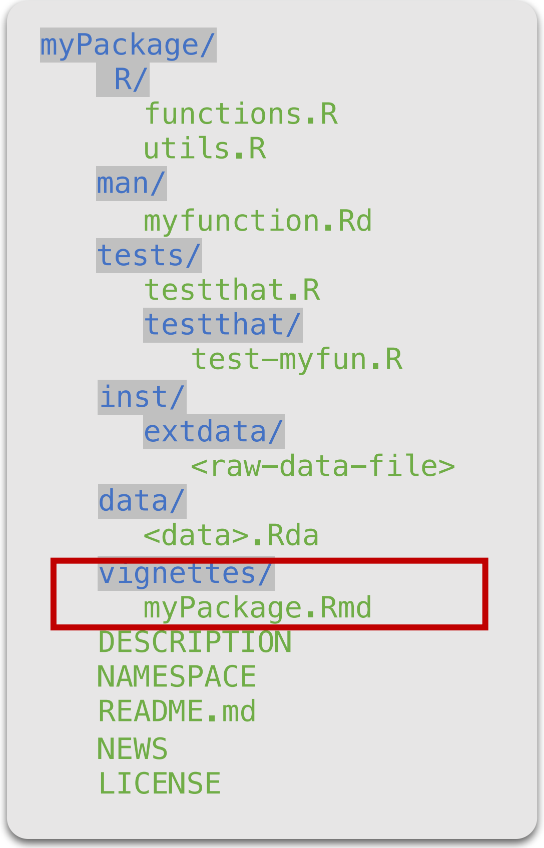
Vignettes

Vignettes are extensive (comprehensive) walkthrough of your package functionalities.

Vignettes

Vignettes are extensive (comprehensive) walkthrough of your package functionalities.

They live in the `vignettes/` folder (duh 🙄) .



```
myPackage/
    R/
      functions.R
      utils.R
    man/
      myfunction.Rd
    tests/
      testthat.R
      testthat/
        test-myfun.R
    inst/
      extdata/
        <raw-data-file>
    data/
      <data>.Rda
    vignettes/
      myPackage.Rmd
    DESCRIPTION
    NAMESPACE
    README.md
    NEWS
    LICENSE
```

Vignettes

Vignettes are extensive (comprehensive) walkthrough of your package functionalities.

They live in the `vignettes/` folder (duh 🙄) .

You can create a vignette boilerplate with either:

- usethis::use_vignette(’<YOUR-PACKAGE>’)
- biocthis::use_bioc_vignette(’<YOUR-PACKAGE>’)

Vignettes

Vignettes are extensive (comprehensive) walkthrough of your package functionalities.

They live in the `vignettes/` folder (duh 🙄) .

You can create a vignette boilerplate with either:

- usethis::use_vignette('<YOUR-PACKAGE>')
- biocthis::use_bioc_vignette('<YOUR-PACKAGE>')

You are not limited to a single vignette. However, you must provide at least one, which should be named <YOUR-PACKAGE>.Rmd.

Vignettes

Vignettes are extensive (comprehensive) walkthrough of your package functionalities.

They live in the `vignettes/` folder (duh 🙄) .

You can create a vignette boilerplate with either:

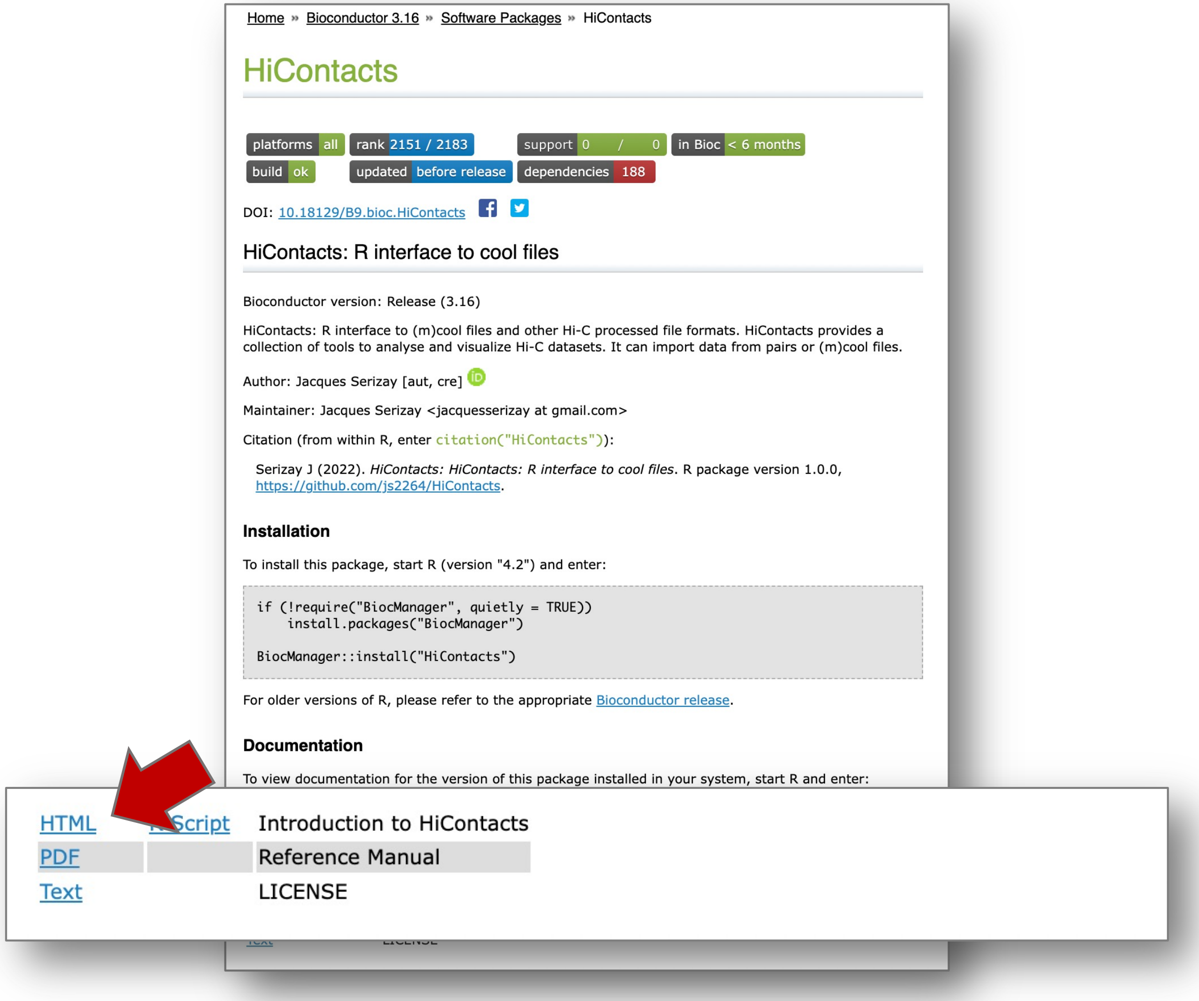
- usethis::use_vignette('<YOUR-PACKAGE>')
- biocthis::use_bioc_vignette('<YOUR-PACKAGE>')

You are not limited to a single vignette. However, you must provide at least one, which should be named <YOUR-PACKAGE>.Rmd.

Be as thorough as possible to describe all your package functionalities.

Vignettes

Once your package is accepted by BioC, your vignette will be compiled by the Bioconductor Single Package Builder into an HTML page, accessible through your package webpage.



Home » Bioconductor 3.16 » Software Packages » HiContacts

HiContacts

DOI: 10.18129/B9.bioc.HiContacts

HiContacts: R interface to cool files

Bioconductor version: Release (3.16)

HiContacts: R interface to (m)cool files and other Hi-C processed file formats. HiContacts provides a collection of tools to analyse and visualize Hi-C datasets. It can import data from pairs or (m)cool files.

Author: Jacques Serizay [aut, cre]

Maintainer: Jacques Serizay <jacquesserizay at gmail.com>

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

Serizay J (2022). *HiContacts: HiContacts: R interface to cool files*. R package version 1.0.0, https://github.com/js2264/HiContacts.

Installation

To install this package, start R (version "4.2") and enter:

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

BiocManager::install("HiContacts")
```

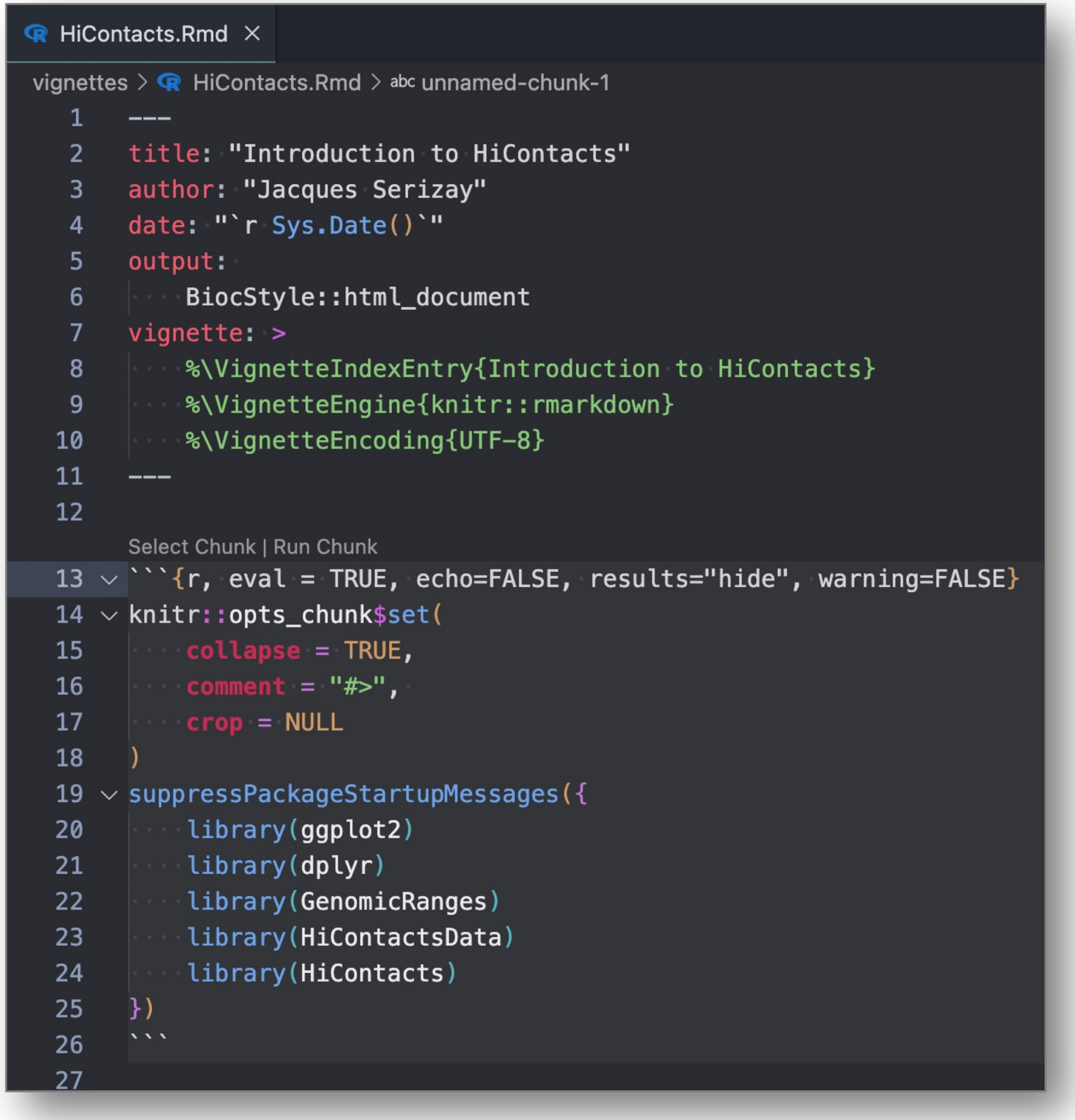
For older versions of R, please refer to the appropriate Bioconductor release.

Documentation

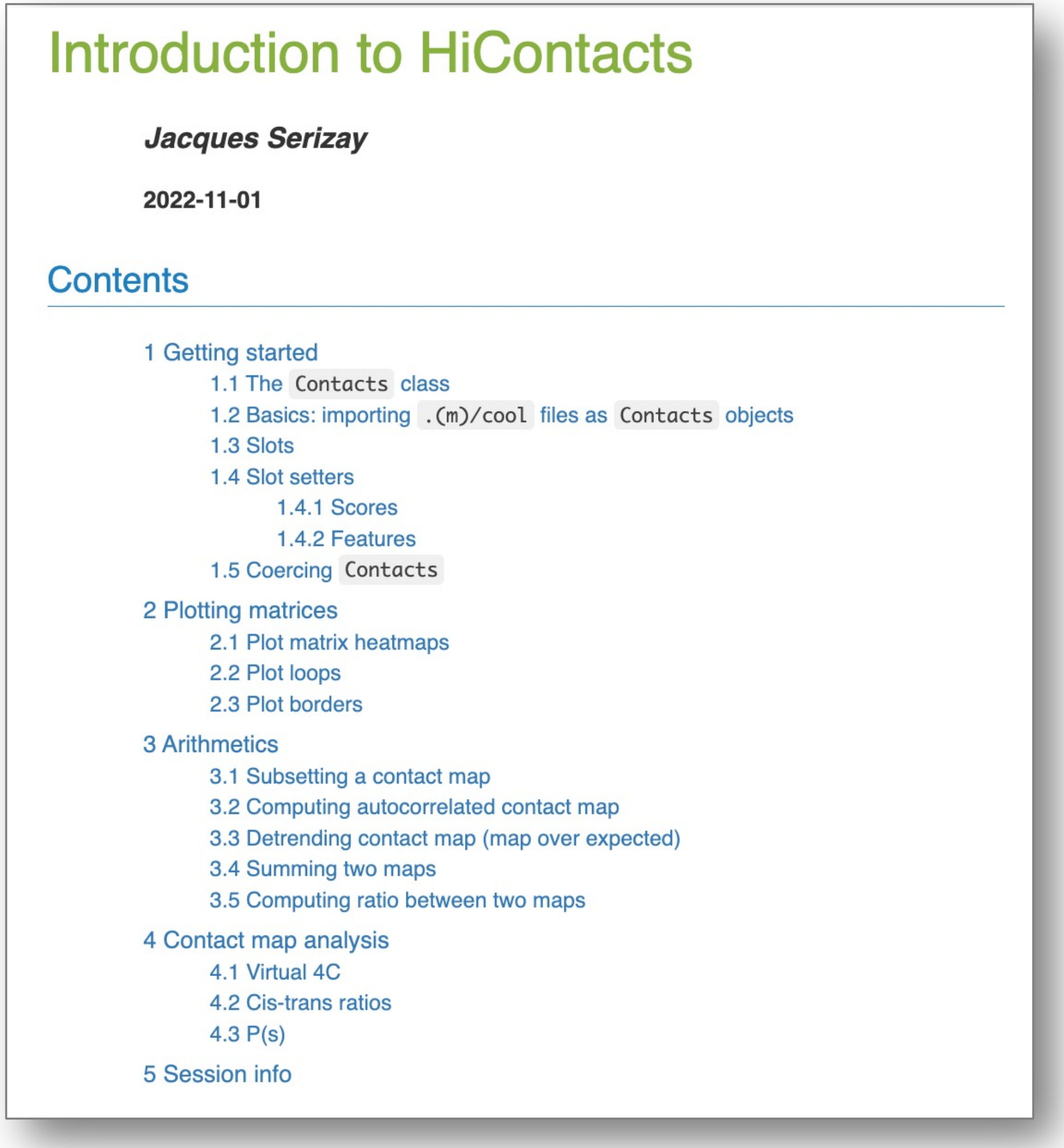
To view documentation for the version of this package installed in your system, start R and enter:

HTML	R Script	Introduction to HiContacts
PDF		Reference Manual
Text		LICENSE

Vignettes



~~~
---
title: "Introduction to HiContacts"
author: "Jacques Serizay"
date: "`r Sys.Date()`"
output: 
    BiocStyle::html_document
vignette: >
    %\VignetteIndexEntry{Introduction to HiContacts}
    %\VignetteEngine{knitr::rmarkdown}
    %\VignetteEncoding{UTF-8}
---

```{r, eval = TRUE, echo=FALSE, results="hide", warning=FALSE}
knitr::opts_chunk$set(
    collapse = TRUE,
    comment = "#>",
    crop = NULL
)
suppressPackageStartupMessages({
    library(ggplot2)
    library(dplyr)
    library(GenomicRanges)
    library(HiContactsData)
    library(HiContacts)
})
```

~~~



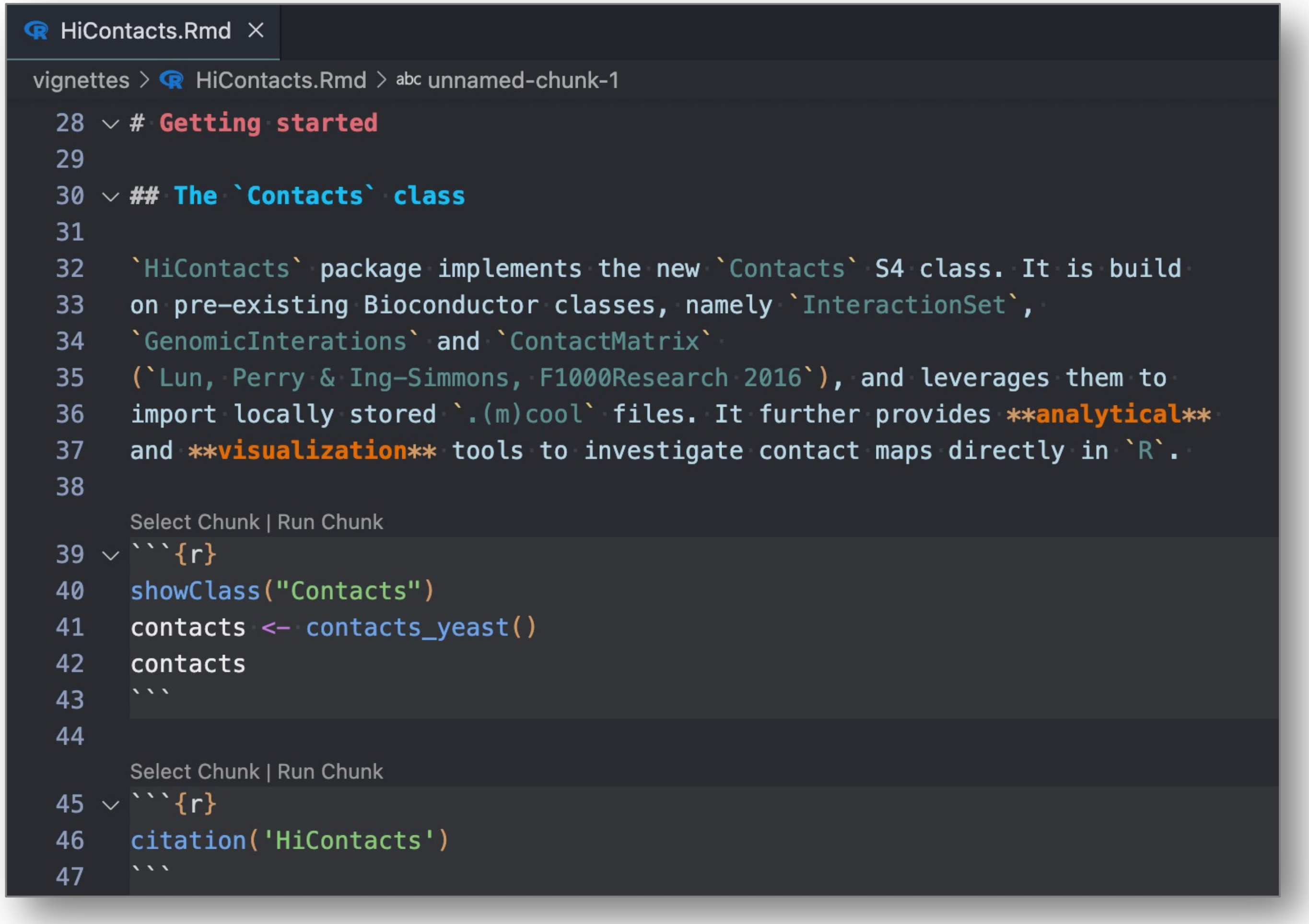
Introduction to HiContacts

Jacques Serizay

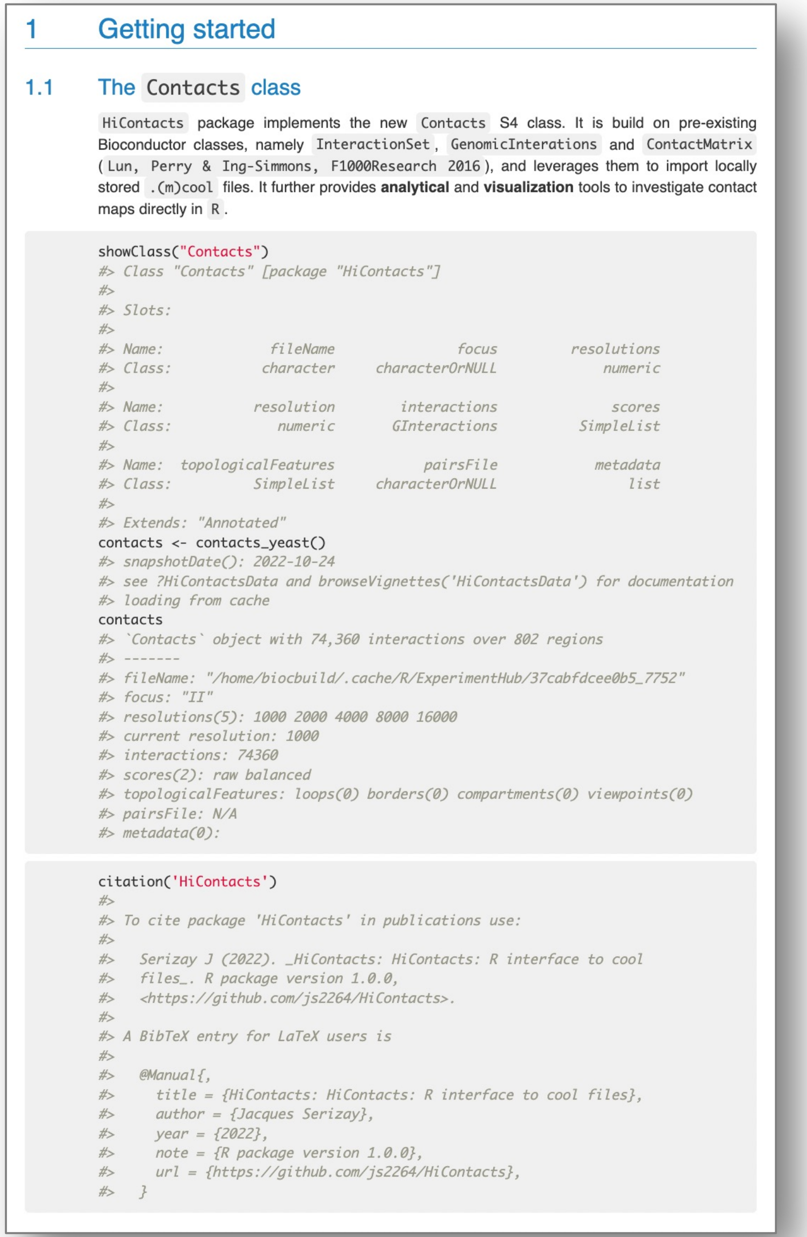
2022-11-01

Contents

Vignettes



~~~
# Getting started

## The `Contacts` class

`HiContacts` package implements the new `Contacts` S4 class. It is build
on pre-existing Bioconductor classes, namely `InteractionSet`,
`GenomicInterations` and `ContactMatrix`
(`Lun, Perry & Ing-Simmons, F1000Research 2016`), and leverages them to
import locally stored `.(m)cool` files. It further provides **analytical**
and **visualization** tools to investigate contact maps directly in `R`.

```{r}
showClass("Contacts")
contacts <- contacts_yeast()
contacts
~~~

```{r}
citation('HiContacts')
```
```



1 Getting started

1.1 The Contacts class

HiContacts package implements the new Contacts S4 class. It is build on pre-existing Bioconductor classes, namely InteractionSet, GenomicInterations and ContactMatrix (Lun, Perry & Ing-Simmons, F1000Research 2016), and leverages them to import locally stored .(m)cool files. It further provides **analytical** and **visualization** tools to investigate contact maps directly in R.

```
showClass("Contacts")
#> Class "Contacts" [package "HiContacts"]
#> 
#> Slots:
#> 
#> Name:             fileName              focus        resolutions
#> Class:           character    characterOrNULL            numeric
#> 
#> Name:           resolution       interactions             scores
#> Class:             numeric      GInteractions         SimpleList
#> 
#> Name:  topologicalFeatures          pairsFile           metadata
#> Class:          SimpleList    characterOrNULL               list
#> 
#> Extends: "Annotated"
contacts <- contacts_yeast()
#> snapshotDate(): 2022-10-24
#> see ?HiContactsData and browseVignettes('HiContactsData') for documentation
#> loading from cache
contacts
#> `Contacts` object with 74,360 interactions over 802 regions
#> -------
#> fileName: "/home/biocbuild/.cache/R/ExperimentHub/37cabfdcee0b5_7752"
#> focus: "II"
#> resolutions(5): 1000 2000 4000 8000 16000
#> current resolution: 1000
#> interactions: 74360
#> scores(2): raw balanced
#> topologicalFeatures: loops(0) borders(0) compartments(0) viewpoints(0)
#> pairsFile: N/A
#> metadata(0):
```

```
citation('HiContacts')
#> 
#> To cite package 'HiContacts' in publications use:
#> 
#>   Serizay J (2022). _HiContacts: HiContacts: R interface to cool
#>   files_. R package version 1.0.0,
#>   <https://github.com/js2264/HiContacts>.
#> 
#> A BibTeX entry for LaTeX users is
#> 
#>   @Manual{,
#>     title = {HiContacts: HiContacts: R interface to cool files},
#>     author = {Jacques Serizay},
#>     year = {2022},
#>     note = {R package version 1.0.0},
#>     url = {https://github.com/js2264/HiContacts},
#>   }
```
```

# Vignettes

This means **all the code** in a vignette **must** work!

# Standard package content

```
myPackage/
    R/
        functions.R
        utils.R
    man/
        myfunction.Rd
    tests/
        testthat.R
        testthat/
            test-myfun.R
    DESCRIPTION
    README.md
    NAMESPACE
    NEWS
    LICENSE
```

➡

```
myPackage/
    R/
        functions.R
        utils.R
    man/
        myfunction.Rd
    tests/
        testthat.R
        testthat/
            test-myfun.R
    inst/
        extdata/
            <raw-data-file>
    data/
        <data>.Rda
    vignettes/
        myPackage.Rmd
    DESCRIPTION
    NAMESPACE
    README.md
    NEWS
    LICENSE
```