Introduction to R environment – 3

Marcin Kierczak

07 October 2016

Working with R

There are several ways to work with/in R:

- from a command line,
- in batch mode,
- from a native GUI,
- using external editor, e.g. RStudio

During this course, we will be focusing on working with RStudio and also in batch mode.

Working from comand line

1. Open Terminal.
2. Type *R*.
3. Type R commands. . .

- Arrows let you browse commands history.
- TAB attempts to autocomplete the command you have just started typing.

The batch mode

If you are working on a computational cluster, such as th e Uppmax, it is very likely you would like to run large jobs that one has to enqueue. This makes interactive work from the console virtually impossible. The solution is to run R code from a file, using the so-called **batch mode**:

1. Create a file with your code and give it extension **.R**.
2. In the console (or in the queue script) write: *R –vanilla* < *mycode.R* [two minus signs in front of *vanilla*].

Should you like to log the output add either:

- *R –vanilla* < *mycode.R* > **output.log** or
- *R –vanilla* < *mycode.R* **tee output.log**

Getting help

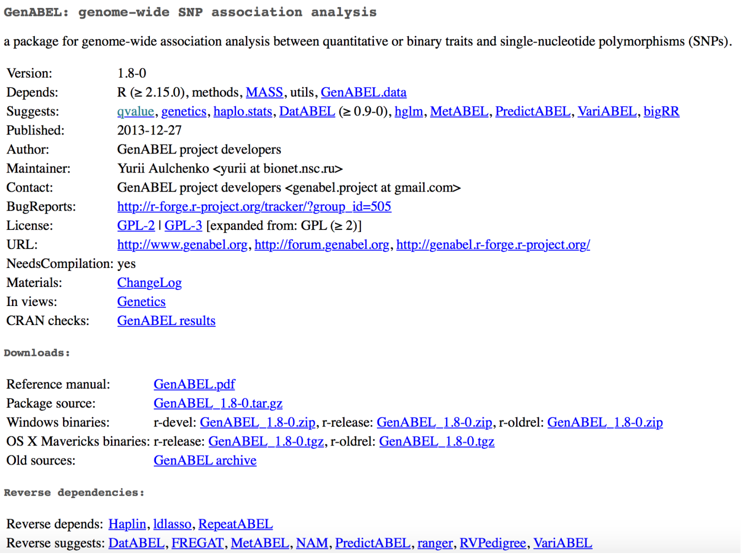
```r
help(t.test) # function level
?t.test # same as above
??t.test # extensive search
vignette("GenABEL") # package level demo(graphics)
example(barplot) # run help examples for barplot
demo() # see all currently available demos
demo('graphics') # run demo for 'graphics'
```

Working with packages

Packages are organised in repositories. The three main repositories are:

- CRAN
- R-Forge
- Bioconductor

Working with packages – CRAN example.



GenABEL: genome-wide SNP association analysis

a package for genome-wide association analysis between quantitative or binary traits and single-nucleotide polymorphisms (SNPs).

Version:	1.8-0
Depends:	R (≥ 2.15.0), methods, MASS, utils, GenABEL.data
Suggests:	qvalue, genetics, haplo.stats, DatABEL (≥ 0.9-0), hglm, MetABEL, PredictABEL, VariABEL, bigRR
Published:	2013-12-27
Author:	GenABEL project developers
Maintainer:	Yurii Aulchenko <yurii at bionet.nsc.ru>
Contact:	GenABEL project developers <genabel.project at gmail.com>
BugReports:	http://r-forge.r-project.org/tracker/?group_id=505
License:	GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]
URL:	http://www.genabel.org, http://forum.genabel.org, http://genabel.r-forge.r-project.org/
NeedsCompilation:	yes
Materials:	ChangeLog
In views:	Genetics
CRAN checks:	GenABEL results

Downloads:

Reference manual:	GenABEL.pdf
Package source:	GenABEL_1.8-0.tar.gz
Windows binaries:	r-devel: GenABEL_1.8-0.zip, r-release: GenABEL_1.8-0.zip, r-oldrel: GenABEL_1.8-0.zip
OS X Mavericks binaries:	r-release: GenABEL_1.8-0.tgz, r-oldrel: GenABEL_1.8-0.tgz
Old sources:	GenABEL archive

Reverse dependencies:

Reverse depends:	Haplin, ldlasso, RepeatABEL
Reverse suggests:	DatABEL, FREGAT, MetABEL, NAM, PredictABEL, ranger, RVPedigree, VariABEL

Figure 1:

Working with packages – installation

Only a few packages are pre-installed:

```
library("XLConnect")
```

```
## Error in library("XLConnect"): there is no package calle
```

In order to install a package from command line, use:

```
install.packages('GenABEL', dependencies=T)
```

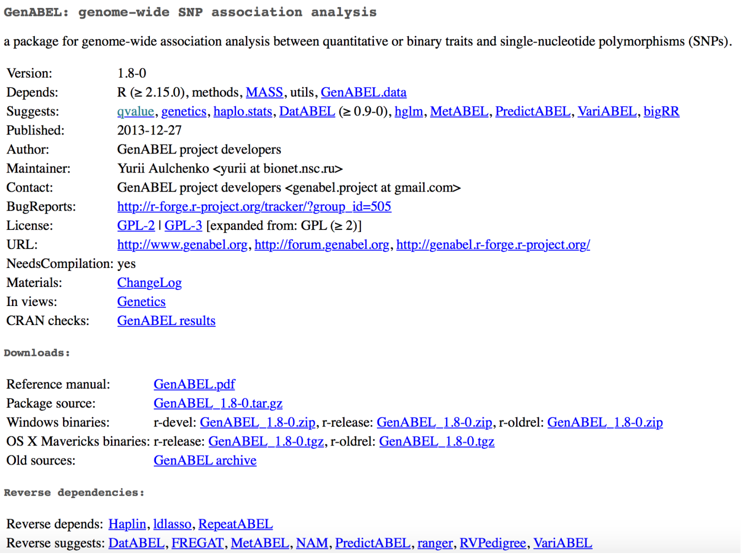
Working with packages – details

It may happen that you want to also specify the repository, e.g. because it is geographically closer to you or because your default mirror is down:

```
install.packages('GenABEL', dependencies=T,
                 repos="http://cran.se.r-project.org")
```

But, sometimes, this does not work either because the package is not available for your platform. In such case, you need to compile it from source.

Working with packages – details cted.



GenABEL: genome-wide SNP association analysis

a package for genome-wide association analysis between quantitative or binary traits and single-nucleotide polymorphisms (SNPs).

Version:	1.8-0
Depends:	R (≥ 2.15.0), methods, MASS, utils, GenABEL.data
Suggests:	qvalue, genetics, haplo.stats, DatABEL (≥ 0.9-0), hglm, MetABEL, PredictABEL, VariABEL, bigRR
Published:	2013-12-27
Author:	GenABEL project developers
Maintainer:	Yurii Aulchenko <yurii at bionet.nsc.ru>
Contact:	GenABEL project developers <genabel.project at gmail.com>
BugReports:	http://r-forge.r-project.org/tracker/?group_id=505
License:	GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]
URL:	http://www.genabel.org, http://forum.genabel.org, http://genabel.r-forge.r-project.org/
NeedsCompilation:	yes
Materials:	ChangeLog
In views:	Genetics
CRAN checks:	GenABEL results

Downloads:

Reference manual:	GenABEL.pdf
Package source:	GenABEL_1.8-0.tar.gz
Windows binaries:	r-devel: GenABEL_1.8-0.zip, r-release: GenABEL_1.8-0.zip, r-oldrel: GenABEL_1.8-0.zip
OS X Mavericks binaries:	r-release: GenABEL_1.8-0.tgz, r-oldrel: GenABEL_1.8-0.tgz
Old sources:	GenABEL archive

Reverse dependencies:

Reverse depends:	Haplin, ldlasso, RepeatABEL
Reverse suggests:	DatABEL, FREGAT, MetABEL, NAM, PredictABEL, ranger, RVPedigree, VariABEL

Figure 2:

Working with packages – installing from source.

0. Make sure you have appropriate tools installed, e.g. XCode or build-essentials.
1. Download the source file, in our example *GenABEL_1.8-0.tar.gz*.
2. Install it:

```
install.packages("path/to/GenABEL_1.8-0.tar.gz",
                 repos=NULL,
                 type='source',
                 dependencies=T)
```

3. Load it:

```
library('GenABEL') # always forces reloading
require('GenABEL') # load only if not already loaded
```

4. Enjoy!

Packages – GitHub

Nowadays, mora and more developers contribute their packages via GitHub. The easiest way to install packages from the GitHub is via the *devtools* package:

1. Install the *devtools* package.
2. Load it.
3. Install.
4. Enjoy!

```
install.packages('devtools', dependencies=T)
library('devtools')
install_github('talgalili/installr')
```

Packages – Bioconductor

Figure 3:

First install Bioconductor:

```
source("https://bioconductor.org/biocLite.R")
# If the above does not work, use http instead of https
biocLite()
```

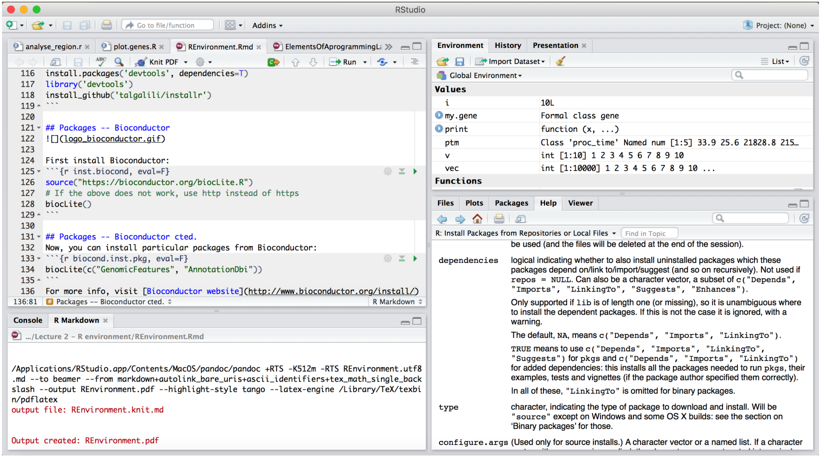
Packages – Bioconductor cted.

Now, you can install particular packages from Bioconductor:

```
biocLite(c("GenomicFeatures", "AnnotationDbi"))
```

For more info, visit Bioconductor website

RStudio – a live demonstration



~~~
install.packages('devtools', dependencies=T)
library('devtools')
install_github('talgalili/installr')
```

## Packages -- Bioconductor
![](logo_bioconductor.gif)

First install Bioconductor:
```{r inst.biocond, eval=F}
source("https://bioconductor.org/biocLite.R")
# If the above does not work, use http instead of https
biocLite()
```

## Packages -- Bioconductor cted.
Now, you can install particular packages from Bioconductor:
```{r biocond.inst.pkg, eval=F}
biocLite(c("GenomicFeatures", "AnnotationDbi"))
```

For more info, visit [Bioconductor website](http://www.bioconductor.org/install/)
~~~

i	10L
my.gene	Formal class gene
print	function (x, ...)
ptm	Class 'proc_time' Named num [1:5] 33.9 25.6 21828.8 215…
v	int [1:10] 1 2 3 4 5 6 7 8 9 10
vec	int [1:10000] 1 2 3 4 5 6 7 8 9 10 ...

```
/Applications/RStudio.app/Contents/MacOS/pandoc/pandoc +RTS -K512m -RTS REnvironment.utf8
.md --to beamer --from markdown+autolink_bare_uris+ascii_identifiers+tex_math_single_back
slash --output REnvironment.pdf --highlight-style tango --latex-engine /Library/TeX/texbi
n/pdflatex
output file: REnvironment.knit.md


Output created: REnvironment.pdf
```

be used (and the files will be deleted at the end of the session).

`dependencies` logical indicating whether to also install uninstalled packages which these packages depend on/link to/import/suggest (and so on recursively). Not used if `repos = NULL`. Can also be a character vector, a subset of `c("Depends", "Imports", "LinkingTo", "Suggests", "Enhances")`.

Only supported if `lib` is of length one (or missing), so it is unambiguous where to install the dependent packages. If this is not the case it is ignored, with a warning.

The default, `NA`, means `c("Depends", "Imports", "LinkingTo")`.

`TRUE` means to use `c("Depends", "Imports", "LinkingTo", "Suggests")` for `pkgs` and `c("Depends", "Imports", "LinkingTo")` for added dependencies: this installs all the packages needed to run `pkgs`, their examples, tests and vignettes (if the package author specified them correctly).

In all of these, `"LinkingTo"` is omitted for binary packages.

`type` character, indicating the type of package to download and install. Will be `"source"` except on Windows and some OS X builds: see the section on 'Binary packages' for those.

`configure.args` (Used only for source installs.) A character vector or a named list. If a character

Figure 4: RStudio screenshot