



## Working in the R environment

R Foundations for Life Scientists

Marcin Kierczak

# 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 command line

1. Open Terminal.
  2. Type `R`.
  3. Type R commands...
  4. Type `q()` to quit R.
- Arrows let you browse throughout the 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 the 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 like this
- `R --vanilla < mycode.R | tee output.log`

# Getting help

```
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'
```

Stackoverflow is a great resource.

# Working with packages

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

- CRAN
- R-Forge
- Bioconductor

We also have [GitHub](#).

## 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

```

# Working with packages -- installation

Only a few packages are pre-installed:

```
library(XLConnect)
```

```
## Error in library(XLConnect): there is no package called 'XLConnect'
```

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

```
install.packages("GenABEL",dependencies=TRUE)
```

## 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=TRUE,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 its *source code*.



# 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 ( $\geq 2.15.0$ ), methods, [MASS](#), [utils](#), [GenABEL.data](#)  
Suggests: [qvalue](#), [genetics](#), [haplo.stats](#), [DatABEL](#) ( $\geq 0.9-0$ ), [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](http://r-forge.r-project.org/tracker/?group_id=505)  
License: [GPL-2](#) | [GPL-3](#) [expanded from: GPL ( $\geq 2$ )]  
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](#)

# Working with packages -- installing from source.

- Make sure you have appropriate tools installed, e.g. XCode or build-essentials.
- Download the source file, in our example *GenABEL\_1.8-0.tar.gz*.
- Install it:

```
install.packages("path/to/GenABEL_1.8-0.tar.gz",  
                 repos=NULL,  
                 type='source',  
                 dependencies=TRUE)
```

- Load it:

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

- Enjoy!

# Packages -- GitHub

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

- Install the *devtools* package.
- Load it.
- Install.
- Enjoy!

```
install.packages('devtools',dependencies=TRUE)  
library('devtools')  
install_github('talgalili/installr')
```

# Packages -- Bioconductor



First install Bioconductor Manager:

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

## Packages -- Bioconductor cted.

Now, you can install particular packages from Bioconductor:

```
BiocManager::install("GenomicRanges")
```

For more info, visit [Bioconductor website](#).

# One package to rule them all -- the magic of `renv`

- first time do `renv::activate()` and `renv::init()`
- while working: `renv::hydrate()` and `renv::snapshot()`

Now, send `renv.lock` to your friend to share the environment and she can:

- restore the environment `renv::restore()`

**Pure magic!**

See you at the next lecture!

Graphics from  freepik.com

Created: 27-Sep-2021 • Roy Francis • SciLifeLab • NBIS

