Package: dartRverse (via r-universe)

November 23, 2026

Title Install and Load the 'dartRverse' Suits of Packages

Version 1.3.0

Date 2026-11-18

Revision Hardy Hedgehog

Description Provides a single function that supports the installation of all packages belonging to the 'dartRverse'. The 'dartRverse' is a set of packages that work together to analyse SNP (single nuclear polymorphism) data. All packages aim to have a similar 'look and feel' and are based on the same type of data structure ('genlight'), with additional metadata for loci and individuals (samples). For more information visit the 'GitHub' pages <https://github.com/green-striped-gecko/dartRverse>.

URL https://github.com/green-striped-gecko/dartRverse

License GPL (>= 3)

Depends R (>= 3.5)

Imports cli (>= 3.4.0), rlang, remotes, httr, BiocManager

Suggests dartR.base, dartR.data, dartR.sim, dartR.captive, dartR.popgen, dartR.spatial, dartR.sexlinked, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Maintainer Bernd Gruber <bernd.gruber@canberra.edu.au>

Author Bernd Gruber [aut, cre], Arthur Georges [aut], Jose L. Mijangos [aut], Carlo Pacioni [aut], Peter J. Unmack [aut], Emily Stringer [aut], Oliver Berry [aut]

Repository https://cran.r-universe.dev

Date/Publication 2026-11-23 03:20:02 UTC

RemoteUrl https://github.com/cran/dartRverse

RemoteRef HEAD

RemoteSha b6978206eb360bae42b0f865699414d78024c5a5

Contents

dartRverse_install 2
gl.download.binary 4

Index **6**

`dartRverse_install` *Checks installed package and Supports installation of CRAN and Github packages of the dartRverse*

Description

This function installs or updates one or several dartR packages from CRAN or from a Github branch (main, beta or dev), including packages that are loaded at the moment. If run with no parameter the installed packages and the versions available on CRAN and Github are printed.

Usage

```
dartRverse_install(
  package = NULL,
  rep = "CRAN",
  branch = "main",
  verbose = TRUE
)
```

Arguments

`package`	Name(s) of the package(s) to install: dartR.base, dartR.data, dartR.sim, dartR.popgen, dartR.spatial, dartR.captive or dartR.sexlinked. Use "all" to install all of them, including those not installed yet, which can take a long time on a new computer. If NULL, the installed packages and the versions available on CRAN and Github are printed [default NULL].
`rep`	Which repository is used: 'CRAN' or 'Github' (partial matching, any case) [default 'CRAN'].
`branch`	The Github branch to install from, usually main, beta or dev. It needs rep = 'Github'; supplying it with rep = 'CRAN' is an error [default 'main'].
`verbose`	If TRUE, the version table is printed when package is NULL; it has no effect otherwise [default TRUE].

Details

Updating without a restart. Loaded dartR packages that are affected are unloaded, the new versions are installed, and the packages are loaded again in their original place on the search path. Objects in your workspace are kept. R has to be restarted in two cases, and the function says so: when a package outside the dartRverse imports the dartR package, and on Windows when the update needs a newer version of a loaded third-party package with compiled code. To update dartRverse itself, use install.packages("dartRverse") and restart R.

Bioconductor packages. Most dartR packages depend on Bioconductor packages, such as SNPRelate, snpStats and LEA, which are not on CRAN. Those that are missing or fail to load are installed first, using the Bioconductor release that BiocManager chooses for your version of R, or the release in the environment variable R_BIOC_VERSION for Github installs if you have set it. If that fails, the function stops before any dartR package is changed. On Linux and macOS these packages can need system libraries, which R cannot install for you.

Failed installs. If an install fails, the function still tries the other requested packages, restores your session, and then stops with an error naming the packages that failed.

Version table. Reading the versions from CRAN and Github can take a moment, and the function says so before it starts. Where the console supports it, such as in RStudio, version numbers link to the package on CRAN or Github, and an install link runs the install. A tip above the table says so, followed by the line with links that install all packages at once. Elsewhere the table is plain text, and the line above it shows the command to type. A package that is installed but fails to load is marked. The function does not attach any package to show the table.

Branches. The 'main' branch on Github is identical with the latest CRAN submission. Important changes and fixes are published under 'beta' and tested there, before they are submitted to CRAN. Hence this might be the best chance to look for fixes. All 'dev' branches are 'risky' meaning they have not been tested.

A Github install changes only the named packages. Packages they depend on, such as dartR.base, are not moved to the same branch, so use "all" if you need matching versions. Dependencies that are out of date are not upgraded.

Value

Invisibly: 1 when the version table is requested, NULL otherwise.

Author(s)

Author(s): Bernd Gruber. Custodian: Bernd Gruber – Post to https://groups.google.com/d/forum/dartr

Examples

```
# returns without printing the table or installing anything
dartRverse_install(verbose = FALSE)
## Not run:
# installed packages and available versions (needs internet)
dartRverse_install()
# install or update all packages from CRAN
dartRverse_install("all")
# install or update all packages from the dev branch on Github
```

```
dartRverse_install("all", rep = "Github", branch = "dev")
# install or update one package from the beta branch on Github
dartRverse_install("dartR.captive", rep = "Github", branch = "beta")

## End(Not run)
```

`gl.download.binary` *Support function to download binaries from github*

Description

This function supports the download of binaries from github. Those binaries are compiled files that allow to run dartR functions that integrate third party software such as epos (gl.run.epos), NeEstimator (gl.LDNe) or Structure (gl.run.structure). Please be aware this is just to allow for easy install and use of dartR functions. We have explicitly asked the authors of the software for permission to include those binaries here. Understandably some authors prefer to direct users to their own websites to download the software. Here a comment is issued by the function how to do so. Please note: **The third party packages are the work of others and please make the effort to cite them accordingly**. You find the citations under the help pages of the respective functions, for example ?gl.run.epos.

Usage

```
gl.download.binary(
  software = NULL,
  os = NULL,
  branch = "main",
  out.dir = tempdir(),
  verbose = 2
)
```

Arguments

`software`	Name of the software package to download, one of "beagle", "coancestry", "colony", "eems", "emibd9", "epos", "faststructure", "gone", "ms", "neestimator", "plink", "popcluster", "stairway2" and "structure". Upper case is converted to lower case. Depending on the software several files will be downloaded. If NULL, a table of the binaries available for each operating system is printed. Where the console supports it, such as in RStudio, clicking a tick in the table downloads that binary; a tip above the table says so [default NULL].
`os`	The operating system to download the binary for: "windows", "mac" or "linux" (partial matching is allowed). Some binaries are not available for all operating systems. If NULL, the current operating system is used [default NULL].
`branch`	Which branch of the dartRverse github repository to download from, or to list the binaries of. There should be no reason to change it [default "main"].
`out.dir`	The path where to save the binary [default tempdir()].
`verbose`	If 0, download progress and messages are suppressed; any other value prints them [default 2].

Details

After a download the function says which software was downloaded, which dartR functions use it (with links to their help pages where the console supports them) and asks you to check the help pages of those functions to cite the program and its methods correctly.

On mac and linux the unzipped files are made executable (mode 777), so the dartR functions that call them, such as gl.run.epos or gl.LDNe, can run them straight away.

Value

When software is given, invisibly, the path of the folder that holds the unzipped binary (out.dir/software). When software is NULL, NULL (the table of available binaries is printed).

Examples

```
## Not run:
# list the binaries available for each operating system
gl.download.binary()
# download PLINK for the current operating system into tempdir()
path <- gl.download.binary("plink")

## End(Not run)
```

Index

∗ **dartRverse installation**
dartRverse_install, 2

dartRverse_install, 2

gl.download.binary, 4