Package ‘Revticulate’

January 20, 2025

Type Package

Title Interaction with ``RevBayes'' in R

Version 1.0.0

Description
Interaction with ``RevBayes'' via R. Objects created in ``RevBayes'' can be passed into the R environment, and many types can be converted into similar R objects. To download ``RevBayes'', go to <https://revbayes.github.io/download>.

License MIT + file LICENSE

Depends R (>= 2.10)

Imports ape, stringr, knitr, utils, comprehenr, jsonlite, usethis, rstudioapi

RoxygenNote 7.1.2

Suggests markdown, rmarkdown, testthat, ggplot2, devtools, RevGadgets,

Encoding UTF-8

NeedsCompilation no

Author Caleb Charpentier [aut, cre],
April Wright [aut]

Maintainer Caleb Charpentier <caleb.charpentier@selu.edu>

Repository CRAN

Date/Publication 2022-03-15 23:30:07 UTC

Contents

`callRev` *Basic Call to RevBayes*

Description

Submits input to the RevBayes executable and returns output to R in string format. If coerce = TRUE, the function coerceRev() will attempt to coerce output to a similar R object.

Usage

```
callRev(
  ...,
  coerce = FALSE,
  path = Sys.getenv("rb"),
  viewCode = FALSE,
  use_wd = TRUE,
  knit = FALSE,
  timeout = 5
)
```

Arguments

`...`	character - input to send to RevBayes.
`coerce`	logical - If TRUE, attempts to coerce output to an R object. If FALSE, output will remain in String format. Default is FALSE.
`path`	character - Path to the RevBayes executable. Default is Sys.getEnv("RevBayesPath"), which is created with initRev().
`viewCode`	logical - If TRUE, the input and output in the temporary file used to interact with RevBayes will be displayed in the viewing pane. This option may be useful for diagnosing code errors.
`use_wd`	logical - If TRUE, sets the working directory in the temporary RevBayes session to the working directory of the active R session. Default is TRUE.
`knit`	logical - Argument used to manage output formatting for knitRev(). This argument should generally be ignored by the user.
`timeout`	integer - Determines how long the system2() call should wait before timing out (seconds). Default is 5.

Details

##NOTE - This function provides very limited interaction with RevBayes, and is mostly intended to provide basic functionalities for other methods. For more effective RevBayes calls, use doRev().

Value

out: character - String formatted output from RevBayes

coercedOut: type varies. R object formatted output from RevBayes. Object type varies according to Rev output (Ex: numeric vector or ape::Phylo object)

Examples

```
## Not run:
callRev("2^3")
callRev("2^3", coerce = FALSE, viewcode = TRUE)

## End(Not run)
```

`callRevFromTerminal` *Execute a .rev file in RevBayes Through an RStudio Terminal*

Description

Accesses an RStudio terminal through the RStudio API and runs a .rev script.

Usage

```
callRevFromTerminal(revscript)
```

Arguments

`revscript`	character - .rev file to execute in RevBayes

Details

Many common RevBayes use-cases, such as generating mcmcs, take a long time to execute. Because of this fact, it is inefficient and inadvisable to run them in knitr documents or with RevBayes calls through repRev(). Instead, tutorial code or code that has been written interactively can be externally saved with saveRev(), and then executed in an RStudio terminal via callRevFromTerminal(). Output files can then be explored and visualized with the RevGadgets package.

Value

termID Unique identifier of the terminal used to call RevBayes

Examples

```
## Not run:
saveRev("archertutorial.rev", use_quit=TRUE)
callRevFromTerminal("archertutorial.rev")

## End(Not run)
```

clearRev *Clear Code from Revticulate History*

Description

Removes code from .Revhistory file used for managing Revticulate history. Clears all code by default, or last ’n’ lines of code specifed by the user.

Usage

```
clearRev(n = NULL, silent = FALSE)
```

Arguments

`n`	integer - How many lines to remove. If n = NULL, all lines are removed.
`silent`	logical - If TRUE, output messages will be silenced. Default is FALSE.

Value

pseudoError: NULL. Message warning user that they attempted to erase more items from the Rev history than exist. message() is used instead of stop() so that clearRev() functions in repRev().

undoRev(n): NULL. Removes n number of lines from .Revhistory and cats the remaining history to the screen.

Examples

```
## Not run:
clearRev() #Clear all Revticulate history
clearRev(n = 3) #Remove the last 3 lines of Revticulate history

## End(Not run)
```

coerceRev *Convert RevBayes Output into R Objects*

Description

Coerces a RevBayes output string into an equivalent R object. Automatically determines R object type based on the structure of the string, and returns the original string if R object equivalent cannot be determined.

Usage

```
coerceRev(revString)
```

Arguments

`revString`	character - Output from RevBayes

Value

rObject: Type varies depending on Rev object type. R object-formatted output coerced from a RevBayes output string.

Examples

```
## Not run:
coerceRev("[1, 2, 3, 4]")

## End(Not run)
```

`doRev` *Call RevBayes with Continuous Session History*

Description

The basic Revticulate function for interacting with RevBayes

Usage

```
doRev(input, coerce = TRUE, evaluate = TRUE, viewCode = FALSE, timeout = 5)
```

Arguments

`input`	character - Rev language expression to execute in RevBayes
`coerce`	logical - If TRUE, the output from RevBayes will be coerced into R format with coerceRev()
`evaluate`	logical - If FALSE, input will be written to the .Revhistory file, but will not be interpreted in RevBayes. Default is FALSE.
`viewCode`	logical - If TRUE, Rev code input and output will be displayed in the viewing pane. Mostly used for development purposes.
`timeout`	integer - Determines how long the system2() call should wait before timing out. Default is 5 seconds. For longer calls, see 'callRevFromTerminal()'

Details

This function allows users to make multiple calls to RevBayes, while maintaining a persistent input history. This means that variables defined in one call can be referenced in another call, giving the feel of a continuous RevBayes session.

Because this function is the basis for repRev() and knitRev(), variables defined in RevBayes with any of these functions can be referenced by any of the others.

Value

now - RevBayes output, type varies. If coerce = TRUE, coerceRev() will attempt to convert RevBayes output into an equivalent R object. If else, the return type is character.

Examples

```
## Not run:
doRev("simTree(16)", coerce=FALSE)
doRev("a <- 10")
doRev("a * 100")

## End(Not run)
```

getRevHistory *Get Revticulate History*

Description

Return the history of RevBayes code submitted with doRev() and similar functions

Usage

```
getRevHistory()
```

Value

lines - character. A vector of the line-by-line Revticulate history

Examples

```
## Not run:
getRevHistory()

cat(getRevHistory(), sep="\n")

## End(Not run)
```

getRevVars *View RevBayes Variables*

Description

Wrapper for RevBayes ls() function. Displays variables that have been defined in the working RevBayes session.

Usage

```
getRevVars()
```

Value

No return - wrapper for RevBayes ls() function to show variables in rb environment

Examples

```
## Not run:

getRevVars()

## End(Not run)
```

knitRev *Knitr Engine for RevBayes*

Description

Creates a knitr engine for evaluating RevBayes code

Usage

```
knitRev()
```

Details

To use RevBayes in a knitr document, type 'library(Revticulate)' followed by 'knitRev()' on the next line in the initialization block. The language header for RevBayes chunks is 'rb'.

The RevBayes knitr engine is built around doRev(), allowing history to persist between chunks. If a variable 'x <- 10' is defined in one RevBayes chunk, 'x' can then be accessed in a later RevBayes chunk. Additionally, it could be accessed via doRev() in an R chunk, allowing for quick R analysis of RevBayes variables.

On top of the standard knitr chunk options, the rb engine provides two extras: rb_eval, and coerce.

If rb_eval = FALSE, the code in the chunk will be saved to the .Revhistory file, but will not be submitted to RevBayes. This option is useful for code chunks containing the final loop of an mcmc, which could be saved to an external file with saveRev() and ran in a terminal with callRevFromTerminal().

If coerce = TRUE, coerceRev() will attempt to convert RevBayes output into equivalent R formatted objects. The default value for both coerce and rb_eval is TRUE.

Value

No return. Initiates knitr engine for RevBayes.

Examples

~~~
## Not run:
```{r setup, include=FALSE}
     knitr::opts_chunk$set(echo = TRUE)
     library(Revticulate)
     knitRev()
```

```{rb rb_eval=TRUE, coerce=FALSE}

x <- simTree(32)

```

```{r}

x <- doRev('x')
plot(x)

```

## End(Not run)
~~~

`loadRevHistory` *Load Revticulate History from a .Rev File*

Description

Reads in the code from a .Rev file and writes it into the current Revticulate history.

Usage

```
loadRevHistory(filepath, overwrite = FALSE)
```

Arguments

filepath
: character - File to load .Revhistory from

overwrite
: logical - If TRUE, the code in the file at ’filepath’ will overwrite current .Revhistory. If FALSE, it will be appended to the end of it.

Value

No return - reads in new code to the .Revhistory file

Examples

```
## Not run:
     loadRevHistory("someCode.rev")

     loadRevHistory("someMoreCode.rev", overwrite=TRUE)

## End(Not run)
```

---

repRev *Interactive Session with RevBayes*

Description

Simulates a continuous, interactive session with RevBayes. While this session is active, all code will be interpreted as Rev code, and attempting to run R code may result in error.

Usage

```
repRev(path = Sys.getenv("rb"), viewCode = FALSE, coerce = TRUE, use_wd = TRUE)
```

Arguments

path
: character - Path to the RevBayes executable. Defaults to Sys.getenv("rb"), which should be assigned upon first loading the package.

viewCode
: logical - If TRUE, code from the temporary file used to interact with RevBayes will be displayed in the viewing pane. Default is FALSE. The option is mostly for developer convenience, and can be ignored by most users.

coerce
: logical - If FALSE, RevBayes output will be printed to the console in character format. If TRUE, coerceRev() will attempt to coerce output into a suitable R object. Default is TRUE.

use_wd
: logical - If TRUE, the simulated Revbayes session will use the same working directory as the active R session. If FALSE, it will use the default for the RevBayes executable. Default is TRUE.

Details

By default, the interactive session uses the present R working directory as the RevBayes working directory. This behavior can be turned off with use_wd = FALSE

The exit the session, type ’quit()’, ’q()’, or hit the ’esc’ key. clearRev(), getRevVars(), and getRevHistory() can still be called from within the session for user convenience

Value

No return. RevBayes variables assigned within the session can be accessed externally via doRev() or viewed with getRevVars().

Examples

```
## Not run:
repRev()

myNumber <- 4
myNumber

posteriorPredictiveProbability(v(2), 3)
getrRev()
clearRev()
quit()

## End(Not run)
```

saveRev *Save Revticulate History*

Description

Captures the current Revticulate history and saves it to an external file.

Usage

```
saveRev(filepath, use_wd = TRUE, use_quit = TRUE)
```

Arguments

`filepath` character - Location of the .Rev file to save Revticulate history in. If the file doesn’t exist, it will be created. Otherwise, it will be overwritten.

`use_wd` logical - If TRUE, the history will be prepended by setwd(’users current working directory’). This function will set the script’s working directory to the user’s current working directory, which is often desirable for files that will be submitted to callRevFromTerminal() and contain output moniters.

`use_quit` logical - If TRUE, q() will be appended to the end of the history before writing it to a file. This will cause the RevBayes session to quit after evaluating the script’s code, and is good practice when running an mcmc in an RStudio terminal.

Details

By providing a file path, the user can save the RevBayes code in their current Revticulate session to an external .Rev script. This script can then be executed in RevBayes via source(), or in an RStudio terminal via callRevFromTerminal(). Usage of this function can be combined with loadRev() to allow for the continual use of a Revticulate history instance between R sessions.

Examples

```
## Not run:
     saveRev("mySavedCode.rev", use_quit=FALSE)
     saveRev("mySavedCode.rev", use_wd=FALSE, use_quit=TRUE)

## End(Not run)
```

`%+%` *Operator for concatenation.*

Description

Rapid string parsing

Usage

```
a %+% b
```

Arguments

a	The first String
b	The second String

Details

Method for quicker pasting of Strings.

Value

character formatted concatenation of both input strings.

Index