Package 'tRnslate'

July 13, 2021

Type Package

Title Translate R Code in Source Files

Version 0.0.3

Date 2021-07-13

Author Mario A. Martinez Araya [aut, cre, cph]
(<https://orcid.org/0000-0002-4821-9314>)

Maintainer Mario A. Martinez Araya <r@marioma.me>

Description Evaluate inline or chunks of R code in template files and replace with their output modifying the resulting template.

License GPL (>= 2)

Depends R (>= 2.10)

Suggests knitr, rmarkdown, markdown

URL <https://marioma.me?i=soft>

BuildVignettes yes

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2021-07-13 15:40:02 UTC

R topics documented:

tRnslate-package *'Translate R Code in Source Files'*

Description

Evaluate inline or chunks of R code in template files and replace with their output modifying the resulting template.

Details

Function translate_r_code receives a character vector with the lines of a template file which contains R code inline and in R chunks. This R code is evaluated in an environment defined by the user and replaces its output in the template returning a character vector with the lines of the resulting template.

Examples

```
library(tRnslate)
# Read template containing R code inline or in chunks
T <- readLines(system.file("examples/template.txt", package = "tRnslate"))
# Create and environment to evaluate the R code in the template.
# Define objects in the environmet which are used to modify the template.
renv <- new.env(parent = parent.frame())
renv$s <- list(
intro = "#SBATCH",
partition = "hpc01",
nodes = 4,
tasks = 10,
memory = "2gb",
time = "01:00:00",
array = FALSE,
modules = 'module load openmpi/chosen/module R/chosen/module',
workdir = 'cd ${SLURM_SUBMIT_DIR}'
)
# Evaluate the R sentences in the template using the objects in the input environment.
TT <- translate_r_code(T, envir = renv)
# See the lines of the resulting template (or using 'cat' and newline as separator)
TT
```

Translate code *Translate code*

Description

Evaluate inline or chunks of R code present in general template files to produce variable content depending on some input arguments.

Usage

```
translate_r_code(x, chunk_prefix = NULL, chunk_char = "@",
        chunk_times = 1, inline_open = "<", inline_char = "@",
        inline_close = ">", char_begin = "", char_clean = "<:NULL:>",
        char_drop = "<:NULL:>",
        envir = new.env(parent = parent.frame()), comments = TRUE,
        reduce = TRUE, allow_file = FALSE, ...)
```

Arguments

x	if allow_file = FALSE (default) it is a character vector with the file lines containing R code. It can also be a unique character element where lines are assumed from the newline character \n. If allow_file = TRUE then it can also be a (unique) file path.
chunk_prefix	prefix character to identify chunks with R code. For example if chunk_prefix = "#" then chunks of R code can be placed only in comments (for an R script file, for instance). If chunk_prefix = NULL then the chunks of R code can be placed anywhere in the file (the default).
chunk_char	character to mark the chunks of R code. Default to "@".
chunk_times	number of times that chunk_char must be repeated (default to 1).
inline_open	character that opens inline R code. Default to <.
inline_char	character in between inline R code is placed. Default to @.
inline_close	character that closes inline R code. Default to >.
char_begin	character to print at the beginning of line before output when adding lines at translating R code. Default is blank, but for comments (if chunk_prefix = "#" for instance) it should start with #.
char_clean	character to print to replace r code with empty or NULL output generating empty line for assignation R code. Default to <:NULL:>.
char_drop	character to print to indicate which lines should be dropped. Can be a regular expression. Default to <:NULL:>.
envir	environment where to evaluate R code.
comments	keep comments before evaluate R code? Default to TRUE.
reduce	logical. Delete consecutive empty lines? Default to TRUE.
allow_file	let x to be a file.path and/or vector of lines read using readLines. Default to FALSE.
...	other arguments to pass to functions. At the moment only debug works to trigger debugging using browser. It can be a logical value to enable/disable debugging at all levels or a character string with the name of the function at whose level we want to trigger debugging.

Details

The input of translate_r_code is a file path, a character vector such as those obtained using readLines or just a unique character element (where each line is assumed using the newline character) with the content of a template file containing inline or chunks with R code. Users can define

an environment (including objects) where to evaluate this R code. Once the template's R code is evaluated, its output is replaced in the template. `translate_r_code` returns a character vector where each element is the corresponding line of the file so that its content can be written to disk easily using `cat`. Characters to identify inline and chunk R code can be defined by the user. Assuming the default values for the input argument of `translate_r_code`, file lines starting with `@r` can contain R code in the whole line while code in between `<r@ code @>` evaluates R code only for the portion in between the opening `<r@` and the closing `@>`. These characters to mark chunks and inline openings and closings can be modified by the user. The R code is evaluated by order of appearance (top to bottom, left to right) and the behaviour of the output depends on the pressence of assignation `<-`. Thus, to control the output, it is necessary to consider two main rules:

- Do not mix assignation (`<-`) with printing (assignation is only evaluated, not printed).
- Separate chunks using an empty line.

For more details see tRnslate package vignette or run `vignette("tRnslate")`. For an example see tRnslate-package.

Value

Once the chunks or inline R code are evaluated by `translate_r_code`, it returns a character vector where each element corresponds to the original line in the template file where the chunks and inline code has been replaced by its output. This content can be seen in console or written to disc, for example, by using `cat` (it requires to use `sep = "\n"`).

Author(s)

Mario Martinez Araya, `<r@marioma.me>`.

Examples

```
## To see an example in R console run:
##
## ?tRnslate::tRnslate
```

Index