Package 'downloadR'

August 30, 2021

Type Package

Title echoverse module: Single- and multi-threaded downloading functions

Version 0.99.0

Description echoverse module: Single- and multi-threaded downloading functions.

URL https://github.com/RajLabMSSM/downloadR

BugReports https://github.com/RajLabMSSM/downloadR/issues

Encoding UTF-8

LazyData false

Depends R (>= 3.6.0)

SystemRequirements Python (>= 3.7.0)

biocViews

Imports echoconda,
utils,
methods,
parallel

Suggests rmarkdown,
remotes,
knitr,
testthat (>= 3.0.0)

Remotes github::RajLabMSSM/echoconda

RoxygenNote 7.1.1

VignetteBuilder knitr

License GPL (>= 3) + file LICENSE

Config/testthat/edition 3

R topics documented:

`downloader` *downloader wrapper*

Description

R wrapper for `"axel"` (multi-threaded) and `"download.file"` (single-threaded) download functions.

Usage

```
downloader(
  input_url,
  output_path = file.path(tempdir(), basename(input_url)),
  download_method = c("axel", "wget", "download.file", "internal", "wininet",
    "libcurl", "wget", "curl"),
  background = FALSE,
  force_overwrite = FALSE,
  quiet = TRUE,
  show_progress = TRUE,
  continue = TRUE,
  nThread = parallel::detectCores() - 1,
  alternate = TRUE,
  check_certificates = TRUE,
  timeout = 30 * 60,
  conda_env = "echoR"
)
```

Arguments

`input_url`	URL to remote file.
`output_path`	The file name you want to save the download as.
`download_method`	• `"axel"` : Multi-threaded • `"wget"` : Single-threaded • `"download.file"` : Single-threaded • `"internal"` : Single-threaded (passed to download.file) • `"wininet"` : Single-threaded (passed to download.file) • `"libcurl"` : Single-threaded (passed to download.file) • `"curl"` : Single-threaded (passed to download.file) or `"download.file"` (single-threaded) .
`background`	Run in background
`force_overwrite`	Overwrite existing file.
`quiet`	Run quietly.
`show_progress`	show_progress.
`continue`	continue.
`nThread`	Number of threads to parallelize over.

alternate	alternate,
check_certificates	check_certificates
timeout	How many seconds before giving up on download. Passed to download.file. Default: 30*60 (30min).
conda_env	Conda environment to use.

Value

Local path to downloaded file.

See Also

Other downloaders: axel(), wget()

Examples

```
rda_url<-"https://github.com/RajLabMSSM/echolocatoR/raw/master/data/BST1.rda"
out_path <- downloadR::downloader(
    input_url = rda_url,
    download_method = "axel"
)
```

---

download_vcf *Download a remote VCF file and its index file*

Description

Download a remote VCF file and its index file

Usage

```
download_vcf(
  vcf_url,
  vcf_dir = tempdir(),
  download_method = "download.file",
  force_new = FALSE,
  quiet = TRUE,
  nThread = parallel::detectCores() - 1
)
```

Arguments

vcf_url	Remote URL to VCF file.
vcf_dir	Where to download VCF file.
download_method	• "axel" : Multi-threaded • "wget" : Single-threaded • "download.file" : Single-threaded • "internal" : Single-threaded (passed to download.file)

- `"wininet"` : Single-threaded (passed to download.file)
- `"libcurl"` : Single-threaded (passed to download.file)
- `"curl"` : Single-threaded (passed to download.file)

or `"download.file"` (single-threaded) .

`force_new`	Overwrite a previously downloaded VCF with the same path name.
`quiet`	Run quietly.
`nThread`	Number of threads to parallelize over.

Value

List containing the paths to the downloaded VCF and its index file.

Examples

```
vcf_url <- "https://gwas.mrcieu.ac.uk/files/ieu-a-298/ieu-a-298.vcf.gz"
out_paths <- download_vcf(vcf_url = vcf_url)
```

`load_rdata` load_rdata

Description

Load processed data (*.rda* format) using a function that assigns it to a specific variable (so you don't have to guess what the loaded variable name is).

Usage

```
load_rdata(fileName)
```

Arguments

`fileName`	Name of the file to load.

Index