

Package ‘sleeper’

September 17, 2026

Title Estimate Sleep Status from Accelerometry Data

Version 0.5.0

Description Wraps the classifier from the Sundararajan (2021) [<doi:10.1038/s41598-020-79217-x>](https://doi.org/10.1038/s41598-020-79217-x) to estimate sleep using a random forest. Users must download the model files from Sundararajan (2020) [<doi:10.5281/zenodo.3752645>](https://doi.org/10.5281/zenodo.3752645) in order to use this method.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.1.0)

Imports assertthat, curl, dplyr, methods, reticulate, rlang

Suggests callr, readr, testthat (>= 3.0.0)

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation no

Author John Muschelli [aut, cre] (ORCID: [<https://orcid.org/0000-0001-6469-1750>](https://orcid.org/0000-0001-6469-1750))

Maintainer John Muschelli <muschelli2@gmail.com>

Repository CRAN

Date/Publication 2026-09-17 09:10:02 UTC

Contents

py_estimate_sleep	2
py_require_sleeper	2
sl_example_model	3
sl_features	3
sl_have_models	4
sl_python_modules_installed	5
Index	6

py_estimate_sleep	<i>Run estimate_sleep with Python</i>
-------------------	---------------------------------------

Description

Run estimate_sleep with Python

Usage

```
py_estimate_sleep(
  ...,
  pyenv_function = function() {
    sleeper::py_require_sleeper()
  },
  show = FALSE
)
```

Arguments

...	arguments to pass to estimate_sleep
pyenv_function	function that loads the forest Python package. By default, it uses py_require_sleeper to import the package. If this function has an args argument, the output of pyenv_function will be re-assigned to args.
show	Logical, whether to show the standard output on the screen while the child process is running, passed to callr::r()

Value

Output from [estimate_sleep\(\)](#), a tibble of times and classification of sleep.

py_require_sleeper	<i>Command for py_require for the full sleeper workflow</i>
--------------------	---

Description

Command for py_require for the full sleeper workflow

Usage

```
py_require_sleeper(...)
```

Arguments

...	arguments to pass to reticulate::py_require()
-----	---

Value

A logical value indicating whether the package is available.

sl_example_model	<i>Compact Example Sleep Models</i>
------------------	-------------------------------------

Description

Compact Example Sleep Models

Usage

```
sl_example_model(  
  model_dir = file.path(tempdir(), "sleeper-example-models"),  
  overwrite = FALSE  
)
```

Arguments

model_dir	Directory in which to save the example models.
overwrite	Logical; overwrite files that are already present.

Value

The downloaded model paths, or NULL when a download fails.

Examples

```
models = sl_example_model()
```

sl_features	<i>Get Sleep Features</i>
-------------	---------------------------

Description

Get Sleep Features

Usage

```
sl_features(data)
```

Arguments

data	A <code>data.frame</code> with columns of timestamp, x, y, z
------	--

Value

A tibble of times and classification of sleep.

Examples

```
file = system.file("extdata", "example_data.csv.gz", package = "sleeper")
if (requireNamespace("readr", quietly = TRUE)) {
  data = readr::read_csv(file, n_max = 3600L)
  feat = sl_features(data)
}
```

sl_have_models

Check for Downloaded Sleep Models

Description

Check for Downloaded Sleep Models

Usage

```
sl_have_models(model_dir)

sl_download_models(
  model_dir,
  quiet = FALSE,
  ...,
  overwrite = FALSE,
  folds = 1:5
)
```

Arguments

model_dir	path to the folder with models from doi:10.5281/zenodo.3752645
quiet	argument passed to curl::curl_download
...	additional arguments to pass to curl::curl_download
overwrite	logical, if TRUE will overwrite existing files
folds	should all model folds be downloaded? If only some folds, specify that (used almost exclusively for testing).

Value

A logical indicating all folds are downloaded.

sl_python_modules_installed

Estimate Sleep from Wrist-Worn Accelerometry

Description

Estimate Sleep from Wrist-Worn Accelerometry

Usage

```
sl_python_modules_installed(component = c("models", "features"))
```

```
estimate_sleep(data, epoch = 30L, model_dir, cores = 1L)
```

Arguments

component	Python dependency set to check. The default, "models", checks all modules needed for sleep classification; "features" checks only the feature-extraction modules.
data	A data.frame with columns of timestamp, x, y, z
epoch	Time in seconds for the time interval estimate
model_dir	path to the folder with models from doi:10.5281/zenodo.3752645
cores	Number of workers to use for model prediction. Any non-zero integer accepted by Python joblib is allowed; -1 uses all available workers. Defaults to one worker for CRAN compatibility.

Value

A tibble of times and classification of category.

Examples

```
if (sl_python_modules_installed("models")) {
  models = sl_example_model()
  if (!is.null(models) && requireNamespace("readr", quietly = TRUE)) {
    file = system.file("extdata", "example_data.csv.gz", package = "sleeper")
    data = readr::read_csv(file, n_max = 3600L)
    # Two minutes is sufficient to demonstrate the workflow and avoids
    # applying the forest to the full 12-hour example recording.
    try({estimate_sleep(data, model_dir = dirname(models$binary[1]))})
  }
}
```

Index

`callr::r()`, 2
`curl::curl_download`, 4

`estimate_sleep`, 2
`estimate_sleep`
 (`sl_python_modules_installed`),
 5
`estimate_sleep()`, 2

`py_estimate_sleep`, 2
`py_require_sleeper`, 2, 2

`reticulate::py_require()`, 2

`sl_download_models(sl_have_models)`, 4
`sl_example_model`, 3
`sl_features`, 3
`sl_have_models`, 4
`sl_python_modules_installed`, 5