Package: asleep (via r-universe)

September 18, 2026

Title Estimate Sleep from 'Accelerometry' Data

Version 0.3.0

Description Interfaces the 'asleep' python module <https://github.com/OxWearables/asleep> from Yuan (2024) <doi:10.1038/s41746-024-01148-y> to estimate sleep from 'accelerometry' data.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports assertthat, curl, lubridate, methods, readr, reticulate (>= 1.42.0), rlang

Suggests tidyr, dplyr, ggplot2, testthat (>= 3.0.0), callr

Config/testthat/edition 3

URL https://github.com/jhuwit/asleep, https://jhuwit.github.io/asleep/

BugReports https://github.com/jhuwit/asleep/issues

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libpng-dev libssl-dev python3 libx11-dev

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

Date/Publication 2026-08-19 18:23:46 UTC

RemoteUrl https://github.com/jhuwit/asleep

RemoteRef HEAD

RemoteSha fba138941ed405ab503f1f145f80697070ba0b2e

Contents

asleep 2
get_sslnet 4
have_asleep 5

py_require_asleep 5
sl_download_models 6
sl_read 6
sl_rename_data 7

Index **8**

`asleep` *Run* `asleep` *Model on Data*

Description

Run `asleep` Model on Data

Run `asleep` with Python

Usage

```
asleep(
  file,
  outdir = NULL,
  min_wear_hours = 22L,
  time_shift = "0",
  report_light_and_temp = FALSE,
  pytorch_device = c("cpu", "cuda:0"),
  sample_rate = NULL,
  verbose = TRUE,
  force_download = FALSE
)

py_asleep(
  ...,
  pyenv_function = function() {
     asleep::py_require_asleep()
 },
  show = FALSE
)
```

Arguments

`file`	accelerometry file to process, including CSV, CWA, GT3X, and `GENEActiv` bin files
`outdir`	output directory for CSVs and outputs
`min_wear_hours`	Min wear time in hours to be eligible for summary statistics computation. The sleepnet paper uses 22
`time_shift`	The number hours to shift forward or backward from the current device time. e.g. +1 or -1

report_light_and_temp	If true, it adds mean temp, and light columns to the predictions
pytorch_device	device to use for prediction for PyTorch.
sample_rate	Sample rate for the data, currently not used.
verbose	print diagnostic messages
force_download	force a download of the model, passed to `sl_download_models()`
...	arguments to pass to asleep
pyenv_function	function that loads the forest Python package. By default, it uses `reticulate::py_import("asleep")` 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

A list of outputs, including summaries, paths, and dataframes.

Examples

```
asleep_check_result = function() {
  res = try({suppressWarnings(asleep_check())})
  if (inherits(res, "try-error")) {
    res = FALSE
  }
  res
}

  file = system.file("extdata/example_sleep.csv.gz", package = "asleep")
  stopifnot(file.exists(file))
  if (asleep_check_result()) {
    out = try({asleep(file = file, verbose = 2L)})
    if (inherits(out, "try-error")) {
      message(out)
      if (interactive()) {
        reticulate::py_last_error()
      }
    } else {
      pred = out$predictions
    }
  }


  file = system.file("extdata/example_sleep.csv.gz", package = "asleep")
  df = readr::read_csv(file)
  if (asleep_check_result()) {
    out = try({asleep(file = df)})
    if (inherits(out, "try-error")) {
      message(out)
    } else {
      st = out$predictions
```

```
    if (requireNamespace("ggplot2", quietly = TRUE) &&
        requireNamespace("tidyr", quietly = TRUE) &&
        requireNamespace("dplyr", quietly = TRUE)) {
      d = st[1:250,] |>
        dplyr::mutate(
          time = lubridate::as_datetime(time),
          time_end = dplyr::lead(time)
        ) |>
        dplyr::filter(!is.na(time_end))
      raw = df |> dplyr::filter(time >= min(d$time) & time <= max(d$time))
      dat = raw |>
        tidyr::gather(axis, value, -time)
      d = d |> dplyr::mutate(activity_y = as.numeric(sleep_wake == "sleep"))
      dat |>
        ggplot2::ggplot(ggplot2::aes(x = time, y = value, colour = axis)) +
        ggplot2::geom_step() +
        ggplot2::geom_segment(
          data = d,
          ggplot2::aes(
            x = time, xend = time_end,
            y = activity_y, yend = activity_y,
            linetype = sleep_wake
          ),
          colour = "black", linewidth = 1, inherit.aes = FALSE
        ) +
        ggplot2::labs(linetype = "Sleep/wake")
    }
  }
}
```

`get_sslnet` *Get SSL Net -*

Description

Get SSL Net -

Usage

```
get_sslnet()

sl_get_sslnet()
```

Value

A `keras` model representing the SSL Net.

have_asleep — *Check the* asleep *Python Module*

Description

Check the asleep Python Module

Usage

```
have_asleep()

asleep_check(...)

asleep_version(...)
```

Arguments

...	additional arguments to pass to reticulate::py_list_packages

Value

A logical value indicating whether the asleep Python module is available.

Examples

```
if (have_asleep()) {
   asleep_version()
}
```

py_require_asleep — *Command for* py_require *for* asleep

Description

Command for py_require for asleep

Usage

```
py_require_asleep(...)
```

Arguments

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

Value

A logical value indicating whether the package is available.

sl_download_models *Download* asleep *models to Cache*

Description

Download asleep models to Cache

Usage

```
sl_download_models(force_download = FALSE)
```

Arguments

force_download	force a download of the model, even if it already exists at model_path

Value

A model from Python. sl_download_model returns a model file path.

sl_read *Read a Data Set for* asleep

Description

Read a Data Set for asleep

Usage

```
sl_read(file, resample_hz = "uniform", keep_pandas = FALSE)
```

Arguments

file	path to the file for reading
resample_hz	Target frequency (Hz) to resample the signal. If "uniform", use the implied frequency (use this option to fix any device sampling errors). Pass NULL to disable. Defaults to "uniform".
keep_pandas	do not convert the data to a data.frame and keep as a pandas data.frame

Value

A list of the data and information about the data

Note

The data P30_wrist100 is from https://ora.ox.ac.uk/objects/uuid:19d3cb34-e2b3-4177-91b6-1bad0e0163e7, where we took the first 180,000 rows, the first 30 minutes of data from that participant as an example.

Examples

```
file = system.file("extdata/example_sleep.csv.gz", package = "asleep")
if (asleep_check()) {
  out = sl_read(file, resample_hz = FALSE)
}
```

`sl_rename_data` *Rename data for* `asleep`

Description

Rename data for `asleep`

Usage

```
sl_rename_data(data)

sl_write_csv(data, path = tempfile(fileext = ".csv"))
```

Arguments

`data`	a `data.frame` of raw accelerometry
`path`	path to the CSV output file

Value

A `data.frame` of renamed columns

Index

asleep, 2, *3*
asleep_check *(*have_asleep*)*, 5
asleep_version *(*have_asleep*)*, 5

callr::r(), *3*

get_sslnet, 4

have_asleep, 5

py_asleep *(*asleep*)*, 2
py_require_asleep, 5

reticulate::py_list_packages, *5*
reticulate::py_require(), *5*

sl_download_models, 6
sl_download_models(), *3*
sl_get_sslnet *(*get_sslnet*)*, 4
sl_read, 6
sl_rename_data, 7
sl_write_csv *(*sl_rename_data*)*, 7