Package: CircaCP (via r-universe)

May 12, 2026

Type Package

Title Sleep and Circadian Metrics Estimation from Actigraphy Data

Version 0.1.2

Maintainer Shanshan Chen <schen3@vcu.edu>

Description A generic sleep–wake cycle detection algorithm for analyzing unlabeled actigraphy data. The algorithm has been validated against event markers using data from the Multi-Ethnic Study of Atherosclerosis (MESA) Sleep study, and its methodological details are described in Chen and Sun (2024) <doi:10.1098/rsos.231468>. The package provides functions to estimate sleep metrics (e.g., sleep and wake onset times) and circadian rhythm metrics (e.g., mesor, phasor, interdaily stability, intradaily variability), as well as tools for screening actigraphy quality, fitting cosinor models, and performing parametric change point detection. The workflow can also be used to segment long actigraphy sequences into regularized structures for physical activity research.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports data.table, pracma, stats, tibble

Suggests ggplot2, knitr, minpack.lm, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 3.5)

NeedsCompilation no

Author Shanshan Chen [aut, cre], Jonathon Jacobs [ctb]

Repository https://shanshanchen-biostat.r-universe.dev

Date/Publication 2025-11-12 11:10:02 UTC

RemoteUrl https://github.com/cran/CircaCP

RemoteRef HEAD

RemoteSha 5906fe87bd58c8e0712b374fd2575a8dedb3166e

Contents

actigraphy 2
cp_detect 3
extract_nonparametric_metrics 3
extract_sleep_metrics 4
extract_sw_period 5
import_acti_file 6
screen_wear 7
sleep_cos 8
sleep_detection 9

Index **10**

`actigraphy` *Example Actigraphy Dataset*

Description

A single-subject actigraphy dataset from the NHANES 2013–2014 study, provided as an example for demonstrating the CircaCP workflow.

Usage

```
actigraphy
```

Format

A data frame with one row per recorded epoch and the following variables:

Date Date of recording (synthetic or reconstructed from NHANES information).

Time Time of day corresponding to each observation.

Lux Light intensity.

SDLux Short-term variability of light intensity.

MIMS Activity magnitude calculated using the MIMS algorithm.

X Raw or calibrated X-axis signal from the accelerometer.

Y Raw or calibrated Y-axis signal from the accelerometer.

Z Raw or calibrated Z-axis signal from the accelerometer.

Source

National Health and Nutrition Examination Survey (NHANES) 2013–2014.

Examples

```
data(actigraphy)
head(actigraphy)
```

`cp_detect` *Detect a single change point (parametric methods)*

Description

Detect a single change point (parametric methods)

Usage

```
cp_detect(M, dist)
```

Arguments

`M`	minute-level Activity vector;
`dist`	Actigraphy data distribution family, including `Gaussian`, `Gamma`, `ZAG` (Zero-Augmented Gamma), `Poisson`, `Exponential`

Value

the location of the single change point

See Also

`sleep_detection()`

`extract_nonparametric_metrics`
Nonparametric circadian metrics (RA, IS, IV, L5, M10)

Description

Computes five nonparametric metrics from minute-level activity: Relative Amplitude (RA), Interdaily Stability (IS), Intradaily Variability (IV), the least-active 5-hour block (L5), and the most-active 10-hour block (M10).

Usage

```
extract_nonparametric_metrics(df, L_window = 5 * 60, M_window = 10 * 60)
```

Arguments

df	cleaned data frame containing the activity varialbe at 1-minute epoch
L_window	Integer window length (minutes) for L5 (default 300).
M_window	Integer window length (minutes) for M10 (default 600).

Value

A list with components:

RA Relative amplitude, $(M10 - L5)/(M10 + L5)$ when denominator > 0.

IS Interdaily stability (0..1).

IV Intradaily variability (≥ 0).

L5_mean Mean activity in the lowest 5-hour block of the 24 h profile.

L5_start_min Minute-of-day (0~1439) at which L5 starts.

M10_mean Mean activity in the highest 10-hour block of the 24 h profile.

M10_start_min Minute-of-day (0~1439) at which M10 starts.

profile_24h Length-1440 vector of minute-of-day means.

See Also

screen_wear(), sleep_detection(), extract_sleep_metrics

---

extract_sleep_metrics *Extract metrics related to sleep and circadian rhythm after using CircaCP algorithm*

Description

From minute-level data with sleep/wake labels (label.sw 1 = sleep, 0 = wake), extracts sleep/wake onsets, episode durations, circular SDs of onset times, Sleep Regularity Index (SRI), cosinor parameters, day/night variance ratio, and nonparametric metrics (RA, IS, IV, L5, M10). Returns one row per episode with scalar metrics repeated per row (tidy format).

Usage

```
extract_sleep_metrics(df, min_sleep_episode_min = 180L)
```

Arguments

df	data.frame with columns id, Date, Time, Activity, label.sw.
min_sleep_episode_min	Minimum duration (minutes) to treat as a main sleep episode.

Value

A `data.frame` with columns including:

id Subject ID obtained from the stem of filename.

period_type sleep or wake

timestamp datetime of sleep onset time and wake onset time

clock_min timestamps presented as minutes of the day since midnight

duration_hours sleep duration or wake duration

SleepTimeSD_hours Standard devation of sleep onset time (calculated by circular statistics)

WakeTimeSD_hours Standard devation of sleep onset time (calculated by circular statistics)

SRI Sleep regularity index

Mesor, Amplitude, Acrophase parameters obtained from the cosinor model

RA, IS, IV, L5_mean, L5_start_min, M10_mean, M10_start_min nonparametrics actigraphy metrics

See Also

`sleep_detection()`, `sleep_cos()`, `extract_nonparametric_metrics()`

`extract_sw_period` *Extract contiguous sleep or wake periods from a labeled minute series*

Description

Splits a minute-level, labeled time series into contiguous episodes of a target state (sleep or wake) and returns the **activity vectors** for those episodes whose length is at least `min_len` minutes.

Usage

```
extract_sw_period(labeled_df, target_state = 1, min_len = 30)
```

Arguments

`labeled_df`	A `data.frame` containing at least: • `Activity` — numeric minute-level activity. • `label.sw` — binary sleep/wake label per minute (`1` = sleep, `0` = wake). `NA` values are treated as breaks between episodes.
`target_state`	Integer `0` or `1`. Use `0` to extract sleep episodes, `1` to extract wake episodes. Default is `1`.
`min_len`	Integer minimum episode length **in minutes** (i.e., number of consecutive samples) required to keep an episode. Default is `30`.

Details

The function uses run-length encoding over `label.sw` to identify contiguous episodes. Any `NA` in `label.sw` is converted to a sentinel and treated as a hard break (i.e., episodes do not cross `NA` gaps). Length filtering is applied on the number of minutes (rows) per episode.

Value

A **list** of numeric vectors. Each element is the `Activity` values for one qualifying episode of the requested state. If no episode qualifies, returns an empty list (`list()`).

See Also

`sleep_detection()`

#' @export

`import_acti_file` *Import actigraphy with header stripping and harmonized* `Activity`

Description

Reads a CSV/TSV, automatically strips any leading header lines, selects user specified `Date` and `Time` columns, and harmonizes one chosen activity column to the canonical name `Activity`. Adds an `id` (defaults to the file stem if not supplied).

Usage

```
import_acti_file(
  file,
  date_col,
  time_col,
  activity_cols,
  id = NULL,
  keep_extra = FALSE,
  drop_original_activity_cols = TRUE
)
```

Arguments

`file`	Character path to the actigraphy file.
`date_col, time_col`	Character names of the date and time columns in `file`.
`activity_cols`	Character vector of candidate activity columns; the first that exists will be used and renamed to `Activity`.
`id`	Optional subject id (character). Defaults to the filename stem.
`keep_extra`	Logical; if `FALSE` keeps only `id, Date, Time, Activity`, otherwise preserves extra columns present in the file.
`drop_original_activity_cols`	Logical; if TRUE, drop the original activity

Details

Internally uses `data.table::fread()` with automatic header detection (skips lines before the first row that looks like a header). No time zone conversion is performed.

Value

A `data.frame` with at least columns `id`, `Date`, `Time`, `Activity`.

See Also

`screen_wear()`

---

screen_wear *Screen wear and extract the longest valid minute-level segment*

Description

Determines the native epoch from the first two time stamps, decimates to 1-minute resolution (if needed), checks total minutes $\geq$ `1440*min_days`, and within that finds the **longest** contiguous segment where runs of consecutive zeros do not exceed `max_zero_run` minutes.

Usage

```
screen_wear(
  df,
  min_days = 5L,
  max_zero_run = 120L,
  date_col = "Date",
  time_col = "Time",
  activity_col = "Activity"
)
```

Arguments

`df`	`data.frame` with columns `Date`, `Time`, `Activity` in time order.
`min_days`	Integer minimum number of whole days required (default 5).
`max_zero_run`	Integer maximum allowed length (minutes) of a run of zeros.
`date_col`	Name of the date column.
`time_col`	Name of the time column.
`activity_col`	Name of the activity column used to determine wear/non-wear.

Details

Decimation rule: 15 s $\rightarrow$ factor 4; 30 s $\rightarrow$ factor 2; 60 s $\rightarrow$ factor 1. The zero-run criterion is applied on the 1-minute `Activity` series.

Value

A list with elements:

status "ok" if a qualifying segment is found, otherwise a message.

epoch_inferred Detected original epoch in seconds (15, 30, 60).

out_idx Indices of the kept segment in the input.

clean_df Minute-level data.frame of the selected segment (if ok).

See Also

import_acti_file(), sleep_cos(), sleep_detection()

---

sleep_cos *Estimates circadian cycle by cosinor fit of minute-level activity (period = 1440)*

Description

Fits $C(t) = m + a\cos\{2\pi(t-\phi)/P\}$ with period $P = 1440$ minutes to a rescaled activity series. Returns fitted curve, binary curve after thresholding, and canonicalized parameters (non-negative amplitude). MESOR (mesor) The baseline or "midline" level around which the rhythm oscillates. Units = same as actigraphy your data Roughly the average activity across the cycle.

Usage

```
sleep_cos(clean_df, thr = 0.2)
```

Arguments

clean_df	cleaned dataframe with cleaned_df$Activity.
thr	Dichotomization threshold for fitted curve.

Details

Amplitude (amp) Half the peak-to-trough swing of the fitted rhythm. Units is the same as actigraphy data. Larger amplitude → stronger rhythmicity (bigger day–night contrast). Negative amp will be flipped, and P/2 will be added to phase. Acrophase (phase) The timing of the peak of the fitted cosine. Units is the epoch units of the data (e.g. minutes). P=1440, interpret phase as minutes-of-day relative to your x origin.

Value

A list with elements:

fitted fitted cosine curve.

label.cos dichotimized cosine curve

cos_para c(Mesor, Amplitude, Acrophase).

rmse Root mean squared error between fitted cosine curve and data.

See Also

screen_wear(), sleep_detection(), extract_sleep_metrics()

`sleep_detection` *Estimate precise sleep_wake cycles using CircaCP algorithm*

Description

It first uses a 24 h cosinor fit (via sleep_cos()) and a chosen thresholding rule to label each minute as sleep (`label.sw = 1`) or wake (`label.sw = 0`). Reference: Shanshan Chen, and Xinxin Sun.Validating CircaCP: a generic sleep–wake cycle detection algorithm for unlabelled actigraphy data. Royal Society Open Science 11, no. 5 (2024): 231468.

Usage

```
sleep_detection(clean_df, thr = 0.2, dist = "ZAG")
```

Arguments

`clean_df`	Minute-level `data.frame` with at least `Activity`. Additional columns are preserved.
`thr`	Numeric threshold in [0, 1] applied to a rescaled cosinor fit.
`dist`	Character method key (e.g., `"ZAG"`); interpreted by your rule set.

Value

a data.frame augmenting the input `df` including the following additional variables:

cosinor fitted cosine curve

label.cos circadian cycle estimated by dichotimized cosine curve

label.sw sleep-wake cycle estimated by CircaCP

Activity_norm range-normalized activity levels

See Also

screen_wear(), sleep_cos(),cp_detect(), extract_sleep_metrics()

Index

∗ **datasets**
actigraphy, 2

actigraphy, 2

cp_detect, 3
cp_detect(), *9*

extract_nonparametric_metrics, 3
extract_nonparametric_metrics(), *5*
extract_sleep_metrics, 4, *4*
extract_sleep_metrics(), *9*
extract_sw_period, 5

import_acti_file, 6
import_acti_file(), *8*

screen_wear, 7
screen_wear(), *4*, *7*, *9*
sleep_cos, 8
sleep_cos(), *5*, *8*, *9*
sleep_detection, 9
sleep_detection(), *3–6*, *8*, *9*