Package 'ActFrag'

January 18, 2026

Type Package

Title Activity Fragmentation Metrics Extracted from Minute Level Activity Data

Version 0.1.2

Maintainer Junrui Di <dijunrui@gmail.com>

Description Recent studies haven shown that, on top of total daily active/sedentary volumes, the time accumulation strategies provide more sensitive information. This package provides functions to extract
commonly used fragmentation metrics to quantify such time accumulation strategies based on minute level
actigraphy-measured activity counts data.

License GPL-3

Imports accelerometry, dplyr, ineq, survival, stats

Depends R (>= 4.4.0),

Suggests knitr, rmarkdown, testthat (>= 2.1.0)

Encoding UTF-8

LazyData TRUE

ByteCompile true

VignetteBuilder knitr

URL https://github.com/junruidi/ActFrag

BugReports https://github.com/junruidi/ActFrag/issues

RoxygenNote 7.3.3

NeedsCompilation no

Author Junrui Di [aut, cre],
John Muschelli [aut],
Vadim zipunnikov [aut]

Repository CRAN

Date/Publication 2026-01-18 06:10:33 UTC

Contents

`example_activity_data` *Activity/Wear Data from 50 Subjects from NHANES 2003 - 2006*

Description

A list of two `data.frames` containing the counts and the weartime for 50 NHANES subjects

Usage

```
example_activity_data
```

Format

A list of two `data.frames` with 1442 columns, which are in the following order:

ID identifier of the person.

Day `numeric` sequence 1,2,.. indicating the order of days within a subject.

MIN1-MIN1440 counts of activity of that specific minute.

`fragmentation` *Fragmentation Metrics*

Description

Fragmentation methods to study the transition between two states, e.g. sedentary v.s. active.

Usage

```
fragmentation(
  x,
  w,
  thresh,
  bout.length = 1,
  metrics = c("mean_bout", "TP", "Gini", "power", "hazard", "all")
)
```

Arguments

x	integer vector of activity data.
w	vector of wear flag data with same dimension as x.
thresh	threshold to binarize the data.
bout.length	minimum duration of defining an active bout; defaults to 1.
metrics	What is the fragmentation metrics to exract. Can be "mean_bout","TP","Gini","power","hazard",or all the above metrics "all".

Details

Metrics include mean_bout (mean bout duration), TP (between states transition probability), Gini (gini index), power (alapha parameter for power law distribution) hazard (average hazard function)

Value

A list with elements

mean_r	mean sedentary bout duration
mean_a	mean active bout duration
SATP	sedentary to active transition probability
ASTP	bactive to sedentary transition probability
Gini_r	Gini index for active bout
Gini_a	Gini index for sedentary bout
h_r	hazard function for sedentary bout
h_a	hazard function for active bout
alpha_r	power law parameter for sedentary bout
alpha_a	power law parameter for active bout

References

Junrui Di, Andrew Leroux, Jacek Urbanek, Ravi Varadhan, Adam P. Spira, Jennifer Schrack, Vadim Zipunnikov. Patterns of sedentary and active time accumulation are associated with mortality in US adults: The NHANES study. bioRxiv 182337; doi: https://doi.org/10.1101/182337

Examples

```
data(example_activity_data)
count1 = c(t(example_activity_data$count[1,-c(1,2)]))
wear1 = c(t(example_activity_data$wear[1,-c(1,2)]))
frag = fragmentation(x = count1, w = wear1, thresh = 100, bout.length = 1, metrics = "mean_bout")
frag = fragmentation(x = count1, w = wear1, thresh = 100,
bout.length = 1, metrics = "all")
res = sapply(c("mean_bout","TP","Gini","power","hazard"), function(x) {
frag = fragmentation(x = count1, w = wear1,
thresh = 100, bout.length = 1, metrics = x)
})
```

```
data(example_activity_data)
count1 = c(t(example_activity_data$count[1,-c(1,2)]))
wear1 = c(t(example_activity_data$wear[1,-c(1,2)]))
count1[ !is.na(count1) & count1 != 0] = 0L
res = sapply(c("mean_bout","TP","Gini","power","hazard", "all"), function(x) {
frag = fragmentation(x = count1, w = wear1,
thresh = 100, bout.length = 1, metrics = x)
})
```

`fragmentation_long` *Fragmentation Metrics for Whole Dataset*

Description

Fragmentation methods to study the transition between two states, e.g. sedentary v.s. active.This function is a whole dataset wrapper for `fragmentation`

Usage

```
fragmentation_long(
  count.data,
  weartime,
  thresh,
  bout.length = 1,
  metrics = c("mean_bout", "TP", "Gini", "power", "hazard", "all"),
  by = c("day", "subject")
)
```

Arguments

`count.data`	`data.frame` of dimension n*1442 containing the 1440 minutes of activity data for all n subject days. The first two columns have to be ID and Day. ID can be either `character` or `numeric`. Day has to be `numeric` indicating the sequency of days within each subject.
`weartime`	`data.frame` with dimension of `count.data`. The first two columns have to be ID and Day.ID can be either `character` or `numeric`. Day has to be `numeric` indicating the sequencey of days within each subject.
`thresh`	threshold to define the two states.
`bout.length`	minimum duration of defining an active bout; defaults to 1.
`metrics`	What is the fragmentation metrics to exract. Can be "mean_bout","TP","Gini","power","hazard",or all the above metrics "all".
`by`	Determine whether fragmentation is calcualted by day or by subjects (i.e. aggregate bouts across days). by-subject is recommended to gain more power.

Details

Metrics include mean_bout (mean bout duration), TP (between states transition probability), Gini (gini index), power (alapha parameter for power law distribution) hazard (average hazard function)

Value

A dataframe with some of the following columns

ID	identifier of the person
Day	numeric vector indicating the sequencey of days within each subject.
mean_r	mean sedentary bout duration
mean_a	mean active bout duration
SATP	sedentary to active transition probability
ASTP	bactive to sedentary transition probability
Gini_r	Gini index for active bout
Gini_a	Gini index for sedentary bout
h_r	hazard function for sedentary bout
h_a	hazard function for active bout
alpha_r	power law parameter for sedentary bout
alpha_a	power law parameter for active bout

Examples

```
data(example_activity_data)
count = example_activity_data$count
wear = example_activity_data$wear
frag_by_day = fragmentation_long(count.data = count,
weartime = wear,thresh = 100,bout.length = 1,
metrics = "all",by = "day")
tp_by_subject = fragmentation_long(count.data = count,
weartime = wear,thresh = 100,bout.length = 1,
metrics = "TP",by = "subject")


res = sapply(c("mean_bout","TP","Gini","power","hazard", "all"), function(x) {
frag_by_day = fragmentation_long(count.data = count,
weartime = wear,thresh = 100,bout.length = 1,
metrics = x,by = "day")
})
res = sapply(c("mean_bout","TP","Gini","power","hazard", "all"), function(x) {
tp_by_subject = fragmentation_long(count.data = count,
weartime = wear,thresh = 100,bout.length = 1,
metrics = x,by = "subject")
})
```

wear_flag *Create Wear/Nonwear Flags*

Description

Determine during which time period, subject should wear the device. It is preferable that user provide their own wear/non wear flag which should has the same dimension as the activity data. This function provide wear/non wear flag based on time of day.

Usage

```
wear_flag(count.data, start = "05:00", end = "23:00")
```

Arguments

count.data	data.frame of dimension n*1442 containing the 1440 minute activity data for all n subject days. The first two columns have to be ID and Day.
start	start time, a string in the format of 24hr, e.g. "05:00"; defaults to "05:00".
end	end time, a string in the format of 24hr, e.g. "23:00"; defaults to "23:00"

Details

Fragmentation metrics are usually defined when subject is awake. The weartime provide time periods on which those features should be extracted. This can be also used as indication of wake/sleep.

Value

A data.frame with same dimension and column name as the count.data, with 0/1 as the elments reprensting wear, nonwear respectively.

Examples

```
data(example_activity_data)
count = example_activity_data$count
weartime = wear_flag(count.data = count)
testthat::expect_error({
weartime = wear_flag(count.data = count, start = "10:00PM")
})
```

Index