# Package ‘ActCR’

January 12, 2026

**Type** Package

**Title** Extract Circadian Rhythms Metrics from Actigraphy Data

**Version** 0.4.0

**Maintainer** Junrui Di <dijunrui@gmail.com>

**Description** Circadian rhythms are rhythms that oscillate about every 24 h, which has been observed in multiple physiological processes including core body temperature, hormone secretion, heart rate, blood pressure, and many others. Measuring circadian rhythm with wearables is based on a principle that there is increased movement during wake periods and reduced movement during sleep periods, and has been shown to be reliable and valid. This package can be used to extract nonparametric circadian metrics like intradaily variability (IV), interdaily stability (IS), and relative amplitude (RA); and parametric cosinor model and extended cosinor model coefficient. Details can be found in Junrui Di et al (2019) <doi:10.1007/s12561-019-09236-4>.

**License** GPL-3

**Imports** zoo, cosinor, cosinor2, dplyr, minpack.lm

**Depends** R (>= 4.4.0),

**Encoding** UTF-8

**LazyData** true

**URL** https://github.com/junruidi/ActCR

**BugReports** https://github.com/junruidi/ActCR/issues

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Author** Junrui Di [aut, cre],
Vadim zipunnikov [aut],
Vincent van Hees [ctb]

**Repository** CRAN

**Date/Publication** 2026-01-12 06:11:19 UTC

# Contents

## `ActCosinor` *Cosinor Model for Circadian Rhythmicity*

### Description

A parametric approach to study circadian rhythmicity assuming cosinor shape.

### Usage

```
ActCosinor(x, window = 1, export_ts = FALSE)
```

### Arguments

`x` `vector` vector of dimension n*1440 which reprsents n days of 1440 minute activity data

`window` The calculation needs the window size of the data. E.g window = 1 means each epoch is in one-minute window.

`export_ts` A Boolean to indicate whether time series should be exported

### Value

A list with elements

`mes` MESOR which is short for midline statistics of rhythm, which is a rhythm adjusted mean. This represents mean activity level.

`amp` amplitude, a measure of half the extend of predictable variation within a cycle. This represents the highest activity one can achieve.

`acro` acrophase, a meaure of the time of the overall high values recurring in each cycle. Here it has a unit of radian. This represents time to reach the peak.

`acrotime` acrophase in the unit of the time (hours) `ndays` Number of days modeled

`cosinor_ts` Exported data frame with time, time over days, original time series, fitted time series using cosinor model

`pr` Percent rhythm, which evaluates the elative power of the rhythm by checking the proportion of the variance explained by the model

`pr_pval` P values for the percent rhythm

### References

Cornelissen, G. Cosinor-based rhythmometry. Theor Biol Med Model 11, 16 (2014). https://doi.org/10.1186/1742-4682-11-16

### Examples

```
count1 = c(t(example_activity_data$count[c(1:2),-c(1,2)]))
cos_coeff = ActCosinor(x = count1, window = 1, export_ts = TRUE)
```

## `ActCosinor_long` *Cosinor Model for Circadian Rhythmicity for the Whole Dataset*

### Description

A parametric approach to study circadian rhythmicity assuming cosinor shape.This function is a whole dataset wrapper for `ActCosinor`.

### Usage

```
ActCosinor_long(count.data, window = 1, export_ts = FALSE)
```

### Arguments

`count.data` `data.frame` of dimension n * (p+2) containing the p dimensional 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 sequence of days within each subject.

`window` The calculation needs the window size of the data. E.g window = 1 means each epoch is in one-minute window.

`export_ts` A Boolean to indicate whether time series should be exported (notice: it takes time and storage space to export time series data for all subject-days. Use this with caution. Suggest to only export time series for selected subjects)

## Value

A data.frame with the following 5 columns

ID ID

ndays number of days

mes MESRO, which is short for midline statistics of rhythm, which is a rhythm adjusted mean. This represents mean activity level.

amp amplitude, a measure of half the extend of predictable variation within a cycle. This represents the highest activity one can achieve.

acro acrophase, a meaure of the time of the overall high values recurring in each cycle. Here it has a unit of radian. This represents time to reach the peak.

acrotime acrophase in the unit of the time (hours)

ndays Number of days modeled

pr Percent rhythm, which evaluates the elative power of the rhythm by checking the proportion of the variance explained by the model

pr_pval P values for the percent rhythm

cosinor_ts Exported data frame with time, time over days, original time series, fitted time series using cosinor model

## Examples

```
counts_1 = example_activity_data$count[c(1:12),]
cos_all_1 = ActCosinor_long(count.data = counts_1, window = 1,export_ts = TRUE)
counts_10 = cbind(counts_1[,1:2],
as.data.frame(t(apply(counts_1[,-c(1:2)], 1,
FUN = bin_data, window = 10, method = ”average”))))
cos_all_10 = ActCosinor_long(count.data = counts_10, window = 10)
```

# ActExtendCosinor *Extended Cosinor Model for Circadian Rhythmicity*

## Description

Extended cosinor model based on sigmoidally transformed cosine curve using anti-logistic transformation

## Usage

```
ActExtendCosinor(
  x,
  window = 1,
  lower = c(0, 0, -1, 0, -3),
  upper = c(Inf, Inf, 1, Inf, 27),
  export_ts = FALSE
)
```

## Arguments

`x` `vector` vector of dimension n*1440 which represents n days of 1440 minute activity data

`window` The calculation needs the window size of the data. E.g window = 1 means each epoch is in one-minute window.

`lower` A numeric vector of lower bounds on each of the five parameters (in the order of minimum, amplitude, alpha, beta, acrophase) for the NLS. If not given, the default lower bound for each parameter is set to `-Inf`.

`upper` A numeric vector of upper bounds on each of the five parameters (in the order of minimum, amplitude, alpha, beta, acrophase) for the NLS. If not given, the default lower bound for each parameter is set to `Inf`

`export_ts` A Boolean to indicate whether time series should be exported

## Value

A list with elements

`minimum` Minimum value of the of the function.

`amp` amplitude, a measure of half the extend of predictable variation within a cycle. This represents the highest activity one can achieve.

`alpha` It determines whether the peaks of the curve are wider than the troughs: when alpha is small, the troughs are narrow and the peaks are wide; when alpha is large, the troughs are wide and the peaks are narrow.

`beta` It dertermines whether the transformed function rises and falls more steeply than the cosine curve: large values of beta produce curves that are nearly square waves.

`acrotime` acrophase is the time of day of the peak in the unit of the time (hours)

`F_pseudo` Measure the improvement of the fit obtained by the non-linear estimation of the transformed cosine model

`UpMesor` Time of day of switch from low to high activity. Represents the timing of the rest- activity rhythm. Lower (earlier) values indicate increase in activity earlier in the day and suggest a more advanced circadian phase.

`DownMesor` Time of day of switch from high to low activity. Represents the timing of the rest-activity rhythm. Lower (earlier) values indicate decline in activity earlier in the day, suggesting a more advanced circadian phase.

`MESOR` A measure analogous to the MESOR of the cosine model (or half the deflection of the curve) can be obtained from mes=min+amp/2. However, it goes through the middle of the peak, and is therefore not equal to the MESOR of the cosine model, which is the mean of the data.

`ndays` Number of days modeled.

`cosinor_ts` Exported data frame with time, time over days, original time series, fitted time series using cosinor model from step 1, and fitted extended cosinor model from step 2

### References

Marler MR, Gehrman P, Martin JL, Ancoli-Israel S. The sigmoidally transformed cosine curve: a mathematical model for circadian rhythms with symmetric non-sinusoidal shapes. Stat Med.

### Examples

```
count1 = c(t(example_activity_data$count[c(1:2),-c(1,2)]))
cos_coeff = ActExtendCosinor(x = count1, window = 1,export_ts = TRUE)
```

## `ActExtendCosinor_long` *Cosinor Model for Circadian Rhythmicity for the Whole Dataset*

### Description

Extended cosinor model based on sigmoidally transformed cosine curve using anti-logistic transformation.This function is a whole dataset wrapper for `ActExtendCosinor`.

### Usage

```
ActExtendCosinor_long(
  count.data,
  window = 1,
  lower = c(0, 0, -1, 0, -3),
  upper = c(Inf, Inf, 1, Inf, 27),
  export_ts = FALSE
)
```

### Arguments

`count.data` `data.frame` of dimension n * (p+2) containing the p dimensional 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 sequence of days within each subject.

`window` The calculation needs the window size of the data. E.g window = 1 means each epoch is in one-minute window. window size as an argument.

`lower` A numeric vector of lower bounds on each of the five parameters (in the order of minimum, amplitude, alpha, beta, acrophase) for the NLS. If not given, the default lower bound for each parameter is set to `-Inf`.

`upper` A numeric vector of upper bounds on each of the five parameters (in the order of minimum, amplitude, alpha, beta, acrophase) for the NLS. If not given, the default lower bound for each parameter is set to `Inf`

`export_ts` A Boolean to indicate whether time series should be exported (notice: it takes time and storage space to export time series data for all subject-days. Use this with caution. Suggest to only export time series for selected subjects)

## Value

A `data.frame` with the following 11 columns

`ID` ID

`ndays` number of days

`minimum` Minimum value of the of the function.

`amp` amplitude, a measure of half the extend of predictable variation within a cycle. This represents the highest activity one can achieve.

`alpha` It determines whether the peaks of the curve are wider than the troughs: when alpha is small, the troughs are narrow and the peaks are wide; when alpha is large, the troughs are wide and the peaks are narrow.

`beta` It dertermines whether the transformed function rises and falls more steeply than the cosine curve: large values of beta produce curves that are nearly square waves.

`acrotime` acrophase is the time of day of the peak in the unit of the time (hours)

`F_pseudo` Measure the improvement of the fit obtained by the non-linear estimation of the transformed cosine model

`UpMesor` Time of day of switch from low to high activity. Represents the timing of the rest- activity rhythm. Lower (earlier) values indicate increase in activity earlier in the day and suggest a more advanced circadian phase.

`DownMesor` Time of day of switch from high to low activity. Represents the timing of the rest-activity rhythm. Lower (earlier) values indicate decline in activity earlier in the day, suggesting a more advanced circadian phase.

`MESOR` A measure analogous to the MESOR of the cosine model (or half the deflection of the curve) can be obtained from mes=min+amp/2. However, it goes through the middle of the peak, and is therefore not equal to the MESOR of the cosine model, which is the mean of the data.

`cosinor_ts` Exported data frame with time, time over days, original time series, fitted time series using cosinor model from step 1, and fitted extended cosinor model from step 2

## Examples

```
counts_1 = example_activity_data$count[c(1:12),]
cos_all_1 = ActExtendCosinor_long(count.data = counts_1, window = 1, export_ts = TRUE)
counts_10 = cbind(counts_1[,1:2],
as.data.frame(t(apply(counts_1[,-c(1:2)], 1,
FUN = bin_data, window = 10, method = "average"))))
cos_all_10 = ActExtendCosinor_long(count.data = counts_10, window = 10, export_ts = FALSE)
```

## bin_data *Bin data into longer windows*

### Description

Bin minute level data into different time resolutions

### Usage

```
bin_data(x = x, window = 1, method = c("average", "sum"))
```

### Arguments

| | |
|---|---|
| `x` | `vector` of activity data. |
| `window` | window size used to bin the original 1440 dimensional data into. Window size should be an integer factor of 1440 |
| `method` | `character` of "sum" or "average", function used to bin the data |

### Value

a vector of binned data

### Examples

```
data(example_activity_data)
count1 = c(t(example_activity_data$count[1,-c(1,2)]))
xbin = bin_data(x = count1, window = 10, method = "average")
```

## 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.

## IS *Interdaily Statbility*

### Description

This function calcualte interdaily stability, a nonparametric metric of circadian rhtymicity

### Usage

```
IS(x)
```

### Arguments

`x` `data.frame` of dimension ndays by p, where p is the dimension of the data.

### References

Junrui Di et al. Joint and individual representation of domains of physical activity, sleep, and circadian rhythmicity. Statistics in Biosciences.

### Examples

```
data(example_activity_data)
count1 = example_activity_data$count[c(1,2,3),-c(1,2)]
is = IS(x = count1)
```

## IS_long *Interdaily Statbility for the Whole Dataset*

### Description

This function calcualte interdaily stability, a nonparametric metric of circadian rhtymicity. This function is a whole dataset wrapper for `IS`

### Usage

```
IS_long(count.data, window = 1, method = c("average", "sum"))
```

### Arguments

`count.data` `data.frame` of dimension n * (1440+2) containing the 1440 dimensional 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.

`window` an `integer` indicating what is the window to bin the data before the function can be apply to the dataset. For details, see `bin_data`.

`method` `character` of "sum" or "average", function used to bin the data

### Value

A data.frame with the following 2 columns

| | |
|---|---|
| ID | ID |
| IS | IS |

### References

Junrui Di et al. Joint and individual representation of domains of physical activity, sleep, and circadian rhythmicity. Statistics in Biosciences.

### Examples

```
data(example_activity_data)
count1 = example_activity_data$count
is_subj = IS_long(count.data = count1, window = 10, method = "average")
```

## IV *Intradaily Variability*

### Description

This function calcualte intradaily variability, a nonparametric metric reprsenting fragmentation of circadian rhtymicity

### Usage

```
IV(x)
```

### Arguments

| | |
|---|---|
| x | vector of activity data |

### Value

IV

### References

Junrui Di et al. Joint and individual representation of domains of physical activity, sleep, and circadian rhythmicity. Statistics in Biosciences.

### Examples

```
data(example_activity_data)
count1 = c(t(example_activity_data$count[1,-c(1,2)]))
iv = IV(x = count1)
```

## IV_long *Intradaily Variability for the Whole Dataset*

### Description

This function calcualte intradaily variability, a nonparametric metric reprsenting fragmentation of circadian rhtymicity. This function is a whole dataset wrapper for `IV`.

### Usage

```
IV_long(count.data, window = 1, method = c("average", "sum"))
```

### Arguments

| | |
|---|---|
| `count.data` | `data.frame` of dimension n * (1440+2) containing the 1440 dimensional 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. |
| `window` | an `integer` indicating what is the window to bin the data before the function can be apply to the dataset. For details, see `bin_data`. |
| `method` | `character` of "sum" or "average", function used to bin the data |

### Value

A `data.frame` with the following 5 columns

| | |
|---|---|
| `ID` | ID |
| `Day` | Day |
| `IV` | IV |

### References

Junrui Di et al. Joint and individual representation of domains of physical activity, sleep, and circadian rhythmicity. Statistics in Biosciences.

### Examples

```
data(example_activity_data)
count1 = example_activity_data$count
iv_subj = IV_long(count.data = count1, window = 10, method = "average")
```

## RA *Relative Amplitude*

### Description

This function calcualte relative amplitude, a nonparametric metric reprsenting fragmentation of circadian rhtymicity

### Usage

```
RA(x, window = 1, method = c("average", "sum"))
```

### Arguments

| | |
|---|---|
| x | vector vector of activity data |
| window | since the caculation of M10 and L5 depends on the dimension of data, we need to include window size as an argument. |
| method | character of "sum" or "average", function used to bin the data |

### Value

A list with elements

| | |
|---|---|
| M10 | Maximum 10 hour activity |
| L5 | Minimum 5 hour activity |
| RA | Relative amplitude |

### References

Junrui Di et al. Joint and individual representation of domains of physical activity, sleep, and circadian rhythmicity. Statistics in Biosciences.

### Examples

```
data(example_activity_data)
count1 = c(t(example_activity_data$count[1,-c(1,2)]))
ra = RA(x = count1, window = 10, method = "average")
```

# RA_long *Relative Amplitude for the Whole Datset*

## Description

This function calcualte relative amplitude, a nonparametric metric of circadian rhtymicity. This function is a whole dataset wrapper for `RA`.

## Usage

```
RA_long(count.data, window = 1, method = c("average", "sum"))
```

## Arguments

`count.data` `data.frame` of dimension n * (p+2) containing the p dimensional 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.

`window` since the caculation of M10 and L5 depends on the dimension of data, we need to include window size as an argument. This function is a whole dataset wrapper for `RA`.

`method` `character` of "sum" or "average", function used to bin the data

## Value

A `data.frame` with the following 3 columns

| | |
|---|---|
| `ID` | ID |
| `Day` | Day |
| `RA` | RA |

## Examples

```
data(example_activity_data)
count1 = example_activity_data$count[1:12,]
ra_all = RA_long(count.data = count1, window = 10, method = "average")
```

# Index