Package ‘rain’

November 22, 2016

Type Package
Title Rhythmicity Analysis Incorporating Non-parametric Methods
Version 1.8.0
Date 2015-09-01
Author Paul F. Thaben, Pål O. Westermark
Maintainer Paul F. Thaben <paul.thaben@charite.de>
Description This package uses non-parametric methods to detect rhythms in time series. It deals with outliers, missing values and is optimized for time series comprising 10-100 measurements. As it does not assume any distinct waveform it is optimal for detecting oscillating behavior (e.g. circadian or cell cycle) in e.g. genome- or proteome-wide biological measurements such as: micro arrays, proteome mass spectrometry, or metabolome measurements.
License GPL-2
Depends R (>= 2.10), gmp, multtest
biocViews TimeCourse, Genetics, SystemsBiology, Proteomics, Microarray, MultipleComparison
Suggests lattice, BiocStyle
Encoding UTF-8
NeedsCompilation no

R topics documented:

rain-package .......................................................... 2
menetRNASeqMouseLiver ......................................... 2
rain ................................................................. 3

Index 6
Description

Package Description

Details

Start with function 'rain'

Author(s)

Paul Thaben <paul.thaben@charite.de>

Usage

menetRNASeqMouseLiver

Format

A data frame containing the normalized expression values for each gene and time point in two repeats. First number shows the time of measurement in 'Zeitgeber Time' (ZT) whereas ZT_0 is the time of light on. The second number specifies the biological replicate.

Author(s)

Paul F. Thaben <paul.thaben@charite.de>

References

rain

Detection of rhythmic behavior in time series.

Description

rain detects rhythms in time-series using non-parametric methods. It uses an extension of the rank test for Umbrella Alternatives (Mack & Wolfe, 1981), based on the Jonckheere-Terpstra test, which tests whether sets of groups have a trend or not. The Umbrella method extends this to independent rising and falling sets.

Usage

```r
rain(x, deltat, period, period.delta = 0, peak.border = c(0.3, 0.7),
    nr.series = 1, measure.sequence = NULL, method = "independent",
    na.rm = FALSE, adjp.method = "ABH", verbose = getOption("verbose"))
```

Arguments

- **x**: numeric array, containing the data. One row per time point, one column per sample. If more than one replicate is done, see `nr.series` for formatting.
- **deltat**: numeric: sampling interval.
- **period**: numeric: Period to search for. The given period is mapped to the best matching number of measurements. A set of periods can be defined by period and `period.delta`.
- **period.delta**: numeric: width of period interval. A interval of different period-length to evaluate is defined by period $-$ `period.delta` and period $+$ `period.delta`. In this interval all possible numbers of time points according to the `deltat` are tested.
- **peak.border**: vector c(min, max): defines the different form of the peak. min and max have to be >0 and <1. The concrete interpretation depends on the chosen method (see Details). (default = c(0.3,0.7))
- **nr.series**: numeric: Number of replicates of the whole time series. If using `nr.series` all series have to have the same length. These multiple time series contained in `x` must be organized timepoint by timepoint using the following format [rit1, rit2, rit1, rit2, ..., ritn, ritm] where rit is the i'th repeat of the j'th time-point.
- **measure.sequence**: numeric array: Numbers of replicates for each time point. By using 'measure.sequence', irregular time courses may be evaluated. A value of 0 is possible and handled correctly. The array determines how many values are present for each time point. The values are ordered in the same format as specified above. `measure.sequence` overwrites `nr.series` if both are set.
- **method**: string ("independent", "longitudinal"): identify the method to use (see Details).
- **na.rm**: boolean: calculate individual statistics for time series containing NAs. The time series of a sample containing NAs is treated as if the time points with NA are not measured. Using this option increases calculation time.
- **adjp.method**: string (see `mt.rawp2adjp`): select the method which is used for the multiple testing correction for the different phases and periods and shapes tested
- **verbose**: status output
Details

The method tests whether the time course consists of alternating rising and falling slopes, repeated with a distinct period. The partitions of the rising part with respect to the whole period are given by `peak.border = c(min, max)`. The value `peak.shape` specifies this partition in the best matching model. The phase is defined as the time point with the peak. There are two versions of umbrella:

- **independent** Multiple periods are interpreted as repeats of one period.
- **longitudinal** The whole time series remains unaffected. Partial slopes in the beginning and end of the time series are evaluated as shorter slopes. This method implicitly rejects underlying trends. This should be used only with longitudinal samples, that may contain strong trends.

Value

An array containing p-values and the description of the best matching model. Each row apply to a sample in `x`.

- `pVal`: The p-Values
- `phase`: The phase of the peak (see Details)
- `peak.shape`: The shape of the curve depending on the method used (see Details)
- `period`: The period length, same unit as `deltat`

Author(s)

Paul F. Thaben

References


Examples

```r
# create a dataset with different noise levels
noise.levels <- c(1, 0.5, 0.2, 0.1, 0.05, 0.02)
period <- 15
testset <- apply(matrix(noise.levels, nrow = 1), 2, function(noise){
  timecourse = 1 + 0.4 * cos((1:30) / period * 2 * pi) + rnorm(30, 0, noise)
})
results <- rain(testset, period=15, deltat=1, method='independent')
plot(-log(results$pVal) ~ noise.levels)
```

```
## Not run:
# testing a biological dataset
data(menetRNASeqMouseLiver)
menet.ossc <- rain(t(menetRNASeqMouseLiver), deltat = 4, period = 24,
  nr.series = 2, peak.border = c(0.3, 0.7), verbose=TRUE)
require('lattice')
best <- order(results$pVal)[1:10]
```
xyplot(as.matrix(menetRNASeqMouseLiver[best, (0:5 * 2 + rep(c(1, 2), each = 6))]
 ~ rep(0:11 * 4 + 2, each = 10) | rownames(menetRNASeqMouseLiver)[best],
scales = list(y = list(relation = 'free')),
layout = c(2, 5), type = 'b', pch = 16, xlab = 'time',
ylab = 'expression value', cex.lab = 1)

## End(Not run)
Index

menetRNASeqMouseLiver, 2
mt.rawp2adjp, 3

rain, 2, 3
rain-package, 2