Package ‘zeitgebr’

April 25, 2020

**Title**  Analysis of Circadian Behaviours

**Date**  2020-04-22

**Version**  0.3.5


**Depends**  R (>= 3.00), behavr

**Imports**  data.table, lomb, pracma, WaveletComp

**Suggests**  testthat, covr, knitr, ggetho, damr, xsp

**License**  GPL-3

**Encoding**  UTF-8

**LazyData**  true

**URL**  https://github.com/rethomics/zeitgebr

**BugReports**  https://github.com/rethomics/zeitgebr/issues

**RoxygenNote**  6.1.1

**NeedsCompilation**  no

**Author**  Quentin Geissmann [aut, cre], Luis Garcia [aut]

**Maintainer**  Quentin Geissmann <qgeissmann@gmail.com>

**Repository**  CRAN

**Date/Publication**  2020-04-25 14:50:03 UTC

---

**R topics documented:**

- cwt_spectrogram ................................................................. 2
- dams_sample ........................................................................... 3


\section*{cwt_spectrogram}

\textit{Computes a spectrogram using CWT}

\subsection*{Description}
A port of Continuous Wavelet transform to rethomics. This function is intended to be used as an argument in the \texttt{spectrogram} wrapper.

\subsection*{Usage}
\begin{verbatim}
cwt_spectrogram(x, period_range = c(hours(1), hours(32)),
    sampling_rate = 1/mins(1), resolution = 1/64,
    summary_time_window = mins(30))
\end{verbatim}

\subsection*{Arguments}
\begin{itemize}
  \item \texttt{x} \quad numeric vector
  \item \texttt{period_range} \quad vector of size 2 defining minimal and maximal range of period to study (in seconds)
  \item \texttt{sampling_rate} \quad the \textendash{} implicitly regular \textendash{} sampling rate of \texttt{x} (in hertz)
  \item \texttt{resolution} \quad the period resolution of the CWT (i.e. the number of suboctaves)
  \item \texttt{summary_time_window} \quad the sampling period after post-processing. Values of power are averaged over this time window, for each period.
\end{itemize}

\subsection*{See Also}
\begin{itemize}
  \item \texttt{spectrogram} \quad to apply this function to all individual, with some preprocessing.
  \item \texttt{WaveletComp::analyze.wavelet} \quad the original function for \texttt{cwt_spectrogram}
\end{itemize}
dams_sample

A behavr table with approximately ten days of DAM2 recording for 32 fruit flies. The first 10, the following 11 and the last 11 animals have long, short and wild type period, respectively (see meta(dams_sample)).

Usage
dams_sample

Format
An object of class behavr (inherits from data.table, data.frame) with 415040 rows and 3 columns.

Author(s)
Luis Garcia

References
Raw data stored at https://github.com/rethomics/zeitgebr/tree/master/raw_data

find_peaks

Find peaks in a periodogram

Description
This function locates the peaks in a pregenerated periodogram. Detection is based on pracma::findpeaks. Only the significant (i.e. power > signif_threshold) peaks are extracted.

Usage
find_peaks(data, n_peaks = 3)

Arguments
data behavr::behavr table representing a periodogram, as returned by periodogram
n_peaks maximal numbers of peak to be detected
periodogram

Value

behavr::behavr table that is data with an extra column peak. peak is filled with zeros except for rows match a peak. In which case, rows have an integer value corresponding to the rank of the peak (e.g. 1 for the first peak).

References

- zeitgebr tutorial – the relevant rehtomics tutorial

See Also

- periodogram – to generate a periodogram in a first place
- ggeth0::geom_peak – a layer to show peaks on a periodogram

Examples

data(dams_sample)
# only four individuals for the sake of the example
dt <- dams_sample[xmv(region_id) %in% c(1, 7, 21, 31)]
per_dt_xs <- periodogram(activity, dt, FUN = chi_sq_periodogram)
per_dt_xs_with_peaks <- find_peaks(per_dt_xs)
per_dt_xs_with_peaks[peak == 1]

periodogram

Computes periodograms

Description

This function builds periodograms, with one of several methods, for each individual of a behavr table

Usage

periodogram(var, data, period_range = c(hours(16), hours(32)),
resample_rate = 1/mins(15), alpha = 0.01, FUN = chi_sq_periodogram,
...)

Arguments

<table>
<thead>
<tr>
<th>argument</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>variable to analyse</td>
</tr>
<tr>
<td>data</td>
<td>behavr table</td>
</tr>
<tr>
<td>period_range</td>
<td>vector of size 2 defining minimal and maximal range of period to study (in seconds)</td>
</tr>
<tr>
<td>resample_rate</td>
<td>frequency to resample (up or down) the data at (in hertz)</td>
</tr>
<tr>
<td>alpha</td>
<td>significance level</td>
</tr>
<tr>
<td>FUN</td>
<td>function used to compute periodogram (see periodogram_methods)</td>
</tr>
<tr>
<td>...</td>
<td>additional arguments to be passed to FUN</td>
</tr>
</tbody>
</table>
Value

A `behavr::behavr` table. In addition to the metadata, it contains data that encodes a periodogram (i.e. power vs period). The data contains the columns:

- `power` – the power the or equivalent (according to `FUN`)
- `period` – the period at which power is computed (in seconds)
- `p_value` – the p value associated to the power estimation
- `signif_threshold` – the threshold above which power is considered significant

References

- `zeitgebr tutorial` – the relevant rehtomics tutorial

See Also

- `periodogram_methods` – the list of built-in methods
- `find_peaks` – to find peaks in the periodogram
- `ggetho::ggperio` – to plot periodograms

Examples

```r
data(dams_sample)
# only four individuals for the sake of the example
dt <- dams_sample[xmv(region_id) %in% c(1, 7, 21, 31)]
pdt <- periodogram(activity, dt, FUN = ls_periodogram, oversampling = 4)
pdt <- periodogram(activity, dt, FUN = chi_sq_periodogram)
require(ggetho)
ggperio(pdt, aes(colour=period_group)) + stat_pop_etho()
```

Methods For Computing Periodograms

Description

These functions provides a series of methods to assess periodicity of circadian processes.

Usage

```r
ac_periodogram(x, period_range = c(hours(16), hours(32)),
               sampling_rate = 1/mins(1), alpha = 0.05)

chi_sq_periodogram(x, period_range = c(hours(16), hours(32)),
                   sampling_rate = 1/mins(1), alpha = 0.05,
                   time_resolution = hours(0.1))
```
cwt_periodogram(x, period_range = c(hours(16), hours(32)),
    sampling_rate = 1/mins(1), alpha = 0.05, resolution = 1/512,
    n_sim = 10)

fourier_periodogram(x, period_range = c(hours(16), hours(32)),
    sampling_rate = 1/mins(1), alpha = 0.05)

ls_periodogram(x, period_range = c(hours(16), hours(32)),
    sampling_rate = 1/mins(1), alpha = 0.05, oversampling = 8)

Arguments

x numeric vector
period_range vector of size 2 defining minimal and maximal range of period to study (in
    seconds)
sampling_rate the – implicitly regular – sampling rate of x (in hertz)
alpha significance level
time_resolution the resolution of periods to scan
resolution the period resolution of the CWT (i.e. the number of suboctaves)
n_sim the number of shuffling simulation to compute p-value (see WaveletComp::analyze.wavelet)
oversampling the oversampling factor (see lomb::lsp)

Value

a data.table with the columns:

- period – the period (in s)
- power – the power (or equivalent) for a given period
- p_value – the significance of the power
- signif_threshold – the significance threshold of the power (at alpha)

References

- zeitgebr tutorial – the relevant rehtomics tutorial

See Also

- lomb::lsp – the orginal function for ls_periodogram
- xsp::chiSqPeriodogram – code modified from
- stats::acf – the orginal function for ac_periodogram
- WaveletComp::analyze.wavelet – the orginal function for cwt_periodogram
**Description**

This function builds spectrogram, using CWT, for each individual of a `behavr` table.

**Usage**

```r
spectrogram(var, data, period_range = c(hours(16), hours(32)),
resample_rate = 1/mins(15), FUN = cwt_spectrogram, ...)
```

**Arguments**

- `var` variable to analyse
- `data` `behavr` table
- `period_range` vector of size 2 defining minimal and maximal range of period to study (in seconds)
- `resample_rate` frequency to resample (up or down) the data at (in hertz)
- `FUN` function used to compute spectrograms (so far, only CWT is implemented via `cwt_spectrogram`)
- `...` additional arguments to be passed to `FUN`

**Details**

A spectrogram is a estimation of the local periodicity of a signal at a given time. In the context of circadian rhythm, it can be useful to understand how infradian rhythms change along the day or, for instance, how circadian rhythm change ver the course of an multi-day experiment.

**Value**

A `behavr::behavr` table. In addition to the metadata, it contains data that encodes a spectrogram (i.e. power vs period). The data contains the columns:

- `t` – the time (in s) (same range the input time)
- `period` – the period at which the `power` is computed, for a given `t` (in s)
- `power` – the power the or equivalent (according to `FUN`
- `ridge` – a logical defining whether the point (`t` and `period`) is a ridge

**References**

- spectrogram tutorial – the relevant rehtomics tutorial
See Also

- `periodogram` – to compute periodogram instead
- `cwt_spectrogram` – The function use to compute individual spectrograms
- `ggetho::ggspectro` – to plot spectrograms

Examples

```r
data(dams_sample)
dt <- dams_sample[id %in% dams_sample[meta=TRUE, ,id[1:5]]]
spect_dt <- spectrogram(activity, dt)

require(ggetho)
ggspectro(spect_dt) +
  stat_tile_etho() +
  scale_y_log10() +
  facet_wrap(~ id)
```
Index

*Topic **datasets**
  dams_sample, 3

ac_periodogram (periodogram_methods), 5

behavr, 4–7
behavr::behavr, 3–5, 7

chi_sq_periodogram
  (periodogram_methods), 5

ls_periodogram (periodogram_methods), 5

ls_periodogram (periodogram_methods), 5

pracma::findpeaks, 3

spectrogram, 2, 7
stats::acf, 6

WaveletComp::analyze.wavelet, 2, 6

xsp::chiSqPeriodogram, 6