

Package ‘spectr’

May 9, 2026

Type Package

Title Calculate the Periodogram of a Time-Course

Version 1.0.1

Description Provides a consistent interface to use various methods to calculate the periodogram and estimate the period of a rhythmic time-course. Methods include Lomb-Scargle, fast Fourier transform, and three versions of the chi-square periodogram. See Tackenberg and Hughey (2021) <[doi:10.1371/journal.pcbi.1008567](https://doi.org/10.1371/journal.pcbi.1008567)>.

URL <https://spectr.hugheylib.org>, <https://github.com/hugheylib/spectr>

License GPL-2

Encoding UTF-8

RoxygenNote 7.1.1

Depends R (>= 3.6)

Imports data.table (>= 1.12.2), foreach (>= 1.5.0), lomb (>= 2.0)

Suggests doParallel (>= 1.0.15), knitr, qs (>= 0.24.1), rmarkdown, testthat (>= 3.0.3)

Config/testthat/edition 3

NeedsCompilation no

Author Jake Hughey [aut, cre],
Michael Tackenberg [aut],
Josh Schoenbachler [ctb]

Maintainer Jake Hughey <jakejhughey@gmail.com>

Repository CRAN

Date/Publication 2022-02-07 12:00:05 UTC

Contents

cspgram	2
Index	4

cspgram

Calculate periodogram

Description

Calculate periodogram for a time-course using Lomb-Scargle, fast Fourier transform, or selected version of chi-square. The `spectr` function is a wrapper for the various methods. `lspgram` is in turn a wrapper for `lomb::lsp()`, and `fftpgram` a wrapper for `stats::spec.pgram()`. Among the versions of chi-square, it is highly recommended to use `greedy`, which has lower bias than `standard` and lower variance than `conservative`.

Usage

```
cspgram(
  x,
  deltat,
  periodRange = c(18, 32),
  method = c("greedy", "conservative", "standard"),
  na.action = stats::na.fail,
  dopar = FALSE
)
```

```
fftpgram(
  x,
  deltat,
  periodRange = c(18, 32),
  pad = 50,
  na.action = stats::na.fail,
  ...
)
```

```
lspgram(x, deltat, time, periodRange = c(18, 32), ofac = 50)
```

```
spectr(
  x,
  deltat,
  time,
  periodRange = c(18, 32),
  method = c("greedy_chisq", "conservative_chisq", "standard_chisq", "lombscargle",
    "fft"),
  ofac = 50,
  pad = 50,
  na.action = stats::na.fail,
  dopar = FALSE,
  ...
)
```

Arguments

<code>x</code>	Numeric vector of measurements.
<code>deltat</code>	Numeric value of the interval between time-points.
<code>periodRange</code>	Numeric vector of the minimum and maximum values of the period to consider, in the same units as <code>deltat</code> or <code>time</code> .
<code>method</code>	Character indicating which method to use. Can be an unambiguous substring of the full name.
<code>na.action</code>	Function specifying how to handle NA values in <code>x</code> . Default is <code>stats::na.fail()</code> , which gives an error if any values are missing. Ignored for Lomb-Scargle.
<code>dopar</code>	Logical indicating whether to run calculations in parallel if a parallel backend is already set up, e.g., using <code>doParallel::registerDoParallel()</code> . Only used for chi-square.
<code>pad</code>	Numeric value of the proportion of the length of <code>x</code> by which to pad <code>x</code> with zeros. Must be > 0 . Only used for FFT.
<code>...</code>	Other arguments passed to <code>stats::spec.pgram()</code> for FFT.
<code>time</code>	Numeric vector of time-points. Can be specified instead of <code>deltat</code> for Lomb-Scargle.
<code>ofac</code>	Integer value of the oversampling factor. Must be ≥ 1 . Only used for Lomb-Scargle.

Value

A `data.table` with various columns depending on the method. For any version of chi-square, columns will be `period`, `chisq`, `df`, and `log_pval`. The log p-value is more reliable than the p-value, since R has finite precision, so p-values less than about $5e-324$ would be set to 0. For Lomb-Scargle and FFT, columns will be `period` and `power`.

Examples

```
library('data.table')

set.seed(1789)
deltat = 0.1
tau = 25
tt = seq(0, 24 * 3, deltat)
x = 3 * sin(tt / tau * 2 * pi) + rnorm(length(tt))

specCsp = spectr(x, deltat, method = 'greedy')
peakCsp = specCsp[which.min(log_pval)]

specLsp = spectr(x, deltat, method = 'lomb')
peakLsp = specLsp[which.max(power)]

specFft = spectr(x, deltat, method = 'fft')
peakFft = specFft[which.max(power)]
```

Index

`cspgram`, 2

`doParallel::registerDoParallel()`, 3

`fftogram(cspgram)`, 2

`lomb::lsp()`, 2

`lspogram(cspgram)`, 2

`spectr(cspgram)`, 2

`stats::na.fail()`, 3

`stats::spec.pgram()`, 2, 3