Package: CWT (via r-universe)

October 26, 2026

Type Package

Title Continuous Wavelet Transformation for Spectroscopy

Version 0.2.3

Maintainer J. Antonio Guzmán Q. <antguz06@gmail.com>

Description Fast application of Continuous Wavelet Transformation ('CWT') on time series with special attention to spectroscopy. It is written using data.table and 'C++' language and in some functions it is possible to use parallel processing to speed-up the computation over samples. Currently, only the second derivative of a Gaussian wavelet function is implemented.

License GPL (>= 3)

URL https://github.com/Antguz/CWT

BugReports https://github.com/Antguz/CWT/issues

Depends R (>= 4.0.0)

Imports data.table (>= 1.14.0), Rcpp

Suggests testthat (>= 3.2.0)

LinkingTo Rcpp, RcppArmadillo

ByteCompile true

Config/testthat/edition 3

Encoding UTF-8

Language en-US

SystemRequirements GNU make

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author J. Antonio Guzmán Q. [cre, aut, cph] (ORCID: <https://orcid.org/0000-0002-0721-148X>)

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2026-07-27 23:20:02 UTC

RemoteUrl https://github.com/cran/CWT

RemoteRef HEAD

RemoteSha e1a670fbcec685ef5f1f67e6716386af3c943e5e

Contents

CWT-package 2
cwt 3
resampling_FWHM 4

Index **6**

`CWT-package` *Continuous Wavelet Transformation for Spectroscopy*

Description

Fast application of Continuous Wavelet Transformation on time series with special attention to spectroscopy. It is written using 'data.table' and 'C++' language and in some functions it is possible to use parallel processing to speed-up the computation over samples.

Author(s)

Maintainer: J. Antonio Guzmán Q. <antguz06@gmail.com> (ORCID) [copyright holder]

Authors:

- J. Antonio Guzmán Q. <antguz06@gmail.com> (ORCID) [copyright holder]

See Also

Useful links:

- https://github.com/Antguz/CWT
- Report bugs at https://github.com/Antguz/CWT/issues

cwt *Continuous Wavelet Transform*

Description

Compute a 1D continuous wavelet transformation using 2st order derivative Gaussian wavelet.

Usage

```
cwt(t, scales, variance = 1, summed_wavelet = FALSE, threads = 1L)
```

Arguments

t	A `data.table`, `matrix`, or `numeric` vector where columns or values represent time (i.e., bands) and rows samples (i.e., pixels). Remember the transformation assume that columns or values are evenly spaced though time (i.e., bands at equal to sampling interval).
scales	A positive `numeric` vector describing the scales to compute. The minimum scale (i.e., scales = 1) is equal to sampling interval between columns.
variance	A positive `numerber` describing the variance of the Gaussian PDF used to scale. Default `variance = 1`.
summed_wavelet	If `TRUE`, it returns the sum of scales. If `FALSE`, each scale is returned.
threads	An `integer` specifying the number of threads to use. Experiment to see what works best for your data on your hardware.

Value

If `summed_wavelet = TRUE`, it returns a `data.table` where columns are the sum of wavelet scales. If `summed_wavelet = FALSE`, it returns an `array` (i.e., time, samples, and scales).

Author(s)

J. Antonio Guzmán Q.

Examples

```
time_series <- sin(seq(0, 20 * pi, length.out = 100))

# Using a numeric vector

cwt(t = time_series,
    scales = c(1, 2, 3, 4, 5),
    summed_wavelet = FALSE)

cwt(t = time_series,
    scales = c(1, 2, 3, 4, 5),
    summed_wavelet = TRUE)
```

```
# Using a matrix

times <- 100
frame <- matrix(rep(time_series, times),
                nrow = times,
                byrow = TRUE)

cwt(t = frame,
    scales = c(1, 2, 3, 4, 5),
    summed_wavelet = FALSE)

cwt(t = frame,
    scales = c(1, 2, 3, 4, 5),
    summed_wavelet = TRUE)
```

`resampling_FWHM` *Full Width Half Maximum Resampling*

Description

It resample spectra data using Full Width Half Maximum (FWHM).

Usage

```
resampling_FWHM(spectra, wavelengths, new_wavelengths, FWHM, threads = 1L)
```

Arguments

`spectra`	A `data.table`, `data.frame`, or `matrix` where columns represent bands and rows samples (i.e., pixels).
`wavelengths`	A `numeric` vector describing the current positioning of the spectral bands within `spectra`.
`new_wavelengths`	A `numeric` vector describing positioning of the new spectral bands to resample.
`FWHM`	A `numeric` vector describing the Full Width Half Maximums of the new spectral bands. The length of this vector should be equal than the length of `new_wavelengths`.
`threads`	An `integer` specifying the number of threads to use. Experiment to see what works best for your data on your hardware.

Value

It returns a `data.table` with the resampled spectra, where columns are the new bands and rows are samples. New bands whose FWHM window falls partially outside the range of `wavelengths` are returned as `NA`.

Author(s)

J. Antonio Guzmán Q.

Examples

```
mean <- 50
sd1 <- 5
sd2 <- 10
n <- 100

test <- matrix(c(dnorm(1:n, mean = mean, sd = sd1),
                 dnorm(1:n, mean = mean, sd = sd2)),
                 nrow = 2,
                 byrow = TRUE)

current_bands <- 1:n
new_bands <- seq(10, 90, by = 5)
FHWM <- rep(5, length(new_bands))

resampling_FWHM(spectra = test,
                 wavelengths = current_bands,
                 new_wavelengths = new_bands,
                 FWHM = FHWM)
```

Index

CWT *(*CWT-package*)*, 2
cwt, 3
CWT-package, 2

resampling_FWHM, 4