Package 'wavScalogram'

March 6, 2019

Type Package

Title Wavelet Scalogram Tools for Time Series Analysis

Version 0.1.0

Author Vicente J. Bolos and Rafael Benitez

Maintainer Vicente J. Bolos <vicente.bolos@uv.es>

Description Provides scalogram based wavelet tools for time series analysis: wavelet power spectrum, scalogram, windowed scalogram, windowed scalogram difference (see Bolos et al. (2017) <doi:10.1016/j.amc.2017.05.046>), scale index and windowed scale index (Benitez et al. (2010) <doi:10.1016/j.camwa.2010.05.010>).

Imports abind, colorRamps, fields, Matrix, parallel, zoo

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-06 15:30:03 UTC

R topics documented:

core *Extracts the center of a vector*

Description

This function is an internal function which extracts from a vector x, the center of the vector of length n. It emulates the Matlab(R) function wkeep. This function is used by the cwt_wst function when the HAAR wavelet is selected.

Usage

```
core(x,n)
```

Arguments

x	A vector from wich the center is extracted.
n	Numeric. The length of the center of x.

cwt_wst *Continuous wavelet transform*

Description

This function computes the continuous wavelet transform for some families of wavelet bases: "MORLET", "DOG", "PAUL" and "HAAR". It is a translation from the Matlab(R) function published by Torrence and Compo (Torrence & Compo, 1998).

The difference between cwt_wst and cwt from package Rwave is that cwt_wst normalizes using L^2 and cwt uses L^1.

Usage

```
cwt_wst(signal,
                dt = 1,
                scales = NULL,
                powerscales = TRUE,
                wname = c("MORLET", "DOG", "PAUL", "HAAR", "HAAR2"),
                wparam = NULL,
                waverad = NULL,
                border_effects = c("BE", "PER", "SYM"),
                makefigure = TRUE,
                time_values = NULL,
                energy_density = FALSE,
                figureperiod = TRUE)
```

Arguments

signal	A vector containing the signal whose wavelet transform is wanted.
dt	Numeric. The time step of the signal.
scales	A vector containing the wavelet scales at wich the CWT is computed. This can be either a vector with all the scales, or (if powerscales is TRUE) following Torrence and Compo 1998, a vector of three elements with the minimum scale, the maximum scale and the number of suboctaves per octave. If NULL, they are automatically computed.
powerscales	Logical. If TRUE (default), construct power 2 scales from scales. If scales is NULL, they are automatically computed.
wname	A string, equal to "MORLET", "DOG", "PAUL", "HAAR" or "HAAR2". The difference between "HAAR" and "HAAR2" is that "HAAR2" is more accurate but slower.
wparam	The corresponding nondimensional parameter for the wavelet function (Morlet, DoG or Paul).
waverad	Numeric. The radius of the wavelet used in the computations for the cone of influence. If it is not specified, it is computed by wavelet_radius for DoG and Paul wavelets. For Haar and Morlet it is assumed to be 1 and 3 respectively.
border_effects	String, equal to "BE", "PER" or "SYM", which indicates how to manage the border effects which arise usually when a convolution is performed on finite-lenght signals. • "BE": Padding time series with zeroes. • "PER": Using boundary wavelets (periodization of the original time series). • "SYM": Using a symmetric catenation of the original time series.
makefigure	Logical. If TRUE (default), a figure with the wavelet power spectrum is plotted.
time_values	A numerical vector of length length(signal) containing custom time values for the figure. If NULL (default), it will be computed starting at 0.
energy_density	Logical. If TRUE (default), divide the wavelet power spectrum by the scales in the figure and so, values for different scales are comparable.
figureperiod	Logical. If TRUE (default), periods are used in the figure instead of scales.

Value

A list with the following fields:

- coefs: A matrix of size length(signal) x length(scales), containing the CWT coefficients of the signal.
- scales: The vector of scales.
- fourier_factor: A factor for converting scales to periods.
- coi_maxscale: A vector of length length(signal) containing the values of the maximum scale from which there are border effects at each time.

References

C. Torrence, G. P. Compo. A practical guide to wavelet analysis. B. Am. Meteorol. Soc. 79 (1998), 61–78.

Examples

```
dt <- 0.1
time <- seq(0, 50, dt)
signal <- c(sin(pi * time), sin(pi * time / 2))
cwt <- cwt_wst(signal = signal, dt = dt, energy_density = TRUE)
```

`pow2scales` *Power 2 scales*

Description

This function constructs power 2 scales from a vector of three elements with the minimum scale, the maximum scale and the number of suboctaves per octave (following Torrence and Compo 1998).

Usage

```
pow2scales(scales)
```

Arguments

`scales`	A vector of three elements with the minimum scale, the maximum scale and the number ofsuboctaves per octave.

Value

A vector with all the scales.

References

C. Torrence, G. P. Compo. A practical guide to wavelet analysis. B. Am. Meteorol. Soc. 79 (1998), 61–78.

Examples

```
scales <- pow2scales(c(2,128,8))
```

scale_index *Scale index of a signal*

Description

This function computes the scale index of a signal in the scale interval $[s_0, s_1]$, for a given set of scale parameters s_1 and taking s_0 as the minimum scale (see Benítez et al. 2010).

The scale index of a signal in the scale interval $[s_0, s_1]$ is given by the quotient

$$\frac{S(s_{min})}{S(s_{max})},$$

where S is the scalogram, $s_{max} in [s_0, s_1]$ is the smallest scale such that $S(s) \leq S(s_{max})$ for all $sin[s_0, s_1]$, and $s_{min} in [s_{max}, 2s_1]$ is the smallest scale such that $S(s_{min}) \leq S(s)$ for all $sin[s_{max}, 2s_1]$.

Usage

```
scale_index(signal,
                  dt = 1,
                  scales = NULL,
                  powerscales = TRUE,
                  s1 = NULL,
                  wname = c("MORLET", "DOG", "PAUL", "HAAR", "HAAR2"),
                  wparam = NULL,
                  waverad = NULL,
                  border_effects = c("BE", "INNER", "PER", "SYM"),
                  makefigure = TRUE,
                  figureperiod = TRUE)
```

Arguments

signal	A vector containing the signal whose scale indices are wanted.
dt	Numeric. The time step of the signals.
scales	A vector containing the wavelet scales at wich the scalogram is computed. This can be either a vector with all the scales, or (if powerscales is TRUE) following Torrence and Compo 1998, a vector of three elements with the minimum scale, the maximum scale and the number of suboctaves per octave. If NULL, they are automatically computed.
powerscales	Logical. If TRUE (default), construct power 2 scales from scales. If scales is NULL, they are automatically computed.
s1	A vector containing the scales s_1. The scale indices are computed in the intervals $[s_0, s_1]$, where s_0 is the minimum scale in scales.
wname	A string, equal to "MORLET", "DOG", "PAUL", "HAAR" or "HAAR2". The difference between "HAAR" and "HAAR2" is that "HAAR2" is more accurate but slower.

wparam	The corresponding nondimensional parameter for the wavelet function (Morlet, DoG or Paul).
waverad	Numeric. The radius of the wavelet used in the computations for the cone of influence. If it is not specified, it is computed by wavelet_radius for DoG and Paul wavelets. For Haar and Morlet it is assumed to be 1 and 3 respectively.
border_effects	A string, equal to "BE", "INNER", "PER" or "SYM", which indicates how to manage the border effects which arise usually when a convolution is performed on finite-lenght signals. • "BE": With border effects, padding time series with zeroes. • "INNER": Normalized inner scalogram with security margin adapted for each different scale. • "PER": With border effects, using boundary wavelets (periodization of the original time series). • "SYM": With border effects, using a symmetric catenation of the original time series.
makefigure	Logical. If TRUE (default), a figure with the scale indices is plotted.
figureperiod	Logical. If TRUE (default), periods are used in the figure instead of scales.

Value

A list with the following fields:

- si: A vector with the scale indices.
- s1: A vector containing the scales s_1.
- smax: A vector with the scales s_{max}.
- smin: A vector with the scales s_{min}.
- scalog_smax: A vector with the maximum scalogram values $S(s_{max})$.
- scalog_smin: A vector with the minimum scalogram values $S(s_{min})$.
- fourier_factor: A factor for converting scales to periods.

References

R. Benítez, V. J. Bolós, M. E. Ramírez. A wavelet-based tool for studying non-periodicity. Comput. Math. Appl. 60 (2010), no. 3, 634-641.

Examples

```
dt <- 0.1
time <- seq(0, 50, dt)
signal <- c(sin(pi * time), sin(pi * time / 2))
si <- scale_index(signal = signal, dt = dt)
```

scalogram *Scalogram of a signal*

Description

This function computes the normalized scalogram of a signal for the scales given. It is important to note that the notion of scalogram here is analogous to the spectrum of the Fourier transform. It gives the contribution of each scale to the total energy of the signal. For each scale s, it is defined as the square root of the integral of the squared modulus of the wavelet transform w.r.t. the time variable t, i.e.

$$S(s) := (\int_{-\infty}^{+\infty} |Wf(t,s)|^2 dt)^{1/2}.$$

"Normalized" means that the scalogram is divided by the square root of the number of times, for comparison purposes between different values of the parameter `border_effects`.

Usage

```
scalogram(signal,
                  dt = 1,
                  scales = NULL,
                  powerscales = TRUE,
                  wname = c("MORLET", "DOG", "PAUL", "HAAR", "HAAR2"),
                  wparam = NULL,
                  waverad = NULL,
                  border_effects = c("BE", "INNER", "PER", "SYM"),
                  energy_density = TRUE,
                  makefigure = TRUE,
                  figureperiod = TRUE)
```

Arguments

`signal`	A vector containing the signal whose scalogram is wanted.
`dt`	Numeric. The time step of the signal.
`scales`	A vector containing the wavelet scales at wich the scalogram is computed. This can be either a vector with all the scales, or (if `powerscales` is TRUE) following Torrence and Compo 1998, a vector of three elements with the minimum scale, the maximum scale and the number of suboctaves per octave. If NULL, they are automatically computed.
`powerscales`	Logical. If TRUE (default), construct power 2 scales from `scales`. If `scales` is NULL, they are automatically computed.
`wname`	A string, equal to "MORLET", "DOG", "PAUL", "HAAR" or "HAAR2". The difference between "HAAR" and "HAAR2" is that "HAAR2" is more accurate but slower.

wparam	The corresponding nondimensional parameter for the wavelet function (Morlet, DoG or Paul).
waverad	Numeric. The radius of the wavelet used in the computations for the cone of influence. If it is not specified, it is computed by wavelet_radius for DoG and Paul wavelets. For Haar and Morlet it is assumed to be 1 and 3 respectively.
border_effects	String, equal to "BE", "INNER", "PER" or "SYM", which indicates how to manage the border effects which arise usually when a convolution is performed on finite-lenght signals. • "BE": With border effects, padding time series with zeroes. • "INNER": Normalized inner scalogram with security margin adapted for each different scale. • "PER": With border effects, using boundary wavelets (periodization of the original time series). • "SYM": With border effects, using a symmetric catenation of the original time series.
energy_density	Logical. If TRUE (default), divide the scalogram by the square root of the scales for convert it into energy density.
makefigure	Logical. If TRUE (default), a figure with the scalogram is plotted.
figureperiod	Logical. If TRUE (default), periods are used in the figure instead of scales.

Value

A list with the following fields:

- scalog: A vector of length length(scales), containing the values of the scalogram at each scale.
- scales: The vector of scales.
- fourier_factor: A factor for converting scales to periods.

References

C. Torrence, G. P. Compo. A practical guide to wavelet analysis. B. Am. Meteorol. Soc. 79 (1998), 61–78.

V. J. Bolós, R. Benítez, R. Ferrer, R. Jammazi. The windowed scalogram difference: a novel wavelet tool for comparing time series. Appl. Math. Comput., 312 (2017), 49-65.

Examples

```
dt <- 0.1
time <- seq(0, 50, dt)
signal <- c(sin(pi * time), sin(pi * time / 2))
scalog <- scalogram(signal = signal, dt = dt, border_effects = "INNER")
```

wavelet_radius *Wavelet radius*

Description

This function computes an approximation of the effective radius of a mother wavelet.

Usage

```
wavelet_radius(wname = c("MORLET", "DOG", "PAUL", "HAAR", "HAAR2"),
                       wparam = NULL,
                       perc = .0025,
                       scale = 100,
                       n = 1000,
                       makefigure = FALSE)
```

Arguments

wname	A string, equal to "MORLET", "DOG", "PAUL", "HAAR" or "HAAR2". The difference between "HAAR" and "HAAR2" is that "HAAR2" is more accurate but slower.
wparam	The corresponding nondimensional parameter for the wavelet function (Morlet, DoG or Paul).
perc	Numeric. The wavelet radius is computed so that the area covered is at least the 100*(1-perc)% of the total area of the mother wavelet.
scale	Numeric. Scale of the wavelet used in the computations. It only affects the accuracy.
n	Numeric. The computations use a time series of length $2n + 1$.
makefigure	Logical. Plots a figure with the real part of the mother wavelet and its modulus.

Value

A list with the following fields:

- left: The radius on the left.
- right: The radius on the right.

Examples

```
waverad <- wavelet_radius(wname = "MORLET", makefigure = TRUE)
```

wavPlot *Wavelet plots*

Description

This function plots a function of two variables (usually times and scales). It is suitable for plotting windowed scalograms, windowed scalogram differences, wavelet coherences and windowed scale indices.

Usage

```
wavPlot(Z,
                X = NULL,
                Y = NULL,
                Ylog = FALSE,
                Yrev = TRUE,
                coi = NULL,
                rdist = NULL,
                sig95 = NULL,
                sig05 = NULL,
                Xname = "X",
                Yname = "Y",
                Zname = "Z")
```

Arguments

Z	A matrix with the images of the function to be plotted.
X	A vector with x-coordinates (times).
Y	A vector with y-coordinates (scales).
Ylog	Logical. Considers logarithmic scale for the y-axis.
Yrev	Logical. Considers reverse the y-axis.
coi	A vector of size `length(X)` with the y-coordinates of the frontier of the cone of influence.
rdist	Numeric. Only for WSD plots, margin in the y-axis where appear border effects.
sig95	Logical matrix with the same size as Z. TRUE if the corresponding point in Z is inside the significance at 95%.
sig05	Logical matrix with the same size as Z. TRUE if the corresponding point in Z is inside the significance at 5%.
Xname	A string with the name of the x-axis.
Yname	A string with the name of the y-axis.
Zname	A string with the name of the function.

Examples

```
nt <- 1500
time <- 1:nt
sd_noise <-  0.2 #% In Bolós et al. 2017 Figure 1, sd_noise = 1.
signal1 <- rnorm(n = nt, mean = 0, sd = sd_noise) + sin(time / 10)
signal2 <- rnorm(n = nt, mean = 0, sd = sd_noise) + sin(time / 10)
signal2[500:1000] = signal2[500:1000] + sin((500:1000) / 2)
## Not run:
wsd <- wsd(signal1 = signal1, signal2 = signal2, mc_nrand = 10, makefigure = FALSE)
wavPlot(Z = -log2(wsd$wsd), X = wsd$t, Y = wsd$scales, Ylog = TRUE, coi = wsd$coi,
        rdist = wsd$rdist, sig95 = wsd$signif95, sig05 = wsd$signif05, Xname = "Time",
        Yname = "Scale", Zname = "-log2(WSD)")

## End(Not run)
```

`windowed_scale_index` *Windowed scale index*

Description

This function computes the windowed scale indices of a signal in the scale interval $[s_0, s_1]$, for a given set of scale parameters s_1 and taking s_0 as the minimum scale (see Benítez et al. 2010).

The windowed scale index of a signal in the scale interval $[s_0, s_1]$ centered at time tc and with time windows radius `windowrad` is given by the quotient

$$\frac{WS_{windowrad}(tc, s_{min})}{WS_{windowrad}(tc, s_{max})},$$

where $WS_{windowrad}$ is the corresponding windowed scalogram with time windows radius `windowrad`, $s_{max} in [s_0, s_1]$ is the smallest scale such that $WS_{windowrad}(tc, s) \leq WS_{windowrad}(tc, s_{max})$ for all $sin[s_0, s_1]$, and $s_{min} in [s_{max}, 2s_1]$ is the smallest scale such that $WS_{windowrad}(tc, s_{min}) \leq WS_{windowrad}(tc, s)$ for all $sin[s_{max}, 2s_1]$.

Usage

```
windowed_scale_index(signal,
                             dt = 1,
                             scales = NULL,
                             powerscales = TRUE,
                             s1 = NULL,
                             windowrad = NULL,
                             delta_t = NULL,
                            wname = c("MORLET", "DOG", "PAUL", "HAAR", "HAAR2"),
                             wparam = NULL,
                             waverad = NULL,
                             border_effects = c("BE", "INNER", "PER", "SYM"),
```

```
                    makefigure = TRUE,
                    time_values = NULL,
                    figureperiod = TRUE)
```

Arguments

signal	A vector containing the signal whose windowed scale indices are wanted.
dt	Numeric. The time step of the signal.
scales	A vector containing the wavelet scales at wich the windowed scalograms are computed. This can be either a vector with all the scales, or (if powerscales is TRUE) following Torrence and Compo 1998, a vector of three elements with the minimum scale, the maximum scale and the number of suboctaves per octave. If NULL, they are automatically computed.
powerscales	Logical. If TRUE (default), construct power 2 scales from scales. If scales is NULL, they are automatically computed.
s1	A vector containing the scales s_1. The windowed scale indices are computed in the intervals $[s_0, s_1]$, where s_0 is the minimum scale in scales.
windowrad	Integer. Time radius for the windows, measured in dt. By default, it is set to $celing(length(signal1)/20)$.
delta_t	Integer. Increment of time for the construction of windows central times, measured in dt. By default, it is set to $celing(length(signal)/256)$.
wname	A string, equal to "MORLET", "DOG", "PAUL", "HAAR" or "HAAR2". The difference between "HAAR" and "HAAR2" is that "HAAR2" is more accurate but slower.
wparam	The corresponding nondimensional parameter for the wavelet function (Morlet, DoG or Paul).
waverad	Numeric. The radius of the wavelet used in the computations for the cone of influence. If it is not specified, it is computed by wavelet_radius for DoG and Paul wavelets. For Haar and Morlet it is assumed to be 1 and 3 respectively.
border_effects	A string, equal to "BE", "INNER", "PER" or "SYM", which indicates how to manage the border effects which arise usually when a convolution is performed on finite-lenght signals. • "BE": With border effects, padding time series with zeroes. • "INNER": Normalized inner scalogram with security margin adapted for each different scale. • "PER": With border effects, using boundary wavelets (periodization of the original time series). • "SYM": With border effects, using a symmetric catenation of the original time series.
makefigure	Logical. If TRUE (default), a figure with the windowed scale indices is plotted.
time_values	A numerical vector of length length(signal) containing custom time values for the figure. If NULL (default), it will be computed starting at 0.
figureperiod	Logical. If TRUE (default), periods are used in the figure instead of scales.

Value

A list with the following fields:

- `wsi`: A matrix of size `length(tcentral)` x `length(s1)` containing the values of the corresponding windowed scale indices.
- `s1`: The vector of scales s_1.
- `smax`: A matrix of size `length(tcentral)` x `length(s1)` containing the scales s_{max}.
- `smin`: A matrix of size `length(tcentral)` x `length(s1)` containing the scales s_{min}.
- `scalog_smax`: A matrix of size `length(tcentral)` x `length(s1)` containing the values of the corresponding scalograms at scales s_{max}.
- `scalog_smin`: A matrix of size `length(tcentral)` x `length(s1)` containing the values of the corresponding scalograms at scales s_{min}.
- `tcentral`: The vector of central times used in the computation of `wsi`.
- `fourier_factor`: A factor for converting scales to periods.
- `coi_maxscale`: A vector of length `length(tcentral)` containing the values of the maximum scale from which there are border effects.

References

R. Benítez, V. J. Bolós, M. E. Ramírez. A wavelet-based tool for studying non-periodicity. Comput. Math. Appl. 60 (2010), no. 3, 634-641.

Examples

```
dt <- 0.1
time <- seq(0, 50, dt)
signal <- c(sin(pi * time), sin(pi * time / 2))
# First, we try with default s1 scales (a vector with a wide range of values for s1).
wsi_full <- windowed_scale_index(signal = signal, dt = dt, figureperiod = FALSE)
# Next, we choose a meaningful s1 value, greater than all relevant scales.
wsi <- windowed_scale_index(signal = signal, dt = dt, s1 = 4, figureperiod = FALSE)
```

`windowed_scalogram` *Windowed scalograms of a signal*

Description

This function computes the normalized windowed scalograms of a signal for the scales given. It is computed using time windows with radius `windowrad` centered at a vector of central times with increment of time `delta_t`. It is important to note that the notion of scalogram here is analogous to the spectrum of the Fourier transform. It gives the contribution of each scale to the total energy of the signal. For each scale s and central time tc, it is defined as the square root of the integral of the squared modulus of the wavelet transform w.r.t the time variable t, i.e.

$$WS_{windowrad}(tc, s) := (\int_{tc-windowrad}^{tc+windowrad} |Wf(t, s)|^2 dt)^{1/2}.$$

"Normalized" means that the windowed scalograms are divided by the square root of the length of the respective time windows in order to be comparable between them.

Usage

```
windowed_scalogram(signal,
                           dt = 1,
                           scales = NULL,
                           powerscales = TRUE,
                           windowrad = NULL,
                           delta_t = NULL,
                           wname = c("MORLET", "DOG", "PAUL", "HAAR", "HAAR2"),
                           wparam = NULL,
                           waverad = NULL,
                           border_effects = c("BE", "INNER", "PER", "SYM"),
                           energy_density = TRUE,
                           makefigure = TRUE,
                           time_values = NULL,
                           figureperiod = TRUE)
```

Arguments

signal	A vector containing the signal whose windowed scalogram is wanted.
dt	Numeric. The time step of the signal.
scales	A vector containing the wavelet scales at wich the windowed scalogram is computed. This can be either a vector with all the scales, or (if powerscales is TRUE) following Torrence and Compo 1998, a vector of three elements with the minimum scale, the maximum scale and the number of suboctaves per octave. If NULL, they are automatically computed.
powerscales	Logical. If TRUE (default), construct power 2 scales from scales. If scales is NULL, they are automatically computed.
windowrad	Integer. Time radius for the windows, measured in dt. By default, it is set to $celing(length(signal1)/20)$.
delta_t	Integer. Increment of time for the construction of windows central times, measured in dt. By default, it is set to $celing(length(signal)/256)$.
wname	A string, equal to "MORLET", "DOG", "PAUL", "HAAR" or "HAAR2". The difference between "HAAR" and "HAAR2" is that "HAAR2" is more accurate but slower.
wparam	The corresponding nondimensional parameter for the wavelet function (Morlet, DoG or Paul).
waverad	Numeric. The radius of the wavelet used in the computations for the cone of influence. If it is not specified, it is computed by wavelet_radius for DoG and Paul wavelets. For Haar and Morlet it is assumed to be 1 and 3 respectively.

`border_effects` String, equal to "BE", "INNER", "PER" or "SYM", which indicates how to manage the border effects which arise usually when a convolution is performed on finite-lenght signals.

- "BE": With border effects, padding time series with zeroes.
- "INNER": Normalized inner scalogram with security margin adapted for each different scale. Although there are no border effects, it is shown as a regular COI the zone in which the length of $J(s)$ (see Benítez et al. 2010) is smaller and it has to be normalized.
- "PER": With border effects, using boundary wavelets (periodization of the original time series).
- "SYM": With border effects, using a symmetric catenation of the original time series.

`energy_density` Logical. If TRUE (default), divide the scalograms by the square root of the scales for convert them into energy density.

`makefigure` Logical. If TRUE (default), a figure with the scalograms is plotted.

`time_values` A numerical vector of length `length(signal)` containing custom time values for the figure. If NULL (default), it will be computed starting at 0.

`figureperiod` Logical. If TRUE (default), periods are used in the figure instead of scales.

Value

A list with the following fields:

- `wsc`: A matrix of size `length(tcentral)` x `length(scales)` containing the values of the windowed scalograms at each scale and at each time window.
- `tcentral`: The vector of central times at which the windows are centered.
- `scales`: The vector of the scales.
- `windowrad`: Radius for the time windows, measured in `dt`.
- `fourier_factor`: A factor for converting scales to periods.
- `coi_maxscale`: A vector of length `length(tcentral)` containing the values of the maximum scale from which there are border effects for the respective central time.

References

C. Torrence, G. P. Compo. A practical guide to wavelet analysis. B. Am. Meteorol. Soc. 79 (1998), 61–78.

V. J. Bolós, R. Benítez, R. Ferrer, R. Jammazi. The windowed scalogram difference: a novel wavelet tool for comparing time series. Appl. Math. Comput., 312 (2017), 49-65.

R. Benítez, V. J. Bolós, M. E. Ramírez. A wavelet-based tool for studying non-periodicity. Comput. Math. Appl. 60 (2010), no. 3, 634-641.

Examples

```
dt <- 0.1
time <- seq(0, 50, dt)
signal <- c(sin(pi * time), sin(pi * time / 2))
wscalog <- windowed_scalogram(signal = signal, dt = dt)
```

---

wsd *Windowed Scalogram Difference*

Description

This function computes the Windowed Scalogram Difference of two signals. The definition and details can be found in (Bolós et al. 2017).

Usage

```
wsd(signal1,
           signal2,
           dt = 1,
           scaleparam = NULL,
           windowrad = NULL,
           rdist = NULL,
           delta_t = NULL,
           normalize = FALSE,
           wname = c("MORLET", "DOG", "PAUL", "HAAR", "HAAR2"),
           wparam = NULL,
           waverad = NULL,
           border_effects = c("BE", "INNER", "PER", "SYM"),
           mc_nrand = 0,
           commutative = TRUE,
           wscnoise = 0.02,
           compensation = 0,
           energy_density = TRUE,
           parallel = FALSE,
           makefigure = TRUE,
           time_values = NULL,
           figureperiod = TRUE)
```

Arguments

signal1	A vector containing the first signal.
signal2	A vector containing the second signal (its length should be equal to that of signal1).
dt	Numeric. The time step of the signals.

`scaleparam` A vector of three elements with the minimum scale, the maximum scale and the number of suboctaves per octave for constructing power 2 scales (following Torrence and Compo 1998). If NULL, they are automatically computed.

`windowrad` Integer. Time radius for the windows, measured in `dt`. By default, it is set to $celing(length(signal1)/20)$.

`rdist` Integer. Log-scale radius for the windows measured in suboctaves. By default, it is set to $ceiling(length(scales)/20)$.

`delta_t` Integer. Increment of time for the construction of windows central times, measured in `dt`. By default, it is set to $ceiling(length(signal1)/256)$.

`normalize` Logical. Set to TRUE if the signals use different units.

`wname` A string, equal to "MORLET", "DOG", "PAUL", "HAAR" or "HAAR2". The difference between "HAAR" and "HAAR2" is that "HAAR2" is more accurate but slower.

`wparam` The corresponding nondimensional parameter for the wavelet function (Morlet, DoG or Paul).

`waverad` Numeric. The radius of the wavelet used in the computations for the cone of influence. If it is not specified, it is computed by `wavelet_radius` for DoG and Paul wavelets. For Haar and Morlet it is assumed to be 1 and 3 respectively.

`border_effects` String, equal to "BE", "INNER", "PER" or "SYM", which indicates how to manage the border effects which arise usually when a convolution is performed on finite-lenght signals.

- "BE": With border effects, padding time series with zeroes.
- "INNER": Normalized inner scalogram with security margin adapted for each different scale.
- "PER": With border effects, using boundary wavelets (periodization of the original time series).
- "SYM": With border effects, using a symmetric catenation of the original time series.

`mc_nrand` Integer. Number of Montecarlo simulations to be performed in order to determine the 95% and 5% significance contours.

`commutative` Logical. If TRUE (default) the commutative windowed scalogram difference. Otherwise a non-commutative (but simpler) version is computed (see Bolós et al. 2017).

`wscnoise` Numeric in $[0, 1]$. If a (windowed) scalogram takes values close to zero, some problems may appear because we are considering relative differences. Specifically, we can get high relative differences that in fact are not relevant, or even divisions by zero.

If we consider absolute differences this would not happen but, on the other hand, using absolute differences is not appropriate for scalogram values not close to zero.

So, the parameter `wscnoise` stablishes a threshold for the scalogram values above which a relative difference is computed, and below which a difference proportional to the absolute difference is computed (the proportionality factor is determined by requiring continuity).

Finally, wscnoise can be interpreted as the relative amplitude of the noise in the scalograms and is chosen in order to make a relative ($= 0$), absolute ($= 1$) or mix (in $(0, 1)$) difference between scalograms. Default value is set to 0.02.

compensation — Numeric in $[0, 1]$. It is an alternative to wscnoise for preventing numerical errors or non-relevant high relative differences when scalogram values are close to zero (see Bolós et al. 2017).

energy_density — Logical. If TRUE (default), divide the scalograms by the square root of the scales for convert them into energy density. Note that it does not affect the results if wscnoise $= 0$.

parallel — Logical. If TRUE (default) uses function parApply from package parallel for the Montecarlo simulations. When FALSE it uses the normal apply function.

makefigure — Logical. If TRUE (default), a figure with the WSD is plotted.

time_values — A numerical vector of length length(signal) containing custom time values for the figure. If NULL (default), it will be computed starting at 0.

figureperiod — Logical. If TRUE (default), periods are used in the figure instead of scales.

Value

A list with the following fields:

- wsd: A matrix of size length(tcentral) x length(scales) containing the values of the windowed scalogram differences at each scale and at each time window.
- tcentral: The vector of central times used in the computations of the windowed scalograms.
- scales: The vector of scales.
- windowrad: Radius for the time windows of the windowed scalograms, measured in dt.
- rdist: The log-scale radius for the windows measured in suboctaves.
- signif95: A logical matrix of size length(tcentral) x length(scales). If TRUE, the corresponding point of the wsd matrix is in the 95% significance.
- signif05: A logical matrix of size length(tcentral) x length(scales). If TRUE, the corresponding point of the wsd matrix is in the 5% significance.
- fourier_factor: A factor for converting scales to periods.
- coi_maxscale: A vector of length length(tcentral) containing the values of the maximum scale from which there are border effects for the respective central time.

References

C. Torrence, G. P. Compo. A practical guide to wavelet analysis. B. Am. Meteorol. Soc. 79 (1998), 61–78.

V. J. Bolós, R. Benítez, R. Ferrer, R. Jammazi. The windowed scalogram difference: a novel wavelet tool for comparing time series. Appl. Math. Comput., 312 (2017), 49-65.

Examples

```
nt <- 1500
time <- 1:nt
sd_noise <-  0.2 #% In BolÃ³s et al. 2017 Figure 1, sd_noise = 1.
signal1 <- rnorm(n = nt, mean = 0, sd = sd_noise) + sin(time / 10)
signal2 <- rnorm(n = nt, mean = 0, sd = sd_noise) + sin(time / 10)
signal2[500:1000] = signal2[500:1000] + sin((500:1000) / 2)
## Not run:
wsd <- wsd(signal1 = signal1, signal2 = signal2)

## End(Not run)
```

Index