Package: fastWavelets (via r-universe)

September 2, 2024

Title Compute Maximal Overlap Discrete Wavelet Transform (MODWT) and À Trous Discrete Wavelet Transform

Version 2.0.0.9000

Description A lightweight package to compute Maximal Overlap Discrete Wavelet Transform (MODWT) and À Trous Discrete Wavelet Transform by leveraging the power of 'Rcpp' to make these operations fast. This package was designed for use in forecasting, and allows users avoid the inclusion of future data when performing wavelet decomposition of time series. See Quilty and Adamowski (2018) <doi:10.1016/j.jhydrol.2018.05.003>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

LinkingTo Rcpp

Imports Rcpp

URL https://github.com/johnswyou/fastWavelets

BugReports https://github.com/johnswyou/fastWavelets/issues

Depends R (>= 2.10)

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

RemoteUrl https://github.com/johnswyou/fastwavelets

RemoteRef HEAD

RemoteSha f5fd955f3f64f592a2b692f32893b63762949ea9

Contents

`atrous_dwt` *A Trous Discrete Wavelet Transform*

Description

This function calculates the wavelet and scaling coefficients of the a trous (AT) version of the Discrete Wavelet Transform (DWT).

Usage

```
atrous_dwt(X, filter, J, remove_boundary_coefs = FALSE)
```

Arguments

`X`	An (N x 1) matrix or a vector
`filter`	Filter name (see Details below) (string)
`J`	Decomposition level (integer, 1 < J < N/2)
`remove_boundary_coefs`	Remove boundary affected coefficients?

Details

The argument `filter` can take one of the following values:

c('bl7', 'bl9', 'bl10', 'beyl', 'coif1', 'coif2', 'coif3', 'coif4', 'coif5', 'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'db11', 'db12', 'db13', 'db14', 'db15', 'db16', 'db17', 'db18', 'db19', 'db20', 'db21', 'db22', 'db23', 'db24', 'db25', 'db26', 'db27', 'db28', 'db29', 'db30', 'db31', 'db32', 'db33', 'db34', 'db35', 'db36', 'db37', 'db38', 'db39', 'db40', 'db41', 'db42', 'db43', 'db44', 'db45', 'fk4', 'fk6', 'fk8', 'fk14', 'fk18', 'fk22', 'han2_3', 'han3_3', 'han4_5', 'han5_5', 'dmey', 'mb4_2', 'mb8_2', 'mb8_3', 'mb8_4', 'mb10_3', 'mb12_3', 'mb14_3', 'mb16_3', 'mb18_3', 'mb24_3', 'mb32_3', 'sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10', 'sym11', 'sym12', 'sym13', 'sym14', 'sym15', 'sym16', 'sym17', 'sym18', 'sym19', 'sym20', 'sym21', 'sym22', 'sym23', 'sym24', 'sym25', 'sym26', 'sym27', 'sym28', 'sym29', 'sym30', 'sym31', 'sym32', 'sym33', 'sym34', 'sym35', 'sym36', 'sym37', 'sym38', 'sym39', 'sym40', 'sym41', 'sym42', 'sym43', 'sym44', 'sym45', 'vaid', 'la8', 'la10', 'la12', 'la14', 'la16', 'la18', 'la20')

Value

Wavelet and scaling coefficients (N x J+1) (wavelet coefficients in first J columns of returned matrix)

References

Benaouda, D., F. Murtagh, J. L. Starck, and O. Renaud (2006), Wavelet-based nonlinear multiscale decomposition model for electricity load forecasting, Neurocomputing, doi:10.1016/j.neucom.2006.04.005.

Maheswaran, R., and R. Khosa (2012), Comparative study of different wavelets for hydrologic forecasting, Comput. Geosci., doi:10.1016/j.cageo.2011.12.015.

Examples

```
N <- 1000 #  number of time series points
J <- 4 # decomposition level
X <- matrix(rnorm(N),N,1)
W <- atrous_dwt(X,'coif1',J)
plot.ts(W) # plot wavelet and scaling coefficients
```

---

circular_convolution *Circular Convolution*

Description

Perform a circular convolution between a filter and a time series.

Usage

```
circular_convolution(a, b)
```

Arguments

a	Filter (typically, the wavelet or scaling filter)
b	A vector

Details

The convolution theorem tells us that the discrete Fourier transform can be leveraged to compute a circular convolution, which is what this function does.

Value

Result of circular convolution, length is the same as `length(b)`

See Also

`fft`

equivalent_filters *Equivalent Wavelet and Scaling Filters*

Description

Equivalent Wavelet and Scaling Filters

Usage

```
equivalent_filters(filter, J, modwt = FALSE)
```

Arguments

filter	Filter name (string)
J	Decomposition level (integer)
modwt	Do you want the MODWT version?, Default: FALSE

Details

Equivalent wavelet and scaling filters are useful for obtaining MODWT wavelet and scaling coefficients for a given filter name and decomposition level.

Value

A list containing the equivalent wavelet and scaling filter and their lengths.

modwt *Maximal Overlap Discrete Wavelet Transform*

Description

Maximal Overlap Discrete Wavelet Transform

Usage

```
modwt(X, filter, J)
```

Arguments

X	An (N x 1) matrix or a vector
filter	Filter name (see Details below) (string)
J	Decomposition level (integer, 1 < J < N/2)

Details

The argument `filter` can take one of the following values:

c('bl7', 'bl9', 'bl10', 'beyl', 'coif1', 'coif2', 'coif3', 'coif4', 'coif5', 'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'db11', 'db12', 'db13', 'db14', 'db15', 'db16', 'db17', 'db18', 'db19', 'db20', 'db21', 'db22', 'db23', 'db24', 'db25', 'db26', 'db27', 'db28', 'db29', 'db30', 'db31', 'db32', 'db33', 'db34', 'db35', 'db36', 'db37', 'db38', 'db39', 'db40', 'db41', 'db42', 'db43', 'db44', 'db45', 'fk4', 'fk6', 'fk8', 'fk14', 'fk18', 'fk22', 'han2_3', 'han3_3', 'han4_5', 'han5_5', 'dmey', 'mb4_2', 'mb8_2', 'mb8_3', 'mb8_4', 'mb10_3', 'mb12_3', 'mb14_3', 'mb16_3', 'mb18_3', 'mb24_3', 'mb32_3', 'sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10', 'sym11', 'sym12', 'sym13', 'sym14', 'sym15', 'sym16', 'sym17', 'sym18', 'sym19', 'sym20', 'sym21', 'sym22', 'sym23', 'sym24', 'sym25', 'sym26', 'sym27', 'sym28', 'sym29', 'sym30', 'sym31', 'sym32', 'sym33', 'sym34', 'sym35', 'sym36', 'sym37', 'sym38', 'sym39', 'sym40', 'sym41', 'sym42', 'sym43', 'sym44', 'sym45', 'vaid', 'la8', 'la10', 'la12', 'la14', 'la16', 'la18', 'la20')

References

Percival, D. B. and A. T. Walden (2000) Wavelet Methods for Time Series Analysis, Cambridge University Press.

`modwt_wavelet_coefs` *MODWT Wavelet Coefficients*

Description

Compute MODWT wavelet coefficients for a specified filter and decomposition level. This function only gives the MODWT wavelet coefficients for the user specified level. To get all MODWT wavelet coefficients up to and including the user specified decomposition level (as well as the MODWT scaling coefficient corresponding to the user specified decomposition level) all in one nice matrix, please use `mo_dwt()`.

Compute MODWT scaling coefficients for a specified filter and decomposition level. This function only gives the MODWT scaling coefficients for the user specified level.

Usage

```
modwt_wavelet_coefs(X, filter, j)

modwt_scaling_coefs(X, filter, j)
```

Arguments

`X`	A vector
`filter`	Filter name (string)
`j`	Decomposition level (integer)

Details

This function implements the first equation in equation 169a (page 169) of Wavelet Methods for Time Series Analysis by Percival and Walden.

This function implements the second equation in equation 169a (page 169) of Wavelet Methods for Time Series Analysis by Percival and Walden.

Value

MODWT wavelet coefficients for specified decomposition level (vector)

MODWT scaling coefficients for specified decomposition level (vector)

`mo_dwt` *Maximal Overlap Discrete Wavelet Transform (MODWT)*

Description

Maximal Overlap Discrete Wavelet Transform (MODWT)

Usage

```
mo_dwt(X, filter, J, remove_boundary_coefs = FALSE)
```

Arguments

`X`	An (N x 1) matrix or a vector
`filter`	Filter name (see Details below) (string)
`J`	Decomposition level (integer, 1 < J < N/2)
`remove_boundary_coefs`	Remove boundary affected coefficients?

Details

The argument `filter` can take one of the following values:

c('bl7', 'bl9', 'bl10', 'beyl', 'coif1', 'coif2', 'coif3', 'coif4', 'coif5', 'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'db11', 'db12', 'db13', 'db14', 'db15', 'db16', 'db17', 'db18', 'db19', 'db20', 'db21', 'db22', 'db23', 'db24', 'db25', 'db26', 'db27', 'db28', 'db29', 'db30', 'db31', 'db32', 'db33', 'db34', 'db35', 'db36', 'db37', 'db38', 'db39', 'db40', 'db41', 'db42', 'db43', 'db44', 'db45', 'fk4', 'fk6', 'fk8', 'fk14', 'fk18', 'fk22', 'han2_3', 'han3_3', 'han4_5', 'han5_5', 'dmey', 'mb4_2', 'mb8_2', 'mb8_3', 'mb8_4', 'mb10_3', 'mb12_3', 'mb14_3', 'mb16_3', 'mb18_3', 'mb24_3', 'mb32_3', 'sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10', 'sym11', 'sym12', 'sym13', 'sym14', 'sym15', 'sym16', 'sym17', 'sym18', 'sym19', 'sym20', 'sym21', 'sym22', 'sym23', 'sym24', 'sym25', 'sym26', 'sym27', 'sym28', 'sym29', 'sym30', 'sym31', 'sym32', 'sym33', 'sym34', 'sym35', 'sym36', 'sym37', 'sym38', 'sym39', 'sym40', 'sym41', 'sym42', 'sym43', 'sym44', 'sym45', 'vaid', 'la8', 'la10', 'la12', 'la14', 'la16', 'la18', 'la20')

References

Percival, D. B. and A. T. Walden (2000) Wavelet Methods for Time Series Analysis, Cambridge University Press.

`n_boundary_coefs` *Number of Boundary Coefficients*

Description

This function calculates the number of boundary coefficients for a particular wavelet/scaling filter and decomposition level.

Usage

```
n_boundary_coefs(filter, J)
```

Arguments

`filter`	Filter name (see Details below) [string]
`J`	Decomposition level [integer]

Details

The argument `filter` can take one of the following values:

c('bl7', 'bl9', 'bl10', 'beyl', 'coif1', 'coif2', 'coif3', 'coif4', 'coif5', 'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'db11', 'db12', 'db13', 'db14', 'db15', 'db16', 'db17', 'db18', 'db19', 'db20', 'db21', 'db22', 'db23', 'db24', 'db25', 'db26', 'db27', 'db28', 'db29', 'db30', 'db31', 'db32', 'db33', 'db34', 'db35', 'db36', 'db37', 'db38', 'db39', 'db40', 'db41', 'db42', 'db43', 'db44', 'db45', 'fk4', 'fk6', 'fk8', 'fk14', 'fk18', 'fk22', 'han2_3', 'han3_3', 'han4_5', 'han5_5', 'dmey', 'mb4_2', 'mb8_2', 'mb8_3', 'mb8_4', 'mb10_3', 'mb12_3', 'mb14_3', 'mb16_3', 'mb18_3', 'mb24_3', 'mb32_3', 'sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10', 'sym11', 'sym12', 'sym13', 'sym14', 'sym15', 'sym16', 'sym17', 'sym18', 'sym19', 'sym20', 'sym21', 'sym22', 'sym23', 'sym24', 'sym25', 'sym26', 'sym27', 'sym28', 'sym29', 'sym30', 'sym31', 'sym32', 'sym33', 'sym34', 'sym35', 'sym36', 'sym37', 'sym38', 'sym39', 'sym40', 'sym41', 'sym42', 'sym43', 'sym44', 'sym45', 'vaid', 'la8', 'la10', 'la12', 'la14', 'la16', 'la18', 'la20')

Value

Number of boundary coefficients [integer]

References

M. Basta (2014),Additive Decomposition and Boundary Conditions in Wavelet-Based Forecasting Approaches, Acta Oeconomica Pragensia, 2, pp. 48-70.

Quilty, J., & Adamowski, J. (2018). Addressing the incorrect usage of wavelet-based hydrological and water resources forecasting models for real-world applications with best practices and a new forecasting framework. Journal of Hydrology, 563, 336–353. https://doi.org/10.1016/j.jhydrol.2018.05.003

Percival, D. B. and A. T. Walden (2000) Wavelet Methods for Time Series Analysis, Cambridge University Press.

Examples

```
J <- 4 # decomposition level
nbc <- n_boundary_coefs('db1', J) # number of boundary-effected coefficients at decomp_level J
```

`periodize_filter` *Periodize a filter*

Description

Periodize a filter

Usage

```
periodize_filter(filter_vector, periodize_to)
```

Arguments

`filter_vector`	The filter vector that is outputted from r_wavelet_filter or r_scaling_filter
`periodize_to`	The length of the time series that is to be decomposed

Details

Output of the function is a zero padded version of the original filter, padded at the end.

Value

Periodized filter

r_scaling_filter *Scaling Filter*

Description

This function returns the user specified scaling filter.

Usage

```
r_scaling_filter(filter, modwt = FALSE)
```

Arguments

filter	Name of the scaling filter desired [string]
modwt	Should the elements of the returned filter be scaled by 1/sqrt(2)?, Default: FALSE

Details

See equation 75a (page 75) of Wavelet Methods for Time Series Analysis by Percival and Walden to see how to convert a wavelet filter into a scaling filter.

The argument filter can take one of the following values:

c('bl7', 'bl9', 'bl10', 'beyl', 'coif1', 'coif2', 'coif3', 'coif4', 'coif5', 'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'db11', 'db12', 'db13', 'db14', 'db15', 'db16', 'db17', 'db18', 'db19', 'db20', 'db21', 'db22', 'db23', 'db24', 'db25', 'db26', 'db27', 'db28', 'db29', 'db30', 'db31', 'db32', 'db33', 'db34', 'db35', 'db36', 'db37', 'db38', 'db39', 'db40', 'db41', 'db42', 'db43', 'db44', 'db45', 'fk4', 'fk6', 'fk8', 'fk14', 'fk18', 'fk22', 'han2_3', 'han3_3', 'han4_5', 'han5_5', 'dmey', 'mb4_2', 'mb8_2', 'mb8_3', 'mb8_4', 'mb10_3', 'mb12_3', 'mb14_3', 'mb16_3', 'mb18_3', 'mb24_3', 'mb32_3', 'sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10', 'sym11', 'sym12', 'sym13', 'sym14', 'sym15', 'sym16', 'sym17', 'sym18', 'sym19', 'sym20', 'sym21', 'sym22', 'sym23', 'sym24', 'sym25', 'sym26', 'sym27', 'sym28', 'sym29', 'sym30', 'sym31', 'sym32', 'sym33', 'sym34', 'sym35', 'sym36', 'sym37', 'sym38', 'sym39', 'sym40', 'sym41', 'sym42', 'sym43', 'sym44', 'sym45', 'vaid', 'la8', 'la10', 'la12', 'la14', 'la16', 'la18', 'la20')

Value

A scaling filter vector

Examples

```
## Not run:
if(interactive()){
 r_scaling_filter("db1")
 }

## End(Not run)
```

`r_wavelet_filter` *Wavelet Filter*

Description

This function returns the user specified wavelet filter.

Usage

```
r_wavelet_filter(filter, modwt = FALSE)
```

Arguments

`filter`	Name of the wavelet filter desired [string]
`modwt`	Should the elements of the returned filter be scaled by `1/sqrt(2)`?, Default: `FALSE`

Details

See section 4.2 (page 68) of Wavelet Methods for Time Series Analysis by Percival and Walden for a detailed discussion of the wavelet filter.

The argument `filter` can take one of the following values:

c('bl7', 'bl9', 'bl10', 'beyl', 'coif1', 'coif2', 'coif3', 'coif4', 'coif5', 'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'db11', 'db12', 'db13', 'db14', 'db15', 'db16', 'db17', 'db18', 'db19', 'db20', 'db21', 'db22', 'db23', 'db24', 'db25', 'db26', 'db27', 'db28', 'db29', 'db30', 'db31', 'db32', 'db33', 'db34', 'db35', 'db36', 'db37', 'db38', 'db39', 'db40', 'db41', 'db42', 'db43', 'db44', 'db45', 'fk4', 'fk6', 'fk8', 'fk14', 'fk18', 'fk22', 'han2_3', 'han3_3', 'han4_5', 'han5_5', 'dmey', 'mb4_2', 'mb8_2', 'mb8_3', 'mb8_4', 'mb10_3', 'mb12_3', 'mb14_3', 'mb16_3', 'mb18_3', 'mb24_3', 'mb32_3', 'sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'sym9', 'sym10', 'sym11', 'sym12', 'sym13', 'sym14', 'sym15', 'sym16', 'sym17', 'sym18', 'sym19', 'sym20', 'sym21', 'sym22', 'sym23', 'sym24', 'sym25', 'sym26', 'sym27', 'sym28', 'sym29', 'sym30', 'sym31', 'sym32', 'sym33', 'sym34', 'sym35', 'sym36', 'sym37', 'sym38', 'sym39', 'sym40', 'sym41', 'sym42', 'sym43', 'sym44', 'sym45', 'vaid', 'la8', 'la10', 'la12', 'la14', 'la16', 'la18', 'la20')

Value

A wavelet filter vector

Examples

```
## Not run:
if(interactive()){
 r_wavelet_filter("coif1")
 }

## End(Not run)
```

Index