Package ‘wavemulcor’

July 4, 2017

Type Package
Title Wavelet Routines for Global and Local Multiple Correlation
Version 2.1.0
Date 2017-07-01
Author Javier Fernández-Macho (UPV/EHU)
Maintainer Javier Fernandez-Macho <javier.fernandezmacho@ehu.es>
Description Wavelet routines that calculate single sets of wavelet multiple correlations (WMC) and cross-correlations (WMCC) out of n variables (either 1D time series, 2D images or 3D arrays). They can later be plotted in single graphs, as an alternative to trying to make sense out of several sets of wavelet correlations or wavelet cross-correlations.

The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in a linear combination of variables for which such coefficient of determination is a maximum.

The code provided here is based on the wave.correlation routine in Brandon Whitcher’s waveshrink R package Version: 1.6.4, which in turn is based on wavelet methodology developed in Percival and Walden (2000); Gençay, Selçuk and Whitcher (2001) and others.

Version 2 incorporates wavelet local multiple correlations (WLMC). These are like the previous global WMC but consisting in one single set of multiscale correlations along time. That is, at each time t, they are calculated by letting a window of weighted wavelet coefficients around t move along time. Six weight functions are provided. Namely, the uniform window, Cleveland’s tricube window, Epanechnikov’s parabolic window, Bartlett’s triangular window and Wendland’s truncated power window and the Gaussian window.

License GPL (>= 2)
Depends waveshrink
Suggests plot3D
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
NeedsCompilation: no
Repository: CRAN
Date/Publication: 2017-07-03 23:15:28 UTC

R topics documented:

- wavemulcor-package
- wave.local.multiple.correlation
- wave.multiple.correlation
- wave.multiple.cross.correlation
- xrand
- xrand1
- xrand2

Index

 wavedelct-srdoc-package  Wavelet package for global and local multiple correlation

Description

Produces estimates of multiscale global or local multiple correlation (as defined below) along with approximate confidence intervals.

Details

Package: wavemulcor
Type: Package
Version: 2.1.0
Date: 2017-07-01
License: GPL (>= 2)
LazyLoad: yes

The wavemulcor package contains three routines, wave.multiple.correlation, wave.multiple.cross.correlation and wave.local.multiple.correlation, that calculate single sets of, respectively, global wavelet multiple correlations, global wavelet multiple cross-correlations and time-localized wavelet multiple correlations out of n variables. They can later be plotted in single graphs, as an alternative to trying to make sense out of n(n-1)/2 sets of global wavelet correlations or n(n-1)/2.J sets of global wavelet cross-correlations or n(n-1)/2.[JxT] sets of local wavelet correlations respectively. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in that linear combination of either global or locally weighted wavelet coefficients for which such coefficient of determination is a maximum.
Note

Dependencies: waveslim

Author(s)

Javier Fernández-Macho (UPV/EHU)
Maintainer: Javier Fernández-Macho <javier.fernandezmacho@ehu.eus>

References


wave.local.multiple.correlation

*Wavelet routine for local multiple correlation*

Description

Produces an estimate of the multiscale local multiple correlation (as defined below) along with approximate confidence intervals.

Usage

```r
wave.local.multiple.correlation(xx, M, window="gauss", p = .975, ymaxr=NULL)
```

Arguments

- **xx**
  A list of *n* (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* `xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)`
- **M**
  length of the weight function or rolling window.
- **window**
  type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
- **p**
  one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
- **ymaxr**
  index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wmc chooses the one maximizing the multiple correlation.
Details

The routine calculates one single set of wavelet multiple correlations out of \( n \) variables that can be plotted in either single heatmap or \( J \) line graphs (the former is usually the best graphic option but the latter is useful if confidence intervals are explicitly needed), as an alternative to trying to make sense out of \( n(n - 1)/2 \cdot [J \times T] \) sets of local wavelet correlations. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in that linear combination of locally weighted wavelet coefficients for which such coefficient of determination is a maximum. The code provided here is based on the wave.multiple.correlation routine in this package which in turn is based on the wave.correlation routine in Brandon Whitcher’s \texttt{waveslim} \texttt{R} package Version: 1.6.4, which in turn is based on wavelet methodology developed in Percival and Walden (2000); Gençay, Selçuk and Whitcher (2001) and others.

Value

List of four elements:

\texttt{val}:

matrix with as many rows as observations and as many columns as levels in the wavelet transform object providing the point estimates for the wavelet local multiple correlation.

\texttt{lo}:

matrix with as many rows as observations and as many columns as levels in the wavelet transform object providing the lower bounds from the confidence interval.

\texttt{up}:

matrix with as many rows as observations and as many columns as levels in the wavelet transform object providing the upper bounds from the confidence interval.

\texttt{YmaxR}:

numeric vector giving, at each wavelet level, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, \texttt{wlmc} chooses at each wavelet level the variable maximizing the multiple correlation.

Note

Needs \texttt{waveslim} package to calculate \texttt{dwt} or \texttt{modwt} coefficients as inputs to the routine (also for data in the example).

Author(s)


References

Examples

## Based on Figure 4 showing correlation structural breaks in Fernandez-Macho (2017).

```r
rm(list = ls())  # clear objects
graphics.off()  # close graphics windows

library(wavemulcor)
library(plot3D)
options(warn = -1)

data(xrand)
N <- length(xrand)
b <- trunc(N/3)
t1 <- 1:b
t2 <- (b+1):(2*b)
t3 <- (2*b+1):N

wf <- "d4"
M <- N/2^3 # sharper with N/2^4
window <- "gaussian"

J <- trunc(log2(N)-3)

# ##############################################################################

cor1 <- cor(xrand[1:t1], xrand[1:t1])
cor2 <- cor(xrand[1:t2], xrand[1:t2])
cor3 <- cor(xrand[1:t3], xrand[1:t3])
cortext <- paste0(round(100*cor1,0)," ",round(100*cor2,0),"",round(100*cor3,0))

ts.plot(cbind(xrand1,xrand2),col=c("red","blue"),xlab="time")

xrand1.modwt <- modwt(xrand1, wf, J)
xrand1.modwt.bw <- brick.wall(xrand1.modwt, wf)

xrand2.modwt <- modwt(xrand2, wf, J)
xrand2.modwt.bw <- brick.wall(xrand2.modwt, wf)

xx <- list(xrand1.modwt.bw,xrand2.modwt.bw)

# ##############################################################################

xy.mulcor <- wave.local.multiple.correlation(xx, M, window=window)

val <- as.matrix(xy.mulcor$val)
lo <- as.matrix(xy.mulcor$lo)
up <- as.matrix(xy.mulcor$up)
YmaxR <- as.matrix(xy.mulcor$YmaxR)

# ##############################################################################

old.par <- par()
```
# # Producing heat plot

```r
colnames(val[1:j]) <- paste0("level", 1:j)
par(mfrow=c(3,2), las=1, pty="m", mar=c(2,3,1,0)+.1, oma=c(1,2,1.2,0,0))
for(i in 1:j) {
  matplot(1:n, val[,i], type="l", lty=1, ylim=c(-1,1), xaxt="n",
  xlab="", ylab="", main=colnames(val)[i])
  if(i<3) {axis(side=1, at=seq(10,nrow(val),by=10))
  #axis(side=2, at=c(-.2, 0, .5, 1))
  lines(lo[,i], lty=1, col=2)  ## Add Connected Line Segments to a Plot
  lines(up[,i], lty=1, col=2)
  abline(h=0)  ## Add Straight horiz and vert Lines to a Plot
  }
}
par(las=0)
mtext('time', side=1, outer=TRUE, adj=0.5)
mtext('Wavelet Local Multiple Correlation', side=2, outer=TRUE, adj=0.5)

# reset graphics parameters
par(old.par)
```

---

**wave.multiple.correlation**

*Wavelet routine for multiple correlation*
wave.multiple.correlation

Description

Produces an estimate of the multiscale multiple correlation (as defined below) along with approximate confidence intervals.

Usage

wave.multiple.correlation(xx, N, p = 0.975, ymaxr=NULL)

Arguments

- xx: A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, i.e. xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)
- N: length of the time series
- p: one minus the two-sided p-value for the confidence interval, i.e. the cdf value.
- ymaxr: index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wmc chooses the one maximizing the multiple correlation.

Details

The routine calculates one single set of wavelet multiple correlations out of n variables that can be plotted in a single graph, as an alternative to trying to make sense out of n(n-1)/2 sets of wavelet correlations. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in the linear combination of variables for which such coefficient of determination is a maximum. The code provided here is based on the wave.correlation routine in Brandon Whitcher’s waveslim R package Version: 1.6.4, which in turn is based on wavelet methodology developed in Percival and Walden (2000); Gençay, Selçuk and Whitcher (2001) and others.

Value

List of two elements:
- xy.mulcor: matrix with as many rows as levels in the wavelet transform object. The first column provides the point estimate for the wavelet multiple correlation, followed by the lower and upper bounds from the confidence interval.
- YmaxR: numeric vector giving, at each wavelet level, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, wmc chooses at each wavelet level the variable maximizing the multiple correlation.

Note

Needs waveslim package to calculate dwt or modwt coefficients as inputs to the routine (also for data in the example).

Author(s)

References


Examples

```r
## Based on data from Figure 7.8 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.

library(wavemulcor)
data(exchange)
returns <- diff(log(as.matrix(exchange)))
returns <- ts(returns, start=1970, freq=12)
wf <- "d4"
J <- 6
demusd.modwt <- modwt(returns[,"DEM.USD"], wf, J)
demusd.modwt.bw <- brick.wall(demusd.modwt, wf)
jpyusd.modwt <- modwt(returns[,"JPY.USD"], wf, J)
jpyusd.modwt.bw <- brick.wall(jpyusd.modwt, wf)
rand.modwt <- modwt(rnorm(length(returns[,"DEM.USD"])), wf, J)
rand.modwt.bw <- brick.wall(rand.modwt, wf)
xx <- list(demusd.modwt.bw, jpyusd.modwt.bw, rand.modwt.bw)
Lst <- wave.multiple.correlation(xx, N = length(xx[[1]][[1]]))
returns.modwt.cor <- Lst$xy.mulcor[1:J,]
YmaxR <- Lst$YmaxR
exchange.names <- c("DEM.USD", "JPY.USD", "RAND")

##Producing plot
par(mfrow=c(1,1), las=0, mar=c(5,4,4,2)+.1)
matplot(2^(0:(J-1)), returns.modwt.cor[-(J+1),], type="b",
        log="x", pch=*
"LU",
xaxt="n", lty=1, col=c(1,4,4),
xlab="Wavelet Scale", ylab="Wavelet Multiple Correlation")
axis(side=1, at=2^(0:7))
abline(h=0)
text(2^(0:7), min(returns.modwt.cor[-(J+1),])-0.03,
     labels=exchange.names[YmaxR], adj=0.5, cex=.5)
```

wave.multiple.cross.correlation

Wavelet routine for multiple cross-correlation

Description

Produces an estimate of the multiscale multiple cross-correlation (as defined below).
wave.multiple.cross.correlation

**Usage**

```
wave.multiple.cross.correlation(xx, lag.max = NULL, ymaxr = NULL)
```

**Arguments**

- **xx**: A list of \( n \) (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.*
  \[ xx \leftarrow \text{list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)} \]

- **lag.max**: maximum lag. If not set, it defaults to half the square root of the length of the original series.

- **ymaxr**: index number of the variable whose correlation is calculated against a linear combination of the rest. Otherwise at each wavelet level wmc chooses the one maximizing the multiple correlation.

**Details**

The routine calculates one single set of wavelet multiple cross-correlations out of \( n \) variables that can be plotted as one single set of graphs (one per wavelet level), as an alternative to trying to make sense out of \( \frac{n(n-1)}{2} \) sets of wavelet cross-correlations. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in a linear combination of variables that includes a lagged variable for which such coefficient of determination is a maximum.

**Value**

List of two elements:

- **xy.mulcor**: matrix with as many rows as levels in the wavelet transform object. The columns provide the point estimates for the wavelet multiple cross-correlations at different lags.

- **YmaxR**: numeric vector giving, at each wavelet level, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, \textit{wmcc} chooses at each wavelet level the variable maximizing the multiple correlation.

**Note**

Needs \textit{waveslim} package to calculate \textit{dwt} or \textit{modwt} coefficients as inputs to the routine (also for data in the example).

**Author(s)**


**References**

Examples

```r
## Based on data from Figure 7.9 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.

library(wavemulcor)
data(exchange)
returns <- diff(log(exchange))
returns <- ts(returns, start=1970, freq=12)
wf <- "d4"
J <- 6
lmax <- 36
n <- dim(returns)[1]

demusd.modwt <- modwt(returns[,"DEM.USD"], wf, J)
demusd.modwt.bw <- brick.wall(demusd.modwt, wf)
jpyusd.modwt <- modwt(returns[,"JPY.USD"], wf, J)
jpyusd.modwt.bw <- brick.wall(jpyusd.modwt, wf)
rand.modwt <- modwt(rnorm(length(returns[,"DEM.USD"])),wf, J)
rand.modwt.bw <- brick.wall(rand.modwt, wf)

#xx <- list(demusd.modwt.bw, jpyusd.modwt.bw)
xx <- list(demusd.modwt.bw, jpyusd.modwt.bw, rand.modwt.bw)

Lst <- wave.multiple.cross.correlation(xx, lmax)
returns.cross.cor <- as.matrix(Lst$xy.mulcor[1:J,])
YmaxR <- Lst$YmaxR
exchange.names <- c("DEM.USD", "JPY.USD", "RAND")
rownames(returns.cross.cor) <- rownames(returns.cross.cor, do.NULL = FALSE, prefix = "Level ")
lags <- length(-lmax:lmax)

lower.ci <- tanh(atanh(returns.cross.cor) - qnorm(0.975) /
                 sqrt(matrix(trunc(n/2*(1:J)), nrow=J, ncol=lags)- 3))
upper.ci <- tanh(atanh(returns.cross.cor) + qnorm(0.975) /
                 sqrt(matrix(trunc(n/2*(1:J)), nrow=J, ncol=lags)- 3))

par(mfrow=c(3,2), las=1, pty="m", mar=c(2.3,1,0)+.1, oma=c(1.2,1.2,0,0))
for(i in J:1) {  
  matplot((1:(J+lmax+1)),returns.cross.cor[i,], type="l", lty=1, ylim=c(-1,1),
  xaxt="n", xlab="", ylab="", main=rownames(returns.cross.cor)[[i]][1])
  if(i<3) {axis(side=1, at=seq(1, 2*lmax+1, by=12),
    labels=seq(-lmax, lmax, by=12))
  #axis(side=2, at=c(-.2, .0, .5, 1))
  lines(lower.ci[i,], lty=1, col=2)  #Add Connected Line Segments to a Plot
  lines(upper.ci[i,], lty=1, col=2)
  abline(h=0,v=lmax+1)  #Add Straight horiz and vert Lines to a Plot
  text(1,1, labels=exchange.names[YmaxR[i]], adj=0.25, cex=.8)
  }
par(las=0)
mtext('Lag (months)', side=1, outer=TRUE, adj=0.5)
mtext('Wavelet Multiple Cross-Correlation', side=2, outer=TRUE, adj=0.5)
```
Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2017).

Usage

data("xrand")

Format

A data frame with 512 observations on the following 2 variables.

xrand1  a numeric vector
xrand2  a numeric vector

Details

$xrand1[t]$ and $xrand2[t]$ are highly correlated at low frequencies (long timescales) but uncorrelated at high frequencies (short timescales). However, during a period of time spanning the second third of the sample ($T/3 < t < 2T/3$) that behavior is reversed so that data become highly correlated at short timescales but uncorrelated at low frequencies.

References


Examples

data(xrand)
## maybe str(xrand); plot(xrand) ...

Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2017).

Usage

data("xrand")
Format

A data frame with 512 observations on 1 variables.

xrand1 a numeric vector

Details

$xrand1[t]$ and $xrand2[t]$ are highly correlated at low frequencies (long timescales) but uncorrelated at high frequencies (short timescales). However, during a period of time spanning the second third of the sample ($T/3 < t < 2T/3$) that behavior is reversed so that data become highly correlated at short timescales but uncorrelated at low frequencies.

References


Examples

data(xrand)
## maybe str(xrand) ; plot(xrand) ...

Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2017).

Usage

data("xrand")

Format

A data frame with 512 observations on 1 variables.

xrand2 a numeric vector

Details

$xrand1[t]$ and $xrand2[t]$ are highly correlated at low frequencies (long timescales) but uncorrelated at high frequencies (short timescales). However, during a period of time spanning the second third of the sample ($T/3 < t < 2T/3$) that behavior is reversed so that data become highly correlated at short timescales but uncorrelated at low frequencies.

References

Examples

data(xrand)

## maybe str(xrand); plot(xrand) ...
Index

*Topic correlation
  wave.multiple.correlation, 6
*Topic cross-correlation
  wave.multiple.cross.correlation, 8
*Topic datasets
  xrand, 11
  xrand1, 11
  xrand2, 12
*Topic local correlation
  wave.local.multiple.correlation, 3
*Topic local regression
  wave.local.multiple.correlation, 3
*Topic moving regression
  wave.local.multiple.correlation, 3
*Topic multivariate
  wave.local.multiple.correlation, 3
  wave.multiple.correlation, 6
  wave.multiple.cross.correlation, 8
*Topic statistics
  wave.local.multiple.correlation, 3
  wave.multiple.correlation, 6
  wave.multiple.cross.correlation, 8
*Topic wavelet
  wave.local.multiple.correlation, 3
  wave.multiple.correlation, 6
  wave.multiple.cross.correlation, 8
  wave.multiple.correlation, 6
  wave.multiple.cross.correlation, 8
  wavemulcor (wavemulcor-package), 2
  wavemulcor-package, 2
  wlmc (wave.local.multiple.correlation), 3
  wmc (wave.multiple.correlation), 6
  wmcc (wave.multiple.cross.correlation), 8

xrand, 11
xrand1, 11