

Package ‘MF DFA’

January 20, 2025

Type Package

Title MultiFractal Detrended Fluctuation Analysis

Version 1.1

Author Mohamed Laib [aut, cre],
Luciano Telesca [aut],
Mikhail Kanevski [aut]

Maintainer Mohamed Laib <laib.med@gmail.com>

Description Contains the MultiFractal Detrended Fluctuation Analysis (MF DFA), MultiFractal Detrended Cross-Correlation Analysis (MFX DFA), and the Multiscale Multifractal Analysis (MMA). The MF DFA() function proposed in this package was used in Laib et al. (<doi:10.1016/j.chaos.2018.02.024> and <doi:10.1063/1.5022737>). See references for more information. Interested users can find a parallel version of the MF DFA() function on GitHub.

License GPL-3

Imports numbers

URL <https://mlaib.github.io>

Note The originale code was in matlab, see details below.

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-15 15:22:53 UTC

Contents

MF DFA	2
MFsim	5
MFX DFA	6
MMA	8

Index	11
--------------	-----------

Description

Applies the MultiFractal Detrended Fluctuation Analysis (MFDFA) to time series.

Usage

```
MFDFA(tsx, scale, m=1, q)
```

Arguments

tsx	Univariate time series (must be a vector).
scale	Vector of scales. There is no default value for this parameter, please add values.
m	An integer of the polynomial order for the detrending (by default m=1).
q	q-order of the moment. There is no default value for this parameter, please add values.

Details

The original code of this function is in Matlab, you can find it on the following website [Mathworks](#).

Value

A list of the following elements:

- Hq Hurst exponent.
- tau_q Mass exponent.
- spec Multifractal spectrum (α and $f(\alpha)$)
- Fq Fluctuation function.

References

- J. Feder, Fractals, Plenum Press, New York, NY, USA, 1988.
- Espen A. F. Ihlen, Introduction to multifractal detrended fluctuation analysis in matlab, *Frontiers in Physiology: Fractal Physiology*, 3 (141),(2012) 1-18.
- J. W. Kantelhardt, S. A. Zschiegner, E. Koscielny-Bunde, S. Havlin, A. Bunde, H. Stanley, Multi-fractal detrended fluctuation analysis of nonstationary time series, *Physica A: Statistical Mechanics and its Applications*, 316 (1) (2002) 87 – 114.
- Kantelhardt J.W. (2012) Fractal and Multifractal Time Series. In: Meyers R. (eds) *Mathematics of Complexity and Dynamical Systems*. Springer, New York, NY.
- M. Laib, L. Telesca and M. Kanevski, Long-range fluctuations and multifractality in connectivity density time series of a wind speed monitoring network, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28 (2018) p. 033108, [Paper](#).
- M. Laib, J. Golay, L. Telesca, M. Kanevski, Multifractal analysis of the time series of daily means of wind speed in complex regions, *Chaos, Solitons & Fractals*, 109 (2018) pp. 118-127, [Paper](#).

Examples

```

## Not run:
## MFDFA package installation: from github ####
install.packages("devtools")
devtools::install_github("mlaib/MFDFA")

## Get the Parellel version:
devtools::source_gist("bb0c09df9593dad16ae270334ec3e7d7", filename = "MFDFA2.r")

## End(Not run)

library(MFDFA)
a<-0.9
N<-1024
tsx<-MFsim(N,a)
scale=10:100
q<--10:10
m<-1
b<-MFDFA(tsx, scale, m, q)

## Not run:
## Results plot ####
dev.new()
par(mai=rep(1, 4))
plot(q, b$Hq, col=1, axes= F, ylab=expression('h'[q]), pch=16, cex.lab=1.8,
      cex.axis=1.8, main="Hurst exponent",
      ylim=c(min(b$Hq),max(b$Hq)))
grid(col="midnightblue")
axis(1)
axis(2)

#####
## Suggestion of output plot: ####
## Supplementary functions: #####
reset <- function(){
par(mfrow=c(1, 1), oma=rep(0, 4), mar=rep(0, 4), new=TRUE)
plot(0:1, 0:1, type="n", xlab="", ylab="", axes=FALSE)}

poly_fit<-function(x,y,n){
  formule<-lm(as.formula(paste('y~',paste('I(x^',1:n,')', sep='',collapse='+'))))
  res1<-coef(formule)
  poly.res<-res1[length(res1):1]
  allres<-list(polyfit=poly.res, model1=formule)
  return(allres)}

#####
## Output plots: #####
dev.new()
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE),heights=c(4, 4))
## b : mfdfa output
par(mai=rep(0.8, 4))

```

```

## 1st plot: Scaling function order Fq (q-order RMS)
p1<-c(1,which(q==0),which(q==q[length(q)]))
plot(log2(scale),log2(b$Fqi[,1]), pch=16, col=1, axes = F, xlab = "s (days)",
      ylab=expression('log'[2]*'(F'[q]*')'), cex=1, cex.lab=1.6, cex.axis=1.6,
      main= "Fluctuation function Fq",
      ylim=c(min(log2(b$Fqi[,c(p1)])),max(log2(b$Fqi[,c(p1)]))))
lines(log2(scale),b$line[,1], type="l", col=1, lwd=2)
grid(col="midnightblue")
axis(2)
lbl<-scale[c(1,floor(length(scale)/8),floor(length(scale)/4),
            floor(length(scale)/2),length(scale))]
att<-log2(lbl)
axis(1, at=att, labels=lbl)
for (i in 2:3){
  k<-p1[i]
  points(log2(scale), log2(b$Fqi[,k]), col=i,pch=16)
  lines(log2(scale),b$line[,k], type="l", col=i, lwd=2)
}
legend("bottomright", c(paste('q', '=',q[p1] , sep=' ')),cex=2,lwd=c(2,2,2),
      bty="n", col=1:3)

## 2nd plot: q-order Hurst exponent
plot(q, b$Hq, col=1, axes= F, ylab=expression('h'[q]), pch=16, cex.lab=1.8,
      cex.axis=1.8, main="Hurst exponent", ylim=c(min(b$Hq),max(b$Hq)))
grid(col="midnightblue")
axis(1, cex=4)
axis(2, cex=4)

## 3rd plot: q-order Mass exponent
plot(q, b$tau_q, col=1, axes=F, cex.lab=1.8, cex.axis=1.8,
      main="Mass exponent",
      pch=16,ylab=expression(tau[q]))
grid(col="midnightblue")
axis(1, cex=4)
axis(2, cex=4)

## 4th plot: Multifractal spectrum
plot(b$spec$hq, b$spec$Dq, col=1, axes=F, pch=16, #main="Multifractal spectrum",
      ylab=bquote("f (~alpha~)"),cex.lab=1.8, cex.axis=1.8,
      xlab=bquote(~alpha))
grid(col="midnightblue")
axis(1, cex=4)
axis(2, cex=4)

x1=b$spec$hq
y1=b$spec$Dq
rr<-poly_fit(x1,y1,4)
mm1<-rr$model1
mm<-rr$polyfit
x2<-seq(0,max(x1)+1,0.01)
curv<-mm[1]*x2^4+mm[2]*x2^3+mm[3]*x2^2+mm[4]*x2+mm[5]
lines(x2,curv, col="red", lwd=2)
reset()

```

```

legend("top", legend="MFDFA Plots", bty="n", cex=2)

## End(Not run)

```

MFsim

Simulated multifractal series.

Description

Generates series using the binomial multifractal model (see references).

Usage

```
MFsim(N,a)
```

Arguments

N	The length of the generated multifractal series.
a	Exponent that takes values in [0.6, 1].

Value

A vector containing the multifractal series.

References

J. Feder, *Fractals*, Plenum Press, New York, NY, USA, 1988.

E.L. Flores-Márquez, A. Ramírez-Rojas, L. Telesca, Multifractal detrended fluctuation analysis of earthquake magnitude series of Mexican South Pacific Region, *Applied Mathematics and Computation*, Volume 265, 2015, Pages 1106-1114, ISSN 0096-3003.

Examples

```

a<-0.9
N<-1024
tsx<-MFsim(N,a)
scale=10:100
q--10:10
m<-1
b<-MFDFA(tsx, scale, m, q)

dev.new()
par(mai=rep(1, 4))
plot(q, b$Hq, col=1, axes= FALSE, ylab=expression('h'[q]), pch=16, cex.lab=1.8,
      cex.axis=1.8, main="q-order Hurst exponent", ylim=c(min(b$Hq),max(b$Hq)))
grid(col="midnightblue")
axis(1)
axis(2)

```

```
## Not run:
## Example with Levy distribution ####
require(rmutil)
tsx <- rlevy(1000, 0, 1)
scale=10:100
q<--10:10
m<-1
b<-MFXDFA(tsx, scale, m, q)

dev.new()
plot(q, b$Hq, col=1, axes= F, ylab=expression('h'[q]), pch=16, cex.lab=1.8,
      cex.axis=1.8, main="Hurst exponent", ylim=c(min(b$Hq),max(b$Hq)))
grid(col="midnightblue")
axis(1, cex=4)
axis(2, cex=4)

## End(Not run)
```

MFXDFA

Multifractal detrended cross-correlation analysis

Description

Applies the MultiFractal Detrended Fluctuation cross-correlation Analysis (MFXDFA) on two time series.

Usage

```
MFXDFA(tsx1, tsx2, scale, m=1, q)
```

Arguments

tsx1	Univariate time series (must be a vector or a ts object).
tsx2	Univariate time series (must be a vector or a ts object).
scale	Vector of scales.
m	Polynomial order for the detrending (by default m=1).
q	q-order of the moment. There is no default value for this parameter, please add values.

Value

A list of the following elements:

- Hq Hurst exponent.
- h Holder exponent.
- Dh Multifractal spectrum.
- Fq Fluctuation function in log.

Note

The original code of this function is in Matlab, you can find it on the following website [Mathworks](#).

References

J. Feder, Fractals, Plenum Press, New York, NY, USA, 1988.

Espen A. F. Ihlen, Introduction to multifractal detrended fluctuation analysis in matlab, *Frontiers in Physiology: Fractal Physiology*, 3 (141),(2012) 1-18.

J. W. Kantelhardt, S. A. Zschiegner, E. Koscielny-Bunde, S. Havlin, A. Bunde, H. Stanley, Multifractal detrended fluctuation analysis of nonstationary time series, *Physica A: Statistical Mechanics and its Applications*, 316 (1) (2002) 87 – 114.

Kantelhardt J.W. (2012) Fractal and Multifractal Time Series. In: Meyers R. (eds) *Mathematics of Complexity and Dynamical Systems*. Springer, New York, NY.

Examples

```
library(MFDFA)
a<-0.6
N<-1024

tsx1<-MFsim(N,a)
b<-0.8
N<-1024

tsx2<-MFsim(N,b)
scale=10:100
q<--10:10
m<-1

## Not run:
b<-MFXDFA(tsx1, tsx2, scale, m=1, q)

## Supplementary functions: #####
reset <- function(){
par(mfrow=c(1, 1), oma=rep(0, 4), mar=rep(0, 4), new=TRUE)
plot(0:1, 0:1, type="n", xlab="", ylab="", axes=FALSE)}

poly_fit<-function(x,y,n){
  formule<-lm(as.formula(paste('y~',paste('I(x^',1:n,')', sep=' ',collapse='+'))))
  res1<-coef(formule)
  poly.res<-res1[length(res1):1]
  allres<-list(polyfit=poly.res, model1=formule)
  return(allres)}

## Plot results: #####
dev.new()
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE),heights=c(4, 4))
## b : mdfa output
par(mai=rep(0.8, 4))
```

```

## 1st plot: Fluctuations function
p1<-which(q==2)
plot(log(scale),b$Fq[,p1], pch=16, col=1, axes = FALSE, xlab = "s",
      ylab=expression('log'*(F'[2]*')), cex=1, cex.lab=1.6, cex.axis=1.6,
      main= "Fluctuation function F for q=2",
      ylim=c(min(b$Fq[,c(p1)]),max(b$Fq[,c(p1)])))
lines(log(scale),b$line[,p1], type="l", col=1, lwd=2)
grid(col="midnightblue")
axis(2)
lbl<-scale[c(1,floor(length(scale)/8),floor(length(scale)/4),
            floor(length(scale)/2),length(scale))]
att<-log(lbl)
axis(1, at=att, labels=lbl)

## 2nd plot: q-order Hurst exponent
plot(q, b$Hq, col=1, axes= FALSE, ylab=expression('h'[q]), pch=16, cex.lab=1.8,
      cex.axis=1.8, main="Hurst exponent", ylim=c(min(b$Hq),max(b$Hq)))
grid(col="midnightblue")
axis(1, cex=4)
axis(2, cex=4)

## 3rd plot: Spectrum
plot(b$h, b$Dh, col=1, axes=FALSE, pch=16, main="Multifractal spectrum",
      ylab=bquote("f (~alpha~)"),cex.lab=1.8, cex.axis=1.8,
      xlab=bquote(~alpha))
grid(col="midnightblue")
axis(1, cex=4)
axis(2, cex=4)

x1=b$h
y1=b$Dh
rr<-poly_fit(x1,y1,4)
mm1<-rr$model1
mm<-rr$polyfit
x2<-seq(min(x1),max(x1)+1,0.01)
curv<-mm[1]*x2^4+mm[2]*x2^3+mm[3]*x2^2+mm[4]*x2+mm[5]
lines(x2,curv, col="red", lwd=2)

## End(Not run)

```

Description

Applies the Multiscale Multifractal Analysis (MMA) on time series.

Usage

```
MMA(tsx, scale, qminmax, overlap=0, m=2)
```


Arguments

tsx	Univariate time series (must be a vector or a ts object).
scale	Vector of scales.
qminmax	Vector of two values min and max of q-order of the moment.
ovlap	Overlapping parameter (By default ovlap=0: no overlapping).
m	Polynomial order for the detrending (by defaults m=2).

Value

A matrix with three columns (q-order, scale (s), and the scale exponent).

Note

The original code of this function is in Matlab, you can find it on the following website [Physionet](#). See references below.

References

- J. Feder, Fractals, Plenum Press, New York, NY, USA, 1988.
- J. Gieraltowski, J. J. Zebrowski, and R. Baranowski, Multiscale multifractal analysis of heart rate variability recordings <http://dx.doi.org/10.1103/PhysRevE.85.021915>
- Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220.
- J. W. Kantelhardt, S. A. Zschiegner, E. Koscielny-Bunde, S. Havlin, A. Bunde, H. Stanley, Multi-fractal detrended fluctuation analysis of nonstationary time series, Physica A: Statistical Mechanics and its Applications, 316 (1) (2002) 87 – 114.
- J. Gieraltowski, J. J. Żebrowski, and R. Baranowski, "Multiscale multifractal analysis of heart rate variability recordings with a large number of occurrences of arrhythmia," Phys. Rev. E 85, 021915 (2012)

Examples

```
## Not run:
library(MFDFA)
library(plotly)
library(plot3D)

a<-0.6
N<-800
tsx<-MFsim(N,a)
scale=10:100
res<-MMA(tsx, scale, qminmax=c(-10,10), ovlap=0, m=2)

## Visualisation 1:
S_exponent <- matrix(res[,3], nrow=length(unique(res[,1])), ncol=length(min(scale):(max(scale)/5)))
m_scale <- unique(res[,2])
```

```
q <- unique(res[,1])
plot_ly() %>% add_surface(x = ~m_scale, y = ~q,
                        z = ~S_exponent)

## Visualisation 2:
image2D(S_exponent, xlab="q", ylab="scale", axes=F)
axis(1, seq(0,1,0.1), round(quantile(q, seq(0, 1, 0.1)), 2))
axis(2, seq(0,1,0.1), round(quantile(m_scale, seq(0, 1, 0.1)), 2))

## End(Not run)
```

Index

MF DFA, 2
MFsim, 5
MFXDFA, 6
MMA, 8