Package ‘MFT’

July 21, 2025

Type Package

Title The Multiple Filter Test for Change Point Detection

Version 2.0

Date 2019-03-11

Author Michael Messer, Stefan Albert, Solveig Plomer, Gaby Schneider

Maintainer Michael Messer <messer@math.uni-frankfurt.de>

Description Provides statistical tests and algorithms for the detection of change points in time series and point processes - particularly for changes in the mean in time series and for changes in the rate and in the variance in point processes. References - Michael Messer, Marietta Kirchner, Julia Schiemann, Jochen Roeper, Ralph Neininger and Gaby Schneider (2014), A multiple filter test for the detection of rate changes in renewal processes with varying variance <doi:10.1214/14-AOAS782>. Stefan Albert, Michael Messer, Julia Schiemann, Jochen Roeper, Gaby Schneider (2017), Multi-scale detection of variance changes in renewal processes in the presence of rate change points <doi:10.1111/jtsa.12254>. Michael Messer, Kaue M. Costa, Jochen Roeper and Gaby Schneider (2017), Multi-scale detection of rate changes in spike trains with weak dependencies <doi:10.1007/s10827-016-0635-3>. Michael Messer, Stefan Albert and Gaby Schneider (2018), The multiple filter test for change point detection in time series <doi:10.1007/s00184-018-0672-1>. Michael Messer, Hendrik Backhaus, Albrecht Stroh and Gaby Schneider (2019+) Peak detection in time series.

License GPL-3

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-11 20:42:55 UTC

Contents

`MFT.filterdata` *MFT.filterdata*

Description

Naive routine to remove trend from the data.

Usage

```
MFT.filterdata(x, filterwidth = NULL, filtersigma = NULL)
```

Arguments

`x`	numeric vector, input sequence of random variables.
`filterwidth`	postive interger, < length(x)/2, number of data points left and right of the current value that are taken into account for Gaussian smoothing.
`filtersigma`	numeric, > 0, standard deviation of Gassian kernel.

Value

invisible

`xfiltered`	filtered data (for filtering the first and last (filterwidth many) data points of the original series cannot be evaluated and are omited)
`xraw`	orignal data, but the first and last (filterwidth many) data point are omitted
`xtrend`	trend that is removed by filtering. That is xfiltered = xraw - xtrend
`x`	orignal data
`filterwidth`	number of data points left and right of the current value that are taken into account for Gaussian smoothing
`filtersigma`	standard deviation of the Gaussian kernel

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Michael Messer, Hendrik Backhaus, Albrecht Stroh and Gaby Schneider (2019+). Peak detection in times series

See Also

MFT.peaks, plot.MFT, summary.MFT, MFT.rate, MFT.variance, MFT.mean

Examples

```
set.seed(0)
# Normally distributed sequence with negative trend
x <- rnorm(1000,mean=seq(5,0,length.out=1000))
MFT.filterdata(x)
MFT.filterdata(x,filterwidth=200,filtersigma=200)
```

MFT.mean *MFT.mean*

Description

The multiple filter test for mean change detection in time series or sequences of random variables.

Usage

```
MFT.mean(X, autoset.H = TRUE, S = NULL, E = NULL, H = NULL,
  alpha = 0.05, method = "asymptotic", sim = 10000,
  rescale = FALSE, Q = NA, perform.CPD = TRUE, print.output = TRUE)
```

Arguments

X	numeric vector, input sequence of random variables
autoset.H	logical, automatic choice of window size H
S	numeric, start of time interval, default: NULL, if NULL then 1 is chosen
E	numeric, end of time interval, default: NULL, if NULL then length(X) is chosen, needs E > S.
H	vector, window set H, all elements must be increasing, the largest element must be =< (T/2). H is automatically set if autoset.H = TRUE
alpha	numeric, in (0,1), significance level
method	either "asymptotic" or "fixed", defines how threshold Q is derived, default: "asymptotic", If "asymptotic": Q is derived by simulation of limit process L (Brownian motion); possible set number of simulations (sim), If "fixed": Q may be set manually (Q)
sim	integer, > 0, No of simulations of limit process (for approximation of Q), default = 10000
rescale	logical, if TRUE statistic G is rescaled to statistic R, default = FALSE
Q	numeric, rejection threshold, default: Q is simulated according to sim and alpha.
perform.CPD	logical, if TRUE change point detection algorithm is performed
print.output	logical, if TRUE results are printed to the console

Value

invisible

M	test statistic
Q	rejection threshold
method	how threshold Q was derived, see 'Arguments' for detailed description
sim	number of simulations of the limit process (approximation of Q)
rescale	states whether statistic G is rescaled to R
CP	set of change points estmated by the multiple filter algorithm, increasingly ordered in time
means	estimated mean values between adjacent change points
S	start of time interval
E	end of time interval
Tt	length of time interval
H	window set
alpha	significance level
perform.CPD	logical, if TRUE change point detection algorithm was performed
tech.var	list of technical variables with processes X and G_ht or R_ht
type	type of MFT which was performed: "mean"

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Michael Messer, Stefan Albert and Gaby Schneider (2018). The multiple filter test for change point detection in time series. Metrika <doi:10.1007/s00184-018-0672-1>

See Also

plot.MFT, summary.MFT, MFT.rate, MFT.variance, MFT.peaks

Examples

```
# Normal distributed sequence with 3 change points of the mean (at n=100, 155, 350)
set.seed(50)
X1   <- rnorm(400,0,1); X2 <- rnorm(400,3,1); X3 <- rnorm(400,5,1); X4 <- rnorm(600,4.6,1)
X    <- c(X1[1:100],X2[101:155],X3[156:350],X4[351:600])
mft  <- MFT.mean(X)
plot(mft)
# Set additional parameters (window set)
mft2 <- MFT.mean(X,autoset.H=FALSE,H=c(80,160,240))
plot(mft2)
```

MFT.m_est *MFT.m_est*

Description

Naive routine for the estimation of the order of serial correlation (m-dependence) in point processes.

Usage

```
MFT.m_est(Phi, n = 200, maxlag = 10, alpha = 0.05, plot = TRUE)
```

Arguments

Phi	point process, vector of time stamps
n	positive integer, number of life times used in segments for estimation of serial correlation
maxlag	non-negative integer, maximal lag up to which serial correlations are calculated
alpha	numeric, in (0,1), significance level
plot	logical, if TRUE, estimation procedure is plotted

Value

m_est	non-negative integer, estimated order of serial correlation (m-dependence)

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Michael Messer, Kaue M. Costa, Jochen Roeper and Gaby Schneider (2017). Multi-scale detection of rate changes in spike trains with weak dependencies. Journal of Computational Neuroscience, 42 (2), 187-201. <doi:10.1007/s10827-016-0635-3>

See Also

MFT.rate, plot.MFT, summary.MFT, MFT.variance, MFT.mean, MFT.peaks

Examples

```
# 1. Independent life times (m=0)
set.seed(117)
n <- 5000
Phi1 <- cumsum(rexp(n,3.5))
Phi2 <- cumsum(rexp(n,5))
Phi3 <- cumsum(rexp(n,2))
Phi  <- c(Phi1[Phi1<=200],Phi2[Phi2>200 & Phi2<400],Phi3[Phi3>400 & Phi3<700])
```

```
MFT.m_est(Phi)

# 2. Point process simulated according to model
# X_i = a_0 X_i + a_1 X_{i-1} + ... a_m X_{i-m}
# with life times X_i gamma-distributed, 2 change points and true m = 3.
set.seed(210)
Tt <- 3000
m <- 3
a <- c(1,0.5,0.25,0.125)
mu <- c(0.5,1,2)/(sum(a))
sigmaX <- sqrt(0.225/(sum(a^2)))
shape <- mu^2/sigmaX^2; rate <- mu/sigmaX^2
len <- 10000
# build auxiliary processes
X1 <- rgamma(len,rate=rate[1],shape=shape[1]); M1 <- embed(X1,m+1)
v1 <- cumsum(as.vector(M1 %*% a)); v1 <- v1[v1<Tt]
X2 <- rgamma(len,rate=rate[2],shape=shape[2]); M2 <- embed(X2,m+1)
v2 <- cumsum(as.vector(M2 %*% a)); v2 <- v2[v2<Tt]
X3 <- rgamma(len,rate=rate[3],shape=shape[3]); M3 <- embed(X3,m+1)
v3 <- cumsum(as.vector(M3 %*% a)); v3 <- v3[v3<Tt]
# build final point process with cps at 100 and 200
Phi <- c(v1[v1<Tt/3],v2[v2>Tt/3 & v2<(2/3)*Tt],v3[v3>(2/3)*Tt])
# estimate m
MFT.m_est(Phi)
```

`MFT.peaks` *MFT.peaks*

Description

The multiple filter test for peak detection in time series or sequences of random variables

Usage

```
MFT.peaks(x, autoset.H = TRUE, S = NULL, E = NULL, H = NULL,
  alpha = 0.05, method = "asymptotic", sim = 10000, Q = NA,
  blocksize = NA, two.sided = FALSE, perform.CPD = TRUE,
  print.output = TRUE)
```

Arguments

`x`	numeric vector, input sequence of random variables
`autoset.H`	logical, automatic choice of window size H
`S`	numeric, start of time interval, default: NULL, if NULL then 1 is chosen
`E`	numeric, end of time interval, default: NULL, if NULL then length(X) is chosen, needs E > S

`H`	vector, window set H, the smallest element must >= 3 be and the largest =< (T/2). H is automatically set if autoset.H = TRUE
`alpha`	numeric, in (0,1), significance level
`method`	either "asymptotic", "bootstrap" or "fixed", defines how threshold Q is derived, default: "asymptotic", If "asymptotic": Q is derived by simulation of limit process L (Gaussian process); possible set number of simulations (sim), If "bootstrap": Q is derived by (Block)-Bootstrapping; possibly set number of simulations (sim) and blocksize (blocksize), If "fixed": Q may be set manually (Q)
`sim`	integer, > 0, No of simulations of limit process (for approximation of Q), default = 10000
`Q`	numeric, rejection threshold, default: Q is simulated according to sim and alpha
`blocksize`	NA or integer >= 1, if method == 'bootstrap', blocksize determines the size of blocks (number of life times) for bootstrapping
`two.sided`	logical, if TRUE a two sided test is performed and also negative peaks are considered in peak detection
`perform.CPD`	logical, if TRUE change point detection algorithm is performed
`print.output`	logical, if TRUE results are printed to the console

Value

invisible

`M`	test statistic
`Q`	rejection threshold
`method`	how threshold Q was derived, see 'Arguments' for detailed description
`sim`	number of simulations of the limit process (approximation of Q)
`blocksize`	size of blocks (number of life times) for bootstrapping (approximation of Q)
`CP`	set of change points estmated by the multiple filter algorithm, increasingly ordered in time
`S`	start of time interval
`E`	end of time interval
`Tt`	length of time interval
`H`	window set
`alpha`	significance level
`two.sided`	logigal, if TRUE also negative peaks are considered
`perform.CPD`	logical, if TRUE change point detection algorithm was performed
`tech.var`	list of technical variables with processes x and D_ht
`type`	type of MFT which was performed: "peaks"

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Michael Messer, Hendrik Backhaus, Albrecht Stroh and Gaby Schneider (2019+). Peak detection in times series

See Also

MFT.filterdata, plot.MFT, summary.MFT, MFT.mean, MFT.rate, MFT.variance

Examples

```
# Normal distributed sequence with 2 peaks
set.seed(12)
m <- c(rep(0,30),seq(0,3,length.out = 100),seq(3,0,length.out = 80),rep(0,10),
       seq(0,6,length.out = 50),seq(6,0,length.out = 50),rep(0,30))
x <- rnorm(length(m),m)
mft <- MFT.peaks(x)
plot(mft)
# Set additional parameters (window set)
mft <- MFT.peaks(x,autoset.H = FALSE, H =c(30,60,90))
plot(mft)
```

MFT.rate *MFT.rate*

Description

The multiple filter test for rate change detection in point processes on the line.

Usage

```
MFT.rate(Phi, m = 0, cutout = TRUE, autoset.d_H = TRUE, S = NULL,
  E = NULL, d = NULL, H = NULL, alpha = 0.05,
  method = "asymptotic", sim = 10000, rescale = FALSE, Q = NA,
  blocksize = NA, perform.CPD = TRUE, print.output = TRUE)
```

Arguments

Phi	numeric vector of increasing events, input point process
m	non-negative integer, dependence parameter: serial corellation rho up to order m estimated
cutout	logical, if TRUE for every point, for which the estimated rho becomes negative, the h-neighborhood of G (resp. R) is set to zero. This might only occur, if m > 0
autoset.d_H	logical, automatic choice of window size H and step size d
S	numeric, start of time interval, default: Smallest multiple of d that lies beyond min(Phi)

`E`	numeric, end of time interval, default: Smallest multiple of d that lies beyond max(Phi), needs E > S.
`d`	numeric, > 0, step size delta at which processes are evaluated. d is automatically set if autoset.d_H = TRUE
`H`	vector, window set H, all elements must be increasing ordered multiples of d, the smallest element must be >= d and the largest =< (T/2). H is automatically set if autoset.d_H = TRUE
`alpha`	numeric, in (0,1), significance level
`method`	either "asymptotic", "bootstrap" or "fixed", defines how threshold Q is derived, default: "asymptotic", If "asymptotic": Q is derived by simulation of limit process L (Brownian motion); possible set number of simulations (sim), If "bootstrap": Q is derived by (Block)-Bootstrapping; possibly set number of simulations (sim) and blocksize (blocksize), If "fixed": Q may be set manually (Q)
`sim`	integer, > 0, No of simulations of limit process (for approximation of Q), default = 10000
`rescale`	logical, if TRUE statistic G is rescaled to statistic R, default = FALSE
`Q`	numeric, rejection threshold, default: Q is simulated according to sim and alpha.
`blocksize`	NA or integer >= 1, if method == 'bootstrap', blocksize determines the size of blocks (number of life times) for bootstrapping
`perform.CPD`	logical, if TRUE change point detection algorithm is performed
`print.output`	logical, if TRUE results are printed to the console

Value

invisible

`M`	test statistic
`Q`	rejection threshold
`method`	how threshold Q was derived, see 'Arguments' for detailed description
`sim`	number of simulations of the limit process (approximation of Q)
`blocksize`	size of blocks (number of life times) for bootstrapping (approximation of Q)
`rescale`	states whether statistic G is rescaled to R
`m`	order of respected serial correlation (m-dependence)
`CP`	set of change points estmated by the multiple filter algorithm, increasingly ordered in time
`rate`	estimated mean rates between adjacent change points
`S`	start of time interval
`E`	end of time interval
`Tt`	length of time interval
`H`	window set
`d`	step size delta at which processes were evaluated
`alpha`	significance level

cutout	states whether cutout was used (see 'Arguments')
perform.CPD	logical, if TRUE change point detection algorithm was performed
tech.var	list of technical variables with processes Phi and G_ht or R_ht
type	type of MFT which was performed: "rate"

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Michael Messer, Marietta Kirchner, Julia Schiemann, Jochen Roeper, Ralph Neininger and Gaby Schneider (2014). A multiple filter test for the detection of rate changes in renewal processes with varying variance. The Annals of Applied Statistics 8(4): 2027-67 <doi:10.1214/14-AOAS782>

Michael Messer, Kaue M. Costa, Jochen Roeper and Gaby Schneider (2017). Multi-scale detection of rate changes in spike trains with weak dependencies. Journal of Computational Neuroscience, 42 (2), 187-201. <doi:10.1007/s10827-016-0635-3>

See Also

MFT.variance, MFT.m_est, plot.MFT, summary.MFT, MFT.mean, MFT.peaks

Examples

```
# Rate change detection in Poisson process
# with three change points (at t = 250, 600 and 680)
set.seed(0)
Phi1 <- runif(rpois(1,lambda=390),0,250)
Phi2 <- runif(rpois(1,lambda=380),250,600)
Phi3 <- runif(rpois(1,lambda=200),600,680)
Phi4 <- runif(rpois(1,lambda=400),680,1000)
Phi  <- sort(c(Phi1,Phi2,Phi3,Phi4))
mft  <- MFT.rate(Phi)
plot(mft)
```

MFT.variance *MFT.variance*

Description

The multiple filter test for variance change detection in point processes on the line.

Usage

```
MFT.variance(Phi, rcp = NULL, autoset.d_H = TRUE, S = NULL,
  E = NULL, d = NULL, H = NULL, alpha = 0.05,
  method = "asymptotic", sim = 10000, Q = NA, perform.CPD = TRUE,
  print.output = TRUE)
```

Arguments

`Phi`	numeric vector of increasing events, input point process
`rcp`	vector, rate CPs of Phi (if MFT for the rates is used: as CP[,1]), default: constant rate
`autoset.d_H`	logical, automatic choice of window size H and step size d
`S`	numeric, start of time interval, default: Smallest multiple of d that lies beyond min(Phi)
`E`	numeric, end of time interval, default: Smallest multiple of d that lies beyond max(Phi), needs E > S
`d`	numeric, > 0, step size delta at which processes are evaluated. d is automatically set if autoset.d_H = TRUE
`H`	vector, window set H, all elements must be increasing ordered multiples of d, the smallest element must be >= d and the largest =< (T/2). H is automatically set if autoset.d_H = TRUE
`alpha`	numeric, in (0,1), significance level
`method`	either "asymptotic", or "fixed", defines how threshold Q is derived, default: "asymptotic". If "asymptotic": Q is derived by simulation of limit process L (Brownian motion); possible set number of simulations (sim). If "fixed": Q may be set manually (Q)
`sim`	integer, > 0, No of simulations of limit process (for approximation of Q), default = 10000
`Q`	numeric, rejection threshold, default: Q is simulated according to sim and alpha
`perform.CPD`	logical, if TRUE change point detection algorithm is performed
`print.output`	logical, if TRUE results are printed to the console

Value

invisible

`M`	test statistic
`varQ`	rejection threshold
`method`	how threshold Q was derived, see 'Arguments' for detailed description
`sim`	number of simulations of the limit process (approximation of Q)
`CP`	set of change points estmated by the multiple filter algorithm, increasingly ordered in time
`var`	estimated variances between adjacent change points

S	start of time interval
E	end of time interval
Tt	length of time interval
H	window set
d	step size delta at which processes were evaluated
alpha	significance level
perform.CPD	logical, if TRUE change point detection algorithm was performed
tech.var	list of technical variables with processes Phi and G_ht
type	type of MFT which was performed: "variance"

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Stefan Albert, Michael Messer, Julia Schiemann, Jochen Roeper and Gaby Schneider (2017) Multi-scale detection of variance changes in renewal processes in the presence of rate change points. Journal of Time Series Analysis, <doi:10.1111/jtsa.12254>

See Also

MFT.rate, plot.MFT, summary.MFT, MFT.mean, MFT.peaks

Examples

```
# Rate and variance change detection in Gamma process
# (rate CPs at t=30 and 37.5, variance CPs at t=37.5 and 52.5)
set.seed(51)
mu <- 0.03; sigma <- 0.01
p1 <- mu^2/sigma^2; lambda1 <- mu/sigma^2
p2 <- (mu*0.5)^2/sigma^2; lambda2 <- (mu*0.5)/sigma^2
p3 <- mu^2/(sigma*1.5)^2; lambda3 <- mu/(sigma*1.5)^2
p4 <- mu^2/(sigma*0.5)^2; lambda4 <- mu/(sigma*0.5)^2
Phi <- cumsum(c(rgamma(1000,p1,lambda1),rgamma(500,p2,lambda2),
rgamma(500,p3,lambda3),rgamma(300,p4,lambda4)))
# rcp  <- MFT.rate(Phi)$CP[,1] # MFT for the rates
rcp <- c(30,37.5) # but here we assume known rate CPs
mft <- MFT.variance(Phi,rcp=rcp) # MFT for the variances
plot(mft)
```

plot.MFT *plot.MFT*

Description

Plot method for class 'mft'.

Usage

```
## S3 method for class 'MFT'
plot(x, col = NULL, ylab1 = NULL, ylab2 = NULL,
  cex.legend = 1.2, cex.diamonds = 1.4, main = TRUE, plot.Q = TRUE,
  plot.M = TRUE, plot.h = TRUE, breaks = NULL, wid = NULL, ...)
```

Arguments

`x`	object of class MFT
`col`	"gray" or vector of colors of length(H). Colors for (G_ht) plot, default: NULL -> rainbow colors from blue to red
`ylab1`	character, ylab for 1. graphic
`ylab2`	character, ylab for 2. graphic
`cex.legend`	numeric, size of annotations in plot
`cex.diamonds`	numeric, size of diamonds that indicate change points
`main`	logical, indicates if title and subtitle are plotted
`plot.Q`	logical, indicates if rejection threshold Q is plotted
`plot.M`	logical, indicates if test statistic M is plotted
`plot.h`	logical, indicates if a legend for the window set H is plotted
`breaks`	integer, >0, number of breaks in rate histogram
`wid`	integer, >0, width of bars in variance histogram
`...`	additional parameters

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Michael Messer, Marietta Kirchner, Julia Schiemann, Jochen Roeper, Ralph Neininger and Gaby Schneider (2014). A multiple filter test for the detection of rate changes in renewal processes with varying variance. The Annals of Applied Statistics 8(4): 2027-67 <doi:10.1214/14-AOAS782>

See Also

`MFT.rate`, `MFT.variance`, `MFT.mean`, `MFT.peaks`, `summary.MFT`

Examples

```
# Rate change detection in Poisson process
# with three change points (at t = 250, 600 and 680)
set.seed(0)
Phi1 <- runif(rpois(1,lambda=390),0,250)
Phi2 <- runif(rpois(1,lambda=380),250,600)
Phi3 <- runif(rpois(1,lambda=200),600,680)
Phi4 <- runif(rpois(1,lambda=400),680,1000)
Phi  <- sort(c(Phi1,Phi2,Phi3,Phi4))
mft  <- MFT.rate(Phi)
plot(mft)
```

---

summary.MFT *summary.MFT*

Description

Summary method for class 'mft'.

Usage

```
## S3 method for class 'MFT'
summary(object, ...)
```

Arguments

object	object of class MFT
...	additional parameters

Author(s)

Michael Messer, Stefan Albert, Solveig Plomer and Gaby Schneider

References

Michael Messer, Marietta Kirchner, Julia Schiemann, Jochen Roeper, Ralph Neininger and Gaby Schneider (2014). A multiple filter test for the detection of rate changes in renewal processes with varying variance. The Annals of Applied Statistics 8(4): 2027-67 <doi:10.1214/14-AOAS782>

See Also

MFT.rate, MFT.variance, MFT.mean, MFT.peaks, plot.MFT

Examples

```
# Rate change detection in Poisson process
# with three change points (at t = 250, 600 and 680)
set.seed(0)
Phi1 <- runif(rpois(1,lambda=390),0,250)
Phi2 <- runif(rpois(1,lambda=380),250,600)
Phi3 <- runif(rpois(1,lambda=200),600,680)
Phi4 <- runif(rpois(1,lambda=400),680,1000)
Phi  <- sort(c(Phi1,Phi2,Phi3,Phi4))
mft  <- MFT.rate(Phi)
summary(mft)
```

Index