Package ‘changepoint’

November 9, 2015

Type Package

Title Methods for Changepoint Detection

Version 2.2

Date 2015-10-23

Maintainer Rebecca Killick <r.killick@lancs.ac.uk>

Description
   Implements various mainstream and specialised changepoint methods for finding single and mul-
   tiple changepoints within data. Many popular non-parametric and frequentist methods are in-
   cluded. The cpt.mean(), cpt.var(), cpt.meanvar() functions should be your first point of call.

Depends R(>= 3.0), methods, stats, zoo

Suggests testthat

License GPL

LazyLoad yes

NeedsCompilation yes

Author Rebecca Killick [aut, cre],
   Kaylea Haynes [aut],
   Idris Eckley [ths, aut],
   Paul Fearnhead [ctb, ths],
   Jamie Lee [ctr]

Repository CRAN

Date/Publication 2015-11-09 18:09:10

R topics documented:

  changepoint-package ........................................ 2
  cpt.mean ..................................................... 3
  cpt.meanvar .................................................. 6
  cpt.var ....................................................... 10
  ftse100 ..................................................... 13
  HC1 ............................................................. 14
  Lai2005fig3 .................................................. 15
changepoint-package

Methods for Changepoint Detection

Description
Implement various mainstream and specialised changepoint methods for finding single and multiple changepoints within data. Many popular non-parametric and frequentist methods are included. Users should start by looking at the documentation for cpt.mean(), cpt.var() and cpt.meanvar().

Details

Package: changepoint
Type: Package
Version: 2.2
Date: 2015-10-23
License: GPL
LazyLoad: yes

Author(s)
Rebecca Killick <r.killick@lancs.ac.uk>, Kaylea Haynes <k.haynes1@lancs.ac.uk> with contributions from Idris A. Eckley <i.eckley@lancs.ac.uk>, Paul Fearnhead <p.fearnhead@lancs.ac.uk>.
Maintainer: Rebecca Killick <r.killick@lancs.ac.uk>

References
 cpt.mean

See Also
cpt.mean,cpt.var,cpt.meanvar

Examples

# change in variance
set.seed(1)
x=c(rnorm(100,0,1),rnorm(100,0,10))
ansvar=cpt.var(x)
plot(ansvar)
print(ansvar) # identifies 1 changepoint at 100

# change in mean
y=c(rnorm(100,0,1),rnorm(100,5,1))
ansmean=cpt.mean(y)
plot(ansmean,cpt.col='blue')
print(ansmean)

# change in mean and variance
z=c(rnorm(100,0,1),rnorm(100,2,10))
ansmeanvar=cpt.meanvar(z)
plot(ansmeanvar,cpt.width=3)
print(ansmeanvar)

cpt.mean Identifying Changes in Mean

Description

Calculates the optimal positioning and (potentially) number of changepoints for data using the user specified method.

Usage

cpt.mean(data,penalty="MBIC",pen.value=0,method="AMOC",Q=5,test.stat="Normal",class=TRUE,param.estimates=TRUE,minseglen=1)

Arguments

data A vector, ts object or matrix containing the data within which you wish to find a changepoint. If data is a matrix, each row is considered a separate dataset.

penalty Choice of "None", "SIC", "BIC", "MBIC", AIC", "Hannan-Quinn", "Asymptotic", "Manual" and "CROPS" penalties. If Manual is specified, the manual penalty is contained in the pen.value parameter. If Asymptotic is specified, the theoretical type I error is contained in the pen.value parameter. If CROPS is specified, the penalty range is contained in the pen.value parameter; note this is a vector of length 2 which contains the minimum and maximum penalty value.
Note CROPS can only be used if the method is "PELT". The predefined penalties listed DO count the changepoint as a parameter, postfix a 0 e.g."SIC0" to NOT count the changepoint as a parameter.

pen.value The theoretical type I error e.g.0.05 when using the Asymptotic penalty. A vector of length 2 (min, max) if using the CROPS penalty. The value of the penalty when using the Manual penalty option - this can be a numeric value or text giving the formula to use. Available variables are, n=length of original data, null=null likelihood, alt=alternative likelihood, tau=proposed changepoint, diffparam=difference in number of alternative and null parameters.

method Choice of "AMOC", "PELT", "SegNeigh" or "BinSeg".

Q The maximum number of changepoints to search for using the "BinSeg" method. The maximum number of segments (number of changepoints + 1) to search for using the "SegNeigh" method.

test.stat The assumed test statistic / distribution of the data. Currently only "Normal" and "CUSUM" supported.

class Logical. If TRUE then an object of class cpt is returned.

param. estimates Logical. If TRUE and class=TRUE then parameter estimates are returned. If FALSE or class=FALSE no parameter estimates are returned.

minseglen Positive integer giving the minimum segment length (no. of observations between changes), default is the minimum allowed by theory.

Details

This function is used to find changes in mean for data using the test statistic specified in the test.stat parameter. The changes are found using the method supplied which can be single changepoint (AMOC) or multiple changepoints using exact (PELT or SegNeigh) or approximate (BinSeg) methods. A changepoint is denoted as the first observation of the new segment / regime.

Value

If class=TRUE then an object of S4 class "cpt" is returned. The slot cpts contains the changepoints that are returned. For class=FALSE the structure is as follows.

If data is a vector (single dataset) then a vector/list is returned depending on the value of method. If data is a matrix (multiple datasets) then a list is returned where each element in the list is either a vector or list depending on the value of method.

If method is AMOC then a vector (one dataset) or matrix (multiple datasets) is returned, the columns are:

cpt The most probable location of a changepoint if a change was identified or NA if no changepoint.

p value The p-value of the identified changepoint.

If method is PELT then a vector is returned containing the changepoint locations for the penalty supplied. This always ends with n. If the penalty is CROPS then a list is returned with elements:
cpt.mean

cpt.out A data frame containing the value of the penalty value where the number of segmentations changes, the number of segmentations and the value of the cost at that penalty value.

changepoints The optimal changepoint for the different penalty values starting with the lowest penalty value.

If method is SegNeigh then a list is returned with elements:

cps Matrix containing the changepoint positions for 1,...,Q changepoints.
op.cpts The optimal changepoint locations for the penalty supplied.
pen Penalty used to find the optimal number of changepoints.
like Value of the -2*log(likelihood ratio) + penalty for the optimal number of changepoints selected.

If method is BinSeg then a list is returned with elements:

cps 2xQ Matrix containing the changepoint positions on the first row and the test statistic on the second row.
op.cpts The optimal changepoint locations for the penalty supplied.
pen Penalty used to find the optimal number of changepoints.

Author(s)
Rebecca Killick

References

See Also
cpt.var,cpt.meanvar,plot-methods cpt
Examples

# Example of a change in mean at 100 in simulated normal data
set.seed(1)
x=c(rnorm(100,0,1),rnorm(100,10,1))
cpt.mean(x,penalty="SIC",method="AMOC",class=FALSE) # returns 100 to show that the null hypothesis
was rejected and the change in mean is at 100
ans=cpt.mean(x,penalty="Asymptotic",pen.value=0.01,method="AMOC")
cpts(ans)# returns 100 to show that the null hypothesis was rejected, the change in mean is at 100
# and we are 99% confident of this result

cpt.mean(x,penalty="Manual",pen.value=0.8,method="AMOC",test.stat="CUSUM")
# returns 101 as the changepoint location

# Example of multiple changes in mean at 50,100,150 in simulated normal data
set.seed(1)
x=c(rnorm(50,0,1),rnorm(50,5,1),rnorm(50,10,1),rnorm(50,3,1))
cpt.mean(x,penalty="Manual",pen.value="2*log(n)",method="BinSeg",Q=5,class=FALSE)
# returns optimal number of changepoints is 3, locations are 50,100,150.

# Example of using the CROPS penalty in data set above
set.seed(1)
x=c(rnorm(50,0,1),rnorm(50,5,1),rnorm(50,10,1),rnorm(50,3,1))
out=cpt.mean(x,pen.value=c(4,1500),penalty="CROPS",method="PELT")
cpts.full(out)# returns 7 segmentations for penalty values between 4 and 1500.
# We find segmentations with 7, 5, 4, 3, 2, 1 and 0 changepoints.
# Note that the empty final row indicates no changepoints.
pen.value.full(out)# gives associated penalty transition points
# CROPS does not give an optimal set of changepoints thus we may wish to explore further
plot(out,diagnostic=TRUE)
# looks like the segmentation with 3 changepoints, 50,100,150 is the most appropriate
plot(out,ncpts=3)

# Example multiple datasets where the first row has multiple changes in mean and the second row has
#no change in mean
set.seed(1)
x=c(rnorm(50,0,1),rnorm(50,5,1),rnorm(50,10,1),rnorm(50,3,1))
y=rnorm(200,0,1)
z=cbind(x,y)
cpt.mean(z,penalty="Asymptotic",pen.value=0.01,method="SegNeigh",Q=5,class=FALSE) # returns list
#that has two elements, the first has 3 changes in mean and variance at 50,100,150 and the second
#has no changes in variance
ans=cpt.mean(z,penalty="Asymptotic",pen.value=0.01,method="PELT")
cpts(ans[[1]]) # same results as for the SegNeigh method.
cpts(ans[[2]]) # same results as for the SegNeigh method.
**Description**

Calculates the optimal positioning and (potentially) number of changepoints for data using the user specified method.

**Usage**

cpt.meanvar(data,penalty="MBIC",pen.value=0,method="AMOC",Q=5,test.stat="Normal",class=TRUE,param.estimates=TRUE,shape=1,minseglen=2)

**Arguments**

data A vector, ts object or matrix containing the data within which you wish to find a changepoint. If data is a matrix, each row is considered a separate dataset.

penalty Choice of "None", "SIC", "BIC", "MBIC", AIC", "Hannan-Quinn", "Asymptotic", "Manual" and "CROPS" penalties. If Manual is specified, the manual penalty is contained in the pen.value parameter. If Asymptotic is specified, the theoretical type I error is contained in the pen.value parameter. If CROPS is specified, the penalty range is contained in the pen.value parameter; note this is a vector of length 2 which contains the minimum and maximum penalty value. Note CROPS can only be used if the method is "PELT". The predefined penalties listed DO count the changepoint as a parameter, postfix a 0 e.g."SIC0" to NOT count the changepoint as a parameter.

pen.value The theoretical type I error e.g.0.05 when using the Asymptotic penalty. A vector of length 2 (min,max) if using the CROPS penalty. The value of the penalty when using the Manual penalty option - this can be a numeric value or text giving the formula to use. Available variables are, n=length of original data, null=null likelihood, alt=alternative likelihood, tau=proposed changepoint, diffparam=difference in number of alternative and null parameters.

method Choice of "AMOC", "PELT", "SegNeigh" or "BinSeg".

Q The maximum number of changepoints to search for using the "BinSeg" method. The maximum number of segments (number of changepoints + 1) to search for using the "SegNeigh" method.

test.stat The assumed test statistic / distribution of the data. Currently only "Normal", "Gamma", "Exponential" and "Poisson" are supported.

class Logical. If TRUE then an object of class cpt is returned.

param.estimates Logical. If TRUE and class=TRUE then parameter estimates are returned. If FALSE or class=FALSE no parameter estimates are returned.

shape Value of the assumed known shape parameter required when test.stat="Gamma".

minseglen Positive integer giving the minimum segment length (no. of observations between changes), default is the minimum allowed by theory.

**Details**

This function is used to find changes in mean and variance for data using the test statistic specified in the test.stat parameter. The changes are found using the method supplied which can be single
changepoint (AMOC) or multiple changepoints using exact (PELT or SegNeigh) or approximate (BinSeg) methods. A changepoint is denoted as the first observation of the new segment / regime.

Value

If class=TRUE then an object of S4 class "cpt" is returned. The slot cpts contains the changepoints that are returned. For class=FALSE the structure is as follows.

If data is a vector (single dataset) then a vector/list is returned depending on the value of method. If data is a matrix (multiple datasets) then a list is returned where each element in the list is either a vector or list depending on the value of method.

If method is AMOC then a vector (one dataset) or matrix (multiple datasets) is returned, the columns are:

- \texttt{cpt}\hspace{2em} The most probable location of a changepoint if a change was identified or NA if no changepoint.
- \texttt{p value}\hspace{2em} The p-value of the identified changepoint.

If method is PELT then a vector is returned containing the changepoint locations for the penalty supplied. This always ends with n. If the penalty is CROPS then a list is returned with elements:

- \texttt{cpt.out}\hspace{2em} A data frame containing the value of the penalty value where the number of segmentations changes, the number of segmentations and the value of the cost at that penalty value.
- \texttt{changepoints}\hspace{2em} The optimal changepoints for the different penalty values starting with the lowest penalty value.

If method is SegNeigh then a list is returned with elements:

- \texttt{cps}\hspace{2em} Matrix containing the changepoint positions for 1,...,Q changepoints.
- \texttt{op.cpts}\hspace{2em} The optimal changepoint locations for the penalty supplied.
- \texttt{pen}\hspace{2em} Penalty used to find the optimal number of changepoints.
- \texttt{like}\hspace{2em} Value of the -2*log(likelihood ratio) + penalty for the optimal number of changepoints selected.

If method is BinSeg then a list is returned with elements:

- \texttt{cps}\hspace{2em} 2xQ Matrix containing the changepoint positions on the first row and the test statistic on the second row.
- \texttt{op.cpts}\hspace{2em} The optimal changepoint locations for the penalty supplied.
- \texttt{pen}\hspace{2em} Penalty used to find the optimal number of changepoints.

Author(s)

Rebecca Killick
References


See Also

cpt.var,cpt.meanvar,plot-methods,cpt

Examples

# Example of a change in scale parameter (mean and variance) at 100 in simulated gamma data
set.seed(1)
x=c(rgamma(100,shape=1,rate=1),rgamma(100,shape=1,rate=5))
cpt.meanvar(x,penalty="SIC",method="AMOC",test.stat="Gamma",class=FALSE,shape=1) # returns 97 to #show that the null hypothesis was rejected and the change in scale parameter is at 97
ans=cpt.meanvar(x,penalty="AIC",method="AMOC",test.stat="Gamma",shape=1)
cpts(ans)
# returns 97 to show that the null hypothesis was rejected, the change in scale parameter is at 97

# Example of multiple changes in mean and variance at 50,100,150 in simulated normal data
set.seed(1)
x=c(rnorm(50,0,1),rnorm(50,5,3),rnorm(50,10,1),rnorm(50,3,10))
cpt.meanvar(x,penalty="Manual",pen.value="4*log(n)",method="BinSeg",Q=5,class=FALSE) # returns optimal number of changepoints is 4, locations are 50,100,150,152.

# Example of using the CROPS penalty in the above example
set.seed(1)
x=c(rnorm(50,0,1),rnorm(50,5,3),rnorm(50,10,1),rnorm(50,3,10))
out=cpt.meanvar(x,pen.value=c(2*log(length(x)),100*log(length(x))),penalty="CROPS",method="PELT")
cpts.full(out)
# returns 6 segmentations for penalty values between 2log(n) and 100log(n).
# We find segmentations with 9, 7, 4, 3, 1 and 0 changepoints.
# Note that the empty final row indicates no changepoints.
pen.value.full(out) # gives associated penalty transition points
# CROPS does not give an optimal set of changepoints thus we may wish to explore further
plot(out,dIagnostic=TRUE)
# looks like the segmentation with 4 changepoints, 50,100,150,200 is the most appropriate
plot(out,ncpts=3)

# Example multiple datasets where the first row has multiple changes in mean and variance and the
#second row has no change in mean or variance
set.seed(1)
x=c(rnorm(50,0,1),rnorm(50,5,3),rnorm(50,10,1),rnorm(50,3,10))
y=rnorm(200,0,1)
z=rbind(x,y)
cpt.meanvar(z,penalty="Asymptotic",pen.value=0.01,method="SegNeigh",Q=5, class=FALSE) # returns list
#that has two elements, the first has 3 changes in mean and variance at 50,100,150 and the second
#has no changes in mean or variance
ans=cpt.meanvar(z,penalty="Asymptotic",pen.value=0.01,method="PELT")
cpts(ans[[1]]) # same results as for the SegNeigh method.
cpts(ans[[2]]) # same results as for the SegNeigh method.

cpt.var

**Identifying Changes in Variance**

**Description**

Calculates the optimal positioning and (potentially) number of changepoints for data using the user specified method.

**Usage**

cpt.var(data,penalty="MBIC",pen.value=0, know.mean=FALSE,mu=NA,method="AMOC",Q=5,
test.stat="Normal",class=TRUE,param. estimates=TRUE,minseglen=2)

**Arguments**

data  
A vector, ts object or matrix containing the data within which you wish to find a changepoint. If data is a matrix, each row is considered a separate dataset.

penalty  
Choice of "None", "SIC", "BIC", "MBIC", AIC", "Hannan-Quinn", "Asymptotic", "Manual" and "CROPS" penalties. If Manual is specified, the manual penalty is contained in the pen.value parameter. If Asymptotic is specified, the theoretical type I error is contained in the pen.value parameter. If CROPS is specified, the penalty range is contained in the pen.value parameter; note this is a vector of length 2 which contains the minimum and maximum penalty value. Note CROPS can only be used if the method is "PELT". The predefined penalties listed DO count the changepoint as a parameter, postfix a 0 e.g."SIC0" to NOT count the changepoint as a parameter.
**pen.value**  The theoretical type I error e.g. 0.05 when using the Asymptotic penalty. A vector of length 2 (min, max) if using the CROPS penalty. The value of the penalty when using the Manual penalty option - this can be a numeric value or text giving the formula to use. Available variables are, n=length of original data, null=null likelihood, alt=alternative likelihood, tau=proposed changepoint, diffparam=difference in number of alternative and null parameters.

**know.mean**  Only required for test.stat="Normal". Logical, if TRUE then the mean is assumed known and mu is taken as its value. If FALSE, and mu=NA (default value) then the mean is estimated via maximum likelihood. If FALSE and the value of mu is supplied, mu is not estimated but is counted as an estimated parameter for decisions.

**mu**  Only required for test.stat="Normal". Numerical value of the true mean of the data. Either single value or vector of length nrow(data). If data is a matrix and mu is a single value, the same mean is used for each row.

**method**  Choice of "AMOC", "PELT", "SegNeigh" or "BinSeg".

**Q**  The maximum number of changepoints to search for using the "BinSeg" method. The maximum number of segments (number of changepoints + 1) to search for using the "SegNeigh" method.

**test.stat**  The assumed test statistic / distribution of the data. Currently only "Normal" and "CSS" supported.

**class**  Logical. If TRUE then an object of class cpt is returned.

**param. estimates**  Logical. If TRUE and class=TRUE then parameter estimates are returned. If FALSE or class=FALSE no parameter estimates are returned.

**minseglen**  Positive integer giving the minimum segment length (no. of observations between changes), default is the minimum allowed by theory.

**Details**

This function is used to find changes in variance for data using the test statistic specified in the test.stat parameter. The changes are found using the method supplied which can be single changepoint (AMOC) or multiple changepoints using exact (PELT or SegNeigh) or approximate (BinSeg) methods. A changepoint is denoted as the first observation of the new segment / regime. Note that for the test.stat="CSS" option the preset penalties are log(.) to allow comparison with test.stat="Normal".

**Value**

If class=TRUE then an object of S4 class "cpt" is returned. The slot cpts contains the changepoints that are returned. For class=FALSE the structure is as follows.

If data is a vector (single dataset) then a vector/list is returned depending on the value of method. If data is a matrix (multiple datasets) then a list is returned where each element in the list is either a vector or list depending on the value of method.

If method is AMOC then a vector (one dataset) or matrix (multiple datasets) is returned, the columns are:
cpt The most probable location of a changepoint if a change was identified or NA if no changepoint.
p value The p-value of the identified changepoint.

If method is PELT then a vector is returned containing the changepoint locations for the penalty supplied. This always ends with n. If the penalty is CROPS then a list is returned with elements:
cpt.out A data frame containing the value of the penalty value where the number of segmentations changes, the number of segmentations and the value of the cost at that penalty value.
segmentations The optimal segmentations for the different penalty values starting with the lowest penalty value

If method is SegNeigh then a list is returned with elements:
cps Matrix containing the changepoint positions for 1,...,Q changepoints.
op.cpts The optimal changepoint locations for the penalty supplied.
pen Penalty used to find the optimal number of changepoints.
like Value of the -2*log(likelihood ratio) + penalty for the optimal number of changepoints selected.

If method is BinSeg then a list is returned with elements:
cps 2xQ Matrix containing the changepoint positions on the first row and the test statistic on the second row.
op.cpts The optimal changepoint locations for the penalty supplied.
pen Penalty used to find the optimal number of changepoints.

Author(s)
Rebecca Killick

References

See Also

cpt.mean,cpt.meanvar,plot-methods,cpt

Examples

# Example of a change in variance at 100 in simulated normal data
set.seed(1)
x=(rnorm(100,0,1),rnorm(100,0,10))
cpt.var(x,penalty="SIC",method="AMOC",class=FALSE) # returns 100 to show that the null hypothesis
# was rejected and the change in variance is at 100
ans=cpt.var(x,penalty="Asymptotic",pen.value=0.01,method="AMOC")
cpts(ans)# returns 100 to show that the null hypothesis was rejected, the change in variance is at
#100 and we are 99% confident of this result

# Example of multiple changes in variance at 50,100,150 in simulated data
set.seed(1)
x=(rnorm(50,0,1),rnorm(50,0,10),rnorm(50,0,5),rnorm(50,0,1))
cpt.var(x,penalty="Manual",pen.value="log(2starlog(n))",method="BinSeg",test.stat="CSS",Q=5,
class=FALSE) # returns optimal number of changepoints is 4, locations are 50,53,99,150.

# Example of using CROPS in the above example
set.seed(1)
x=(rnorm(50,0,1),rnorm(50,0,10),rnorm(50,0,5),rnorm(50,0,1))
out=cpt.var(x,pen.value=c(log(length(x)),100*log(length(x))),penalty="CROPS",method="PELT")
cpts.full(out)# returns 7 segmentations for penalty values between log(n) and 100log(n).
# We find segmentations with 7, 5, 4,3,2,1 and 0 changepoints.
# Note that the empty final row indicates no changepoints.
pen.value.full(out)# gives associated penalty transition points
# CROPS does not give an optimal set of changepoints thus we may wish to explore further
plot(out,diagnostic=TRUE)
# looks like the segmentation with 3 changepoints, 50,100,150 is the most appropriate
plot(out,ncpts=3)

# Example multiple datasets where the first row has multiple changes in variance and the second row
# has no change in variance
set.seed(10)
x=(rnorm(50,0,1),rnorm(50,0,10),rnorm(50,0,5),rnorm(50,0,1))
y=rnorm(200,0,1)
z=rbind(x,y)
cpt.var(z,penalty="Asymptotic",pen.value=0.01,method="SegNeigh",Q=5,class=FALSE) # returns list that
# has two elements, the first has 3 changes in variance at 50,100,149 and the second has no changes
# in variance
ans=cpt.var(z,penalty="Asymptotic",pen.value=0.01,method="PELT")
cpts(ans[[1]]) # same results as for the SegNeigh method.
cpts(ans[[2]]) # same results as for the SegNeigh method.
**HC1**

**Description**

This dataset gives the daily returns \((c_{t+1}/c_t - 1)\) of the UK FTSE 100 index from 2nd April 1984 until the 13th September 2012.

**Usage**

`ftse100`

**Format**

A matrix of dimension 7187 x 2 where the first column is the Date and the second column is the Daily Return.

**Source**

Yahoo! Finance

---

**HC1**

**G+C Content in Human Chromosome 1**

**Description**

This dataset gives the G+C content in 3kb windows along the Human Chromosome from 10Mb to 33Mb (no missing data).

**Usage**

`HC1`

**Format**

A vector of length 23553.

**Source**

Lai2005fig3

Normalized glioblastoma profile for chromosome 13

Description
This dataset is taken from Lai W, Johnson MJ, Kucherlapati R, Park PJ, Bioinformatics, 2005. The paper states that the original source of the data is from Bredel et al. (2005). The data is Chromosome 13 in GBM31.

Usage
Lai2005fig3

Format
A matrix of dimensions 797 x 5. The columns are Spot, CH, POS.start, POS.end, GBM31.

Source
http://compbio.med.harvard.edu/Supplements/Bioinformatics05b/Profiles/Chrom_13_GBM31.xls

Lai2005fig4
Normalized glioblastoma profile for an excerpt of chromosome 7, the EGFR locus.

Description
This dataset is taken from Lai W, Johnson MJ, Kucherlapati R, Park PJ, Bioinformatics, 2005. The paper states that the original source of the data is from Bredel et al. (2005). The data is an excerpt of chromosome 7 in GBM29 from 40 to 65 Mb.

Usage
Lai2005fig4

Format
A matrix of dimensions 193 x 5. The columns are Spot, CH, POS.start, POS.end, GBM31.

Source
http://compbio.med.harvard.edu/Supplements/Bioinformatics05b/Profiles/Chrom_7_from40_to65Mb_GBM29.xls
ncpts  

 Generic Function - ncpts  

Description

Generic function

Usage

ncpts(object)

Arguments

object  Depending on the class of object depends on the method used (and if one exists)

Details

Generic Function

Value

Depends on the class of object, see individual methods

Author(s)

Rebecca Killick

See Also

ncpts-methods

Examples

x = new("cpt") # new cpt object
ncpts(x) # returns the number of changepoints (i.e. length of the cpts slot in x minus 1)
Description

Generic function

Usage

nseg(object)

Arguments

object Depending on the class of object depends on the method used (and if one exists)

Details

Generic Function

Value

Depends on the class of object, see individual methods

Author(s)

Rebecca Killick

See Also

nseg-methods

Examples

x=new("cpt") # new cpt object
nseg(x) # returns the number of segments (i.e. length of the cpts slot)
Description

Generic function

Usage

seg.len(object)

Arguments

object Depending on the class of object depends on the method used (and if one exists)

Details

Generic Function

Value

Depends on the class of object, see individual methods

Author(s)

Rebecca Killick

See Also

seg.len-methods

Examples

x=new("cpt") # new cpt object
seg.len(x) # returns the length of each segment in the data (i.e. no. of obs between changepoints)
wave.c44137  Wave data from buoy c44137

Description
This dataset gives the significant wave heights from buoy c44137 obtained from the Fisheries and Oceans Canada, East Scotian Slop. The data are taken at hourly intervals from January 2005 until September 2012.

Usage
wave.c44137

Format
A vector of length 63651.

Source
Index

+ Topic changepoint
  changepoint-package, 2
+ Topic cpt
  ncpts, 16
  nseg, 17
  seg.len, 18
+ Topic datasets
  ftse100, 13
  HC1, 14
  Lai2005fig3, 15
  Lai2005fig4, 15
  wave.c44137, 19
+ Topic methods
  cpt.mean, 3
  cpt.meanvar, 6
  cpt.var, 10
  ncpts, 16
  nseg, 17
  seg.len, 18
+ Topic models
  cpt.mean, 3
  cpt.meanvar, 6
  cpt.var, 10
+ Topic segmentation
  changepoint-package, 2
+ Topic ts
  cpt.mean, 3
  cpt.meanvar, 6
  cpt.var, 10
+ Topic univar
  cpt.mean, 3
  cpt.meanvar, 6
  cpt.var, 10

changepoint (changepoint-package), 2
changepoint-package, 2
cpt, 5, 9, 13
cpt.mean, 3, 9, 13
cpt.meanvar, 3, 5, 6, 13
cpt.var, 3, 5, 9, 10

ftse100, 13
HC1, 14
Lai2005fig3, 15
Lai2005fig4, 15
ncpts, 16
nseg, 17
seg.len, 18
wave.c44137, 19