Package 'MuChPoint'

October 12, 2022

Type Package

Title Multiple Change Point

Version 0.6.3

Maintainer Brault Vincent <vincent.brault@univ-grenoble-alpes.fr>

Description Nonparametric approach to estimate the location of block boundaries (change-points) of non-overlapping blocks in a random symmetric matrix which consists of random variables whose distribution changes from block to block.
BRAULT Vincent, OUADAH Sarah, SANSONNET Laure and LEVY-LEDUC Celine (2017) <doi:10.1016/j.jmva.2017.12.005>.

Imports Matrix, capushe, shiny, utils, methods, Rcpp

Collate MuChPoint_Class.R RcppExports.R MuChPoint.R

URL https://github.com/Lionning/MuChPoint

BugReports https://github.com/Lionning/MuChPoint/issues

License GPL-2

Encoding UTF-8

RoxygenNote 7.1.2

LinkingTo Rcpp

NeedsCompilation yes

Author Brault Vincent [cre, aut],
Cougoulat Glenn [ctb],
Ouadah Sarah [ctb],
Sansonnet Laure [ctb]

Repository CRAN

Date/Publication 2022-04-08 14:22:31 UTC

R topics documented:

Compute_Cn1n2   *Compute the Delta of the dynamic programming*

Description

Compute the Delta of the dynamic programming in Rcpp

Usage

```
Compute_Cn1n2(x)
```

Arguments

x   the matrix of rank

MuChPoint   *MuChPoint fitting procedure*

Description

Produce a block-wise estimation of a symmetric matrix.

Usage

```
MuChPoint(Y, Lmax = nrow(Y)/2, N = NULL, cores = 1, verbose = TRUE)
```

Arguments

Y	symmetric matrix of observations.
Lmax	a positive integer less than number of columns (and number of rows). By default, nrow(Y)/2.
N	a positive integer vector less than number of columns (and number of rows). N is used when the break-points are known. By default, NULL.
cores	a positive integer giving the number of cores used. If you use windows, the parallelization is impossible. By default, 1.
verbose	logical. To display the progression bars. By default TRUE.

References

Article: BRAULT V., OUADAH S., SANSONNET L. and LEVY-LEDUC C. Nonparametric homogeneity tests and multiple change-point estimation for analyzing large Hi-C data matrices. Journal of Multivariate Analysis, 2017

Examples

```
require(MuChPoint)
mu=c(rep(c(rep(1,25),rep(0,25)),3))%*%t(rep(c(rep(0,25),rep(1,25)),3))
Y=matrix(rnorm(150^2,0,5),150)+mu+t(mu)
Y=as.matrix(Matrix::forceSymmetric(Y))
res=MuChPoint(Y)
plot(res,Y,L=5,shiny=FALSE)
plot(res,Y,L=1:10,shiny=FALSE,ask=FALSE)
```

MuChPoint-class   *Class "MuChPoint"*

Description

Class of object returned by the `MuChPoint` function.

Usage

```
## S4 method for signature 'MuChPoint'
show(object)
```

Arguments

object   an object with class `MuChPoint`

Slots

S a vector object of type `numeric`, giving the values of the statistics $S_n(n_1,...,n_L)$ following the number L.

N a numeric vector with the position of the different break-points.

bt an inferior triangular matrix containing the positions of break-points following the number of break-points (in rows).

References

Article: BRAULT V., OUADAH S., SANSONNET L. and LEVY-LEDUC C. Nonparametric homogeneity tests and multiple change-point estimation for analyzing large Hi-C data matrices. Journal of Multivariate Analysis, 2017

See Also

See also `plot,MuChPoint-method` and `MuChPoint`.

`plot,MuChPoint-method`   *Produce a plot of two-dimensional segmentation of a* `MuChPoint` *fit.*

Description

Produce a plot of two-dimensional segmentation of a `MuChPoint` fit.

Usage

```
## S4 method for signature 'MuChPoint'
plot(x, y, shiny = TRUE, col = "Color", L = NULL, ask = TRUE)
```

Arguments

x	an object of class `MuChPoint`.
y	used for S4 compatibility represented the matrix (typically, the matrix used in the program `MuChPoint`).
shiny	for a representation with a shiny application.
col	for the colors of the representations.
L	the summarized matrix with L break-points (L can be a vector).
ask	If `TRUE`, to hit will be necessary to see next plot.

References

Article: BRAULT V., OUADAH S., SANSONNET L. and LEVY-LEDUC C. Nonparametric homogeneity tests and multiple change-point estimation for analyzing large Hi-C data matrices. Journal of Multivariate Analysis, 2017

See Also

`MuChPoint`, `capushe`.

Examples

```
require(MuChPoint)
mu=c(rep(c(rep(1,25),rep(0,25)),3))%*%t(rep(c(rep(0,25),rep(1,25)),3))
Y=matrix(rnorm(150^2,0,2),150)+mu+t(mu)
Y=as.matrix(Matrix::forceSymmetric(Y))
res=MuChPoint(Y)
plot(res,Y,L=5,shiny=FALSE)
plot(res,Y,L=1:10,shiny=FALSE,ask=FALSE)
```

```
print,MuChPoint-method
```
Print for the class of object returned by the MuChPoint *function.*

Description

Print for the class of object returned by the MuChPoint function.

Usage

```
## S4 method for signature 'MuChPoint'
print(x, N = NULL)
```

Arguments

x	an object with class MuChPoint
N	a numeric between 1 and length(x@N) for the number of break-points desired.

```
summary,MuChPoint-method
```
Summary of a MuChPoint *object.*

Description

Summary of a MuChPoint object.

Usage

```
## S4 method for signature 'MuChPoint'
summary(object)
```

Arguments

object	an object of class MuChPoint.

See Also

[MuChPoint](#).

Examples

```
require(MuChPoint)
mu=c(rep(c(rep(1,25),rep(0,25)),3))%*%t(rep(c(rep(0,25),rep(1,25)),3))
Y=matrix(rnorm(150^2,0,2),150)+mu+t(mu)
Y=as.matrix(Matrix::forceSymmetric(Y))
res=MuChPoint(Y)
summary(res)
```

Index