

Package ‘MSigSeg’

July 21, 2025

Type Package

Title Multiple SIGNAL SEGmentation

Version 0.2.0

Description Traditional methods typically detect breakpoints from individual signals, which means that when applied separately to multiple signals, the breakpoints are not aligned. However, this package implements a common breakpoint detection approach for multiple piecewise constant signals, resulting in increased detection sensitivity and specificity. By employing various techniques, optimal performance is ensured, and computation is accelerated. We hope that this package will be beneficial for researchers in signal processing, bioinformatics, economy, and other related fields. The `segmentation()`, `lambda_estimator()` functions are the main functions of this package.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Collate 'MSigSeg_Class.R' 'allGenerics.R' 'brkps_method.R' 'change.R' 'data.R' 'data.input_method.R' 'data.output_method.R' 'funtion_colsum_colmean.R' 'lambda_estimator.R' 'lambda_method.R' 'multi_plot.R' 'noisegen.R' 'print_method.R' 'seg.len_method.R' 'segmentation.R' 'summary_method.R'

Depends R (>= 4.0.0), methods

Imports MASS, ggpubr, ggplot2

NeedsCompilation no

Author Xuanyu Liu [aut, cre],
Junbo Duan [aut]

Maintainer Xuanyu Liu <lxy382198251@stu.xjtu.edu.cn>

Repository CRAN

Date/Publication 2023-11-13 15:03:21 UTC

Contents

brkps	2
change	3
data.input	3
data.output	4
data_test	5
lambda	5
lambda_estimator	6
MSigSeg-class	6
multi_plot	7
NCHSData	8
noisegen	8
print	9
seg.len	9
segmentation	10
stock	11
summary	11
T16M	12
T16P	12
Index	13

brkps	<i>Generic Function-brkps.</i>
-------	--------------------------------

Description

This function returns the brkps slot of MSigSeg object.

Usage

```
## S4 method for signature 'MSigSeg'
brkps(object)
```

Arguments

object A 'MSigSeg' object.

Details

This function is a S4 method for MSigSeg object. It retrieves brkps slot, which contains the locations of break points.

Value

The brkps slot of MSigSeg object.

Examples

```
x=new("MSigSeg") # Creating a new MSigSeg object.  
brkps(x)
```

change	<i>Breakpoints matrix generation.</i>
--------	---------------------------------------

Description

Generate matrix based on specified breakpoints.

Usage

```
change(M, p = 0.01)
```

Arguments

M	A matrix users aimed to add breakpoints.
p	Probability of occurrence of breakpoints.

Details

Generate matrix with common breakpoints, based on specific probability of occurrence.

Value

A list containing the matrix with specified change point and the location of breakpoints.

data.input	<i>Generic Function-data.input.</i>
------------	-------------------------------------

Description

This function returns the data.input slot of MSigSeg object.

Usage

```
## S4 method for signature 'MSigSeg'  
data.input(object)
```

Arguments

object	A MSigSeg object.
--------	-------------------

Details

This function is a S4 method for MSigSeg object. It retrieves data.input slot, which contains the data users input.

Value

The data.input slot of MSigSeg object.

Examples

```
x=new("MSigSeg") # Creating a new MSigSeg object.  
data.input(x)
```

data.output

Generic Function-data.output.

Description

This function returns the data.output slot of MSigSeg object.

Usage

```
## S4 method for signature 'MSigSeg'  
data.output(object)
```

Arguments

object A MSigSeg object.

Details

This function is a S4 method for MSigSeg object. It retrieves data.output slot, which contains the input data which has been smoothed..

Value

The data.output slot of MSigSeg object.

Examples

```
x=new("MSigSeg") # Creating a new MSigSeg object.  
data.output(x)
```

data_test	<i>A simulated data set used for testing.</i>
-----------	---

Description

A simulated data set used for testing.

Usage

```
data_test
```

Format

A matrix with 1000 rows and 20 columns.

Examples

```
data("data_test", package = "MSigSeg")
```

lambda	<i>Generic Function-lambda.</i>
--------	---------------------------------

Description

This function returns the lambda slot of MSigSeg object.

Usage

```
## S4 method for signature 'MSigSeg'  
lambda(object)
```

Arguments

object	A MSigSeg object.
--------	-------------------

Details

This function is a S4 method for MSigSeg object. It retrieves lambda slot, which contains penalty coefficient to prevent over fitting.

Value

The lambda slot of MSigSeg object.

Examples

```
x=new("MSigSeg") # Creating a new MSigSeg object.  
lambda(x)
```

lambda_estimator	<i>Detecting common breakpoints with designated number.</i>
------------------	---

Description

Automatic estimation of penalty parameter lambda for user defined breakpoints number.

Usage

```
lambda_estimator(Y, K)
```

Arguments

Y	An data.frame/matrix containing the data to be segmented. Each column stores a signal.
K	Number of change points users want to detect.

Details

This function is based on the segmentation() function. Number of breakpoints are defined by users and lambda is calculated by algorithm automatically.

Value

An object of S4 class "MSigSeg".

Examples

```
data(data_test)
lambda_estimator(data_test,5)
```

MSigSeg-class	<i>An S4 class to encapsulation the result of breakpoints analysis.</i>
---------------	---

Description

An S4 class to encapsulation the result of breakpoints analysis.

Slots

`data.input` An data.frame/matrix containing the data to be segmented. Each column stores a signal.

`data.output` A matrix containing the input data which has been smoothed.

`lambda` A penalty term, small value leads to large number of breakpoints, and vice versa.

`brkps` A vector containing the locations of common breakpoints.

`fmin` A numeric containing the optimal numerical value calculated.

`date` Character string containing date information.

multi_plot	<i>Plot function of MSigSeg package.</i>
------------	--

Description

Graph signals and breakpoints based on ggplot2 and ggarange packages.

Usage

```
multi_plot(m, ncol, nrow)
```

Arguments

<code>m</code>	An object of S4 class "MSigSeg".
<code>ncol</code>	Column numbers of signals arrangement in the graph.
<code>nrow</code>	Row numbers of signals arrangement in the graph

Value

A list, first item in the list is a graphic objects with all signals drawn and second is a list with individual signals.

Examples

```
data(data_test)
m <- segmentation(data_test,100)
p <- multi_plot(m,4,5)
```

NCHSData	<i>influenza data set from CDC used as an example.</i>
----------	--

Description

influenza data set from CDC used as an example.

Usage

NCHSData

Format

A matrix with 52 rows and 10 columns.

Examples

```
data("NCHSData",package = "MSigSeg")
```

noisegen	<i>Noisegen.</i>
----------	------------------

Description

Generate matrix based on signal-to-noise ratio.

Usage

```
noisegen(X, SNR)
```

Arguments

- X A matrix users aimed to add signal-to-noise ratio.
- SNR Signal-to-noise ratio.

Value

A matrix with specified signal-to-noise ratio.

print	<i>Generic Function-print.</i>
-------	--------------------------------

Description

This function print the basic information of MSigSeg object.

Usage

```
## S4 method for signature 'MSigSeg'  
print(object)
```

Arguments

object A MSigSeg object.

Details

This function is a S4 method for MSigSeg object. It prints class, slots, created date and summary of MSigSeg object.

Value

The the basic information of MSigSeg object.

Examples

```
x=new("MSigSeg") # Creating a new MSigSeg object.  
print(x)
```

seg.len	<i>Generic Function-seg.len.</i>
---------	----------------------------------

Description

This function returns the length of segmentation.

Usage

```
## S4 method for signature 'MSigSeg'  
seg.len(object)
```

Arguments

object A MSigSeg object.

Details

This function is a S4 method for MSigSeg object. It calculates the distance between each change points.

Value

A vector contains length of segmentation.

Examples

```
x=new("MSigSeg") # Creating a new MSigSeg object.
seg.len(x)
```

segmentation

Detecting common change points for multiple signals.

Description

Calculates the optimal positioning and number of common breakpoints for multiple signals.

Usage

```
segmentation(Y, lambda, flag = TRUE, return_smooth_signals = TRUE)
```

Arguments

Y	An data.frame/matrix containing the data to be segmented. Each column stores a signal.
lambda	A penalty term, small value leads to large number of breakpoints, and vice versa.
flag	Logical. If True then use th PELT method. If False then use the OP method.
return_smooth_signals	Logical. If True then smoothed signals are returned.

Details

This function uses modified PELT method to find optimal common change points for multiple signals.

Value

An object of S4 class "MSigSeg"

Examples

```
data(data_test)
segmentation(data_test,100)
```

stock	<i>A stock data set used as an example.</i>
-------	---

Description

A stock data set used as an example.

Usage

```
stock
```

Format

A data.frame with 757 rows and 488 columns.

Examples

```
data("stock",package = "MSigSeg")
```

summary	<i>Generic Function-summary.</i>
---------	----------------------------------

Description

This function summarize the information of MSigSeg object.

Usage

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

Arguments

object	A MSigSeg object.
--------	-------------------

Details

This function is a S4 method for MSigSeg object. It summarizes the number of signals, length of signals, number of change points and fmin.

Value

A summary of MSigSeg object.

Examples

```
x=new("MSigSeg") # Creating a new MSigSeg object.  
summary(x)
```

T16M

A chromosome sequencing data set used as an example.

Description

A chromosome sequencing data set used as an example.

Usage

T16M

Format

A data.frame with 2928 rows and 22 columns.

References

Navin N, Kendall J, Troge J, et al. Tumour evolution inferred by single-cell sequencing. *Nature*. 2011;472(7341):90-94. doi:10.1038/nature09807

Examples

```
data("T16M", package = "MSigSeg")
```

T16P

A chromosome sequencing data set used as an example.

Description

A chromosome sequencing data set used as an example.

Usage

T16P

Format

A data.frame with 2928 rows and 16 columns.

References

Navin N, Kendall J, Troge J, et al. Tumour evolution inferred by single-cell sequencing. *Nature*. 2011;472(7341):90-94. doi:10.1038/nature09807

Examples

```
data("T16P", package = "MSigSeg")
```

Index

* datasets

data_test, [5](#)

NCHSData, [8](#)

stock, [11](#)

T16M, [12](#)

T16P, [12](#)

brkps, [2](#)

brkps,MSigSeg-method (brkps), [2](#)

change, [3](#)

data.input, [3](#)

data.input,MSigSeg-method (data.input),
[3](#)

data.output, [4](#)

data.output,MSigSeg-method
(data.output), [4](#)

data_test, [5](#)

lambda, [5](#)

lambda,MSigSeg-method (lambda), [5](#)

lambda_estimator, [6](#)

MSigSeg-class, [6](#)

multi_plot, [7](#)

NCHSData, [8](#)

noisegen, [8](#)

print, [9](#)

print,MSigSeg-method (print), [9](#)

seg.len, [9](#)

seg.len,MSigSeg-method (seg.len), [9](#)

segmentation, [10](#)

stock, [11](#)

summary, [11](#)

summary,MSigSeg-method (summary), [11](#)

T16M, [12](#)

T16P, [12](#)