

Package: IDetect (via r-universe)

September 11, 2024

Type Package

Title Isolate-Detect Methodology for Multiple Change-Point Detection

Version 0.1.0

Depends R (>= 3.3.0)

Imports splines

Description Provides efficient implementation of the Isolate-Detect methodology for the consistent estimation of the number and location of multiple change-points in one-dimensional data sequences from the ``deterministic + noise" model. For details on the Isolate-Detect methodology, please see Anastasiou and Fryzlewicz (2018) https://docs.wixstatic.com/ugd/24cdcc_6a0866c574654163b8255e272bc0001b.pdf. Currently implemented scenarios are: piecewise-constant signal with Gaussian noise, piecewise-constant signal with heavy-tailed noise, continuous piecewise-linear signal with Gaussian noise, continuous piecewise-linear signal with heavy-tailed noise.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests testthat

NeedsCompilation no

Author Andreas Anastasiou [aut, cre], Piotr Fryzlewicz [aut]

Maintainer Andreas Anastasiou <a.anastasiou@lse.ac.uk>

Repository CRAN

Date/Publication 2018-03-09 10:02:07 UTC

Contents

cplm_ic	2
---------------	---

cplm_th	4
est_signal	5
ht_ID_cplm	7
ht_ID_pcm	8
ID	10
IDetect	13
ID_cplm	13
ID_pcm	15
normalise	17
pcm_ic	18
pcm_th	19
resid_ID	21
sol_path_cplm	22
sol_path_pcm	23
s_e_points	24
win_cplm_th	25
win_pcm_th	26

Index **29**

cplm_ic	<i>Multiple change-point detection in a continuous piecewise-linear signal via minimising an information criterion</i>
---------	--

Description

This function performs the Isolate-Detect methodology based on an information criterion approach, in order to detect multiple change-points in a noisy, continuous, piecewise-linear data sequence, with the noise being Gaussian. More information on how this approach works as well as the relevant literature reference are given in Details.

Usage

```
cplm_ic(x, th_const = 1.25, Kmax = 200, penalty = c("ssic_pen",
"sic_pen"), points = 10)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
th_const	A positive real number with default value equal to 1.25. It is used to define the threshold value that will be used at the first step of the model selection based Isolate-Detect method; see Details for more information.
Kmax	A positive integer with default value equal to 200. It is the maximum allowed number of estimated change-points in the solution path; see sol_path_cplm for more details.
penalty	A character vector with names of penalty functions used.

`points` A positive integer with default value equal to 10. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Details

The approach followed in `cplm_ic` in order to detect the change-points is based on identifying the set of change-points that minimise an information criterion. At first, we employ `sol_path_cplm`, which overestimates the number of change-points using `th_const` in order to define the threshold and then sorts the obtained estimates in a way that the estimate, which is most likely to be correct appears first, whereas the least likely to be correct, appears last. Let J be the number of estimates that this overestimation approach returns. We will obtain a vector $b = (b_1, b_2, \dots, b_J)$, with the estimates ordered as explained above. We define the collection $\{M_j\}_{j=0,1,\dots,J}$, where M_0 is the empty set and $M_j = \{b_1, b_2, \dots, b_j\}$. Among the collection of models $M_j, j = 0, 1, \dots, J$, we select the one that minimises a predefined Information Criterion. The obtained set of change-points is apparently a subset of the solution path given in `sol_path_cplm`. More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A list with the following components:

`sol_path` A vector containing the solution path.
`ic_curve` A list with values of the chosen information criteria.
`cpt_ic` A list with the change-points detected for each information criterion considered.
`no_cpt_ic` The number of change-points detected for each information criterion considered.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

`ID_cplm` and `ID`, which employ this function. In addition, see `pcm_ic` for the case of detecting changes in a piecewise-constant signal using the information criterion based approach.

Examples

```
single.cpt <- c(seq(0, 999, 1), seq(998.5, 499, -0.5))
single.cpt.noise <- single.cpt + rnorm(2000)
cpt.single.ic <- cplm_ic(single.cpt.noise)

three.cpt <- c(seq(0, 499, 1), seq(498.5, 249, -0.5), seq(250,1249,2), seq(1248,749,-1))
three.cpt.noise <- three.cpt + rnorm(2000)
cpt.three.ic <- cplm_ic(three.cpt.noise)
```

cplm_th *Multiple change-point detection in a continuous, piecewise-linear signal via thresholding*

Description

This function performs the Isolate-Detect methodology with the thresholding-based stopping rule in order to detect multiple change-points in a continuous, piecewise-linear noisy data sequence, with noise that is Gaussian. See Details for a brief explanation of the Isolate-Detect methodology (with the relevant reference) and of the thresholding-based stopping rule.

Usage

```
cplm_th(x, sigma = stats::mad(diff(diff(x)))/sqrt(6), thr_const = 1.4,
        thr_fin = sigma * thr_const * sqrt(2 * log(length(x))), s = 1,
        e = length(x), points = 3, k_l = 1, k_r = 1)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
sigma	A positive real number. It is the estimate of the standard deviation of the noise in x. The default value is $\text{mad}(\text{diff}(\text{diff}(x)))/\sqrt{6}$, where $\text{mad}(x)$ denotes the median absolute deviation of x computed under the assumption that the noise is independent and identically distributed from the Gaussian distribution.
thr_const	A positive real number with default value equal to 1.4. It is used to define the threshold; see thr_fin.
thr_fin	With T the length of the data sequence, this is a positive real number with default value equal to $\text{sigma} * \text{thr_const} * \sqrt{2 * \log(T)}$. It is the threshold, which is used in the detection process.
s, e	Positive integers with s less than e, which indicate that you want to check for change-points in the data sequence with subscripts in [s, e]. The default values are s equal to 1 and e equal to T, with T the length of the data sequence.
points	A positive integer with default value equal to 3. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively; see Details for more information.
k_l, k_r	Positive integer numbers that get updated whenever the function calls itself during the detection process. They are not essential for the function to work, and we include them only to reduce the computational time.

Details

The change-point detection algorithm that is used in `cplm_th` is the Isolate-Detect methodology described in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint. The concept is simple and is split into two stages; firstly, isolation of each of the true change-points in subintervals of the data domain, and secondly their

detection. ID first creates two ordered sets of $K = \lceil T/\text{points} \rceil$ right- and left-expanding intervals as follows. The j^{th} right-expanding interval is $R_j = [1, j \times \text{points}]$, while the j^{th} left-expanding interval is $L_j = [T - j \times \text{points} + 1, T]$. We collect these intervals in the ordered set $S_{RL} = \{R_1, L_1, R_2, L_2, \dots, R_K, L_K\}$. For a suitably chosen contrast function, ID first identifies the point with the maximum contrast value in R_1 . If its value exceeds a certain threshold, then it is taken as a change-point. If not, then the process tests the next interval in S_{RL} and repeats the above process. Upon detection, the algorithm makes a new start from estimated location.

Value

A numeric vector with the detected change-points.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[win_cplm_th](#), [ID_cplm](#), and [ID](#), which employ this function. In addition, see [pcm_th](#) for the case of detecting changes in a piecewise-constant signal via thresholding.

Examples

```
single.cpt <- c(seq(0, 999, 1), seq(998.5, 499, -0.5))
single.cpt.noise <- single.cpt + rnorm(2000)
cpt.single.th <- cplm_th(single.cpt.noise)

three.cpt <- c(seq(0, 499, 1), seq(498.5, 249, -0.5), seq(251,1249,2), seq(1248,749,-1))
three.cpt.noise <- three.cpt + rnorm(2000)
cpt.three.th <- cplm_th(three.cpt.noise)

multi.cpt <- rep(c(seq(0,49,1), seq(48,0,-1)),20)
multi.cpt.noise <- multi.cpt + rnorm(1980)
cpt.multi.th <- cplm_th(multi.cpt.noise)
```

est_signal

Estimate the signal

Description

This function estimates the signal in a given data sequence x with change-points at cpt . The type of the signal depends on whether the change-points represent changes in a piecewise-constant or continuous, piecewise-linear signal. For more information see Details below.

Usage

```
est_signal(x, cpt, type = c("mean", "slope"))
```

Arguments

x	A numeric vector containing the given data.
cpt	A positive integer vector with the locations of the change-points. If missing, the <code>ID_pcm</code> or the <code>ID_cp1m</code> function (depending on the type of the signal) is called internally to extract the change-points in x.
type	A character string, which defines the type of the detected change-points. If <code>type = "mean"</code> , then the change-points represent the locations of changes in the mean of a piecewise-constant signal. If <code>type = "slope"</code> , then the change-points represent the locations of changes in the slope of a continuous, piecewise-linear signal.

Details

The data points provided in x are assumed to follow

$$X_t = f_t + \sigma \epsilon_t; t = 1, 2, \dots, T,$$

where T is the total length of the data sequence, X_t are the observed data, f_t is a one-dimensional, deterministic signal with abrupt structural changes at certain points, and ϵ_t is white noise. We denote by r_1, r_2, \dots, r_N the elements in `cpt` and by $r_0 = 0$ and $r_{N+1} = T$. Depending on the value that has been passed to `type`, the returned value is calculated as follows.

- For `type = "mean"`, in each segment (r_j+1, r_{j+1}) , f_t for $t \in (r_j+1, r_{j+1})$ is approximated by the mean of X_t calculated over $t \in (r_j+1, r_{j+1})$.
- For `type = "slope"`, f_t is approximated by the linear spline fit with knots at r_1, r_2, \dots, r_N minimising the l_2 distance between the fit and the data.

Value

A numeric vector with the estimated signal.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

Examples

```
single.cpt.pcm <- c(rep(4,1000),rep(0,1000))
single.cpt.pcm.noise <- single.cpt.pcm + rnorm(2000)
cpt.single.pcm <- ID_pcm(single.cpt.pcm.noise)
fit.cpt.single.pcm <- est_signal(single.cpt.pcm.noise, cpt.single.pcm$cpt, type = "mean")

three.cpt.pcm <- c(rep(4,500),rep(0,500),rep(-4,500),rep(1,500))
three.cpt.pcm.noise <- three.cpt.pcm + rnorm(2000)
cpt.three.pcm <- ID_pcm(three.cpt.pcm.noise)
fit.cpt.three.pcm <- est_signal(three.cpt.pcm.noise, cpt.three.pcm$pcm, type = "mean")

single.cpt.plm <- c(seq(0,999,1),seq(998.5,499,-0.5))
single.cpt.plm.noise <- single.cpt.plm + rnorm(2000)
cpt.single.plm <- ID_cp1m(single.cpt.plm.noise)
fit.cpt.single.plm <- est_signal(single.cpt.plm.noise, cpt.single.plm$cpt, type = "slope")
```

ht_ID_cplm	<i>Apply the Isolate-Detect methodology for multiple change-point detection in a continuous, piecewise-linear vector with non Gaussian noise</i>
------------	--

Description

Using the Isolate-Detect methodology, this function estimates the number and locations of multiple change-points in the noisy, continuous, piecewise-linear input vector x , with noise that is not normally distributed. It also gives the estimated signal, as well as the solution path defined in [sol_path_cplm](#) (see Details for the relevant literature reference).

Usage

```
ht_ID_cplm(x, s.ht = 3, q.ht = 300, ht_thr_id = 1.4, ht_th_ic_id = 1.25,
           p_thr = 1, p_ic = 3)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
s.ht	A positive integer number with default value equal to 3. It is used to define the way we pre-average the given data sequence. For more information see Details.
q.ht	A positive integer number with default value equal to 300. If the length of x is less than or equal to $q.ht$, then no pre-averaging will take place.
ht_thr_id	A positive real number with default value equal to 1.4. It is used to define the threshold, if the thresholding approach (described in cplm_th) is to be followed.
ht_th_ic_id	A positive real number with default value equal to 1.25. It is useful only if the model selection based Isolate-Detect method is to be followed and it is used to define the threshold value that will be used at the first step (change-point overestimation) of the model selection approach described in cplm_ic . It is applied to the new data, which are obtained after we take average values on x .
p_thr	A positive integer with default value equal to 1. It is used only when the threshold based approach (described in cplm_th) is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.
p_ic	A positive integer with default value equal to 3. It is used only when the information criterion based approach (described in cplm_ic) is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Details

Firstly, in this function we call [normalise](#), in order to create a new data sequence, \tilde{x} , by taking averages of observations in x . Then, we employ [ID_cplm](#) on \tilde{x}_q to obtain the change-points, namely

$\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_{\hat{N}}$ in increasing order. To obtain the original location of the change-points with, on average, the highest accuracy we define

$$\hat{r}_k = (\tilde{r}_k - 1) * s.ht + \lfloor s.ht/2 + 0.5 \rfloor, k = 1, 2, \dots, \hat{N}.$$

More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A list with the following components:

cpt	A vector with the detected change-points.
no_cpt	The number of change-points detected.
fit	A numeric vector with the estimated continuous piecewise-linear signal.
solution_path	A vector containing the solution path.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[ID_cp1m](#) and [normalise](#), which are functions that are used in [ht_ID_cp1m](#). In addition, see [ht_ID_pcm](#) for the case of piecewise-constant mean signals.

Examples

```
single.cpt <- c(seq(0, 1999, 1), seq(1998, -1, -1))
single.cpt.student <- single.cpt + rt(4000, df = 5)
cpt.single <- ht_ID_cp1m(single.cpt.student)

three.cpt <- c(seq(0, 3998, 2), seq(3996, -2, -2), seq(0, 3998, 2), seq(3996, -2, -2))
three.cpt.student <- three.cpt + rt(8000, df = 5)
cpt.three <- ht_ID_cp1m(three.cpt.student)
```

ht_ID_pcm	<i>Apply the Isolate-Detect methodology for multiple change-point detection in the mean of a vector with non Gaussian noise</i>
-----------	---

Description

Using the Isolate-Detect methodology, this function estimates the number and locations of multiple change-points in the mean of the noisy, piecewise-constant input vector x , with noise that is not normally distributed. It also gives the estimated signal, as well as the solution path defined in [sol_path_pcm](#). See Details for the relevant literature reference.

Usage

```
ht_ID_pcm(x, s.ht = 3, q.ht = 300, ht_thr_id = 1, ht_th_ic_id = 0.9,
          p_thr = 1, p_ic = 3)
```

Arguments

<code>x</code>	A numeric vector containing the data in which you would like to find change-points.
<code>s.ht</code>	A positive integer number with default value equal to 3. It is used to define the way we pre-average the given data sequence (see Details).
<code>q.ht</code>	A positive integer number with default value equal to 300. If the length of <code>x</code> is less than or equal to <code>q.ht</code> , then no pre-averaging will take place.
<code>ht_thr_id</code>	A positive real number with default value equal to 1. It is used to define the threshold, if the thresholding approach is to be followed; see pcm_th for more details on the thresholding approach.
<code>ht_th_ic_id</code>	A positive real number with default value equal to 0.9. It is useful only if the model selection based Isolate-Detect method is to be followed and it is used to define the threshold value that will be used at the first step (change-point over-estimation) of the model selection approach described in pcm_ic . It is applied to the new data, which are obtained after we pre-average <code>x</code> .
<code>p_thr</code>	A positive integer with default value equal to 1. It is used only when the threshold based approach (as described in pcm_th) is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.
<code>p_ic</code>	A positive integer with default value equal to 3. It is used only when the information criterion based approach (described in pcm_ic) is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Details

Firstly, in this function we call [normalise](#), in order to create a new data sequence, \tilde{x} , by taking averages of observations in `x`. Then, we employ [ID_pcm](#) on \tilde{x}_q to obtain the change-points, namely $\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_{\hat{N}}$ in increasing order. To obtain the original location of the change-points with, on average, the highest accuracy we define $\hat{r}_k = (\tilde{r}_k - 1) * s.ht + \lfloor s.ht/2 + 0.5 \rfloor, k = 1, 2, \dots, \hat{N}$. More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A list with the following components:

<code>cpt</code>	A vector with the detected change-points.
<code>no_cpt</code>	The number of change-points detected.
<code>fit</code>	A numeric vector with the estimated piecewise-constant signal.

`solution_path` A vector containing the solution path.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[ID_pcm](#) and [normalise](#), which are functions that are used in [ht_ID_pcm](#). In addition, see [ht_ID_cplm](#) for the case of continuous and piecewise-linear signals.

Examples

```
single.cpt <- c(rep(4,3000),rep(0,3000))
single.cpt.student <- single.cpt + rt(6000, df = 5)
cpts_detect <- ht_ID_pcm(single.cpt.student)

three.cpt <- c(rep(4,2000),rep(0,2000),rep(-4,2000),rep(0,2000))
three.cpt.student <- three.cpt + rt(8000, df = 5)
cpts_detect_three <- ht_ID_pcm(three.cpt.student)
```

ID

Multiple change-point detection in piecewise-constant or continuous, piecewise-linear signals using the Isolate-Detect methodology

Description

This is the main, general function of the package. It employs more specialised functions in order to estimate the number and locations of multiple change-points in the noisy, piecewise-constant or continuous, piecewise-linear input vector `xd`. The noise can either follow the Gaussian distribution or not. The approach that is followed is a hybrid between the thresholding approach (explained in [pcm_th](#) and [cplm_th](#)) and the information criterion approach (explained in [pcm_ic](#) and [cplm_ic](#)) and estimates the change-points taking into account both these approaches. Further to the number and the location of the estimated change-points, `ID`, returns the estimated signal, as well as the solution path. For more information and the relevant literature reference, see [Details](#).

Usage

```
ID(xd, th.cons = 1, th.cons_lin = 1.4, th.ic = 0.9, th.ic.lin = 1.25,
    lambda = 3, lambda.ic = 10, contrast = c("mean", "slope"), ht = FALSE,
    scale = 3)
```

Arguments

<code>xd</code>	A numeric vector containing the data in which you would like to find change-points.
<code>th.cons</code>	A positive real number with default value equal to 1. It is used to define the threshold, if the thresholding approach (explained in pcm_th) is to be followed to detect the change-points in the scenario of piecewise-constant signals.

th.cons_lin	A positive real number with default value equal to 1.4. It is used to define the threshold, if the thresholding approach (explained in cplm_th) is to be followed to detect the change-points in the scenario of continuous, piecewise-linear signals.
th.ic	A positive real number with default value equal to 0.9. It is useful only if the model selection based Isolate-Detect method (described in pcm_ic) is to be followed for the scenario of piecewise-constant signals. It is used to define the threshold value that will be used at the first step (change-point overestimation) of the model selection approach.
th.ic.lin	A positive real number with default value equal to 1.25. It is useful only if the model selection based Isolate-Detect method (described in cplm_ic) is to be followed for the scenario of continuous, piecewise-linear signals. It is used to define the threshold value that will be used at the first step (change-point overestimation) of the model selection approach.
lambda	A positive integer with default value equal to 3. It is used only when the threshold based approach is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.
lambda.ic	A positive integer with default value equal to 10. It is used only when the information criterion based approach is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.
contrast	A character string, which defines the type of the contrast function to be used in the Isolate-Detect algorithm. If contrast = ``mean'`, then the algorithm looks for changes in a piecewise-constant signal. If contrast = ``slope'`, then the algorithm looks for changes in a continuous, piecewise-linear signal.
ht	A logical variable with default value equal to FALSE. If FALSE, the noise is assumed to follow the Gaussian distribution. If TRUE, then the noise is assumed to follow a distribution that has tails heavier than those of the Gaussian distribution.
scale	A positive integer number with default value equal to 3. It is used to define the way we pre-average the given data sequence only if ht = TRUE. See the Details in ht_ID_pcm for more information on how we pre-average.

Details

The data points provided in `xd` are assumed to follow

$$X_t = f_t + \sigma \epsilon_t; t = 1, 2, \dots, T,$$

where T is the total length of the data sequence, X_t are the observed data, f_t is a one-dimensional, deterministic signal with abrupt structural changes at certain points, and ϵ_t are independent and identically distributed random variables with mean zero and variance one. In this function, the following scenarios for f_t are implemented.

- Piecewise-constant signal with Gaussian noise.
Use contrast = ``mean'` and ht = FALSE here.

- Piecewise-constant signal with heavy-tailed noise.
Use `contrast = ``mean``` and `ht = TRUE` here.
- Continuous, piecewise-linear signal with Gaussian noise.
Use `contrast = ``slope``` and `ht = FALSE` here.
- Continuous, piecewise-linear signal with heavy-tailed noise.
Use `contrast = ``slope``` and `ht = TRUE` here.

In the case where `ht = FALSE`: the function firstly detects the change-points using `win_pcm_th` (for the case of piecewise-constant signal) or `win_cplm_th` (for the case of continuous, piecewise-linear signal). If the estimated number of change-points is greater than 100, then the result is returned and we stop. Otherwise, `ID` proceeds to detect the change-points using `pcm_ic` (for the case of piecewise-constant signal) or `cplm_ic` (for the case of continuous, piecewise-linear signal) and this is what is returned.

In the case where `ht = TRUE`: First we pre-average the given data sequence using `normalise` and then, on the obtained data sequence, we follow exactly the same procedure as the one when `ht = FALSE` above.

More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A list with the following components:

<code>cpt</code>	A vector with the detected change-points.
<code>no_cpt</code>	The number of change-points detected.
<code>fit</code>	A numeric vector with the estimated signal.
<code>solution_path</code>	A vector containing the solution path.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

`ID_pcm`, `ID_cplm`, `ht_ID_pcm`, and `ht_ID_cplm`, which are the functions that are employed in `ID`, depending on which scenario is imposed by the input arguments.

Examples

```
single.cpt.mean <- c(rep(4,3000),rep(0,3000))
single.cpt.mean.normal <- single.cpt.mean + rnorm(6000)
single.cpt.mean.student <- single.cpt.mean + rt(6000, df = 5)
cpt.single.mean.normal <- ID(single.cpt.mean.normal)
cpt.single.mean.student <- ID(single.cpt.mean.student, ht = TRUE)

single.cpt.slope <- c(seq(0, 1999, 1), seq(1998, -1, -1))
single.cpt.slope.normal <- single.cpt.slope + rnorm(4000)
single.cpt.slope.student <- single.cpt.slope + rt(4000, df = 5)
cpt.single.slope.normal <- ID(single.cpt.slope.normal, contrast = "slope")
cpt.single.slope.student <- ID(single.cpt.slope.student, contrast = "slope", ht = TRUE)
```

IDetect	<i>IDetect: Multiple generalised change-point detection using the Isolate-Detect methodology</i>
---------	--

Description

The IDetect package implements the Isolate-Detect methodology for multiple generalised change-point detection, or sequence segmentation, in one-dimensional data following the “deterministic signal + noise” model. The different structures that are implemented are: piecewise-constant signal with Gaussian noise, piecewise-constant signal with heavy tailed noise, piecewise-linear and continuous signal with Gaussian noise, and piecewise-linear and continuous signal with heavy-tailed noise. The main routine of the package is [ID](#).

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>, Piotr Fryzlewicz, <p.fryzlewicz@lse.ac.uk>

References

“Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

See Also

[ID](#), [ID_pcm](#), [ID_cplm](#), [ht_ID_pcm](#), and [ht_ID_cplm](#).

Examples

#See Examples for ID.

ID_cplm	<i>Multiple change-point detection for a continuous, piecewise-linear signal using the Isolate-Detect methodology</i>
---------	---

Description

This function estimates the number and locations of multiple change-points in the noisy, continuous and piecewise-linear input vector x , using the Isolate-Detect methodology. The noise follows the normal distribution. The estimated signal, as well as the solution path defined in [sol_path_cplm](#) are also given. The function is a hybrid between the thresholding approach of [win_cplm_th](#) and the information criterion approach of [cplm_ic](#) and estimates the change-points taking into account both these approaches (see Details for more information and the relevant literature reference).

Usage

```
ID_cplm(x, thr_id = 1.4, th_ic_id = 1.25, pointsth = 3, pointsic = 10)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
thr_id	A positive real number with default value equal to 1.4. It is used to define the threshold, if the thresholding approach is to be followed; see cplm_th for more details.
th_ic_id	A positive real number with default value equal to 1.25. It is useful only if the model selection based Isolate-Detect method is to be followed and it is used to define the threshold value that will be used at the first step (change-point overestimation) of the model selection approach described in cplm_ic .
pointsth	A positive integer with default value equal to 3. It is used only when the threshold based approach is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.
pointsic	A positive integer with default value equal to 10. It is used only when the information criterion based approach is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Details

Firstly, this function detects the change-points using [win_cplm_th](#). If the estimated number of change-points is larger than 100, then the result is returned and we stop. Otherwise, [ID_cplm](#) proceeds to detect the change-points using [cplm_ic](#) and this is what is returned. To sum up, [ID_cplm](#) returns a result based on [cplm_ic](#) if the estimated number of change-points is less than 100. Otherwise, the result comes from thresholding. More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A list with the following components:

cpt	A vector with the detected change-points.
no_cpt	The number of change-points detected.
fit	A numeric vector with the estimated continuous piecewise-linear signal.
solution_path	A vector containing the solution path.

Author(s)

Andreas Anastasiou, <a.anastasiou@else.ac.uk>

See Also

[win_cplm_th](#) and [cplm_ic](#) which are the functions that [ID_cplm](#) is based on. In addition, see [ID_pcm](#) for the case of detecting changes in the mean of a piecewise-constant signal. The main function [ID](#) of the package employs [ID_cplm](#).

Examples

```

single.cpt <- c(seq(0, 999, 1), seq(998.5, 499, -0.5))
single.cpt.noise <- single.cpt + rnorm(2000)
cpt.single <- ID_cplm(single.cpt.noise)

three.cpt <- c(seq(0, 499, 1), seq(498.5, 249, -0.5), seq(250,1249,2), seq(1248,749,-1))
three.cpt.noise <- three.cpt + rnorm(2000)
cpt.three <- ID_cplm(three.cpt.noise)

multi.cpt <- rep(c(seq(0,49,1), seq(48,0,-1)),20)
multi.cpt.noise <- multi.cpt + rnorm(1980)
cpt.multi <- ID_cplm(multi.cpt.noise)

```

ID_pcm	<i>Multiple change-point detection in the mean of a vector using the Isolate-Detect methodology</i>
--------	---

Description

This function estimates the number and locations of multiple change-points in the mean of the noisy piecewise-constant input vector x , using the Isolate-Detect methodology. The noise is Gaussian. The estimated signal, as well as the solution path defined in [sol_path_pcm](#) are also given. The function is a hybrid between the thresholding approach of [win_pcm_th](#) and the information criterion approach of [pcm_ic](#) and estimates the change-points taking into account both these approaches (see Details for more information and the relevant literature reference).

Usage

```
ID_pcm(x, thr_id = 1, th_ic_id = 0.9, pointsth = 3, pointsic = 10)
```

Arguments

<code>x</code>	A numeric vector containing the data in which you would like to find change-points.
<code>thr_id</code>	A positive real number with default value equal to 1. It is used to define the threshold, if the thresholding approach is to be followed; see pcm_th for more details.
<code>th_ic_id</code>	A positive real number with default value equal to 0.9. It is useful only if the model selection based Isolate-Detect method is to be followed. It is used to define the threshold value that will be used at the first step (change-point over-estimation) of the model selection approach described in pcm_ic .
<code>pointsth</code>	A positive integer with default value equal to 3. It is used only when the threshold based approach is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

`pointsic` A positive integer with default value equal to 10. It is used only when the information criterion based approach is to be followed and it defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Details

Firstly, this function detects the change-points using `win_pcm_th`. If the estimated number of change-points is larger than 100, then the result is returned and we stop. Otherwise, `ID_pcm` proceeds to detect the change-points using `pcm_ic` and this is what is returned. To sum up, `ID_pcm` returns a result based on `pcm_ic` if the estimated number of change-points is less than 100. Otherwise, the result comes from thresholding. More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A list with the following components:

<code>cpt</code>	A vector with the detected change-points.
<code>no_cpt</code>	The number of change-points detected.
<code>fit</code>	A numeric vector with the estimated piecewise-constant signal.
<code>solution_path</code>	A vector containing the solution path.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

`win_pcm_th` and `pcm_ic` which are the functions that `ID_pcm` is based on. In addition, see `ID_cp1m` for the case of detecting changes in a continuous, piecewise-linear signal. The main function `ID` of the package employs `ID_pcm`.

Examples

```
single.cpt <- c(rep(4,1000),rep(0,1000))
single.cpt.noise <- single.cpt + rnorm(2000)
cpts_detect <- ID_pcm(single.cpt.noise)

three.cpt <- c(rep(4,500),rep(0,500),rep(-4,500),rep(1,500))
three.cpt.noise <- three.cpt + rnorm(2000)
cpts_detect_three <- ID_pcm(three.cpt.noise)

multi.cpt <- rep(c(rep(0,50),rep(3,50)),20)
multi.cpt.noise <- multi.cpt + rnorm(2000)
cpts_detect_multi <- ID_pcm(multi.cpt.noise)
```

normalise

Transform the noise to be closer to the Gaussian distribution

Description

This function pre-processes the given data in order to obtain a noise structure that is closer to satisfying the Gaussianity assumption. See details for more information and for the relevant literature reference.

Usage

```
normalise(x, sc = 3)
```

Arguments

x A numeric vector containing the data.

sc A positive integer number with default value equal to 3. It is used to define the way we pre-average the given data sequence.

Details

For a given natural number sc and data x of length T , let us denote by $Q = \lceil T/sc \rceil$. Then, `normalise` calculates

$$\tilde{x}_q = 1/sc \sum_{t=(q-1)*sc+1}^{q*sc} x_t,$$

for $q = 1, 2, \dots, Q - 1$, while

$$\tilde{x}_Q = (T - (Q - 1) * sc)^{-1} \sum_{t=(Q-1)*sc+1}^T x_t.$$

More details can be found in the preprint “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018).

Value

The “normalised” vector \tilde{x} of length Q , as explained in Details.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[ht_ID_pcm](#) and [ht_ID_cplm](#), which are functions that employ `normalise`.

Examples

```
t5 <- rt(n = 10000, df = 5)
n5 <- normalise(t5, sc = 3)
```

pcm_ic	<i>Multiple change-point detection in the mean via minimising an information criterion</i>
--------	--

Description

This function performs the Isolate-Detect methodology based on an information criterion approach, in order to detect multiple change-points in the mean of a noisy data sequence, with the noise following the Gaussian distribution. More information on how this approach works as well as the relevant literature reference are given in Details.

Usage

```
pcm_ic(x, th_const = 0.9, Kmax = 200, penalty = c("ssic_pen", "sic_pen"),
       points = 10)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
th_const	A positive real number with default value equal to 0.9. It is used to define the threshold value that will be used at the first step of the model selection based Isolate-Detect method; see Details for more information.
Kmax	A positive integer with default value equal to 200. It is the maximum allowed number of estimated change-points in the solution path algorithm, described in Details below.
penalty	A character vector with names of the penalty functions used.
points	A positive integer with default value equal to 10. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Details

The approach followed in `pcm_ic` in order to detect the change-points is based on identifying the set of change-points that minimise an information criterion. At first, we employ `sol_path_pcm`, which overestimates the number of change-points using `th_const` in order to define the threshold, and then sorts the obtained estimates in a way that the estimate, which is most likely to be correct appears first, whereas the least likely to be correct, appears last. Let J be the number of estimates that this overestimation approach returns. We will obtain a vector $b = (b_1, b_2, \dots, b_J)$, with the estimates ordered as explained above. We define the collection $\{M_j\}_{j=0,1,\dots,J}$, where M_0 is the empty set and $M_j = \{b_1, b_2, \dots, b_j\}$. Among the collection of models $M_j, j = 0, 1, \dots, J$, we select the one that minimises a predefined Information Criterion. The obtained set of change-points

is apparently a subset of the solution path given in [sol_path_pcm](#). More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A list with the following components:

`sol_path` A vector containing the solution path.
`ic_curve` A list with values of the chosen information criteria.
`cpt_ic` A list with the change-points detected for each information criterion considered.
`no_cpt_ic` The number of change-points detected for each information criterion considered.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[ID_pcm](#) and [ID](#), which employ this function. In addition, see [cplm_ic](#) for the case of detecting changes in a continuous, piecewise-linear signal using the information criterion based approach.

Examples

```
single.cpt <- c(rep(4,1000),rep(0,1000))
single.cpt.noise <- single.cpt + rnorm(2000)
cpt.single.ic <- pcm_ic(single.cpt.noise)

three.cpt <- c(rep(4,500),rep(0,500),rep(-4,500),rep(1,500))
three.cpt.noise <- three.cpt + rnorm(2000)
cpt.three.ic <- pcm_ic(three.cpt.noise)
```

pcm_th

Multiple change-point detection in the mean via thresholding

Description

This function performs the Isolate-Detect methodology (see Details for the relevant literature reference) with the thresholding-based stopping rule in order to detect multiple change-points in the mean of a noisy input vector x , with Gaussian noise. See Details for a brief explanation of the Isolate-Detect methodology, and of the thresholding-based stopping rule.

Usage

```
pcm_th(x, sigma = stats::mad(diff(x)/sqrt(2)), thr_const = 1,
      thr_fin = sigma * thr_const * sqrt(2 * log(length(x))), s = 1,
      e = length(x), points = 3, k_l = 1, k_r = 1)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
sigma	A positive real number. It is the estimate of the standard deviation of the noise in x. The default value is the median absolute deviation of x computed under the assumption that the noise is independent and identically distributed from the Gaussian distribution.
thr_const	A positive real number with default value equal to 1. It is used to define the threshold; see thr_fin.
thr_fin	With T the length of the data sequence, this is a positive real number with default value equal to $\text{sigma} * \text{thr_const} * \sqrt{2 * \log(T)}$. It is the threshold, which is used in the detection process.
s, e	Positive integers with s less than e, which indicate that you want to check for change-points in the data sequence with subscripts in [s, e]. The default values are s equal to 1 and e equal to T, with T the length of the data sequence.
points	A positive integer with default value equal to 3. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively; see Details for more information.
k_l, k_r	Positive integer numbers that get updated whenever the function calls itself during the detection process. They are not essential for the function to work, and we include them only to reduce the computational time.

Details

The change-point detection algorithm that is used in [pcm_th](#) is the Isolate-Detect methodology described in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint. The concept is simple and is split into two stages; firstly, isolation of each of the true change-points in subintervals of the data domain, and secondly their detection. ID first creates two ordered sets of $K = \lceil T/\text{points} \rceil$ right- and left-expanding intervals as follows. The j^{th} right-expanding interval is $R_j = [1, j \times \text{points}]$, while the j^{th} left-expanding interval is $L_j = [T - j \times \text{points} + 1, T]$. We collect these intervals in the ordered set $S_{RL} = \{R_1, L_1, R_2, L_2, \dots, R_K, L_K\}$. For a suitably chosen contrast function, ID first identifies the point with the maximum contrast value in R_1 . If its value exceeds a certain threshold, then it is taken as a change-point. If not, then the process tests the next interval in S_{RL} and repeats the above process. Upon detection, the algorithm makes a new start from estimated location.

Value

A numeric vector with the detected change-points.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[win_pcm_th](#), [ID_pcm](#), and [ID](#), which employ this function. In addition, see [cplm_th](#) for the case of detecting changes in a continuous, piecewise-linear signal via thresholding.

Examples

```

single.cpt <- c(rep(4,1000),rep(0,1000))
single.cpt.noise <- single.cpt + rnorm(2000)
cpt.single.th <- pcm_th(single.cpt.noise)

three.cpt <- c(rep(4,500),rep(0,500),rep(-4,500),rep(1,500))
three.cpt.noise <- three.cpt + rnorm(2000)
cpt.three.th <- pcm_th(three.cpt.noise)

multi.cpt <- rep(c(rep(0,50),rep(3,50)),20)
multi.cpt.noise <- multi.cpt + rnorm(2000)
cpt.multi.th <- pcm_th(multi.cpt.noise)

```

resid_ID

*Calculate the residuals related to the estimated signal***Description**

This function returns the difference between x and the estimated signal with change-points at `cpt`. The input in the argument `type_chg` will indicate the type of changes in the signal.

Usage

```
resid_ID(x, cpt, type_chg = c("mean", "slope"), type_res = c("raw",
  "standardised"))
```

Arguments

<code>x</code>	A numeric vector containing the data.
<code>cpt</code>	A positive integer vector with the locations of the change-points. If missing, the <code>ID</code> function is called internally to detect any change-points that might be present in x .
<code>type_chg</code>	A character string, which defines the type of the detected change-points. If <code>type_chg = "mean"</code> , then the change-points represent the locations of changes in the mean of a piecewise-constant signal. If <code>type_chg = "slope"</code> , then the change-points represent the locations of changes in the slope of a piecewise-linear and continuous signal.
<code>type_res</code>	A choice of "raw" and "standardised" residuals.

Value

If `type_res = "raw"`, the function returns the difference between the data and the estimated signal. If `type_res = "standardised"`, then the function returns the difference between the data and the estimated signal, divided by the estimated standard deviation.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

Examples

```

single.cpt.pcm <- c(rep(4,1000),rep(0,1000))
single.cpt.pcm.noise <- single.cpt.pcm + rnorm(2000)
cpt_detect <- ID(single.cpt.pcm.noise, contrast = "mean")

residuals_cpt_raw <- resid_ID(single.cpt.pcm.noise, cpt = cpt_detect$cpt, type_chg = "mean",
type_res = "raw")

residuals_cpt_stand. <- resid_ID(single.cpt.pcm.noise, cpt = cpt_detect$cpt, type_chg = "mean",
type_res = "standardised")

plot(residuals_cpt_raw)
plot(residuals_cpt_stand.)

```

sol_path_cplm

The solution path for the case of continuous piecewise-linear signals

Description

This function starts by over-estimating the number of true change-points. After that, following an approach based on the values of a suitable contrast function, it sorts the estimated change-points in a way that the estimation, which is most-likely to be correct appears first, whereas the least likely to be correct, appears last. The routine is typically not called directly by the user; it is employed in [cplm_ic](#). For more details, see References.

Usage

```
sol_path_cplm(x, thr_ic = 1.25, points = 3)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
thr_ic	A positive real number with default value equal to 1.25. It is used to define the threshold. The change-points are estimated by thresholding with threshold equal to $\sigma * thr_ic * \sqrt{2 * \log(T)}$, where T is the length of the data sequence x and $\sigma = \text{mad}(\text{diff}(\text{diff}(x)))/6$. Because, we would like to overestimate the number of the true change-points in x, it is suggested to keep thr_ic smaller than 1.4, which is the default value used as the threshold constant in the function win_cplm_th .
points	A positive integer with default value equal to 3. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Value

The solution path for the case of continuous piecewise-linear signals.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

References

Anastasiou, A. and Fryzlewicz, P. (2018). Detecting multiple generalized change-points by isolating single ones.

Examples

```
three.cpt <- c(seq(0, 499, 1), seq(498.5, 249, -0.5), seq(250.5, 999, 1.5), seq(998, 499, -1))
three.cpt.noise <- three.cpt + rnorm(2000)
solution.path <- sol_path_cplm(three.cpt.noise)
```

sol_path_pcm

The solution path for the case of piecewise-constant signals

Description

This function starts by overestimating the number of true change-points. After that, following a CUSUM-based approach, it sorts the estimated change-points in a way that the estimate, which is most-likely to be correct appears first, whereas the least likely to be correct, appears last. The routine is typically not called directly by the user; it is employed in [pcm_ic](#). For more information, see References.

Usage

```
sol_path_pcm(x, thr_ic = 0.9, points = 3)
```

Arguments

x	A numeric vector containing the data in which you would like to find change-points.
thr_ic	A positive real number with default value equal to 0.9. It is used to define the threshold. The change-points are estimated by thresholding with threshold equal to $\sigma * thr_ic * \sqrt{2 * \log(T)}$, where T is the length of the data sequence x and $\sigma = \text{mad}(\text{diff}(x)/\sqrt{2})$. Because we would like to overestimate the number of true change-points in x, it is suggested to keep thr_ic smaller than 1, which is the default value used as the threshold constant in the function pcm_th .
points	A positive integer with default value equal to 3. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.

Value

The solution path for the case of piecewise-constant signals.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

References

Anastasiou, A. and Fryzlewicz, P. (2018). Detecting multiple generalized change-points by isolating single ones.

Examples

```
three.cpt <- c(rep(4,4000),rep(0,4000),rep(-4,4000),rep(1,4000))
three.cpt.noise <- three.cpt + rnorm(16000)
solution.path <- sol_path_pcm(three.cpt.noise)
```

s_e_points

Derives a subset of integers from a given set

Description

This function finds two subsets of integers in a given interval $[s, e]$. The routine is typically not called directly by the user; its result is used in order to construct the expanding intervals, where the Isolate-Detect method is going to be applied. For more details on how the Isolate-Detect methodology works, see References.

Usage

```
s_e_points(r, l, s, e)
```

Arguments

r	A positive integer vector containing the set, from which the end-points of the expanding intervals are to be chosen.
l	A positive integer vector containing the set, from which the start-points of the expanding intervals are to be chosen.
s	A positive integer indicating the starting position, in the sense that we will choose the elements from r and l that are greater than s.
e	A positive integer indicating the finishing position, in the sense that we will choose the elements from r and l that are less than e.

Value

e_points A vector containing the points that will be used as end-points, in order to create the left-expanding intervals. It consists of the input e and all the elements in the input vector r that are in (s, e) .

s_points A vector containing the points that will be used as start-points, in order to create the left-expanding intervals. It consists of the input s and all the elements in the input vector l that are in (s, e) .

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

References

Anastasiou, A. and Fryzlewicz, P. (2018). Detecting multiple generalized change-points by isolating single ones.

Examples

```
s_e_points(r = seq(10,1000,10), l = seq(991,1,-10), s=435, e = 786)
s_e_points(r = seq(3,100,3), l = seq(98,1,-3), s=43, e = 86)
```

win_cplm_th	<i>A windows-based approach for multiple change-point detection in a continuous, piecewise-linear signal via thresholding</i>
-------------	---

Description

This function performs the windows-based variant of the Isolate-Detect methodology with the thresholding-based stopping rule in order to detect multiple change-points in a continuous, piecewise-linear noisy data sequence, with the noise being Gaussian. It is particularly helpful for very long data sequences, as due to applying Isolate-Detect on moving windows, the computational time is reduced. See Details for a brief explanation of this approach and for the relevant literature reference.

Usage

```
win_cplm_th(xd, sigma = stats::mad(diff(diff(xd)))/sqrt(6), thr_con = 1.4,
  c_win = 3000, w_points = 3, l_win = 12000)
```

Arguments

xd	A numeric vector containing the data in which you would like to find change-points.
sigma	A positive real number. It is the estimate of the standard deviation of the noise in xd. The default value is $\text{mad}(\text{diff}(\text{diff}(xd)))/\sqrt{6}$, where $\text{mad}(xd)$ denotes the median absolute deviation of xd computed under the assumption that the noise is independent and identically distributed from the Gaussian distribution.
thr_con	A positive real number with default value equal to 1.4. It is used to define the threshold. The change-points are estimated by thresholding with threshold equal to $\text{sigma} * \text{thr_con} * \sqrt{2 * \log(T)}$, where T is the length of the data sequence xd.
c_win	A positive integer with default value equal to 3000. It is the length of each window for the data sequence in hand. Isolate-Detect will be applied in segments of the form $[(i-1) * c_win + 1, i * c_win]$, for $i = 1, 2, \dots, K$, where K depends on the length T of the data sequence.

w_points	A positive integer with default value equal to 3. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.
l_win	A positive integer with default value equal to 12000. If the length of the data sequence is less than or equal to l_win, then the windows-based approach will not be applied and the result will be obtained by the classical Isolate-Detect methodology based on thresholding.

Details

The method that is implemented by this function is based on splitting the given data sequence uniformly into smaller parts (windows), to which Isolate-Detect, based on the thresholding stopping rule (see [cplm_th](#)), is then applied.

Value

A numeric vector with the detected change-points.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[cplm_th](#), which is the function that [win_cplm_th](#) is based on. Also, see [ID_cplm](#) and [ID](#), which employ [win_cplm_th](#). In addition, see [win_pcm_th](#) for the case of detecting changes in a piecewise-constant signal via thresholding.

Examples

```
single.cpt <- c(seq(0, 999, 1), seq(998.5, 499, -0.5))
single.cpt.noise <- single.cpt + rnorm(2000)
cpt.single.th <- win_cplm_th(single.cpt.noise)

three.cpt <- c(seq(0, 3999, 1), seq(3998.5, 1999, -0.5), seq(2001, 9999, 2), seq(9998, 5999, -1))
three.cpt.noise <- three.cpt + rnorm(16000)
cpt.three.th <- win_cplm_th(three.cpt.noise)
```

win_pcm_th	<i>A windows-based approach for multiple change-point detection in the mean via thresholding</i>
------------	--

Description

This function performs the windows-based variant of the Isolate-Detect methodology with the thresholding-based stopping rule in order to detect multiple change-points in the mean of a noisy data sequence, with noise that is Gaussian. It is particularly helpful for very long data sequences, as due to applying Isolate-Detect on moving windows, the computational time is reduced. See Details for a brief explanation of this approach and for the relevant literature reference.

Usage

```
win_pcm_th(xd, sigma = stats::mad(diff(xd)/sqrt(2)), thr_con = 1,  
           c_win = 3000, w_points = 3, l_win = 12000)
```

Arguments

xd	A numeric vector containing the data in which you would like to find change-points.
sigma	A positive real number. It is the estimate of the standard deviation of the noise in xd. The default value is the median absolute deviation of xd computed under the assumption that the noise is independent and identically distributed from the Gaussian distribution.
thr_con	A positive real number with default value equal to 1. It is used to define the threshold, which is equal to $\text{sigma} * \text{thr_con} * \sqrt{2 * \log(T)}$, where T is the length of the data sequence xd.
c_win	A positive integer with default value equal to 3000. It is the length of each window for the data sequence in hand. Isolate-Detect will be applied in segments of the form $[(i-1) * c_win + 1, i * c_win]$, for $i = 1, 2, \dots, K$, where K depends on the length T of the data sequence.
w_points	A positive integer with default value equal to 3. It defines the distance between two consecutive end- or start-points of the right- or left-expanding intervals, respectively.
l_win	A positive integer with default value equal to 12000. If the length of the data sequence is less than or equal to l_win, then the windows-based approach will not be applied and the result will be obtained by the classical Isolate-Detect methodology based on thresholding.

Details

The method that is implemented by this function is based on splitting the given data sequence uniformly into smaller parts (windows), to which Isolate-Detect, based on the threshold stopping rule (see [pcm_th](#)), is then applied. An idea of the computational improvement that this structure offers over the classical Isolate-Detect in the case of large data sequences is given in the supplement of “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2018), preprint.

Value

A numeric vector with the detected change-points.

Author(s)

Andreas Anastasiou, <a.anastasiou@lse.ac.uk>

See Also

[pcm_th](#), which is the function that [win_pcm_th](#) is based on. Also, see [ID_pcm](#) and [ID](#), which employ [win_pcm_th](#). In addition, see [win_cplm_th](#) for the case of detecting changes in the slope of a piecewise-linear and continuous signal via thresholding.

Examples

```
single.cpt <- c(rep(4,1000),rep(0,1000))
single.cpt.noise <- single.cpt + rnorm(2000)
cpt.single.th <- win_pcm_th(single.cpt.noise)

three.cpt <- c(rep(4,4000),rep(0,4000),rep(-4,4000),rep(1,4000))
three.cpt.noise <- three.cpt + rnorm(16000)
cpt.three.th <- win_pcm_th(three.cpt.noise)
```

Index

`cplm_ic`, [2](#), [3](#), [7](#), [10–14](#), [19](#), [22](#)
`cplm_th`, [4](#), [4](#), [7](#), [10](#), [11](#), [14](#), [20](#), [26](#)

`est_signal`, [5](#)

`ht_ID_cplm`, [7](#), [8](#), [10](#), [12](#), [13](#), [17](#)
`ht_ID_pcm`, [8](#), [8](#), [10–13](#), [17](#)

`ID`, [3](#), [5](#), [10](#), [10](#), [12–14](#), [16](#), [19–21](#), [26](#), [28](#)
`ID_cplm`, [3](#), [5–8](#), [12](#), [13](#), [13](#), [14](#), [16](#), [26](#)
`ID_pcm`, [6](#), [9](#), [10](#), [12–14](#), [15](#), [16](#), [19](#), [20](#), [28](#)
`IDetect`, [13](#)
`IDetect-package (IDetect)`, [13](#)

`normalise`, [7–10](#), [12](#), [17](#), [17](#)

`pcm_ic`, [3](#), [9–12](#), [15](#), [16](#), [18](#), [18](#), [23](#)
`pcm_th`, [5](#), [9](#), [10](#), [15](#), [19](#), [20](#), [23](#), [27](#), [28](#)

`resid_ID`, [21](#)

`s_e_points`, [24](#)
`sol_path_cplm`, [2](#), [3](#), [7](#), [13](#), [22](#)
`sol_path_pcm`, [8](#), [15](#), [18](#), [19](#), [23](#)

`win_cplm_th`, [5](#), [12–14](#), [22](#), [25](#), [26](#), [28](#)
`win_pcm_th`, [12](#), [15](#), [16](#), [20](#), [26](#), [26](#), [28](#)