# Package: IDetect (via r-universe)

June 7, 2026

**Type** Package

**Title** Isolate-Detect Method for Multiple Change-Point Detection

**Version** 0.1.1

**Depends** R (>= 3.4.2)

**Imports** splines

**Description** The IDetect provides efficient implementation of the ID methodology for the consistent estimation of the number and location of multiple change-points in one-dimensional data sequences from the `deterministic + noise' model. Currently implemented scenarios are: piecewise-constant signal, piecewise-constant signal with a heavy-tailed noise, continuous piecewise-linear signal, continuous piecewise-linear signal with a heavy-tailed noise.

**License** GPL-3

**Encoding** UTF-8

**Suggests** testthat (>= 3.0.0)

**Config/roxygen2/version** 8.0.0

**Config/testthat/edition** 3

**NeedsCompilation** no

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

**Maintainer** Andreas Anastasiou <anastasiou.andreas@ucy.ac.cy>

**Repository** https://anastasiou-andreas.r-universe.dev

**Date/Publication** 2026-05-07 15:07:48 UTC

**RemoteUrl** https://github.com/cran/IDetect

**RemoteRef** HEAD

**RemoteSha** c1ed44b7b3f2d00e0e5c55c2b664a6031f604591

# Contents

IDetect-package . . . . . 2
cpt_ic_pcm . . . . . 3
cpt_ic_plm . . . . . 4
cumsum_lin . . . . . 6
cusum_function . . . . . 7
cusum_one . . . . . 8
est_signal . . . . . 9
ht_ID_pcm . . . . . 10
ht_ID_plm . . . . . 12
ID . . . . . 14
ID_pcm . . . . . 17
ID_plm . . . . . 18
linear_contr_one . . . . . 20
log_lik_slope . . . . . 21
normalise . . . . . 22
pcm_th . . . . . 23
plm_th . . . . . 24
resid . . . . . 26
s_e_points . . . . . 27
sic_pen . . . . . 28
sol_path_pcm . . . . . 29
sol_path_plm . . . . . 30
ssic_pen . . . . . 31
wind_pcm_th . . . . . 32
wind_plm_th . . . . . 33

**Index** **36**

---

`IDetect-package` *IDetect: Multiple generalised change-point detection using the Isolate-Detect methodology*

---

### Description

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

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### References

"Detecting multiple generalized change-points by isolating single ones", Anastasiou and Fryzlewicz (2017), preprint.

### See Also

ID, ID_pcm, ID_plm, ht_ID_pcm, and ht_ID_plm.

### Examples

```
#See Examples for ID.
```

---

## cpt_ic_pcm — *Multiple change-point detection in the mean via minimising an information criterion*

---

### 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 given data sequence. The relevant literature reference is given in details.

### Usage

```
cpt_ic_pcm(
  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. |
| Kmax | A positive integer with default value equal to 200. It defines the maximum number of change-points allowed to be detected. In addition, it is the maximum allowed number of estimated change-points in the solution path. |
| 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 cpt_ic_pcm in order to detect the change-points is based on identifying the set of change-point that minimise an information criterion. The obtained set of change-points is a subset of the solution path, which is given by sol_path_pcm. More details can be found in "Detecting multiple generalized change-points by isolating single ones", Anastasiou and Fryzlewicz (2017), 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, <anastasiou.andreas@ucy.ac.cy>

### See Also

ID_pcm and ID, which employ this function. In addition, see cpt_ic_plm for the case of detecting changes in the slope of a piecewise-linear and continuous 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 <- cpt_ic_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)
cpt.three.ic <- cpt_ic_pcm(three.cpt.noise)
```

---

## cpt_ic_plm — *Multiple change-point detection in the slope of a continuous piecewise-linear mean signal via minimising an information criterion*

---

### Description

This function performs the Isolate-Detect methodology based on an information criterion approach, in order to detect multiple change-points in the slope of a given data sequence. The relevant literature reference is given in details.

## Usage

```
cpt_ic_plm(
  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. |
| Kmax | A positive integer with default value equal to 200. It defines the maximum number of change-points allowed to be detected. In addition, it is the maximum allowed number of estimated change-points in the solution path. |
| 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 `cpt_ic_plm` in order to detect the change-points is based on identifying the set of change-point that minimise an information criterion. The obtained set of change-points is a subset of the solution path, which is given by `sol_path_plm`. More details can be found in "Detecting multiple generalized change-points by isolating single ones", Anastasiou and Fryzlewicz (2017), 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, <anastasiou.andreas@ucy.ac.cy>

### See Also

ID_plm and ID, which employ this function. In addition, see cpt_ic_pcm for the case of detecting changes in the mean of 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 <- cpt_ic_plm(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 <- cpt_ic_plm(three.cpt.noise)
```

---

## cumsum_lin — *Calculate the contrast function that is used in continuous piecewise-linear mean signals*

---

### Description

This function returns the values of the contrast function, which is used for for change-point detection in continuous piecewise-linear mean signals. See Details for more information.

### Usage

```
cumsum_lin(x)
```

### Arguments

| | |
|---|---|
| x | A numeric vector containing the data. |

### Details

The mathematical expression of the result returned by `cumsum_lin` is rather large. Therefore, for the exact formula please see the relevant subsection for piecewise-linearity in the preprint "Detecting multiple generalized change-points by isolating single ones", Anastasiou and Fryzlewicz (2017).

### Value

A numeric vector with the contrast function values at $b = 1, 2, ..., T - 1$, where $T$ is the length of `x`. Note that due to the structure of the signal (piecewise-linear mean), the value of the contrast function statistic at $b = 1$ is equal to zero.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

`cusum_function` for the calculation of the CUSUM statistic, which is the contrast function used in the case of piecewise-constant mean signals.

### Examples

```
no.cpt.noise <- rnorm(2000)
cf.no.cpt <- IDetect:::cumsum_lin(no.cpt.noise)

single.cpt <- c(seq(0, 999, 1), seq(998.5, 499, -0.5))
single.cpt.noise <- single.cpt + rnorm(2000)
cf.single.cpt <- IDetect:::cumsum_lin(single.cpt.noise)
#*** Notice that the maximum in absolute value of \code{csm.single.cpt}
#*** occurs in a neighbourhood of the true change-point, which is 1000.
which.max(abs(cf.single.cpt))
```

---

## `cusum_function` *Calculate the CUMSUM statistic*

---

### Description

This function returns the CUMSUM statistic for a given data sequence. See Details for more information.

### Usage

```
cusum_function(x)
```

### Arguments

| | |
|---|---|
| `x` | A numeric vector containing the data. |

### Details

The CUSUM statistic for `x` at a location $b$ is defined as

$$\tilde{X}_{s,e}^{b} = \sqrt{\frac{e-b}{n(b-s+1)}}\sum_{t=s}^{b} X_t - \sqrt{\frac{b-s+1}{n(e-b)}}\sum_{t=b+1}^{e} X_t,$$

where $1 \leq s \leq b < e \leq T$ and $n = e - s + 1$. In `cusum_function`, we have $s = 1, e = T$.

### Value

A numeric vector with the CUSUM statistic values at $b = 1, 2, ..., T-1$, where $T$ is the length of `x`.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

cumsum_lin for the calculation of the contrast function that is used in the case of piecewise-linear mean signals.

### Examples

```
no.cpt.noise <- rnorm(2000)
csm.no.cpt <- IDetect:::cusum_function(no.cpt.noise)

single.cpt <- c(rep(4,1000),rep(0,1000))
single.cpt.noise <- single.cpt + rnorm(2000)
csm.single.cpt <- IDetect:::cusum_function(single.cpt.noise)
#*** Notice that the maximum in absolute value of \code{csm.single.cpt}
#*** occurs in a neighbourhood of the true change-point, which is 1000.
which.max(abs(csm.single.cpt))
```

---

## cusum_one *Calculate the CUMSUM statistic at specific values*

---

### Description

This function returns the CUMSUM statistic at predefined positions of a given data sequence. The routine is typically not called directly by the user; its result is used in the derivation of the solution path in the case of a piecewise-constant mean signal, which is carried out in sol_path_pcm.

### Usage

```
cusum_one(x, s, e, b)
```

### Arguments

| | |
|---|---|
| x | A numeric vector containing the data. |
| s, e, b | Positive integer vectors, all of the same length $l_b$, with $s_j \leq b_j < e_j, j = 1, 2, ..., l_b$. They indicate that for each $j = 1, 2, ..., l_b$, the function needs to calculate the CUSUM statistic value at position b_j, with start- and end-points at positions s_j and e_j, respectively. |

### Value

A numeric vector of length $l_b$, of which the $j^{th}$ element is the CUSUM statistic value at $b_j$, when the start- and end-points are $s_j$ and $e_j$, respectively.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## See Also

cusum_function for the calculation of the CUSUM statistic for all data points of x. Also, see linear_contr_one for a function that has the same purpose, but for the case of the contrast function for continuous and piecewise-linear mean signals.

## Examples

```
no.cpt.noise <- rnorm(2000)
ex1 <- IDetect:::cusum_one(no.cpt.noise, s = c(1, 5, 9), e = c(30, 56, 71), b = c(20, 40, 45))
```

---

# 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 the mean of a piecewise-constant signal or a 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 ID_pcm or the ID_plm 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 type = ``mean'', then the change-points represent the locations of changes in the mean of a piecewise-constant signal. If type = ``slope'', then the change-points represent the locations of changes in the slope of a piecewise-linear and continuous signal. |

## Details

The data points provided in x are assumed to follow

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

,

where $T$ is the total length of the data sequence, $X_t$ are the observed data, $f_t$ is an one-dimensional, deterministic signal with abrupt structural changes at certain points, and $\epsilon_t$ is white noise. We denote by $r_1, r_2, ..., r_N$ 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 $r_1, r_2, ..., r_N$ minimising the $l_2$ distance between the fit and the data.

### Value

A numeric vector with the estimated signal.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### 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_plm(single.cpt.plm.noise)
fit.cpt.single.plm <- est_signal(single.cpt.plm.noise, cpt.single.plm$cpt, type = "slope")
```

---

## ht_ID_pcm — *Apply the Isolate-Detect methodology for multiple change-point detection in the mean of a vector with non Gaussian noise*

---

### Description

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

### Usage

```
ht_ID_pcm(
  x,
  s_ht = 3,
  l_ht = 300,
  ht_thr_id = 1,
```

```
  ht_th_ic_id = 0.9,
  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.
- `l_ht` A positive integer number with default value equal to 300. If the length of `x` is less than or equal to `l_ht`, then no pre-averaging will take place.
- `ht_thr_id` 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. In this case, the change-points are estimated by thresholding with threshold equal to `sigma * thr_id * sqrt(2 * log(l))`, where `l` is the length of the newly obtained data, after pre-averaging takes place through the `normalise` function.
- `ht_th_ic_id` 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 overestimation) of the model selection approach. 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 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 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 `link{ID_pcm}` on $\tilde{x}_q$ to obtain the change-points, namely $\tilde{r}_1, \tilde{r}_2, ..., \tilde{r}_{\hat{N}}$ in an increasing order. To obtain the original location of the change-points with, on average, the highest accuracy we define

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

More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2017), 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 piecewise-constant mean signal.

solution_path A vector containing the solution path.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## See Also

ID_pcm and normalise, which are functions that are used in ht_ID_pcm. In addition, see ht_ID_plm for the case of continuous and piecewise-linear mean 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)
```

---

ht_ID_plm *Apply the Isolate-Detect methodology for multiple change-point detection in the slope of a vector with non Gaussian noise*

---

## Description

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

## Usage

```
ht_ID_plm(
  x,
  s_ht = 3,
  l_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.
- `l_ht` — A positive integer number with default value equal to 300. If the length of `x` is less than or equal to `l_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 is to be followed. In this case, the change-points are estimated by thresholding with threshold equal to `sigma * thr_id * sqrt(2 * log(l))`, where `l` is the length of the newly obtained data, after pre-averaging takes place through the `normalise` function.
- `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. 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 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 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 `link{ID_plm}` on $\tilde{x}_q$ to obtain the change-points, namely $\tilde{r}_1, \tilde{r}_2, ..., \tilde{r}_{\hat{N}}$ in an increasing order. To obtain the original location of the change-points with, on average, the highest accuracy we define

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

More details can be found in “Detecting multiple generalized change-points by isolating single ones”, Anastasiou and Fryzlewicz (2017), 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 piecewise-linear mean signal.

`solution_path` A vector containing the solution path.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## See Also

ID_plm and normalise, which are functions that are used in ht_ID_plm. 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_plm(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_plm(three.cpt.student)
```

---

ID *Multiple change-point detection in the mean or the slope of a vector 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 either piecewise-constant or piecewise-linear mean of a noisy input vector xd. The noise can either follow the Gaussian distribution or not. Further to 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,
  lam = 3,
  lam.ic = 10,
  contrast = c("mean", "slope"),
  ht = FALSE,
  scale = 3
)
```

## Arguments

- `xd` — A numeric vector containing the data in which you would like to find change-points.
- `th.cons` — 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) in the scenario of piecewise-constant mean signals. In this case, the change-points are estimated by thresholding with threshold equal to `sigma * th.cons * sqrt(2 * log(l))`, where `l` is the length of the data sequence `xd` and `sigma` is equal to `mad(diff(xd)/sqrt(2))`.
- `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 is to be followed) in the scenario of piecewise-linear mean signals. In this case, the change-points are estimated by thresholding with threshold equal to `sigma * th.cons_lin * sqrt(2 * log(l))`, where `l` is the length of the data sequence `xd` and `sigma` is equal to `mad(diff(diff(xd)))/sqrt(6)`.
- `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 is to be followed for the scenario of piecewise-constant mean 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 is to be followed for the scenario of piecewise-linear mean 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.
- `lam` — 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.
- `lam.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 the mean of a piecewise-constant signal. If `contrast = ``slope''`, then the algorithm looks for changes in the slope of a piecewise-linear and continuous 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`.

## Details

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

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

where $T$ is the total length of the data sequence, $X_t$ are the observed data, $f_t$ is an one-dimensional, deterministic signal with abrupt structural changes at certain points, and $\epsilon_t$ are independent and identically distributed random variables with mean zero and variance equal to 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.
- Piecewise-linear and continuous signal with Gaussian noise.
  Use `contrast = "slope"` and `ht = FALSE` here.
- Piecewise-linear and continuous signal with heavy-tailed noise.
  Use `contrast = "slope"` and `ht = TRUE` here.

## 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 piecewise-linear mean signal.

`solution_path` A vector containing the solution path.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## See Also

`ID_pcm`, `ID_plm`, `ht_ID_pcm`, and `ht_ID_plm`, which are the functions that are employed in 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)
```

ID_pcm — *Multiple change-point detection in the mean of a vector using the Isolate-Detect method*

## Description

This function estimates the number and locations of multiple change-points in the piecewise-constant mean of the noisy input vector `x`, using the Isolate-Detect methodology. It also gives the estimated signal, as well as the solution path (see Details for the relevant literature reference).

## Usage

```
ID_pcm(x, thr_id = 1, th_ic_id = 0.9, 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. It is used to define the threshold, if the thresholding approach is to be followed. In this case, the change-points are estimated by thresholding with threshold equal to `sigma * thr_id * sqrt(2 * log(l))`, where `l` is the length of the data sequence `x`. |
| `th_ic_id` | 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 overestimation) of the model selection approach. |
| `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 `wind_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 `cpt_ic_pcm` and this is what is returned. To sum up, `ID_pcm` returns a result based on `cpt_ic_pcm` 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 (2017), 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 piecewise-constant mean signal.

`solution_path` A vector containing the solution path.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

`wind_pcm_th` and `cpt_ic_pcm` which are the functions that `ID_pcm` is based on. In addition, see `ID_plm` for the case of detecting changes in the slope of a piecewise-linear and continuous 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)
```

---

## ID_plm — *Multiple change-point detection in the slope of a vector using the Isolate-Detect method*

---

### Description

This function estimates the number and locations of multiple change-points in the slope of a continuous piecewise-linear mean of the noisy input vector `x`, using the Isolate-Detect methodology. It also gives the estimated signal, as well as the solution path (see Details for the relevant literature reference).

### Usage

```
ID_plm(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. In this case, the change-points are estimated by thresholding with threshold equal to sigma * thr_id * sqrt(2 * log(l)), where l is the length of the data sequence x and sigma is equal to mad(diff(diff(x)))/sqrt(6).

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.

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 wind_plm_th. If the estimated number of change-points is larger than 100, then the result is returned and we stop. Otherwise, ID_plm proceeds to detect the change-points using cpt_ic_plm and this is what is returned. To sum up, ID_plm returns a result based on cpt_ic_plm 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 (2017), 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 mean signal.

solution_path A vector containing the solution path.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## See Also

wind_plm_th and cpt_ic_plm which are the functions that ID_plm 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_plm.

## Examples

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

---

# linear_contr_one — *Calculate the contrast function for the continuous piecewise-linear mean case at specific values*

---

## Description

This function returns, at predefined positions, the values of the contrast function for a given data sequence with under the scenario of continuous, piecewise-linear mean signals. The routine is typically not called directly by the user; its result is used in the derivation of the solution path in the case of a piecewise-linear mean signal, which is carried out in `sol_path_plm`.

## Usage

```
linear_contr_one(x, s, e, b)
```

## Arguments

| | |
|---|---|
| x | A numeric vector containing the data. |
| s, e, b | Positive integer vectors, all of the same length $l_b$, with $s_j \leq b_j < e_j, j = 1, 2, ..., l_b$. They indicate that for each $j = 1, 2, ..., l_b$, the function needs to calculate the contrast function value at position `b_j`, with start- and end-points at positions `s_j` and `e_j`, respectively. |

## Value

A numeric vector of length $l_b$, of which the $j^{th}$ element is the contrast function value at $b_j$, when the start- and end-points are $s_j$ and $e_j$, respectively.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

cumsum_lin for the calculation of the contrast function for all data points of x. Also, see cusum_one for a function that has the same purpose, but for the case of the CUSUM statistic, which is used in piecewise-constant mean signals.

### Examples

```
noise <- rnorm(2000)
ex.lin <- IDetect:::linear_contr_one(noise, s = c(1, 5, 9), e = c(6, 56, 71), b = c(4, 40, 45))
```

---

## log_lik_slope — *Calculate the log-likelihood in the case of a continuous piecewise-linear mean signal*

---

### Description

This function calculates the Gaussian log-likelihood for the continuous piecewise-linear mean signal estimated using est_signal with the changepoints at cpt and for type = ``slope''.

### Usage

```
log_lik_slope(x, cpt)
```

### Arguments

| | |
|---|---|
| x | A numeric vector containing the data. |
| cpt | A positive integer vector with the locations of the change-points. If missing, the ID function is called internally to detect any change-points that might be present in x. |

### Value

The Gaussian log-likelihood for the continuous piecewise-linear mean signal estimated using est_signal with the changepoints at cpt.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### Examples

```
single.cpt.plm <- c(seq(0, 999, 1), seq(998.5, 499, -0.5))
single.cpt.plm.noise <- single.cpt.plm + rnorm(2000)
cpt_detect <- ID(single.cpt.plm.noise, contrast = "slope")
loglik_cpt <- IDetect:::log_lik_slope(single.cpt.plm.noise, cpt_detect$cpt)
```

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 $q = 1, 2, ..., Q-1$, 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 (2017).

## Value

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

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## See Also

ht_ID_pcm, ht_ID_plm, and ID, which are functions that employ normalise.

### Examples

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

---

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 given data sequence.

### 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. The change-points are estimated by thresholding with threshold equal to sigma * thr_const * sqrt(2 * log(l)), where l is the length of the data sequence x. |
| thr_fin | A positive real number with default value equal to sigma * thr_const * sqrt(2 * log(l)), where l is the length of the data sequence x. 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 l, with l 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.

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 (2017), preprint. The concept is simple and is split into two stages; firstly, isolation of each of the true changepoints in small intervals, and secondly their detection.

### Value

A numeric vector with the detected change-points.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

wind_pcm_th, ID_pcm, and ID, which employ this function. In addition, see plm_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 <- 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)
```

---

## plm_th — *Multiple change-point detection in the slope of a piecewise-linear mean signal 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 slope of a piecewise-linear mean of a given data sequence.

### Usage

```
plm_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 mad(diff(diff(x)))/sqrt(6), where 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. The change-points are estimated by thresholding with threshold equal to sigma * thr_const * sqrt(2 * log(l)), where l is the length of the data sequence x. |
| thr_fin | A positive real number with default value equal to sigma * thr_const * sqrt(2 * log(l)). 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 l, with l 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. |
| 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 plm_th is the Isolate-Detect methodology described in "Detecting multiple generalized change-points by isolating single ones", Anastasiou and Fryzlewicz (2017), preprint. The concept is simple and is split into two stages; firstly, isolation of each of the true changepoints in small intervals, and secondly their detection.

### Value

A numeric vector with the detected change-points.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

wind_plm_th, ID_plm, and ID, which employ this function. In addition, see pcm_th for the case of detecting changes in the mean of 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 <- plm_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 <- plm_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 <- plm_th(multi.cpt.noise)
```

---

## resid *Calculate the residuals related to the estimated signal*

---

### Description

This function returns a 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(
  x,
  cpt,
  type_chg = c("mean", "slope"),
  type_res = c("raw", "standardised")
)
```

### Arguments

| | |
|---|---|
| x | A numeric vector containing the data. |
| cpt | A positive integer vector with the locations of the change-points. If missing, the ID function is called internally to detect any change-points that might be present in x. |

type_chg — A character string, which defines the type of the detected change-points. If `type_chg = ``mean''`, then the change-points represent the locations of changes in the mean of a piecewise-constant signal. If `type_chg = ``slope''`, then the change-points represent the locations of changes in the slope of a piecewise-linear and continuous signal.

type_res — 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, <anastasiou.andreas@ucy.ac.cy>

## 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(single.cpt.pcm.noise, cpt = cpt_detect$cpt, type_chg = "mean",
type_res = "raw")

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

plot(residuals_cpt_raw)
plot(residuals_cpt_stand.)
```

---

# 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.

## 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 that we will choose the elements from r and l that are greater than s. |
| e | A positive integer indicating the finishing position, in 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, <anastasiou.andreas@ucy.ac.cy>

### 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)
```

---

## sic_pen *Schwarz Information Criterion penalty*

---

### Description

This function evaluates the penalty term for the Schwarz Information Criterion. The routine is typically not called directly by the user; its name can be passed as an argument to cpt_ic_pcm and cpt_ic_plm.

### Usage

```
sic_pen(n, n_param)
```

### Arguments

| | |
|---|---|
| n | The number of observations. |
| n_param | The number of parameters in the model for which the penalty is evaluated. |

### Value

The penalty term `log(n) * n_param`.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

`ssic_pen` for the strengthened Schwarz Information Criterion penalty.

### Examples

```
three.cpt <- c(rep(4,400),rep(0,400),rep(-4,400),rep(1,400))
three.cpt.noise <- three.cpt + rnorm(1600)
detected_cpts <- cpt_ic_pcm(three.cpt.noise, penalty = "sic_pen")
```

---

## sol_path_pcm *The solution path for the case of piecewise-constant mean signals*

---

### Description

This function starts by over-estimating the number of true change-points. After that, following a CUSUM-based approach, 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 `cpt_ic_pcm`.

### 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_const * sqrt(2 * log(l))`, where `l` is the length of the data sequence `x`. 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, which is the default value used as the threshold constant in the function `wind_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 mean signals.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## 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)
```

---

`sol_path_plm` *The solution path for the case of continuous piecewise-linear mean 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 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 `cpt_ic_plm`.

## Usage

```
sol_path_plm(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_const * sqrt(2 * log(l))`, where `l` is the length of the data sequence `x`. 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 `wind_plm_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 mean signals.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### Examples

```
three.cpt <- c(seq(0, 499, 1.2), 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_plm(three.cpt.noise)
```

---

## ssic_pen *Strengthened Schwarz Information Criterion penalty*

---

### Description

This function evaluates the penalty term for the strengtened Schwarz Information Criterion proposed in Fryzlewicz (2014). The routine is typically not called directly by the user; its name can be passed as an argument to `cpt_ic_pcm` and `cpt_ic_plm`.

### Usage

```
ssic_pen(n, n_param, alpha = 1.01)
```

### Arguments

| | |
|---|---|
| `n` | The number of observations. |
| `n_param` | The number of parameters in the model for which the penalty is evaluated. |
| `alpha` | A real number greater than one. |

### Details

The strengthened Schwarz Information Criterion was introduced in Fryzlewicz (2014). Taking `alpha` = 1 will give the known Schwarz Information Criterion of `sic_pen`.

### Value

The penalty term $log(n)^alpha * n_param$.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### References

Fryzlewicz, P. (2014). Wild Binary Segmentation for multiple change-point detection. Annals of Statistics, Vol. 42, No. 6, 2243-2281.

### See Also

`sic_pen` for the Schwarz Information Criterion penalty.

## Examples

```
three.cpt <- c(rep(4,400),rep(0,400),rep(-4,400),rep(1,400))
three.cpt.noise <- three.cpt + rnorm(1600)
detected_cpts <- cpt_ic_pcm(three.cpt.noise, penalty = "ssic_pen")
```

---

# wind_pcm_th — *A window-based approach for multiple change-point detection in the mean via thresholding*

---

## 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 given data sequence. It is particularly helpful for very long data sequences, as due to applying Isolate-Detect on moving windows, it reduces the computational time (see Details for the relevant literature reference).

## Usage

```
wind_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 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_con | A positive real number with default value equal to 1. It is used to define the threshold. The change-points are estimated by thresholding with threshold equal to `sigma * thr_const * sqrt(2 * log(l))`, where `l` is the length of the data sequence `x`. |
| 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 $i = 1, 2, ..., K$, 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 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 explained in the supplement of "Detecting multiple generalized change-points by isolating single ones", Anastasiou and Fryzlewicz (2017), preprint.

## Value

A numeric vector with the detected change-points.

## Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

## See Also

pcm_th, which is the function that wind_pcm_th is based on. Also, see ID_pcm and ID, which employ wind_pcm_th. In addition, see wind_plm_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 <- wind_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 <- wind_pcm_th(three.cpt.noise)
```

---

wind_plm_th — *A window-based approach for multiple change-point detection in the slope of a picewise-linear mean signal via thresholding*

---

### 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 slope of a given data sequence. It is particularly helpful for very long data sequences, as due to applying Isolate-Detect on moving windows, it reduces the computational time (see Details for the relevant literature reference).

### Usage

```
wind_plm_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 `x`. The default value is `mad(diff(diff(x)))/sqrt(6)`, where `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_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 `sigma * thr_const * sqrt(2 * log(l))`, where `l` is the length of the data sequence `x`. |
| `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 $i = 1, 2, ..., K$, 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 is then applied.

### Value

A numeric vector with the detected change-points.

### Author(s)

Andreas Anastasiou, <anastasiou.andreas@ucy.ac.cy>

### See Also

plm_th, which is the function that wind_plm_th is based on. Also, see ID_plm and ID, which employ wind_plm_th. In addition, see wind_pcm_th for the case of detecting changes in the mean of 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 <- wind_plm_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 <- wind_plm_th(three.cpt.noise)
```

# Index

cpt_ic_pcm, 3, *6*, *18*, *28, 29*, *31*
cpt_ic_plm, 4, *4*, *19*, *28*, *30, 31*
cumsum_lin, 6, *8*, *21*
cusum_function, 7, *7*, *9*
cusum_one, 8, *21*

est_signal, 9

ht_ID_pcm, *3*, 10, *14*, *16*, *22*
ht_ID_plm, *3*, 12, *12*, *16*, *22*

ID, *2–4*, *6*, 14, *18, 19*, *22*, *24*, *26*, *33*, *35*
ID_pcm, *3, 4*, *12*, *16*, 17, *19*, *24*, *33*
ID_plm, *3*, *6*, *14*, *16*, 18, *18*, *26*, *35*
IDetect *(*IDetect-package*)*, 2
IDetect-package, 2

linear_contr_one, *9*, 20
log_lik_slope, 21

normalise, *11–14*, 22

pcm_th, 23, *26*, *33*
plm_th, 24, *24*, *35*

resid, 26

s_e_points, 27
sic_pen, 28, *31*
sol_path_pcm, *8*, 29
sol_path_plm, *20*, 30
ssic_pen, *29*, 31

wind_pcm_th, *17, 18*, *24*, *29*, 32, *35*
wind_plm_th, *19*, *26*, *30*, 33, *33*