R package *stepR*
Multiscale change-point inference

Florian Pein[1] and Axel Munk[1,2]

[1]Institute for Mathematical Stochastics, Georg-August-University of Goettingen, Goldschmidtstr. 7, 37077 Göttingen, Germany
[2]Max Planck Institute for Biophysical Chemistry, Am Fassberg 11, 37077 Göttingen, Germany

November 3, 2019

```
## Warning:  package 'knitr' was built under R version 4.0.0
```

1 Introduction

Multiple change-point detection (fitting a piecewise constant function to serial observations is relevant to many applications, for instance to genetics or finance. Accordingly, this is a long standing task in statistical research and related areas. In addition to estimation / regression, statistical inference (obtaining confidence statements) for locations and size of segments is of high practical importance. The **stepR** package provides both by implementing the multiscale regression estimators SMUCE from Frick et al. (2014) and HSMUCE from Pein et al. (2017). See the introduction of these two papers for a biblography, more details on the history, theory and applications of multiple change-point methods.
The regression model underlying the methodolodgy is described in Section 2. The multiscale regression estimator, its confidence statements and technical quantities are introduced in Section 3. The functions of this package together with small examples are listed in Section 4. Finally, Sections 5 and 6 give an overview about the supported parametric families and interval systems, respectively.

2 Model

This package deals with the estimation of a piecewise constant (step) function

$$\gamma(t) = \sum_{k=0}^{K} \gamma_k \mathbb{1}_{[\tau_k, \tau_{k+1})}(t), \tag{2.1}$$

where the number of change-points $K \in \mathbb{N}_0$, the change-point locations $\tau_1 < \cdots < \tau_K$ and the function values $\gamma_0, \ldots, \gamma_K \in \mathbb{R}$ are all unknown and have to be estimated from the data. Start and end point τ_0 and τ_{K+1} are known. In other words, the set of all possible candidate functions is

$$\Gamma := \{\gamma(t) = \sum_{k=0}^{K} \gamma_k \mathbb{1}_{[\tau_k, \tau_{k+1})}(t),\ K \in \mathbb{N}_0,\ \tau_0 < \cdots < \tau_{K+1},\ \gamma_0, \ldots, \gamma_K \in \mathbb{R}\}. \tag{2.2}$$

To this end, it is assumed that a vector of length $n \in \mathbb{N}_0$ of noisy observations $y = (y_1, \ldots, y_n)$ from a known *parametric family* $\mathcal{P}$ is given, where the function values of γ at known design points $x = (x_1, \ldots, x_n)$ are (unknown) parameters of the family, i.e.

$$y = (y_1, \ldots, y_n) \sim \mathcal{P}(\gamma) = \mathcal{P}((\gamma(x_1), \ldots, \gamma(x_n))). \tag{2.3}$$

For instance, the (homogeneous) gaussian regression model with equidistant design points $x_i = i/n$

$$y_i = \mu(i/n) + \sigma_0 \epsilon_i, \quad i = 1, \ldots, n$$

is supported. For an overview about the supported parametric families see Section 5.

```
set.seed(1)
n <- 100L
x <- seq(1 / n, 1, 1 / n)
mu <- stepfit(cost = 0, family = "gauss", value = c(0, 3, 0, -2, 0), param = NULL,
              leftEnd = x[c(1, 21, 26, 71, 81)],
              rightEnd = x[c(20, 25, 70, 80, 100)], x0 = 0,
              leftIndex = c(1, 21, 26, 71, 81),
              rightIndex = c(20, 25, 70, 80, 100))
sigma0 <- 0.5
epsilon <- rnorm(n, 0, sigma0)
y <- fitted(mu) + epsilon
plot(x, y, pch = 16, col = "grey30", ylim = c(-3, 4))
lines(mu, lwd = 3)
```

3 Methodolodgy

The *multiscale regression estimator* $\hat{\gamma}$ is defined as the restricted maximizer of a functional $L^{\mathcal{P}}(Y)$ (for most parametric families $L^{\mathcal{P}}$ is the likelihood function, the exact functional for each parametric family is given in Section 5), where the candidate set Γ is restricted to all solutions of the (non-convex) optimisation problem to minimize the number of change-points under the constraint that the candidate function is accepted by a multiscale test. In formulas, the multiscale regression estimator $\hat{\gamma}$ is defined by

$$\hat{\gamma} := \operatorname*{argmax}_{\gamma \in C(y,q)} L^{\mathcal{P}}(y, \gamma) \tag{3.1}$$

with

$$C(y, q) := \{\operatorname*{argmin}_{\gamma \in \Gamma} |\gamma|_0 \text{ s.t. } T_n(y, \gamma, q) \leq 0\}, \tag{3.2}$$

with $|\gamma|_0$ the number of change-points of γ, $q = (q_1, \ldots, q_{d_n})$ a vector of *critical values* (see Section 3.2), and *multiscale test*

$$T_n(y, \gamma, q) := \max_{I \in \mathcal{I}:\ \gamma_{|I} \equiv \gamma_I} T_I^{\mathcal{P}}(y, \gamma_I) - q_{|I|_0}. \tag{3.3}$$

Here $\mathcal{I}$ is the *interval set* on which the candidate function is tested (to be defined in Section 3.1), $\gamma_{|I}$ denotes the function γ restricted to the interval I, $T_I^{\mathcal{P}}$ is a *local test statistic* depending on the parametric family (often the corresponding likelihood ratio test, the exact test statistic for each parametric family is given in Section 5) and $|I|_0$ is the number of observations on the interval I, i.e. $|I|_0 := |\{i \in \{1, \ldots, n\} : x_i \in I\}|$.

Minimizing the number of change-point in (3.2) aims to seek for the most persimonious solution, whereas the multiscale constraint aims to detect all true changes of the underlying parameter by guaranteeing that the estimate $\hat{\gamma}$ describes the data everywhere locally well by testing on each interval (of the interval set $\mathcal{I}$) whether the function value corresponds to the parameter underlying the data. More precisely, on each interval $I \in \mathcal{I}$ on which the candidate function γ is constant with value γ_I the hypothesis $\tilde{\gamma} = \gamma_I$ versus the alternative $\tilde{\gamma} \neq \gamma_I$ is tested assuming the observations $\{y_i : x_i \in I,\ i = 1, \ldots, n\}$ have parameter $\tilde{\gamma}$.

The set $C(y, q)$ (3.2) is an asymptotic confidence set for γ at level $1-\alpha$ if q satisfies (3.4), see (Frick et al., 2014, corollary 2) and (Pein et al., 2017, theorem 3.8), but difficult to visualise. Therefore, simultaneous *confidence intervals* $[L_k, R_k]$ for the change-point locations τ_k and a *confidence band* $B(q)$ for γ are provided, see (Frick et al., 2014, section 3.2) and (Pein et al., 2017, supplement section A.1) as well as (Frick et al., 2014, theorem 7 and corollary 3) and (Pein et al., 2017, theorem 3.9) for theoretical verification, in particular it is shown that the union $I(q) = \{\hat{K}, B(q), [L_k, R_k]_{k=1,\ldots,\hat{K}}\}$ is sequentially honest with respect to certain subspaces Γ_n, see (Frick et al., 2014, corollary 3) for details.

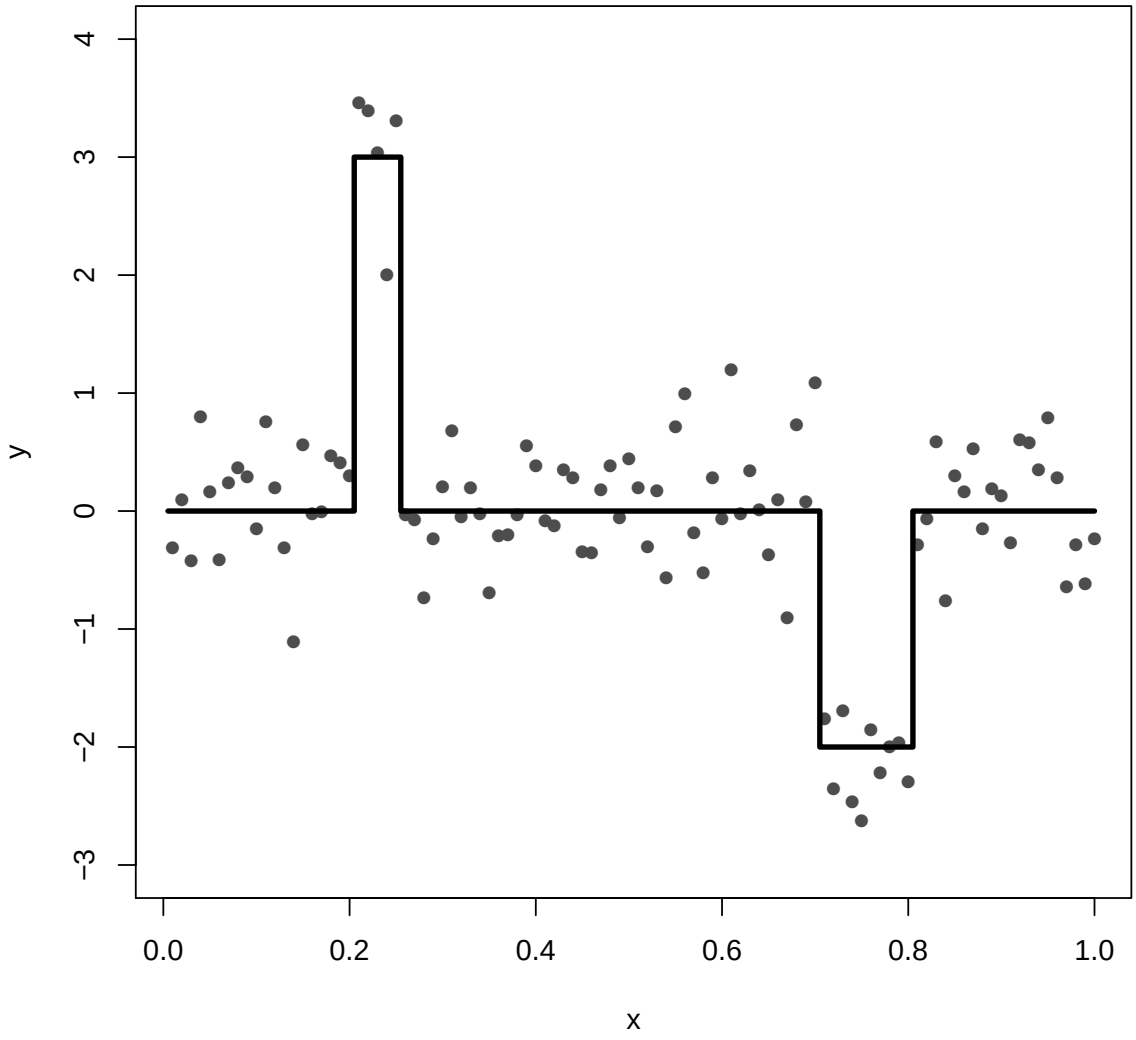

Figure 1: Observations (grey points) and underlying function (black line).

```r
fit <- stepFit(y, x = x, alpha = 0.5, jumpint = TRUE, confband = TRUE)

plot(x, y, pch = 16, col = "grey30", ylim = c(-3, 4))
lines(mu, lwd = 3)
lines(fit, lwd = 3, col = "red", lty = "22")

# confidence intervals for the change-point locations
points(jumpint(fit), col = "red")
# confidence band
lines(confband(fit), lty = "22", col = "darkred", lwd = 2)
```

3.1 Interval set

The user can choose the interval set $\mathcal{I}$ by specifying an *interval system*, see Section 6 for an overview about the supported interval systems, and a set of *lengths* $l = \{l_1, \dots, l_{d_n}\} \subseteq \{1, \dots, n\}$. Here, d_n denotes the *number of scales*, i.e. the number of different interval lengths on which will be tested. The interval set will contain all intervals of the interval system with a length that is in the set of lengths. Note that not all lengths are possible for each interval system (see Section 6 for which ones are possible for which system) and also not for each parametric family (see Section 5 for which ones are possible for which family).

3.2 Critical values, penalisation and Monte-Carlo simulations

The vector of critical values $q = (q_1, \dots, q_{d_n})$ is chosen such that the multiscale test is a test at *significance level* $\alpha \in (0, 1)$, i.e. for $\gamma_0 \equiv 0$ and $z \sim \mathcal{P}(\gamma_0)$

$$\mathbb{P}(T_n(z, \gamma_0, q) < 0) \leq \alpha \tag{3.4}$$

is required. (3.4) implies the overestimation control that the probability to overestimate the number of change-points is uniformly over all candidate functions bounded by the significance level α, i.e.

$$\sup_{\gamma \in \Gamma} \mathbb{P}_\gamma(\hat{K} > K) \leq \alpha. \tag{3.5}$$

Here $\mathbb{P}_\gamma$ denotes that $\hat{K}$ is estimated based on $y \sim \mathcal{P}(\gamma)$, see (Frick et al., 2014, (17)) and (Pein et al., 2017, theorem 3.3). Hence, if a strict overestimation control of the number of change-points K is desirable α should be chosen small, e.g. 0.05 or 0.1. This might come at the expense of missing (smaller) change-points but with large probability not detecting too many. If change-point screening is the primarily goal, i.e. missing of change-points should be avoided, α should be increased, e.g. $\alpha = 0.5$ or even higher, since the error probability to underestimate the number of change-points decreases with increasing α, see (Frick et al., 2014, theorem 2 and (23)) and (Pein et al., 2017, theorem 3). If model selection, i.e. $\hat{K} = K$, is the major aim, an intermediate level that balances the over- and underestimation error should be chosen, e.g. α between 0.1 and 0.5. For a detailed discussion of the choice of α see (Frick et al., 2014, section 4) and (Pein et al., 2017, section 3.4). This package offers two approaches to satisfy (3.4) and to balance different scales (statistics on intervals of different lengths): *scale penalisation* and *balancing by weights*.

3.2.1 Scale penalisation

This approach, proposed in Frick et al. (2014), balances different scales by a penalty function $p_{|I|_0,n}(\cdot)$ leading to the *penalised multiscale statistic*

$$T_n^{p_{|I|_0,n}}(y, \gamma) := \max_{I \in \mathcal{I}:\ \gamma_{|I} \equiv \gamma_I} p_{|I|_0,n}(T_I^{\mathcal{P}}(y, \gamma_I)) \tag{3.6}$$

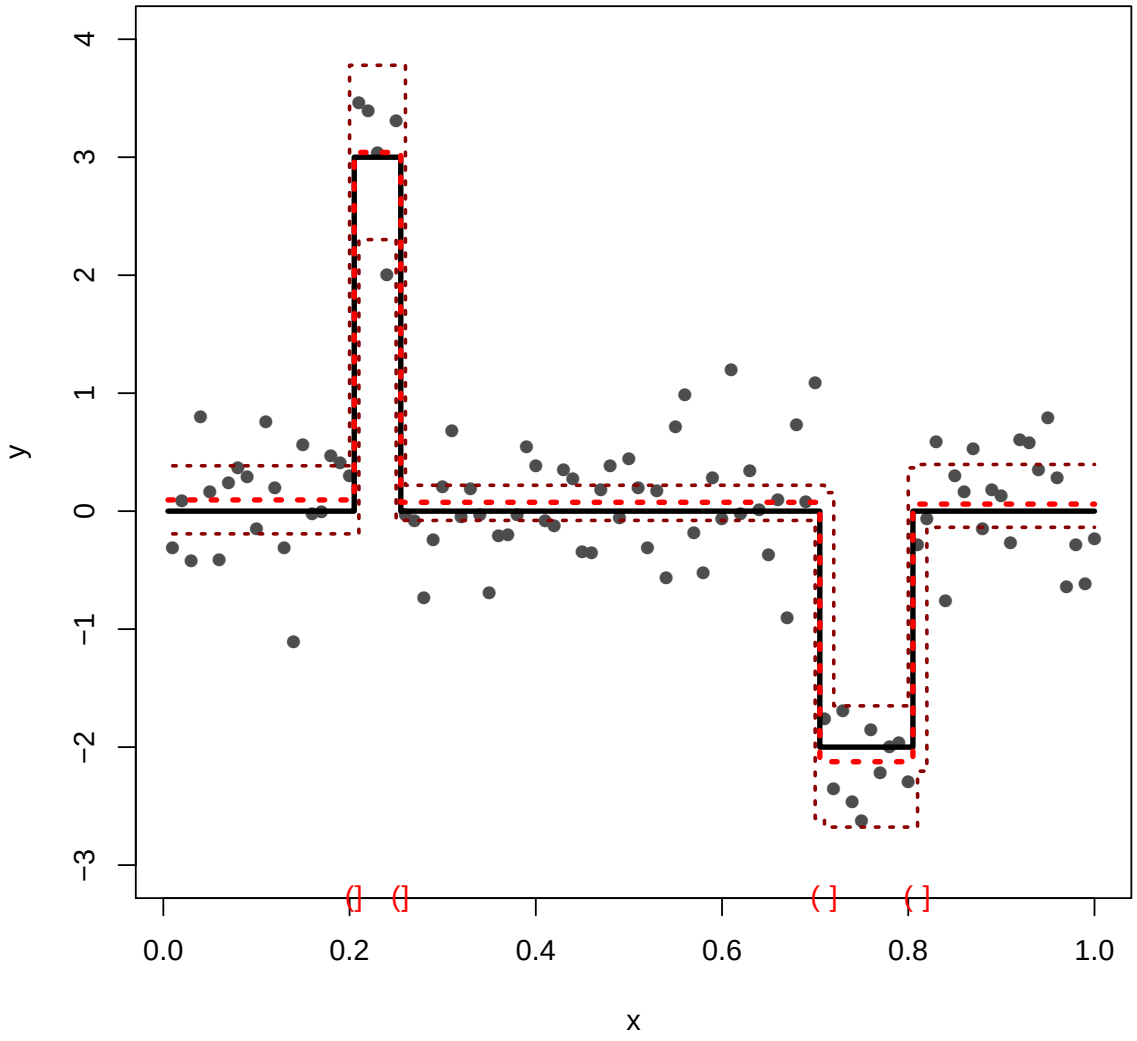

Figure 2: Observations (grey points), underlying function (black line), fit by the multiscale estimator (red line), its confidence intervals for the change-point locations (red brackets) and its confidence band for the underlying function (darkred lines).

and the *multiscale vector of penalised statistics*

$$T_n^{l,p_{|I|_0,n}}(y,\gamma) := \left(\max_{\substack{I\in\mathcal{I}:\ |I|_0=l_1,\\ \gamma_{|I}\equiv\gamma_I}} p_{|I|_0,n}(T_I^{\mathcal{P}}(y,\gamma_I)), \ldots, \max_{\substack{I\in\mathcal{I}:\ |I|_0=l_{d_n},\\ \gamma_{|I}\equiv\gamma_I}} p_{|I|_0,n}(T_I^{\mathcal{P}}(y,\gamma_I)) \right). \tag{3.7}$$

Then, the *global quantile* q_α at significance level α is defined as the $(1-\alpha)$ quantile of $T_n^{p_{|I|_0,n}}(z,\gamma_0)$ and the critical values are determined by $q_i = p_{i,n}^{-1}(q_\alpha)$. The most common penalisation $p_{|I|_0,n}(t) = \sqrt{2t} - \sqrt{2\log(exp(1)n/|I|_0)}$, called "sqrt" in this package, implies for the families "gauss" and "mDependentPS" that

$$T_n^{p_{|I|_0,n}}(y,\gamma) \xrightarrow{D} M, \tag{3.8}$$

where M is a functional of the standard Brownian motion and finite almost surely. For a precise definition and more details see (Frick et al., 2014, section 2.2). See also its minimax optimality in (Frick et al., 2014, theorems 5 and 6). Consequently, this penalty guarantees appropriate scale balancing and, hence, is the default penalty for these families. Note, that this does not apply to inhomogeneous Gaussian observations underlying the parametric family "hsmuce", hence, this penalty is not recommended here. This package also supports the penalties $p_{|I|_0,n}(t) = t - \log(exp(1)n/|I|_0)$, called "log" and $p_{|I|_0,n}(t) = t$, called "none", where the statistics are not penalized and, hence, the multiscale test is dominated by smaller scales. For a discussion how these penalties influence the final outcome, see (Frick et al., 2014, section 6.2).

3.2.2 Balancing by weights

For this approach, proposed in (Pein et al., 2017, section 2) and chosen by penalty = "weights" (although this approach does not correspond to a penalty in a literal sense), the *multiscale vector of statistics* is defined as

$$T_n^{l}(y,\gamma) := \left(\max_{\substack{I\in\mathcal{I}:\ |I|_0=l_1,\\ \gamma_{|I}\equiv\gamma_I}} T_I^{\mathcal{P}}(y,\gamma), \ldots, \max_{\substack{I\in\mathcal{I}:\ |I|_0=l_{d_n},\\ \gamma_{|I}\equiv\gamma_I}} T_I^{\mathcal{P}}(y,\gamma) \right). \tag{3.9}$$

Moreover, for given weights

$$\beta_1,\ldots,\beta_{d_n} > 0, \text{ with } \sum_{k=1}^{d_n} \beta_k = 1, \tag{3.10}$$

in addition to (3.4),

$$\frac{1-F_1(q_1)}{\beta_1} = \cdots = \frac{1-F_{d_n}(q_{d_n})}{\beta_{d_n}}, \tag{3.11}$$

with F_i the cumulative distribution function of the i-th entry of $T_n^l(z,\gamma_0)$, is required. The weights determine the fractions between the probabilities that a test on a certain scale rejects, and hence regulate the allocation of the level α among the single scales. Then, for any $\alpha \in (0,1)$ and for any weights $\beta_1,\ldots,\beta_{d_n}$, s.t. (3.10) holds, there exits a unique vector of critical values $q = (q_1,\ldots,q_{d_n}) \in \mathbb{R}_+^{d_n}$ which fulfils the equations (3.4) and (3.11), see (Pein et al., 2017, lemma 2.1). This vector can be computed by (Pein et al., 2017, algorithm 1 in supplement A.2) based on Monte-Carlo simulations. This penalty is the default and only meaningful penalty for the parametric family "hsmuce", but can also be used for other families. As default uniform weights $\beta_1 = \cdots = \beta_n = 1/n$ are proposed, but other choices can be used to incorporate prior knowledge on the scales, see also (Pein et al., 2017, section 3.4) for more details on the choice of the weights.

3.2.3 Monte-Carlo simulations

Calculating the critical values analytically appears to be a very hard task, therefore for $y = z \in \mathcal{P}(\gamma_0)$ and $\gamma = \gamma_0$ the distribution of either the penalised multiscale statistic (3.6) or of the multiscale vector of statistics (3.9), where l is the set of all lengths allowed by the interval system and by the parametric family, is simulated by Monte-Carlo simulations. This means to generate r, the *number of repetitions*, independent samples of $z \sim \mathcal{P}(\gamma_0)$ and to compute for each the corresponding statistic. Resulting in a d_n

times r matrix, containing r independent simulations of the multiscale vector of statistics (3.9), an object of class *"MCSimulationVector"*, or in an r dimensional vector, containing r independent simulations of the penalised multiscale statistic (3.6), an object of class *"MCSimulationMaximum"*. Since Monte-Carlo simulations can last very long, this package offers several possibilities to store them, see Section 3.4 for more details.

3.3 Dynamic program

The multiscale regression estimator $\hat{\gamma}$ can be computed efficiently by a pruned dynamic program. The acceptance region of each local test is an interval with lower *bound* l_I and upper *bound* u_I, i.e.

$$[l_I, u_I] := \{\gamma_I \in \mathbb{R} : T_I^{\mathcal{P}}(y, \gamma_I) \leq q_{|I|_0}\}. \tag{3.12}$$

Hence, to be accepted by the multiscale test a candidate function has to satisfy these constraints on all intervals of the interval set $\mathcal{I}$ on which it is constant. For more details on the computation of the estimator and its confidence statements based on these bounds see (Pein et al., 2017, supplement section A.1) and (Frick et al., 2014, section 3).

3.4 Storing of the Monte-Carlo simulations

Since a Monte-Carlo simulation lasts potentially much longer than the main calculations of the estimator, this package offers multiple possibilities for saving and loading the simulations. Both, an object of the class *"MCSimulationVector"*, i.e. a d_n times r matrix, containing r independent simulations of the multiscale vector of statistics (3.9), and an object of the class *"MCSimulationMaximum"*, i.e. an r dimensional vector, containing r independent simulations of the penalised multiscale statistic (3.6) (for penalties "sqrt", "log" and "none"), can be simulated, saved and loaded. An object of the class *"MCSimulationVector"* is more flexible, since critical values for all penalties and all set of lengths can be derived from it, but requires much more storage space and slightly larger storing and loading times. The Monte-Carlo simulations depend on the number of observations, the parametric family and the interval system, for *"MCSimulationMaximum"* additionally the set of lengths and the used penalty matters. Note that Monte-Carlo simulations can only be saved and loaded if they are generated with the default function for generating the observations.
Monte-Carlo simulations can also be performed for a (slightly) larger number of observations n_q, this means to calculate $T_{n_q}^{p_{|I|_0}, n_q}(z, \gamma_0)$ or $T_{n_q}^{l}(z, \gamma_0)$ with $z = (z_1, \ldots, z_{n_q}) \sim \mathcal{P}(\gamma_0)$, which avoids extensive resimulations for only a little bit varying number of observations. The overestimation control (3.5) is still satisfied but the detection power is (slightly) smaller. The number n_q can be specified by the variable *nq*, as default, the next larger dyadic number minus one, i.e. $n_q = 2^{\lceil \log_2(n) \rceil} - 1$, is used. Additionally, the user can decide whether simulations (if required) will be performed with n or with n_q, see below.
The simulations can either be stored persistently on the file system for which the package *R.cache* is used or in the workspace in the global variable *critValStepRTab*. Loading from the workspace is faster, but either the user has to store the workspace manually or in a new session simulations have to be performed again. Moreover, storing in and loading from variables and *rds* files is supported. Finally, an initial collection of simulations can be accessed by installing the package *stepRdata* available from `http://www.stochastik.math.uni-goettingen.de/stepRdata_1.0-0.tar.gz`.
Whenever computation of the critical values / the global quantile will be required, i.e. when the variable q is not given, it will be looked for avaiable outcomes of the required Monte-Carlo simulations in the following order: They can explicitly be given in the variable *stat*, they can be given as an *rds* file in the variable *RDSfile*, they can be loaded from the workspace, they can be loaded from the file system, they can be accessed from the *stepRdata* package (if installed) and if no other option was available they will be simulated by the function *monteCarloSimulation*. For the workspace and the file system it will first be looked for a vector of the penalised multiscale statistic in the workspace and then on the file system, afterwards in both for a matrix of the multiscale vector of statistics for n observations, afterwards in both for a vector of the penalised multiscale statistic with n_q observations and then for a matrix of the multiscale vector of statistics with n_q observations. All searches can be disabled by specifying the variable *options* accordingly, please see the documentation of the function *critVal* for technical details.
The user can decide by the variable *options*, too, whether a vector of the penalised multiscale statistic or

matrix of the multiscale vector of statistics with n or with n_q observations will be simulated if simulations are required. Please take into account the explanations regarding computation time and flexibility from the beginning of this section for this choice.
All available simulations (either simulated, loaded or computed in the case of a vector of the penalised statistic) can be saved by all available options, please use again the variable *options* for a decision which will be stored and see the documentation of the function *critVal* for technical details.
As default, the matrix of the multiscale vector of statistics with n_q observations will be stored on the file system and the penalised multiscale statistic in the work space. If required the matrix of the multiscale vector of statistics with n_q observations will be simulated and it will be attempted to load the required simulations from any option in the previously described order.

4 Methods

This section gives a brief overview about the available functions in the package, for more detailed information see the documentation of each function itself.

stepFit Computes the multiscale regression estimator $\hat{\gamma}$ (3.1), confidence intervals for the change-point locations and confidence bands for γ, see Section 3, by a pruned dynamic program, see Section 3.3. The computation of the confidence intervals and the confidence band can be requested by the variables *jumpint* and *confband*, respectively. An example was given in Section 3.

critVal Computes the vector of critical values q and the global quantile q_α at significance level α for a given penalisation, see Section 3.2.

```
# was called in stepFit, can be called explicitly,
# for instance outside of a for loop to save computation time
qVector <- critVal(length(y), alpha = 0.5)
identical(stepFit(y, x = x, q = qVector, jumpint = TRUE, confband = TRUE), fit)

## [1] TRUE

qValue <- critVal(length(y), alpha = 0.5, output = "value")
identical(stepFit(y, x = x, q = qValue, jumpint = TRUE, confband = TRUE), fit)

## [1] TRUE
```

computeStat Computes the multiscale vector of penalised statistics $T_n^{l,p_{|I|_0,n}}(y,\gamma)$ (3.7) and the penalised multiscale statistic $T_n^{p_{|I|_0,n}}(y,\gamma)$ (3.6) for given $\gamma \in \Gamma$.

```
# fit satisfies the multiscale contraint, i.e.
# the computed penalized multiscale statistic is not larger than the global quantile
computeStat(y, signal = fit, output = "maximum") < qValue

## [1] TRUE

# multiscale vector of statistics is componentwise not larger than
# the vector of critical values
all(computeStat(y, signal = fit, output = "vector") < qVector)

## [1] TRUE
```

computeBounds Computes the multiscale contraint, i.e. the bounds (3.12) for all intervals I in the interval set $\mathcal{I}$.

```
# the multiscale contraint
bounds <- computeBounds(y, alpha = 0.5)
```

monteCarloSimulation Performs Monte-Carlo simulations of the multiscale vector of statistics (3.9) and of the penalised multiscale statistic (3.6) for $y = z \sim \mathcal{P}(\gamma_0)$ and $\gamma = \gamma_0$, see Section 3.2.3.

```
# monteCarloSimulation will be called in critVal, can be called explicitly
# object of class MCSimulationVector
stat <- monteCarloSimulation(n = length(y))
identical(critVal(n = length(y), alpha = 0.5, stat = stat),
          critVal(n = length(y), alpha = 0.5,
                  options = list(load = list(), simulation = "matrix")))

## [1] TRUE

# object of class MCSimulationMaximum
stat <- monteCarloSimulation(n = length(y), output = "maximum")
identical(critVal(n = length(y), alpha = 0.5, stat = stat),
          critVal(n = length(y), alpha = 0.5,
                  options = list(load = list(), simulation = "vector")))

## [1] TRUE
```

Moreover, the functions *compareBlocks*, *neighbours*, *sdrobnorm*, *stepcand*, *steppath*, *stepsel* from version 1.0-0 are still available, please see the documentation of these functions for more details.

4.1 Deprecated functions

The following functions are deprecated, but still working, however, they might be removed in a further version. There are two groups of deprecated functions. First of all, the functions *BesselPolynomial*, *contMC*, *dfilter*, *jsmurf*, *transit* are mainly used for patchclamp recordings and might be moved to a specialised package. Secondly, the functions *MRC*, *bounds*, *smuceR*, *stepbound* are deprecated and functions of the new version should be used instead. Please see the documentation of the functions itself for more information and how they can be replaced.

5 Parametric families

This package supports several parametric families (models and fitting methods). Please note that some require additional parameters. For technical details see also the documentation *parametricFamily*. In addition to these families the function *smuceR* supports "gaussvar", "poisson", "binomial", "gaussKern". They will be added to the new functions in a further version, too.

"gauss" This family implements the **S**imulataneous **MU**ltiscale **C**hange-point **E**stimator (SMUCE) Frick et al. (2014) for independent Gaussian observations. More precisely, independent normally distributed data y with unknown mean function $\gamma := \mu$ but known, constant standard deviation $\sigma_0 > 0$ are assumed, i.e.

$$y_i = \mu(x_i) + \sigma_0 \epsilon_i, \quad i = 1, \ldots, n, \tag{5.1}$$

with $\epsilon_1, \ldots, \epsilon_n$ independent standard normal distributed errors. The standard deviation σ_0 has to be either given by the user or will be estimated using the difference-based estimator

$$\frac{\mathrm{IQR}(|y_1 - y_2|, \ldots, |y_{n-1} - y_n|)}{\sqrt{2}\,\mathrm{IQR}(\mathcal{N})} \tag{5.2}$$

with IQR(x) the interquartile range of the sample x, i.e. the 75%-quantile minus the 25%-quantile of x, and IQR($\mathcal{N}$) the theoretical interquartile range of the standard normal distribution, i.e. the 75%-quantile minus the 25%-quantile of the standard normal distribution. This estimator is rather robust against changes of the underlying mean function and is implemented in the function *sdrobnorm*. For this family the multiscale regression estimator $\hat{\gamma}$ is defined as the restricted maximum likelihood estimator, i.e.

$$L^{\mathcal{P}}(y,\gamma) := \sum_{i=1}^{n} \big(y_i - \gamma(x_i)\big)^2. \tag{5.3}$$

The local test statistics are the statistics of the likelihood ratio test

$$T_I^{\mathcal{P}}(y,\gamma_I) := \frac{|I|_0 \,(\overline{y}_I - \gamma_I)^2}{2\sigma^2}, \tag{5.4}$$

with $\overline{y}_I := \sum_{i:\ x_i \in I} y_i/|I|_0$, hence, the bounds are derived by

$$[l_I, u_I] := \left[\overline{y}_I - \sigma\sqrt{\frac{2q_{|I|_0}}{|I|_0}}, \overline{y}_I + \sigma\sqrt{\frac{2q_{|I|_0}}{|I|_0}}\right]. \tag{5.5}$$

All lengths $\{1,\dots,n\}$ are allowed for this family.

"hsmuce" This family implements the **H**eterogeneous **S**imulataneous **MU**ltiscale **C**hange-point **E**stimator (HSMUCE) Pein et al. (2017). More precisely, independent normal distributed data y with unknown mean function $\gamma := \mu$ and also unknown standard deviation function σ are assumed, i.e.

$$y_i = \mu(x_i) + \sigma(x_i)\epsilon_i, \quad i = 1,\dots,n, \tag{5.6}$$

with $\epsilon_1,\dots,\epsilon_n$ independent standard normal distributed errors. Here, the unknown mean function is the parameter of interest, whereas the standard deviation is considered as a nuisance parameter. Moreover, the standard deviation is assumed to change only at the change-point locations of μ, i.e. function pairs from the set

$$\left\{ \big(\mu,\sigma\big) = \left(\sum_{k=0}^{K} \mu_k \mathbb{1}_{[\tau_k,\tau_{k+1})},\ \sum_{k=0}^{K} \sigma_k \mathbb{1}_{[\tau_k,\tau_{k+1})}\right),\ K \in \mathbb{N}_0,\ \tau_0 < \dots < \tau_{K+1},\ \mu_0,\dots,\mu_K \in \mathbb{R},\ \sigma_0,\dots,\sigma_K > 0 \right\}, \tag{5.7}$$

with $\mu_i \neq \mu_{i+1}$, but $\sigma_i = \sigma_{i+1}$ allowed, are assumed. For this family the multiscale regression estimator $\hat{\gamma}$ is defined as the restricted maximum likelihood estimator, i.e.

$$L^{\mathcal{P}}(y,\gamma) := \sum_{i=1}^{n} \big(y_i - \gamma(x_i)\big)^2/\hat{\sigma}_i^2, \tag{5.8}$$

with $\hat{\sigma}_i = \sum_{j\in I^i} (y_j - \gamma(x_i))^2/|I^i|_0$, $I^i = \{j \in \{1,\dots,n\}\ :\ \gamma(x_k) = \gamma(x_i)\ \forall\ k = j,\dots,i\}$. The local test statistics are the statistics of the likelihood ratio test

$$T_I^{\mathcal{P}}(y,\gamma_I) := \frac{|I|_0 \,(\overline{y}_I - \gamma_I)^2}{2\hat{\sigma}_I^2}, \tag{5.9}$$

with $\overline{y}_I := \sum_{i:\ x_i \in I} y_i/|I|_0$ and $\hat{\sigma}_I^2 := \sum_{i:\ x_i\in I} (y_i - \overline{y}_I)^2/(|I|_0 - 1)$, hence, the bounds are derived by

$$[l_I, u_I] := \left[\overline{y}_I - \hat{\sigma}_I\sqrt{\frac{2q_{|I|_0}}{|I|_0}}, \overline{y}_I + \hat{\sigma}_I\sqrt{\frac{2q_{|I|_0}}{|I|_0}}\right]. \tag{5.10}$$

All lengths larger than 1, $\{2,\dots,n\}$, are allowed for this family.

"mDependentPS" This family implements the **S**imulataneous **MU**ltiscale **C**hange-point **E**stimator (SMUCE) Frick et al. (2014) for m-dependent Gaussian observations with known covariance structure. More precisely, m-dependent normal distributed data y with unknown mean function $\gamma := \mu$ but known covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$ are assumed, i.e.

$$y_i = \mu(x_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{5.11}$$

with centred errors $\mathbb{E}[\epsilon_i] = 0$ and known covariance $\Sigma_{i,j} = \mathrm{Cov}[\epsilon_i, \epsilon_j] = \sigma^2_{|j-i|} > 0$ if $|j - i| \leq m$ and $\Sigma_{i,j} = \mathrm{Cov}[\epsilon_i, \epsilon_j] = 0$ else. $\sigma^2 = (\sigma_0^2, \ldots, \sigma_m^2)$ is called the *vector of covariances*. For this family the multiscale regression estimator $\hat{\gamma}$ is defined as the restricted least squares estimator, i.e.

$$L^{\mathcal{P}}(y, \gamma) := \sum_{i=1}^{n} \big(y_i - \gamma(x_i)\big)^2. \tag{5.12}$$

The local test statistics are the statistics of the partial sum test

$$T_I^{\mathcal{P}}(y, \gamma_I) := \frac{\big(\sum_{i:\ x_i\in I} y_i - \gamma_I\big)^2}{2\,\mathrm{Var}\big(\sum_{i:\ x_i\in I} y_i\big)}, \tag{5.13}$$

with $\mathrm{Var}\big(\sum_{i:\ x_i\in I} y_i\big) = |I|_0\sigma_0^2 + 2\sum_{k=1}^{m}(|I|_0 - k)_+\sigma_k^2$, $x_+ = \max(x, 0)$, hence, the bounds are derived by

$$[l_I, u_I] := \left[\overline{y}_I - \sqrt{\frac{2\,\mathrm{Var}\big(\sum_{i:\ x_i\in I} y_i\big) q_{|I|_0}}{|I|_0}}, \overline{y}_I + \sqrt{\frac{2\,\mathrm{Var}\big(\sum_{i:\ x_i\in I} y_i\big) q_{|I|_0}}{|I|_0}}\right], \tag{5.14}$$

with $\overline{y}_I := \sum_{i:x_i\in I} y_i/|I|_0$. All lengths $\{1, \ldots, n\}$ are allowed for this family.

6 Interval systems

This package supports the following interval systems.

"all" The system of all intervals (with start and end points at the observation grid). More precisely, the set

$$\{[x_i, x_j],\ 1 \leq i \leq j \leq n\}. \tag{6.1}$$

This system allows all lengths $1, \ldots, n$.

"dyaLen" The system of all intervals of dyadic length. More precisely, the set

$$\{[x_i, x_j],\ 1 \leq i \leq j \leq n \text{ s.t. } \exists\ k \in \mathbb{N}_0 :\ j - i + 1 = 2^k\}. \tag{6.2}$$

This system allows all dyadic lengths $2^0, \ldots, 2^{\lfloor \log_2(n)\rfloor}$.

"dyaPar" The system of the dyadic partition, i.e. all disjoint intervals of dyadic length. More precisely, the set

$$\{[x_{(i-1)2^k+1}, x_{i2^k}],\ i = 1, \ldots, \lfloor n/2^k\rfloor,\ k = 0, \ldots, \lfloor \log_2(n)\rfloor\}. \tag{6.3}$$

This system allows all dyadic lengths $2^0, \ldots, 2^{\lfloor \log_2(n)\rfloor}$.

References

Frick, K., Munk, A., and Sieling, H. (2014). Multiscale change-point inference. With discussion and rejoinder by the authors. *J. Roy. Statist. Soc. Ser. B*, 76(3):495–580.

Pein, F., Sieling, H., and Munk, A. (2017). Heterogeneous change point inference. *J. Roy. Statist. Soc. Ser. B*, 79(4):1207–1227.