

Change-in-Slope Optimal Partitioning Algorithm in a Finite-Size Parameter Space

Vincent Runge

Université Paris-Saclay, CNRS, Univ Evry,
Laboratoire de Mathématiques et Modélisation d'Evry,
91037, Evry-Courcouronnes, France*

Marco Pascucci

Université Paris-Saclay, CNRS, Univ Evry,
Institut des neurosciences Paris-Saclay, 91400, Orsay, France.

Nicolas Deschamps de Boishebert

Université Paris-Saclay, CNRS, Univ Evry,
Laboratoire de Mathématiques et Modélisation d'Evry,
91037, Evry-Courcouronnes, France

Abstract

We consider the problem of detecting change-points in univariate time series by fitting a continuous piecewise linear signal using the residual sum of squares. Values of the inferred signal at slope breaks are restricted to a finite set of size m . Using this finite parameter space, we build a dynamic programming algorithm with a controlled time complexity of $O(m^2n^2)$ for n data points. Some accelerating strategies can be used to reduce the constant before n^2 . The adapted classic inequality-based pruning is outperformed by a simpler “channel” method on simulations. Besides, our finite parameter space setting allows an easy introduction of constraints on the inferred signal. For example, imposing a minimal angle between consecutive segment slopes provides robustness to model misspecification and outliers. We test our algorithm with an isotonic constraint on an antibiogram image analysis problem for which algorithmic efficiency is a cornerstone in the emerging context of mobile-health and embedded medical devices. For this application, a finite state approach can be a valid compromise.

Keywords: Multiple change-point detection, change in slope, pruned dynamic programming, isotonic constraint, unimodal constraint, robust inference

1 Introduction

Detecting change-points in time series is a long-standing problem in statistics that have been tackled in many different ways since the 1950s. Originally developed for quality control in manufacturing [31], it has spread to the most modern sciences such as genomics [19, 6], neuroscience [27, 8] or climate change [3, 47, 33, 5] among many others. Due to the wide range of possible modeling assumptions and problem settings, change-point detection remains today an active scientific field, especially challenging for large data [30].

*E-mail: vincent.runge@univ-evry.fr

1.1 Search Algorithms

Last decades, researchers have been principally focused on the multiple change-point problem which consists in finding the location and the number of changes in a time series $y_{1:n} = (y_1, \dots, y_n)$ of length n . Among the 2^{n-1} possible change-point vectors, the problem requires to select one vector for which a criterion is minimized or close to the minimum. Algorithmic efficiency is then a central challenge in order to rapidly infer a good change-point vector candidate.

Binary segmentation (BS) is one of the best known algorithm [38, 39] for solving this problem. It returns an approximate solution in time $O(n \log(n))$. Many other approaches have been proposed: see [45] for a review and [12] for some comparisons. A well-known alternative method is to use a dynamic programming approach [4] called optimal partitioning (OP) [22] which returns the best change-point vector minimizing the criterion¹. Its principle consists in finding the best last segment in consecutive truncated time series $y_{1:t}$ for t from 1 to n . Time complexity of this algorithm is of order $O(n^2)$ as for each $y_{1:t}$ we test the t available positions (candidates) for potential best last change-point.

In recent years, pruning ideas were proposed to reduce the set of candidates to consider for each truncated time series $y_{1:t}$, so that the complexity of OP became comparable with BS in some simulation studies. Two main classes of pruning are available: inequality-based pruning implemented in the PELT algorithm [25] and functional pruning [34, 28] implemented in the FPOP algorithm. PELT is based on a recursion on the position of the last change and remains close to OP (only a conditional expression is added). PELT is efficient only when the number of change-points is proportional to data length. FPOP is based on a recursion on the value of the last segment parameter and is more difficult to code: we need to update a functional cost at any step, that is a continuous piecewise quadratic function (in Gaussian model). However, this strategy achieves the best possible pruning and the log-linear time complexity on many data-sets [28].

1.2 Change in Slope With Continuity Constraint

Many of these dynamic programming algorithms were developed to solve the change-in-mean problem for which we fit a piecewise constant signal. Another problem deals with changes in trend, also called changes in slope or changes in regression. It consists in fitting a piecewise linear signal to the time series. Many algorithms were developed for this task [40] using for example Bayesian inference [37] or Hypothesis testing [29] but most of them do not consider a continuity constraint between successive segments. This literature on change-point detection is often related to the estimation of climate variations [3].

The change-in-slope problem with continuity constraint² is much less studied, even though many applications need such models [47, 23, 41, 44]. A few approaches directly tackle this problem, using ideas of the Wild Binary Segmentation method [14, 2], trend filtering [26, 42], Bayesian approaches [32], or dynamic programming [11]. The latter is based on an FPOP-like algorithm (CPOP) and fits a continuous piecewise linear signal based on the minimization of the residual sum of squares of each segment. It introduces a functional cost parametrized by the signal value associated to the current analysed data point. Using this cost, the authors derive a one-parametric functional update and eventually get a time complexity between $O(n^2)$ and $O(n^3)$ (see simulations in Section 5.2).

¹The problem can have many best change-point vectors with the same minimal value given a criterion. However, this multiple solution is an event of zero probability with real-valued time series.

²In the following, we often write "change in slope" to mean "change in slope with continuity constraint".

1.3 Contribution

The guideline of our work was the construction of a simplified efficient algorithm for change-in-slope problem with continuity constraint, capable of returning the optimal change-point vector in a controlled time complexity of order $O(n^2)$. For this purpose, we use a dynamic programming approach with values of the inferred signal at slope breaks restricted to a finite set of values (called states). Furthermore, we were also interested in the possibility to easily constrain the inference as for change-in-mean algorithm GFPOP [10]. Our algorithm can restrict the inference to an increasing signal but also force a minimal angle value between consecutive segment slopes. This latter option provides stability for signal inference (filtering outliers) which is a desired features for many applications. The use of states is not only a bypass for reducing running time and enforce constraints: this restricted inference can match the expected level of complexity for some applications as for example in beat detection, for which the musician only needs integer beat-per-minute information [16, 35].

1.4 Outline

The outline of this paper is as follows. Section 2 introduces the model and describes the optimal partitioning algorithm for changes in slope with a simple description of the continuity constraint. We also give the three proposed constrained inference modes: isotonic, unimodal and minimal angle. Pruning-like approaches for speeding up the algorithm are exposed in Section 3. Section 4 is dedicated to variance estimation of time series with slopes. In the simulation study in Section 5 we show the benefit of using the continuity hypothesis by measuring the mean squared error (MSE) and the Adjusted Rand Index (ARI) between the true signal and the inferred one. We also search for the range of penalties minimizing the MSE for 4 different simulation scenarios. We then compare the efficiency of the two proposed pruning methods and the time complexity with the FPOP-like challenging algorithm CPOP [11]. Eventually, we show how using the minimal-angle constraint enhances the stability of the inference. In last Section 6 we apply our algorithm to an antibiogram image analysis problem to find a unique change-point. In this application where non-decreasing signals are expected, the use of a monotonicity constraint improves detection precision. In this example, although the true signal is not expected to have a finite number of states, the finite-state trade-off allows a sensible gain in time and still a good analysis precision. In the context of mobile-health applications and hardware with limited computing capacity, where efficiency is important, such a trade-off could be fundamental.

Our change-in-slope algorithms are available into an R package on CRAN called slopeOP³ and on github⁴.

2 Change-in-slope Optimal Partitioning

2.1 Model and Cost Function

We define the set of states \mathcal{S} as a finite set of accessible real values for beginning and ending values in inferred segments. When we write $s = s_{min}, \dots, s_{max}$, the variable s goes through all the values of \mathcal{S} from the smallest one to the biggest one. For computational efficiency we recommend to have $\#\mathcal{S} = m \ll n$ but this is not mandatory. The key idea is that limiting the number of states decreases computational time, while still capturing some useful information

³<https://CRAN.R-project.org/package=slopeOP>

⁴<https://github.com/vrunge/slopeOP>

(see Section 6). Notice that with too many states, the challenging algorithm CPOP would be more efficient.

Data are generated by the model [11]:

$$Y_t = s_i + \frac{s_{i+1} - s_i}{\tau_{i+1} - \tau_i}(t - \tau_i) + \epsilon_t, \quad t = \tau_i + 1, \dots, \tau_{i+1}, \quad i = 0, \dots, k,$$

with $0 = \tau_0 < \tau_1 < \dots < \tau_k < \tau_{k+1} = n$, $s_0, \dots, s_{k+1} \in \mathcal{S}$ and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ identically and independently distributed. Observations are denoted in lowercase y_1, \dots, y_n . The change-point positions are given by integer vector $(\tau_0, \dots, \tau_{k+1})$ and define a subdivision in $k + 1$ consecutive segments $(y_{\tau_i+1}, \dots, y_{\tau_{i+1}})$. This modeling imposes to define the cost for fitting data $y_{(\tau+1):t}$, $\tau < t$, with linear interpolation from value $s_1 \in \mathcal{S}$ to value $s_2 \in \mathcal{S}$ as a residual sum of squares:

$$\mathcal{C}(y_{(\tau+1):t}, s_1, s_2) = \sum_{i=\tau+1}^t \left(y_i - \left(s_1 + (s_2 - s_1) \frac{i - \tau}{t - \tau} \right) \right)^2. \quad (1)$$

Notice that the cost $(y_\tau - s_1)^2$ that would have been obtained at index τ is not present in the summation, unlike the right bound $(y_t - s_2)^2$. This choice of cost function will lead to an update rule with a simple description of the continuity condition for consecutive segments.

2.2 Optimization Problem

The *slopeOP* problem consists of finding the optimal partitioning of a time series $y_{1:n} = (y_1, \dots, y_n) \in \mathbb{R}^n$ of size n into $k + 1$ consecutive segments $(y_{\tau_i+1}, \dots, y_{\tau_{i+1}})$ optimizing a penalized risk

$$Q_n^{slope} = \min_{\substack{\tau=(\tau_1, \dots, \tau_k) \in \mathbb{N}^k \\ \tau_0=0, \tau_{k+1}=n \\ (s_0, \dots, s_{k+1}) \in \mathcal{S}}} \sum_{i=0}^k \{ \mathcal{C}(y_{(\tau_i+1):\tau_{i+1}}, s_i, s_{i+1}) + \beta \} - \beta, \quad (2)$$

where the states defined inside the cost function provide the continuity constraint between successive segments and β is a positive penalty. This penalty controls the amount of evidence we need to add a change: the greater this quantity, the less is k . The "minus beta" term is added in order to get the null value in case of a perfect fit (null residuals) with no change. We emphasise that k is an unknown quantity. The penalty value is often set to the BIC penalty, $2\hat{\sigma}^2 \log(n)$ [43, 49], as we have a result of asymptotic consistency for change-in-mean problems [46]. The quantity $\hat{\sigma}^2$ is an estimation of the true variance σ^2 . Some other penalty values can be also relevant [9, 48]. For the change-in-slope problem, theorems for a similar penalty have been given in asymptotic regime [49]. As variance estimation is needed, we proposed a robust estimator of the variance based on an adaptation to the difference estimator of Hall et al. [17] (see Table 1). In non-asymptotic regime, setting the penalty is a difficult task as it may also depends on signal shape (see Figure 4), that is the form of the deterministic part in the time series ($t \mapsto Y_t - \epsilon_t$).

2.3 Dynamic Programming Algorithm

To address the continuity constraint in an efficient algorithm we introduce the function

$$Q_t : \begin{cases} \mathcal{S} & \rightarrow \mathbb{R} \\ u & \mapsto Q_t(u), \end{cases}$$

which is the optimal penalized cost up to position t with the inferred value at this position equal to u . Our goal is then to update the set

$$\mathcal{Q}_t = \{Q_t(u), u = s_{min}, \dots, s_{max}\},$$

at any time step $t \in \{1, \dots, n\}$, where s_{min} and s_{max} are bounds that can be determined in a pre-processing step. With y_- and y_+ the minimal and maximal values of the time-series, setting for example the bounds around $y_{\pm} \pm \frac{y_+ - y_-}{2}$ include all extreme intersection points between consecutive segments (segment of two points (y_-, y_+) followed by (y_+, y_-) gives the upper bound). We find the objective value Q_n^{slope} with the minimization $Q_n^{slope} = \min_{s_{min} \leq u \leq s_{max}} \{Q_n(u)\}$.

Proposition 2.1. *The update rule with continuity constraint takes the form*

$$Q_t(v) = \min_{0 \leq t' < t} \left(\min_{s_{min} \leq u \leq s_{max}} \{Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta\} \right), \quad (3)$$

where the state u in $Q_{t'}$ and the cost function including states realizes the continuity constraint. At the initial step we have $Q_0(v) = -\beta$ for all $v \in \mathcal{S}$.

The proof is detailed in Appendix A and the corresponding optimal partitioning algorithm based on (3) is given in Algorithm 1.

Algorithm 1: Change-in-slope Optimal Partitioning (slopeOP)

Input: data $y_{1:n}$, set of states $\mathcal{S} = \{s_{min}, \dots, s_{max}\}$ and penalty $\beta > 0$

- 1 Q = matrix of size $(n + 1) \times m$
- 2 cp = matrix of size $n \times m$
- 3 U = matrix of size $n \times m$
- 4 **for** $v = s_{min}$ **to** s_{max} **do**
- 5 | $Q(0, v) \leftarrow -\beta$
- 6 **end**
- 7 **for** $t = 1$ **to** n **do**
- 8 | **for** $v = s_{min}$ **to** s_{max} **do**
- 9 | | $Q(t, v) \leftarrow \min_{0 \leq t' < t} \left(\min_{s_{min} \leq u \leq s_{max}} \{Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta\} \right)$
- 10 | | $cp(t, v) \leftarrow \operatorname{argmin}_{0 \leq t' < t} \left(\min_{s_{min} \leq u \leq s_{max}} \{Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta\} \right)$
- 11 | | $U(t, v) \leftarrow \operatorname{argmin}_{s_{min} \leq u \leq s_{max}} \left(\min_{0 \leq t' < t} \{Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta\} \right)$
- 12 | **end**
- 13 **end**
- 14 **return** cp, U and $Q(n, \cdot)$

The slopeOP algorithm is then an OP algorithm searching for the best last change-point in $y_{1:t}$ for t from 1 to n . But, differently from OP, it looks for it in a finite set of couples (state,time) of size $m \times t$, where the state is in the finite states set. Summing up for all states and for t from 1 to n , we get an overall running time of order $O(m^2n^2)$. In next Section, we propose two accelerating strategies to reduce this initial m^2n^2 time complexity.

The backtracking step is a little different from the standard optimal partitioning algorithm as we need to take into account the states.

Algorithm 2: Backtracking Algorithm

Input: Output of Algorithm 1 : cp, U and $\{Q(n, v), v \in \mathcal{S}\}$

- 1 $chpts \leftarrow ()$
- 2 $states \leftarrow (\operatorname{argmin}_{s_{min} \leq v \leq s_{max}} \{Q(n, v)\})$
- 3 $t \leftarrow n$
- 4 **while** $t > 0$ **do**
- 5 $chpts \leftarrow (t, chpts)$
- 6 $states \leftarrow (U(t, states(0)), states)$
- 7 $t \leftarrow cp(t, states(1))$
- 8 **end**
- 9 **return** $chpts$ and $states$

2.4 Inference With Signal Shape Constraints

Besides its unambiguous computational time efficiency in $O(m^2n^2)$, one of the benefit of this finite-state approach is the possibility to simply enforce some constraints in the inference. The quantity to optimize is then

$$Q_n^{ctt} = \min_{\substack{\text{vectors } \tau \text{ and } s \\ (\tau, s) \in U_{k+1}}} \sum_{i=0}^k \{\mathcal{C}(y_{(\tau_i+1):\tau_{i+1}}, s_i, s_{i+1}) + \beta\} - \beta, \quad (4)$$

where U_{k+1} is the set of constraints containing the positions (couples) that can be used in the minimization. We use notation ν_i for a couple (τ_i, s_i) position-state, $(\tau, s) = ((\tau', n), (s', v))$ is a sequence of couples with the last one being (n, v) and $U_{k+1}^{(n, v)}$ is the set U_{k+1} of sequences of couples with a last couple equal to (n, v) . A dynamic programming algorithm solving exactly (4) can be built if in $U_{k+1}^{(n, v)}$ the information over the path (ν_1, \dots, ν_k) can be conveyed alongside this path to the current ν_k position. That is:

$$\begin{aligned} U_{k+1}^{(n, v)} &= \{(\tau, s) \mid (\tau', s') \in U_k^{\nu_k}, f(\nu_1, \dots, \nu_k, (n, v)) = 1\} \\ &= \{(\tau, s) \mid (\tau', s') \in U_k^{\nu_k}, g(M(\nu_k), \nu_k, (n, v)) = 1\}, \end{aligned}$$

with $g : \mathbb{R} \times (\{0, \dots, n\} \times \mathcal{S})^2 \rightarrow \{0, 1\}$ a validity test associated to the constraint. Function f is reduced to g . The ‘‘memory’’ function $M : \{0, \dots, n\} \times \mathcal{S} \rightarrow \mathbb{R}$ summarizes the information of the path (ν_1, \dots, ν_k) and is associated in practice to the cost $Q_{t'}(u)$ of the update rule. If g equals 0, the couple $\nu_k = (t', u)$ cannot be considered in the minimization.

With this definition for U_{k+1} we get the update rule:

$$Q_t^{ctt}(v) = \min_{\substack{\tilde{v}=(t', u) \in \{0, \dots, t-1\} \times \mathcal{S} \\ \nu=(t, v), g(M(\tilde{v}), \tilde{v}, \nu)=1}} \left(Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta \right). \quad (5)$$

In this rule, the positions \tilde{v} of the minimization operator are taken into account according to the g function value depending only on the current couple $\nu = (t, v)$ and past events memorized by function $M(\cdot)$. We illustrate this constraint approach by the examples of isotonic, unimodal and stable inferences.

Isotonic case. The update rule for isotonic constraint is:

$$Q_t^{iso}(v) = \min_{0 \leq t' < t} \left(\min_{s_{min} \leq u \leq v} \{Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta\} \right), \quad (6)$$

which is close to (3) but with a constrained minimization for variable u . Here $M(\tilde{\nu}) = 0$ for all $\tilde{\nu}$ and $g(0, \tilde{\nu}, \nu) = 1$ if and only if $u \leq v$ for $\tilde{\nu} = (t', u)$ and $\nu = (t, v)$. We illustrate the use of the isotonic constraint in Section 6 with an antibiogram image analysis problem.

Unimodal case. $M(\tilde{\nu})$ is equal to 1 if at position $\tilde{\nu} = (t', u)$ no decreasing segment has been yet inferred. Otherwise $M(\tilde{\nu}) = 0$. $g(1, \cdot, \cdot) = 1$ and

$$g(0, \tilde{\nu} = (t', u), \nu = (t, v)) = \begin{cases} 0 & \text{if } u < v, \\ 1 & \text{if } u \geq v. \end{cases}$$

Value $M(\nu)$ is determined just after setting $Q(\nu) = Q_t(v)$ using the slope sign information on the segment $(\tilde{\nu}, \nu)$.

Minimal angle case. $M(\tilde{\nu}) \in \mathbb{R}$ is the slope value of the last inferred segment at position $\tilde{\nu}$. Here the value $g(M(\tilde{\nu}), \tilde{\nu}, \nu)$ computes the angle deviation in degree between slope $M(\tilde{\nu})$ and slope formed by $(\tilde{\nu}, \nu)$. If the obtained value is less than a threshold we have $g(M(\tilde{\nu}), \tilde{\nu}, \nu) = 0$ and $g(M(\tilde{\nu}), \tilde{\nu}, \nu) = 1$ otherwise. With this constraint, we tend to be robust to outliers with the right level of smoothness (the threshold). We also hope to get some kind of robustness to model misspecification (see simulations in Section 5.3).

In the R package slopeOP, we also provide a {Segment Neighborhood dynamic programming method returning the best change-point vector with a given fixed number of segments. With k changes, its time complexity is about k times greater than the complexity of slopeOP.

3 Accelerating Strategies

As revealed by Algorithm 1 the double loop for variables t and s (lines 7 and 8) is time consuming. We first give a formula for a constant-time computation of cost (1). Then we propose two accelerating strategies aiming at reducing the set of values to consider for the search for minimum in matrix Q . The first one is a PELT-like method based on inequalities of type “ $Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) > Q_t(v)$ ”. The second one considers direct comparisons between elements of the minimization in (3). We test the efficiency of those two accelerating rules on simulations in Section 5.2. As the constraints can force the choice of non-minimal elements in (5) these methods can only be implemented in non-constraint setting (at the exception of the isotonic constraint).

3.1 Efficient Cost Computation

Proposition 3.1. *The cost function (1) can be computed in constant time with the formula*

$$\begin{aligned} \mathcal{C}(y_{(t'+1):t}, u, v) = & S_t^2 - S_{t'}^2 - \frac{2}{t-t'} \left((ut - vt')(S_t^1 - S_{t'}^1) + (v - u)(S_t^+ - S_{t'}^+) \right) \\ & + \frac{v^2 - u^2}{2} + \frac{u^2 + uv + v^2}{3}(t - t') + \frac{(v - u)^2}{6(t - t')}, \end{aligned}$$

where $t' < t \in \{0, \dots, n\}$, $(u, v) \in \mathcal{S}^2$ and

$$S_t^1 = \sum_{i=1}^t y_i \quad , \quad S_t^2 = \sum_{i=1}^t y_i^2 \quad \text{and} \quad S_t^+ = \sum_{i=1}^t iy_i \quad \text{for all } t \in \{1, \dots, n\}.$$

The proof is straightforward by direct (tedious) calculations. The vectors $(S_t^1)_{t=1,\dots,n}$, $(S_t^2)_{t=1,\dots,n}$ and $(S_t^+)_{t=1,\dots,n}$ are computed in a pre-processing step in $O(n)$ time and saved. This is a marginal cost compared to the cost for Algorithm 1. Notice that the $O(m^2n^2)$ time complexity is obtained using the result of Proposition 3.1.

3.2 Inequality-based Pruning

With the following Proposition 3.2 we build a standard inequality-based pruning rule that has the specificity to take into account future data. Our cost function (1) doesn't use a minimisation as in PELT: we cannot guarantee that splitting a segment into two segments would result in a smaller overall cost. This property is at the core of the PELT pruning rule.

At each position $v \in \mathcal{S}$, we need to update a set of positions:

$$\mathcal{U}(t, v) \subset \{(t', u), t' \in \{0, \dots, t-1\}, u \in \{s_{min}, \dots, s_{max}\}\},$$

to get $\mathcal{U}(t+1, v)$. There are m such sets. Therefore, we actually chose to transfer information at constant state value.

Proposition 3.2. *At time t and position v , suppose that there exists $t' < t$ and $u \neq v$ such that*

$$Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) > Q_t(v). \quad (7)$$

We define $S_t^{t'} = \sum_{i=t'+1}^t (i-t')y_i$ and coefficients $(\alpha_t^-, \alpha_t^+, \gamma_t^-, \gamma_t^+)$ such that

$$\alpha_t^+ T + \gamma_t^+ \leq \sum_{i=t+1}^{T-1} y_i \frac{T-i}{T-t} \leq \alpha_t^- T + \gamma_t^-, \quad T = t+1, \dots, n.$$

We also define for all integers $T > t$ the affine in T functions:

$$\begin{cases} f^+(T) = \left(\alpha_t^+ - \frac{u+2v}{6} \right) T + t' \frac{u+2v}{6} - \frac{v-u}{12(t-t')} + \frac{S_t^{t'}}{t-t'} + \gamma_t^+, \\ f^-(T) = \left(\alpha_t^- - \frac{u+2v}{6} \right) T + t' \frac{u+2v}{6} - \frac{v-u}{12(t-t')} + \frac{S_t^{t'}}{t-t'} + \gamma_t^-. \end{cases}$$

If $f^+(t+1) \geq 0$ and $f^+(n) \geq 0$ (case $v-u > 0$) or if $f^-(t+1) \leq 0$ and $f^-(n) \leq 0$ (case $v-u < 0$) then the position (t', u) doesn't have to be considered for further iterations $T > t$ for computing $Q_T(v)$ in Algorithm 1. We remove (t', u) from $\mathcal{U}(T, v)$ for all $T > t$.

The proof is given in Appendix B.

The use of additional parameters (the states) in cost function leads to a more complex pruning rule than the one for standard PELT [25] as the overall cost is not guarantee to decrease as the number of change-point increases. However, in case $u = v$ we have a simpler result:

Proposition 3.3. *At time t , if there exists $t' \in \{0, \dots, t-1\}$ such that*

$$Q_{t'}(v) + \mathcal{C}(y_{(t'+1):t}, v, v) > Q_t(v), \quad (8)$$

then the position t' doesn't have to be considered for further iterations $T > t$ for computing $Q_T(v)$ in Algorithm 1.

Proof. We use the fact that we have

$$\mathcal{C}(y_{(t'+1):t}, v, v) + \mathcal{C}(y_{(t+1):T}, v, v) = \mathcal{C}(y_{(t'+1):T}, v, v),$$

for any $t' < t < T$. Adding the quantity $\mathcal{C}(y_{(t+1):T}, v, v)$ to (8) and using this equality we get

$$Q_{t'}(v) + \mathcal{C}(y_{(t'+1):T}, v, v) + \beta > Q_t(v) + \mathcal{C}(y_{(t+1):T}, v, v) + \beta,$$

which means that (t', v) is a choice less optimal than (t, v) for the minimization in $Q_T(v)$. \square

We have a second pruning rule for v -constant positions which is original in the sense that a recent position can be pruned by an older one.

Proposition 3.4. *At time t , if there exists $t' \in \{0, \dots, t-1\}$ such that*

$$Q_{t'}(v) + \mathcal{C}(y_{(t'+1):t}, v, v) \leq Q_t(v), \quad (9)$$

then the position t doesn't have to be considered for further iterations $T > t$ for computing $Q_T(v)$ in Algorithm 1.

The proof is the same as for Proposition 3.3.

Remark 3.1. *One of the two inequalities (8) and (9) is always satisfied so that the subset $\mathcal{U}(t, v) \cap \{(t', v), t' = 1, \dots, t-1\}$ contains at most one element.*

3.3 Channel Pruning

We propose a new rule for speeding-up the algorithm, called the “channel method”. It does not prune the position (t', u) in the classical sense, as a non-considered value can be further reintegrated in the minimization (in a set $\mathcal{U}(t, v)$). However its simplicity could help to reduce time complexity.

We expect the cost Q to verify some structural properties. In particular,

$$q_t = \begin{cases} \mathcal{S} & \rightarrow \mathbb{R} \\ v & \mapsto Q_t(v) \end{cases}$$

should have a minimum for v “near” the data, which means that q_t is, for most t , decreasing until a state v_l and increasing from a state $v_u \geq v_l$. The function

$$C_{(t'+1):t}^{\tilde{v}} = \begin{cases} \mathcal{S} & \rightarrow \mathbb{R} \\ v & \mapsto \mathcal{C}(y_{(t'+1):t}, v, \tilde{v}) \end{cases}$$

has a global minimum that can be easily found (it's a quadratic function). The idea is to study the variations of $q_{t'} + C_{(t'+1):t}^{\tilde{v}}$ in order to leave out the state values for which we know they are away from the argminimum.

Proposition 3.5. *We consider the function $q_{t'} + C_{(t'+1):t}^{\tilde{v}}$ with $t' + 1 < t$. The minimum of this function is inside the interval*

$$I_{t'}^{\tilde{v}} = \mathcal{S} \setminus (S_{min} \cup S_{max}),$$

with

$$\begin{cases} S_{min} = \{s_{min}, \dots, \min(v_l^{-1}, [v^*]^{-1})\}, \\ S_{max} = \{\max(v_u^{+1}, [v^*]^{+1}), \dots, s_{max}\}, \end{cases}$$

where $[\cdot]$ is the nearest “value in set \mathcal{S} ” operator and $(\cdot)^{+1}$ (resp. $(\cdot)^{-1}$) denotes the following (resp. preceding) value in the ordered set \mathcal{S} . Function $q_{t'}$ is decreasing on $\{s_{min}, \dots, v_l\}$ and increasing on $\{v_u, \dots, s_{max}\}$. We also know that the argminimum of $C_{(t'+1):t}^{\tilde{v}}$ is given by formula

$$v^* = \frac{6}{(t-t'-1)(2(t-t')-1)} \sum_{i=t'+1}^t \left[(t-i)y_i \right] - \tilde{v} \frac{t-t'+1}{2(t-t')-1}.$$

Proof. v^* is easy to compute as the argminimum of the quadratic function $C_{(t'+1):t}^{\tilde{v}}$. The set S_{min} (resp. S_{max}) corresponds to the set on which we know that function $q_{t'} + C_{(t'+1):t}^{\tilde{v}}$ is decreasing (resp. increasing) with the next (resp. previous) value in \mathcal{S} giving a smaller output. \square

Notice that the closest state search $[v^*]$ is a $O(1)$ operation if \mathcal{S} is made of consecutive integers or a $O(\log(n))$ operation in a general ordered list. We define the matrix $\mathcal{Q} \in \mathbb{R}^{m \times n}$ with $\mathcal{Q}_{ij} = Q_j(s_i)$. The name “channel” comes from the fact that the matrix \mathcal{Q} is restricted by a channel with fixed v_u and v_l states for each column of this matrix. For all $\mathcal{U}(t, v)$, this channel is updated to $I_{t'}^{\tilde{v}}$ after computing the value v^* and using Proposition 3.5.

4 Variance Estimation

With real data-sets the noise level σ^2 in time series is an unknown quantity that has to be estimated. In change-in-mean problems there exist many estimators for the variance of a time-series as for example the mean absolute deviation (MAD) or HALL estimators [17]. They have the property to be slightly sensitive to change-points when the number of changes is small compared to data length. In slope problems we face the additional difficulty to remove the slope effect. We suggest to adapt the HALL estimator to our change-in-slope problem. In classic mean problem, we have for HALL of order 3:

$$\hat{\sigma}_{mean}^2 = \frac{1}{n-3} \sum_{j=1}^{n-3} \left(\sum_{k=0}^3 d_k Y_{j+k} \right)^2,$$

with $d_0 = 0.1942$, $d_1 = 0.2809$, $d_2 = 0.3832$ and $d_3 = -0.8582$. We apply Hall on the successive differences (“HALL Diff”):

$$\hat{\sigma}_{slope}^2 = \frac{1}{(n-4)\Delta} \sum_{j=1}^{n-4} \left(\sum_{k=0}^3 d_k (Y_{j+k+1} - Y_{j+k}) \right)^2,$$

with $\Delta = d_0^2 + (d_1 - d_0)^2 + (d_2 - d_1)^2 + (d_3 - d_2)^2 + d_3^2 = 1.527507$ a normalization coefficient.

To evaluate the quality of the HALL Diff estimator, we simulate time series of length 100 with 2 different signals: the first one is piecewise linear and the second one is sinusoidal, as shown in Figure 1.

In Table 1, we present results for estimating the standard deviation σ for 10^4 simulated time series for each level of noise ($\sigma = 1, \dots, 5$). We easily conclude that the HALL Diff method gives very accurate results, even with a non-linear (sinusoidal) model. We also have a better precision of the estimated sigma for HALL Diff than for MAD. Simulations with more change-points (10) yield similar results (see Appendix C).

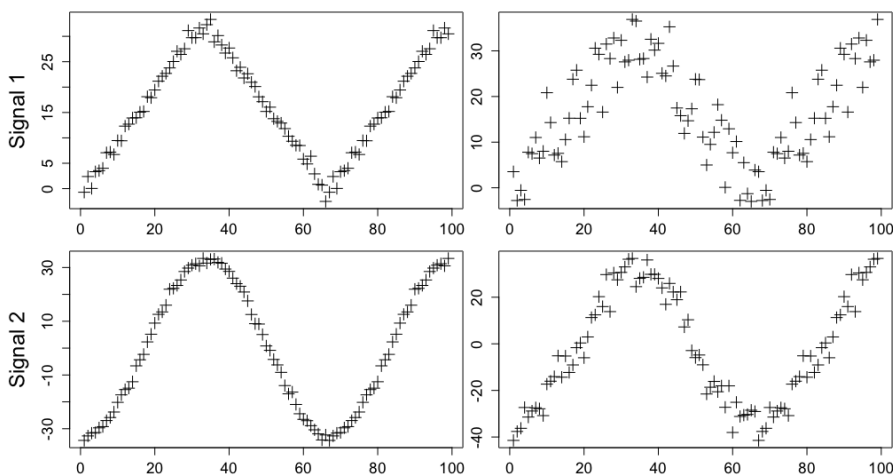


Figure 1: Examples for the linear and sinusoidal time series with noise $\sigma = 1$ (left) and $\sigma = 5$ (right).

method	signal	σ	1	2	3	4	5
MAD	slope	mean($\hat{\sigma}$)	1.23	2.11	3.08	4.05	5.05
		sd($\hat{\sigma}$)	0.13	0.25	0.39	0.52	0.64
	sinus	mean($\hat{\sigma}$)	1.93	2.54	3.36	4.26	5.22
		sd($\hat{\sigma}$)	0.16	0.26	0.38	0.51	0.64
HALL	slope	mean($\hat{\sigma}$)	1.84	2.52	3.37	4.28	5.23
		sd($\hat{\sigma}$)	<i>0.048</i>	<i>0.13</i>	<i>0.21</i>	<i>0.29</i>	<i>0.37</i>
	sinus	mean($\hat{\sigma}$)	3.58	3.97	4.56	5.27	6.06
		sd($\hat{\sigma}$)	<i>0.026</i>	<i>0.082</i>	<i>0.16</i>	<i>0.24</i>	<i>0.32</i>
HALL Diff	slope	mean($\hat{\sigma}$)	1.01	2.00	3.00	3.99	4.99
		sd($\hat{\sigma}$)	0.087	0.18	0.27	0.36	0.44
	sinus	mean($\hat{\sigma}$)	1.02	2.00	3.00	3.99	4.99
		sd($\hat{\sigma}$)	0.087	0.18	0.26	0.36	0.44

Table 1: Variance estimation with MAD, HALL and HALL Diff estimators. The closest values to the true sigma are in bold; the smallest standard deviations in italic.

5 Simulation Study

Our simulation study is split into three parts. We first compare the mean squared error (MSE) and the Adjusted Rand Index (ARI) between the true signal and the inferred one with many different algorithms and study the impact of choosing different β penalty values. In the next part, we compare the two proposed pruning rules and also select our main competitor, CPOP, to challenge slopeOP in terms of computational efficiency. Finally, we consider misspecified time series with an heavy-tailed noise and explore the capacity of slopeOP with a minimal angle between segments to infer a good model on a range of penalties.

5.1 MSE Competition with Other Algorithms

We consider the following five methods:

- OP-2D : we fit the data with the PELT algorithm but the cost function on a segment is the residual sum of squares between data and a linear regression (2-dimensional fit). In

that case, we don't have any continuity constraint;

- FPOP : The standard efficient gaussian FPOP algorithm [28] used on differenced data $z_t = y_{t+1} - y_t$, $t = 1, \dots, n - 1$. The continuity is obtained by construction;
- RFPOP : Same algorithm as the previous FPOP but with robust biweight loss (with robust parameter $K = 3\sigma$) [13];
- CPOP : A FPOP-like algorithm for changes in slope with continuity constraint [11]. The set of states is here infinite-dimensional (\mathbb{R}) and not finite as for slopeOP;
- slopeOP : Our finite-state change-in-slope OP algorithm.

The first idea to detect change in slope consists of looking at changes in data $z_t = y_{t+1} - y_t$ with any well-known change-in-mean algorithm. We use here the efficient FPOP and robust FPOP algorithms for the inference. We highlight in Figure 2 the fact that with a simple hat-shaped model and a low level of noise, the FPOP algorithms give bad performances as the number of datapoint increases. Therefore, we left out these two methods and study only the behavior of three algorithms: OP-2D, CPOP and slopeOP.

We now consider 4 scenarios of increasing complexity with 3 levels of noise as shown in Figure 3. The beginning and ending values of all segments are integers in $\{0, \dots, 60\}$ and we choose a set of states $\mathcal{S} = \{-10, \dots, 70\}$ to reduce side effects in limit points (0 and 60). We simulate 150 time series of length 500 for each of the couple (scenario, noise) and draw in Figure 4 the graph of the MSE, the ARI and computation time with respect to $b = \frac{\beta}{\sigma^2 \log(n)}$, where σ is the chosen known level of noise. For each of the 150 time series, the 3 algorithms are run for $b = 0.1, 0.2, \dots, 4.9, 5$ and each point of the curves ($\log(\text{MSE})$, ARI or time) is the mean over the 150 independent simulations.

We discuss the results for the intermediate level of noise ($\sigma = 12$) exposed in Figure 4, other results are exposed in Appendix D. For all scenarios CPOP and slopeOP give very close and similar good results compared to OP-2D. The lack of continuity constraint explains the higher MSE value and lower ARI for OP-2D in scenarios 1, 3 and 4. For all scenarios and all noise levels, the optimal penalty in our simulations with $n = 500$ is close to $b = 2$ for slopeOP and CPOP as expected in asymptotic regime [49]. Execution time is smaller for CPOP in most scenarios as data length is here limited to 500 (see next Subsection for more details). In Figure 5, we plotted the segmentation obtained by OP-2D and slopeOP at the smallest MSE value. The result highlight the need for a continuity constraint to get a better inference.

5.2 Computational Efficiency

We first compare time efficiency for the two accelerating methods: channel-based rule and inequality-based pruning. We consider hat-shaped time series with a signal from 10 to 50 and 61 integers states ($\mathcal{S} = \{0, \dots, 60\}$) for inference. In Figure 6, we ran 100 repetitions for each sigma value with data of length 500 and two hat-shaped signals (with 1 hat or 25 regularly spaced hats). We easily understand that the channel rule is more efficient for all noise levels. It may be possible to get similar pruning efficiency in some very particular situations but we conclude with these simulations that the default rule should always be the channel method. Moreover, this latter is simpler as it processes each element in the cost matrix quicker (compare Propositions 3.2 and 3.5).

We now compare time efficiency between slopeOP and CPOP algorithms with two noise levels and two different hat-shaped signals over 100 simulations for each data length regularly sampled between $n = 100$ and $n = 1500$ on the log scale. In Figure 7, we plot the mean time for each algorithm with a 1-hat signal and two levels of noise in log-log scale. The results

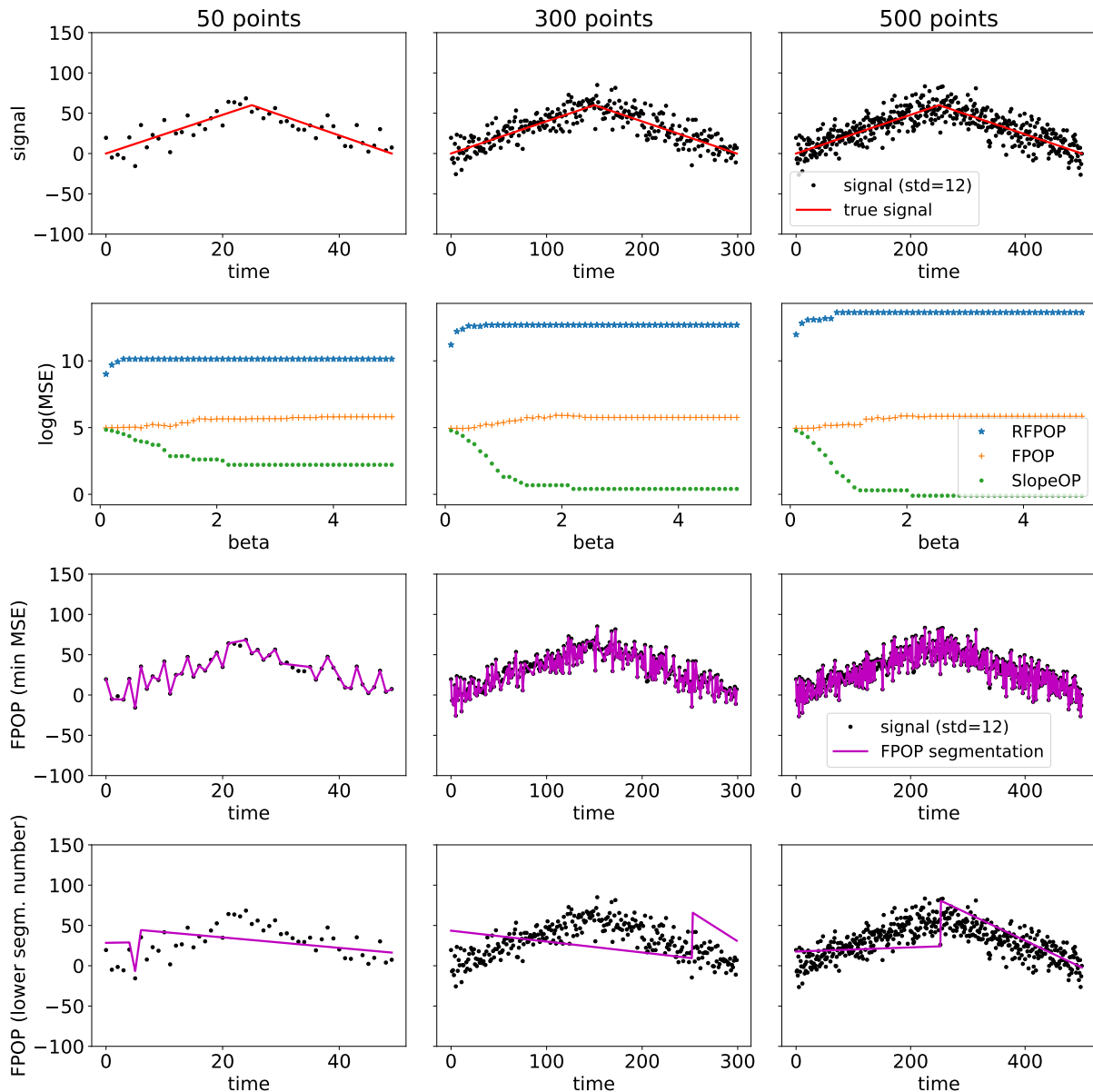


Figure 2: The first row presents 3 time-series of size 50, 300 and 500 that are obtained by the black hat-shaped signal with an additive Gaussian noise with $\sigma = 12$. For a range of values for $b = \frac{\beta}{\sigma^2 \log(n)}$ we plot the average $\log(\text{MSE})$ curve for algorithms FPOP, RFPOP and slopeOP for 10 simulated time series. The first two algorithms give performances far away from slopeOP. In the third row we show an example of an inferred signal by FPOP with a b value corresponding to the minimum in $\log(\text{MSE})$ curves. In last row we plot the segments obtained by the biggest possible b value before having a unique segment with the same data. We observe that we fail to obtain only 2 segments and that we obviously obtain an unrealistic result.

confirm the quadratic complexity for slopeOP and a complexity for CPOP between n^2 and n^3 (closer to n^3 with a higher level of noise). In this simulation, slopeOP outperforms CPOP for time series of length greater than 550 with $\sigma = 3$ and 1000 with $\sigma = 24$ (for example, with $n = 1500$ and $\sigma = 3$ we have a mean time of 32s for CPOP and 10s for slopeOP). The coefficient q in complexity $O(n^q)$ is equal to 2.88 and 2.94 for CPOP for $\sigma = 3$ and $\sigma = 24$,

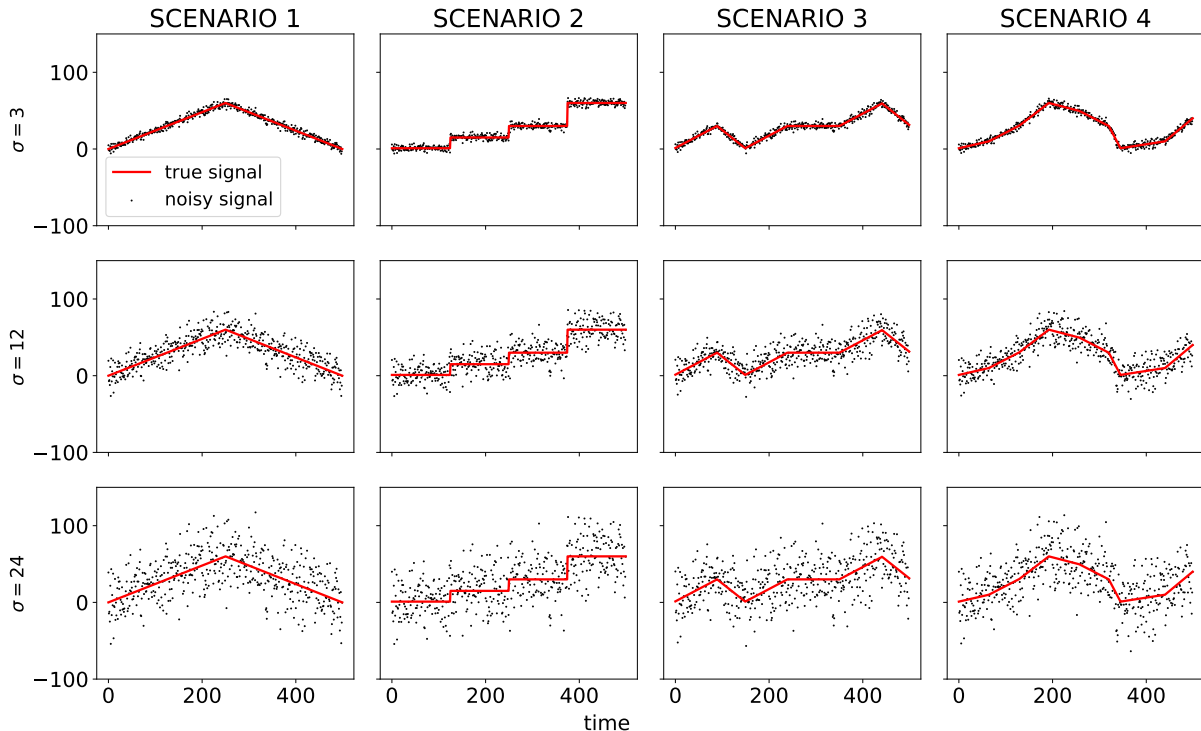


Figure 3: The plain black line is the true signal. For each scenario we simulate time series with an additive Gaussian noise of increasing standard deviation ($\sigma = 3, \sigma = 12, \sigma = 24$). Each plot shows a realisation of this procedure with 500 data points. The number of segments for scenarios 1 to 4 is respectively 2, 7, 6 and 8.

respectively. For slopeOP we get 1.95 and 2.04 with the channel pruning option. With a signal of 10 regular hats, CPOP is faster up to $n = 1500$ with power coefficient for CPOP $q_{\sigma=3} = 2.16$ & $q_{\sigma=24} = 2.47$ and for slopeOP $q_{\sigma=3} = 1.96$ & $q_{\sigma=24} = 2.03$. The corresponding plot is presented in Appendix E.

5.3 A Minimum Angle Between Consecutive Segments

We generate time series in scenario 1 of the hat-shaped signal with an heavy-tailed noise: a Student distribution with a degree of freedom equal to 3 with $\sigma = 24$. Simple computations give an angle of about 153° between the two segments. We choose to fix a minimal angle between segments to 130° . We explore the value of the MSE returned by the standard slopeOP algorithm compared with the same algorithm with smoothing option for a range of penalty values. We also return the number of inferred segments. Each data-point is the mean over 100 simulations.

Results presented in Figure 8 show that the MSE as well as the number of segments is lower with the min-angle option, whatever the penalty value. With a misspecified model or in presence of outliers, the minimum angle option (when we expect no small angle) improves the result of slopeOP. The penalty value has a reduced impact on the algorithm, which can be important for applications when the time series to analyse does not have a Gaussian noise structure.

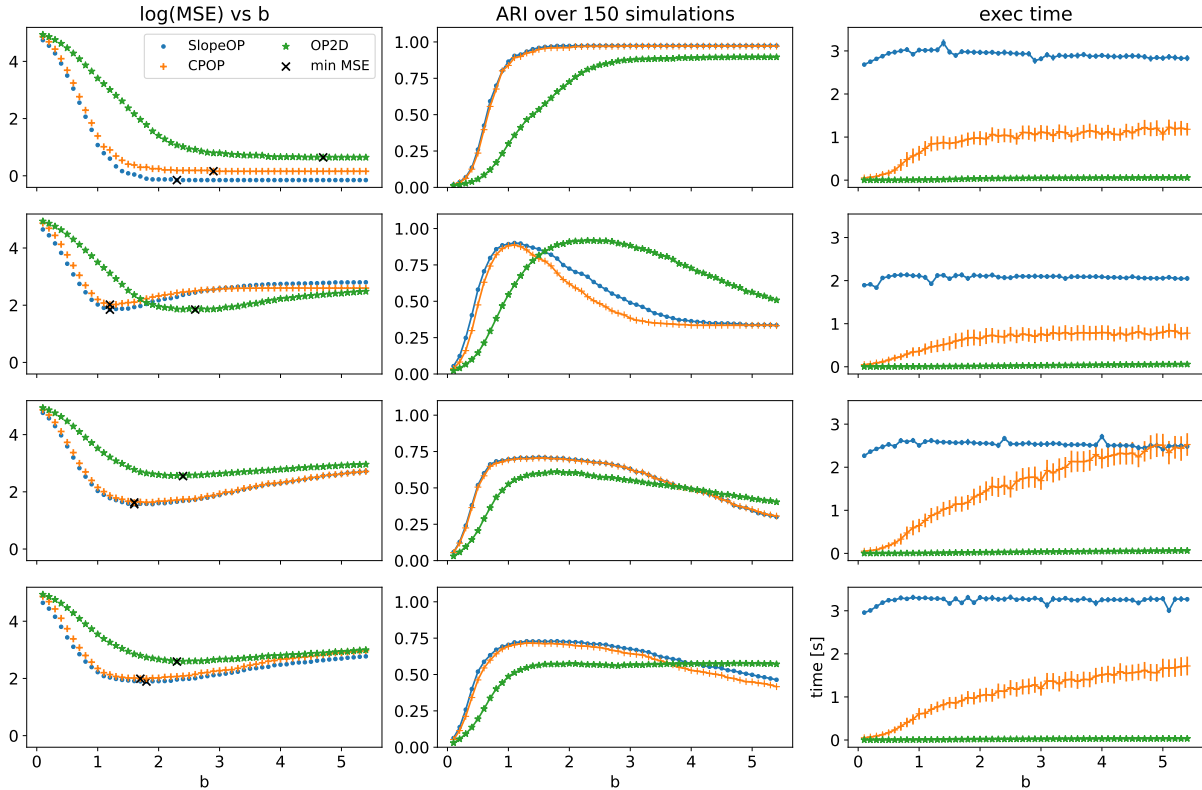


Figure 4: Results for scenarios 1 to 4 with noise $\sigma = 12$. Each point of the MSE curve is the mean over 150 independent simulations. Column 2 shows the Adjusted Rand Index (ARI) of the clustering obtained with SlopeOP and CPOP compared with the true signal. The last column shows the execution time vs b . The results of SlopeOP and CPOP are very similar. SlopeOP is slower than CPOP in most cases, but its complexity does not depend on b .

6 Application to Antibigram Image Analysis

Antibiotic Susceptibility Testing (AST) is a microbiology test used to guide antibiotic prescription in bacterial diseases by determining the susceptibility of bacteria to different antibiotics.

In disk diffusion AST, or antibiogram, cellulose disks impregnated with specific antibiotics are placed on the surface of Petri-dishes previously inoculated with a microorganism. Agar plates are incubated so that bacteria can grow everywhere, except around the cellulose disks that contain antibiotics to which the bacteria are susceptible. Then the diameter of the zone of non-growth (inhibition) surrounding each antibiotic disk is measured and compared to known minimum diameters to determine the susceptibility (the comparison is done with a sensibility of one millimeter)[24]. Diameters can be read by eye with a ruler, but several automatic reading systems exist which process pictures of incubated plates [36, 21].

The inhibition zone boundary is usually clear and easy to measure, but it can sometimes be hazy and fuzzy. In most cases, zone edges should be read at the point of complete inhibition[1], but determining this point can be challenging.

Several image processing procedures have been proposed for measuring inhibition diameters[18, 15, 7], but none of them focuses specifically on the issue of fuzzy borders. We propose the use of slopeOP to measure inhibition diameters in these cases. To our knowledge, a MSE-minimizing optimal segmentation algorithm has never been used to measure AST's inhibition diameters.

We designed a processing pipeline that uses slopeOP to read inhibition diameters from the

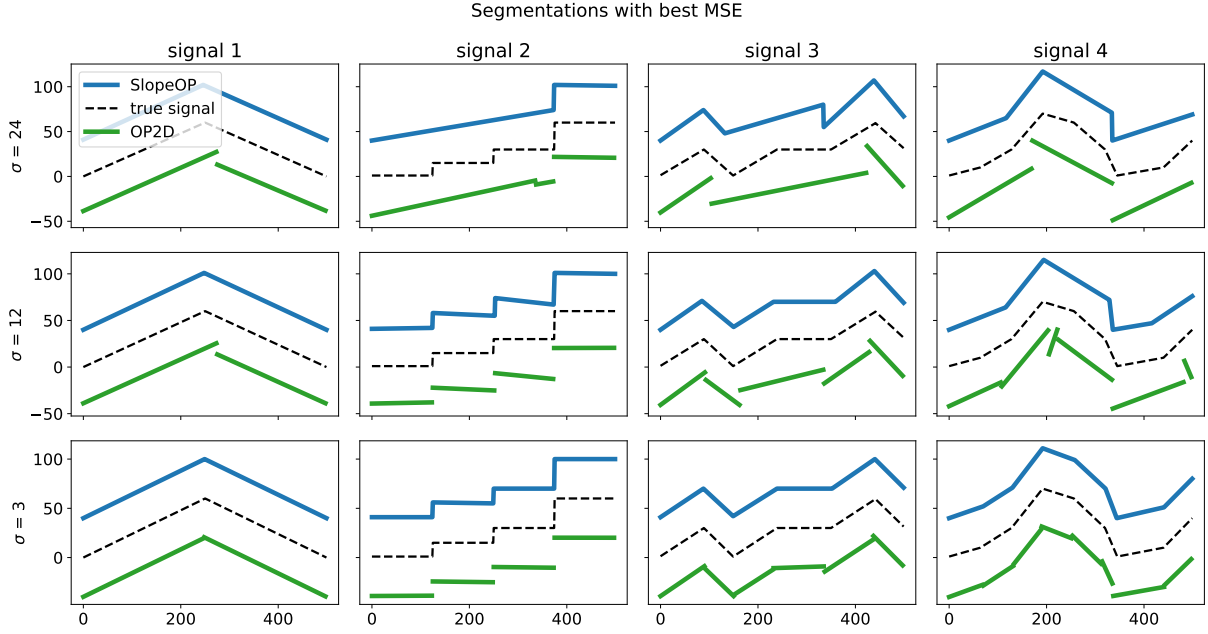


Figure 5: Segmentation obtained with values of b that minimise the MSE. These values correspond to the black crosses in figure 4. The curve are deliberately y-shifted for better appreciation of the shape differences.

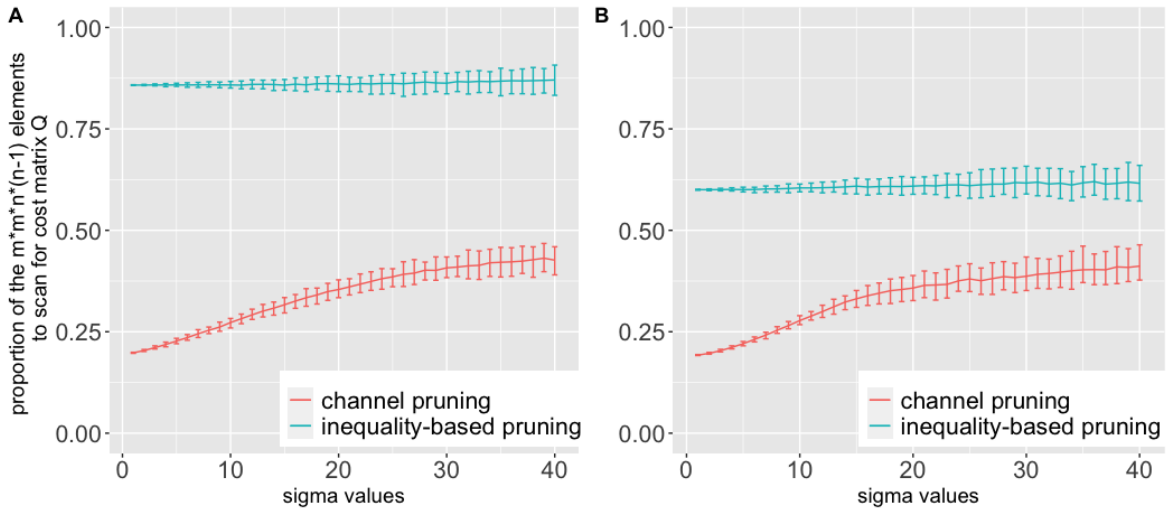


Figure 6: Without pruning the dynamic programming algorithm needs to scan $m^2n(n-1)$ elements for finding minima at each step. We compare the proportion of elements that the algorithm scans for channel-based and inequality-based prunings with different noise levels. In panel A we use a hat-shaped signal of length 500 with 1 hat and a signal with 25 regular hats in panel B. For each noise level, we simulate 100 time series and plot the error bar using the 95% empirical confidence interval. For all noise levels, the channel method is more efficient.

picture of an AST and tested it on one hundred inhibition zones presenting fuzzy borders. Fifty inhibition zones were taken from standard Mueller-Hinton growth medium antibiograms, the other fifty from blood agar antibiograms, which are darker and less contrasted. After normalizing the intensity of the AST picture, a sub-region of each inhibition zone was selected and centered on the the antibiotic disk. Then the radial intensity profile $I(r)$ was extracted:

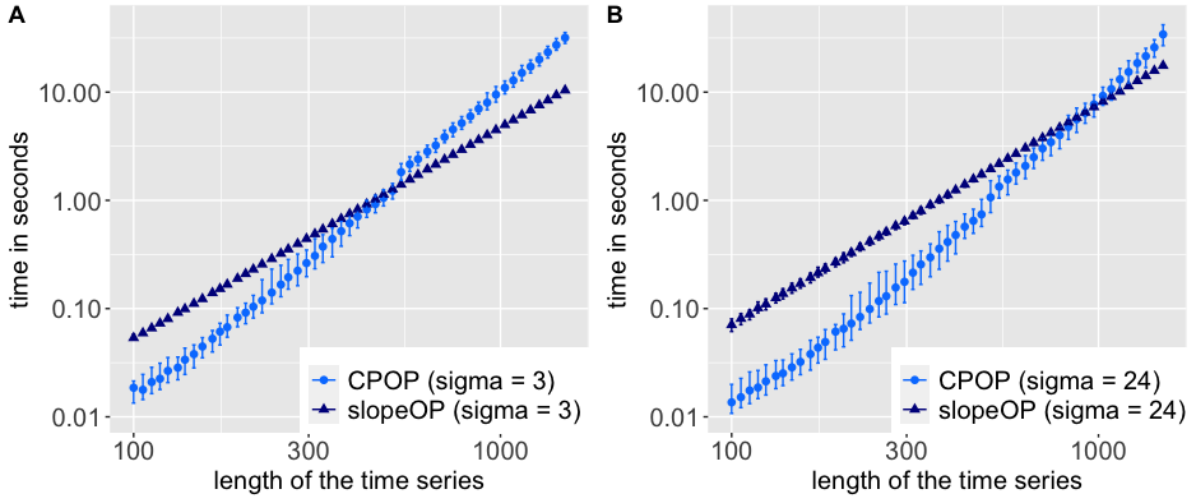


Figure 7: Computational time of CPOP versus slopeOP for one-hat-shaped data with noise level $\sigma = 3$ and $\sigma = 24$ in log-log scale. The coefficient q in complexity $O(n^q)$ is equal to 2.88 and 2.94 for CPOP for $\sigma = 3$ and $\sigma = 24$, respectively. For slopeOP we get 1.95 and 2.04 with the channel pruning option. We plot the error bar using the 95% empirical confidence interval.

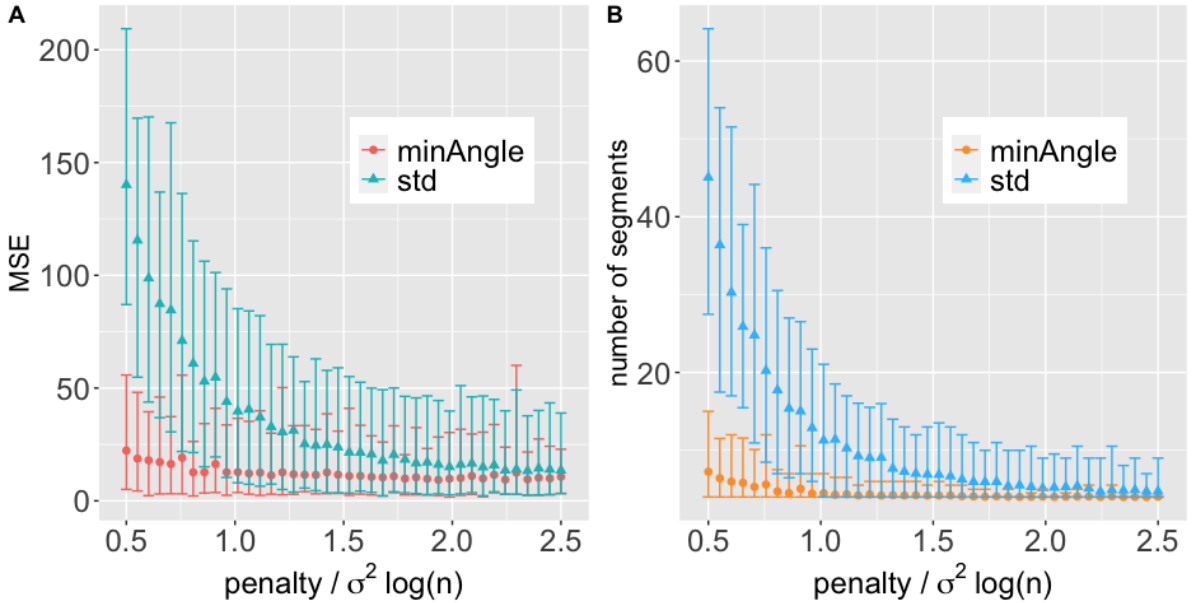


Figure 8: For a range of penalties from $\beta = 0.5\sigma^2 \log(n)$ to $\beta = 2.5\sigma^2 \log(n)$, 100 one-hat-shaped time series have been simulated with $n = 500$ for each penalty value and we ran slopeOP with a smoothing option ($minAngle = 130$) and without constraint (std). On the left panel A, we compare the MSE between the inferred signal and the true one for the two methods; on the right panel B, we compare the number of inferred segments for the two methods. Results highlight the stability with respect to the penalty of the algorithm with minimal angle option.

we recorded I , the average intensity value of the 10 most intense pixels (the scale of the images is $\simeq 10 \frac{pixel}{mm}$) lying at fixed distance r from the sub-region center (see figure 9). This method is a simple way to detect bacteria even if the inhibition zone is not a perfect disk. The average profile size in the data-set is 230 point.

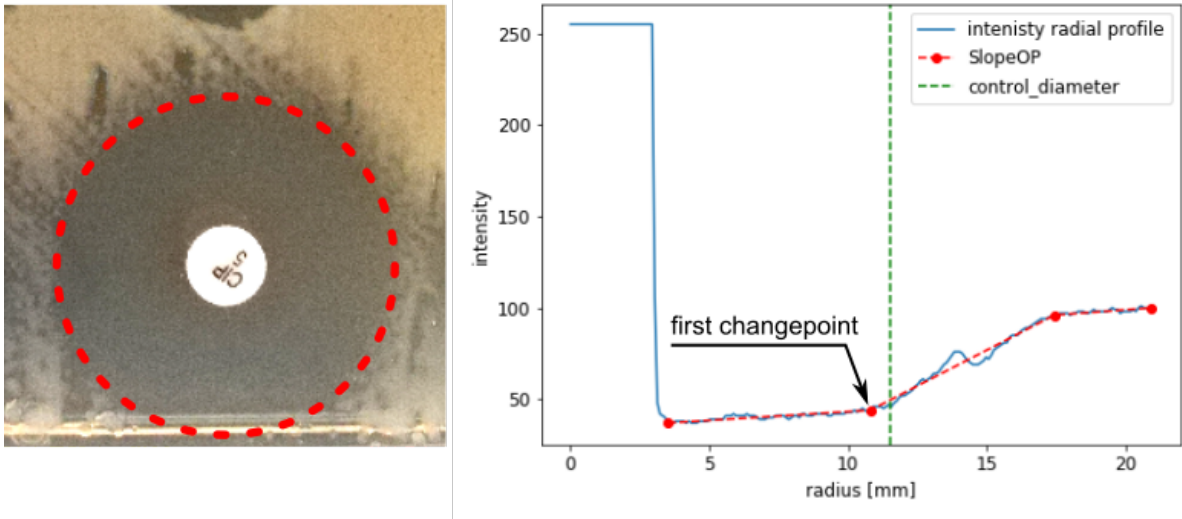


Figure 9: Picture of a part of an antibiogram showing an antibiotic disk and its inhibition zone (left). On the right a plot of the intensity radial profile and the segmentation with slopeOP with isotonic constraint.

slopeOP was used to segment the radial profile and determine the inhibition zone radius. For the segmentation with slopeOP we take the signal starting at 3.5mm, i.e. just after the plateau corresponding to the pellet disk (from this point, the ideal signal is supposed to be isotonic).

slopeOP is used with $\mathcal{S} = \{I(r)_{min}, \dots, I(r)_{max}\}$ and a penalty $\beta = 255$. We chose a high penalty in order to decrease the chance of detecting any changepoint before the bacterial halo (because of small imperfections in the sample or picture). As a comparison, we report in the appendix the results obtained with $\beta = 2\sigma^2 \log(n)$ (cfr **F**).

We tested both with and without the isotonic constraint (as the observed signal is supposed to be isotonic).

After the segmentation, we measure the inhibition radius r_{inib} as the distance corresponding to the first change-point. The inhibition diameter is obtained as $d_{inhib} = 2 \times r_{inhib}$.

As a comparison, we measured the diameters with the method suggested by Gavaille et al.[15] which uses a student t-test and considers both the intensity and the texture of the pixels around the antibiotic disk.

The so measured diameters were compared with manual (by eye) measurements, by calculating the absolute difference.

The results (Table 2 and Figure 10) show a good consistency with the manual measurements. In the case of Mueller-Hinton antibiograms, half of the diameter differences between our procedure and control are below the test sensibility (1mm) and 75% are smaller than 1.5mm, which is comparable with the acknowledged inter-operator variability due to the subjectivity of the measurement (± 1 mm) [20]. The results on the blood agar ASTs are slightly worse because of the lower contrast in the images and a consequent decreased signal-noise ratio. In both cases the isotonic constraint yields improved results.

Successively we tested the performance of slopeOP with isotonic constraint when reducing the density of states by 2,4 and 8. We defined the states space as $\mathcal{S} = \{I_{min}, I_{min} + s, I_{min} + 2s, \dots, I_{max}\}$ and repeated the measurements on the data-set at each value of $s = 2, 4, 8$. The results are reported in Table 3 and show a neat improvement in the execution speed by reducing the states density. Although the precision of the measurement decreases with decreasing states

AST type	algorithm	diameter diff quantiles		
		25%	50%	75%
M-H	CPOP	1.60	3.60	6.00
	SlopeOP	0.29	0.91	2.92
	SlopeOPi	0.24	0.69	1.71
	Ttest	0.50	1.50	3.25
blood	deltaCPOP	1.80	3.20	5.55
	deltaSlopeOP	0.72	1.72	3.00
	deltaSlopeOPi	0.47	1.28	2.38
	deltaTtest	2.00	9.50	16.75

Table 2: Agreement between auto and manual reading. We observe the distribution quantiles of $\Delta d = |d_a - d_c|$ the absolute difference between the automatically measured diameter d_a and the control value d_c . The smallest values are embolded.

agar	state density	diameter diff quantiles			speed	Changepoints
		25%	50%	75%		
M-H	1	0.24	0.69	1.71	1.00	4.25
	1/2	0.19	0.64	1.84	2.97	4.33
	1/4	0.21	0.77	1.55	7.74	4.24
	1/8	0.38	0.83	2.57	17.92	4.25
blood	1	0.47	1.28	2.38	1.00	3.92
	1/2	0.43	1.04	1.84	2.76	3.90
	1/4	0.70	1.52	2.34	6.69	3.94
	1/8	1.07	2.05	3.83	14.53	3.96

Table 3: Performance of slopeOP-isotonic at various states density. The calculation speed is reported as the inverse of the average execution time (normalized to density=1). Changepoints=average number of changepoints found. Mean execution time for slopeOP-iso1 is 5ms @ 2,3 GHz Intel Core i5.

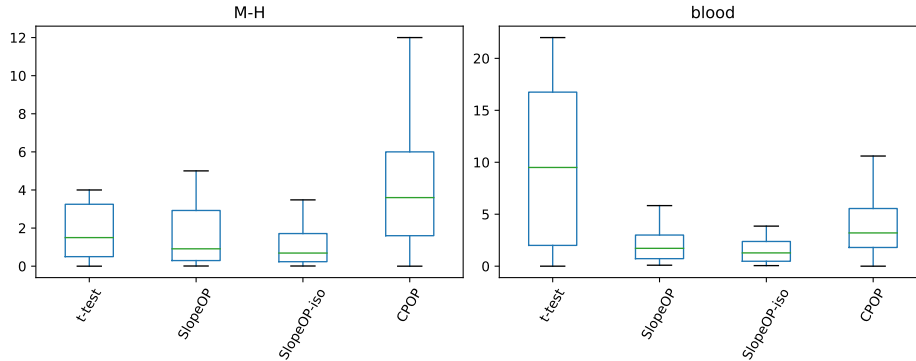


Figure 10: Distribution of absolute diameter differences across the tested algorithms. The isotonic constraint visibly improves the accuracy of the measurement.

density, the measured diameter are still in reasonable accord with the control values.

7 Conclusion

Multiple change-point detection for change in slope with continuity constraint is a challenging problem in time-series analysis. The dynamic programming approach developed in CPOP [11] exactly optimize the penalized likelihood but suffers from a quadratic-to-cubic time complexity in data length. We proposed a novel approach with quadratic complexity restricting the set of possible slope and intercept values to a finite set. In addition, this new problem allows us to introduce constraints on successive segments. State values for the spatial discretization has to be chosen for each time-series and leads to two difficulties: how close are we to the continuous solution of CPOP? How to make a appropriate choice of state values? These questions remains widely open and could be a subject for further developments. A natural extension of this work would consider a dynamic programming algorithm following at each iteration a two-continuous parameter function (instead of one as in CPOP). In this parametrization the challenging question of pruning rules is yet open.

Acknowledgement

We thank Guillem Rigail (LaMME Evry, IPS2 Paris-Saclay) for joint mentoring of our intern (co-author Nicolas Deschamps de Boishebert), for the careful reading of this work and the many useful feedbacks.

A Proof of Proposition 2.1

The update rule is obtained by direct computations (we write “vectors τ and s ” to mean the constraints in (2)):

$$\begin{aligned}
 Q_t(v) &= \min_{\substack{\text{vectors } \tau \text{ and } s \\ \tau_{k+1}=\ell, s_{k+1}=v}} \sum_{i=0}^k \{ \mathcal{C}(y_{(\tau_i+1):\tau_{i+1}}, s_i, s_{i+1}) + \beta \} - \beta \\
 &= \min_{\text{vectors } \tau \text{ and } s} \sum_{i=0}^{k-1} \{ \mathcal{C}(y_{(\tau_i+1):\tau_{i+1}}, s_i, s_{i+1}) + \beta \} - \beta + \mathcal{C}(y_{(\tau_k+1):t}, s_k, v) + \beta
 \end{aligned}$$

$$\begin{aligned}
&= \min_{\substack{0 \leq t' < t \\ s_{\min} \leq u \leq s_{\max}}} \min_{\substack{\text{vectors } \tau \text{ and } s \\ \tau_k = t', s_k = u}} \sum_{i=0}^{k-1} \{ \mathcal{C}(y_{(\tau_i+1):\tau_{i+1}}, s_i, s_{i+1}) + \beta \} - \beta + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta \\
&= \min_{\substack{0 \leq t' < t \\ s_{\min} \leq u \leq s_{\max}}} (Q_{t'}(u) + \mathcal{C}(y_{(t'+1):t}, u, v) + \beta) .
\end{aligned}$$

B Proof of Proposition 3.2

In order to prune we need to force the inequality

$$\mathcal{C}(y_{(t'+1):T}, u, v) \geq \mathcal{C}(y_{(t'+1):t}, u, v) + \mathcal{C}(y_{(t+1):T}, v, v),$$

to be true for all $T \geq t+1$ and fixed t', t, u, v . Using the cost definition (1), the inequality is equivalent to

$$\begin{aligned}
(v-u) \left(\sum_{i=t'+1}^t y_i \frac{(i-t')(T-t)}{(t-t')(T-t')} + \sum_{i=t+1}^{T-1} y_i \frac{T-i}{T-t'} \right) \\
\geq (v-u)(T-t) \left(\frac{u+2v}{6} + \frac{v-u}{12(t-t')(T-t')} \right)
\end{aligned}$$

and then

$$(v-u)g_t(T) \geq (v-u) \left((T-t') \left(\frac{u+2v}{6} \right) + \frac{v-u}{12(t-t')} - \frac{S'_t}{t-t'} \right) \quad (10)$$

with

$$g_t(T) = \begin{cases} \sum_{i=t+1}^{T-1} y_i \frac{T-i}{T-t} & \text{for } T = t+2, \dots, n, \\ 0 & \text{for } T = t+1. \end{cases}$$

As the right hand side of inequality (10) is linear in T we choose the following strategy: find coefficients $(\alpha_t^-, \alpha_t^+, \gamma_t^-, \gamma_t^+)$ of an upper and lower linear approximation such that

$$\alpha_t^+ T + \gamma_t^+ \leq g_t(T) \leq \alpha_t^- T + \gamma_t^-, \quad T = t+1, \dots, n.$$

We then introduce the linear-in- T functions f^- and f^+ defined in the Proposition. To prove that the inequality $f^+(T) \geq 0$ holds for all $T = t+1$ to n we only need to have $f^+(t+1) \geq 0$ and $f^+(n) \geq 0$ due to the linearity of f^+ . With the same argument for f^- the result is proven.

C Variance estimation: complement with 10 segments

With smaller segments as presented in Figure 11, the estimator "HALL diff" remains the more efficient but overestimates the variance in case of a sinusoidal signal, in particular for lower noise levels. Nevertheless, this estimation remains less biased than the two others.

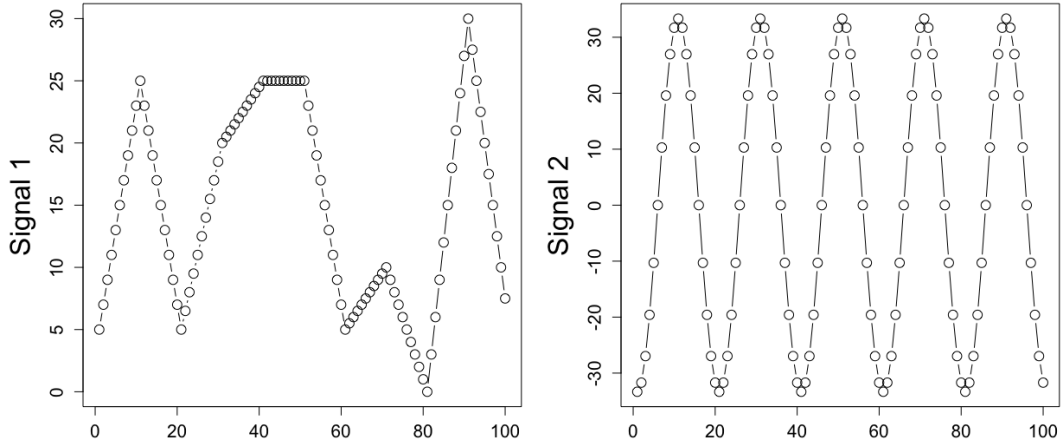


Figure 11: Linear and sinusoidal signals of length 100 with 10 segments.

method	signal	σ	1	2	3	4	5
MAD	slope (signal 1)	$\text{mean}(\hat{\sigma})$	1.71	2.37	3.26	4.19	5.163
		$\text{sd}(\hat{\sigma})$	0.15	0.26	0.39	0.52	0.65
	sinus (signal 2)	$\text{mean}(\hat{\sigma})$	7.60	7.30	7.21	7.37	7.75
		$\text{sd}(\hat{\sigma})$	0.26	0.35	0.45	0.56	0.67
HALL	slope (signal 1)	$\text{mean}(\hat{\sigma})$	2.71	3.21	3.92	4.72	5.59
		$\text{sd}(\hat{\sigma})$	<i>0.056</i>	<i>0.13</i>	<i>0.2</i>	<i>0.28</i>	<i>0.37</i>
	sinus (signal 2)	$\text{mean}(\hat{\sigma})$	11.2	11.3	11.5	11.8	12.2
		$\text{sd}(\hat{\sigma})$	0.048	<i>0.097</i>	<i>0.15</i>	<i>0.21</i>	<i>0.27</i>
HALL Diff	slope (signal 1)	$\text{mean}(\hat{\sigma})$	1.18	2.09	3.06	4.04	5.02
		$\text{sd}(\hat{\sigma})$	0.085	0.17	0.26	0.35	0.44
	sinus (signal 2)	$\text{mean}(\hat{\sigma})$	2.43	2.98	3.72	4.56	5.46
		$\text{sd}(\hat{\sigma})$	<i>0.039</i>	0.12	0.22	0.31	0.40

Table 4: Variance estimation with MAD, HALL and HALL Diff estimators based on the two signals presented in Figure 11. The closest values to the true sigma are in bold; the smallest standard deviations in italic. We simulated 10^4 time-series for each experiment. We notice that all estimators struggle to return an unbiased estimation with the sinusoidal signal, in particular with the MAD and HALL estimators for which the estimation seems no more sigma dependent.

D Simulation Results for the 4 Scenarios

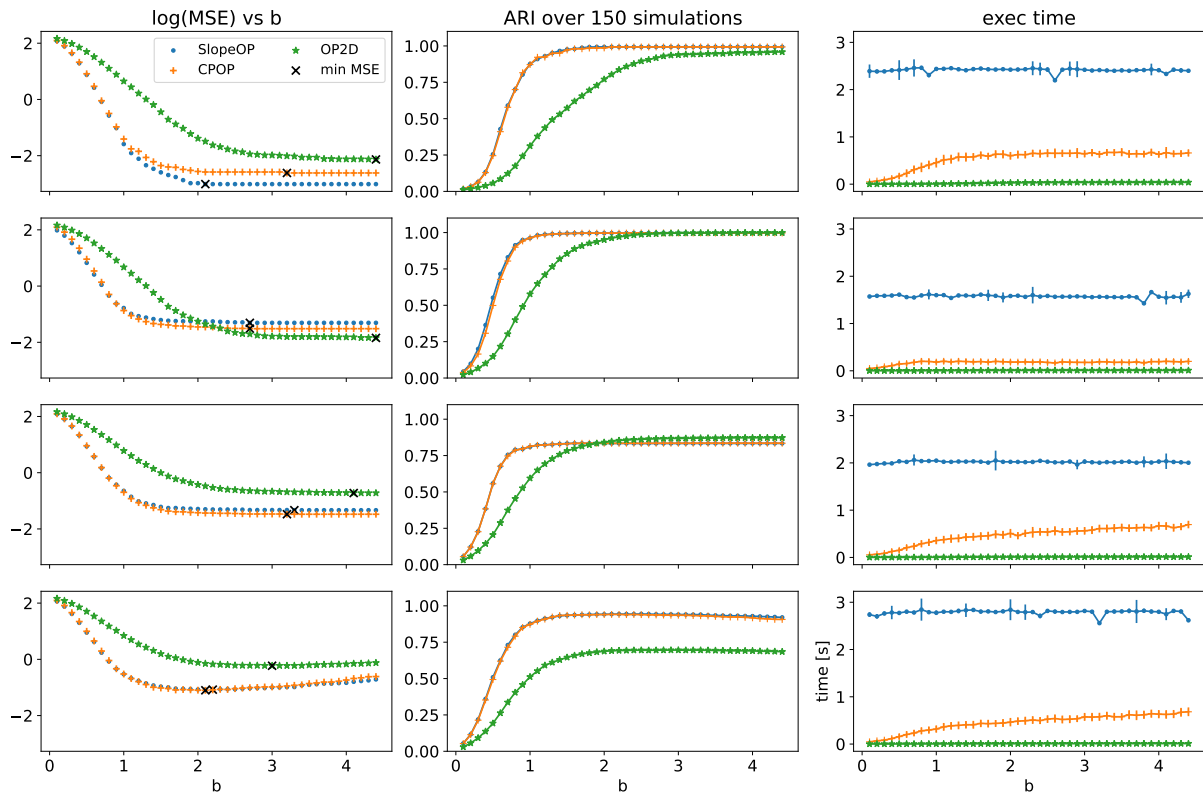


Figure 12: Results for scenarios 1 to 4 with noise $\sigma = 3$

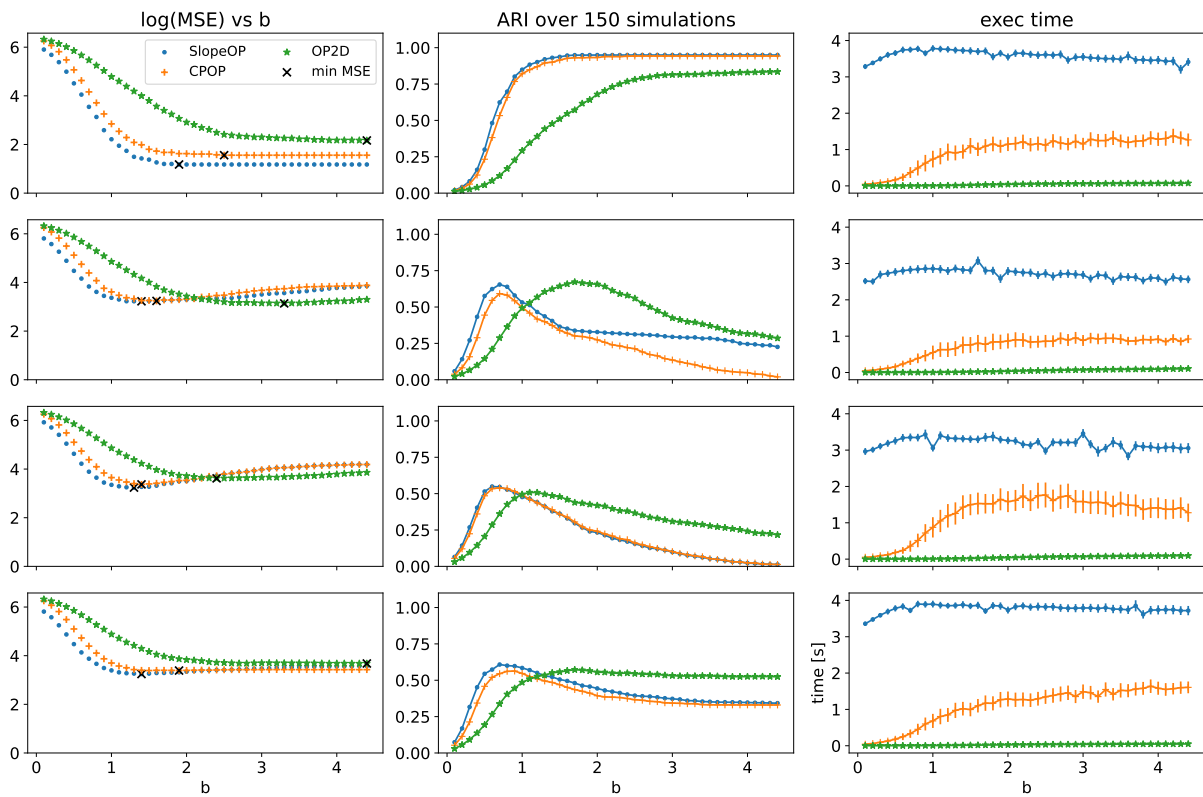


Figure 13: Results for scenarios 1 to 4 with noise $\sigma = 24$

E Time complexity with 10-hat-shaped signals

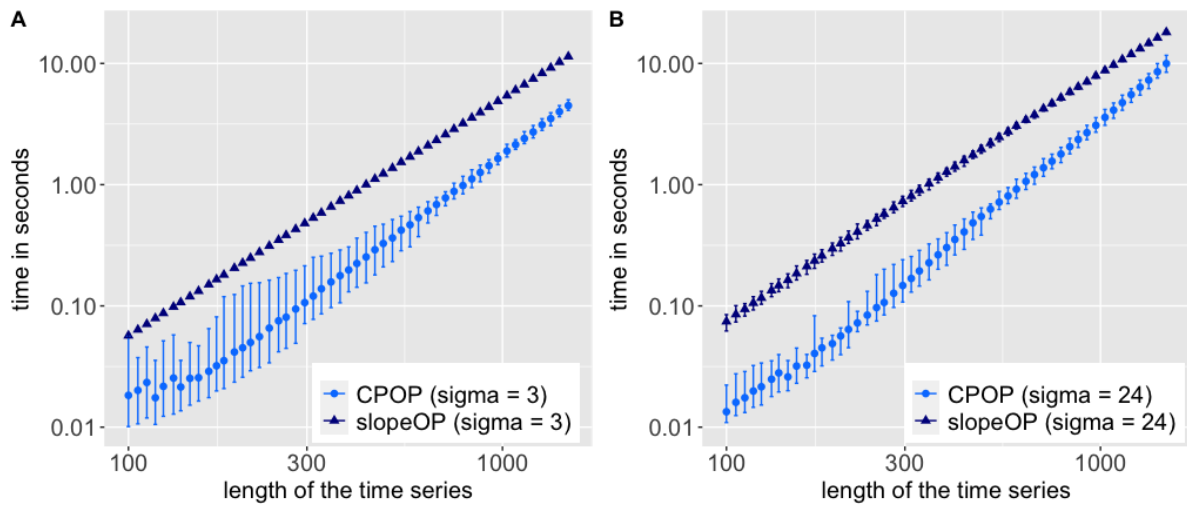


Figure 14: Computational time of CPOP versus slopeOP for 10-hat-shaped data with noise level $\sigma = 3$ and $\sigma = 24$ in log-log scale. The coefficient q in complexity $O(n^q)$ is equal to 2.16 and 2.47 for CPOP for $\sigma = 3$ and $\sigma = 24$, respectively. For slopeOP we get 1.96 and 2.03 with the channel pruning option.

AST type	algorithm	diameter diff. quantiles		
		25%	50%	75%
blood	deltaCPOP	8.00	12.50	18.00
	deltaSlopeOP	8.35	12.10	15.28
	deltaSlopeOPi	1.63	3.44	7.23
	deltaTtest	2.00	9.50	16.75
M-H	deltaCPOP	9.00	16.00	18.00
	deltaSlopeOP	6.46	12.05	17.01
	deltaSlopeOPi	1.89	8.49	15.56
	deltaTtest	0.50	1.50	3.25

Table 5: Agreement of auto and manual reading.

F AST application - complementary results

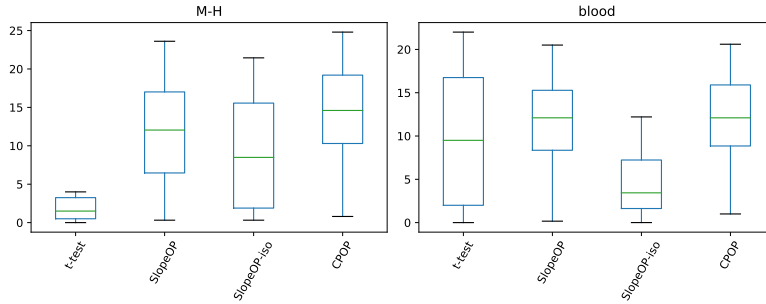


Figure 15: Distribution of absolute diameter differences across the tested algorithms. For M-H antibiograms, Gavaille’s t-test method shows the best results, whereas in the case of blood enriched culture medium, SlopeOP with isotonic constraint outperforms the others. Both SlopeOP unconstrained and CPOP give poor measurements, underlining the fact that the isotonic constraint improves the results

We repeated the automatic experiment of Section 6 with a variable penalty calculated as $\beta = 2\sigma^2 \log(n)$ where the variance σ^2 is estimated with the HallDiff estimator presented in this paper (cfr. 4). The average penalty value obtained is 4 which results in the detection of more changepoints in the signal (compared to Section 6. In this case, the assumption that the first detected changepoint is close the bacterial boundary is not appropriate and results in much worse measurement for all tested segmentation algorithms. Nevertheless, the results (Table 5 and Figure 15) show a neat improvement in the measurement accuracy when using the isotonic constraint. In the case of Mueller-Hinton antibiotics, even if the isotonic constraint ameliorates the result of the segmentation, none of the proposed segmentation algorithms achieves good measurements compared to Gavaille’s t-test method. In the case of blood enriched culture medium instead, where the t-test method gives less accurate results, the isotonic constraint of SlopeOP shows the best measurement accuracy.

	agar	states density	abs. diameter difference			speed	av. changepoints number
			25%	50%	75%		
blood	1		1.63	3.44	7.23	1.00	7.24
	1/2		0.93	4.43	7.87	2.87	5.60
	14		2.03	4.20	6.46	7.41	8.76
	1/8		1.76	3.04	6.97	17.02	8.14
mh	1		1.89	8.49	15.56	0.43	9.76
	1/2		2.54	8.65	15.19	1.00	8.29
	14		1.12	5.00	11.15	2.80	10.98
	1/8		1.94	3.69	10.62	6.96	10.82

Table 6: Performance of slopeOP-isotonic at various states density. The calculation speed is reported as the inverse of the average execution time (normalized to density=1). Change-points=average number of changepoints found. NOTE: Mean execution time for slopeOP-iso1 is 5ms @ 2,3 GHz Intel Core i5.

References

- [1] Reading guide - eucast disk diffusion method for antimicrobial susceptibility testing. page 3. EUCAST, 2020.
- [2] R. Baranowski, Y. Chen, and P. Fryzlewicz. Narrowest-over-threshold detection of multiple change points and change-point-like features. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(3):649–672, 2019.
- [3] C. Beaulieu, J. Chen, and J. L. Sarmiento. Change-point analysis as a tool to detect abrupt climate variations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1962):1228–1249, 2012.
- [4] R. E. Bellman and S. Dreyfus. Applied dynamic programming, princeton unlv. *Press, Princeton, NJ*, 1962.
- [5] N. Cahill, S. Rahmstorf, and A. C. Parnell. Change points of global temperature. *Environmental Research Letters*, 10(8):084002, 2015.
- [6] H. Cao and W. Biao Wu. Changepoint estimation: another look at multiple testing problems. *Biometrika*, 102(4):974–980, 2015.
- [7] L. F. Costa, E. S. da Silva, V. T. Noronha, I. Vaz-Moreira, O. C. Nunes, and M. M. de Andrade. Development of an automatic identification algorithm for antibiogram analysis. *Computers in biology and medicine*, 67:104–115, 2015.
- [8] I. Cribben and Y. Yu. Estimating whole-brain dynamics by using spectral clustering. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(3):607–627, 2017.
- [9] R. A. Davis, T. C. M. Lee, and G. A. Rodriguez-Yam. Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101(473):223–239, 2006.
- [10] T. Dylan Hocking, G. Rigaiil, P. Fearnhead, and G. Bourque. A log-linear time algorithm for constrained changepoint detection. *arXiv e-prints*, pages arXiv-1703, 2017.

- [11] P. Fearnhead, R. Maidstone, and A. Letchford. Detecting changes in slope with an l_0 penalty. *Journal of Computational and Graphical Statistics*, pages 1–11, 2018.
- [12] P. Fearnhead and G. Rigai. Relating and comparing methods for detecting changes in mean. *Stat*, page e291.
- [13] P. Fearnhead and G. Rigai. Change-point detection in the presence of outliers. *Journal of the American Statistical Association*, 9 2017.
- [14] P. Fryzlewicz. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.
- [15] A. Gavaille, B. Bardy, and A. Andremont. Measurement of inhibition zone diameter in disk susceptibility tests by computerized image analysis. *Computers in biology and medicine*, 24(3):179–188, 1994.
- [16] S. W. Hainsworth and M. D. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Advances in Signal Processing*, 2004(15):1–11, 2004.
- [17] P. Hall, J. Kay, and D. Titterinton. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, 77(3):521–528, 1990.
- [18] G. Hejblum, V. Jarlier, J. Grosset, and A. Aurengo. Automated interpretation of disk diffusion antibiotic susceptibility tests with the radial profile analysis algorithm. *Journal of clinical microbiology*, 31(9):2396–2401, 1993.
- [19] T. Hocking, G. Rigai, and G. Bourque. Peakseg: constrained optimal segmentation and supervised penalty learning for peak detection in count data. In *International Conference on Machine Learning*, pages 324–332, 2015.
- [20] M. Hombach, F. P. Maurer, T. Pfiffner, E. C. Böttger, and R. Furrer. Standardization of operator-dependent variables affecting precision and accuracy of the disk diffusion method for antibiotic susceptibility testing. *Journal of clinical microbiology*, 53(12):3864–3869, 2015.
- [21] M. Hombach, R. Zbinden, and E. C. Böttger. Standardisation of disk diffusion results for antibiotic susceptibility testing using the sirscan automated zone reader. *BMC microbiology*, 13(1):225, 2013.
- [22] B. Jackson, J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumoussis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, 2005.
- [23] S. Jamali, P. Jönsson, L. Eklundh, J. Ardö, and J. Seaquist. Detecting changes in vegetation trends using time series segmentation. *Remote Sensing of Environment*, 156:182–195, 2015.
- [24] J. H. Jorgensen and J. D. Turnidge. Susceptibility test methods: Dilution and disk diffusion methods*. In *Manual of Clinical Microbiology, Eleventh Edition*, pages 1253–1273. American Society of Microbiology, 2015.
- [25] R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of change-points with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

- [26] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM review*, 51(2):339–360, 2009.
- [27] L. Koepcke, G. Ashida, and J. Kretzberg. Single and multiple change point detection in spike trains: comparison of different cusum methods. *Frontiers in systems neuroscience*, 10:51, 2016.
- [28] R. Maidstone, T. Hocking, G. Rigaiil, and P. Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2):519–533, Mar 2017.
- [29] A. F. Militino, M. Moradi, and M. D. Ugarte. On the performances of trend and change-point detection methods for remote sensing data. *Remote Sensing*, 12(6):1008, 2020.
- [30] National Research Council. *Frontiers in massive data analysis*, 2013.
- [31] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.
- [32] P. Papastamoulis, T. Furukawa, N. Van Rhijn, M. Bromley, E. Bignell, and M. Rattray. Bayesian detection of piecewise linear trends in replicated time-series with application to growth data modelling. *The international journal of biostatistics*, 16(1), 2020.
- [33] J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of applied meteorology and climatology*, 46(6):900–915, 2007.
- [34] G. Rigaiil. A pruned dynamic programming algorithm to recover the best segmentations with 1 to k_{\max} change-points. *Journal de la Société Française de Statistique*, 156(4):180–205, 2015.
- [35] A. Robertson. Decoding tempo and timing variations in music recordings from beat annotations. In *ISMIR*, pages 475–480, 2012.
- [36] M. Sanchez, B. S. Del Saz, E. Loza, F. Baquero, and R. Canton. Evaluation of the osiris video reader system for disk diffusion susceptibility test reading. *Clinical microbiology and infection*, 7(7):352–357, 2001.
- [37] N. Schütz and M. Holschneider. Detection of trend changes in time series using bayesian inference. *Physical Review E*, 84(2):021120, 2011.
- [38] A. J. Scott and M. Knott. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 30(3):507–512, 1974.
- [39] A. Sen and M. S. Srivastava. On tests for detecting change in mean. *The Annals of statistics*, pages 98–108, 1975.
- [40] S. Sharma, D. A. Swayne, and C. Obimbo. Trend analysis and change point techniques: a survey. *Energy, Ecology and Environment*, 1(3):123–130, 2016.
- [41] Y. Sowa, A. D. Rowe, M. C. Leake, T. Yakushi, M. Homma, A. Ishijima, and R. M. Berry. Direct observation of steps in rotation of the bacterial flagellar motor. *Nature*, 437(7060):916–919, 2005.
- [42] R. J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

- [43] S. Tickle, I. Eckley, P. Fearnhead, and K. Haynes. Parallelization of a common changepoint detection method. *Journal of Computational and Graphical Statistics*, 29(1):149–161, 2020.
- [44] R. Todd Ogden and G. L. Collier. On detecting and modeling deterministic drift in long run sequences of tapping data. *Communications in Statistics-Theory and Methods*, 28(3-4):977–987, 1999.
- [45] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.
- [46] Y.-C. Yao. Estimating the number of change-points via schwarz’criterion. *Statistics & Probability Letters*, 6(3):181–189, 1988.
- [47] M. Yu and E. Ruggieri. Change point analysis of global temperature records. *International Journal of Climatology*, 39(8):3679–3688, 2019.
- [48] N. R. Zhang and D. O. Siegmund. A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1):22–32, 2007.
- [49] C. Zheng, I. A. Eckley, and P. Fearnhead. Consistency of a range of penalised cost approaches for detecting multiple changepoints. *arXiv preprint arXiv:1911.01716*, 2019.