Relating and comparing methods for detecting changes in mean

Paul Fearnhead1 | Guillem Rigaill2,3,4

1Department of Mathematics and Statistics, Lancaster University, Lancaster, LA1 4YF, UK
2Université Paris-Saclay, CNRS, INRAE, Univ Evry, Institute of Plant Sciences Paris-Saclay (IP52), Orsay, 91405, France
3Université de Paris, CNRS, INRAE, Institute of Plant Sciences Paris-Saclay (IP52), Orsay, 91405, France
4Université Paris-Saclay, CNRS, Univ Evry, Laboratoire de Mathématiques et Modélisation d’Evry, Evry, 91037, France

Correspondence
Paul Fearnhead, Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, UK.
Email: p.fearnhead@lancaster.ac.uk

Funding information
Saclay Plant Sciences; Engineering and Physical Sciences Research Council, Grant/Award Number: EP/N031938/1; Genopole, Grant/Award Number: ATIGE grant

1 INTRODUCTION

In recent years, there has been an explosion in methods for detecting changepoints (also known as breakpoints or structural breaks) in data streams (see Truong, Oudre, & Vayatis, 2019; Aue & Horváth, 2013, for recent reviews of the area). This has been motivated by the wide range of applications where changepoint methods are needed. These include genomics (Muggeo & Adelfio, 2010; Cao & Wu, 2015), neuroscience (Koepcke, Ashida, & Kretzberg, 2016), climate science (Reeves, Chen, Wang, Lund, & Lu, 2007), and econometrics (Bai, 1997), amongst many others.

An important question for a practitioner is how to choose a suitable method for their specific application. This is made difficult by the often contradictory empirical results across the various papers on changepoint methods.

All methods for detecting changepoints also require a practitioner to set some threshold. Although the form of threshold will vary between methods, in all cases, it governs how much evidence for a changepoint is needed before that method will detect it. Our experience is that the choice of this threshold will often be more important, in terms of the accuracy of the results, than the choice of the method. Furthermore, and as we discuss later, the disparity in the empirical results across different papers appears to be driven by different papers using different choices for these thresholds.

This paper focuses on the simplest, but most widespread, changepoint problem of detecting changes in mean in a univariate data stream. We overview some of the more popular approaches to this problem, with the aim of emphasizing the statistical similarities between them and, in particular, pointing out links between threshold choices for different methods. These links can be used when performing comparisons between methods to ensure comparable thresholds are used for the different methods being tested. We also provide a thorough comparison of these methods on a wide range of scenarios, taken from the literature. R code is provided in the Supporting Information, and the full results are available at https://github.com/guillemr/SomeSegSimu, so that an interested user can extend this study to other scenarios, other measures of evaluating the accuracy of a set of estimated changepoints, or new methods.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. Stat published by John Wiley & Sons, Ltd.

Stat. 2020;9:e291.
https://doi.org/10.1002/sta4.291

KEYWORDS
binary segmentation, breakpoints, dynamic programming, model error, penalized likelihood, structural breaks
Most existing methods come with theoretical guarantees, which has motivated default threshold values. However, this theory, and the validity of the default threshold values, often is based on strong distributional assumptions for the data. In particular, defaults are based on assuming that the noise in the data is independent, identically distributed (IID), and Gaussian or sub-Gaussian. Our results show that when this assumption is true, all methods perform reasonably well in terms of estimating the number and position of the changepoints. However, all methods are sensitive to violations of this assumption, and the presence of heavier tailed noise and/or positive auto-correlation in the noise can lead to over-estimation of the number of changepoints. Methods that try to compare the fit with the data obtained for a range of threshold values seem to be more reliable across a range of scenarios.

2 | DETECTING A SINGLE CHANGE

To understand the ideas behind various multiple changepoint methods, and their similarities and differences, it is helpful to first consider the simpler problem of detecting a single change. We are focusing on detecting changes in mean of univariate data, and so we assume we have data \( y_{1:n} = (y_1, \ldots, y_n) \), with \( y_t = \mu_t + \epsilon_t \) for \( t = 1, \ldots, n \). Here, \( \mu_t \) is the mean of the \( t \)th data point, and \( \epsilon_{1:n} \) is a zero-mean error process. We wish to compare two models for the data:

(M0) There is no changepoint, so \( \mu_t = \theta_0 \) for \( t = 1, \ldots, n \).

(M1) There is a changepoint at some unknown time \( \tau \in \{1, \ldots, n-1\} \), such that \( \mu_t = \theta_0 \) for \( t \leq \tau \) and \( \mu_t = \theta_1 \) for \( t > \tau \), with \( \theta_1 \neq \theta_0 \).

If we decide there is a changepoint, we would then wish to estimate its location, \( \tau \).

The most common approach to this problem is to model the noise process as consisting of IID Gaussian random variables with variance \( \sigma^2 \) and then perform a likelihood ratio test between the two models. If we define \( \gamma(y; \theta, \sigma) = \frac{1}{\sigma^2} (y - \theta)^2 + \log \sigma^2 \), to be minus twice the log-likelihood for an observation, \( y \), under our Gaussian model with mean \( \theta \) and variance \( \sigma^2 \), then the log-likelihood ratio statistic for testing between M0 and M1 if \( \sigma^2 \) is known is

\[
\text{LR}_1 = \min_{\tau \in \{1, \ldots, n-1\}} \left\{ \sum_{t=1}^{\tau} \gamma(y_t; \theta_0, \sigma) + \sum_{t=\tau+1}^{n} \gamma(y_t; \theta_1, \sigma) \right\}
\]

where \( y_{\tau:t} \) denotes the sample mean of \( y_{\tau:t} \). As \( \gamma \) is defined in terms of the negative log-likelihood, we are minimizing it, which is equivalent to maximizing the log-likelihood. The last equality comes from straightforward algebraic manipulation and motivates the cusum statistic

\[
C_\tau = \left( \frac{\tau(n-\tau)}{\sigma^4} \right)^{1/2} |\bar{y}_{1:\tau} - \bar{y}_{\tau+1:n}|.
\]

which is the natural estimate of the absolute change in mean of the data before and after \( \tau \), scaled to have a variance that does not depend on \( \tau \) or \( n \). The likelihood ratio statistic, \( \text{LR}_1 \), is monotonically increasing in \( \max C_\tau \).

If we perform the likelihood ratio test under the assumption that \( \sigma \) is unknown, straightforward calculation gives the test statistic in terms of the cusum statistic as

\[
\text{LR}_2 = \max_{\tau} n \log \left\{ \frac{1}{n} \left( 1 - \frac{C_\tau^2}{\sum_{t=1}^{n} (y_t - \bar{y}_{1:n})^2} \right)^{-1} \right\}.
\]

This test statistic is also monotonically increasing in \( \max C_\tau \).

To perform a test, we then need to specify a threshold, \( \beta \), say, such that we detect a change if, say, \( \text{LR}_1 > \beta \). If we detect a change, the natural estimator of the location of the change will be the arg-max for \( \tau \) within the calculation of the likelihood ratio statistic, or, equivalently, the value of \( \tau \) for which \( C_\tau \) is the largest. As both test statistics are monotonic in \( \max C_\tau \), the form of the test will be the same regardless of whether we assume \( \sigma^2 \) is known or not. Furthermore, both of these tests will be equivalent to a test based on the cusum statistic, that is, that detects a changepoint if \( \max C_\tau \) is larger than some pre-specified threshold. It is also clear that there will be a correspondence between thresholds such that a test based on \( \text{LR}_1 \) will be identical to the one based on \( \text{LR}_2 \) and also to the one based on \( \max C_\tau \).

To specify an appropriate threshold for one of these tests requires knowledge of the distribution of the test statistic under the null assumption of no changepoint. This distribution will depend on the distribution of the noise variables. The likelihood ratio test is non-regular, and so asymptotically does not have a chi-squared distribution under the null, even if the noise variables are IID Gaussians. Instead, under the Gaussian assumption the null distribution can be related to the maximum of a scaled Brownian bridge process (e.g., Hinkley, 1971). Importantly, these distributions may be inaccurate in practice if either of the Gaussian or independence assumptions do not hold.

3 | DETECTING MULTIPLE CHANGES

We now focus on the more challenging problem of detecting multiple changes in mean. As before, we have data \( y_{1:n} \) with \( y_t = \mu_t + \epsilon_t \) for \( t = 1, \ldots, n \). Let \( m' \) be the, unknown, number of changepoints and \( \tau_{1:m'} \) be their locations. If we further define \( \tau_0 = 0 \) and \( \tau_{m'+1} = n \) then we
can define \( m^* + 1 \) segments, with the \( j \)th segment containing observations \( y_{i_j+1:t_j} \) for \( j = 0, \ldots, m^* \). We let \( \theta_j^* \) be the mean for data within this segment, so \( \mu_i = \theta_j^* \) if and only if \( i_0^* < i \leq i_{j+1}^* \).

We now overview a range of methods for estimating \( m^* \) and \( \tau_{1:m^*}^* \). Each of these can be viewed as a way of extending the approach, described above, for detecting a single change in mean to this multiple changepoint setting. To aid comparison, for some methods, we will state asymptotic theoretical properties. These will tend to relate to the performance of the method under in-fill asymptotics, that is where we fix \( m^* \), the proportion of the data within each segment, that is, \( \tau_j^*/n \), and the segment means as \( n \to \infty \). Unless stated otherwise, results will further assume the noise random variables are IID Gaussians. A method is consistent if its estimators \( \hat{m} \) and \( \hat{\tau}_{1:M} \) satisfy that \( \hat{m} = m^* \) and \( \max |\hat{\tau}_j - \tau_j^*|/n \to 0 \) with probability tending to 1 as \( n \to \infty \).

### 3.1 Binary segmentation and variants

Possibly the simplest way of extending a method for detecting a single changepoint to the multiple changepoint setting is to repeatedly apply such a method. We first apply the method to detect a single change in the data. If we detect one, we can split the data at the inferred changepoint location and apply the method separately to data before and after the change. This can be repeated each time we detect a change, until no further changepoints are found. This approach is called binary segmentation (Scott & Knott, 1974).

Implementing binary segmentation requires specifying a threshold for our test statistic that we use for detecting a change. If we are willing to make assumptions on the distribution of the noise, this choice can be guided by the asymptotic distribution for the test of a single change. However, even in this case, the choice of threshold is still complicated by the multiple tests that are being performed within binary segmentation, which makes it hard to relate the significance level of the test for a single change to inferential properties of the procedure when detecting multiple changepoints. There is theory showing that binary segmentation can consistently estimate the number of changepoints under in-fill asymptotics (Venkatraman, 1992; Fryzlewicz, 2014).

One issue with binary segmentation is that, because it adds changes one at a time, it can have low power to detect short segments in the data (Venkatraman, 1992; Olshen, Venkatraman, Lucito, & Wigler, 2004). This has led to a number of adaptations of binary segmentation (Olshen et al., 2004; Fryzlewicz, 2014; Baranowski, Chen, & Fryzlewicz, 2019; Anastasiou & Fryzlewicz, 2019). One of these is wild binary segmentation (Venkatraman, 1992; Fryzlewicz, 2014). Wild binary segmentation randomly chooses \( M \) regions of data \( y_{i_{r+1}:t} \) for \( i_{r+1} < \ldots < i_M \), with \( i_r < i_{r+1} \). It then applies the cusum test to detect a change on each \( y_{i_{r+1}:t} \) and records the value of the maximum of the cusum statistic and the estimated changepoint location for each region. The regions are then ordered by the value of the test statistic, and these regions are processed in order. As we process the \( i_{r+1} \)th region, we add the detected changepoint from the test for that region provided the cusum test statistic is above our threshold and no previously detected changepoints lie within the \( i_{r+1} \)th region.

The rationale for this procedure is that, if \( M \) is large enough, then for each true changepoint there should be at least one of the \( M \) regions that contains it and no other changepoint. We hope that our test for detecting a single change will have sufficient power to detect that change within one of these regions. When processing the results of the test for the regions, we ignore those in which we have already detected a change to avoid potentially estimating multiple changes associated with just one true change. Extensions of wild binary segmentation include seeded binary segmentation (Kovács, Li, Bühlmann, & Munk, 2020) that uses a deterministic choice of regions of the data, Isolate-Detect (Anastasiou & Fryzlewicz, 2019) that chooses regions and searches for changes from the start or end of the data, and a recursive version (Fryzlewicz, 2018a) that uses information from estimated changepoint locations to help choose new regions.

One nice property of binary segmentation and wild binary segmentation is that, rather than pre-specifying a threshold for the test statistic, it is quick to consider how the estimated number and locations of changepoints would vary as we vary the threshold. We can then use other procedures to choose between our different estimates of the number and locations of the changes. In particular, if \( \phi_{1:M} \) denotes the estimated locations of the changepoints from wild binary segmentation, say, with the threshold specified so that we estimate \( m \) changepoints, then Fryzlewicz (2014) suggests estimating the number of changes, for some suitably specified \( \beta \), by

\[
\hat{m} = \arg \max_m \left\{ \min_{\phi_{1:M}} \left( \sum_{\phi_j} \min_{\tilde{\tau}_j} \sum_{t_{\tilde{\tau}_j+1}} \gamma(y_{t:\tilde{\tau}_j}, \theta_j) + \beta m \right) \right\},
\]

where \( \gamma(y_{t:\tilde{\tau}_j}, \theta_j, \sigma) \) is minus twice the log-likelihood under a Gaussian model, and we have defined \( r_0^m = 0 \) and \( r_{m+1}^m = n \). This condition is similar to a penalized log-likelihood or information criteria. Fryzlewicz (2014) shows that such a procedure will be consistent under in-fill asymptotics if \( \beta \) increases with \( n \) faster than \( \log n \) but slower than \( n \). Specifically, Fryzlewicz (2014) suggest \( \beta = 2(\log n)^{0.1} \), which is almost identical to the choice of \( 2 \log n \) that would correspond to using a Schwarz/Bayesian information criteria. There are also other ways of post-processing the output from a binary segmentation based algorithm. For example, the Isolate-Detect method of Anastasiou and Fryzlewicz (2019) suggests using the locations of estimated changepoints to prune changepoint estimates that are too close to each other.

Binary segmentation can be viewed as a divisive clustering method, as it initially clusters all data together in a single segment and then recursively considers splitting segments. Fryzlewicz (2018b) considers an approach to estimating changepoints based on an agglomerative clustering approach: Initially, each data point is in its own segment, and these are recursively merged together to create fewer and larger clusters.
Such a method requires the choice of a threshold that determines when to stop merging segments. For in-fill asymptotics, Fryzlewicz (2018b) shows that for a suitable choice of threshold, the method will be consistent. The theory behind this consistency result uses the standard, $2 \log n$, bound for the segment means for independent standard normal data (Yao, 1988) that is used to motivate the use of a $2 \log n$ in the penalized cost methods we now introduce.

### 3.2 Penalized cost methods

An alternative way of extending the test for detecting a single changepoint to the multiple changepoint setting is by relating the test to the minimization of the following optimization problem

$$\min_{m, \tau} \left\{ \left( \sum_{j=0}^{m} \min_{\theta_j} \sum_{t=\tau_j+1}^{\tau_{j+1}} \gamma(y_t; \theta_j, \sigma) \right) + \beta m \right\}, \quad (2)$$

where, as previously, we use the notation that $\tau_0 = 0$ and $\tau_{m+1} = n$. If we minimize subject to the constraint that $m \leq 1$, then the value of $m$ that achieves the minimum will correspond to whether a test for a single changepoint, using the test statistic $LR_1$ and threshold $\beta$, would detect a changepoint ($m = 1$) or not ($m = 0$).

If we wish to estimate multiple changepoints we could perform the minimization of (2) without any constraint on the value that $m$ can take. The value of $m$ and the associated values $r_{1:m}$ that minimize (2) would then be our estimate of the number and locations of the changes. As $\gamma$ is defined in terms of the negative of the log-likelihood for a model for the data, this can be viewed as a penalized likelihood approach. However, it generalizes to other scenarios, for example, Fearnhead and Rigaill (2019) suggest using the bi-weight loss,

$$\gamma(y; \theta, \sigma) = \min \left\{ \frac{1}{\sigma^2} (y - \theta)^2, K \right\}, \quad (3)$$

for some chosen $K > 0$, so as to make detection of changepoints robust to point outliers. Thus, the general approach of minimizing (2) is often termed a penalized cost method. The costs we are using assume that the noise variance, $\sigma$, is known; in practice, this often has to be estimated, for example, based on a robust estimate of the variance of the first differences of the data (see, e.g., Fryzlewicz, 2014). How important the method for estimating $\sigma$ is has had little attention in the literature, but it is simple to show that this commonly used estimator is sensitive to deviations from IID Gaussian assumptions, and this is one reason why methods based on comparing segmentations with differing numbers of changepoints (as described above and below) may be preferred in real applications.

A key practical question is how feasible is minimizing (2). It turns out that there are efficient dynamic programming algorithms for minimizing (2) (Jackson et al., 2005; Killick, Fearnhead, & Eckley, 2012; Maidstone, Hocking, Rigaill, & Fearnhead, 2017), with the FPOP algorithm (Maidstone et al., 2017) particularly efficient for the change in mean problem we are considering.

There is a close relationship between this penalized cost approach and the binary segmentation approach. Binary segmentation is equivalent to a greedy approach to approximately minimizing (2): At each iteration, we add the changepoint, if any, that will reduce our objective function by as much as possible. Furthermore, the approach suggested by Fryzlewicz (2014) for choosing the number of changepoints given the output of wild binary segmentation involves minimizing a very similar objective function. The main difference here is that for wild binary segmentation, we have a pre-chosen set of changepoints corresponding to each value of $m$, whereas when minimizing (2), we are choosing the changepoint locations that are best in terms of minimizing this criteria. As with the binary segmentation methods, minimizing the penalized cost has been shown to give consistent estimates under in-fill asymptotics but with tighter results on the choice of $\beta$, which needs to be at least $2 \log n$, rather than increasing faster than $\log n$.

As with binary segmentation methods, rather than pre-specifying the threshold/penalty, $\beta$, we may wish to compare possible segmentations obtained with differing $\beta$ or differing $m$. Haynes, Eckley, and Fearnhead (2017) give an efficient way of solving (2) for all values of $\beta$ within any interval—though, unlike with binary segmentation or wild binary segmentation, this involves re-running a dynamic programming algorithm a number of times (corresponding roughly to the number of different segmentations that are optimal as you vary $\beta$ within the interval). Alternatively, though at slightly higher computational cost, one can solve the minimization problem (2) for a fixed value of $m$ and repeat this for $m = 0, 1, \ldots, m_{\text{max}}$, for some chosen $m_{\text{max}}$ (Auger & Lawrence, 1989; Rigaill, 2015). For each $m$, the output will be the choice of locations of the $m$ changepoints that minimizes the cost or, for our change in mean problem, maximizes the log-likelihood.

Looking at various values for $m$ is needed when considering penalties that are not linear in the number of changes $m$ (Lebarbier, 2005; Arlot, Celisse, & Harchaoui, 2012). In that case, the goal is to minimize

$$\min_{m, \tau} \left\{ \left( \sum_{j=0}^{m} \min_{\theta_j} \sum_{t=\tau_j+1}^{\tau_{j+1}} \gamma(y_t; \theta_j, \sigma) \right) + C_{\text{pen}}(m) \right\}, \quad (4)$$
where \( \text{pen}(m) \) is the penalty and \( C \) a constant. For example, the penalty of Lebarbier (2005) is \( C \text{pen}(m) = \sigma^2 m(c_1 \log(n/m) + c_2) \); simulations suggest to take \( c_1 = 2 \) and \( c_2 = 5 \). Importantly, non-asymptotic results show that these penalties control the risk (Massart, 2007), retrieve the correct number of changepoints, and estimate the changepoint locations at the optimal rate (Garreau & Arlot, 2018).

Using such a penalty is non-trivial as the penalty is only known up to a constant, \( C \). Determining this constant can be done using the slope heuristic (Lebarbier, 2005; Arlot, Brault, Baudry, Maugis, & Michel, 2016; Arlot, 2019) by looking at how the segmentations vary as we change \( C \). Two variants exist in the R package \texttt{capushe}. The first, called \texttt{Djump}, looks for the unique largest jump of the function mapping \( C \) to the \( m \) minimizing (4). The second, called \texttt{Ddse}, considers segmentations with many changes and uses the fact that as we add false changepoints, the empirical risk reduces roughly linearly with \( m \). Deviations from this linear function indicate when real changes are being identified. These heuristics, although theoretically justified in some cases, sometimes fail (see Arlot, 2019, for example). Also, it is important to estimate segmentations with an \( m_{\max} \) number of changepoints that is at least two to three times larger than \( m^* \); too small a choice of \( m_{\max} \) is the reason why this approach performs poorly in some of the empirical results of Fryzlewicz (2018b).

Given either a set of estimated changepoint locations corresponding to different choices of \( m \) or different choices of \( \beta \), Haynes et al. (2017) and Lavielle (2005) suggest looking at how the cost for these segmentations, defined as the right-hand side of (2) with \( \beta = 0 \), changes as the number of changepoints increases. The idea is that the cost should decrease more when detecting true changes than when detecting spurious ones. Thus, we can estimate the number of changepoints, \( m \), by locating the “elbow” in the plot. Figure 1 shows examples of such plots for two data sets. The left-hand plots show the ideal scenario for such an approach to estimating \( m \), where we have data simulated with a number of similar sized changes. Here, an elbow is clearly seen and corresponds to the true number of changepoints. The right-hand plots show a more challenging, and potentially more realistic, scenario where there are many changes of differing sizes. In this case, identifying the location of an elbow is more subjective.

Estimating \( m \) by eye from such plots has obvious disadvantages, in particular such an approach cannot be applied to automated analyses of large numbers of data sets, as is often required in applications. However, this informal approach has been formalized into an automated method: it is similar to the idea behind the \texttt{Ddse} approach described above, the slope heuristic methods of Lebarbier (2005) and Lavielle (2005), and the steepest drop method of Fryzlewicz (2018a).
3.3 | Scan statistics and multi-scale methods

Another way of detecting multiple changepoints is to apply a test for a single change to regions of data of a chosen fixed width and then move these regions through the data from start to end. These are called scan statistics (Yau & Zhao, 2016) as they can be viewed as scanning through the data. One example of these methods is the MOSUM procedure of Eichinger and Kirch (2018). For the change in mean problem, this involves using the cusum-based test for a change at $t$ based on data from a region of width $2h$ say centred on $t$. The cusum test statistic can be calculated for each $t$, and the idea is that changes can be detected as local maxima in the cusum statistic, providing this maxima is above a suitable threshold. One of the challenges with this approach is avoiding estimating multiple changes around a single changepoint, which may happen due to the inherent fluctuations in the cusum statistic as you vary $t$. This can be achieved by not allowing two estimated changepoints to be too close to one another.

Scan statistic ideas can be extended to multi-scale methods by considering regions of differing lengths. A multi-scale version of MOSUM is suggested by Cho and Kirch (2019); but perhaps the most common multi-scale method is SMUCE (Frick, Munk, & Sieling, 2014), which brings together aspects of scan statistics and the penalized cost methods. SMUCE splits the data into regions and performs a test for the presence of a change on each of these regions. These regions are chosen to range in size, and the test statistics are aggregated appropriately across these differing scales, hence the multi-scale nature of the procedure. The number of changepoints is estimated as the smallest number of changes needed so that there is at least one changepoint within each region of data for which our test has detected a change. The locations of the changes are then estimated by minimizing a penalized cost (2) subject to requiring changes in regions of data where it was indicated by the tests.

One benefit of the SMUCE approach is that, as well as estimating the number and locations of the changes, it is able to quantify uncertainty in these estimates. For example, by appropriately choosing the threshold, and hence the significance level, for the tests, this approach immediately gives frequentist bounds on the probability of over-estimating the number of changes. However, SMUCE does not perform well for applications where there are many changes, particularly if some of these have small signal, as it tends to be too conservative in detecting changepoints. In some situations (e.g., Li, Munk, & Sieling, 2016), it can substantially underestimate the number of changepoints. To overcome this, Li et al. (2016) adapt the criteria for estimating the number of changepoints. They propose a method called FDR-Seg that controls a version of the false discovery rate for the estimated changepoints.

4 | COMPARISON OF METHODS

We now present a comparison of the performance of these different methods across a range of scenarios. Presenting these comparisons is challenging due to wishing to consider a number of distinct scenarios for the mean (differing numbers of changes, sizes of change, lengths of segment, etc.); the noise (IID Gaussian, heavy-tailed and auto-correlated); different measures of accuracy (error of estimating the segmentation of the data or the mean function); and the large number of methods that have been proposed for this problem, each of which can have a different tuning parameter that impacts on the number of changepoints detected.

As a result, we present indicative results in this section, with the complete results available in Supporting Information. Moreover, the Supporting Information is available as a markdown document with associated code available. The full results are presented in a way such that it is possible to extend the results to additional methods and/or additional scenarios. The indicative results we present below are based on comparing some distinct approaches but focusing on those approaches that appear to be most accurate.

4.1 | Results: Known number of changepoints

To isolate the accuracy of a method at estimating the changepoints’ locations from the tuning of the method to estimate the number of changepoints, we first present results where we assume the number of changepoints is known. We consider 4 of the 11 scenarios for the mean function that are given in the Supporting Information (see the top tow of Figure 2). These are representative of the different behaviours and include a scenario with a random mix of segment lengths and sizes of change; one structured to have increasing segment sizes with decreasing sizes of change; and two with small segments and a constant size of change, the first alternating up and down changes and the latter having only up changes. All these scenarios are taken from studies in the recent changepoint papers.

For each scenario, we consider varying the number of data points, $n$. This is done so that a doubling of $n$ will mean twice as many data points per segment but that the variance of the noise for each data point is also doubled; hence, the amount of information about the mean is roughly constant as we vary $n$. Furthermore, we compare three cases for the noise: The first is that it is IID Gaussian; second, we have IID $t$-distributed noise with 10 degrees of freedom, and third, we have an AR(1) noise process with lag-1 correlation of 0.3. The marginal variance of the noise is the same in each case. These cases relate in turn to the model most commonly assumed in theory for changepoint methods, a heavier tailed alternative, and a model with auto-correlation.

The results in Figure 2 give a comparison of four methods in terms of their mean square error of estimating the mean function, though qualitatively similar results are seen when the methods are compared in terms of the accuracy of the segmentation of the data that they obtain. The four methods we compare include three methods that measure fit to data based on the square error loss (or equivalently based on the log-likelihood for IID Gaussian noise): the best fitting segmentation in terms of the square error loss as obtained by dynamic programming; binary segmentation; and wild binary segmentation. The final method is the best segmentation in terms of minimizing the bi-weight loss (3).
FIGURE 2  Performance of different methods, in terms of mean square error of the fitted mean, for a range of scenarios with known numbers of changepoints. Top row shows the mean (red line) and example data for the results for each column for one value of the number of data points. Results are shown for IID Gaussian noise (second row), IID Student-t noise (third row), and AR(1) noise (fourth row). Plots in each show the mean square error for each method against $n$, where varying $n$ affects the number of data points per segment. We compare maximum likelihood under IID Gaussian noise model (black full line); minimization of the bi-weight loss (blue dot-dashed line); binary segmentation (green dotted line); and wild binary segmentation (red dashed line)

For the scenarios in the first three columns of Figure 2, there is a consistent picture. The fit given by binary segmentation is substantially and consistently worse than the other methods. The other methods give very similar performance except that the bi-weight loss performs noticeably better for the t-distributed noise, which is as expected, as this loss is designed to give robustness to outliers. Perhaps more surprising is that, for the IID Gaussian noise case, using the bi-weight loss does not lead to a noticeable loss of accuracy relative to the square error loss.

The results for the scenario in the final column are substantially different, as binary segmentation is competitive with the alternative methods in this case. This is consistent with theory that suggests binary segmentation loses power to detect changes in scenarios where you have changes of alternating signs, whereas this scenario has changes that are all the same sign.

4.2  Results: Unknown number of changepoints

Figure 3 gives results when we repeat the above comparison for methods that also estimate $m$, the number of changepoints. We compare four methods, minimizing the square error loss with BIC penalty; minimizing the square error loss for different values of $m$; and estimating $m$ using the penalty of Lebarbier (2005) with a difference based estimator of the variance (Hall, Kay, & Titterington, 1990); wild binary segmentation, with the
FIGURE 3  Performance of different methods, in terms of mean square error of fitted mean, for a range of scenarios with unknown numbers of changepoints. Columns are for the same scenarios as in Figure 2. Results are shown for IID Gaussian noise (top row), IID Student-t noise (middle row), and AR(1) noise (bottom row). Plots in each row show the mean square error for each method against \( n \), where varying \( n \) affects the number of data points per segment. We compare wild binary segmentation (blue dot-dashed line); FDR-Seg (green dotted line); minimization of the square error loss with BIC penalty (red dashed line); and minimization of the square error loss with the penalty of Lebarbier (2005) with a difference based estimator of the variance (Hall et al., 1990) (light blue long dashed line). To show the effect of estimating the number of changes, we also show results for the maximum likelihood under IID Gaussian noise models with a known number of changepoints (black full line).

The first two scenarios (two left columns of Figure 3) give similar results. First, all methods perform well when the noise is IID Gaussian: There is little to choose between the methods, and all perform nearly as well as for the case where \( m \) is known. This is because the model we simulated from satisfies the standard assumptions underpinning the theory for the approaches for estimating \( m \), and in these scenarios, the changes are relatively straightforward to detect. We see some drop off in performance for FDR-Seg and wild binary segmentation for the \( t \)-distributed noise, and a much larger drop in performance for the auto-correlated noise case: These and other simulation results suggest that FDR-Seg is particularly sensitive to the noise assumptions it makes being correct. The pattern of performance of wild binary segmentation is largely due to our choice of threshold for estimating the number of changes, \( m \). The results we present are for the version of wild binary segmentation that performed best in the IID Gaussian case, but it tends to over-estimate the number of changes when we have auto-correlated or heavy-tailed noise. By comparison, using the Schwarz Information Criteria, (1) with \( \beta = 2 \log n \) estimates fewer changes and gives a better performance for the auto-correlated noise case (see Supporting Information), highlighting how conclusions about a changepoint algorithm can be sensitive to the choice of method for estimating \( m \).

The results in the latter two scenarios (two rightmost columns of Figure 3) give quite different results. In particular, all methods tend to perform better when there is correlated noise than when the noise is IID Gaussian. This counter-intuitive performance has a simple explanation. These two scenarios have many short segments. All the methods we compare tend to under-estimate \( m \) in these cases, and this under-estimation of \( m \) is the reason for the big difference in performance compared with a method that assumes \( m \) is known. By comparison, the impact of auto-correlation in the noise is to encourage methods to fit too many changes: fluctuations in the data caused by auto-correlation will have similar patterns to adding a small change in mean. In these two scenarios, the effects seem to cancel each other out.
FIGURE 4  Performance of different methods that assume a minimum segment, in terms of mean square error of the fitted mean, for a range of scenarios with an unknown number of changepoints. Columns are for the same scenarios as in Figure 2. Results are shown for IID Gaussian noise (top row), IID Student-t noise (middle row), and AR(1) noise (bottom row). Plots in each row show the mean square error for each method against $n$, where varying $n$ affects the number of data points per segment. We compare MOSUM (red dashed line); iDetect (green dotted line), and penalized square error cost with minimum segment length of 4 (blue dot-dashed line). To see the impact of setting a minimum segment length, we also show results for the penalized square error cost with no minimum segment length (black full line).

Across all scenarios, we can see that using the slope heuristic, that is, comparing fit to data as we vary the number of changepoints, performs well, and this type of method for estimating $m$ appears to be promising and to give some robustness to the type of scenario and/or to the violation of the IID Gaussian assumption on which most default thresholds for estimating $m$ are based.

The results in the Supporting Information show that both iDetect and MOSUM approaches have very strong performance for many scenarios when either we have auto-correlated or heavy-tailed noise. Results for the four scenarios we are focusing on are shown in Figure 4. The reason for this appears to be because both methods have an implicit minimum segment length, due to rules that mean they avoid detecting changes that are too close to each other. For scenarios with segments longer than this minimum segment length, this appears to make the methods robust to violations of the IID Gaussian assumptions. To demonstrate this, we also show in Figure 4 the strong performance of minimizing a penalized version of the square error loss with an explicitly imposed minimum segment length of 4, and we can directly see the improvement we obtain relative to the same approach without any minimum segment length. The results for MOSUM are somewhat erratic for the fourth scenario in Figure 4, which appears to be driven by how the choice of window length used in the scan statistic relates to the actual segment lengths as we vary $n$.

In some applications, the computational cost of the algorithm used is important. We present timing results for the changepoint algorithms in the Supporting Information, with a subset of results presented in Figure 5. First, we observe that most algorithms have an empirical computational cost that is close to linear in the number of observations, $n$; though across these, there is an order of magnitude difference in computational cost. Exceptions include methods that use the slope heuristic and require multiple runs of an algorithm to estimate segmentations with differing numbers of changepoints. Often, the number of different segmentations used by these methods is linear in $n$, and thus, these methods have a computational cost that is closer to quadratic in $n$. Also, the FDR-Seg method has a computational cost that empirically is worse than quadratic; for large $n$, FDR-Seg was substantially slower than all of its competitors. Furthermore, FDR-Seg requires simulation to choose appropriate threshold values—the timings in Figure 5 ignore this considerable upfront cost.
FIGURE 5  Log-log plot of CPU cost (in seconds) for a range of changepoint algorithms against the data set size (the number of data points, n). Left-hand plot shows a scenario with segments of length 64 so the number of changepoints increases linearly with n, and the right-hand plot shows a scenario with no changepoints. The methods compared are FPOP (black), WBS with the SIC penalty (red), MOSUM (green), the agglomerative Tail Greedy Unbalanced Haar approach of Fryzlewicz (2018b) (blue), a slope heuristic using multiple runs of FPOP with the CROPS algorithm of Haynes et al. (2017) (light blue), and FDR-Seg (pink). The dashed grey lines correspond to CPU cost increasing linearly and quadratically with n.

5 | DISCUSSION

Our results suggest that care is needed when comparing different changepoint methods if one wishes to understand when and why one method is better than another—as it is rare that any method is uniformly better than another.

First, it is important to recognize that detecting changes has two aspects: estimating the number of changes and estimating their locations. Many published studies look at these together or compare different algorithms each with a different way of estimating the number of changes. In such results, it is hard to determine whether one algorithm is better than another or whether it is the estimate of the number of changes that is better. We would recommend that comparisons between algorithms should include comparisons which assume the number of changes is known.

Second, most methods have default settings that are based on theory for the case of IID Gaussian noise. These settings may not work well in applications, where often there will be features such as auto-correlation or heavy-tailed noise. Furthermore, the presence of auto-correlation can even affect the commonly used estimate of the marginal variance of the noise, obtained by the median-absolute-deviation estimate from the first differences of the data, which is used to standardize the data. We think that methods that estimate segmentations for a range of values of the number of changepoints are more reliable in situations where the IID Gaussian assumption does not hold, though the best way of using the information from these different segmentations is still an open problem. In applications where there is training data, using such data to train the tuning parameters so as to give reliable estimates of the number of changepoints (Hocking, Rigaill, Vert, & Bach, 2013) may be best.

Finally, assuming a minimum segment length, if the assumption is correct, can noticeably improve performance and also give some robustness to heavy-tailed noise or auto-correlation. Care is needed when comparing methods that either explicitly or implicitly assume a minimum segment length with those that do not—as often it will be whether or not this assumption holds that will be the primary reason why the method is more accurate or not.

ACKNOWLEDGEMENTS

This work is supported by EPSRC Grant EP/N031938/1 and an ATIGE grant from Genopole. The IPS2 benefits from the support of the LabEx Saclay Plant Sciences-SPS.

DATA AVAILABILITY STATEMENT

R code for the simulations study is provided in the Supporting Information, and full code and results are available at https://github.com/guillemr/SomeSegSimu.

ORCID

Paul Fearnhead https://orcid.org/0000-0002-9386-2341

REFERENCES

Anastasiou, A., & Fryzlewicz, P. (2019). Detecting multiple generalized change-points by isolating single ones.
Arlot, S. (2019). Minimal penalties and the slope heuristics: A survey. arXiv:1901.07277.
Arlot, S., Brault, V., Baudry, J.-P., Maugis, C., & Michel, B. (2016). capushe: Calibrating penalties using slope heuristics. R package version 1.1.1.

Arlot, S., Celisse, A., & Harchaoui, Z. (2012). A kernel multiple change-point algorithm via model selection. arXiv:1202.3878.

Aue, A., & Horváth, L. (2013). Structural breaks in time series. Journal of Time Series Analysis, 34(1), 1–16.

Auger, I. E., & Lawrence, C. E. (1989). Algorithms for the optimal identification of segment neighborhoods. Bulletin of Mathematical Biology, 51(1), 39–54.

Bai, J. (1997). Estimating multiple breaks one at a time. Econometric Theory, 13(3), 315–352.

Baranowski, R., Chen, Y., & Fryzlewicz, P. (2019). Narrowest-over-threshold detection of multiple change-points and change-point-like features. Journal of the Royal Statistical Society: Series B, 81(3), 649–672.

Cao, H., & Wu, W. B. (2015). Changepoint estimation: Another look at multiple testing problems. Biometrika, 102(4), 974–980.

Cho, H., & Kirch, C. (2019). Localised pruning for data segmentation based on multiscale change point procedures. arXiv.1910.12486.

Eichinger, B., & Kirch, C. (2018). A MOSUM procedure for the estimation of multiple random change points. Bernoulli, 24(1), 526–564.

Fearnhead, P., & Rigail, G. (2019). Change point detection in the presence of outliers. Journal of the American Statistical Association, 114(525), 169–183.

Frick, K., Munk, A., & Sieling, H. (2014). Multiscale change-point inference. Journal of the Royal Statistical Society: Series B, 76(3), 495–580.

Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. The Annals of Statistics, 42(6), 2243–2281.

Fryzlewicz, P. (2018a). Detecting possibly frequent change-points: Wild Binary Segmentation 2 and steepest-drop model selection. arXiv.1812.06880.

Fryzlewicz, P. (2018b). Tail-greedy bottom-up data decompositions and fast multiple change-point detection. The Annals of Statistics, 46(6B), 3390–3421.

Garreau, D., & Arlot, S. (2018). Consistent change-point detection with kernels. Electronic Journal of Statistics, 12(2), 4440–4486.

Hall, P., Kay, J., & Titterington, D. (1990). Asymptotically optimal difference-based estimation of variance in nonparametric regression. Biometrika, 77(3), 521–528.

Haynes, K., Eckley, I. A., & Fearnhead, P. (2017). Computationally efficient changepoint detection for a range of penalties. Journal of Computational and Graphical Statistics, 26(5), 134–143.

Hinkley, D. V. (1971). Inference about the change-point from cumulative sum tests. Biometrika, 58(3), 509–523.

Hocking, T., Rigail, G., Vert, J.-P., & Bach, F. (2013). Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research, (Vol. 28, pp. 172–180). Atlanta, Georgia.

Jackson, B., Scargle, JD, Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., ..., & Tsai, T. T. (2005). An algorithm for optimal partitioning of data on an interval. IEEE Signal Processing Letters, 12(2), 103–108.

Killick, R., Fearnhead, P., & Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500), 1590–1598.

Koeckele, L., Ashida, G., & Kretzberg, J. (2016). Single and multiple change point detection in spike trains: Comparison of different CUSUM methods. Frontiers in Systems Neuroscience, 10, 51.

Kovács, S., Li, H., Bühlmann, P., & Munk, A. (2020). Seeded binary segmentation: A general methodology for fast and optimal change point detection. arXiv.2002.06633.

Lavielle, M. (2005). Using penalized contrasts for the change-point problem. Signal Processing, 85(8), 1501–1510.

Lebarbier, É. (2005). Detecting multiple change-points in the mean of Gaussian process by model selection. Signal Processing, 85(4), 717–736.

Li, H., Munk, A., & Sieling, H. (2016). FDR-control in multiscale change-point segmentation. Electronic Journal of Statistics, 10(1), 918–959.

Maidstone, R., Hocking, T., Rigail, G., & Fearnhead, P. (2017). On optimal multiple changepoint algorithms for large data. Statistics and Computing, 27(2), 521–533.

Massart, P. (2007). Concentration inequalities and model selection. Berlin, Heidelberg: Springer.

Muggeo, V. M., & Adelfio, G. (2010). Efficient change point detection for genomc sequences of continuous measurements. Bioinformatics, 27(2), 161–166.

Olshen, A. B., Venkatraman, E. S., Lucito, R., & Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics, 5(4), 557–72.

Reeves, J., Chen, J., Wang, X. L., Lund, R., & Lu, Q. Q. (2007). A review and comparison of changepoint detection techniques for climate data. Journal of Applied Meteorology and Climatology, 46(6), 900–915.

Rigail, G. (2015). 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.

Scott, A. J., & Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. Biometrics, 30, 507–512.

Truong, C., Oudre, L., & Vayatis, N. (2019). Selective review of offline change point detection methods. Signal Processing, 167, 107299.

Venkatraman, E. S. (1992). Consistency results in multiple change-point problems. (Ph.D. Thesis), Stanford University.

Yao, Y.-C. (1988). Estimating the number of change-points via Schwarz's criterion. Statistics & Probability Letters, 6(3), 181–189.

Yau, C. Y., & Zhao, Z. (2016). Inference for multiple change points in time series via likelihood ratio scan statistics. Journal of the Royal Statistical Society: Series B, 78(4), 895–916.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of the article.

How to cite this article: Fearnhead P., Rigail G. Relating and comparing methods for detecting changes in mean. Stat. 2020:9:e291. https://doi.org/10.1002/sta4.291