Multiple change-point detection with localised pruning

Haeran Cho* †
Joint work with Claudia Kirch (OvGU Magdeburg) ‡

Computational strategies for
large-scale statistical data analysis
ICSM, July 2018

---

*School of Mathematics, University of Bristol, UK.
†Supported by EP/N024435/1.
‡Supported by DFG-GRK 2297 and DFG-KI 1443/3-1.

Change-point problem

- Change-point problems have been of interest to statisticians for many decades.
- Detection of change-points in mean, variance, regression coefficients, second-order structure and distribution in general, in univariate, multivariate or high-dimensional data, in both a posteriori (retrospective) and sequential manner.
- We are concerned with the classical, a posteriori multiple change-point detection problem in univariate data:

$$X_t = \sum_{j=0}^{q_n} f_j \cdot \mathbb{I}(k_j + 1 \leq t \leq k_{j+1}) + \varepsilon_t, \quad t = 1, \ldots, n.$$

- $k_0 + 1 = 1 \leq k_1 < k_2 < \ldots < k_{q_n} < n = k_{q_n+1}$.
- Total number (q_n) as well as locations ($k_1, \ldots, k_{q_n}$) of change-points are unknown and to be estimated.
- Errors $\varepsilon_1, \ldots, \varepsilon_n \sim (0, \sigma^2)$.

‘multiple’ change-point estimation

Considerably more challenging than single change-point estimation.

1. Estimation of the **total number** of the change-points itself is difficult.
   - Information criterion: Yao (1988), Lee (1995), Serbinowska (1996), Liu et al. (1997), Bai (1998), Kühn (2001), Ninomiya (2005), Pan & Chen (2006), Zhang & Siegmund (2007), Hannart & Naveau (2012), Fryzlewicz (2014)...
   - Typically requires the maximum number of change-points as an input.
2. Often, computing change-point detectors over an interval $=$ fitting a stump function.
   - May lead to undesirable consequences when the interval is ‘contaminated’ by **more than one change-points**.
   - Reflected in stronger assumption on the size of change-points for their detection.

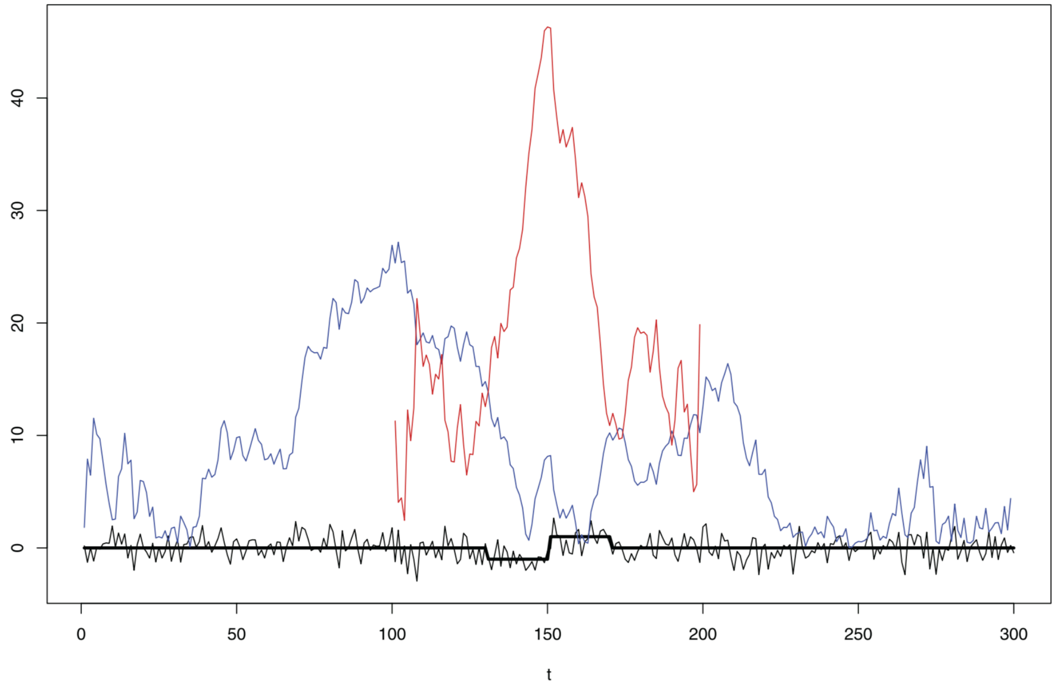


FIG. 1. *True function* f_t, $t = 1, \ldots, T = 300$ (*thick black*), *observed* X_t (*thin black*), $|\tilde{X}^b_{1,300}|$ *plotted for* $b = 1, \ldots, 299$ (*blue*), *and* $|\tilde{X}^b_{101,200}|$ *plotted for* $b = 101, \ldots, 199$ (*red*).

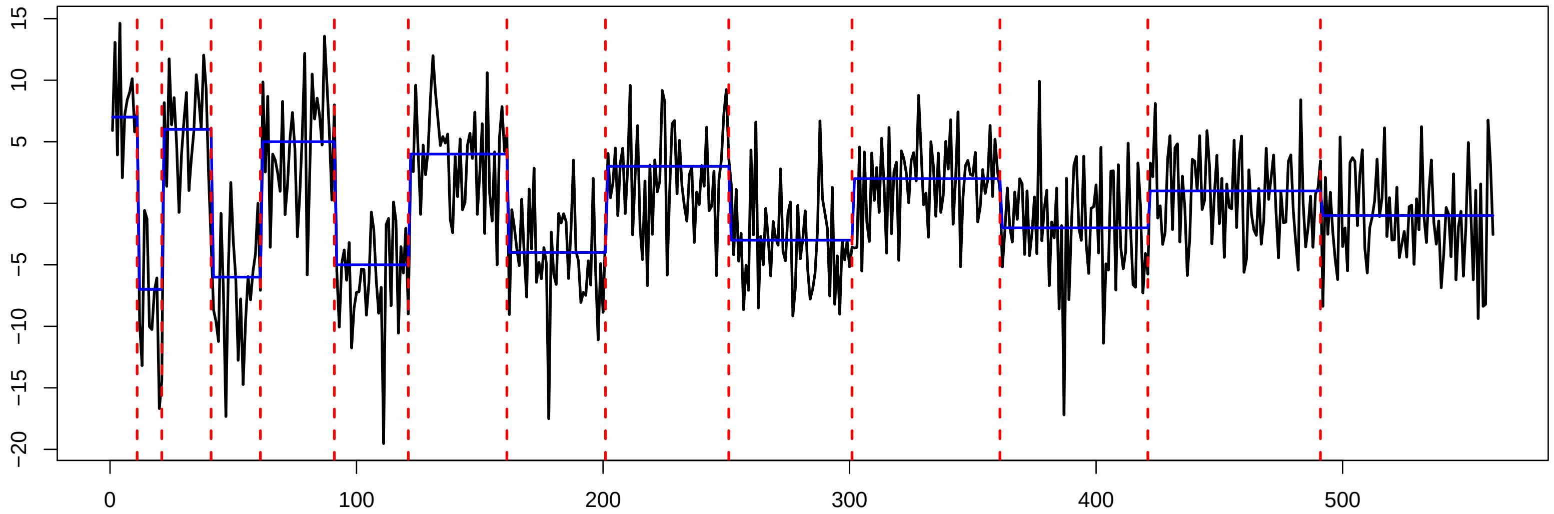
(Fryzlewicz, 2014))

Localised approaches to change-point estimation

Aim: isolate each change-point in an interval sufficiently large for its detection.

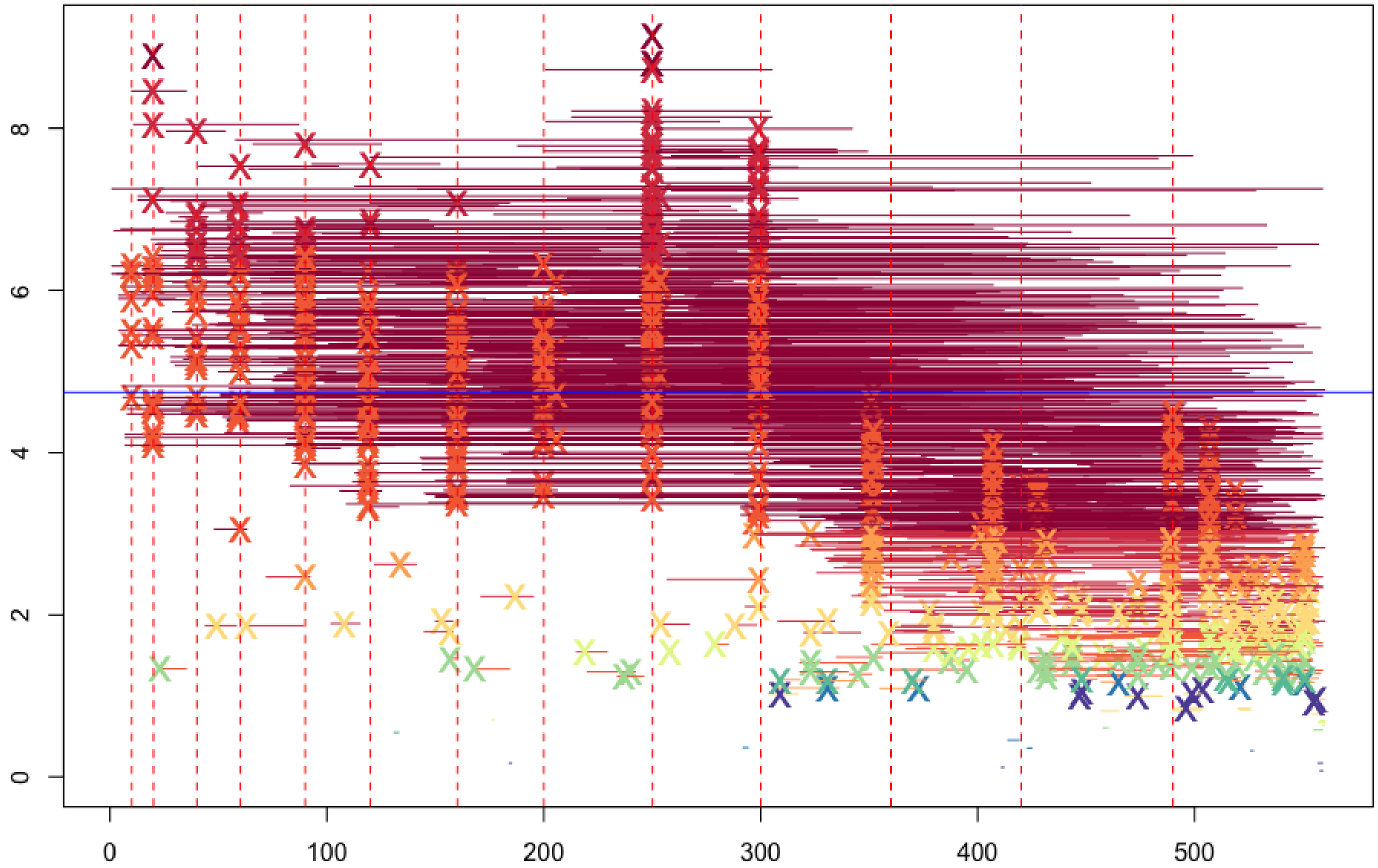
- **Wild Binary Segmentation** (Fryzlewicz, 2014): draws a large number of intervals randomly.
  - With high probability, for each k_j, there exists at least one interval which contains k_j only and is sufficiently large, so that its presence is detected as well as its location being accurately estimated, for all $j = 1, \ldots, q_n$.
- **MOSUM procedure** (Eichinger & Kirch, 2017): contrasts the behaviour of left and right summation windows using a moving window.
  - With bandwidth G satisfying $\min_j(k_{j+1} - k_j) > 2G$, summation windows contain at most a single change-point.
  - Multiscale extension with multiple bandwidths.

`mix` test signal $=$ large jumps over short intervals $+$ small jumps over long intervals.



Conflicting change-point estimators

CUSUM statistics over random intervals along with change-point candidates ('x').

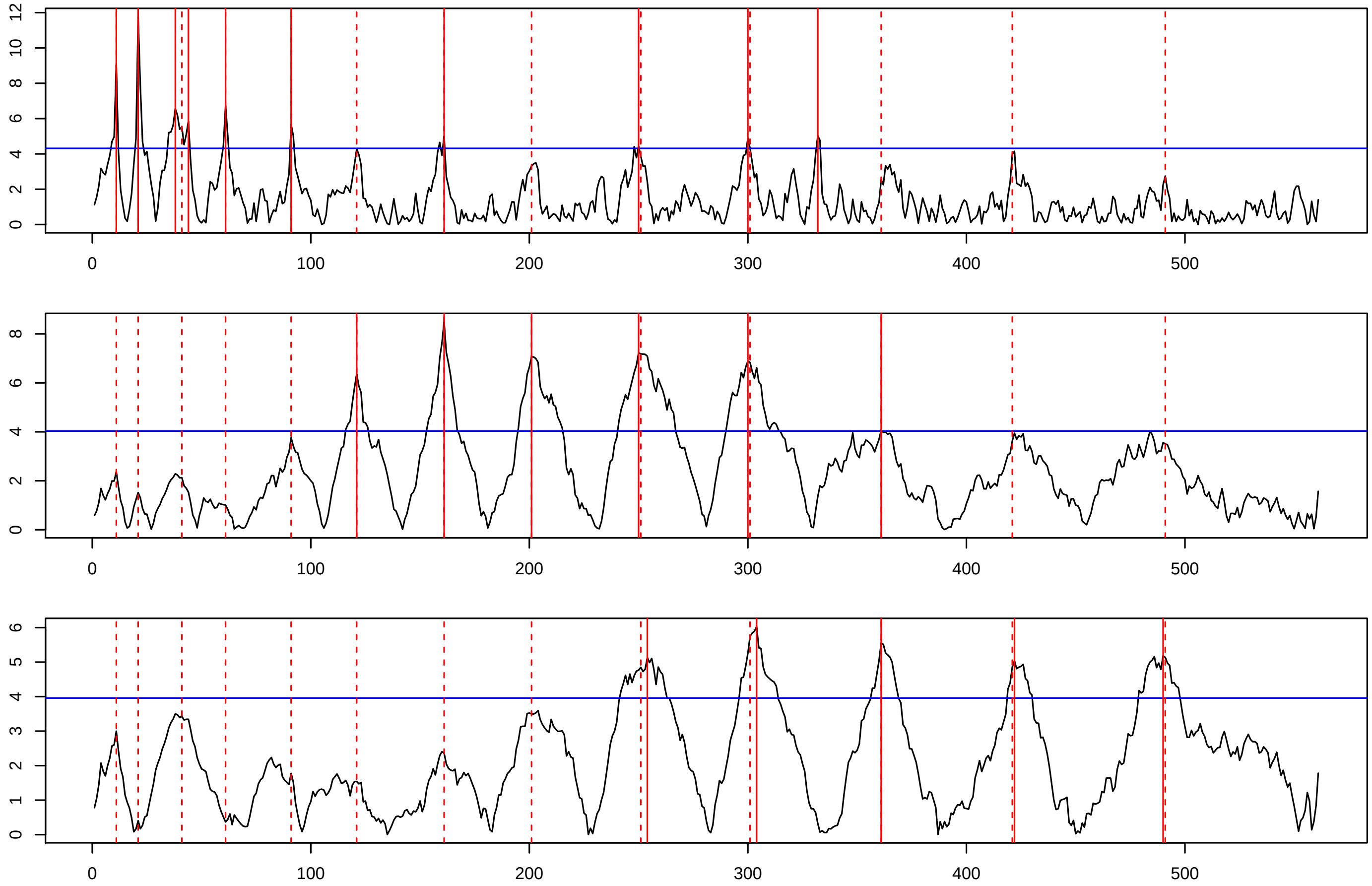


WBS resolves this issue by adopting a binary segmentation algorithm, which selects the interval with the largest CUSUM statistic and segments the data in an iterative manner.

- NOT (narrowest-over-threshold, Baranowski et al., 2016) selects the narrowest interval among those with CUSUM statistics exceeding a threshold.
- Final model selection depends on the choice of **threshold**, or the application of an **information criterion** to a sequence of nested models indexed by the number of change-points.

Conflicting change-point estimators

Multiscale MOSUM procedure with bandwidths $\in \{10, 30, 60\}$.



Messer et al. (2014) proposed 'bottom-up' merging of change-point candidates from the multiscale MOSUM procedure, starting from those detected with the smallest bandwidth.

– Cannot be generalised to asymmetric (left summation window $\neq$ right summation window) MOSUM procedure.
– Cannot remove some spurious change-point estimates.

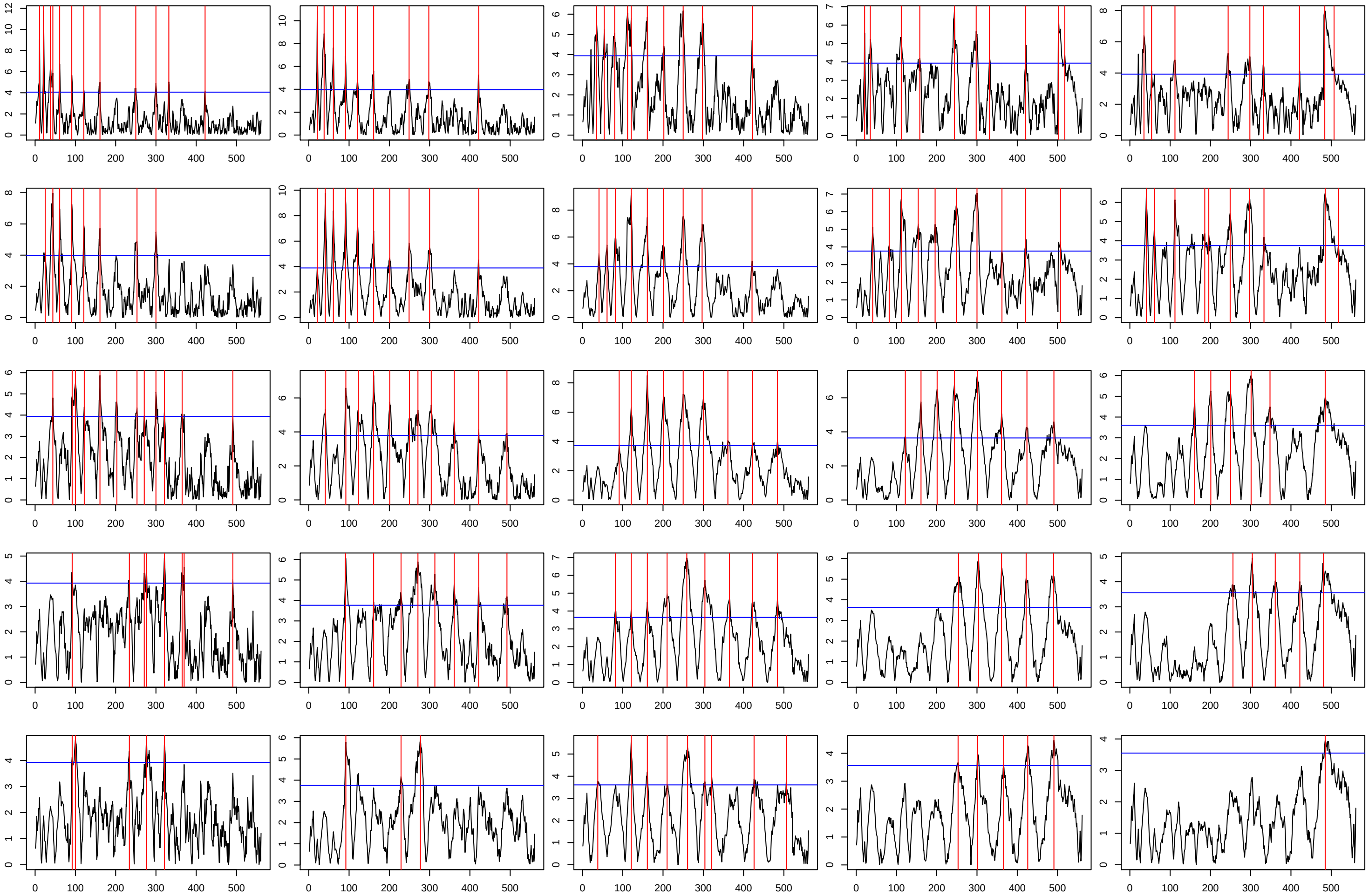
Localised pruning

A **generic** procedure applicable to change-point candidates returned by a class of change-point methodologies based on the principle of **isolating each change-point for its estimation**.

'Ingredients'

- $\widehat{\mathcal{B}}_{\text{pool}}$, the pool of all change-point estimates.

$\widehat{B}_{\text{pool}}$ from **multiscale** and **asymmetric** extension of MOSUM procedure with $\mathcal{H} = \{10, 20, 40, 60, 80\}$.



Localised pruning

A **generic** procedure applicable to change-point candidates returned by a class of change-point methodologies based on the principle of **isolating each change-point for its estimation**.

'Ingredients'

- $\widehat{\mathcal{B}}_{\text{pool}}$, the pool of all change-point estimates.
- Schwarz criterion (Schwarz, 1978): for $\mathcal{A} \subset \widehat{\mathcal{B}}_{\text{pool}}$,

$$\text{SC}(\mathcal{A}) = \frac{n}{2} \log \left\{ \frac{\text{RSS}(\mathcal{A})}{n} \right\} + \xi(n) \cdot |\mathcal{A}|.$$

$\widehat{B}_{\textbf{pool}} +$ **SC** $\Rightarrow$ **exhaustive search?**

$\widehat{\mathcal{B}}_{\text{pool}}$ may be pruned down via an exhaustive search using SC evaluated at every possible combination of estimates $\mathcal{A} \subset \widehat{\mathcal{B}}_{\text{pool}}$.

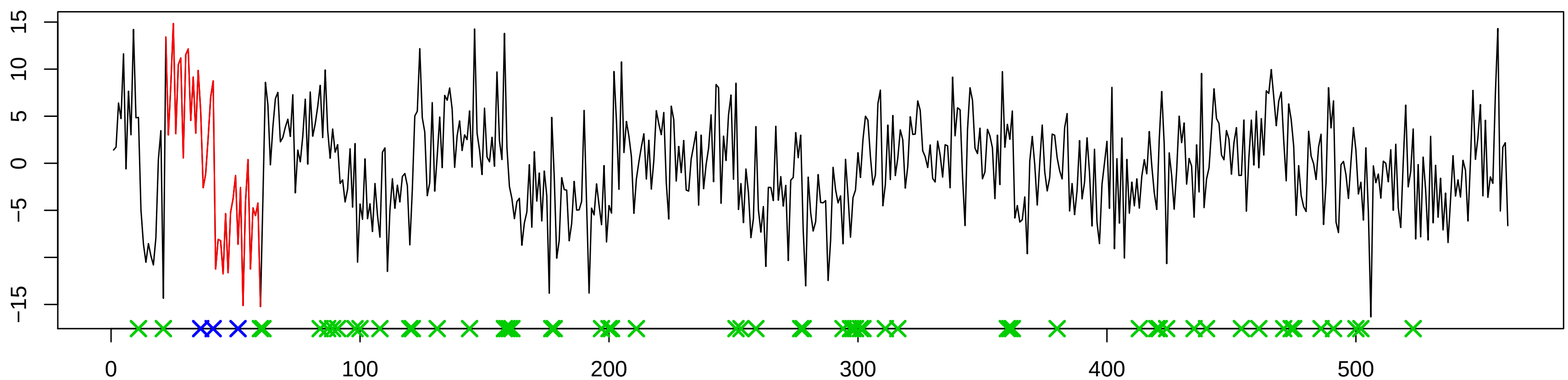
- Computationally expensive!
  - For this example, in total 177 change-point estimates ($= |\widehat{\mathcal{B}}_{\text{pool}}|$) while $q_n = 13$!
  - 2^{177} combinations!

```
        CP
  [1,]  11
  [2,]  21
  [3,]  41
...
[176,] 491
[177,] 301
```

$\widehat{B}_{\textbf{pool}} + \textbf{SC} \Rightarrow$ exhaustive search?

$\widehat{\mathcal{B}}_{\text{pool}}$ may be pruned down via an exhaustive search using SC evaluated at every possible combination of estimates $\mathcal{A} \subset \widehat{\mathcal{B}}_{\text{pool}}$.

- Computationally expensive!
- Ignores that each $\widehat{k}$ is detected within a local interval.



$\Rightarrow$ Motivates a **localised** pruning approach.

Localised pruning

A **generic** procedure applicable to change-point candidates returned by a class of change-point methodologies based on the principle of **isolating each change-point for its estimation**.

'Ingredients'

- $\widehat{\mathcal{B}}_{\text{pool}}$, the pool of all change-point estimates.
- Schwarz criterion (Schwarz, 1978): for $\mathcal{A} \subset \widehat{\mathcal{B}}_{\text{pool}}$,

$$\text{SC}(\mathcal{A}) = \frac{n}{2} \log \left\{ \frac{\text{RSS}(\mathcal{A})}{n} \right\} + \xi(n) \cdot |\mathcal{A}|.$$

- Such change-point estimation methods attach extra information to their change-point estimates, namely the interval within which each candidate is estimated.
  - For each $\widehat{k} \in \widehat{\mathcal{B}}_{\text{pool}}$, its detection interval $\mathcal{I}(\widehat{k}) \equiv [\widehat{k} - G_l + 1, \widehat{k} + G_r]$, with some $G_l = G_l(\widehat{k})$ and $G_r = G_r(\widehat{k})$.

Localised pruning: step-by-step

Step 0: assign the set of candidates $\mathcal{P} = \widehat{B}_{\text{pool}}$ (for future consideration), and the set of currently 'alive' estimates $\mathcal{C} = \widehat{B}_{\text{pool}}$ (survived pruning & for future consideration).

Step 1: identify $\widehat{k}^* \in \mathcal{P}$ with {largest jump size, smallest p-value}.

```
> sort CP according to jump size
        CP G_l G_r      p-value      jump
  [1,]  41  20  10 9.820958e-08 3.4338066
  [2,]  41  10  10 9.023790e-06 3.2922756
  [3,]  61  20  10 2.813026e-07 3.2901970
  [4,]  21  10  10 1.049419e-05 3.2684876
  [5,]  41  20  20 4.971903e-08 3.0141912
  ...
```

We can 'sort' the change-point candidates according to

- (scaled) jump size

$$\mathcal{J}(\widehat{k}) = \frac{1}{\widehat{\sigma}} \left| \frac{1}{G_l} \sum_{t=\widehat{k}-G_l+1}^{\widehat{k}} X_t - \frac{1}{G_r} \sum_{t=\widehat{k}+1}^{\widehat{k}+G_r} X_t \right|.$$

- p-value $= p(\widehat{k})$ from the (asymptotic) distribution of the test statistic under H_0.
  † $\widehat{k}$ with small $p(\widehat{k})$ / large $\mathcal{J}(\widehat{k})$ is preferable.

```
> sort CP according to jump size
       CP G_l G_r      p-value      jump
  [1,] 41  20  10 9.820958e-08 3.4338066
  [2,] 41  10  10 9.023790e-06 3.2922756
  [3,] 61  20  10 2.813026e-07 3.2901970
  [4,] 21  10  10 1.049419e-05 3.2684876
  [5,] 41  20  20 4.971903e-08 3.0141912
  ...
```

Step 2: select candidates $\mathcal{D}$ conflicting with $\widehat{k}$ as

$$\widehat{k} < \widehat{k}^* \text{ and } \widehat{k}^* - \widehat{k} \leq \min\{G_l(\widehat{k}^*), G_r(\widehat{k})\}, \text{ or}$$
$$\widehat{k} > \widehat{k}^* \text{ and } \widehat{k} - \widehat{k}^* \leq \min\{G_r(\widehat{k}^*), G_l(\widehat{k})\},$$

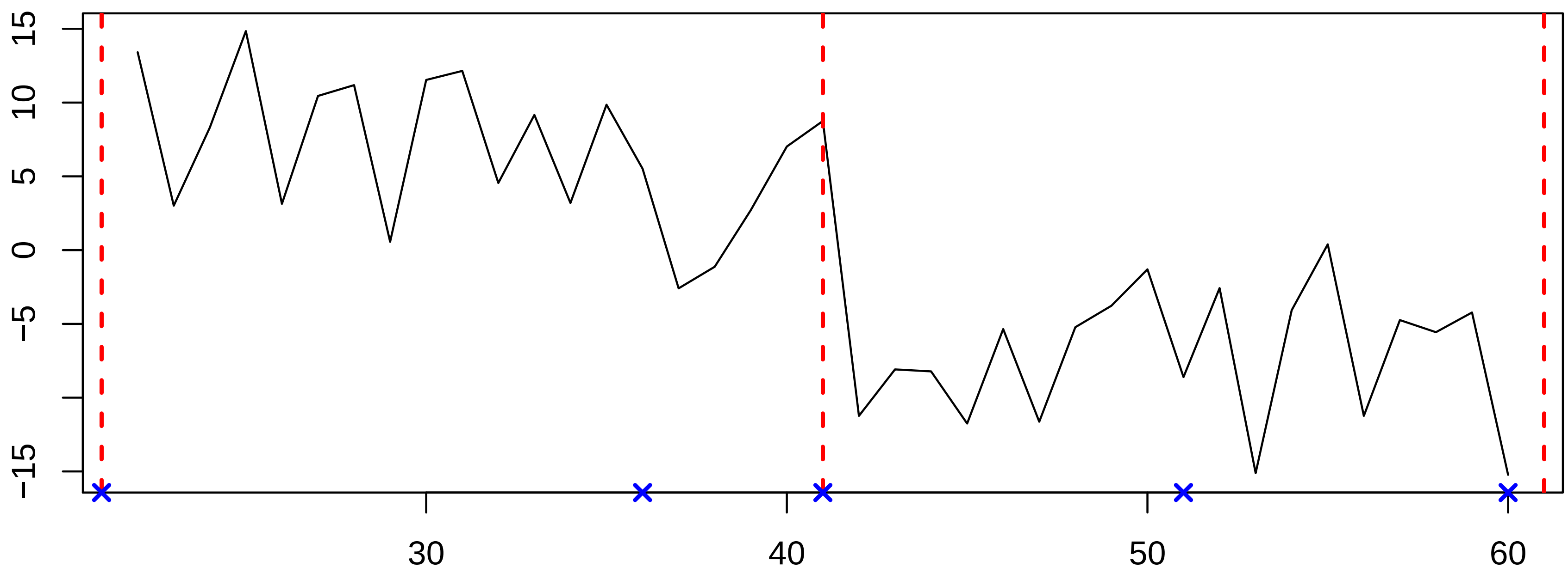
such that $\mathcal{D} = \{\widehat{k}_l, \ldots, \widehat{k}_r\}$.

† Detection intervals of $\widehat{k}$ contain $\widehat{k}^*$ and vice versa.

Step 3: identify a local environment containing $\mathcal{D}$, as

$$\widehat{k}_L = \text{largest estimate in } \mathcal{C} \text{ to the left of } \widehat{k}_l,$$
$$\widehat{k}_R = \text{smallest estimate in } \mathcal{C} \text{ to the right of } \widehat{k}_r.$$

Steps 2–3.



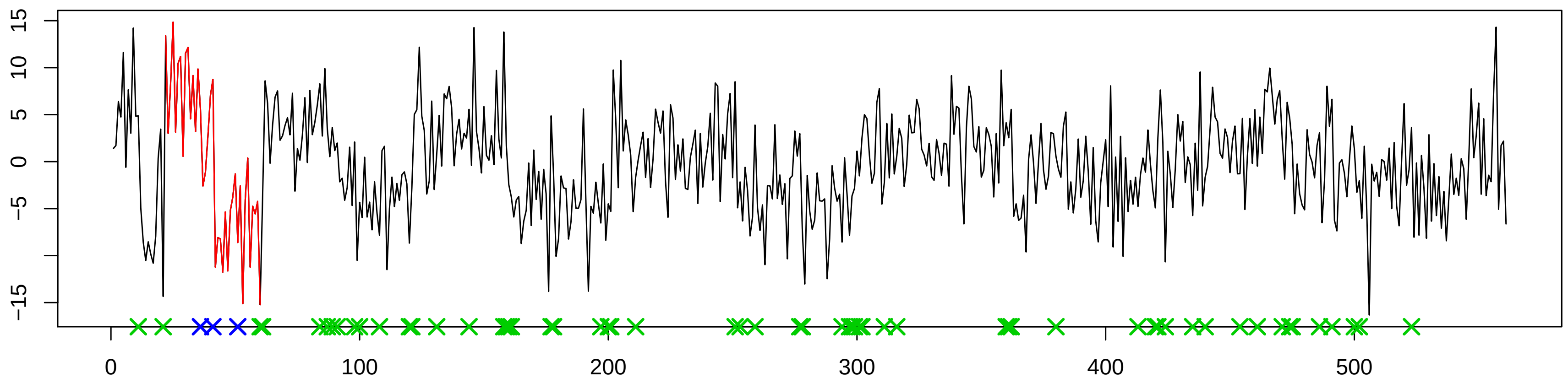
$$x_t,\ t \in [\widehat{k}_L + 1, \widehat{k}_R] = [22, 60] \text{ with } \widehat{k}^* = 41 \text{ and } \mathcal{D} = \{36, 41, 51\}.$$

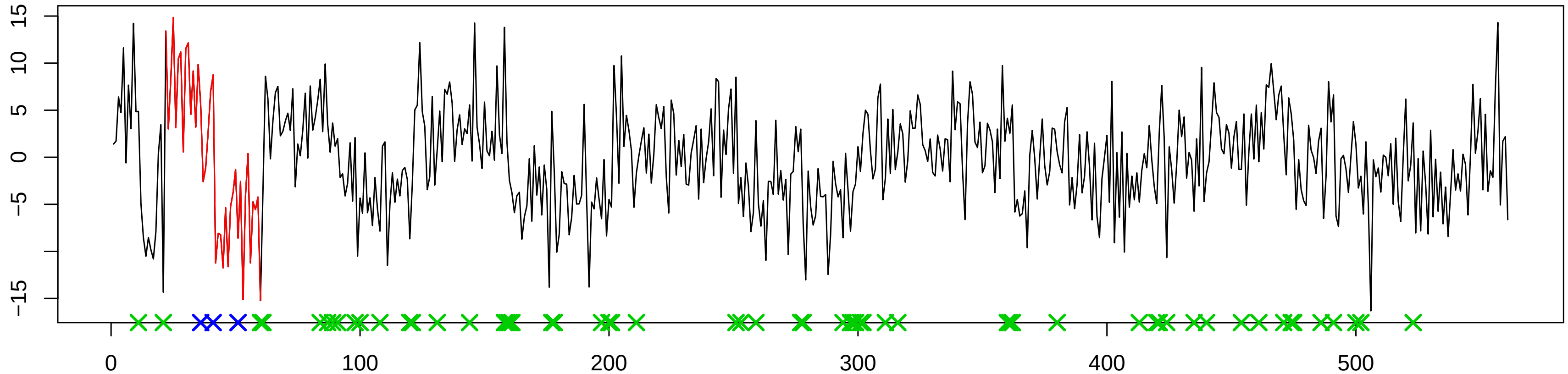
```
     CP G_l G_r      p-value      jump
[1,] 41  20  10 9.820958e-08 3.4338066
```

Step 4: Letting $\mathcal{I} = [\widehat{k}_L + 1, \widehat{k}_R]$, calculate SC at each subset of the conflicting candidates, $\mathcal{A} \subset \mathcal{D}$:

$$\mathrm{SC}(\mathcal{A}, \mathcal{I}, \mathcal{C}) = (n/2) \log \left\{ \frac{\mathrm{RSS}(\mathcal{A} \cup (\mathcal{C} \setminus \mathcal{I}))}{n} \right\} + \xi(n) \cdot (|\mathcal{A}| + |\mathcal{C} \setminus \mathcal{I}|),$$

where $\mathcal{A} = \emptyset, \{36\}, \{41\}, \{51\}, \{36, 41\}, \{36, 51\}, \{41, 51\}, \{36, 41, 51\}$.





- SC takes the **whole** $X_t,\ t = 1, \ldots, n$ into account.
- Using thus-defined SC, we perform an **adaptively chosen subset** of exhaustive search over all possible subsets of $\widehat{\mathcal{B}}_{\text{pool}}$ using the information criterion.
- Does not require maximum number of change-points as an input.
- Due to how we define the local environment $\mathcal{I}$, the computation is facilitated as we only need to compute and store $\sum_{t=\widehat{k}_j+1}^{\widehat{k}_{j+1}} X_t$ and $\sum_{t=\widehat{k}_j+1}^{\widehat{k}_{j+1}} X_t^2$.

Step 4 (cont'd): look for $\widehat{\mathcal{A}} \subset \mathcal{D}$ such that

(a) adding further candidates to $\widehat{\mathcal{A}}$ **monotonically increases** SC;

(b) removing **any single estimate** from $\widehat{\mathcal{A}}$ increases SC;

(c) $|\widehat{\mathcal{A}}| = \min\{|\mathcal{A}| : \mathcal{A} \subset \mathcal{D}$ satisfies (a)–(b)$\}$.

```
      36 41 51       sc
[1,]   1  1  1 1223.574
[2,]   0  1  1 1219.207
[3,]   1  0  1 1227.901
[4,]   0  0  1 1253.316
[5,]   1  1  0 1216.017
[6,]   0  1  0 1211.649
[7,]   1  0  0 1221.799
[8,]   0  0  0 1261.026
```

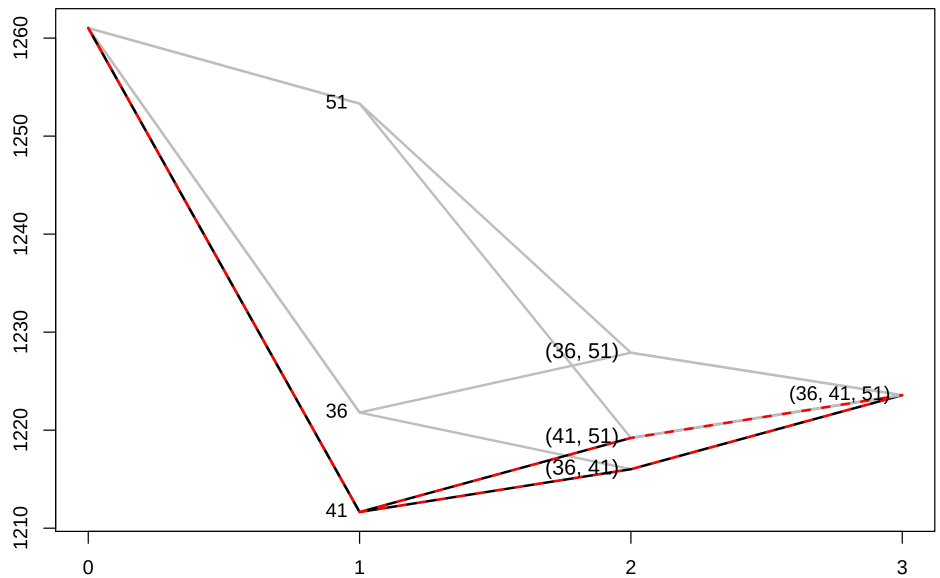
Step 4 (cont'd): look for $\widehat{\mathcal{A}} \subset \mathcal{D}$ such that

(a) adding further candidates to $\widehat{\mathcal{A}}$ **monotonically increases** SC;

(b) removing **any single estimate** from $\widehat{\mathcal{A}}$ increases SC;

(c) $|\widehat{\mathcal{A}}| = \min\{|\mathcal{A}| : \mathcal{A} \subset \mathcal{D}$ satisfies (a)–(b)$\}$.

† Such $\widehat{\mathcal{A}}$ does not always coincides with $\arg\min_{\mathcal{A}\subset\mathcal{D}}$ SC$(\mathcal{A}, \mathcal{I}, \mathcal{C})$.



† Efficient computation via bitwise iteration.

Step 5: add $\widehat{\mathcal{A}}$ to $\widehat{\mathcal{B}}$, remove $\widehat{\mathcal{A}}$ from $\mathcal{P}$ as well as those $\widehat{k} \in \mathcal{D}$ whose detection intervals are contained in $[\widehat{k}_L + 1, \widehat{k}_R]$, update $\mathcal{C} \leftarrow \mathcal{P} \cup \widehat{\mathcal{B}}$ and proceed to the next iteration.

Repeat the steps 1–5 until $\mathcal{P}$ is empty.

Steps 1–3: identify $\widehat{k}^* \in \mathcal{P}$ with {largest $\mathcal{J}(\widehat{k}^*)$, smallest $p(\widehat{k}^*)$}, its neighbours $\mathcal{D}$ and local environment $\mathcal{I} = [\widehat{k}_L + 1, \widehat{k}_R]$.

Step 4: Look for a subset $\widehat{\mathcal{A}} \subset \mathcal{D}$ so that

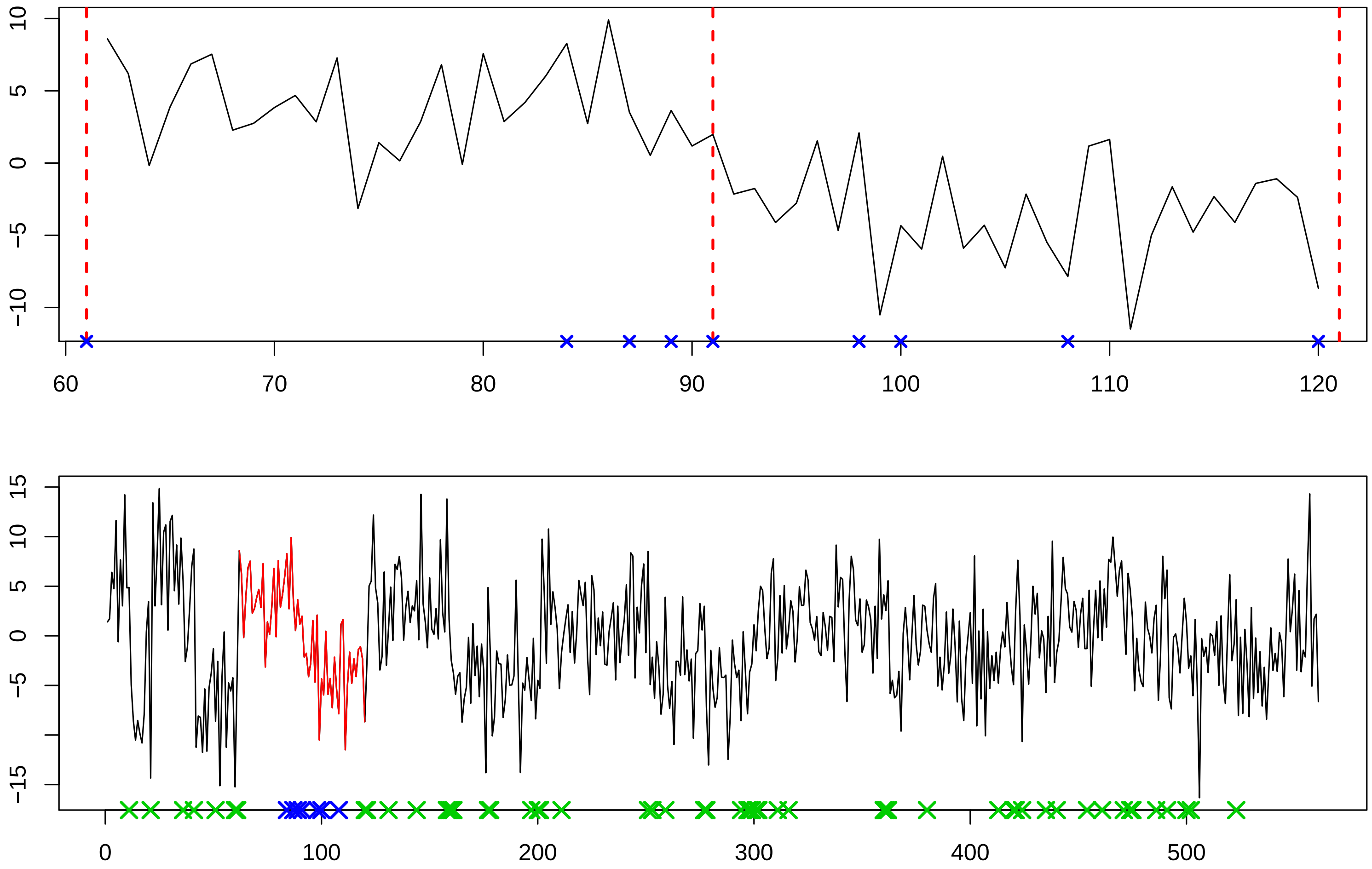
(a) adding further candidates to $\widehat{\mathcal{A}}$ **monotonically increases** SC;

(b) removing **any single estimate** from $\widehat{\mathcal{A}}$ increases SC;

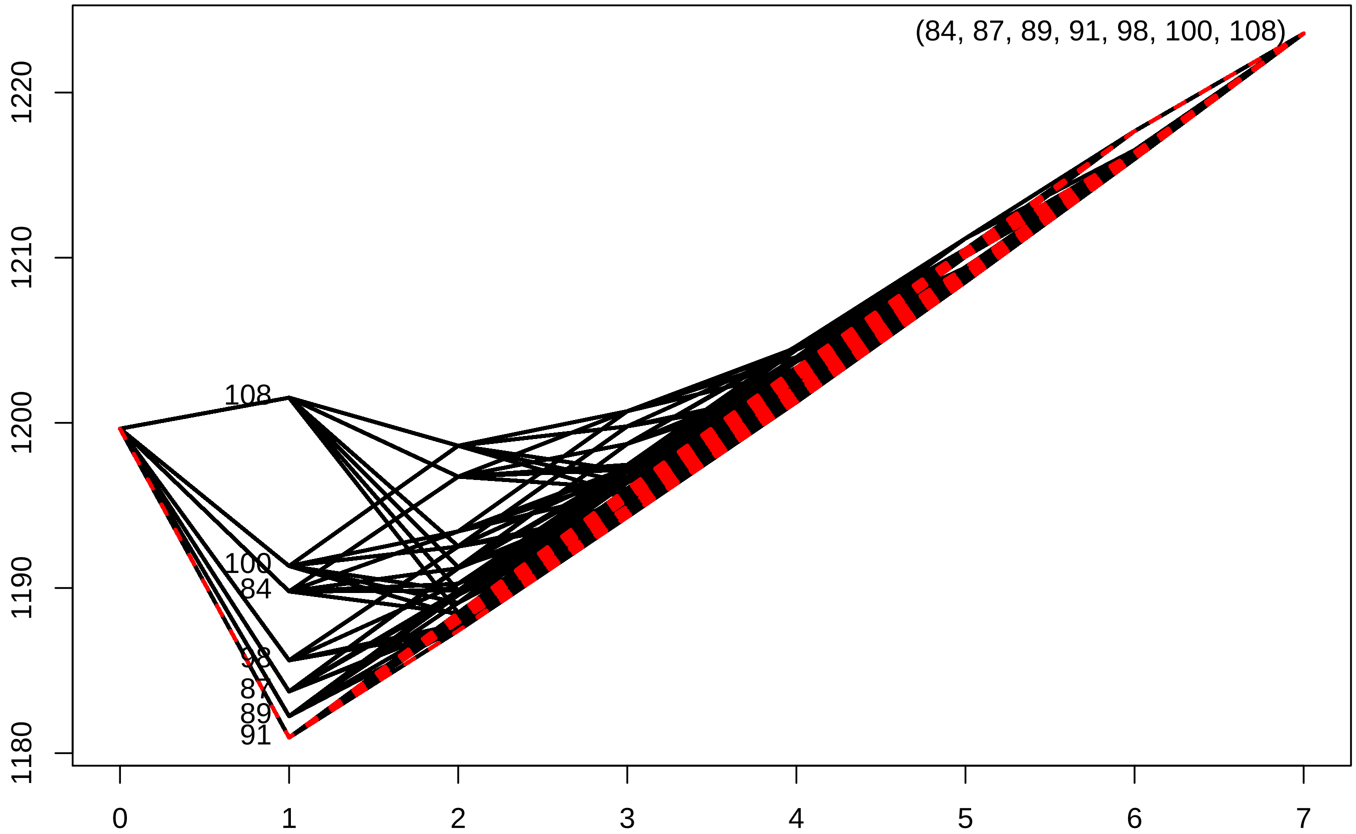
(c) $|\widehat{\mathcal{A}}| = \min\{|\mathcal{A}| : \mathcal{A} \subset \mathcal{D}$ satisfies (a)–(b)$\}$.

Step 5: update $\widehat{\mathcal{B}}$, $\mathcal{P}$ and $\mathcal{C}$.

Next iteration



Next iteration (cont'd)



Multiscale MOSUM procedure with localised pruning

Multiscale **MOSUM procedure** with localised pruning

With $\mathbf{G} = (G_l, G_r)$ as the bandwidth, MOSUM detector:

$$T_{k,n}(\mathbf{G}) = \sqrt{\frac{G_l G_r}{G_l + G_r}} \left| \frac{1}{G_l} \sum_{t=k-G_l+1}^{k} X_t - \frac{1}{G_r} \sum_{t=k+1}^{k+G_r} X_t \right|$$

for $k = 1, \ldots, n-1$.

Asymptotic distribution under H_0

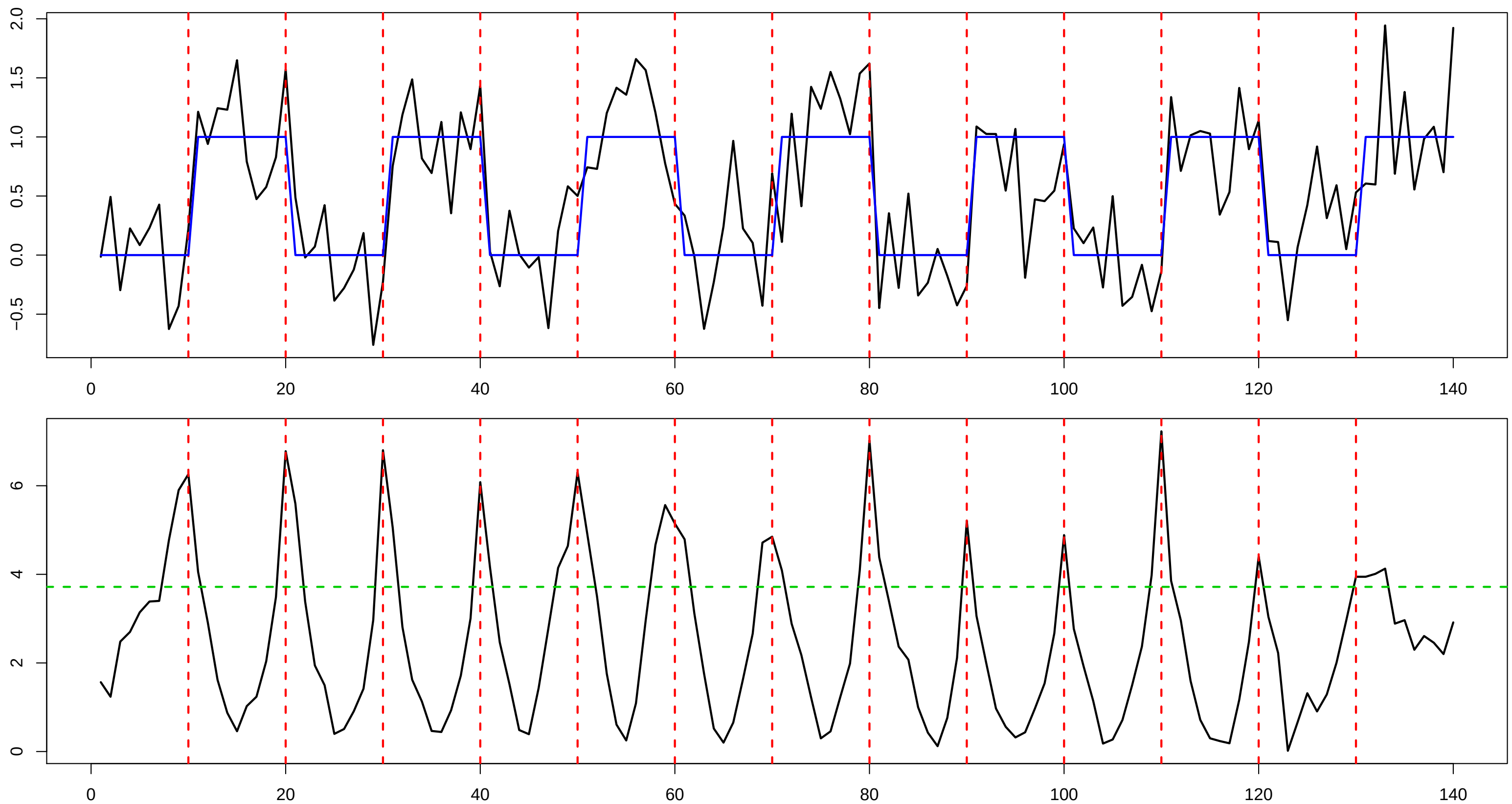
Under H_0: $q_n = 0$ (no change-points), we have $T_n(\mathbf{G}) = \max_k \sigma^{-1} T_{k,n}(\mathbf{G})$ satisfy

$$a(n, \mathbf{G}) T_n - b(n, \mathbf{G}) \to_d \Gamma,$$

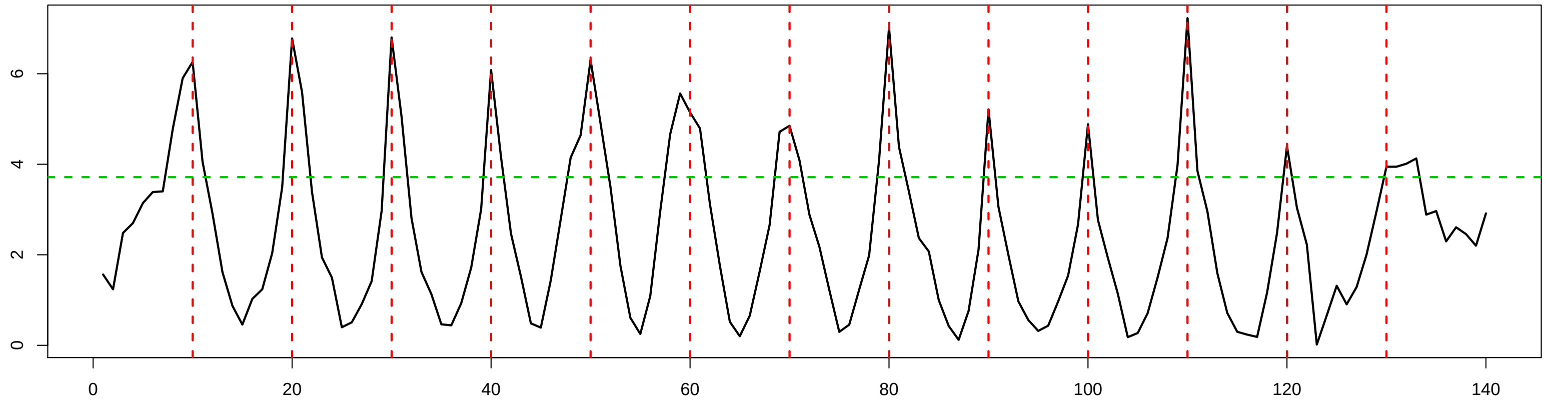
where Γ follows a Gumbel extreme value distribution.

(Eichinger & Kirch, 2017; Meier, Kirch & Cho, 2018)

Multiple change-point estimation



Multiple change-point estimation



Locate multiple change-points as $\widehat{k}_j^{\mathbf{G}}$, $j = 1, \ldots, \widehat{q}_{\mathbf{G}}$, where

(a) $\widehat{\sigma}^{-1} T_{\widehat{k}_j^{\mathbf{G}}, n}(\mathbf{G}) > D_n(\mathbf{G}, \alpha)$ (from the asymptotic distribution under H_0), and

(b) $\widehat{\sigma}^{-1} T_{\widehat{k}_j^{\mathbf{G}}, n}(\mathbf{G})$ is the local maximum over $\widehat{k}_j^{\mathbf{G}} - \eta G_l < k < \widehat{k}_j^{\mathbf{G}} + \eta G_r$ for some $\eta \in (0, 1]$.

For each estimate $\widehat{k}_j^{\mathbf{G}}$, we have $\mathcal{I}(\widehat{k}_j^{\mathbf{G}}) = [\widehat{k}_j^{\mathbf{G}} - G_l + 1, \widehat{k}_j^{\mathbf{G}} + G_r]$, $p(\widehat{k}_j^{\mathbf{G}})$ and $\mathcal{J}(\widehat{k}_j^{\mathbf{G}})$.
$\Longrightarrow$ input to the localised pruning.

Theoretical consistency for multiscale MOSUM procedure with localised pruning

Conditions

For $X_t = f_t + \varepsilon_t$, assume

(i) **Invariance principle:** for a standard Wiener process $\{W(k) : 1 \le k \le n\}$ and $\lambda_n = o(\sqrt{n})$,

$$\max_{1\le k\le n} \Big| \sum_{t=1}^{k} \varepsilon_t - \tau W(k) \Big| = O(\lambda_n) \quad \text{a.s.}$$

with $\tau^2 = \sigma^2 + \sum_{h>0} \mathrm{Cov}(\varepsilon_0, \varepsilon_h)$.

(ii) For some fixed $\gamma > 2$ and $C_0 > 0$, it holds for any $-\infty < l \le r < \infty$,

$$\mathrm{E} \left|\sum_{t=l}^{r} \varepsilon_t\right|^{\gamma} \le C_0 (r - l + 1)^{\gamma/2}.$$

Consistency of single-bandwidth MOSUM procedure

Let $d_j = |f_j - f_{j-1}|$ (jump size) and suppose

(i) $\min_j (k_{j+1} - k_j) > 2G$,

(ii) $\min_j d_j^2 G \geq c q_n^{2/\gamma} \log n$ for some $c > 0$,

(iii) $n/G \to \infty$ and $\lambda_n^2 \log n/G \to 0$.

Then, estimated change-points $\widehat{\mathcal{B}}(G) = \{\widehat{k}_j^G,\, j = 1, \ldots, \widehat{q}_G\}$ satisfy

$$\mathsf{P}\left\{\widehat{q}_G = q_n;\ |k_j - \widehat{k}_j^G| < c_0 q_n^{2/\gamma} d_j^{-2} \log n\, \forall\, j = 1, \ldots, q_n\right\} \to 1$$

for some fixed $c_0 > 0$.

† With fixed d_j and q_n, we have minimax optimality in estimating k_j up to a logarithmic factor.

Theoretical consistency for multiscale MOSUM procedure with localised pruning

Consistency of multiscale MOSUM procedure

For a set of bandwidths $\mathcal{H}$, assume that for each k_j, there exists $G_{h(j)} \in \mathcal{H}$ such that

(i') $\min(k_j - k_{j-1}, k_{j+1} - k_j) > 2G_{h(j)}$,

(ii') $d_j^2 G_{h(j)} \geq c q_n^{2/\gamma} \log\, n$.

(iii') $n/(\max_j G_{h(j)}) \to \infty$ and $\lambda_n^2 \log\, n/(\min_j G_{h(j)}) \to 0$.

† If there are multiple such bandwidths for k_j, let $G_{h(j)}$ be their minimum.

Then, **pooling all change-point estimates** $\widehat{\mathcal{B}}_{\text{pool}} = \cup_{G \in \mathcal{H}} \widehat{\mathcal{B}}(G)$, we have

$$\mathsf{P}\left\{ \min_{\widehat{k} \in \widehat{\mathcal{B}}_{\text{pool}}} |k_j - \widehat{k}| < c_0 q_n^{2/\gamma} d_j^{-2} \log\, n \text{ for all } j = 1, \ldots, q_n \right\} \to 1.$$

$\Longrightarrow$ $\exists$ at least one 'valid' estimate for each k_j in $\widehat{\mathcal{B}}_{\text{pool}}$.

Theoretical consistency for multiscale MOSUM procedure with localised pruning

Choice of penalty in SC

Closely related to the behaviour of $\max_{1\le l<r\le n}(r-l+1)^{-1/2}|\sum_{t=l}^{r}\varepsilon_t|$.

- If ε_t i.i.d. with mgf, set $\xi(n)=\log^{1+\delta} n$ for some small $\delta>0$ (Shao, 1995).
- If $\mathrm{E}|\varepsilon_t|^{2+\Delta}<\infty$ for some $\Delta>0$, set $\xi(n)=n^{\frac{2}{3+\Delta}+\delta}$ (Mikosch & Račkauskas, 2010).

Consistency of multiscale MOSUM procedure with localised pruning

(ii") $d_j^2 G_{h(j)} \geq c\xi(n)$.

Result 1. For any given interval $[s, e]$ and $\bar{c} \in (0, 1]$, define

$$\mathcal{B}_{s,e} = \{k_{j_s+j} \in [s, e];\ \min(k_{j_s+j} - s, e - k_{j_s+j}) \geq \bar{c} G_{h(j_s+j)}\}$$

with $q_{s,e} = |\mathcal{B}_{s,e}|$. Then, **exhaustive search in Step 4** yields $\widehat{\mathcal{B}}_{s,e} = \{\widehat{k}_{j_s+j},\ j = 1, \ldots, \widehat{q}_{s,e}\}$ that consistently estimates $\mathcal{B}_{s,e}$: $\widehat{q}_{s,e} = q_{s,e}$ and

$$|\widehat{k}_{j_s+j} - k_{j_s+j}| < c_0 q_n^{2/\gamma} d_{j_s+j}^{-2} \log n \text{ for all } j = 1, \ldots, q_{s,e}$$

with probability tending to one.

Result 2. Repeated application of Steps 1–5 yields $\widehat{\mathcal{B}} = \{\widehat{k}_1, \ldots, \widehat{k}_{\widehat{q}}\}$ satisfying

$$\mathsf{P}\left\{\widehat{q} = q_n;\ |\widehat{k}_j - k_j| < c_0 q_n^{2/\gamma} d_j^{-2} \log n \text{ for all } j = 1, \ldots, q_n\right\} \to 1.$$

Simulation study

Compare the **multiscale MOSUM procedure with localised pruning ('CK')** against the **bottom-up** merging approach (Messer et al. 2014), Pruned Exact Linear Time (**PELT**, Killick et al. 2012), Wild Binary Segmentation combined with BIC (**WBS**, Fryzlewicz 2014), pruned dynamic programming algorithm (**S3IB**, Rigaill 2010), **cumSeg** (Muggeo & Adelfio 2010), Tail-Greedy Unbalanced Haar (**TGUH**, Fryzlewicz 2018+), its hybrid with adaptive WBS (**hybrid**), and multiscale segmentation method (**FDRSeg**, Li et al. 2016).

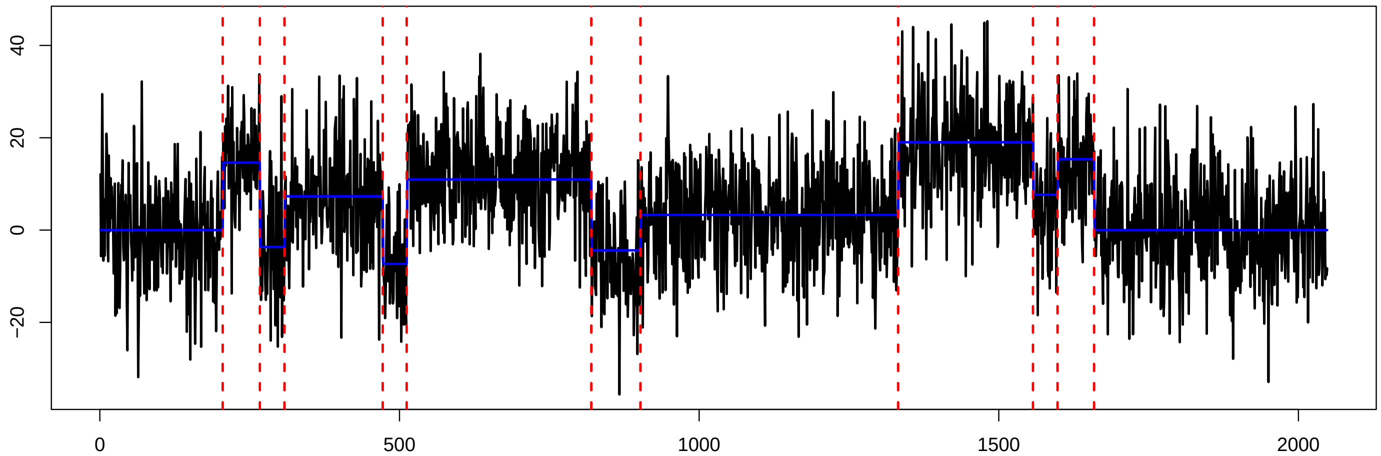
Choice of parameters: $\mathcal{H} = \{10, 20, 40, 60, 80, 100\}$ and $\alpha = 0.2$.

Report the **true positive rate** ($|\widehat{\mathcal{B}} \cap \mathcal{B}|/q_n$), **false positive rate** ($|\widehat{\mathcal{B}} \cap \mathcal{B}^c|/\widehat{q}$), **Adjusted Rand Index (ARI)** of the estimated segmentation, **mean squared error (MSE)** of $\widehat{f}$, **Hausdorff distance** between $\mathcal{B}$ and $\widehat{\mathcal{B}}$, and weighted average of **trimmed distances** $\delta_{\text{trim}} = (\sum_{j=1}^{q_n} d_j^2)^{-1} \sum_{j=1}^{q_n} d_j^2 \cdot \delta_{\text{trim},j}$ where

$$\delta_{\text{trim},j} = \left(\frac{k_{j+1} - k_j}{2} \wedge \frac{k_j - k_{j-1}}{2} \wedge \min_{1 \le j' \le \widehat{q}} |\widehat{k}_{j'} - k_j| \right)$$

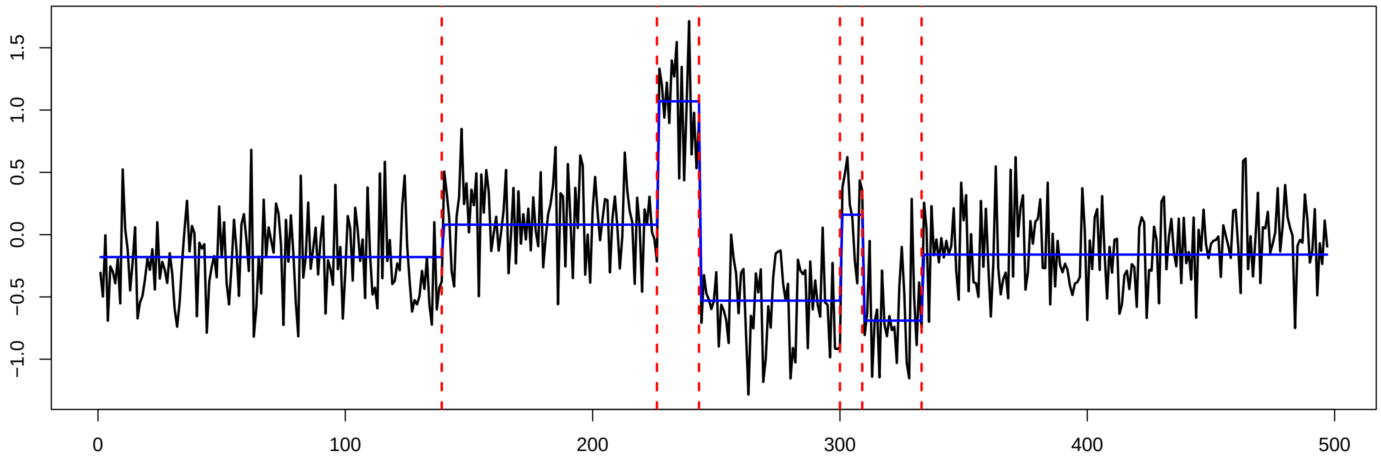
averaged over 1000 replications.

`blocks`: $n = 2048$, $q_n = 11$, Gaussian noise, $\xi(n) = \log^{1.01} n$.



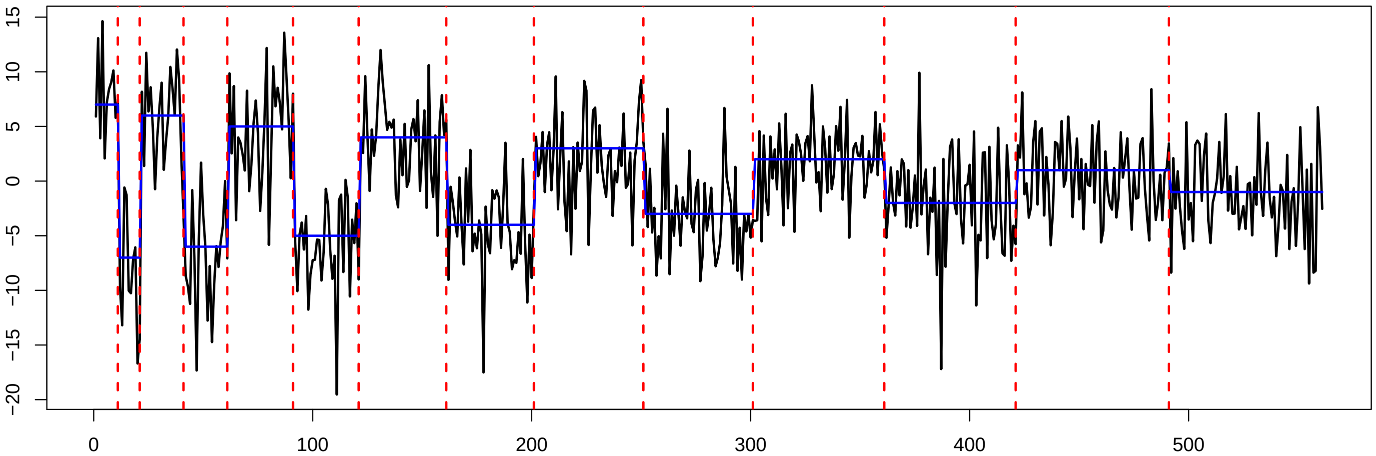
method	TPR	FPR	ARI	MSE	d_H	δ_{trim}
CK	0.96	0.02	0.973	4.975	38.452	339.349
`bottom.up`	0.959	0.288	0.864	5.94	144.947	353.589
WBS	0.954	0.008	0.979	5.105	29.785	374.093
PELT	0.876	0.001	0.96	6.355	58.619	582.304
S3IB	0.971	0.02	0.979	4.749	30.868	312.663
cumSeg	0.775	0.002	0.914	12.914	83.402	1706.121
TGUH	0.947	0.025	0.965	6.519	42.168	499.076
hybrid	0.945	0.027	0.971	5.342	45.131	370.507
FDRSeg	0.973	0.081	0.956	6.033	67.488	450.989

`fms`: $n = 497$, $q_n = 6$, Gaussian noise, $\xi(n) = \log^{1.01} n$.



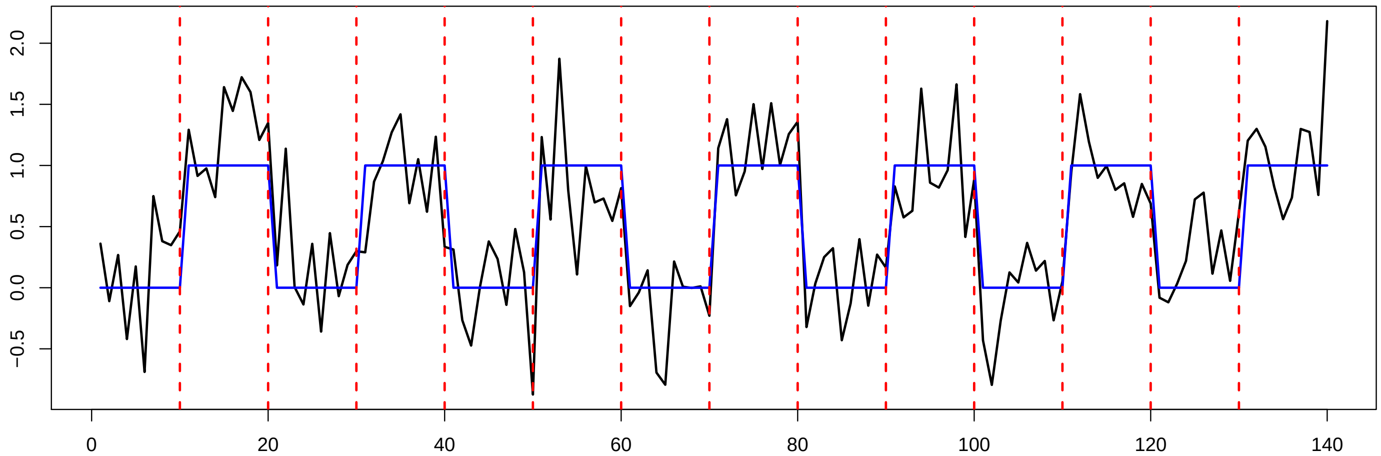
method	TPR	FPR	ARI	MSE	d_H	δ_{trim}
CK	0.996	0.037	0.949	4.315	16.215	0.155
`bottom.up`	0.974	0.301	0.817	6.355	60.737	0.351
WBS	0.996	0.006	0.968	3.675	7.477	0.138
PELT	0.927	0.001	0.953	5.012	14.24	0.431
S3IB	1	0.097	0.942	4.777	27.953	0.104
cumSeg	0.731	0.011	0.916	14.234	30.087	1.961
TGUH	0.996	0.039	0.945	4.765	15.782	0.159
hybrid	0.997	0.037	0.954	3.916	14.705	0.127
FDRSeg	0.999	0.099	0.941	13.165	20.812	1.671

$\texttt{mix}$: $n = 560$, $q_n = 13$, Gaussian noise, $\xi(n) = \log^{1.01} n$.



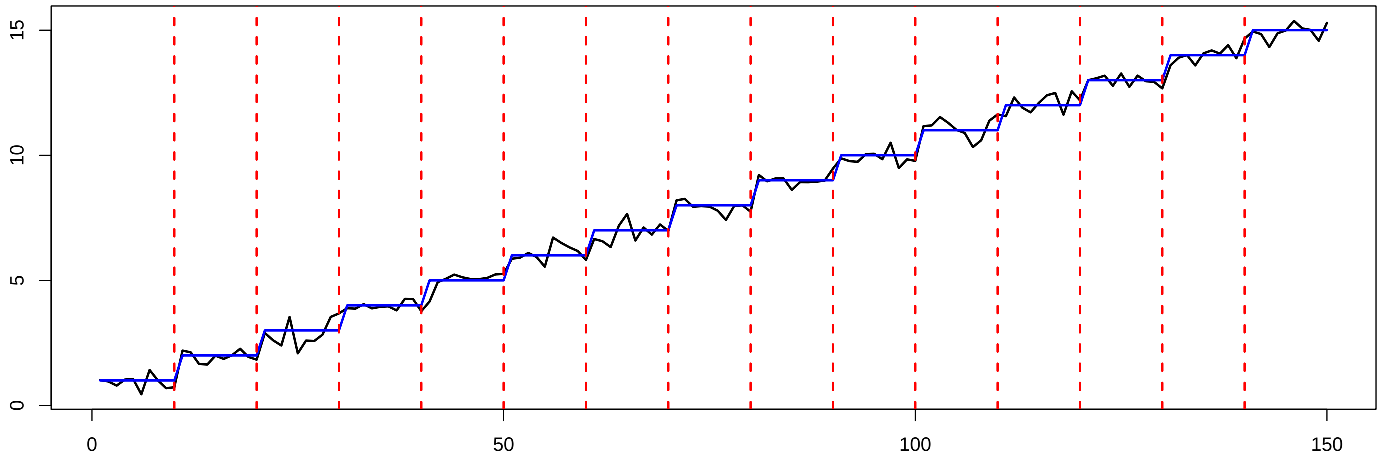
method	TPR	FPR	ARI	MSE	d_H	δ_{trim}
CK	0.941	0.014	0.792	4.145	57.094	30.099
`bottom.up`	0.961	0.06	0.823	4.298	43.313	32.504
WBS	0.91	0.008	0.733	4.377	80.408	33.351
PELT	0.772	0.002	0.462	6.135	184.846	46.501
S3IB	0.961	0.073	0.817	4.771	43.354	31.722
cumSeg	0.315	0	0.255	25.587	101.915	796.096
TGUH	0.9	0.026	0.697	5.474	88.063	46.775
hybrid	0.905	0.02	0.723	4.53	83.754	33.750
FDRSeg	0.936	0.086	0.756	8.164	58.865	136.397

`teeth10`: $n = 140$, $q_n = 13$, Gaussian noise, $\xi(n) = \log^{1.01} n$.



method	TPR	FPR	ARI	MSE	d_H	δ_{trim}
CK	0.968	0.001	0.932	2.389	3.591	0.298
`bottom.up`	0.967	0.004	0.942	2.584	4.524	0.282
WBS	0.943	0.02	0.865	4.277	4.628	0.641
PELT	0.378	0.007	0.274	13.16	43.907	3.256
S3IB	0.997	0.099	0.904	4.014	3.882	0.387
cumSeg	0.001	0	0	18.382	1.881	4.994
TGUH	0.961	0.017	0.866	4.396	5.161	0.634
hybrid	0.964	0.014	0.886	3.917	4.644	0.543
FDRSeg	0.964	0.054	0.743	7.986	5.714	1.280

`stairs10`: $n = 140$, $q_n = 13$, Gaussian noise, $\xi(n) = \log^{1.01} n$.



method	TPR	FPR	ARI	MSE	d_H	δ_{trim}
CK	0.998	0.003	0.977	2.184	1.216	0.113
`bottom.up`	0.995	0.006	0.968	2.559	1.985	0.152
WBS	1	0.036	0.958	2.937	2.207	0.168
PELT	0.992	0	0.965	2.715	1.882	0.179
S3IB	1	0.089	0.953	3.078	3.245	0.135
cumSeg	0.986	0.006	0.877	7.351	3.253	0.646
TGUH	1	0.009	0.962	2.873	1.575	0.179
hybrid	0.999	0.011	0.956	3.221	1.944	0.214
FDRSeg	1	0.051	0.79	11.52	2.739	1.020

ℓ_1-error between $\widehat{\mathcal{B}}$ and $\mathcal{B}$ conditional on our method ('CK') and the competitor correctly estimating all q_n change-points.

model			WBS	S3IB	cumSeg	TGUH	hybrid	FDRSeg
`blocks`	ℓ_1	CK	2.039	2.098	NA	2.038	2.03	2.093
		competitor	2.14	2.061	NA	3.106	2.153	2.555
	-	all detection	0.401	0.413	0	0.328	0.314	0.326
`fms`	ℓ_1	CK	1.188	1.143	1.17	1.183	1.18	1.188
		competitor	1.113	1.023	1.641	1.591	1.116	1.744
	-	all detection	0.81	0.526	0.155	0.725	0.73	0.688
`mix`	ℓ_1	CK	1.784	1.843	NA	1.8	1.748	1.671
		competitor	1.781	1.805	NA	2.319	1.742	2.303
	-	all detection	0.22	0.159	0	0.159	0.162	0.129
`teeth10`	ℓ_1	CK	0.127	0.132	NA	0.126	0.126	0.133
		competitor	0.329	0.315	NA	0.404	0.33	1.085
	-	all detection	0.499	0.218	0	0.5	0.512	0.413
`stairs10`	ℓ_1	CK	0.1	0.099	0.102	0.101	0.101	0.103
		competitor	0.157	0.124	0.559	0.167	0.195	1.020
	-	all detection	0.578	0.298	0.729	0.837	0.819	0.765

Heavy-tail

t_5 noise, $\xi(n) = \log^{1.1} n$ (light) and $\xi(n) = n^{2/5.1}$ (heavy).

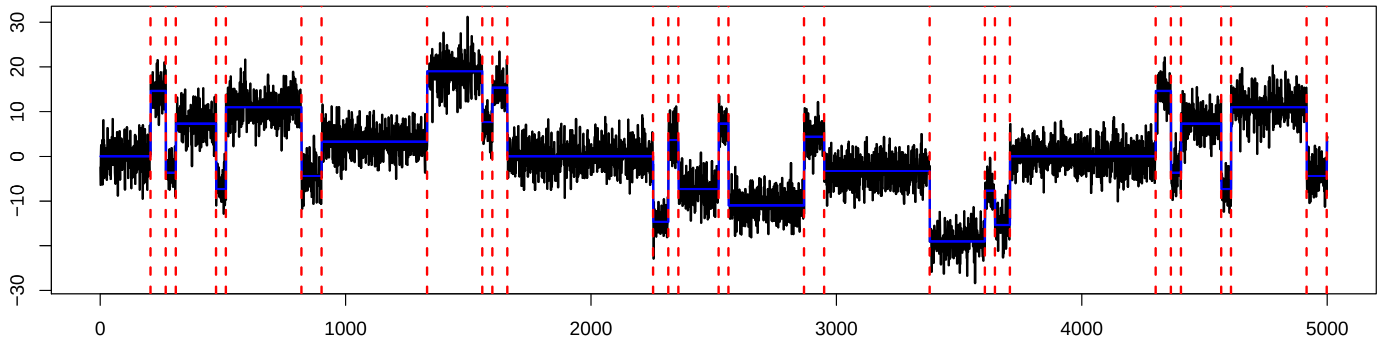
model	method	penalty	TPR	FPR	ARI	MSE	d_H	δ_{trim}
`blocks`	CK	light	0.943	0.005	0.978	5.113	35.346	346.064
		heavy	0.768	0	0.923	11.484	78.559	1068.298
	`bottom.up`	-	0.979	0.354	0.822	6.143	184.263	321.292
	cumSeg	-	0.773	0.003	0.913	13.428	85.287	1737.013
`fms`	CK	light	0.987	0.013	0.964	4.281	10.196	0.205
		heavy	0.951	0.002	0.938	5.483	17.847	0.313
	`bottom.up`	-	0.988	0.38	0.757	6.76	88.055	0.283
	cumSeg	-	0.737	0.013	0.915	14.94	30.354	1.889
`mix`	CK	light	0.914	0.005	0.747	4.092	75.677	34.143
		heavy	0.855	0.001	0.634	4.72	118.595	39.439
	`bottom.up`	-	0.983	0.099	0.851	4.016	33.257	33.985
	cumSeg	-	0.32	0	0.258	24.689	104.038	788.816
`teeth10`	CK	light	0.92	0.001	0.875	3.154	6.74	0.522
		heavy	0.894	0.001	0.847	3.534	8.572	0.642
	`bottom.up`	-	0.981	0.004	0.953	2.227	3.13	0.226
	cumSeg	-	0.001	0	0	18.516	1.92	4.994
`stairs10`	CK	light	0.995	0.002	0.973	2.328	1.639	0.135
		heavy	0.994	0.001	0.973	2.338	1.66	0.137
	`bottom.up`	-	0.991	0.004	0.962	2.864	2.434	0.189
	cumSeg	-	0.982	0.007	0.878	7.513	3.312	0.645

Scalability

Proposed methodology is less affected by increasing n due to its localised approach.

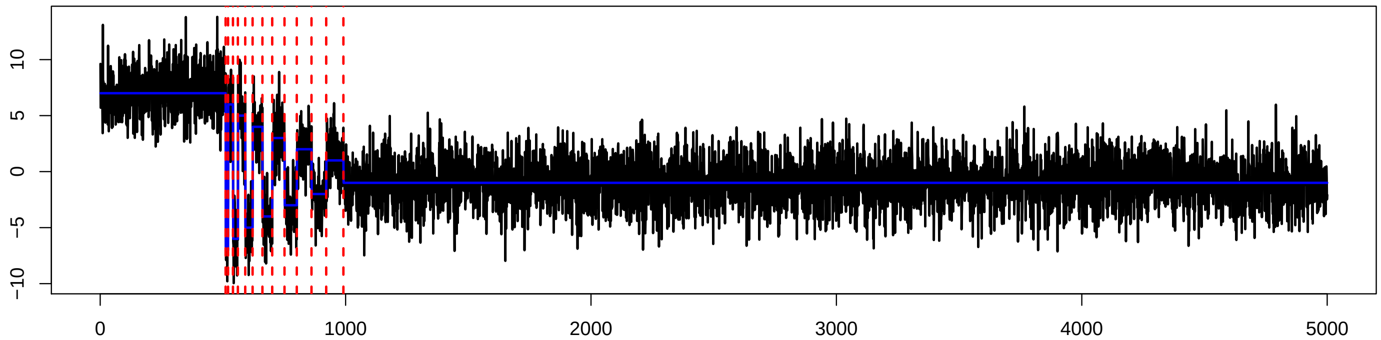
Dense: repeat the five test signals until $n > 2 \times 10^4$.
Sparse: embed the test signals in a sequence of $n = 2 \times 10^4$ at $t = 500$.

Dense `blocks`: $n = 20480$, $q_n = 110$, Gaussian noise, $\xi(n) = \log^{1.01} n$.



method	TPR	FPR	ARI	MSE	d_H	δ_{trim}	speed
CK	0.928	0.004	0.98	5.002	87.97	2.364	1.901
`bottom.up`	0.919	0.207	0.906	6.013	205.63	2.58	0.225
`wbs.c`	0.854	0.029	0.944	7.627	171.92	4.288	32.416
`wbs.sbic`	0.927	0.035	0.959	5.602	165.82	2.576	85.722
PELT	0.808	0	0.954	8.107	107.747	6.451	0.017
TGUH	0.92	0.006	0.974	6.269	98.24	3.259	1.028
hybrid	0.895	0.033	0.955	6.282	165.2	2.989	22.742

Sparse `mix`: $n = 2 \times 10^4$, $q_n = 13$, Gaussian noise, $\xi(n) = \log^{1.01} n$.



method	TPR	FPR	ARI	MSE	d_H	δ_{trim}	speed
CK	0.89	0.017	0.871	4.81	1,203.516	0.781	0.373
`bottom.up`	0.909	0.353	0.072	5.829	15,764.76	0.785	0.190
`wbs.c`	0.727	0.002	0.912	10.494	223.741	2.414	30.748
`wbs.sbic`	0.834	0.008	0.907	6.432	519.809	1.232	44.235
PELT	0.668	0.001	0.846	9.209	248.531	2.234	0.025
TGUH	0.538	NA	0.703	46.868	369.404	12.014	1.002
hybrid	0.537	NA	0.712	48.392	499.946	12.638	24.926

Conclusions

- Computationally efficient approach to pruning down (potentially) a large number of conflicting change-point estimates.
- Scalable to increasing number of observations and change-points.
- Achieves consistency in estimating both the total number and locations of change-points.
- Applicable in combination with any method that provides additional information about the local environment in which each change-point is obtained.
- Extendable to problems admitting the minimisation of cost $+$ penalty on the model complexity.
- R package `mosum` available on CRAN.

References

H. Cho & C. Kirch (2018+) Multiple change-point detection via multiscale MOSUM procedure with localised pruning. *In preparation*.

A. Meier, C. Kirch & **H. Cho** (2018) mosum: A package for moving sums in change point analysis. *In submission*.

R. Baranowski, Y. Chen & P. Fryzlewicz (2016) Narrowest-over-threshold detection of multiple change-points and change-point-like features, arXiv preprint arXiv:1609.00293.

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

P. Fryzlewicz (2014) Wild binary segmentation for multiple change-point detection. *AoS*, **42**: 2243–2281.

P. Fryzlewicz (2018+) Tail-greedy bottom-up data decompositions and fast multiple change-point detection. *To appear in AoS*.

R. Killick, P. Fearnhead & I. Eckley (2012) Optimal detection of changepoints with a linear computational cost. *JASA*, **107**: 1590–1598.

H. Li, A. Munk & H. Sieling (2016) FDR-control in multiscale change-point segmentation. *EJS*, **10**: 918–959.

M. Messer et al. (2014) A multiple filter test for the detection of rate changes in renewal processes with varying variance. *Annals of Applied Statistics*, **8**: 2027–2067.

V. Muggeo & G. Adelfio (2011). Efficient change point detection for genomic sequences of continuous measurements. Bioinformatics, **27**: 161–166.

G. Rigaill (2010) Pruned dynamic programming for optimal multiple change-point detection. Preprint.