Kernseg: A new efficient change-points detection procedure for analyzing biological data

ALAIN CELISSE

[1]UMR 8524 CNRS - Université Lille 1

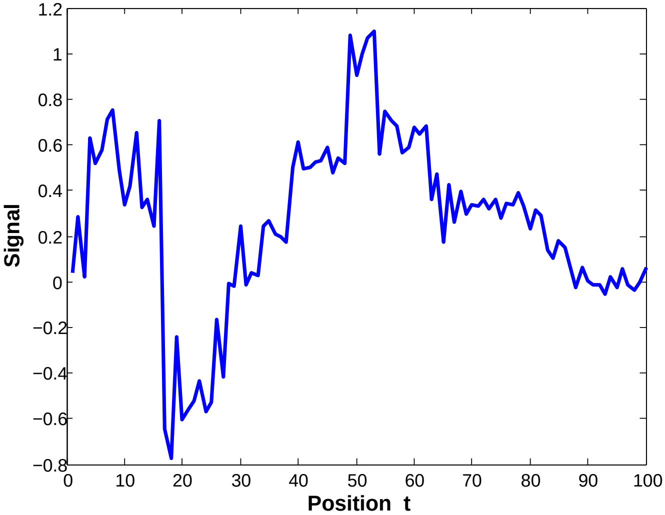
[2]MODAL INRIA team-project

joint work with G. Rigaill, M. Pierre-Jean, and G. Marot

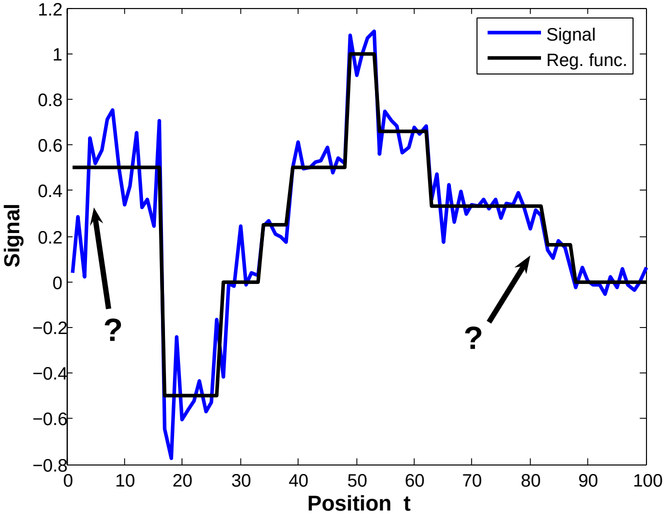
Stat4Bio – LaMME

Évry, April 3rd, 2019

Change-point detection: 1-D signal (example)



Change-point detection: 1-D signal (example)

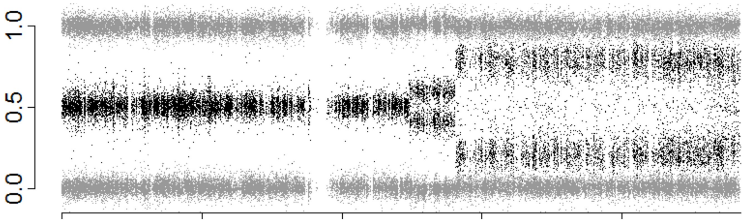


Detect abrupt changes. . .

General purposes:

1. Detect changes in (features of) the distribution (not only in the mean)

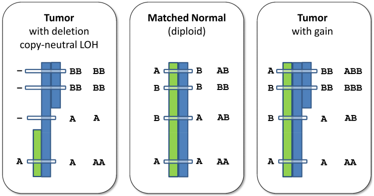
Example 1: Changes in the distribution



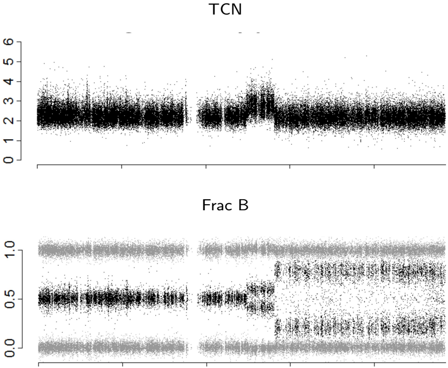
⟶ Detecting changes in the mean is useless

Example 1: Changes in the distribution

Total copy number (TCN) and Allele B fraction (Frac B)



$$(Frac\ B)_t = \frac{N_{B,t}}{N_{A,t}+N_{B,t}} \ .$$



Detect abrupt changes...

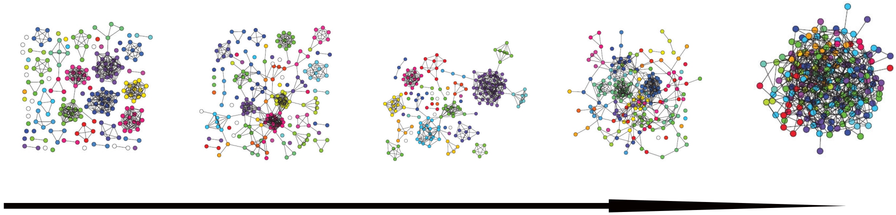
General purposes:

1. Detect changes in (features of) the distribution (not only in the mean)

2. Complex data:
    - High-dimension: measures in $\mathbb{R}^d$, curves,...
    - Structured: audio/video streams, graphs, DNA sequence,...

Motivating example 2: Structured objects

Observe networks along the time



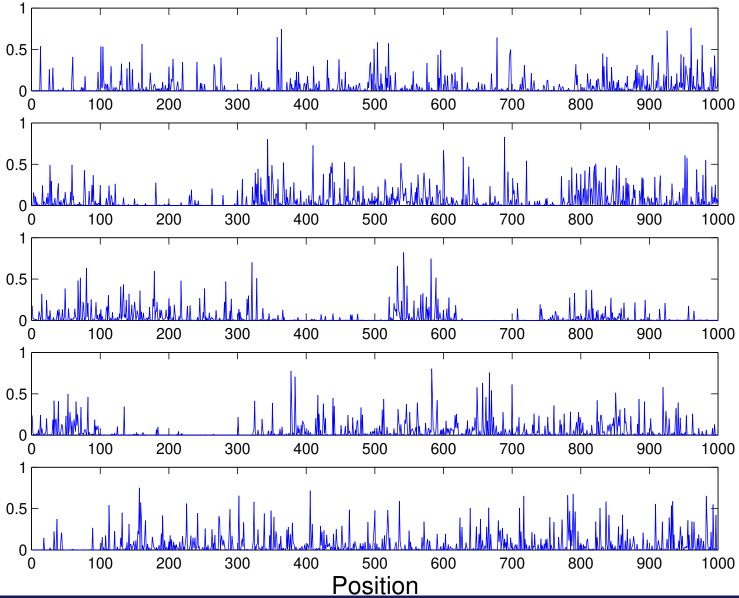
Goal:

Detect abrupt changes in some features of the network

Ex:

- Each network represented by one histogram
- Columns $\leftrightarrow$ counts of specific motifs (stars, triangles,...)

Motivating example 2: dynamic networks



Detect abrupt changes...

General purposes:

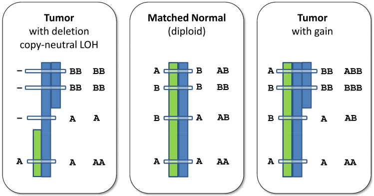
1. Detect changes in (features of) the distribution (not only in the mean)

2. Complex data:
   - High-dimension: measures in $\mathbb{R}^d$, curves,...
   - Structured: audio/video streams, graphs, DNA sequence,...

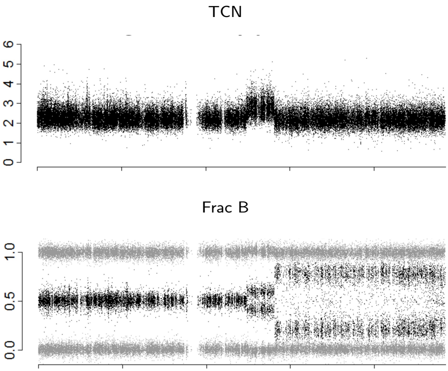
3. Fusion of heterogeneous data
   - Deal simultaneously with different types of complex data

Example 1 (Cont'd): Fusion of TCN and Frac B

Total copy number (TCN) and Allele B fraction (Frac B)



$$(Frac\ B)_t = \frac{N_{B,t}}{N_{A,t}+N_{B,t}} \ .$$



Detect abrupt changes. . .

General purposes:

1. Detect changes in (features of) the distribution (not only in the mean)

2. Complex data:
   - High-dimension: measures in $\mathbb{R}^d$, curves,. . .
   - Structured: audio/video streams, graphs, DNA sequence,. . .

3. Fusion of heterogeneous data
   - Deal simultaneously with different types of complex data

4. Efficient algorithm allowing to deal with large data sets ("Big data" challenge)

Outline

1. KCP: Kernel change-points detection proc.
2. Dyn. programming and reproducing kernels
3. Faster approximate algorithm
4. How many change-points? $\rightarrow$ penalty
5. Experiments on biological data

$\mathbb{I}$ Kernel change-points procedure

Basic notations

- Segmentation: $\tau = (\tau_1, \ldots, \tau_D)$

$$\{1, \ldots, n\} = [\tau_1, \tau_2[\cup [\tau_2, \tau_3[\cup \ldots \cup [\tau_{D-1}, \tau_D[\cup [\tau_D, \tau_{D+1}[$$

with $\tau_1 = 1$ and $\tau_{D+1} = n + 1$.

- $\mathcal{T}_n = \{\tau : \text{ segmentation of } \{1, \ldots, n\}\}$
- D_τ: number fo segments of τ
- $\mathcal{T}_n^D$: segmentations with D segments

Basic notations

- Segmentation: $\tau = (\tau_1, \ldots, \tau_D)$

  $$\{1, \ldots, n\} = [\tau_1, \tau_2[\cup [\tau_2, \tau_3[\cup \ldots \cup [\tau_{D-1}, \tau_D[\cup [\tau_D, \tau_{D+1}[$$

  with $\tau_1 = 1$ and $\tau_{D+1} = n + 1$.

- $\mathcal{T}_n = \{\tau : \text{ segmentation of } \{1, \ldots, n\}\}$

- D_τ: number fo segments of τ

- $\mathcal{T}_n^D$: segmentations with D segments

Basic notations

- Segmentation: $\tau = (\tau_1, \ldots, \tau_D)$

  $$\{1, \ldots, n\} = [\tau_1, \tau_2[\cup [\tau_2, \tau_3[\cup \ldots \cup [\tau_{D-1}, \tau_D[\cup [\tau_D, \tau_{D+1}[$$

  with $\tau_1 = 1$ and $\tau_{D+1} = n + 1$.

- $\mathcal{T}_n = \{\tau : \text{ segmentation of } \{1, \ldots, n\}\}$

- D_τ: number fo segments of τ

- $\mathcal{T}_n^D$: segmentations with D segments

Model selection-based procedure: KCP

Outline

1. For every $1 \leq D \leq D_{\max}$,

$$\widehat{\tau}^D \in \mathrm{Argmin}_{\tau \in \mathcal{T}_n^D} \mathcal{R}_{emp}(\tau) \ ,$$

2. Define

$$\widehat{D} = \mathrm{Argmin}_{1 \leq D \leq D_{\max}} \left\{ \mathcal{R}_{emp}(\widehat{\tau}^D) + \mathrm{pen}(\widehat{\tau}^D) \right\} \ .$$

3. Final segmentation:

$$\widehat{\tau} := \widehat{\tau}^{\widehat{D}}.$$

Rks

- $\mathcal{R}_{emp}(\tau)$ quantifies the mistakes (cost) of τ
- $\mathrm{pen}(\tau)$: penalty to be made precise

(Arlot, Celisse, Harchaoui (2018))

Model selection-based procedure: KCP

Outline

1. For every $1 \leq D \leq D_{\max}$,

$$\widehat{\tau}^D \in \operatorname{Argmin}_{\tau \in \mathcal{T}_n^D} \mathcal{R}_{emp}(\tau) \quad ,$$

2. Define

$$\widehat{D} = \operatorname{Argmin}_{1 \leq D \leq D_{\max}} \left\{ \mathcal{R}_{emp}(\widehat{\tau}^D) + \operatorname{pen}(\widehat{\tau}^D) \right\} \quad .$$

3. Final segmentation:

$$\widehat{\tau} := \widehat{\tau}^{\widehat{D}} .$$

Rks

- $\mathcal{R}_{emp}(\tau)$ quantifies the mistakes (cost) of τ
- $\operatorname{pen}(\tau)$: penalty to be made precise

(Arlot, Celisse, Harchaoui (2018))

Model selection-based procedure: KCP

Outline

1. For every $1 \leq D \leq D_{\max}$,

$$\widehat{\tau}^D \in \operatorname{Argmin}_{\tau \in \mathcal{T}_n^D} \mathcal{R}_{emp}(\tau) \enspace ,$$

2. Define

$$\widehat{D} = \operatorname{Argmin}_{1 \leq D \leq D_{\max}} \left\{ \mathcal{R}_{emp}(\widehat{\tau}^D) + \mathrm{pen}(\widehat{\tau}^D) \right\} \enspace .$$

3. Final segmentation:

$$\widehat{\tau} := \widehat{\tau}^{\widehat{D}} .$$

Rks

- $\mathcal{R}_{emp}(\tau)$ quantifies the mistakes (cost) of τ
- $\mathrm{pen}(\tau)$: penalty to be made precise

(Arlot, Celisse, Harchaoui (2018))

Model selection-based procedure: KCP

Outline

1. For every $1 \leq D \leq D_{\max}$,

$$\widehat{\tau}^D \in \operatorname{Argmin}_{\tau \in \mathcal{T}_n^D} \mathcal{R}_{emp}(\tau) \quad ,$$

2. Define

$$\widehat{D} = \operatorname{Argmin}_{1 \leq D \leq D_{\max}} \left\{ \mathcal{R}_{emp}(\widehat{\tau}^D) + \operatorname{pen}(\widehat{\tau}^D) \right\} \quad .$$

3. Final segmentation:

$$\widehat{\tau} := \widehat{\tau}^{\widehat{D}}.$$

Rks

- $\mathcal{R}_{emp}(\tau)$ quantifies the mistakes (cost) of τ
- $\operatorname{pen}(\tau)$: penalty to be made precise

(Arlot, Celisse, Harchaoui (2018))

Cost of a segmentation

Reproducing kernel

- $X_1, \ldots, X_n$: initial observations (structured objects)
  Ex: DNA sequences, networks, texts, ...

- $k(\cdot,\cdot) \to \mathbb{R}$: reproducing kernel (sdp, ...)
- $k(\cdot,\cdot)$ similarity measure between "objects"
  Ex : Gaussian kernel

$$k_\alpha(x_i, x_j) = \exp\left[-(x_i - x_j)^2/h\right], \quad h > 0$$

Empirical risk

$\mathcal{R}_{emp}(\tau)$

$$= \frac{1}{n}\sum_{i=1}^{n} k(X_i, X_i) - \frac{1}{n}\sum_{\ell=1}^{D} \underbrace{\left[\frac{1}{\tau_{\ell+1}-\tau_\ell}\sum_{i=\tau_\ell}^{\tau_{\ell+1}-1}\sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)\right]}_{=\text{Cost of } [\tau_\ell, \tau_{\ell+1}[}$$

Cost of a segmentation

Reproducing kernel

- $X_1, \ldots, X_n$: initial observations (structured objects)
  Ex: DNA sequences, networks, texts, ...

- $k(\cdot,\cdot) \to \mathbb{R}$: reproducing kernel (sdp, ...)
- $k(\cdot,\cdot)$ similarity measure between "objects"
  Ex : Gaussian kernel

$$k_\alpha(x_i, x_j) = \exp\left[-(x_i - x_j)^2/h\right], \ h > 0$$

Empirical risk

$$\mathcal{R}_{emp}(\tau) = \frac{1}{n}\sum_{i=1}^{n} k(X_i, X_i) - \frac{1}{n}\sum_{\ell=1}^{D} \underbrace{\left[\frac{1}{\tau_{\ell+1} - \tau_\ell}\sum_{i=\tau_\ell}^{\tau_{\ell+1}-1}\sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)\right]}_{=\text{Cost of } [\tau_\ell, \tau_{\ell+1}[}$$

Instances of reproducing kernels

- Polynomial kernel:

$$k_{\alpha,c}(x,y) = (x.y + c)^{\alpha}, \quad c, \alpha \geq 0 \ .$$

- χ^2-kernel:

$$k_I(p,q) = \exp\left[-\sum_{i=1}^{I} \frac{(p_i - q_i)^2}{p_i + q_i} \right] \ .$$

- Combination of kernels:
  With $x = (x_1, x_2)$ and $y = (y_1, y_2)$, $(\alpha \in [0,1])$

$$k_{\alpha}(x,y) = \alpha k_1(x_1, y_1) + (1-\alpha) k_2(x_2, y_2).$$

Model selection-based procedure: KCP

Outline

1. For every $1 \leq D \leq D_{\max}$,

$$\widehat{\tau}^{D} \in \mathrm{Argmin}_{\tau \in \mathcal{T}_n^D} \mathcal{R}_{emp}(\tau) \enspace ,$$

$\longrightarrow$ Dynamic programming

2. Define

$$\widehat{D} = \mathrm{Argmin}_{1 \leq D \leq D_{\max}} \left\{ \mathcal{R}_{emp}(\widehat{\tau}^{D}) + \mathrm{pen}(\widehat{\tau}^{D}) \right\} \enspace .$$

3. Final segmentation:

$$\widehat{\tau} := \widehat{\tau}^{\widehat{D}}.$$

$\mathbb{III}$ Dynamic prog. and reproducing kernels

Find the best segmentation with D segments

Dynamic programming (Step 1 in KCP)

- Solving $\widehat{\tau}^D \in \operatorname{Argmin}_{\tau \in \mathcal{T}_n^D} \{\mathcal{R}_{emp}(\tau)\}$: computationally hard ($1 \leq D \leq D_{\max}$)
- General principle:

$$L(D+1;t) = \min_{1 \leq s \leq t-1} \{L(D;s) + C(s,t)\}$$

  - $L(D;s)$: cost of the best segmentation of $[1, s+1[$ with D segments
  - $C(s,t)$: cost of segment $[s, t+1[$
- Outputs the exact solution

Classical computational complexity

- Time complexity: $O(n^2)$ (if evaluating $C(s,t)$ is linear time)
- Space complexity: $O(n^2)$ (storing the cost matrix)

Find the best segmentation with D segments

Dynamic programming (Step 1 in KCP)

- Solving $\widehat{\tau}^D \in \operatorname{Argmin}_{\tau \in \mathcal{T}_n^D} \{\mathcal{R}_{emp}(\tau)\}$: computationally hard $(1 \leq D \leq D_{\max})$
- General principle:

$$L(D+1; t) = \min_{1 \leq s \leq t-1} \{L(D; s) + C(s, t)\}$$

  - $L(D; s)$: cost of the best segmentation of $[1, s+1[$ with D segments
  - $C(s, t)$: cost of segment $[s, t+1[$

- Outputs the exact solution

Classical computational complexity

- Time complexity: $O(n^2)$ (if evaluating $C(s, t)$ is linear time)
- Space complexity: $O(n^2)$ (storing the cost matrix)

Find the best segmentation with D segments

Dynamic programming (Step 1 in KCP)

- Solving $\widehat{\tau}^D \in \operatorname{Argmin}_{\tau \in \mathcal{T}_n^D} \{\mathcal{R}_{emp}(\tau)\}$: computationally hard $(1 \leq D \leq D_{\max})$
- General principle:

$$L(D+1; t) = \min_{1 \leq s \leq t-1} \{L(D; s) + C(s, t)\}$$

  - $L(D; s)$: cost of the best segmentation of $[1, s+1[$ with D segments
  - $C(s, t)$: cost of segment $[s, t+1[$

- Outputs the exact solution

Classical computational complexity

- Time complexity: $O(n^2)$ (if evaluating $C(s, t)$ is linear time)
- Space complexity: $O(n^2)$ (storing the cost matrix)

Find the best segmentation with D segments

Dynamic programming (Step 1 in KCP)

- Solving $\widehat{\tau}^D \in \mathrm{Argmin}_{\tau \in \mathcal{T}_n^D} \{\mathcal{R}_{emp}(\tau)\}$: computationally hard $(1 \leq D \leq D_{\max})$
- General principle:

$$L(D+1;t) = \min_{1 \leq s \leq t-1} \{L(D;s) + C(s,t)\}$$

  - $L(D;s)$: cost of the best segmentation of $[1, s+1[$ with D segments
  - $C(s,t)$: cost of segment $[s, t+1[$
- Outputs the exact solution

Classical computational complexity

- Time complexity: $O(n^2)$ (if evaluating $C(s,t)$ is linear time)
- Space complexity: $O(n^2)$ (storing the cost matrix)

Find the best segmentation with D segments

Dynamic programming (Step 1 in KCP)

- Solving $\widehat{\tau}^D \in \operatorname{Argmin}_{\tau \in \mathcal{T}_n^D} \{\mathcal{R}_{emp}(\tau)\}$: computationally hard ($1 \leq D \leq D_{\max}$)
- General principle:

$$L(D+1;t) = \min_{1 \leq s \leq t-1} \{L(D;s) + C(s,t)\}$$

  - $L(D;s)$: cost of the best segmentation of $[1, s+1[$ with D segments
  - $C(s,t)$: cost of segment $[s, t+1[$
- Outputs the exact solution

Classical computational complexity

- Time complexity: $O(n^2)$ (if evaluating $C(s,t)$ is linear time)
- Space complexity: $O(n^2)$ (storing the cost matrix)

Embedding dynamic programming in the kernel framework

Main limitations of the naive formulation

$$C\left(\tau_\ell, \tau_{\ell+1}\right) = \frac{1}{\tau_{\ell+1} - \tau_\ell} \sum_{i=\tau_\ell}^{\tau_{\ell+1}-1} \sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)$$

- Computing $C\left(\tau_\ell, \tau_{\ell+1}\right)$ is quadratic
- A naive formulation of the dyn. prog. is $O(n^4)$ in time and $O(n^2)$ in space (if we store the cost matrix $\{C(i,j)\}_{1 \leq i,j \leq n+1}$).

Improved formulation

- At round t, only store $C(\cdot, t) \in \mathbb{R}^n$.
- Update $C(\cdot, t+1)$ from $C(\cdot, t)$ on the fly.

$\longrightarrow$ Reduced time and space complexity to $O(n^2)$ and $O(n)$ resp.

Embedding dynamic programming in the kernel framework

Main limitations of the naive formulation

$$C\left(\tau_\ell, \tau_{\ell+1}\right) = \frac{1}{\tau_{\ell+1} - \tau_\ell} \sum_{i=\tau_\ell}^{\tau_{\ell+1}-1} \sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)$$

- Computing $C\left(\tau_\ell, \tau_{\ell+1}\right)$ is quadratic
- A naive formulation of the dyn. prog. is $O(n^4)$ in time and $O(n^2)$ in space (if we store the cost matrix $\{C(i,j)\}_{1 \leq i,j \leq n+1}$).

Improved formulation

- At round t, only store $C(\cdot, t) \in \mathbb{R}^n$.
- Update $C(\cdot, t+1)$ from $C(\cdot, t)$ on the fly.

$\longrightarrow$ Reduced time and space complexity to $O(n^2)$ and $O(n)$ resp.

Embedding dynamic programming in the kernel framework

Main limitations of the naive formulation

$$C\left(\tau_\ell, \tau_{\ell+1}\right) = \frac{1}{\tau_{\ell+1} - \tau_\ell} \sum_{i=\tau_\ell}^{\tau_{\ell+1}-1} \sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)$$

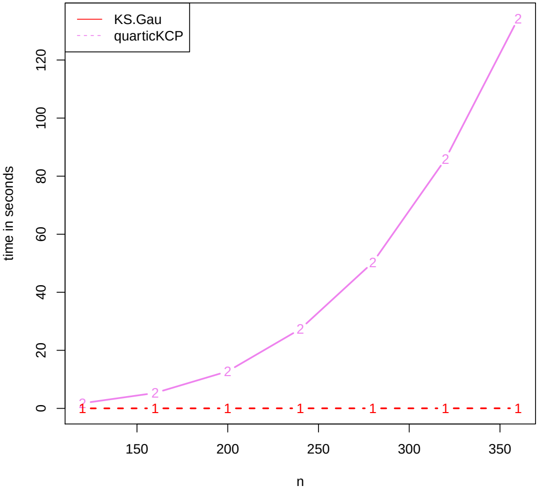
- Computing $C\left(\tau_\ell, \tau_{\ell+1}\right)$ is quadratic
- A naive formulation of the dyn. prog. is $O(n^4)$ in time and $O(n^2)$ in space (if we store the cost matrix $\{C(i,j)\}_{1\leq i,j\leq n+1}$).

Improved formulation

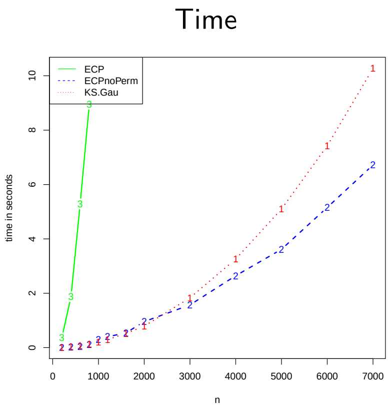
- At round t, only store $C(\cdot, t) \in \mathbb{R}^n$.
- Update $C(\cdot, t+1)$ from $C(\cdot, t)$ on the fly.

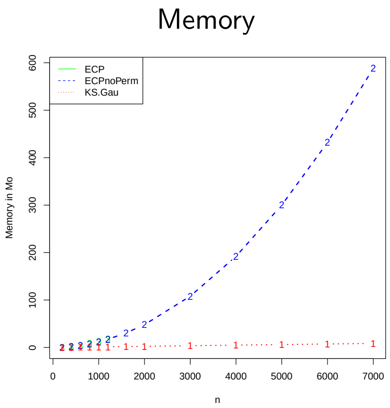
$\longrightarrow$ Reduced time and space complexity to $O(n^2)$ and $O(n)$ resp.

Runtime of the improved dyn. prog.: Kernseg



Comparison Kernseg - ECP ($D_{max} = 100$)





Rks:

- ECP: based on a kernel and Binary Segmentation
- Chooses among candidate change-points using permutations

IIII Faster approx. optimization algo.

From quadratic-to linear-time cost

Computational limitation

$$C\left(\tau_\ell, \tau_{\ell+1}\right) = \frac{1}{\tau_{\ell+1} - \tau_\ell} \sum_{i=\tau_\ell}^{\tau_{\ell+1}-1} \sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)$$

- With general kernel, the cost of $[\tau_\ell, \tau_{\ell+1}]$ is quadratic
  $\longrightarrow \quad n \approx 5.10^4$ in less than 2 minutes
- $n \geq 10^6$ not realistic with a $O(n^2)$ time complexity
- If $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathbb{R}^p}$, then

$$\sum_{1 \leq i,j \leq s} k(x_i, x_j) = \left\langle \sum_{i=1}^{s} \phi(x_i), \sum_{j=1}^{s} \phi(x_j) \right\rangle_{\mathbb{R}^p} = \left\| \sum_{i=1}^{s} \phi(x_i) \right\|_{\mathbb{R}^p}^2$$

From quadratic-to linear-time cost

Computational limitation

$$C\left(\tau_\ell, \tau_{\ell+1}\right) = \frac{1}{\tau_{\ell+1} - \tau_\ell} \sum_{i=\tau_\ell}^{\tau_{\ell+1}-1} \sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)$$

- With general kernel, the cost of $[\tau_\ell, \tau_{\ell+1}]$ is quadratic
  $\longrightarrow \quad n \approx 5.10^4$ in less than 2 minutes
- $n \geq 10^6$ not realistic with a $O(n^2)$ time complexity
- If $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathbb{R}^p}$, then

$$\sum_{1 \leq i,j \leq s} k(x_i, x_j) = \left\langle \sum_{i=1}^{s} \phi(x_i), \sum_{j=1}^{s} \phi(x_j) \right\rangle_{\mathbb{R}^p} = \left\| \sum_{i=1}^{s} \phi(x_i) \right\|_{\mathbb{R}^p}^2$$

Low-rank matrix approximation (Drineas Mahoney (05))

- Gram matrix: $K = \{k(X_i, X_j)\}_{1 \leq i,j \leq n}$
- $I, J \subset \{1, \ldots, n\}$ with $I = \{1, \ldots, n\}$ and $\mathrm{Card}(J) = p$.
- $K^{-}_{J,J}$: pseudo-inverse of $K_{J,J}$

Nyström approximation of size p

$$K \approx \widetilde{K} = K_{I,J} \times K^{-}_{J,J} \times K_{J,I} .$$

Then

$$\widetilde{K} = Z^{\top} Z, \quad \text{where } Z \in \mathcal{M}_{p,n}(\mathbb{R}),$$

which means that

$$\widetilde{k}(x_i, x_j) = Z_i^{\top} Z_j, \quad \text{with } Z_i, Z_j \in \mathbb{R}^p .$$

Approximate optimization algorithm

Fast (approximate) procedure (ApKern)

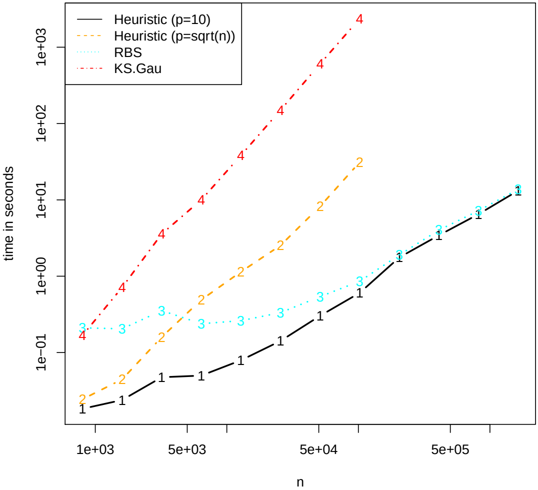
1. Approximate $K \approx \widetilde{K} = Z^\top \cdot Z$
2. Apply Binary Segmentation to the Z_is and output a sequence of approximate solutions:

$$\left\{ \widehat{\tau}_{BS}^{D} \right\}_{1 \leq D \leq D_{\max}} .$$

Computational complexity

- Time: $O(n \log(n) \vee np^2)$
- Space: $O(n)$
- Allows for $n \geq 10^6$

Comparison ApKern - RBS ($D_{max} = 100$)



$\mathbb{IV}$ How many chgpts? $\rightarrow$ designing a penalty

Model selection-based procedure: KCP

Outline

1. For every $1 \leq D \leq D_{\max}$,

$$\widehat{\tau}^D \in \mathrm{Argmin}_{\tau \in \mathcal{T}_n^D} \mathcal{R}_{emp}(\tau) \ ,$$

2. Define

$$\widehat{D} = \mathrm{Argmin}_{1 \leq D \leq D_{\max}} \left\{ \mathcal{R}_{emp}(\widehat{\tau}^D) + \mathrm{pen}(\widehat{\tau}^D) \right\} \ .$$

$\longrightarrow$ Model selection result

3. Final segmentation:

$$\widehat{\tau} := \widehat{\tau}^{\widehat{D}}.$$

Penalty shape and minimal length

- Arlot, C., Harchaoui (18) proved an oracle inequality for

$$\text{pen}(\tau) = c_1 \frac{D_\tau}{n} + c_2 \frac{1}{n} \log \underbrace{\binom{n-1}{D_\tau - 1}}_{=\text{Card}\left(\mathcal{T}_n^{D_\tau}\right)}$$

- $c_1, c_2 > 0$: estimated by slope heuristic
- With a constraint on the minimal length ℓ of any segment:

$$\text{pen}_\ell(\tau) = c_1 \frac{D_\tau}{n} + c_2 \frac{1}{n} \log \binom{n - D_\tau(\ell - 1) - 1}{D_\tau - 1}$$

$\longrightarrow$ Particularly relevant with low signal-to-noise ratio data.

Penalty shape and minimal length

- Arlot, C., Harchaoui (18) proved an oracle inequality for

$$\mathrm{pen}(\tau) = c_1 \frac{D_\tau}{n} + c_2 \frac{1}{n} \log \underbrace{\binom{n-1}{D_\tau - 1}}_{=\mathrm{Card}\left(\mathcal{T}_n^{D_\tau}\right)}$$

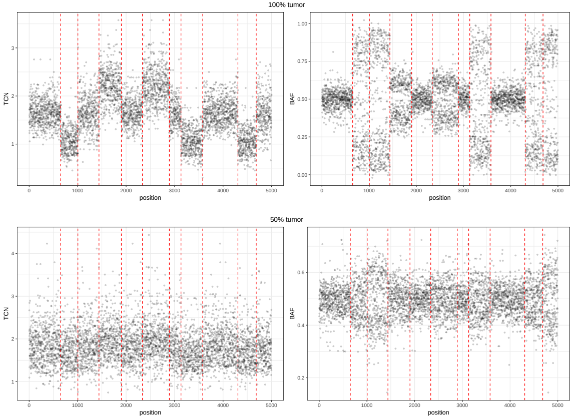
- $c_1, c_2 > 0$: estimated by slope heuristic
- With a constraint on the minimal length ℓ of any segment:

$$\mathrm{pen}_\ell(\tau) = c_1 \frac{D_\tau}{n} + c_2 \frac{1}{n} \log \binom{n - D_\tau(\ell - 1) - 1}{D_\tau - 1}$$

$\longrightarrow$ Particularly relevant with low signal-to-noise ratio data.

$\mathbb{V}$ Biological experiments: LOH

Data from ACNR package (Pierre-Jean, Neuvial (14))



- $n = 5\,000$, $D^{\star} = 11$, purity is 100% and 50%
- $k(x, y) = k_{TCN}(x_1, y_1) + k_{BAF}(x_2, y_2)$

Parameters values and accuracy

Tuning the kernel parameters

- k_{TCN} and k_{BAF}: Gaussian kernels $e^{-\frac{(x-y)^2}{h}}$
- Bandwidth:

$$h = 2\left(\frac{\widehat{\sigma}}{\sqrt{2}}\right)^2, \qquad \text{with} \qquad \widehat{\sigma}^2 = \frac{1}{n/2}\sum_{i=1}^{n/2}\left(X_{2i} - X_{2i-1}\right)^2$$

Accuracy measure of the segmentation

- Segmentation τ as a matrix $M^\tau = \left\{M^\tau_{i,j}\right\}$ such that

$$M^\tau_{i,j} = \sum_{d=1}^{D^\tau} \frac{\mathbb{1}_{\{\tau_d \leq i,j < \tau_{d+1}\}}}{\tau_{d+1} - \tau_d}$$

- $\|M\|_F = \sqrt{\mathrm{tr}\left(M^\top M\right)}$
-

$$\mathrm{Accuracy}(\tau) = \left\| M^\tau - M^{\tau^\star} \right\|_F$$

Parameters values and accuracy

Tuning the kernel parameters

- k_{TCN} and k_{BAF}: Gaussian kernels $e^{-\frac{(x-y)^2}{h}}$
- Bandwidth:

$$h = 2\left(\frac{\widehat{\sigma}}{\sqrt{2}}\right)^2, \qquad \text{with} \qquad \widehat{\sigma}^2 = \frac{1}{n/2}\sum_{i=1}^{n/2}(X_{2i} - X_{2i-1})^2$$

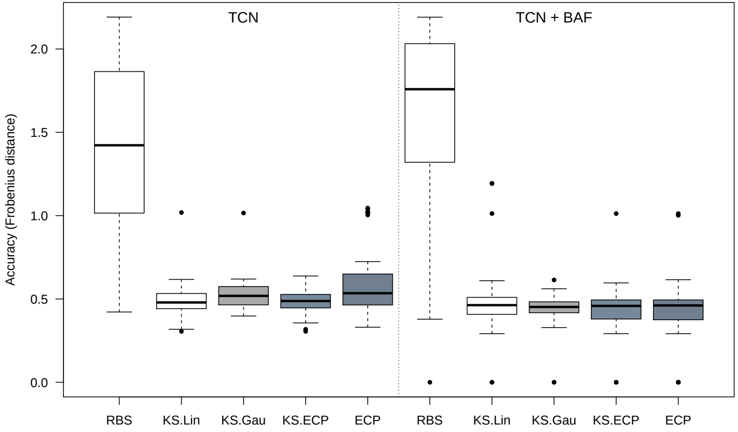
Accuracy measure of the segmentation

- Segmentation τ as a matrix $M^\tau = \left\{M^\tau_{i,j}\right\}$ such that

$$M^\tau_{i,j} = \sum_{d=1}^{D^\tau} \frac{\mathbb{1}_{\{\tau_d \leq i,j < \tau_{d+1}\}}}{\tau_{d+1} - \tau_d}$$

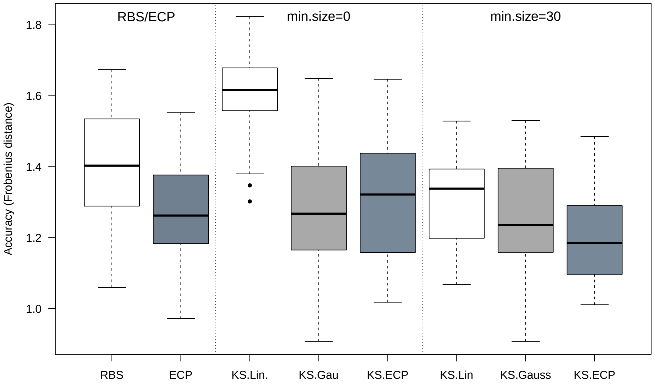
- $\|M\|_F = \sqrt{\operatorname{tr}\left(M^\top M\right)}$
-
$$\operatorname{Accuracy}(\tau) = \left\| M^\tau - M^{\tau^\star} \right\|_F$$

Easy case: high purity



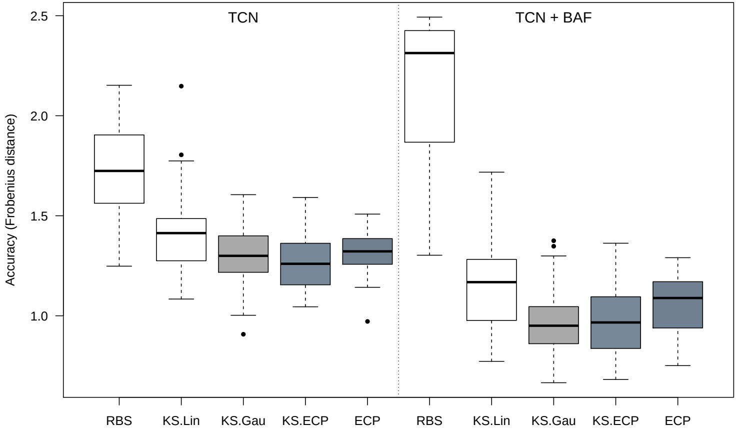
- Accuracy (chpts location) expressed with Frobenius norm
- Everyone works well at $\widehat{D}$ (except RBS)
- (TCN,BAF) not significantly better! (easy case)

Difficult case (low purity): Minimum length



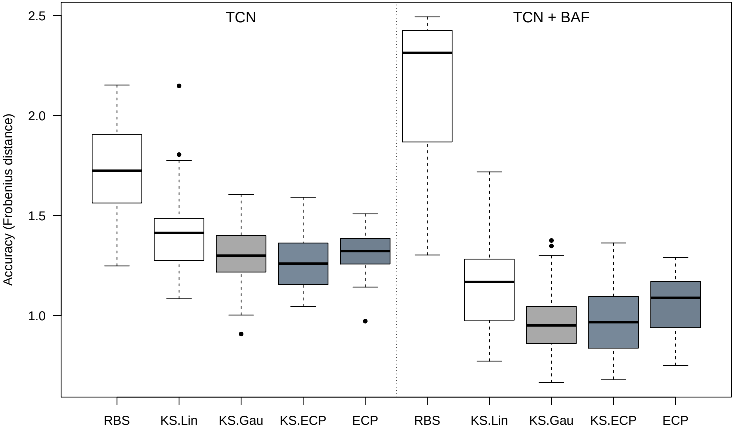
- ECP: $\ell = 30$ constraint encoded by default
- Global improvement with $\ell = 30$ (especially for linear kernel)
- Gaussian and ECP kernels: best performance

Influence of the kernel – joint signal



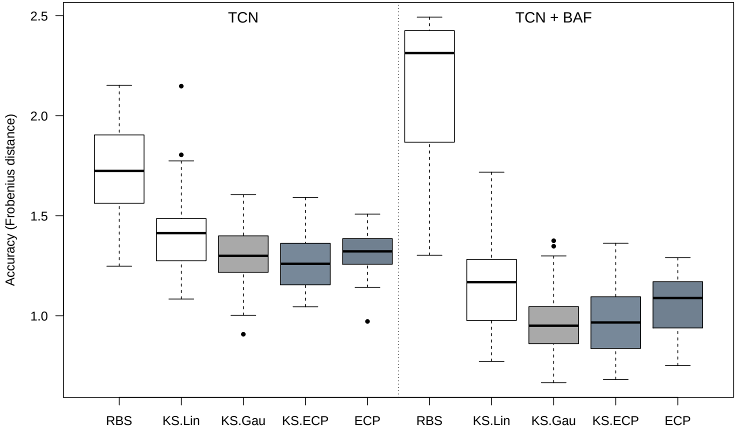
- RBS fails: bad choice of $\widehat{D}$
- Best perf.: Gaussian and ECP kernels (characteristic)
- Strong improvement with (TCN, BAF) (difficult case)

Influence of the kernel – joint signal



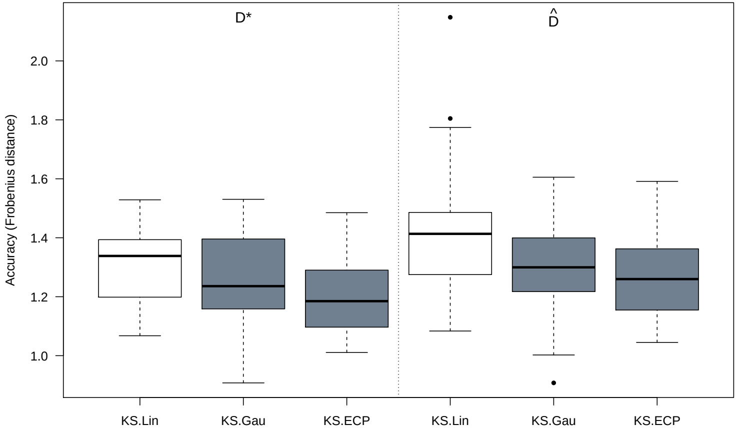
- RBS fails: bad choice of $\widehat{D}$
- Best perf.: Gaussian and ECP kernels (characteristic)
- Strong improvement with (TCN, BAF) (difficult case)

Influence of the kernel – joint signal



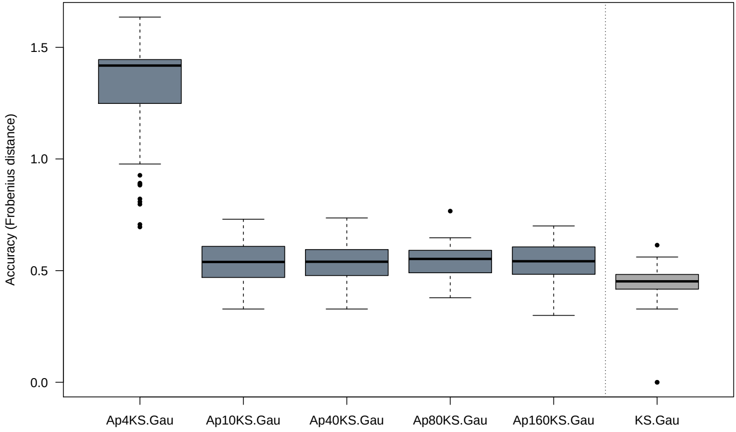
- RBS fails: bad choice of $\widehat{D}$
- Best perf.: Gaussian and ECP kernels (characteristic)
- Strong improvement with (TCN, BAF) (difficult case)

Estimated the number of segments quality



- Slight loss of accuracy when estimating
- Does not modify the ranking of kernel-based procedures

Quality of the approximate procedure (ApKern)



- p has to be large enough ($p \geq \mathrm{rk}(K)$?)
- Close to the optimal solution
- Quite stable results w.r.t. p

Pros/cons of ApKern

Assets

- Dealing with $n \geq 10^6$ in a few seconds is possible!
- Enjoys a good accuracy
- Reducing the dimension (low-rank approx.) allows for using any available discrete optimization algorithm (dynamic programming, BS, ...)

Drawbacks

- Choosing p (and the X_is) remains an open practical question
- No ongoing model selection result to choose the number fo segments (Binary Segmentation)

Pros/cons of ApKern

Assets

- Dealing with $n \geq 10^6$ in a few seconds is possible!
- Enjoys a good accuracy
- Reducing the dimension (low-rank approx.) allows for using any available discrete optimization algorithm (dynamic programming, BS, . . .)

Drawbacks

- Choosing p (and the X_is) remains an open practical question
- No ongoing model selection result to choose the number fo segments (Binary Segmentation)

Take-home message/discussion

Recap

- KCP: no (strong) distributional assumption and theoretical guarantees
- Kernseg: R package with an improved complexity
  - Time: $O(n^2)$ (exact) or $O(n \log n)$ (approx.)
  - Space: $O(n)$
- Achieves state-of-the-art performances on biological data generated with the ACNR package (Characteristic kernels)

Open questions

- Explore other structured objects (dynamic networks, ...)
- Optimize the kernel (approx.?): supervised or not
- ...

Thank you!

Take-home message/discussion

Recap

- KCP: no (strong) distributional assumption and theoretical guarantees
- Kernseg: R package with an improved complexity
  - Time: $O(n^2)$ (exact) or $O(n \log n)$ (approx.)
  - Space: $O(n)$
- Achieves state-of-the-art performances on biological data generated with the ACNR package (Characteristic kernels)

Open questions

- Explore other structured objects (dynamic networks, ...)
- Optimize the kernel (approx.?): supervised or not
- ...

Thank you!

Take-home message/discussion

Recap

- KCP: no (strong) distributional assumption and theoretical guarantees
- Kernseg: R package with an improved complexity
  - Time: $O(n^2)$ (exact) or $O(n \log n)$ (approx.)
  - Space: $O(n)$
- Achieves state-of-the-art performances on biological data generated with the ACNR package (Characteristic kernels)

Open questions

- Explore other structured objects (dynamic networks, . . .)
- Optimize the kernel (approx.?): supervised or not
- . . .

Thank you!