Electronic Journal of Statistics
Vol. 11 (2017) 1570–1599
ISSN: 1935-7524
DOI: 10.1214/17-EJS1270

Efficient block boundaries estimation in block-wise constant matrices: An application to HiC data[*]

Vincent Brault

Univ. Grenoble Alpes, LJK, F-38000 Grenoble, France
CNRS, LJK, F-38000 Grenoble, France
e-mail: vincent.brault@imag.fr

Julien Chiquet and Céline Lévy-Leduc

UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay
e-mail: julien.chiquet@inra.fr*;* celine.levy-leduc@agroparistech.fr

Abstract: In this paper, we propose a novel modeling and a new methodology for estimating the location of block boundaries in a random matrix consisting of a block-wise constant matrix corrupted with white noise. Our method consists in rewriting this problem as a variable selection issue. A penalized least-squares criterion with an ℓ_1-type penalty is used for dealing with this problem. Firstly, some theoretical results ensuring the consistency of our block boundaries estimators are provided. Secondly, we explain how to implement our approach in a very efficient way. This implementation is available in the R package blockseg which can be found in the Comprehensive R Archive Network. Thirdly, we provide some numerical experiments to illustrate the statistical and numerical performance of our package, as well as a thorough comparison with existing methods. Fourthly, an empirical procedure is proposed for estimating the number of blocks. Finally, our approach is applied to HiC data which are used in molecular biology for better understanding the influence of the chromosomal conformation on the cells functioning.

MSC 2010 subject classifications: Primary 62-07, 62F30, 62P10; secondary 62F12, 62J07.
Keywords and phrases: Change-points, high-dimensional sparse linear model, HiC experiments.

Received September 2016.

Contents

[*] ABS4NGS project (ANR-11-BINF-0001-06) for support.

1. Introduction

Detecting automatically the block boundaries in large block wise constant matrices corrupted with noise is a very important issue which may have several applications. One of the main situations in which this problem occurs is in the study of HiC data. It corresponds to one of the most recent chromosome conformation capture technologies that have been developed to better understand the influence of the chromosomal conformation on the cells functioning. This technology is based on a deep sequencing approach and provides read pairs corresponding to pairs of genomic loci that physically interacts in the nucleus, see [13] for more details. The raw measurements provided by HiC data are often summarized as a square matrix where each entry at row i and column j stands for the total number of read pairs matching in position i and position j, respectively, see [5] for further details. Positions refer here to a sequence of non-overlapping windows of equal sizes covering the genome.

Blocks of different intensities arise among this matrix, revealing interacting genomic regions among which some have already been confirmed to host co-regulated genes. The purpose of the statistical analysis is then to provide a fully automated and efficient strategy to determine a decomposition of the matrix in non-overlapping blocks, which gives, as a by-product, a list of non-overlapping interacting chromosomic regions. In the following, our goal will thus be to design an efficient and fully automated method to find the block boundaries of non-overlapping blocks in very large matrices which can be modeled as block wise constant matrices corrupted with white noise. As a natural extension to the two-dimensional case, the positions in columns and rows of the block boundaries within the observation matrix will also be called change-points. For a more precise definition, we refer the reader to the beginning of Section 2.

An abundant literature is dedicated to the change-point detection issue for one-dimensional data both from a theoretical and practical point of view. From a practical point of view, the standard approach for estimating the change-point locations is based on least- square fitting, performed via a dynamic programming algorithm (DP). Indeed, for a given number of change-points K, the dynamic programming algorithm, proposed by [3] and [7], takes advantage of the intrinsic additive nature of the least-square objective to recursively compute the optimal change-points locations with a complexity of $O(Kn^2)$ in time, see [1] and [11]. This complexity has recently been improved by [20] and [15] in some specific cases. Another very popular approach in the one-dimensional case is the Binary Segmentation method proposed by [22]. However, contrary to the DP approach, it does not necessary provide the optimal solution of the least-square minimization problem.

However, in general one-dimensional situations, the computational burden of the DP based methods is prohibitive to handle very large data sets. In this situation, [9] proposed to rephrase the change-point estimation issue as a variable selection problem. This approach has also been extended by [24] to find shared change-points between several signals. In the two-dimensional case, namely when matrices have to be processed, no method has been proposed, to the best of our knowledge, for providing the block boundaries of non overlapping blocks of very large $n \times n$ matrices. In order to be able to process observation matrices coming from HiC experiments, we aim at being able to handle 5000×5000 matrices, which corresponds to matrices having 2.5×10^7 entries. The only statistical approach proposed for retrieving such non-overlapping block boundaries in this two-dimensional framework is the one devised by [12] but it is limited to the case where the block wise matrix is assumed to be block wise constant on the diagonal and constant outside the diagonal blocks.

The difficulties that we have to face with in the two-dimensional framework are the following. Firstly, it has to be noticed that the classical dynamic programming algorithm cannot be applied in such a framework since the Markov property does not hold anymore. Secondly, the group-lars approach of [24] cannot be used in this framework since it would only provide change-points in columns and not in rows. Thirdly, although very efficient for image denoising, neither the generalized Lasso approach devised by [23] nor the fused Lasso signal approximator of [10], which are implemented in the R packages `genlasso` and `flsa`, respectively, give access to the boundaries of non-overlapping blocks of a noisy block wise constant matrix. This fact is illustrated in Figure 2. The first column of this figure contains the block wise constant matrix of Figure 1 corrupted with additional noise in high signal to noise ratio contexts. The denoising of these noisy matrices obtained by the packages `genlasso` and `flsa` is displayed in the second and third columns of Figure 1, respectively. Note that, for obtaining these results, we used the default parameters of these packages and for the parameter λ we used the one giving the denoised matrix being the closest to the original one in terms of recovered blocks.

In this paper, our goal is thus to design a statistical method for estimating the location of the boundaries of non-overlapping blocks from a block wise constant matrix corrupted with white noise. To the best of our knowledge, there is indeed no statistical procedure for answering this specific question in the literature that is both computationally and statistically efficient.

The paper is organized as follows. In Section 2, we first describe how to rephrase the problem of two-dimensional change-point estimation as a high di-

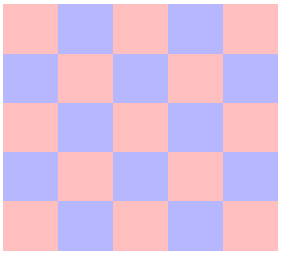

Fig 1. *Block wise constant matrix without noise.*

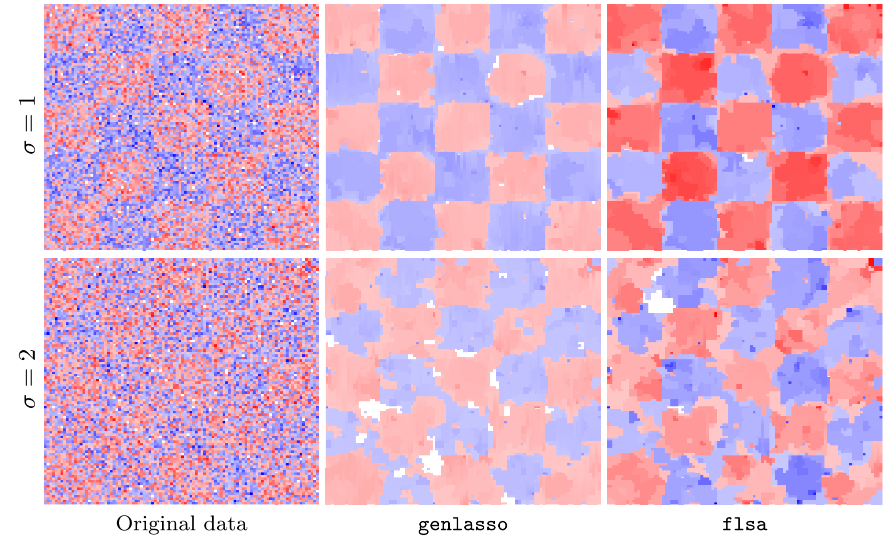

Fig 2. *Left: Matrix of Figure 1 corrupted with Gaussian white noise of variance σ. Middle: Denoising obtained with* `genlasso`*. Right: Denoising obtained with* `flsa`*.*

mensional sparse linear model and give some theoretical results which prove the consistency of our change-point estimators. In Section 3, we describe how to efficiently implement our method. Then, we provide in Section 4 experimental evidence of the relevance of our approach on synthetic data. We conclude in Section 6 by a thorough analysis of a HiC dataset.

2. Statistical framework

2.1. Statistical modeling

In this section, we explain how the two-dimensional retrospective change-point estimation issue can be seen as a variable selection problem. Our goal is to estimate $\mathbf{t}_1^\star = (t_{1,1}^\star, \dots, t_{1,K_1^\star}^\star)$ and $\mathbf{t}_2^\star = (t_{2,1}^\star, \dots, t_{2,K_2^\star}^\star)$ from the random matrix

$\mathbf{Y} = (Y_{i,j})_{1 \leq i,j \leq n}$ defined by

$$\mathbf{Y} = \mathbf{U} + \mathbf{E}, \tag{2.1}$$

where $\mathbf{U} = (U_{i,j})$ is a blockwise constant matrix such that

$$U_{i,j} = \mu^{\star}_{k,\ell} \quad \text{if } t^{\star}_{1,k-1} \leq i \leq t^{\star}_{1,k} - 1 \text{ and } t^{\star}_{2,\ell-1} \leq j \leq t^{\star}_{2,\ell} - 1,$$

with the convention $t^{\star}_{1,0} = t^{\star}_{2,0} = 1$ and $t^{\star}_{1,K^{\star}_1+1} = t^{\star}_{2,K^{\star}_2+1} = n + 1$. An example of such a matrix $\mathbf{U}$ is displayed in Figure 3. The entries $E_{i,j}$ of the matrix $\mathbf{E} = (E_{i,j})_{1 \leq i,j \leq n}$ are iid zero-mean random variables. With such a definition the $Y_{i,j}$ are assumed to be independent random variables with a blockwise constant mean.

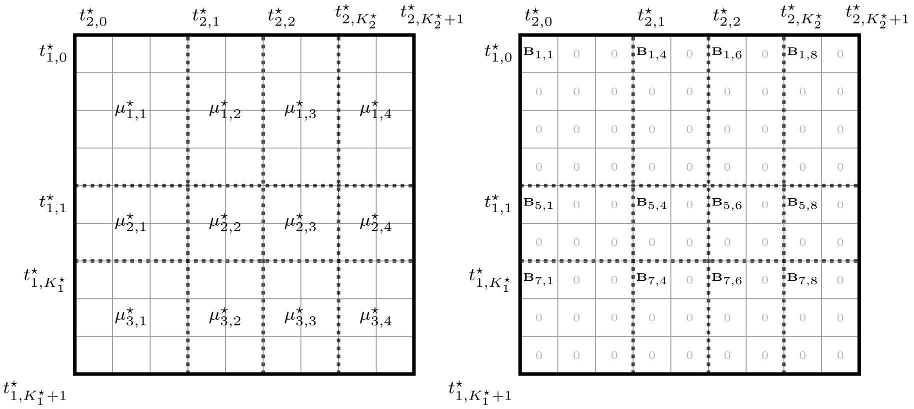

FIG 3. *Left: An example of a matrix* $\mathbf{U}$ *with* $n = 9$, $K^{\star}_1 = 2$ *and* $K^{\star}_2 = 3$. *Right: The matrix* $\mathbf{B}$ *associated to this matrix* $\mathbf{U}$.

Let $\mathbf{T}$ be a $n \times n$ lower triangular matrix with nonzero elements equal to one and $\mathbf{B}$ a sparse matrix containing null entries except for the $\mathbf{B}_{i,j}$ such that $(i,j) \in \{t^{\star}_{1,0}, \ldots, t^{\star}_{1,K^{\star}_1}\} \times \{t^{\star}_{2,0}, \ldots, t^{\star}_{2,K^{\star}_2}\}$. Then, (2.1) can be rewritten as follows:

$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{T}^{\top} + \mathbf{E}, \tag{2.2}$$

where $\mathbf{T}^{\top}$ denotes the transpose of the matrix $\mathbf{T}$. For an example of a matrix $\mathbf{B}$, see Figure 3. Let $\mathrm{Vec}(\mathbf{X})$ denotes the vectorization of the matrix $\mathbf{X}$ formed by stacking the columns of $\mathbf{X}$ into a single column vector then $\mathrm{Vec}(\mathbf{Y}) = \mathrm{Vec}(\mathbf{T}\mathbf{B}\mathbf{T}^{\top}) + \mathrm{Vec}(\mathbf{E})$. Hence, by using that $\mathrm{Vec}(\mathbf{A}\mathbf{X}\mathbf{C}) = (\mathbf{C}^{\top} \otimes \mathbf{A})\mathrm{Vec}(\mathbf{X})$, where $\otimes$ denotes the Kronecker product, (2.2) can be rewritten as:

$$\mathcal{Y} = \mathcal{X}\mathcal{B} + \mathcal{E}, \tag{2.3}$$

where $\mathcal{Y} = \mathrm{Vec}(\mathbf{Y})$, $\mathcal{X} = \mathbf{T} \otimes \mathbf{T}$, $\mathcal{B} = \mathrm{Vec}(\mathbf{B})$ and $\mathcal{E} = \mathrm{Vec}(\mathbf{E})$. Thanks to these transformations, Model (2.1) has thus been rephrased as a sparse high dimensional linear model where $\mathcal{Y}$ and $\mathcal{E}$ are $n^2 \times 1$ column vectors, $\mathcal{X}$ is a

$n^2 \times n^2$ matrix and $\mathcal{B}$ is $n^2 \times 1$ sparse column vectors. Multiple change-point estimation Problem (2.1) can thus be addressed as a variable selection problem:

$$\widehat{\mathcal{B}}(\lambda_n) = \operatorname*{Argmin}_{\mathcal{B}\in\mathbb{R}^{n^2}} \left\{ \|\mathcal{Y} - \mathcal{X}\mathcal{B}\|_2^2 + \lambda_n \|\mathcal{B}\|_1 \right\}, \tag{2.4}$$

where $\|u\|_2^2$ and $\|u\|_1$ are defined for a vector u in $\mathbb{R}^N$ by $\|u\|_2^2 = \sum_{i=1}^N u_i^2$ and $\|u\|_1 = \sum_{i=1}^N |u_i|$. Criterion (2.4) is related to the popular Least Absolute Shrinkage and Selection Operator (LASSO) in least-square regression. Thanks to the sparsity enforcing property of the ℓ_1-norm, the estimator $\widehat{\mathcal{B}}$ of $\mathcal{B}$ is expected to be sparse and to have non-zero elements matching with those of $\mathcal{B}$. Hence, retrieving the positions of the non zero elements of $\widehat{\mathcal{B}}$ thus provides estimators of $(t_{1,k}^\star)_{1\leq k\leq K_1^\star}$ and of $(t_{2,k}^\star)_{1\leq k\leq K_2^\star}$. More precisely, let us define by $\widehat{\mathcal{A}}(\lambda_n)$ the set of active variables:

$$\widehat{\mathcal{A}}(\lambda_n) = \left\{ j \in \{1,\ldots,n^2\} : \widehat{\mathcal{B}}_j(\lambda_n) \neq 0 \right\}.$$

For each j in $\widehat{\mathcal{A}}(\lambda_n)$, consider the Euclidean division of $(j-1)$ by n, namely $(j-1) = nq_j + r_j$ then

$$\begin{aligned} \widehat{\mathbf{t}}_1 = (\widehat{t}_{1,k})_{1\leq k\leq |\widehat{\mathcal{A}}_1(\lambda_n)|} &\in \{r_j + 1 : j \in \widehat{\mathcal{A}}(\lambda_n)\}, \\ \widehat{\mathbf{t}}_2 = (\widehat{t}_{2,\ell})_{1\leq \ell\leq |\widehat{\mathcal{A}}_2(\lambda_n)|} &\in \{q_j + 1 : j \in \widehat{\mathcal{A}}(\lambda_n)\} \\ \text{where } \widehat{t}_{1,1} < \widehat{t}_{1,2} < \cdots < \widehat{t}_{1,|\widehat{\mathcal{A}}_1(\lambda_n)|}, &\quad \widehat{t}_{2,1} < \widehat{t}_{2,2} < \cdots < \widehat{t}_{2,|\widehat{\mathcal{A}}_2(\lambda_n)|}. \end{aligned} \tag{2.5}$$

In (2.5), $|\widehat{\mathcal{A}}_1(\lambda_n)|$ and $|\widehat{\mathcal{A}}_2(\lambda_n)|$ correspond to the number of distinct elements in $\{r_j : j \in \widehat{\mathcal{A}}(\lambda_n)\}$ and $\{q_j : j \in \widehat{\mathcal{A}}(\lambda_n)\}$, respectively.

As far as we know, neither thorough practical implementation nor theoretical grounding have been given so far to support such an approach for change-point estimation in the two-dimensional case. In the following section, we give theoretical results supporting the use of such an approach.

2.2. Theoretical results

In order to establish the consistency of the estimators $\widehat{\mathbf{t}}_1$ and $\widehat{\mathbf{t}}_2$ defined in (2.5), we shall use assumptions (**A1**–**A4**). These assumptions involve the two following quantities

$$\begin{aligned} I_{\min}^\star &= \min_{0\leq k\leq K_1^\star} |t_{1,k+1}^\star - t_{1,k}^\star| \wedge \min_{0\leq k\leq K_2^\star} |t_{2,k+1}^\star - t_{2,k}^\star|, \\ J_{\min}^\star &= \min_{1\leq k\leq K_1^\star, 1\leq \ell\leq K_2^\star+1} |\mu_{k+1,\ell}^\star - \mu_{k,\ell}^\star| \wedge \min_{1\leq k\leq K_1^\star+1, 1\leq \ell\leq K_2^\star} |\mu_{k,\ell+1}^\star - \mu_{k,\ell}^\star|, \end{aligned}$$

which corresponds to the smallest length between two consecutive change-points and to the smallest jump size between two consecutive blocks, respectively.

(**A1**) The random variables $(E_{i,j})_{1\leq i,j\leq n}$ are iid zero mean random variables such that there exists a positive constant β such that for all ν in $\mathbb{R}$, $\mathbb{E}[\exp(\nu E_{1,1})] \leq \exp(\beta\nu^2)$.

(**A2**) The sequence (δ_n) is a non increasing and positive sequence tending to zero such that $n\delta_n {J^\star_{\min}}^2/\log(n) \to \infty$, as n tends to infinity.

(**A3**) The sequence (λ_n) appearing in (2.4) is such that $(n\delta_n J^\star_{\min})^{-1}\lambda_n \to 0$, as n tends to infinity.

(**A4**) $I^\star_{\min} \geq n\delta_n$.

The following proposition ensures that the distance between each estimated and true change-point is less than $n\delta_n$, where δ_n is a non increasing and positive sequence tending to zero defined in (A2), (A3) and (A4), with a probability tending to 1, as $n \to \infty$.

Proposition 1. *Let $(Y_{i,j})_{1\leq i,j\leq n}$ be defined by (2.1) and $\widehat{t}_{1,k}$, $\widehat{t}_{2,k}$ be defined by (2.5). Assume that (A1)–(A4) hold. Assume also that $|\widehat{\mathcal{A}}_1(\lambda_n)| = K^\star_1$ and that $|\widehat{\mathcal{A}}_2(\lambda_n)| = K^\star_2$, with probabilty tending to one. Then,*

$$\mathbb{P}\left(\left\{\max_{1\leq k\leq K^\star_1} \left|\widehat{t}_{1,k} - t^\star_{1,k}\right| \leq n\delta_n\right\} \cap \left\{\max_{1\leq k\leq K^\star_2} \left|\widehat{t}_{2,k} - t^\star_{2,k}\right| \leq n\delta_n\right\}\right) \to 1,$$
$$\textit{as } n \to \infty. \tag{2.6}$$

The proof of Proposition 1 is based on the two following lemmas. The first one comes from the Karush-Kuhn-Tucker conditions of the optimization problem stated in (2.4). The second one allows us to control the supremum of the empirical mean of the noise.

Lemma 2. *Let $(Y_{i,j})_{1\leq i,j\leq n}$ be defined by (2.1). Then, $\widehat{\mathcal{U}} = \mathcal{X}\widehat{\mathcal{B}}$, where $\mathcal{X}$ and $\widehat{\mathcal{B}}$ are defined in (2.3) and (2.4) respectively, is such that*

$$\sum_{k=r_j+1}^{n} \sum_{\ell=q_j+1}^{n} Y_{k,\ell} - \sum_{k=r_j+1}^{n} \sum_{\ell=q_j+1}^{n} \widehat{\mathcal{U}}_{k,\ell} = \frac{\lambda_n}{2} sign(\widehat{\mathcal{B}}_j), \ \ if\ \widehat{\mathcal{B}}_j \neq 0, \tag{2.7}$$

$$\left|\sum_{k=r_j+1}^{n} \sum_{\ell=q_j+1}^{n} Y_{k,\ell} - \sum_{k=r_j+1}^{n} \sum_{\ell=q_j+1}^{n} \widehat{\mathcal{U}}_{k,\ell}\right| \leq \frac{\lambda_n}{2}, \ \ if\ \widehat{\mathcal{B}}_j = 0, \tag{2.8}$$

where q_j and r_j are the quotient and the remainder of the Euclidean division of $(j-1)$ by n, respectively, that is $(j-1) = nq_j + r_j$. In (2.7), sign denotes the function which is defined by $sign(x) = 1$, if $x > 0$, -1, if $x < 0$ and 0 if $x = 0$. Moreover, the matrix $\widehat{\mathbf{U}}$, which is such that $\widehat{\mathcal{U}} = Vec(\widehat{\mathbf{U}})$, is blockwise constant and satisfies $\widehat{U}_{i,j} = \widehat{\mu}_{k,\ell}$, if $\widehat{t}_{1,k-1} \leq i \leq \widehat{t}_{1,k} - 1$ and $\widehat{t}_{2,\ell-1} \leq j \leq \widehat{t}_{2,\ell} - 1$, $k \in \{1,\ldots,|\widehat{\mathcal{A}}_1(\lambda_n)|\}$, $\ell \in \{1,\ldots,|\widehat{\mathcal{A}}_2(\lambda_n)|\}$, where the $\widehat{t}_{1,k}$, $\widehat{t}_{2,k}$, $\widehat{\mathcal{A}}_1(\lambda_n)$ and $\widehat{\mathcal{A}}_2(\lambda_n)$ are defined in (2.5).

Lemma 3. *Let $(E_{i,j})_{1\leq i,j\leq n}$ be random variables satisfying (A1). Let also (v_n) and (x_n) be two positive sequences such that $v_n x_n^2/\log(n)\to\infty$, then*

$$\mathbb{P}\left(\max_{\substack{1\leq r_n<s_n\leq n\\ |r_n-s_n|\geq v_n}}\left|(s_n-r_n)^{-1}\sum_{j=r_n}^{s_n-1}E_{n,j}\right|\geq x_n\right)\to 0,\ \ as\ n\to\infty,$$

the result remaining valid if $E_{n,j}$ is replaced by $E_{j,n}$.

The proofs of Proposition 1, Lemmas 2 and 3 are given in Appendix A.

Remark. If $\mathbf{Y}$ is a non square matrix having n_1 rows and n_2 columns, with $n_1\neq n_2$, the result of Proposition 1 remains valid if in Assumption (A2) δ_n is replaced by δ_{n_1,n_2} satisfying $n_1\delta_{n_1,n_2}{J^\star_{\min}}^2/\log(n_2)\to\infty$ and $n_2\delta_{n_1,n_2}{J^\star_{\min}}^2/\log(n_1)\to\infty$, as n_1 and n_2 tend to infinity.

3. Implementation

In order to identify a series of change-points we look for the whole path of solutions in (2.4), *i.e.*, $\{\hat{\mathcal{B}}(\lambda),\lambda_{\min}<\lambda<\lambda_{\max}\}$ such that $|\hat{\mathcal{A}}(\lambda_{\max})|=0$ and $|\hat{\mathcal{A}}(\lambda_{\min})|=s$ with s a predefined maximal number of activated variables. To this end it is natural to adopt the famous homotopy/LARS strategy of [18, 6]. Such an algorithm identifies in Problem (2.4) the successive values of λ that correspond to the activation of a new variable, or the deletion of one that became irrelevant. However, the existing implementations do not apply here since the size of the design matrix $\mathcal{X}$ – even for reasonable n – is challenging both in terms of memory requirement and computational burden. To overcome these limitations, we need to take advantage of the particular structure of the problem. In the following lemmas (which are proved in Appendix A), we show that the most involving computations in the LARS can be made extremely efficiently thanks to the particular structure of $\mathcal{X}$.

Lemma 4. *For any vector $\mathbf{v}\in\mathbb{R}^{n^2}$, computing $\mathcal{X}\mathbf{v}$ and $\mathcal{X}^\top\mathbf{v}$ requires at worse $2n^2$ operations.*

Lemma 5. *Let $\mathcal{A}=\{a_1,\dots,a_K\}$ and for each j in $\mathcal{A}$ let us consider the Euclidean division of $j-1$ by n given by $j-1=nq_j+r_j$, then*

$$\left(\left(\mathcal{X}^\top\mathcal{X}\right)_{\mathcal{A},\mathcal{A}}\right)_{1\leq k,\ell\leq K}=\left((n-(q_{a_k}\vee q_{a_\ell}))\times(n-(r_{a_k}\vee r_{a_\ell}))\right)_{1\leq k,\ell\leq K}. \tag{3.1}$$

Moreover, for any non empty subset $\mathcal{A}$ of distinct indices in $\{1,\dots,n^2\}$, the matrix $\mathcal{X}_{\mathcal{A}}^\top\mathcal{X}_{\mathcal{A}}$ is invertible.

Lemma 6. *Assume that we have at our disposal the Cholesky factorization of $\mathcal{X}_{\mathcal{A}}^\top\mathcal{X}_{\mathcal{A}}$. The updated factorization on the extended set $\mathcal{A}\cup\{j\}$ only requires solving a $|\mathcal{A}|$-size triangular system, with complexity $\mathcal{O}(|\mathcal{A}|^2)$. Moreover, the downdated factorization on the restricted set $\mathcal{A}\backslash\{j\}$ requires a rotation with negligible cost to preserve the triangular form of the Cholesky factorization after a column deletion.*

Remark. We were able to obtain a closed-form expression of the inverse $(\mathcal{X}_{\mathcal{A}}^\top \mathcal{X}_{\mathcal{A}})^{-1}$ for some special cases of the subset $\mathcal{A}$, namely, when the quotients/ratios associated with the Euclidean divisions of the elements of $\mathcal{A}$ are endowed with a particular ordering. Moreover, for addressing any general problem, we rather solve systems involving $\mathcal{X}_{\mathcal{A}}^\top \mathcal{X}_{\mathcal{A}}$ by means of a Cholesky factorization which is updated along the homotopy algorithm. These updates correspond to adding or removing an element at a time in $\mathcal{A}$ and are performed efficiently as stated in Lemma 6.

These lemmas are the building blocks for our LARS implementation given in Algorithm 1, where we detail the leading complexity associated with each part. The global complexity is in $\mathcal{O}(mn^2 + ms^2)$ where m is the final number of steps in the while loop. These steps include all the successive additions and deletions needed to reach s, the final targeted number of active variables. At the end of day, we have m block wise prediction $\hat{\mathbf{Y}}$ associated with the series of m estimations of $\hat{\mathcal{B}}(\lambda)$. The above complexity should be compared with the usual complexity of the LARS algorithm, when no particular structure is at play in Problem (2.4): in such a case, a implementation of the LARS as in [2] would be at least in $\mathcal{O}(mn^4 + ms^2)$.

Algorithm 1: Fast LARS for two-dimensional change-point estimation

```
Input: data matrix $\mathbf{Y}$, maximal number of active variables $s$.
// Initialization
Start with no change-point $\mathcal{A} \leftarrow \emptyset$, $\hat{\mathcal{B}} = \mathbf{0}$
Compute current correlations $\hat{\mathbf{c}} = \mathcal{X}^\top \mathcal{Y}$ with Lemma 4    // $\mathcal{O}(n^2)$
while $\lambda > 0$ or $|\mathcal{A}| < s$ do
    // Update the set of active variables
    Determine next change-point(s) by setting $\lambda \leftarrow \|\hat{\mathbf{c}}\|_\infty$ and $\mathcal{A} \leftarrow \{j : \hat{\mathbf{c}}_j = \lambda\}$
    Update the Cholesky factorization of $\mathcal{X}_{\mathcal{A}}^\top \mathcal{X}_{\mathcal{A}}$ with Lemma 5    // $\mathcal{O}(|\mathcal{A}|^2)$

    // Compute the direction of descent
    Get the unormalized direction $\tilde{w}_{\mathcal{A}} \leftarrow \left(\mathcal{X}_{\cdot\mathcal{A}}^\top \mathcal{X}_{\cdot\mathcal{A}}\right)^{-1} \mathrm{sign}(\hat{c}_{\mathcal{A}})$    // $\mathcal{O}(|\mathcal{A}|^2)$
    Normalize $w_{\mathcal{A}} \leftarrow \alpha \tilde{w}_{\mathcal{A}}$ with $\alpha \leftarrow 1/\sqrt{\tilde{w}_{\mathcal{A}}^\top \mathrm{sign}(\hat{c}_{\mathcal{A}})}$
    Compute the equiangular vector $u_{\mathcal{A}} = \mathcal{X}_{\mathcal{A}} w_{\mathcal{A}}$ and $\mathbf{a} = \mathcal{X}^\top u_{\mathcal{A}}$ with Lemma 4
    // $\mathcal{O}(n^2)$

    // Compute the direction step
    Find the maximal step preserving equicorrelation $\gamma_{\mathrm{in}} \leftarrow \min^+_{j \in \mathcal{A}^c} \left\{ \frac{\lambda - \mathbf{c}_j}{\alpha - a_j}, \frac{\lambda + \mathbf{c}_j}{\alpha + a_j} \right\}$
    Find the maximal step preserving the signs $\gamma_{\mathrm{out}} \leftarrow \min^+_{j \in \mathcal{A}} \left\{ -\hat{\mathcal{B}}_{\mathcal{A}} / w_{\mathcal{A}} \right\}$
    The direction step that preserves both is $\hat{\gamma} \leftarrow \min(\gamma_{\mathrm{in}}, \gamma_{\mathrm{out}})$
    Update the correlations $\hat{\mathbf{c}} \leftarrow \hat{\mathbf{c}} - \hat{\gamma}\mathbf{a}$ and $\hat{\mathcal{B}}_{\mathcal{A}} \leftarrow \hat{\mathcal{B}}_{\mathcal{A}} + \hat{\gamma} w_{\mathcal{A}}$ accordingly    // $\mathcal{O}(n)$

    // Drop variable crossing the zero line
    if $\gamma_{\mathrm{out}} < \gamma_{\mathrm{in}}$ then
        Remove existing change-point(s) $\mathcal{A} \leftarrow \mathcal{A} \backslash \left\{ j \in \mathcal{A} : \hat{\mathcal{B}}_j = 0 \right\}$
        Downdate the Cholesky factorization of $\mathcal{X}_{\mathcal{A}}^\top \mathcal{X}_{\mathcal{A}}$    // $\mathcal{O}(|\mathcal{A}|)$
Output: Sequence of triplet $(\mathcal{A}, \lambda, \hat{\mathcal{B}})$ recorded at each iteration.
```

In all our experiments the number of steps m required to reach a given number of change-points s was always of the same order as s, as argued in the original LARS paper of [6]. However, there is no theoretical guarantee for m to be small: Indeed, it is possible to build artificial designs where m is equal to s^3, see [16]. Mairal and Yu in [16] propose a slight modification of the LARS to overcome this issue, that can be applied to our implementation.

Concerning the memory requirements, we only need to store the $n \times n$ data matrix $\mathbf{Y}$ once. Indeed, since we have at our disposal the analytic form of any sub matrix extracted from $\mathcal{X}^\top \mathcal{X}$, we never need to compute neither store this large $n^2 \times n^2$ matrix. This paves the way for quickly processing data with thousands of rows and columns.

4. Simulation study

In this Section, we conduct a set of simulation studies to assess the performances of our proposal. First, we report the computational performances of Algorithm 1 and of its practical implementation in terms of timings. Second, we report the statistical performances of our estimators (2.5) for recovering the true change-points by means of Receiver Operating Characteristic (ROC) curves.

4.1. Data generation

All synthetic data are generated from Model (2.1). We control the computational difficulty of the problem by varying the sample size n. The statistical difficulty is controlled by varying σ, the standard deviation of the Gaussian noise $\mathbf{E}$. We chose different patterns for the true matrix $\mathbf{U}^\star$ designed to mimic the variety of block matrix structures met in Hi-C data. These patterns are obtained by changing the parameters $\mu^\star_{k,\ell}$s, each of whom controlling the intensity in block (k,ℓ) of $\mathbf{U}^\star$. We consider four different scenarios, all with $K_1^\star = 4$ change-points along the rows and $K_2^\star = 4$ change-points along the columns.

$$\left(\mu_{k,\ell}^{\star,(1)}\right) = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}, \qquad \left(\mu_{k,\ell}^{\star,(2)}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$
$$\left(\mu_{k,\ell}^{\star,(3)}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad \left(\mu_{k,\ell}^{\star,(4)}\right) = \begin{pmatrix} 0 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 \\ -1 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{4.1}$$

Examples of matrices $\mathbf{Y}$ are displayed in Figure 4 for these four scenarios, with $n = 100$ and $\sigma = 1$ which corresponds to a relatively small level of noise in this problem.

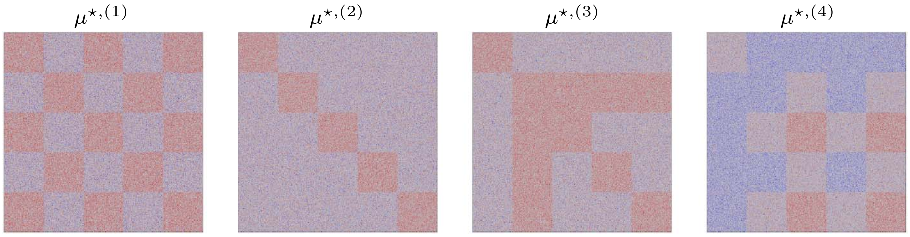

Fig 4. *Data matrices* $\mathbf{Y}$ *drawn from Model 2.1 for* $\sigma = 1, n = 100$ *and various block wise pattern for* $\mathbf{U}^{\star}$.

The first ($\mu^{\star,(1)}_{k,\ell}$) corresponds to a "checkerboard-shaped" matrix, that is, a natural two dimensional extension of a one dimensional piece-wise constant problem.

The second ($\mu^{\star,(2)}_{k,\ell}$) defines a block diagonal model that mimics the *cis-interactions* in the human Hi-C experiments: these are the most usual interactions found in the cell, which occur between nearby elements along the genome.

The third ($\mu^{\star,(3)}_{k,\ell}$) and fourth ($\mu^{\star,(4)}_{k,\ell}$) configurations describe more complex patterns that can be found when *trans-interactions* occur in Hi-C experiments. They also correspond to more difficult change-points problems.

4.2. Competitors and implementation details

In our experiments, we compare our methodology with popular methods for segmentation and variable selection that we adapted to the specific problem of two-dimensional change-points detection:

1. First, we adapt Breiman et al.'s classification and regression trees [4] (hereafter called `CART`) by using the successive boundaries provided by CART as change-points for the two-dimensional data. We use the implementation provided by the publicly available R package `rpart`.
2. Second, we adapt Harchaoui and Lévy-Leduc's method [9] (hereafter `HL`), which is the exact one-dimensional counterpart of our approach. To analyse two-dimensional data, we apply this procedure to each row of $\mathbf{Y}$ in order to recover the change-points of each row. The change-points appearing in the different rows are claimed to be change-points for the two-dimensional data either if they appear at least in one row (variant `HL1`) or if they appear in $([n/2]+1)$ rows (variant `HL2`). This approach is fitted by solving n Lasso problems (one per row of $\mathbf{Y}$) by means of the R package `glmnet`.
3. Third, we consider an adaptation of the fused-Lasso (hereafter `FL2D`). Indeed, as illustrated in the introduction, the basic 2-dimensional fused-Lasso for signal approximator is not tailored for recovering change points. We thus consider the following variant, which applied a fused-Lasso penalty

on the following linear model:

$$\mathcal{Y} = \underbrace{\begin{pmatrix} \mathbb{1}_n & 0_n & \cdots & \cdots & 0_n & \mathbb{I}_n \\ 0_n & \mathbb{1}_n & \ddots & & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \mathbb{1}_n & 0_n & \vdots \\ 0_n & \cdots & \cdots & 0_n & \mathbb{1}_n & \mathbb{I}_n \end{pmatrix}}_{\mathcal{X}^{(FL)}} \underbrace{\begin{pmatrix} \beta_1^{(FL)} \\ \vdots \\ \beta_n^{(FL)} \\ \beta_{n+1}^{(FL)} \\ \vdots \\ \beta_{2n}^{(FL)} \end{pmatrix}}_{\mathcal{B}^{(FL)}} + \mathcal{E}$$

where $\mathbb{1}_n$ (resp. 0_n) is a size-n column vector of ones (resp. zeros), $\mathbb{I}_n$ a $n \times n$-diagonal matrix of ones and $\mathcal{Y}, \mathcal{E}$ are defined as in Equation (2.3). The `FL2D` method detects a change-point in columns (resp. in row) if two successive values $\beta_i^{(FL)}$ and $\beta_{i+1}^{(FL)}$ with $1 \leq i \leq n-1$ (resp. $n+1 \leq i \leq 2n-1$) are different. To solve this problem, we must fit a general fused-Lasso problem. We rely on the R package `genlasso` for this task.

4. Finally, our own procedure, that we call `blockseg`, is implemented in the R package `blockseg` which is available from the Comprehensive R Archive Network (CRAN, [19]). Most of the computation are performed in `C++` using the library `armadillo` for linear algebra [21].

In what follows, all experiments were conducted on a Linux workstation with Intel Xeon 2.4 GHz processor and 8 GB of memory.

4.3. Numerical performances

We start by presenting in Figure 5 the computational times for 100 runs of each method applied to a matrix drawn from the "checkerboard" scenario, with $n = 100$ and $\sigma = 5$. Each run provides the estimated change-points in rows and columns for all the possible number of change-points in rows and in columns, that is all the values between 1 and n.

Independent of its statistical performance, we can see on this small problem that the adaptation of the fused-Lasso cannot be used for analyzing real Hi-C problems. On the other hand, our modified CART procedure is extremely fast. However, we will see that its statistical performances are quite poor. Finally, our implementation `blockseg` is quite efficient as it clearly outperforms `HL`. This should be emphasized since `blockseg` is a two-dimensional method dealing with data with size n^2, while `HL` is a 1-dimensional approach that addresses two univariate problems of size n.

We now consider `blockseg` on its own in order to study the scalability of our approach regarding the problem dimension. To this end, we generated "checkerboard" matrix $\left(\mu_{k,\ell}^{\star,(1)}\right)$ given in (4.1) with various sizes n (from 100 to 5000) and various values of the maximal number of activated variables s (from 50 to 750). The median runtimes obtained from 4 replications (+ 2 for warm-up)

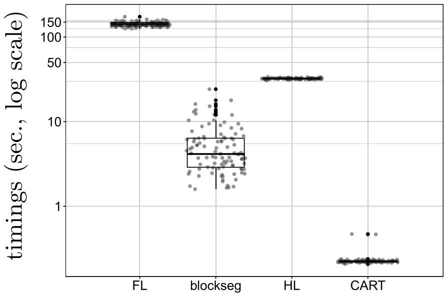

Fig 5. *Boxplots of the computational times for 100 runs (one point per run) of each procedure: CART methodology (`CART`), adaptation of [9] (`HL`), our method (`blockseg`) and fused LASSO (`FL`).*

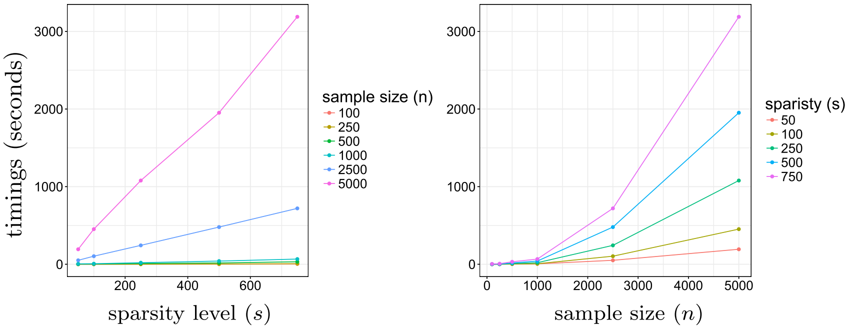

Fig 6. *Left: Computational time (in seconds) for various values of n as a function of the sparsity level $s = |\mathcal{A}|$ reached at the end of the algorithm. Right: Computation time (in seconds) as a function of sample size n.*

are reported in Figures 6. The left (resp. the right) panel gives the runtimes in seconds as a function of s (resp. of n). These results give experimental evidence for the theoretical complexity $\mathcal{O}(mn^2 + ms^2)$ that we established in Section 3 and thus for the computational efficiency of our approach: applying `blockseg` to matrices containing 10^7 entries takes less than 2 minutes for $s = 750$.

4.4. Statistical performances

We evaluate the performance of the different competitors for recovering the true change-points in the 4 scenarios defined in Section 4.1 for an increasing level of difficulty. We draw 1000 datasets for each scenario for a varying level of noise $\sigma \in \{1, 2, 5, 10\}$ and for a problem size of $n = 100$. Note that we use this relatively small problem size to allow the comparison with methods `HL` and `FL2D` that would not work for greater values of n.

Figure 7 shows the results in terms of receiver operating characteristic (ROC) curves for recovering the change-points in rows, averaged over the 1000 runs.

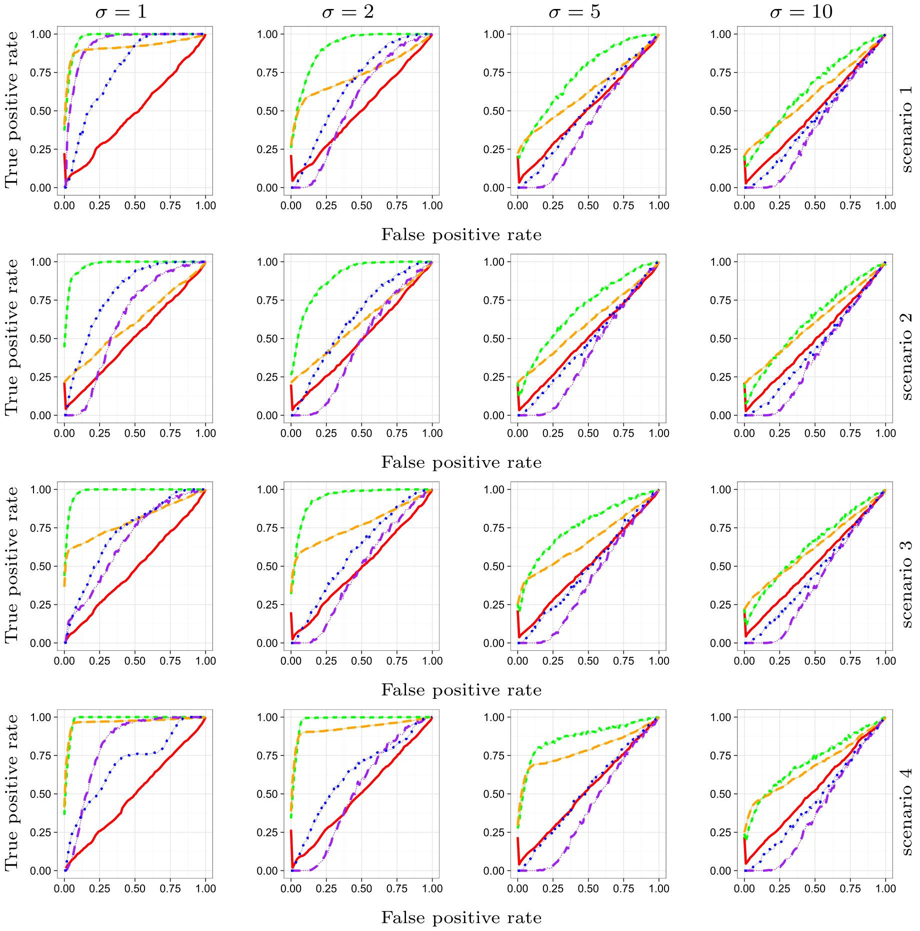

Fig 7. *ROC curves for the estimated change-points in rows for* `blockseg` *(dotted green),* `HL1` *(double-dashed purple),* `HL2` *(in dotted blue),* `CART` *(solid red) and* `FL2D` *(long-dashed orange). Each row is associated to a scenario depicted in Section 4.1.*

Similar results hold for the change-points in columns. This Figure exhibits the very good performance of our method, which outperforms its competitors by retrieving the change-points with a very small error rate even in high noise level frameworks. Moreover, our method seems to be less sensitive to the block pattern shape in matrix $\mathbf{U}$ than the other ones. In order to further assess our approach we give in Figure 8 the boxplots of the Area Under Curve (AUC) for the different ROC curves. We also give in Table 1 the mean of the AUC and the associated standard deviation.

In order to further compare the different approaches we generated matrices $\mathbf{Y}$ satisfying Model (2.1) with a "checkerboard" matrix $\left(\mu_{k,\ell}^{\star,(1)}\right)$ given in (4.1) for $n \in \{50, 100, 250\}$. We observe from Table 2 that the performance of our

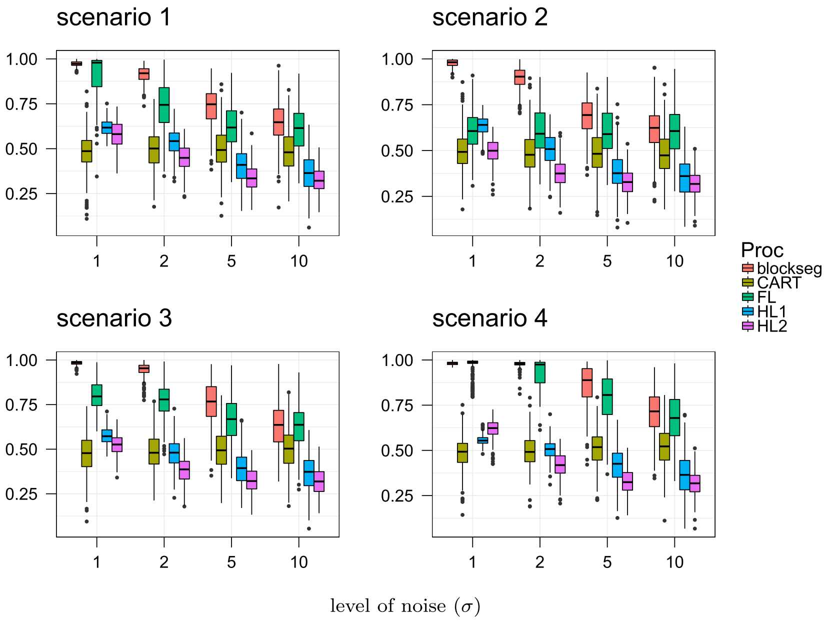

Fig 8. *Boxplots of the area under the ROC curve for the different scenarios and the different algorithms as a function of the noise variance.*

Table 1
Mean and standard deviation of the area under the ROC curve for the different scenarios, different algorithms and different values of the noise variance.

	Scenario 1				Scenario 2			
	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$
blockseg	0.972 (0.0145)	0.913 (0.0421)	0.733 (0.0988)	0.644 (0.118)	0.977 (0.0206)	0.896 (0.0555)	0.689 (0.107)	0.617 (0.123)
FL2D	0.918 (0.102)	0.738 (0.139)	0.623 (0.127)	0.608 (0.13)	0.608 (0.116)	0.603 (0.125)	0.601 (0.127)	0.603 (0.127)
HL1	0.618 (0.0427)	0.535 (0.0708)	0.407 (0.102)	0.363 (0.108)	0.635 (0.0535)	0.505 (0.0874)	0.382 (0.105)	0.351 (0.107)
HL2	0.576 (0.0744)	0.448 (0.0713)	0.337 (0.0734)	0.323 (0.072)	0.498 (0.0653)	0.374 (0.0777)	0.326 (0.0727)	0.317 (0.0745)
CART	0.482 (0.107)	0.497 (0.107)	0.498 (0.117)	0.486 (0.119)	0.496 (0.112)	0.487 (0.124)	0.491 (0.126)	0.484 (0.118)

	Scenario 3				Scenario 4			
	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$
blockseg	0.983 (0.0114)	0.945 (0.0391)	0.758 (0.113)	0.63 (0.125)	0.983 (0.00927)	0.977 (0.0179)	0.866 (0.102)	0.707 (0.124)
FL2D	0.799 (0.0855)	0.772 (0.0956)	0.667 (0.121)	0.623 (0.121)	0.969 (0.051)	0.931 (0.0722)	0.789 (0.135)	0.68 (0.134)
HL1	0.575 (0.0458)	0.479 (0.0819)	0.391 (0.0981)	0.368 (0.105)	0.556 (0.0252)	0.504 (0.0514)	0.418 (0.0974)	0.368 (0.11)
HL2	0.524 (0.0612)	0.384 (0.0711)	0.326 (0.0716)	0.319 (0.0738)	0.616 (0.0527)	0.416 (0.0696)	0.327 (0.0714)	0.316 (0.067)
CART	0.474 (0.106)	0.485 (0.11)	0.495 (0.114)	0.502 (0.115)	0.484 (0.0905)	0.493 (0.0889)	0.512 (0.0985)	0.516 (0.111)

TABLE 2

Mean and standard deviation of the area under the ROC curve as a function of the standard deviation of the noise, the algorithms and the size of the matrices. The crosses correspond to cases where the results are not available.

	$\sigma = 1$			$\sigma = 2$		
	$n = 50$	$n = 100$	$n = 250$	$n = 50$	$n = 100$	$n = 250$
blockseg	0.896 (0.0425)	0.972 (0.0162)	0.993 (0.00463)	0.791 (0.0789)	0.923 (0.0398)	0.982 (0.00865)
FL2D	0.814 (0.132)	0.906 (0.0997)	X	0.679 (0.133)	0.753 (0.128)	X
HL1	0.574 (0.0598)	0.619 (0.0426)	0.66 (0.0255)	0.467 (0.0899)	0.527 (0.084)	0.611 (0.0513)
HL2	0.56 (0.101)	0.573 (0.0642)	0.59 (0.0432)	0.424 (0.0972)	0.451 (0.0713)	0.472 (0.0467)
CART	0.445 (0.123)	0.479 (0.108)	0.498 (0.0589)	0.487 (0.125)	0.487 (0.114)	0.512 (0.0708)

	$\sigma = 5$			$\sigma = 10$		
	$n = 50$	$n = 100$	$n = 250$	$n = 50$	$n = 100$	$n = 250$
blockseg	0.646 (0.127)	0.739 (0.11)	0.91 (0.0394)	0.577 (0.112)	0.642 (0.124)	0.766 (0.0867)
FL2D	0.631 (0.132)	0.629 (0.125)	X	0.602 (0.118)	0.616 (0.115)	X
HL1	0.382 (0.106)	0.397 (0.107)	0.481 (0.0909)	0.364 (0.103)	0.35 (0.108)	0.386 (0.115)
HL2	0.333 (0.0905)	0.342 (0.0775)	0.341 (0.0451)	0.325 (0.083)	0.313 (0.0729)	0.317 (0.0539)
CART	0.488 (0.115)	0.501 (0.119)	0.497 (0.0917)	0.466 (0.129)	0.483 (0.131)	0.48 (0.117)

method are on a par with those of FL2D for $n = 50$ and 100. However, for $n = 250$ the computational burden of FL2D is so large that the results are not available, see the blue crosses in Table 2. The AUC are also displayed with boxplots in Figure 9.

5. Model selection

In the previous experiments we did not need to explain how to choose the number of estimated change-points since we used ROC curves for comparing the methodologies. However, in real data applications, it is necessary to propose a methodology for estimating the number of change-points. This is what we explain in the following.

In practice, we take $s = K_{\max}^2$ where $K_{\max}$ is an upper bound for $K_1^\star$ and $K_2^\star$. For choosing the final change-points we shall adapt the well-known *stability selection* approach devised by [17]. More precisely, we randomly choose M times $n/2$ columns and $n/2$ rows of the matrix $\mathbf{Y}$ and for each subsample we select $s = K_{\max}^2$ active variables. Finally, after the M data resamplings, we keep the change-points which appear a number of times larger than a given threshold. By the definition of the change-points given in (2.5), a change-point $\widehat{t}_{1,k}$ or $\widehat{t}_{2,\ell}$ may appear several times in a given set of resampled observations. Hence, the

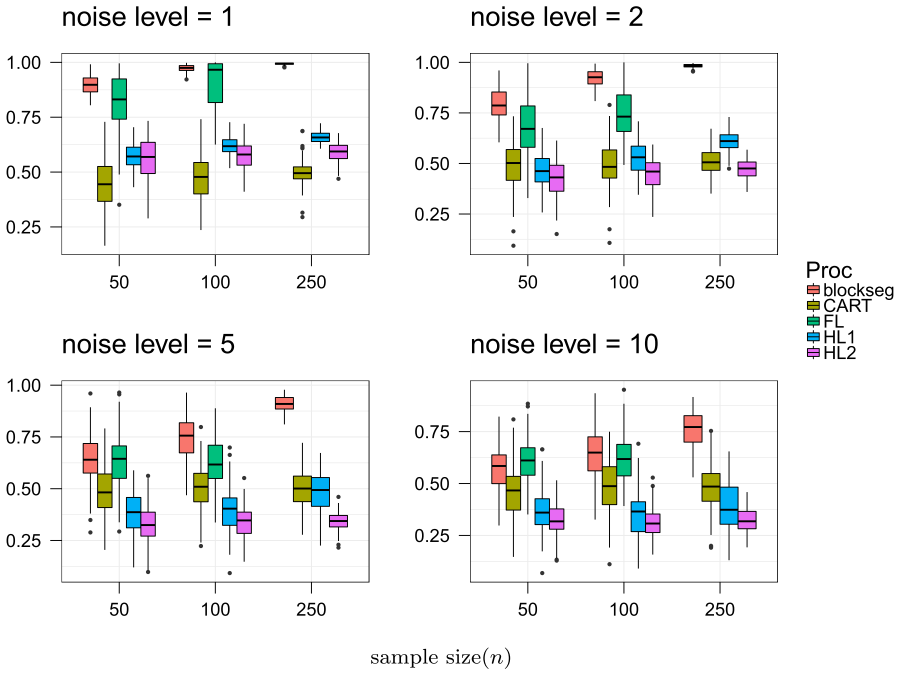

FIG 9. *Boxplots of the area under the ROC curve as a function of the standard deviation of the noise, the size of the matrices and the methods.*

score associated with each change-point corresponds to the sum of the number of times it appears in each of the M subsamplings.

To evaluate the performances of this methodology, we generated observations according to the "checkerboard" model defined in (2.1) with $(\mu^{\star,(1)}_{k,\ell})$ defined in (4.1), $s = 225$ and $M = 100$. The results are given in Figure 10 which displays the score associated to each change-point for a given matrix $\mathbf{Y}$. We can see from this figure that there are some spurious change-points close to the true change-point positions. In order to identify the most representative change-point in a given neighborhood, we keep the one with the largest score among a set of contiguous candidates. The result of such a post-processing is displayed in the first row of Figure 11. More precisely the boxplots associated to the estimation of $K^\star_1$ (resp. the histograms of the estimated change-points in rows) are displayed for different values of σ and different thresholds expressed as a percentage of the largest score. We can see from these figures that when the threshold is in the interval $[20, 40]$ the number and the location of the change-points are very well estimated even in the high noise level case.

In order to further assess our methodology including the post-processing step and to be in a framework closer to our real data application, we generated observations following (2.1) with $n = 1000$ and $K^\star_1 = K^\star_2 = 100$ where we used for the matrix $\mathbf{U}$ the same shape as the one of the matrix $(\mu^{\star,(1)}_{k,\ell})$ except

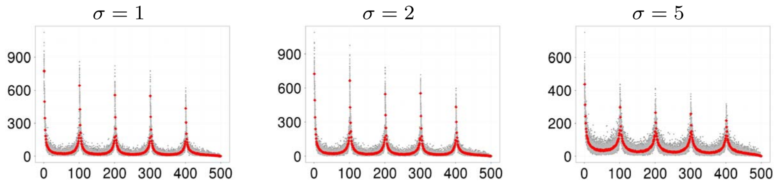

FIG 10. *Scores associated to each estimated change-points for different values of σ; the true change-point positions in rows and columns are located at 101, 201, 301 and 401.*

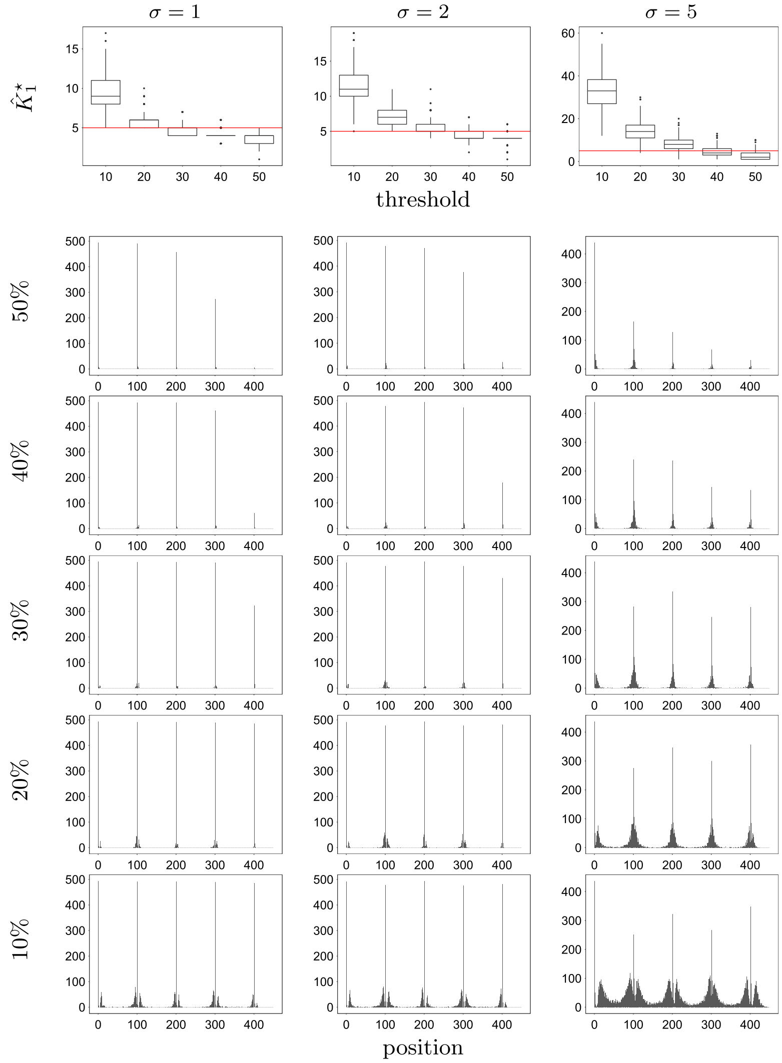

FIG 11. *Top row: Boxplots of the estimation of $K_1^\star$ for different values of σ and threshold after the post-processing step; 3 bottom rows: Barplots of the estimated change-points for different values of σ (columns) and different thresholds (rows) for the model $\mu^{\star,(1)}$.*

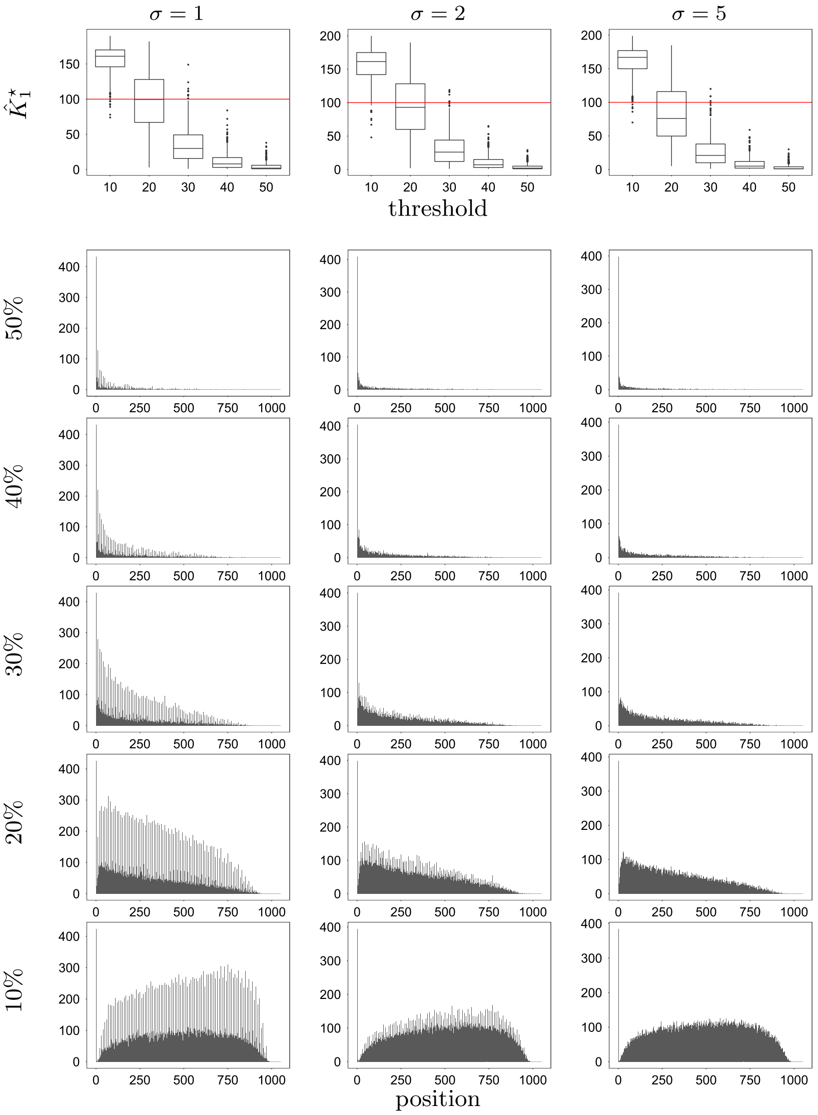

Fig 12. *Top row: Boxplots of the estimation of $K_1^\star$ for different values of σ and thresholds after the post-processing step; 3 bottom rows: Barplots of the estimated change-points for different variances (columns) and different thresholds (rows) in the case where $n = 1000$ and $K_1^\star = K_2^\star = 100$.*

that $K_1^\star = K_2^\star = 100$. In this framework, the proportion of change-points is thus ten times larger than the one of the previous case. The corresponding results are displayed in Figures 12 and 13. We can see from the last figure that taking a threshold equal to 20% provides the best estimations of the number and of the change-point positions. This threshold corresponds to the lower bound

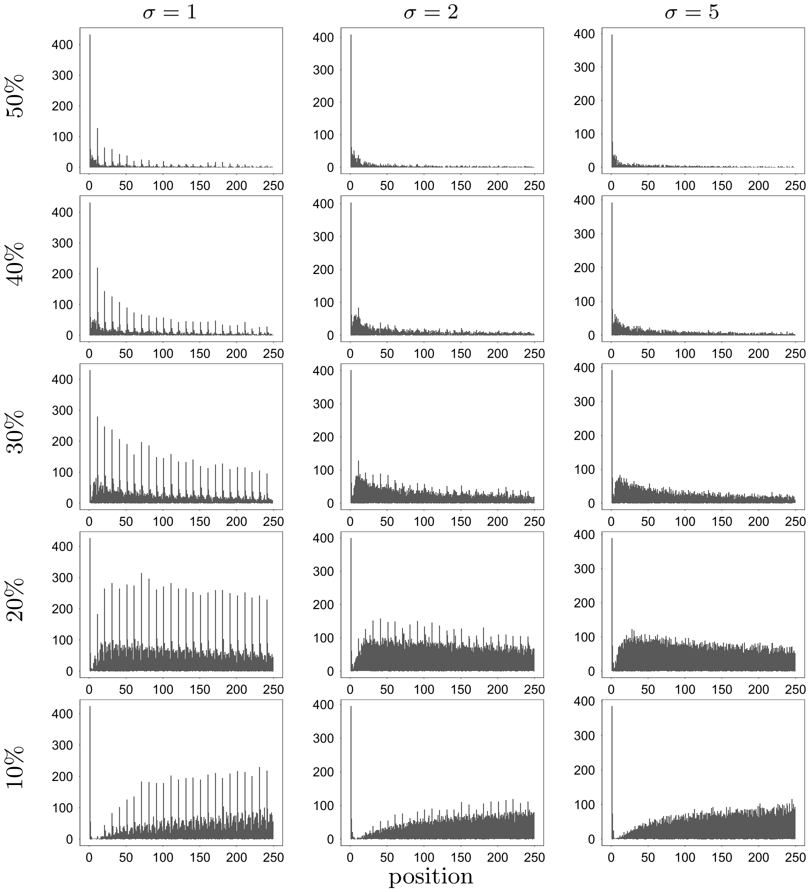

FIG 13. *Zoom of the barplots of Figure 12.*

of the thresholds interval obtained in the previous configuration. Our package `blockseg` provides an estimation of the matrix $\mathbf{U}$ for any threshold given by the user as we shall explain in the next section.

6. Application to HiC data

In this section, we apply our methodology to publicly available HiC data (`http://chromosome.sdsc.edu/mouse/hi-c/download.html`) already studied by [5]. This technology is based on a deep sequencing approach and provides read pairs corresponding to pairs of genomic loci that physically interacts in the nucleus, see [13] for more details. The raw measurements provided by HiC data is therefore a list of pairs of locations along the chromosome, at the nucleotide resolution. These measurement are often summarized as a square matrix where each entry at row i and column j stands for the total number of read pairs

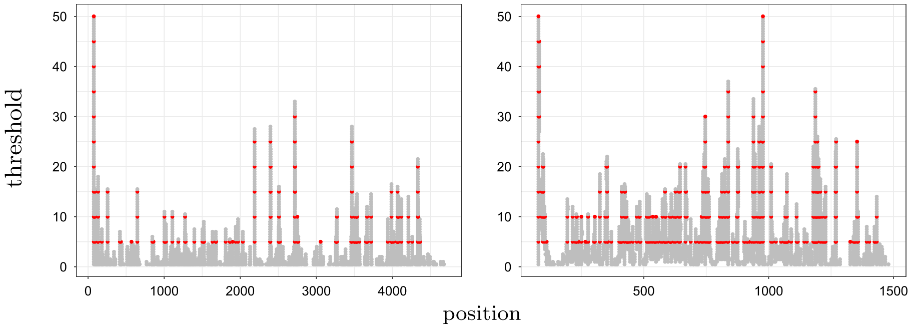

Fig 14. *Plots of the estimated change-points locations (x-axis) for different thresholds (y-axis) from 0.5% to 50% by 0.5% for Chromosome 1 (left) and Chromosome 19 (right). The estimated change-point locations associated to threshold which are multiples of 5% are displayed in red.*

matching in position i and position j, respectively. Positions refer here to a sequence of non-overlapping windows of equal sizes covering the genome. The number of windows may vary from one study to another: [13] considered a Mb resolution, whereas [5] went deeper and used windows of 40kb (called hereafter the resolution).

In our study, we processed the interaction matrices of Chromosomes 1 and 19 of the mouse cortex at a resolution 40 kb and we compared the number and the location of the estimated change-points found by our approach with those obtained by [5] on the same data since no ground truth is available. More precisely, in the case of Chromosome 1, $n = 4930$ and in the case of Chromosome 19, $n = 1534$.

Let us first give the results obtained by using our methodology. Figure 14 displays the change-point locations obtained for the different values of the threshold used in our adaptation of the stability selection approach and defined in Section 5. The corresponding estimated matrices $\widehat{\mathbf{Y}} = \widehat{\mathbf{U}}$ for Chromosome 1 and 19 are displayed in Figure 15 when the thresholds are equal to 10, 15 and 20%, which correspond to the red horizontal levels in Figure 14.

In order to compare our approach with the technique devised by [5], we display in Figure 16 the number of change-points in rows found by our methodology as a function of the threshold and a red line corresponding to the number of change-points found by [5]. Note that we did not display the change-points in columns in order to save space since they are similar.

We also compute the two parts of the Hausdorff distance for the change-points in rows which is defined by

$$d\left(\widehat{\boldsymbol{t}}_B, \widehat{\boldsymbol{t}}\right) = \max\left(d_1\left(\widehat{\boldsymbol{t}}_B, \widehat{\boldsymbol{t}}\right), d_2\left(\widehat{\boldsymbol{t}}_B, \widehat{\boldsymbol{t}}\right)\right) , \tag{6.1}$$

where $\widehat{\boldsymbol{t}}$ and $\widehat{\boldsymbol{t}}_B$ are the change-points in rows found by our approach and [5], respectively. In (6.1),

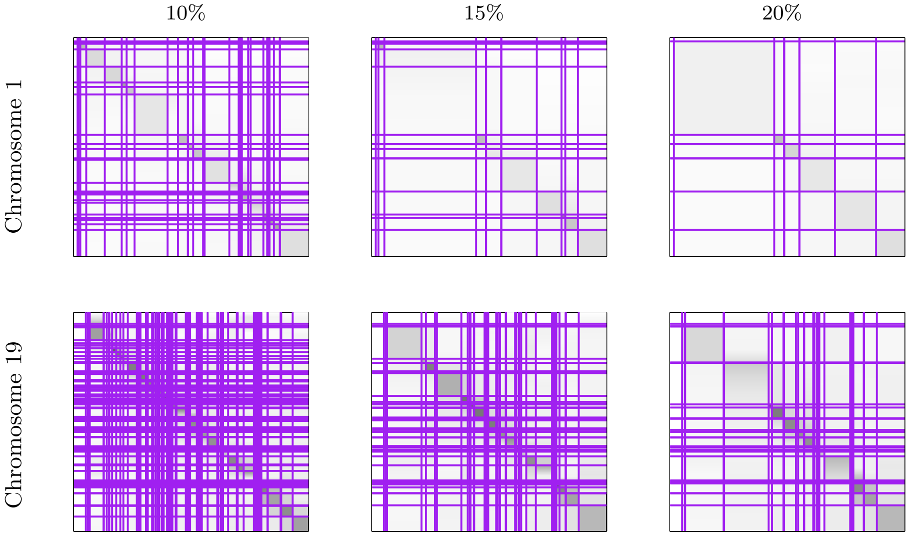

Fig 15. *Estimated matrices* $\widehat{\mathbf{Y}} = \widehat{\mathbf{U}}$ *for Chromosomes 1 and 19 for the thresholds 10, 15 and 20%.*

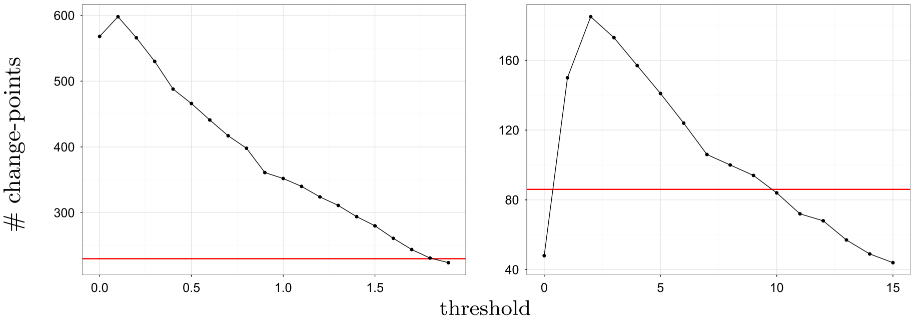

Fig 16. *Number of change-points in rows found by our approach as a function of the threshold (in %) for the interaction matrices of Chromosome 1 (left) and Chromosome 19 (right) of the mouse cortex. The red line corresponds to the number of change-points found by [5].*

$$d_1(\mathbf{a}, \mathbf{b}) = \sup_{b \in \mathbf{b}} \inf_{a \in \mathbf{a}} |a - b|, \tag{6.2}$$

$$d_2(\mathbf{a}, \mathbf{b}) = d_1(\mathbf{b}, \mathbf{a}). \tag{6.3}$$

More precisely, Figure 17 displays the boxplots of the d_1 and d_2 parts of the Hausdorff distance without taking the supremum in orange and blue, respectively.

We can observe from Figure 17 that some differences indeed exist between the segmentations produced by the two approaches but that the boundaries of the blocks are quite close when the number of estimated change-points are the same, which is the case when thresh = 1.8% (left) and 10% (right).

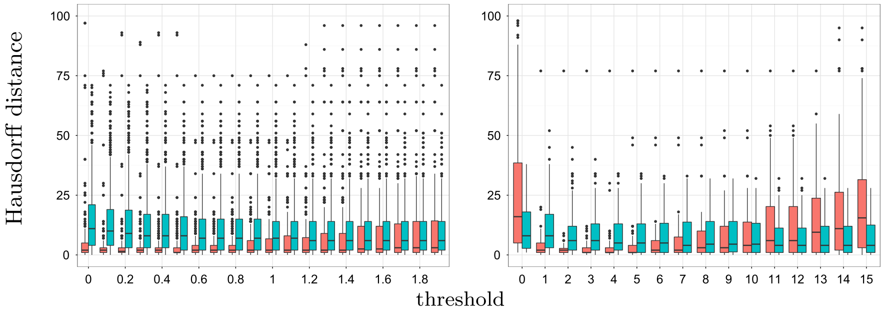

Fig 17. *Boxplots for the infimum parts of the Hausdorff distances d_1 (orange) and d_2 (blue) between the change-points found by [5] and our approach for the Chromosome 1 (left) and the Chromosome 19 (right) of the mouse cortex for the different thresholds in %.*

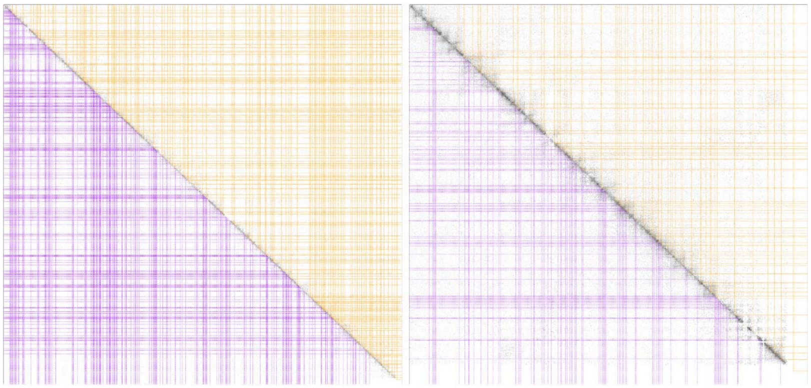

Fig 18. *Topological domains detected by [5] (upper triangular part of the matrix) and by our method (lower triangular part of the matrix) from the interaction matrix of Chromosome 1 (left) and Chromosome 19 (right) of the mouse cortex with a threshold giving 232 (resp 85) estimated change-points in rows and columns.*

In the case where the number of estimated change-points are on a par with those of [5], we can see from Figure 18 that the change-points found with our strategy present a lot of similarities with those found by the HMM based approach of [5]. However, contrary to our method, the approach of [5] can only deal with binned data at the resolution of several kilobases of nucleotides. The very low computational burden of our strategy paves the way for processing data collected at a very high resolution, namely at the nucleotide resolution, which is one of the main current challenges of molecular biology.

7. Conclusion

In this paper, we proposed a novel approach for retrieving the boundaries of a block wise constant matrix corrupted with noise by rephrasing this problem as a variable selection issue. Our approach is implemented in the R package `blockseg` which is available from the Comprehensive R Archive Network (CRAN).

In the course of this study, we have shown that our method has two main features which make it very attractive. Firstly, it is very efficient both from the theoretical and practical point of view. Secondly, its very low computational burden makes its use possible on very large data sets coming from molecular biology.

However, in view of applying our approach to HiC data experiments, it could be interesting to develop a methodology which could deal with change-points that do not affect all rows and columns simultaneously. This will be the subject of a future work.

Appendix A: Proofs

A.1. Proofs of statistical results

Proof of Lemma 2. A necessary and sufficient condition for a vector $\widehat{\mathcal{B}}$ in $\mathbb{R}^{n^2}$ to minimize the function Φ defined by: $\Phi(\mathcal{B}) = \sum_{i=1}^{n^2}(\mathcal{Y}_i - (\mathcal{X}\mathcal{B})_i)^2 + \lambda_n \sum_{i=1}^{n^2} |\mathcal{B}_i|$, is that the zero vector in $\mathbb{R}^{n^2}$ belongs to the subdifferential of Φ at $\widehat{\mathcal{B}}$ that is:

$$\begin{aligned} \left(\mathcal{X}^\top(\mathcal{Y} - \mathcal{X}\widehat{\mathcal{B}})\right)_j &= \frac{\lambda_n}{2}, \quad \text{if } \widehat{\mathcal{B}}_j \neq 0, \\ \left|\left(\mathcal{X}^\top(\mathcal{Y} - \mathcal{X}\widehat{\mathcal{B}})\right)_j\right| &\leq \frac{\lambda_n}{2}, \quad \text{if } \widehat{\mathcal{B}}_j = 0. \end{aligned}$$

Using that $\mathcal{X}^\top\mathcal{Y} = (\mathbf{T} \otimes \mathbf{T})^\top \mathcal{Y} = (\mathbf{T}^\top \otimes \mathbf{T}^\top)\mathcal{Y} = \text{Vec}(\mathbf{T}^\top \mathbf{Y} \mathbf{T})$, where $(\mathbf{T}^\top \mathbf{Y} \mathbf{T})_{i,j} = \sum_{k=i}^{n} \sum_{\ell=j}^{n} Y_{k,\ell}$, and that $\widehat{\mathcal{U}} = \mathcal{X}\widehat{\mathcal{B}}$, Lemma 2 is proved. □

Proof of Lemma 3. Note that

$$\begin{aligned} &\mathbb{P}\left(\max_{\substack{1 \leq r_n < s_n \leq n \\ |r_n - s_n| \geq v_n}} \left|(s_n - r_n)^{-1} \sum_{j=r_n}^{s_n - 1} E_{n,j}\right| \geq x_n\right) \\ &\qquad \leq \sum_{\substack{1 \leq r_n < s_n \leq n \\ |r_n - s_n| \geq v_n}} \mathbb{P}\left(\left|(s_n - r_n)^{-1} \sum_{j=r_n}^{s_n - 1} E_{n,j}\right| \geq x_n\right). \end{aligned}$$

By (A1) and the Markov inequality, we get that for all positive η,

$$\begin{aligned} \mathbb{P}\left((s_n - r_n)^{-1} \sum_{j=r_n}^{s_n - 1} E_{n,j} \geq x_n\right) &\leq \exp[-\eta(s_n - r_n)x_n](\mathbb{E}(\exp(\eta E_{1,1})))^{s_n - r_n} \\ &\leq \exp[-\eta(s_n - r_n)x_n + \beta\eta^2(s_n - r_n)]. \end{aligned}$$

By taking $\eta = x_n/(2\beta)$, we get that

$$\mathbb{P}\left((s_n - r_n)^{-1}\sum_{j=r_n}^{s_n-1} E_{n,j} \geq x_n\right) \leq \exp[-x_n^2(s_n - r_n)/(4\beta)].$$

Since the same result is valid for $-E_{n,j}$, we get that

$$\mathbb{P}\left(\max_{\substack{1\leq r_n<s_n\leq n\\ |r_n-s_n|\geq v_n}}\left|(s_n - r_n)^{-1}\sum_{j=r_n}^{s_n-1} E_{n,j}\right| \geq x_n\right) \leq 2n^2\exp[-x_n^2 v_n/(4\beta)],$$

which concludes the proof of Lemma 3. □

Proof of Proposition 1. Since

$$\begin{aligned}&\mathbb{P}\left(\left\{\max_{1\leq k\leq K_1^\star}\left|\widehat{t}_{1,k} - t_{1,k}^\star\right| > n\delta_n\right\} \cup \left\{\max_{1\leq k\leq K_2^\star}\left|\widehat{t}_{2,k} - t_{2,k}^\star\right| > n\delta_n\right\}\right)\\ &\quad\leq \mathbb{P}\left(\max_{1\leq k\leq K_1^\star}\left|\widehat{t}_{1,k} - t_{1,k}^\star\right| > n\delta_n\right) + \mathbb{P}\left(\max_{1\leq k\leq K_2^\star}\left|\widehat{t}_{2,k} - t_{2,k}^\star\right| > n\delta_n\right),\end{aligned} \tag{A.1}$$

it is enough to prove that both terms in (A.1) tend to zero for proving (2.6). We shall only prove that the second term in the rhs of (A.1) tends to zero, the proof being the same for the first term. Since $\mathbb{P}(\max_{1\leq k\leq K_2^\star}|\widehat{t}_{2,k} - t_{2,k}^\star| > n\delta_n) \leq \sum_{k=1}^{K_2^\star}\mathbb{P}(|\widehat{t}_{2,k} - t_{2,k}^\star| > n\delta_n)$, it is enough to prove that for all k in $\{1,\dots,K_2^\star\}$, $\mathbb{P}(A_{n,k}) \to 0$, where $A_{n,k} = \{|\widehat{t}_{2,k} - t_{2,k}^\star| > n\delta_n\}$. Let C_n be defined by

$$C_n = \left\{\max_{1\leq k\leq K_2^\star}|\widehat{t}_{2,k} - t_{2,k}^\star| < I_{\min,2}^\star/2\right\}. \tag{A.2}$$

It is enough to prove that, for all k in $\{1,\dots,K_2^\star\}$, $\mathbb{P}(A_{n,k}\cap C_n)$ and $\mathbb{P}(A_{n,k}\cap\overline{C_n})$ tend to 0, as n tends to infinity.

Let us first prove that for all k in $\{1,\dots,K_2^\star\}$, $\mathbb{P}(A_{n,k}\cap C_n) \to 0$. Observe that (A.2) implies that $t_{2,k-1}^\star < \widehat{t}_{2,k} < t_{2,k+1}^\star$, for all k in $\{1,\dots,K_2^\star\}$. For a given k, let us assume that $\widehat{t}_{2,k} \leq t_{2,k}^\star$. Applying (2.7) and (2.8) with $r_j + 1 = n$, $q_j + 1 = \widehat{t}_{2,k}$ on the one hand and $r_j + 1 = n$, $q_j + 1 = t_{2,k}^\star$ on the other hand, we get that

$$\left|\sum_{j=\widehat{t}_{2,k}}^{t_{2,k}^\star-1} Y_{n,j} - \sum_{j=\widehat{t}_{2,k}}^{t_{2,k}^\star-1}\widehat{\mathcal{U}}_{n,j}\right| \leq \lambda_n.$$

Hence using (2.1), the notation: $\mathbf{E}([a,b];[c,d]) = \sum_{i=a}^{b}\sum_{j=c}^{d} E_{i,j}$ and the definition of $\widehat{\mathcal{U}}$ given by Lemma 2, we obtain that

$$\left|(t_{2,k}^\star - \widehat{t}_{2,k})(\mu_{K_1^\star+1,k}^\star - \widehat{\mu}_{K_1^\star+1,k+1}) + \mathbf{E}(n;[\widehat{t}_{2,k}, t_{2,k}^\star - 1])\right| \leq \lambda_n,$$

which can be rewritten as follows

$$\Big|(t^\star_{2,k}-\widehat{t}_{2,k})(\mu^\star_{K_1^\star+1,k}-\mu^\star_{K_1^\star+1,k+1})+(t^\star_{2,k}-\widehat{t}_{2,k})(\mu^\star_{K_1^\star+1,k+1}-\widehat{\mu}_{K_1^\star+1,k+1}) \\ +\mathbf{E}(n;[\widehat{t}_{2,k},t^\star_{2,k}-1])\Big| \leq \lambda_n.$$

Thus,

$$\begin{aligned}\mathbb{P}(A_{n,k}\cap C_n) &\leq \mathbb{P}(\lambda_n/(n\delta_n)\geq|\mu^\star_{K_1^\star+1,k}-\mu^\star_{K_1^\star+1,k+1}|/3) \\ &+\mathbb{P}(\{|\mu^\star_{K_1^\star+1,k}-\widehat{\mu}_{K_1^\star+1,k+1}|\geq|\mu^\star_{K_1^\star+1,k}-\mu^\star_{K_1^\star+1,k+1}|/3\}\cap C_n) \\ &+\mathbb{P}(\{|\mathbf{E}(n;[\widehat{t}_{2,k},t^\star_{2,k}-1])|/|t^\star_{2,k}-\widehat{t}_{2,k}|\geq|\mu^\star_{K_1^\star+1,k}-\mu^\star_{K_1^\star+1,k+1}|/3\}\cap A_{n,k}).\end{aligned} \tag{A.3}$$

The first term in the rhs of (A.3) tends to 0 by (A3). By Lemma 3 with $x_n = J^\star_{\min}/3$, $v_n = n\delta_n$ and (A2) the third term in the rhs of (A.3) tends to 0. Applying Lemma 2 with $r_j+1=n$, $q_j+1=\widehat{t}_{2,k}$ on the one hand and $r_j+1=n$, $q_j+1=(t^\star_{2,k}+t^\star_{2,k+1})/2$ on the other hand, we get that

$$\left|\sum_{j=t^\star_{2,k}}^{(t^\star_{2,k}+t^\star_{2,k+1})/2-1} Y_{n,j}-\sum_{j=t^\star_{2,k}}^{(t^\star_{2,k}+t^\star_{2,k+1})/2-1}\widehat{\mathcal{U}}_{n,j}\right|\leq\lambda_n.$$

Since $\widehat{t}_{2,k}\leq t^\star_{2,k}$, $\widehat{\mathcal{U}}_{n,j}=\widehat{\mu}_{K_1^\star+1,k+1}$ within the interval $[t^\star_{2,k},(t^\star_{2,k}+t^\star_{2,k+1})/2-1]$ and we get that

$$(t^\star_{2,k+1}-t^\star_{2,k})|\mu^\star_{K_1^\star+1,k+1}-\widehat{\mu}_{K_1^\star+1,k+1}|/2\leq\lambda_n+|\mathbf{E}(n,|[t^\star_{2,k},(t^\star_{2,k}+t^\star_{2,k+1})/2-1])|.$$

Therefore the second term in the rhs of (A.3) can be bounded by

$$\mathbb{P}\left(\lambda_n\geq(t^\star_{2,k+1}-t^\star_{2,k})|\mu^\star_{K_1^\star+1,k}-\mu^\star_{K_1^\star+1,k+1}|/12\right) \\ +\mathbb{P}\left((t^\star_{2,k+1}-t^\star_{2,k})^{-1}\left|\mathbf{E}(n,,|[t^\star_{2,k},(t^\star_{2,k}+t^\star_{2,k+1})/2-1])\right| \right. \\ \left.\geq|\mu^\star_{K_1^\star+1,k}-\mu^\star_{K_1^\star+1,k+1}|/6\right)$$

By Lemma 3 and (A3), (A2) and (A4), we get that both terms tend to zero as n tends to infinity. We thus get that $\mathbb{P}(A_{n,k}\cap C_n)\to 0$, as n tends to infinity.

Let us now prove that $\mathbb{P}(A_{n,k}\cap\overline{C_n})$ tend to 0, as n tends to infinity. Observe that

$$\mathbb{P}(A_{n,k}\cap\overline{C_n})=\mathbb{P}(A_{n,k}\cap D_n^{(\ell)})+\mathbb{P}(A_{n,k}\cap D_n^{(m)})+\mathbb{P}(A_{n,k}\cap D_n^{(r)}),$$

where

$$\begin{aligned}D_n^{(\ell)}&=\left\{\exists p\in\{1,\ldots,K^\star\},\ \widehat{t}_{2,p}\leq t^\star_{2,p-1}\right\}\cap\overline{C_n}, \\ D_n^{(m)}&=\left\{\forall k\in\{1,\ldots,K^\star\},\ t^\star_{2,k-1}<\widehat{t}_{2,k}<t^\star_{2,k+1}\right\}\cap\overline{C_n},\end{aligned}$$

$$D_n^{(r)} = \left\{\exists p \in \{1,\dots,K^\star\},\ \widehat{t}_{2,p} \geq t^\star_{2,p+1}\right\} \cap \overline{C_n}.$$

Using the same arguments as those used for proving that $\mathbb{P}(A_{n,k} \cap C_n) \to 0$, we can prove that $\mathbb{P}(A_{n,k} \cap D_n^{(m)}) \to 0$, as n tends to infinity. Let us now prove that $\mathbb{P}(A_{n,k} \cap D_n^{(\ell)}) \to 0$. Note that

$$\begin{aligned}\mathbb{P}(D_n^{(\ell)}) \quad \leq \quad & \sum_{k=1}^{K_2^\star - 1} \mathbb{P}(\{t^\star_{2,k} - \widehat{t}_{2,k} > I^\star_{\min}/2\} \cap \{\widehat{t}_{2,k+1} - t^\star_{2,k} > I^\star_{\min}/2\}) \\ & \qquad + \mathbb{P}(t^\star_{2,K_2^\star} - \widehat{t}_{2,K_2^\star} > I^\star_{\min}/2). \qquad (A.4)\end{aligned}$$

Applying (2.7) and (2.8) with $r_j + 1 = n$, $q_j + 1 = \widehat{t}_{2,k}$ on the one hand and $r_j + 1 = n$, $q_j + 1 = t^\star_{2,k}$ on the other hand, we get that

$$\left| \sum_{j=\widehat{t}_{2,k}}^{t^\star_{2,k}-1} Y_{n,j} - \sum_{j=\widehat{t}_{2,k}}^{t^\star_{2,k}-1} \widehat{\mathcal{U}}_{n,j} \right| \leq \lambda_n.$$

Thus,

$$\begin{aligned} & \mathbb{P}(\{t^\star_{2,k} - \widehat{t}_{2,k} > I^\star_{\min}/2\} \cap \{\widehat{t}_{2,k+1} - t^\star_{2,k} > I^\star_{\min}/2\}) \\ \leq \quad & \mathbb{P}(\lambda_n/(n\delta_n) \geq |\mu^\star_{K_1^\star+1,k} - \mu^\star_{K_1^\star+1,k+1}|/3) \\ & + \mathbb{P}(\{|\mu^\star_{K_1^\star+1,k} - \widehat{\mu}_{K_1^\star+1,k+1}| \geq |\mu^\star_{K_1^\star+1,k} - \mu^\star_{K_1^\star+1,k+1}|/3\} \\ & \qquad \cap \{\widehat{t}_{2,k+1} - t^\star_{2,k} > I^\star_{\min}/2\}) \\ & + \mathbb{P}(\{|\mathbf{E}(n; [\widehat{t}_{2,k}, t^\star_{2,k} - 1])|/(t^\star_{2,k} - \widehat{t}_{2,k}) \geq |\mu^\star_{K_1^\star+1,k} - \mu^\star_{K_1^\star+1,k+1}|/3\} \\ & \qquad \qquad \cap \{t^\star_{2,k} - \widehat{t}_{2,k} > I^\star_{\min}/2\}). \qquad (A.5)\end{aligned}$$

Using the same arguments as previously we get that the first and the third term in the rhs of (A.5) tend to zero as n tends to infinity. Let us now focus on the second term of the rhs of (A.5). Applying (2.7) and (2.8) with $r_j + 1 = n$, $q_j + 1 = \widehat{t}_{2,k+1}$ on the one hand and $r_j + 1 = n$, $q_j + 1 = t^\star_{2,k}$ on the other hand, we get that

$$\left| \sum_{j=t^\star_{2,k}}^{\widehat{t}_{2,k+1}-1} Y_{n,j} - \sum_{j=t^\star_{2,k}}^{\widehat{t}_{2,k+1}-1} \widehat{\mathcal{U}}_{n,j} \right| \leq \lambda_n.$$

Hence,

$$|(\mu^\star_{K_1^\star+1,k} - \widehat{\mu}_{K_1^\star+1,k+1})(\widehat{t}_{2,k+1} - t^\star_{2,k}) + \mathbf{E}(n, [t^\star_{2,k}; \widehat{t}_{2,k+1} - 1])| \leq \lambda_n.$$

The second term of the rhs of (A.5) is thus bounded by

$$\begin{aligned} & \mathbb{P}(\{\lambda_n(\widehat{t}_{2,k+1} - t^\star_{2,k})^{-1} \geq |\mu^\star_{K_1^\star+1,k} - \mu^\star_{K_1^\star+1,k+1}|/6\} \\ & \qquad \cap \{\widehat{t}_{2,k+1} - t^\star_{2,k} > I^\star_{\min}/2\}) \\ & + \mathbb{P}(\{(\widehat{t}_{2,k+1} - t^\star_{2,k})^{-1}|\mathbf{E}(n, [t^\star_{2,k}; \widehat{t}_{2,k+1} - 1])| \geq |\mu^\star_{K_1^\star+1,k} - \mu^\star_{K_1^\star+1,k+1}|/6\}\end{aligned}$$

$$\cap\{\widehat{t}_{2,k+1} - t^{\star}_{2,k} > I^{\star}_{\min}/2\}),$$

which tend to zero by Lemma 3, (A3), (A2) and (A4). It is thus proved that the first term in the rhs of (A.4) tends to zero as n tends to infinity. The same arguments can be used for addressing the second term in the rhs of (A.4) since $\widehat{t}_{2,K^{\star}_2+1} = n$ and hence $\widehat{t}_{2,K^{\star}_2+1} - t^{\star}_{2,K^{\star}_2} > I^{\star}_{\min}/2$.

Using similar arguments, we can prove that $\mathbb{P}(A_{n,k} \cap D^{(r)}_n) \to 0$, which concludes the proof of Proposition 1. □

A.2. Proofs of computational lemmas

Proof of Lemma 4. Consider $\mathcal{X}\mathbf{v}$ for instance (the same reasoning applies for $\mathcal{X}^{\top}\mathbf{v}$): we have $\mathcal{X}\mathbf{v} = (\mathbf{T} \otimes \mathbf{T})\mathbf{v} = \mathrm{Vec}(\mathbf{T}\mathbf{V}\mathbf{T}^{\top})$ where $\mathbf{V}$ is the $n \times n$ matrix such that $\mathrm{Vec}(\mathbf{V}) = \mathbf{v}$. Because of its triangular structure, $\mathbf{T}$ operates as a cumulative sum operator on the columns of $\mathbf{V}$. Hence, the computations for the jth column is done by induction in n operations. The total cost for the n columns of $\mathbf{TV}$ is thus n^2. Similarly, right multiplying a matrix by $\mathbf{T}^{\top}$ boils down to perform cumulative sums over the rows. The final cost for $\mathcal{X}\mathbf{v} = \mathrm{Vec}(\mathbf{T}\mathbf{V}\mathbf{T}^{\top})$ is thus $2n^2$ in case of a dense matrix $\mathbf{V}$, and possibly less when $\mathbf{V}$ is sparse. □

Proof of Lemma 5. Let $\mathcal{A} = \{a_1, \ldots, a_K\}$, then

$$\left(\mathcal{X}^{\top}\mathcal{X}\right)_{\mathcal{A},\mathcal{A}} = (\mathbf{T} \otimes \mathbf{T})^{\top}_{\bullet,\mathcal{A}}(\mathbf{T} \otimes \mathbf{T})_{\bullet,\mathcal{A}}, \tag{A.6}$$

where $(\mathbf{T} \otimes \mathbf{T})_{\bullet,\mathcal{A}}$ (resp. $(\mathbf{T} \otimes \mathbf{T})^{\top}_{\bullet,\mathcal{A}}$) denotes the columns (resp. the rows) of $\mathbf{T} \otimes \mathbf{T}$ lying in $\mathcal{A}$. For j in $\mathcal{A}$, let us consider the Euclidean division of $j-1$ by n given by: $(j-1) = nq_j + r_j$, then $(\mathbf{T} \otimes \mathbf{T})_{\bullet,j} = \mathbf{T}_{\bullet,q_j+1} \otimes \mathbf{T}_{\bullet,r_j+1}$. Hence, $(\mathbf{T} \otimes \mathbf{T})_{\bullet,\mathcal{A}}$ is a $n^2 \times K$ matrix defined by:

$$\begin{aligned}&(\mathbf{T} \otimes \mathbf{T})_{\bullet,\mathcal{A}}\\ &\quad = \left[\mathbf{T}_{\bullet,q_{a_1}+1} \otimes \mathbf{T}_{\bullet,r_{a_1}+1}; \mathbf{T}_{\bullet,q_{a_2}+1} \otimes \mathbf{T}_{\bullet,r_{a_2}+1}; \ldots; \mathbf{T}_{\bullet,q_{a_K}+1} \otimes \mathbf{T}_{\bullet,r_{a_K}+1}\right].\end{aligned}$$

Thus,

$$(\mathbf{T} \otimes \mathbf{T})_{\bullet,\mathcal{A}} = \mathbf{T}_{\bullet,Q_{\mathcal{A}}} * \mathbf{T}_{\bullet,R_{\mathcal{A}}}, \quad \text{where} \quad \begin{aligned} Q_{\mathcal{A}} &= \{q_{a_1}+1, \ldots, q_{a_K}+1\},\\ R_{\mathcal{A}} &= \{r_{a_1}+1, \ldots, r_{a_K}+1\}\end{aligned}$$

and $*$ denotes the Khatri-Rao product, which is defined as follows for two $n \times n$ matrices A and B

$$A * B = [a_1 \otimes b_1; a_2 \otimes b_2; \ldots; a_n \otimes b_n],$$

where the a_i (resp. b_i) are the columns of A (resp. B). Using (25) of Theorem 2 in [14], we get that

$$(\mathbf{T} \otimes \mathbf{T})^{\top}_{\bullet,\mathcal{A}}(\mathbf{T} \otimes \mathbf{T})_{\bullet,\mathcal{A}} = \left(\mathbf{T}^{\top}_{\bullet,Q_{\mathcal{A}}}\mathbf{T}_{\bullet,Q_{\mathcal{A}}}\right) \circ \left(\mathbf{T}^{\top}_{\bullet,R_{\mathcal{A}}}\mathbf{T}_{\bullet,R_{\mathcal{A}}}\right),$$

where $\circ$ denotes the Hadamard or entry-wise product. Observe that by definition of $\mathbf{T}$, $(\mathbf{T}_{\bullet,Q_{\mathcal{A}}}^{\top}\mathbf{T}_{\bullet,Q_{\mathcal{A}}})_{k,\ell} = n - (q_{a_k} \vee q_{a_\ell})$ and $(\mathbf{T}_{\bullet,R_{\mathcal{A}}}^{\top}\mathbf{T}_{\bullet,R_{\mathcal{A}}})_{k,\ell} = n - (r_{a_k} \vee r_{a_\ell})$. By (A.6), $\left(\mathcal{X}^{\top}\mathcal{X}\right)_{\mathcal{A},\mathcal{A}}$ is a Gram matrix which is positive and definite since the vectors $\mathbf{T}_{\bullet,q_{a_1}+1} \otimes \mathbf{T}_{\bullet,r_{a_1}+1}$, $\mathbf{T}_{\bullet,q_{a_2}+1} \otimes \mathbf{T}_{\bullet,r_{a_2}+1}$, $\ldots$, $\mathbf{T}_{\bullet,q_{a_K}+1} \otimes \mathbf{T}_{\bullet,r_{a_K}+1}$ are linearly independent. □

Proof of Lemma 6. The operations of adding/removing a column to a Cholesky factorization are classical and well treated in books of numerical analysis, see e.g. [8]. An advantage of our settings is that there is no additional computational cost for computing $\mathcal{X}^{\top}\mathcal{X}_{\bullet j}$ when entering a new variable j thanks to the closed-form expression (3.1). □

Acknowledgements

V. Brault would like to thank the French National Research Agency ANR, which partly supported this research through the ABS4NGS project (ANR-11-BINF-0001-06).

References

[1] Auger, I. E. and C. E. Lawrence (1989). Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology 51*(1), 39–54. MR0978902

[2] Bach, F., R. Jenatton, J. Mairal, and G. Obozinski (2012). Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning 4*(1), 1–106.

[3] Bellman, R. (1961). On the approximation of curves by line segments using dynamic programming. *Commun. ACM 4*(6), 284–286.

[4] Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees.* Statistics/Probability Series. Belmont, California, U.S.A.: Wadsworth Publishing Company.

[5] Dixon, J. R., S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren (2012). Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature 485*(7398), 376–380.

[6] Efron, B., T. Hastie, I. Johnstone, R. Tibshirani, et al. (2004). Least angle regression. *The Annals of statistics 32*(2), 407–499. MR2060166

[7] Fisher, W. D. (1958). On grouping for maximum homogeneity. *Journal of the American Statistical Association 53*(284), 789–798.

[8] Golub, G. H. and C. F. Van Loan (2012). *Matrix computations.* JHU Press. 3rd edition.

[9] Harchaoui, Z. and C. Lévy-Leduc (2010). Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association 105*(492), 1480–1493.

[10] Hoefling, H. (2010). A path algorithm for the fused lasso signal approximator. *J. Comput. Graph. Statist. 19*(4), 984–1006.

[11] Kay, S. (1993). *Fundamentals of statistical signal processing: detection theory*. Prentice-Hall, Inc.

[12] Lévy-Leduc, C., M. Delattre, T. Mary-Huard, and S. Robin (2014). Two-dimensional segmentation for analyzing hi-c data. *Bioinformatics 30*(17), i386–i392.

[13] Lieberman-Aiden, E., N. L. Van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, et al. (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science 326*(5950), 289–293.

[14] Liu, S. and G. Trenkler (2008). Hadamard, khatri-rao, kronecker and other matrix products. *Int. J. Inform. Syst. Sci. 4*, 160–177.

[15] Maidstone, R., T. Hocking, G. Rigaill, and P. Fearnhead (2016). On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 1–15. MR3599687

[16] Mairal, J. and B. Yu (2012). Complexity analysis of the lasso regularization path. In *Proceedings of the 29th ICML*.

[17] Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(4), 417–473. MR2758523

[18] Osborne, M. R., B. Presnell, and B. A. Turlach (2000). A new approach to variable selection in least squares problems. *IMA journal of numerical analysis 20*(3), 389–403.

[19] R Core Team (2015). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

[20] Rigaill, G. (2015). A pruned dynamic programming algorithm to recover the best segmentations with 1 to kmax change-points. *Journal de la Société Française de Statistique 156*(4), 180–205. MR3436653

[21] Sanderson, C. (2010). Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Technical report, NICTA.

[22] Scott, A. J. and M. Knott (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics 30*(3), 507–512.

[23] Tibshirani, R. J. and J. Taylor (2011). The solution path of the generalized lasso. *Ann. Statist. 39*(3), 1335–1371.

[24] Vert, J.-P. and K. Bleakley (2010). Fast detection of multiple change-points shared by many signals using group lars. In *Advances in Neural Information Processing Systems*, pp. 2343–2351.