Fast Detection of Block Boundaries in Block-Wise Constant Matrices

Vincent Brault, Julien Chiquet, Céline Lévy-Leduc

▶ **To cite this version:**

Vincent Brault, Julien Chiquet, Céline Lévy-Leduc. Fast Detection of Block Boundaries in Block-Wise Constant Matrices. 12th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM 2016), Petra Perner, Jul 2016, New York, NY, United States. pp.214-228, 10.1007/978-3-319-41920-6_16 . hal-01574362

HAL Id: hal-01574362
https://hal.science/hal-01574362

Submitted on 14 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Fast Detection of Block Boundaries in Block-Wise Constant Matrices

Vincent Brault(✉), Julien Chiquet, and Céline Lévy-Leduc

UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay, 75005 Paris, France
vincentbrault@agroparistech.fr

Abstract. We propose a novel approach for estimating the location of block boundaries (change-points) in a random matrix consisting of a block wise constant matrix observed in white noise. Our method consists in rephrasing this task as a variable selection issue. We use a penalized least-squares criterion with an ℓ_1-type penalty for dealing with this problem. We first provide some theoretical results ensuring the consistency of our change-point estimators. Then, we explain how to implement our method in a very efficient way. Finally, we provide some empirical evidence to support our claims and apply our approach to data coming from molecular biology which can be used for better understanding the structure of the chromatin.

Keywords: Change-points · High-dimensional sparse linear model · HiC experiments

1 Introduction

Detecting automatically the block boundaries in a block wise constant matrix corrupted with noise is a very important issue which may have several applications. One of the main situations in which this problem occurs is the detection of chromosomal regions having close spatial location in the nucleus. Detecting such regions will improve our understanding of the influence of the chromosomal conformation on the cells functioning. The data provided by the most recent technology called HiC consist of a list of pairs of locations along the chromosome which are often summarized as a square matrix such that each entry corresponds to the number of interactions between two positions along the chromosome, see [3]. Since this matrix can be modeled as a block wise matrix corrupted by some additional noise, it is of particular interest to design an efficient and fully automated method to find the block boundaries of large matrices, which may typically have several thousands of rows and columns, in order to identify the interacting chromosomal regions.

A large literature is dedicated to the change-point detection issue for one-dimensional data. This problem can be addressed from a sequential (online) [13] or from a retrospective (off-line) [2] point of view. Many off-line approaches are based on the dynamic programming algorithm which retrieves K change-points

© Springer International Publishing Switzerland 2016
P. Perner (Ed.): MLDM 2016, LNAI 9729, pp. 214–228, 2016.
DOI: 10.1007/978-3-319-41920-6_16

within n observations of a one-dimensional signal with a complexity of $O(Kn^2)$ in time [7]. Such a complexity is however prohibitive for dealing with very large data sets. In this situation, [5] proposed to rephrase the change-point estimation issue as a variable selection problem. This approach has also been extended by [15] to find shared change-points between several signals. To the best of our knowledge no method has been proposed for addressing the case of two-dimensional data where the number of rows and columns may be very large ($n \times n \approx 5000 \times 5000$, namely 2×10^7 observations). The only statistical approach proposed for retrieving the change-point positions in the two-dimensional framework is the one devised by [8] but it is limited to the case where the blockwise matrix is assumed to be blockwise constant on the diagonal and constant outside the diagonal blocks.

It has first to be noticed that the classical dynamic programming algorithm cannot be applied in such a framework since the Markov property does not hold anymore. Moreover, the group-lars approach of [15] cannot be used in this framework since it would only provide change-points in columns and not in rows. As for the generalized Lasso recently devised by [14] or the two dimensional fused Lasso of [6], they are very helpful for image denoising but do not give access to the change-point positions since they are not derived to provide a partitioning of a matrix in rectangular blocks.

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 dimensional 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. In Section 4, we provide experimental evidence of the relevance of our approach on synthetic and real data coming from molecular biology.

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 $\mathbf{t}_1^\star = (t_{1,1}^\star, \ldots, t_{1,K_1^\star}^\star)$ and $\mathbf{t}_2^\star = (t_{2,1}^\star, \ldots, t_{2,K_2^\star}^\star)$ from the random matrix $\mathbf{Y} = (Y_{i,j})_{1 \leq i,j \leq n}$ defined by

$$\mathbf{Y} = \mathbf{U} + \mathbf{E}, \tag{1}$$

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

$$U_{i,j} = \mu_{k,\ell}^\star \quad \text{if} \quad \begin{array}{l} t_{1,k-1}^\star \leq i \leq t_{1,k}^\star - 1 \\ \text{and } t_{2,\ell-1}^\star \leq j \leq t_{2,\ell}^\star - 1, \end{array}$$

with the convention $t_{1,0}^\star = t_{2,0}^\star = 1$ and $t_{1,K_1^\star+1}^\star = t_{2,K_2^\star+1}^\star = n+1$. An example of such a matrix $\mathbf{U}$ is displayed in Figure 1 (left). 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

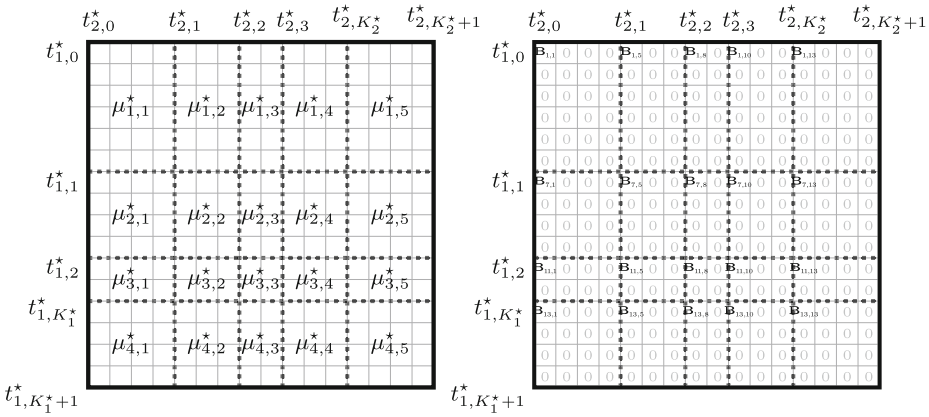

Fig. 1. Left: An example of a matrix $\mathbf{U}$ with $n = 16$, $K_1^\star = 3$ and $K_2^\star = 4$. Right: The matrix $\mathbf{B}$ associated to this matrix $\mathbf{U}$.

$Y_{i,j}$ are assumed to be independent random variables with a blockwise constant mean.

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 $(i,j) \in \{t^\star_{1,0}, \ldots, t^\star_{1,K_1^\star}\} \times \{t^\star_{2,0}, \ldots, t^\star_{2,K_2^\star}\}$. Then, (1) can be rewritten as follows:

$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{T}^\top + \mathbf{E}, \tag{2}$$

where $\mathbf{T}^\top$ denotes the transpose of the matrix $\mathbf{T}$. For an example of a matrix $\mathbf{B}$, see Figure 1 (right). 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) can be rewritten as:

$$\mathcal{Y} = \mathcal{X}\mathcal{B} + \mathcal{E}, \tag{3}$$

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 (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 (1) can thus be addressed as a variable selection problem:

$$\widehat{\mathcal{B}}(\lambda_n) = \underset{\mathcal{B} \in \mathbb{R}^{n^2}}{\mathrm{Argmin}} \left\{ \|\mathcal{Y} - \mathcal{X}\mathcal{B}\|_2^2 + \lambda_n \|\mathcal{B}\|_1 \right\}, \tag{4}$$

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 (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 $(t^\star_{1,k})_{1\le k\le K_1^\star}$ and of $(t^\star_{2,k})_{1\le k\le K_2^\star}$. More precisely, let us define by $\widehat{\mathcal{A}}(\lambda_n)$ the set of active variables:

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

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

$$\begin{aligned} \widehat{\mathbf{t}}_1 = (\widehat{t}_{1,k})_{1\le k\le |\widehat{\mathcal{A}}_1(\lambda_n)|} \in \{r_j + 1 : j \in \widehat{\mathcal{A}}(\lambda_n)\}, \\ \widehat{\mathbf{t}}_2 = (\widehat{t}_{2,\ell})_{1\le \ell\le |\widehat{\mathcal{A}}_2(\lambda_n)|} \in \{q_j + 1 : j \in \widehat{\mathcal{A}}(\lambda_n)\} \\ \text{where } \widehat{t}_{1,1} < \widehat{t}_{1,2} < \cdots < \widehat{t}_{1,|\widehat{\mathcal{A}}_1(\lambda_n)|}, \widehat{t}_{2,1} < \widehat{t}_{2,2} < \cdots < \widehat{t}_{2,|\widehat{\mathcal{A}}_2(\lambda_n)|}. \end{aligned} \quad (5)$$

In (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 (5), we shall use assumptions (**A1**–**A4**). These assumptions involve the two following quantities

$$\begin{aligned} I^\star_{\min} &= \min_{0\le k\le K_1^\star} |t^\star_{1,k+1} - t^\star_{1,k}| \wedge \min_{0\le k\le K_2^\star} |t^\star_{2,k+1} - t^\star_{2,k}|, \\ J^\star_{\min} &= \min_{1\le k\le K_1^\star, 1\le \ell\le K_2^\star+1} |\mu^\star_{k+1,\ell} - \mu^\star_{k,\ell}| \wedge \min_{1\le k\le K_1^\star+1, 1\le \ell\le K_2^\star} |\mu^\star_{k,\ell+1} - \mu^\star_{k,\ell}|, \end{aligned}$$

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

(**A2**) The sequence (λ_n) appearing in (4) is such that $(n\delta_n J^\star_{\min})^{-1}\lambda_n \to 0$, as n tends to infinity.

(**A3**) 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.

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

Proposition 1. *Let $(Y_{i,j})_{1\leq i,j\leq n}$ be defined by (1) and $\widehat{t}_{1,k}$, $\widehat{t}_{2,k}$ be defined by (5). Assume that $|\widehat{\mathcal{A}}_1(\lambda_n)| = K_1^\star$ and that $|\widehat{\mathcal{A}}_2(\lambda_n)| = K_2^\star$, with probabilty tending to one, then,*

$$\mathbb{P}\left(\left\{\max_{1\leq k\leq K_1^\star}\left|\widehat{t}_{1,k}-t_{1,k}^\star\right|\leq n\delta_n\right\}\cap\left\{\max_{1\leq k\leq K_2^\star}\left|\widehat{t}_{2,k}-t_{2,k}^\star\right|\leq n\delta_n\right\}\right)\underset{n\to\infty}{\longrightarrow}1. \quad (6)$$

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 (4). The second one allows us to control the supremum of the empirical mean of the noise.

Lemma 1. *Let $(Y_{i,j})_{1\leq i,j\leq n}$ be defined by (1). Then, $\widehat{\mathcal{U}} = \mathcal{X}\widehat{\mathcal{B}}$, where $\mathcal{X}$ and $\widehat{\mathcal{B}}$ are defined in (3) and (4) respectively, is such that*

$$\sum_{k=r_j+1}^{n}\sum_{\ell=q_j+1}^{n}Y_{k,\ell}-\sum_{k=r_j+1}^{n}\sum_{\ell=q_j+1}^{n}\widehat{\mathcal{U}}_{k,\ell}=\frac{\lambda_n}{2}sign(\widehat{\mathcal{B}}_j),\ \text{if } \widehat{\mathcal{B}}_j\neq 0, \quad (7)$$

$$\left|\sum_{k=r_j+1}^{n}\sum_{\ell=q_j+1}^{n}Y_{k,\ell}-\sum_{k=r_j+1}^{n}\sum_{\ell=q_j+1}^{n}\widehat{\mathcal{U}}_{k,\ell}\right|\leq\frac{\lambda_n}{2},\ \text{if } \widehat{\mathcal{B}}_j=0, \quad (8)$$

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 (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 (5).

Lemma 2. *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*

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

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

The proofs of Proposition 1, Lemmas 1 and 2 can be seen as a natural extension of the results of [5].

3 Implementation

In order to identify a series of change-points we look for the whole path of solutions in (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 [4,10]. Such an algorithm identifies in Problem (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, we show that the most involving computations in the LARS can be made extremely efficiently thanks to the particular structure of $\mathcal{X}$.

Lemma 3. *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 4. *Let $\mathcal{A} = \{a_1, \ldots, a_K\}$ 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*

$$\begin{aligned}&\left((\mathcal{X}^\top\mathcal{X})_{\mathcal{A},\mathcal{A}}\right)_{1\le k,\ell\le K}\\ &\quad = ((n - (q_{a_k} \vee q_{a_\ell})) \times (n - (r_{a_k} \vee r_{a_\ell})))_{1\le k,\ell\le K}\,. \end{aligned} \tag{9}$$

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

Lemma 5. *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 an $|\mathcal{A}|$-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 1. 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. For addressing any general problem though, we rather solve system 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 5.

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 $\hat{\mathcal{B}}(\lambda)$.

Algorithm 1. Fast LARS for two-dimensional change-point detection

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 3 ; // $\mathcal{O}(n^2)$

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 4; // $\mathcal{O}(|\mathcal{A}|^2)$

  // Compute the direction of descent

  Get the unormalized direction $\tilde{w}_\mathcal{A} \leftarrow \left(\mathcal{X}_{\cdot\mathcal{A}}^\top \mathcal{X}_{\cdot\mathcal{A}}\right)^{-1} \operatorname{sign}(\hat{c}_\mathcal{A})$; // $\mathcal{O}(|\mathcal{A}|^2)$

  Normalize $w_\mathcal{A} \leftarrow \alpha \tilde{w}_\mathcal{A}$ with $\alpha \leftarrow 1/\sqrt{\tilde{w}_\mathcal{A}^\top \operatorname{sign}(\hat{c}_\mathcal{A})}$;

  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 3; // $\mathcal{O}(n^2)$

  // Compute the direction step

  Find the maximal step preserving equicorrelation $\gamma_{\text{in}} \leftarrow \min_{j \in \mathcal{A}^c}^+ \left\{ \frac{\lambda - \mathbf{c}_j}{\alpha - a_j}, \frac{\lambda + \mathbf{c}_j}{\alpha + a_j} \right\}$;

  Find the maximal step preserving the signs $\gamma_{\text{out}} \leftarrow \min_{j \in \mathcal{A}}^+ \left\{ -\hat{\mathcal{B}}_\mathcal{A} / w_\mathcal{A} \right\}$;

  The direction step that preserves both is $\hat{\gamma} \leftarrow \min(\gamma_{\text{in}}, \gamma_{\text{out}})$;

  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 $\gamma_{\text{out}} < \gamma_{\text{in}}$ **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.

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 Numerical Experiments

4.1 Synthetic Data

The goal of this section is to assess the statistical and numerical performances of our methodology. We generated observations according to Model (1) where $\mathbf{U}$ is a symmetric blockwise constant matrix defined by

$$\left(\mu^\star_{k,\ell}\right)_{k \in \{1,\dots K_1^\star+1\}, \ell \in \{1,\dots K_2^\star+1\}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}, \tag{10}$$

and the $E_{i,j}$ are zero mean i.i.d. Gaussian random variables of variance σ^2 where σ is in $\{1, 2, 5\}$. Some examples of data generated from this model with $n = 500$, $K_1^\star = K_2^\star = 4$ can be found in Figure 2 (Top).

Statistical Performances. For each value of σ in $\{1, 2, 5\}$, we generated 1000 matrices following the model described in Section 4.1 with $(t^\star_{1,k})_{1\le k\le K^\star_1} = ([nk/(K^\star_1+1)]+1)_{1\le k\le K^\star_1}$ and $(t^\star_{2,k})_{1\le k\le K^\star_2} = ([nk/(K^\star_2+1)]+1)_{1\le k\le K^\star_2}$, where $[x]$ denotes the integer part of x and $K^\star_1 = K^\star_2 = 4$ and $n = 500$. Figure 2 (middle) displays the mean square error $n^{-2}\|\mathcal{B} - \widehat{\mathcal{B}}\|_2^2$ for the different samples (in gray) and the median of the mean square errors in thick as a function of the number of active variables s defined in Algorithm 1. We can see from this figure that even in the high noise level case, the mean square error is small. Moreover, the ROC curves displayed in the bottom part of Figure 2 ensure that the change-points in rows: $(t^\star_{1,k})_{1\le k\le K^\star_1}$ are properly retrieved with a very small error rate even in high noise level frameworks. The same results hold for the change-points in columns: $(t^\star_{2,k})_{1\le k\le K^\star_2}$ but are not displayed in order to save space.

Since, to the best of our knowledge, no two-dimensional method are available, we propose to compare our approach to an adaptation of the CART procedure of [1] and to an adaptation of [5] (HL) dedicated to univariate observations. We adapt the CART methodology by using the successive boundaries provided by CART as change-points for the two-dimensional data. The associated ROC curve is displayed with '•' in Figure 3. For adapting the HL methodology, we apply it to each row of the data matrix and for each λ, we obtain 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 (the associated ROC curve for this approach called HL1 is displayed with '+' in Figure 3) or if they appear in $([n/2]+1)$ rows (the associated ROC curve for this approach called HL2 is displayed with '□' in Figure 3). Since the procedures HL1 and HL2 are much slower than ours, the ROC curves are displayed for matrices of size 250×250. We can see from Figure 3 that our method outperforms the other ones.

Numerical Performances. We implemented Algorithm 1 in C++ using the library **armadillo** for linear algebra [12] and also provide an interface to the R platform [11] through the R package **blockseg** which is available from the Comprehensive R Archive Network (CRAN). All experiments were conducted on Linux workstation with Intel Xeon 2.4 GHz processor and 8 GB of memory.

We generated data as in Model (10) for different values of n: $n \in \{100, 250, 500, 1000, 2500, 5000\}$ and different values of the maximal number of activated variables: $s \in \{50, 100, 250, 500, 750\}$. The median runtimes obtained from 4 replications (+ 2 for warm-up) are reported in Figures 4. Left of Figure 4 (resp. right) 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.

Model Selection. In practice, we take $s = K^2_{\max}$ where $K_{\max}$ is an upper bound for $K^\star_1$ and $K^\star_2$. For choosing the final change-points we shall adapt the well-known *stability selection* approach devised by [9]. More precisely, we randomly

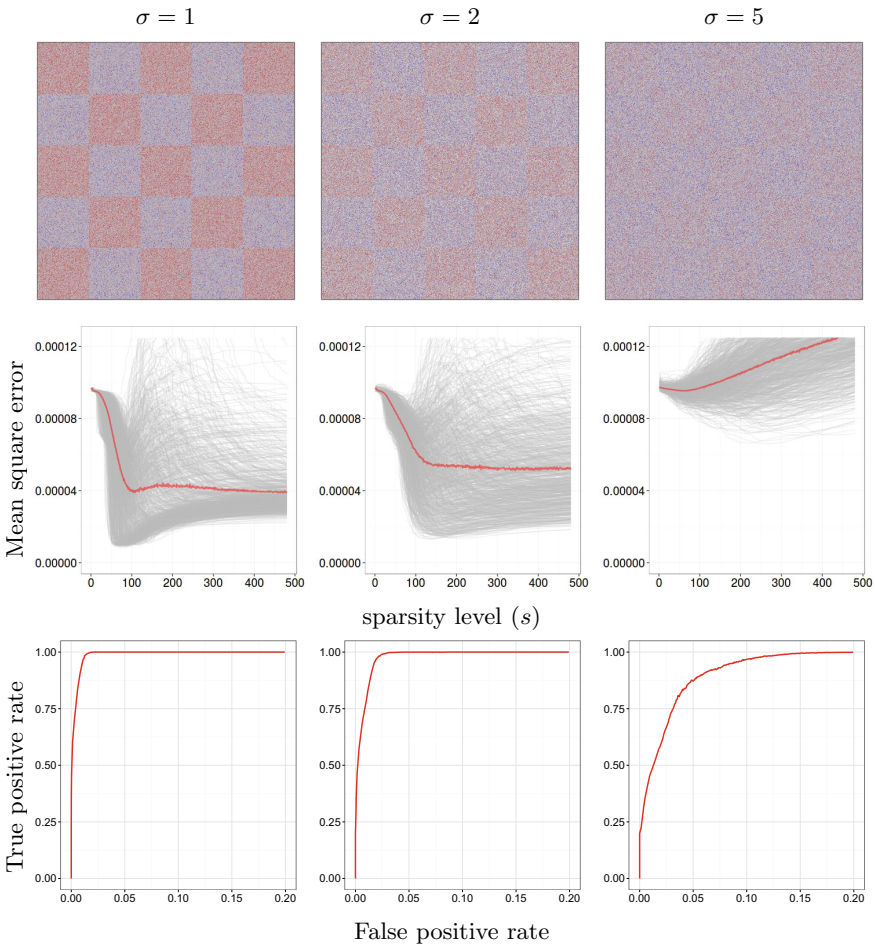

Fig. 2. Top: Examples of 500×500 matrices $\mathbf{Y}$ generated from the model described in Section 4.1. Middle: Mean square errors $n^{-2}\|\mathcal{B} - \widehat{\mathcal{B}}\|_2^2$ for the different realizations in gray and the median of the mean square errors in thick line as a function of the number of nonzero elements in $\widehat{\mathcal{B}}$ for each scenario. Bottom: ROC curves for the estimated change-points in rows.

choose M times $n/2$ columns and $n/2$ rows of the matrix $\mathbf{Y}$ and for each set of observations thus generated 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 (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 score associated with each change-point corresponds to the sum of the number of times it appears in each of the M resamplings.

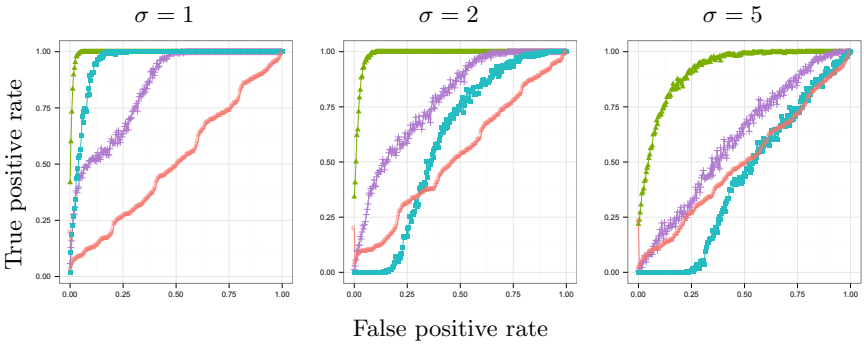

Fig. 3. ROC curves for the estimated change-points in rows for our method ('△'), HL1 ('+'), HL2 ('□') and CART ('•').

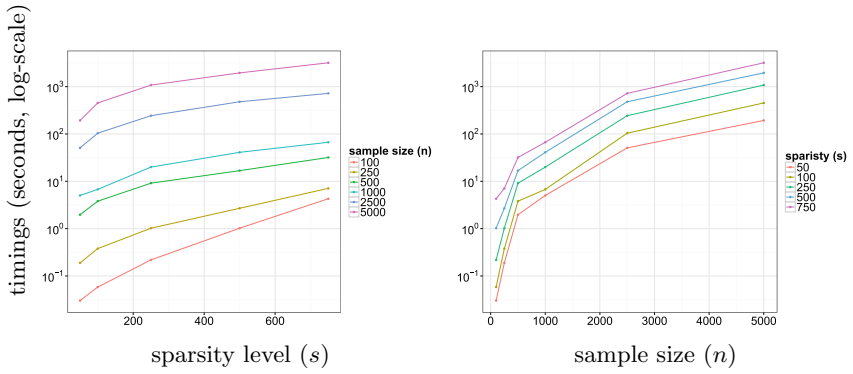

Fig. 4. Left: Computation time (in seconds) for various value of n as a function of the sparsity level $s = |\mathcal{A}|$ reached at the end of the algorithm. The cuvres for $n = 100$ to 5000 are displayed from bottom to top. Right: Computation time (in seconds) as a function of sample size n. The curves for $s = 50$ to 750 are displayed from bottom to top.

To evaluate the performances of this methodology, we generated observations according to the model defined in Section 4.1 with $s = 225$ and $M = 100$. The results are given in Figure 5 which displays the score associated to each change-point for a given matrix $\mathbf{Y}$ (top). We can see from the top part of Figure 5 some spurious change-points appearing near from 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 second and third rows of Figure 5. More precisely the boxplots associated to the estimation of $K_1^\star$ (resp. the histograms of the estimated change-points in rows) are displayed in the middle (resp. bottom) part of Figure 5 for different threshold (resp. when the threshold is equal to $T = 30\%$ of the largest score). We can see from these

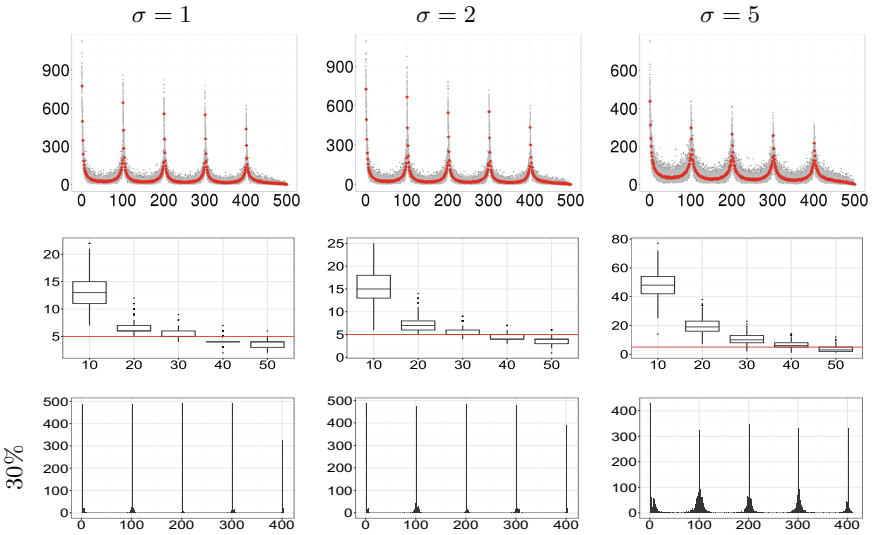

Fig. 5. Top: Scores associated to each estimated change-points for different values of σ; the true change-point positions in rows and columns are located at 101, 201, 301 and 401. Middle: Boxplots of the estimation of $K_1^\star$ for different values of σ and thresh after the post-processing step. The horizontal line corresponds to the true value of $K_1^\star$. Bottom: Histograms of the estimated change-points in rows for different values of σ after the post-processing step with thresh=30%.

figures that when thresh 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.

4.2 Application to HiC Data

In this section, we applied our methodology to publicly available data (http://chromosome.sdsc.edu/mouse/hi-c/download.html) already studied by [3]. More precisely, we studied 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 with our approach with those obtained by [3] on the same data since no ground truth is available. The matrices of these interaction matrices are displayed in Figure 6. We can see from this figure that modeling these matrices as block wise constant matrices corrupted with white seems to be relevant.

We display in Figure 7 the number of change-points in rows found by our approach as a function of the threshold thresh used in our adaptation of the stability selection approach presented in the previous section. We also display in this figure a red line corresponding to the number of change-points found by [3].

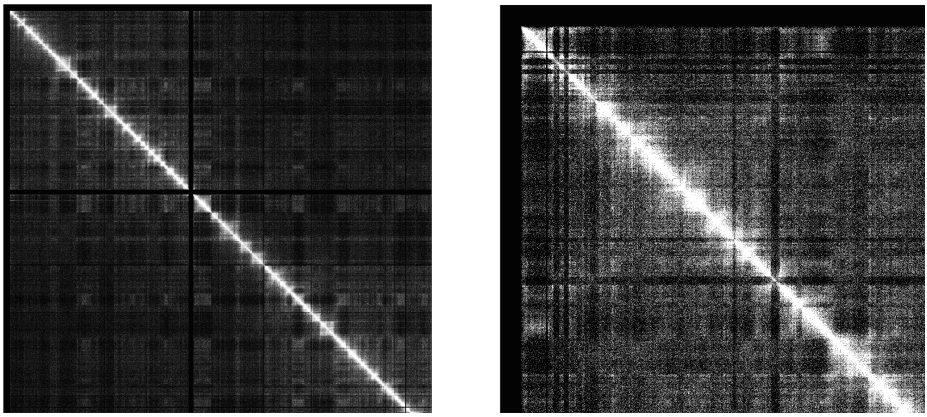

Fig. 6. Raw interaction matrices of Chromosome 1 (left) and Chromosome 19 (right) for the mouse cortex. The darkest entries correspond to the lowest values.

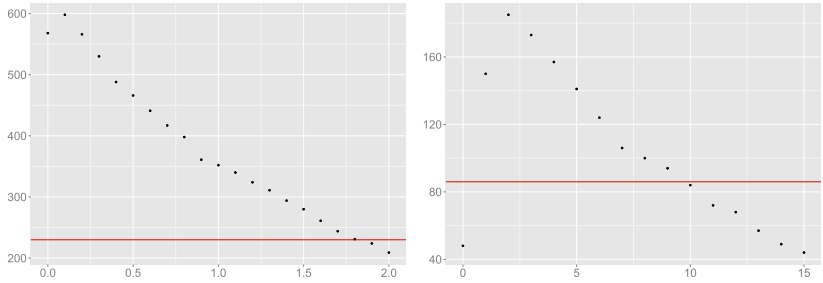

Fig. 7. Number of change-points in rows found by our approach as a function of the threshold ('•') in % for the interaction matrices of Chromosome 1 (left) and Chromosome 19 (right) of the mouse cortex. The horizontal line corresponds to the number of change-points found by [3].

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

$$d\left(\widehat{\boldsymbol{t}}_B, \widehat{\boldsymbol{t}}\right) = \max\left(d_1\left(\widehat{\boldsymbol{t}}_B, \widehat{\boldsymbol{t}}\right), d_2\left(\widehat{\boldsymbol{t}}_B, \widehat{\boldsymbol{t}}\right)\right), \tag{11}$$

where $\widehat{\boldsymbol{t}}$ and $\widehat{\boldsymbol{t}}_B$ are the change-points in rows found by our approach and [3], respectively. In (11),

$$d_1(\mathbf{a}, \mathbf{b}) = \sup_{b \in \mathbf{b}} \inf_{a \in \mathbf{a}} |a - b|, \tag{12}$$

$$d_2(\mathbf{a}, \mathbf{b}) = d_1(\mathbf{b}, \mathbf{a}). \tag{13}$$

More precisely, Figure 8 displays the boxplots of the d_1 and d_2 parts of the Hausdorff distance without taking the supremum in orange and blue, respectively.

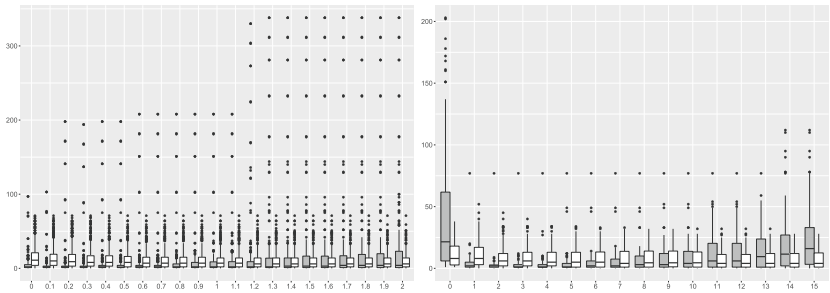

Fig. 8. Boxplots for the infimum parts of the Hausdorff distances d_1 and d_2 between the change-points found by [3] and our approach for the Chromosome 1 (left) and the Chromosome 19 (right) of the mouse cortex for the different thresholds in %. In each plot, the boxplot on the left corresponds to d_1 and the boxplot on the right corresponds to d_2.

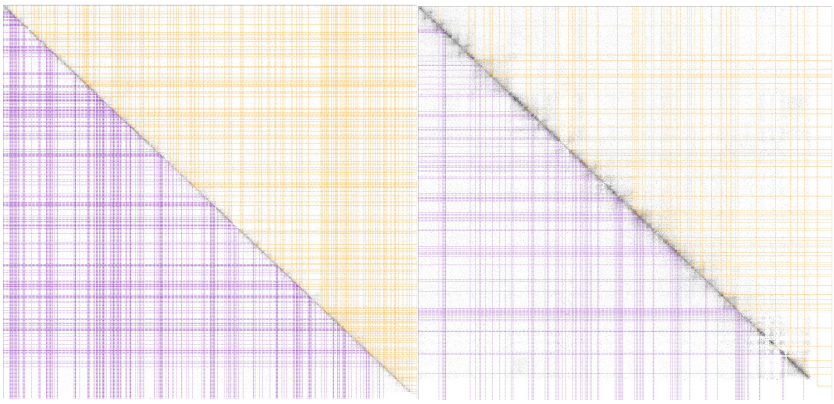

Fig. 9. Topological domains detected by [3] (upper triangular part of the matrix) and by our method (lower triangular part of the matrix) from the interaction matrix of Chromosome 1 (left) and Chromosome 19 (right) of the mouse cortex with a threshold giving 232 (resp. 85) estimated change-points in rows and columns.

We can observe from Figure 8 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).

In the case where the number of estimated change-points are on a par with those of [3], we can see from Figure 9 that the change-points found with our strategy present a lot of similarities with those found by the HMM based approach of [3].

Our method also gives access to the estimated change-point positions for different values of the thresholds. Figure 10 displays the different change-point locations that can be obtained for these different values of the threshold.

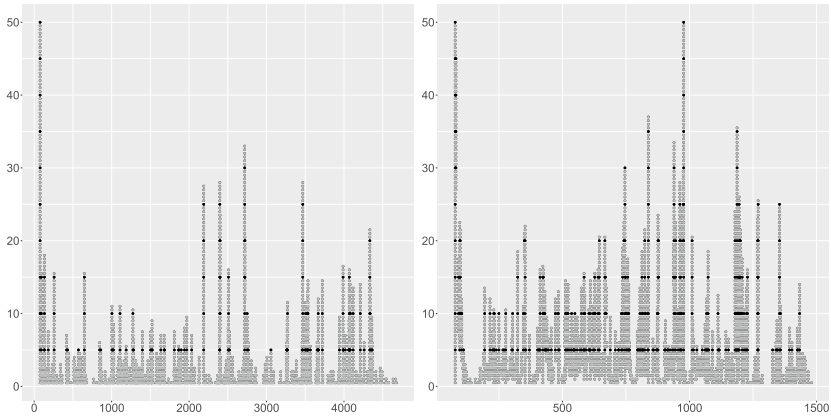

Fig. 10. Plots of the estimated change-points locations (x-axis) for different thresholds (y-axis) from 0.5% to 50% by 0.5%. The estimated change-point locations associated to threshold which are multiples of 5% are displayed with black points.

However, contrary to our method, the approach of [3] 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.

5 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 will be 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.

References

1. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Statistics/Probability Series. Wadsworth Publishing Company, Belmont (1984)
2. Brodsky, B., Darkhovsky, B.: Non-parametric statistical diagnosis: problems and methods. Kluwer Academic Publishers (2000)

3. Dixon, J.R., Selvaraj, S., Yue, F., Kim, A., Li, Y., Shen, Y., Hu, M., Liu, J.S., Ren, B.: Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature **485**(7398), 376–380 (2012)
4. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. The Annals of statistics **32**(2), 407–499 (2004)
5. Harchaoui, Z., Lévy-Leduc, C.: Multiple change-point estimation with a total variation penalty. Journal of the American Statistical Association **105**(492), 1480–1493 (2010)
6. Hoefling, H.: A path algorithm for the fused lasso signal approximator. J. Comput. Graph. Statist. **19**(4), 984–1006 (2010)
7. Kay, S.: Fundamentals of statistical signal processing: detection theory. Prentice-Hall, Inc. (1993)
8. Lévy-Leduc, C., Delattre, M., Mary-Huard, T., Robin, S.: Two-dimensional segmentation for analyzing hi-c data. Bioinformatics **30**(17), i386–i392 (2014)
9. Meinshausen, N., Bühlmann, P.: Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **72**(4), 417–473 (2010)
10. Osborne, M.R., Presnell, B., Turlach, B.A.: A new approach to variable selection in least squares problems. IMA Journal of Numerical Analysis **20**(3), 389–403 (2000)
11. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2015). http://www.R-project.org/
12. Sanderson, C.: Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Tech. rep, NICTA (2010)
13. Tartakovsky, A., Nikiforov, I., Basseville, M.: Sequential Analysis: Hypothesis Testing and Changepoint Detection. CRC Press, Taylor & Francis Group (2014)
14. Tibshirani, R.J., Taylor, J.: The solution path of the generalized lasso. Ann. Statist. **39**(3), 1335–1371 (2011)
15. Vert, J.P., Bleakley, K.: Fast detection of multiple change-points shared by many signals using group lars. In: Advances in Neural Information Processing Systems, pp. 2343–2351 (2010)