The group fused Lasso for multiple change-point detection

Kevin Bleakley, Jean-Philippe Vert

The group fused Lasso for multiple change-point detection

Kevin Bleakley
INRIA Saclay, Orsay, France
kevbleakley@gmail.com

Jean-Philippe Vert
Mines ParisTech CBIO, Fontainebleau, France
Institut Curie, Paris, France
INSERM U900, Paris, France
Jean-Philippe.Vert@mines.org

Abstract

We present the group fused Lasso for detection of multiple change-points shared by a set of co-occurring one-dimensional signals. Change-points are detected by approximating the original signals with a constraint on the multidimensional total variation, leading to piecewise-constant approximations. Fast algorithms are proposed to solve the resulting optimization problems, either exactly or approximately. Conditions are given for consistency of both algorithms as the number of signals increases, and empirical evidence is provided to support the results on simulated and array comparative genomic hybridization data.

1 Introduction

Finding the place (or time) where most or all of a set of one-dimensional signals (or *profiles*) jointly change in some specific way is an important question in several fields. A common situation is when we want to find change-points in a multidimensional signal, e.g., in audio and image processing [1, 2], to detect intrusion in computer networks [3, 4], or in financial and economics time series analysis [5]. Another important situation is when we are confronted with several 1-dimensional signals which we believe share common change-points, e.g., genomic profiles of a set of patients. The latter application is increasingly important in biology and medicine, in particular for the detection of copy-number variation along the genome [6], or the analysis of microarray and genetic linkage studies [7]. The common thread in biological applications is the search for data patterns shared by a set of individuals, such as cancer patients, at precise places on the genome; in particular, sudden changes in measured values. As opposed to the segmentation of multidimensional signals such as speech, where the dimension is fixed and collecting more data means having longer profiles, the length of signals in genomic studies (i.e., the number of probes measured along the genome) is fixed for a given technology while the number of signals (i.e., the number of individuals) can increase when we collect data about more patients. From a statistical point of view, it is therefore of interest to develop methods that identify multiple change-points shared by several signals that can benefit from increasing the number of signals.

There exists a vast literature on the change-point detection problem [8, 9]. Here we focus on computationally efficient methods to segment a multidimensional signal by approximating it with a piecewise-constant one, using quadratic error criteria. It is well-known that, in this case, the optimal segmentation of a p-dimensional signal of length n into k segments can be obtained in $O(n^2pk)$ by dynamic programming [10, 11, 12]. However, the quadratic complexity in n is prohibitive in applications such as genomics, where n can be in the order of 10^5 to 10^7 with current technology. An alternative to such *global* procedures, which estimate change-points as solutions of a global optimization problem, are fast *local* procedures such as binary segmentation [13], which detect breakpoints by iteratively applying a method for single change-point detection to the segments obtained after the previous change-point is detected. While such recursive methods can be extremely fast, in the order of $O(np\log(k))$ when the single change-point detector is $O(np)$, quality of segmentation is questionable when compared with global procedures [14].

For $p = 1$ (a single signal), an interesting alternative to these global and local procedures is to express the optimal segmentation as the solution of a convex optimization problem, using the (convex) total variation instead of the (non-convex) number of jumps to penalize a piecewise-constant function in order to approximate the original signal [15, 16]. The resulting piecewise-constant approximation of the signal, defined as the global minimum of the objective function, benefits from theoretical guaranties in terms of correctly detecting change-points [17, 18], and can be implemented efficiently in $O(nk)$ or $O(n\log(n))$ [19, 17, 20].

In this paper we propose an extension of total-variation based methods for single signals to the multidimensional setting, in order to approximate a multidimensional signal with a piecewise-constant signal with multiple change-points. We define the approximation as the solution of a convex optimization problem which involves a quadratic approximation error penalized by the sum of the Euclidean norms of the multidimensional increments of the function. The problem can be reformulated as a group Lasso [21], which we show how to solve exactly and efficiently. Alternatively, we provide an approximate yet often computationally faster solution to the problem using a group LARS procedure [21]. In the latter case, using the particular structure of the design matrix, we can find the first k change-points in $O(npk)$, thus extending the method of [17] to the multidimensional setting.

Unlike most previous theoretical investigations of change-point methods (e.g., [17, 18]), we are not interested in the case where the dimension p is fixed and the length of the profiles n increases, but in the opposite situation where n is fixed and p increases. Indeed, this corresponds to the case in genomics where, for example, n would be the fixed number of probes used to measure a signal along the genome, and p the number of samples or patients analyzed. We want to design a method that benefits from increasing p in order to identify shared change-points, even though the signal-to-noise ratio may be very low within each signal. As a first step towards this question, we give conditions under which our method is able to consistently identify a single change-point as p increases. We also show by simulation that the method is able to correctly identify multiple change-points as $p \to +\infty$, validating its relevance in practical settings.

The paper is organized as follows. After fixing notation in Section 2, we present the group fused Lasso method in Section 3. We propose two efficient algorithms to solve it in Section 4, and discuss its theoretical properties in Section 5. Lastly, we provide an empirical evaluation of the method and a comparison with other methods in the study of copy number variations in cancer in Section 6. A preliminary version of this paper was published in [22].

2 Notation

For any two integers $u \leq v$, we denote by $[u, v]$ the interval $\{u, u+1, \ldots, v\}$. For any $u \times v$ matrix M we note $M_{i,j}$ its (i,j)-th entry, and $\|M\| = \sqrt{\sum_{i=1}^{u}\sum_{j=1}^{v} M_{i,j}^2}$ its Frobenius norm (or Euclidean norm in the case of vectors). For any subsets of indices $A = \left(a_1, \ldots, a_{|A|}\right) \in [1, u]^{|A|}$ and $B = \left(b_1, \ldots, b_{|B|}\right) \in [1, v]^{|B|}$, we denote by $M_{A,B}$ the $|A| \times |B|$ matrix with entries M_{a_i,b_j} for $(i,j) \in [1, |A|] \times [1, |B|]$. For simplicity we will use $\bullet$ instead of $[1, u]$ or $[1, v]$, i.e., $A_{i,\bullet}$ is the i-th row of A and $A_{\bullet,j}$ is the j-th column of A. We note $\mathbf{1}_{u,v}$ the $u \times v$ matrix of ones, and $\mathbf{I}_p$ the $p \times p$ identity matrix.

3 Formulation

We consider p real-valued profiles of length n, stored in an $n \times p$ matrix Y. The i-th profile $Y_{\bullet,i} = (Y_{1,i}, \ldots, Y_{n,i})$ is the i-th column of Y. We model each profile as a piecewise-constant signal corrupted by noise, and assume that change-point locations tend to be shared across profiles. Our goal is to detect these shared change-points, and benefit from the possibly large number p of profiles to increase the statistical power of change-point detection.

3.1 Segmentation with a total variation penalty

When $p = 1$ (a single profile), a popular method to find change-points in a signal is to approximate it by a piecewise-constant function using a quadratic error criterion, i.e., to solve

$$\min_{U \in \mathbb{R}^n} \| Y - U \|^2 \quad \text{subject to} \quad \sum_{i=1}^{n-1} \delta(U_{i+1} - U_i) \leq k \,, \tag{1}$$

where δ is the Dirac function, equal to 0 if its argument is null, 1 otherwise. In other words, (1) expresses the best approximation of Y by a piecewise-constant profile U with at most k jumps. It is well-known that (1) can be solved in $O(n^2 k)$ by dynamic programming [10, 11, 12]. Although very fast when n is of moderate size, the quadratic dependency in n renders it impractical in current computers when n reaches millions or more, which is often the case in many application such as segmentation of genomic profiles.

An alternative to the combinatorial optimization problem (1) is to relax it to a convex optimization problem, by replacing the number of jumps by the convex total variation (TV) [15], i.e., to consider:

$$\min_{U \in \mathbb{R}^n} \frac{1}{2} \| Y - U \|^2 + \lambda \sum_{i=1}^{n-1} | U_{i+1} - U_i | \,. \tag{2}$$

For a given $\lambda > 0$, the solution $U \in \mathbb{R}^n$ of (2) is again piecewise-constant. Recent work has shown that (2) can be solved much more efficiently than (1): [19] proposed a fast coordinate descent-like method, [17] showed how to find the first k change-points iteratively in $O(nk)$, and [20] proposed a $O(n \ln(n))$ method to find all change-points. Adding penalties proportional to the ℓ_1 or ℓ_2 norm of U to (2) does not change the position of the change-points detected [16, 23], and the capacity of TV denoising to correctly identify change-points when n increases has been investigated in [17, 18].

Here, we propose to generalize TV denoising to multiple profiles by considering the following convex optimization problem, for $Y \in \mathbb{R}^{n \times p}$:

$$\min_{U \in \mathbb{R}^{n \times p}} \frac{1}{2} \| Y - U \|^2 + \lambda \sum_{i=1}^{n-1} \| U_{i+1,\bullet} - U_{i,\bullet} \| \,. \tag{3}$$

The second term in (3) can be considered a multidimensional TV: it penalizes the sum of Euclidean norms of the increments of U, seen as a time-dependent multidimensional vector, and reduces to the classical 1-dimensional TV when $p = 1$. Intuitively, when λ increases, this penalty will enforce many increment vectors $U_{i+1,\bullet} - U_{i,\bullet}$ to collapse to 0, just like the total variation in (2) in the case of 1-dimensional signals. This implies that the positions of non-zero increments will be the same for all profiles. As a result, the solution to (3) provides an approximation of the profiles Y by an $n \times p$ matrix of piecewise-constant profiles U which share change-points.

While (3) is a natural multidimensional generalization of the classical TV denoising method (2), we more generally investigate the following variant:

$$\min_{U \in \mathbb{R}^{n \times p}} \frac{1}{2} \| Y - U \|^2 + \lambda \sum_{i=1}^{n-1} \frac{\| U_{i+1,\bullet} - U_{i,\bullet} \|}{d_i} \,, \tag{4}$$

where $(d_i)_{i=1,\ldots,n-1}$ are position-dependant weights which affect the penalization of the jump differently at different positions. While (4) boils down to (3) for uniform weights $d_i = 1$, $i = 1, \ldots, n-1$, we will see that the unweighted version suffers from boundary effects and that position-dependent schemes such as:

$$\forall i \in [1, n-1], \quad d_i = \sqrt{\frac{n}{i(n-i)}} \,, \tag{5}$$

are both theoretically and empirically better choices.

To illustrate the grouping effect of the penalty in (4), Figure 1 compares the segmentation of three simulated profiles obtained with and without enforced sharing of change-points across profiles. We simulated three piecewise-constant signals corrupted by independent additive Gaussian noise. All profiles have length 500 and share the same 5 change-points, though with different amplitudes, at positions 38, 139, 268, 320 and 397. On the left-hand side, we show the first 5 change-points captured by TV denoising with weights (5) applied to each signal independently. On the right, we show the first 5 change-points captured by formulation (4). We see that the latter formulation finds the correct change-points, whereas treating each profile independently leads to errors. For example, the first two change-points have a small amplitude in the second profile and are therefore very difficult to detect from the profile only, while they are very apparent in the first and third profiles.

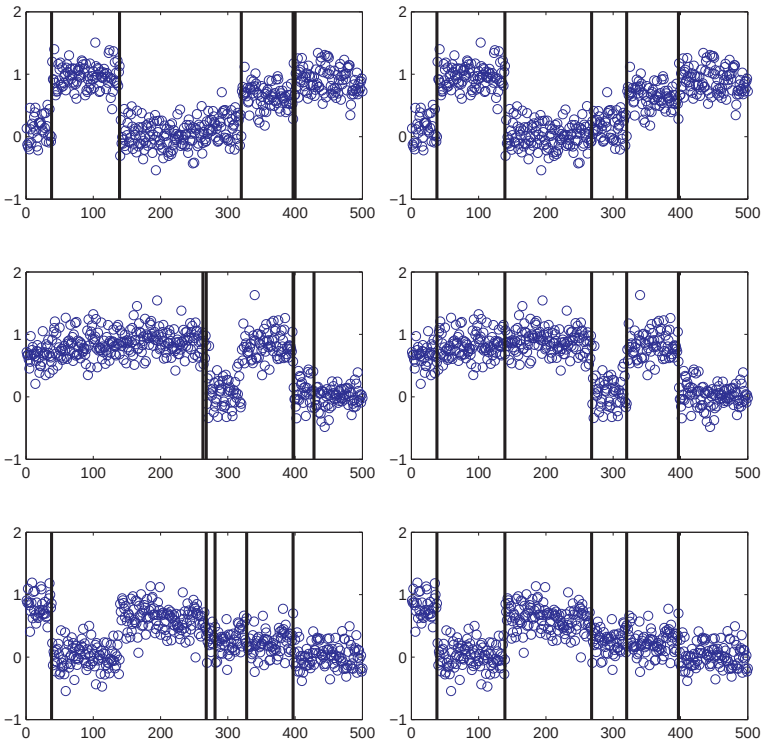

Figure 1: First 5 change-points detected on three simulated profiles by TV denoising of each profile (left) and by joint TV denoising (right).

3.2 Reformulation as a group Lasso problem

It is well-known that the 1-dimensional TV denoising problem (2) can be reformulated as a Lasso regression problem by an appropriate change of variable [17]. We now show that our generalization (4) can be reformulated as a group Lasso regression problem, which will be convenient for theoretical analysis and implementation [21]. To this end, we make the change of variables $(\beta, \gamma) \in \mathbb{R}^{(n-1)\times p} \times \mathbb{R}^{1\times p}$ given by:

$$\begin{aligned} \gamma &= U_{1,\bullet}\,, \\ \beta_{i,\bullet} &= \frac{U_{i+1,\bullet} - U_{i,\bullet}}{d_i} \quad \text{for } i = 1, \ldots, n-1\,. \end{aligned}$$

In other words $d_i\beta_{i,j}$ is the jump between the i-th and the $(i+1)$-th positions of the j-th profile. We immediately get an expression for U as a function of β and γ:

$$\begin{aligned} U_{1,\bullet} &= \gamma\,, \\ U_{i,\bullet} &= \gamma + \sum_{j=1}^{i-1} d_j \beta_{j,\bullet} \quad \text{for } i = 2, \ldots, n\,. \end{aligned}$$

This can be rewritten in matrix form as

$$U = \mathbf{1}_{n,1}\gamma + X\beta\,, \tag{6}$$

where X is the $n \times (n-1)$ matrix with entries $X_{i,j} = d_j$ for $i > j$, and 0 otherwise. Making this change of variable, we can re-express (4) as follows:

$$\min_{\beta \in \mathbb{R}^{(n-1)\times p}, \gamma \in \mathbb{R}^{1\times p}} \frac{1}{2} \| Y - X\beta - \mathbf{1}_{n,1}\gamma \|^2 + \lambda \sum_{i=1}^{n-1} \| \beta_{i,\bullet} \|\,. \tag{7}$$

For any $\beta \in \mathbb{R}^{(n-1)\times p}$, the minimum in γ is attained with $\gamma = \mathbf{1}_{1,n}(Y - X\beta)/n$. Plugging this into (7), we get that the matrix of jumps β is solution of

$$\min_{\beta \in \mathbb{R}^{(n-1)\times p}} \frac{1}{2} \| \bar{Y} - \bar{X}\beta \|^2 + \lambda \sum_{i=1}^{n-1} \| \beta_{i,\bullet} \|\,, \tag{8}$$

where $\bar{Y}$ and $\bar{X}$ are obtained from Y and X by centering each column.

Equation (8) is now a classical group Lasso regression problem [21], with a specific design matrix $\bar{X}$ and groups of features corresponding to the rows of the matrix β. The solution β of (8) is related to the solution U of our initial problem (4) by equation (6).

4 Implementation

Although (4) and (8) are convex optimization problems that can in principle be solved by general-purpose solvers [24], we want to be able to work in dimensions that reach millions or more, making this computationally difficult. In particular, the design matrix $\bar{X}$ in (8) is a non-sparse matrix of size $n \times (n-1)$, and cannot even fit in a computer's memory when n is large. Moreover, we would ideally like to obtain solutions for various values of λ, corresponding to various numbers of change-points, in order to be able to select the optimal number of change-points using statistical criteria. In the single profile case ($p = 1$), fast implementations in $O(nk)$ or $O(n \ln n)$ have been proposed [19, 17, 20]. However, none of these methods is applicable directly to the $p > 1$ setting since they all rely on specific properties of the $p = 1$ case, such as the fact that the solution is piecewise-affine in λ and that the set of change-points is monotically decreasing with λ.

In this section we propose two algorithms to respectively *exactly* or *approximately* solve (4) efficiently. We adopt the algorithms suggested by [21] to solve the group Lasso problem (8) and show how they can be implemented very efficiently in our case due to the particular structure of the regression problem. We have placed in Annex A several technical lemmas which show how to efficiently perform several operations with the given design matrix $\bar{X}$ that will be used repeatedly in the implementations proposed below.

4.1 Exact solution by block coordinate descent

A first possibility to solve the group Lasso problem (8) is to follow a block coordinate descent approach, where each group is optimized in turn with all other groups fixed. It can be shown that this strategy

converges to the global optimum, and is reported to be stable and efficient [21, 25]. As shown by [21], it amounts to iteratively applying the following equation to each block $i = 1, \ldots, n-1$ in turn, until convergence:

$$\beta_{i,\bullet} \leftarrow \frac{1}{\gamma_i}\left(1 - \frac{\lambda}{\| S_i \|}\right)_+ S_i \,, \tag{9}$$

where $\gamma_i = \| \bar{X}_{\bullet,i} \|^2 = i(n-i)d_i^2/n$ and $S_i = \bar{X}_{\bullet,i}^\top \left(\bar{Y} - \bar{X}\beta^{-i}\right)$, and where β^{-i} denotes the $(n-1)\times p$ matrix equal to β except for the i-th row $\beta^{-i}_{i,\bullet} = 0$. The convergence of the procedure can be monitored by the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} -\bar{X}_{\bullet,i}^\top \left(\bar{Y} - \bar{X}\beta\right) + \frac{\lambda\beta_{i,\bullet}}{\| \beta_{i,\bullet} \|} = 0 \quad \forall \beta_{i,\bullet} \neq 0 \,, \\ \| -\bar{X}_{\bullet,i}^\top \left(\bar{Y} - \bar{X}\beta\right) \| \leq \lambda \quad \forall \beta_{i,\bullet} = 0 \,. \end{aligned} \tag{10}$$

Since the number of blocks n can be very large and we expect only a fraction of non-zero blocks at the optimum (corresponding to the change-points), we implemented this block coordinate descent with an active set strategy. In brief, a set of active groups $\mathcal{A}$ corresponding to non-zero groups is maintained, and the algorithm alternates between optimizing β over the active groups in $\mathcal{A}$ and updating $\mathcal{A}$ by adding or removing groups based on violation of the KKT conditions. The resulting pseudo-code is shown in Algorithm 1. The inner loop (lines 3-7) corresponds to the optimization of β on the current active groups, using iteratively block coordinate descent (9). After convergence, groups that have been shrunk to 0 are removed from the active set (line 8), and the KKT conditions are checked outside of the active set (lines 9-10). If they are not fulfilled, the group that most violates the conditions is added to the active set (line 11), otherwise the current solution satisfies all KKT conditions and is therefore the global optimum (line 13).

Although it is difficult to estimate the number of iterations needed to reach convergence for a certain level of precision, we note that by Lemma 5 (Annex A), computation of $\bar{X}^\top\bar{Y}$ in line 1 can be done in $O(np)$, and each group optimization iteration (lines 3-7) requires computing $\bar{X}_{\bullet,i}^\top X_{\bullet,\mathcal{A}}$ (line 5), done in $O(|\mathcal{A}|)$ (see Lemma 6 in Annex A), then computing S_i (line 5) in $O(|\mathcal{A}|p)$ and soft-thresholding (line 6) in $O(p)$. The overall complexity of each group optimization iteration is therefore $O(|\mathcal{A}|p)$. Since each group in $\mathcal{A}$ must typically be optimized several times, we expect complexity that is at least quadratic in $|\mathcal{A}|$ and linear in p for each optimization over an active set $\mathcal{A}$ (lines 3-7). To check optimality of a solution after optimization over an active set $\mathcal{A}$, we need to compute $\bar{X}^\top\bar{X}\beta$ (line 9) which takes $O(np)$ (see Lemma 7, Annex A). Although it is difficult to upper bound the number of iterations needed to optimize over $\mathcal{A}$, this shows that a best-case complexity to find k change-points, if we correctly add groups one by one to the active set, would be $O(npk)$ to check k times the KKT conditions and find the next group to add, and $O(pk^3)$ in total if each optimization over an active set $\mathcal{A}$ is in $O(p|\mathcal{A}|^2)$. In Section 6, we provide some empirical results on the behavior of this block coordinate descent strategy.

4.2 Group fused LARS implementation

Since exactly solving the group Lasso with the method described in Section 4.1 can be computationally intensive, it may be of interest to find fast, approximate solutions to (8). We propose to implement a strategy based on the group LARS, proposed in [21] as a good way to approximately find the regularization path of the group Lasso. More precisely, the group LARS approximates the solution path of (8) with a piecewise-affine set of solutions and iteratively finds change-points. The resulting algorithm is presented here as Algorithm 2, and is intended to approximately solve (8). Change-points are added one by one (lines 4 and 8), and for a given set of change-points the solution moves straight along a descent direction (line 6) with a given step (line 7) until a new change-point is added (line 8). We refer to [21] for more details and justification for this algorithm.

While the original group LARS method requires storage and manipulation of the design matrix [21], implausible for large n here, we can again benefit from the computational tricks provided in Annex A

Algorithm 1 Block coordinate descent algorithm

Require: centered data $\bar{Y}$, regularization parameter λ.

1: Initialize $\mathcal{A} \leftarrow \emptyset$, $\beta = 0$, $C \leftarrow \bar{X}^\top \bar{Y}$.
2: **loop**
3: **repeat**
4: Pick $i \in \mathcal{A}$.
5: Compute $S_i \leftarrow C_{i,\bullet} - \bar{X}_{\bullet,i}^\top \bar{X} \beta^{-i}$.
6: Update $\beta_{i,\bullet}$ according to (9).
7: **until** convergence
8: Remove inactive groups: $\mathcal{A} \leftarrow \mathcal{A} \backslash \{i \in \mathcal{A} : \beta_{i,\bullet} = 0\}$.
9: Check KKT: $S \leftarrow C - \bar{X}^\top \bar{X} \beta$.
10: $\hat{u} \leftarrow \operatorname{argmax}_{i \notin A} \| S_{i,\bullet} \|^2$, $M = \| S_{\hat{u},\bullet} \|^2$.
11: **if** $M > \lambda^2$ **then**
12: Add a new group: $\mathcal{A} \leftarrow \mathcal{A} \cup \{\hat{u}\}$.
13: **else**
14: **return** β.
15: **end if**
16: **end loop**

to efficiently run the fast group LARS method. Computing $\bar{X}^\top \bar{Y}$ in line 1 can be done in $O(np)$ using Lemma 5. To compute the descent direction (line 6), we first compute w in $O(|\mathcal{A}|p)$ using Lemma 8, then a in $O(np)$ using Lemma 7. To find the descent step (line 7), we need to solve n polynomial equations of degree 2, the coefficients of which are computed in $O(p)$, resulting in a $O(np)$ complexity. Overall the main loop for each new change-point (lines 2–10) takes $O(np)$ in computation and memory, resulting in $O(npk)$ complexity in time and $O(np)$ in memory to find the first k change-points. We provide in Section 6 empirical results that confirm this theoretical complexity.

Algorithm 2 Group fused LARS algorithm

Require: centered data $\bar{Y}$, number of breakpoints k.

Initialize $\mathcal{A} \leftarrow \emptyset$, $\hat{c} \leftarrow \bar{X}^\top \bar{Y}$.
for $i = 1$ to k **do**
 if i=1 **then**
 First change-point : $\hat{u} \leftarrow \operatorname{argmin}_{j \in [1, n-1]} \| \hat{c}_{j,\bullet} \|$, $\mathcal{A} \leftarrow \{\hat{u}\}$.
 end if
 Descent direction: compute $w \leftarrow \left(\bar{X}_{\bullet,\mathcal{A}}^\top \bar{X}_{\bullet,\mathcal{A}} \right)^{-1} \hat{c}_{\mathcal{A},\bullet}$, then $a = \bar{X}^\top \bar{X}_{\mathcal{A}} w$.
 Descent step: for each $u \in [1, n-1] \setminus \mathcal{A}$, find if it exists the smallest positive solution α_u of the second-order polynomial in α:

$$\| \hat{c}_{u,\bullet} - \alpha a_{u,\bullet} \|^2 = \| \hat{c}_{v,\bullet} - \alpha a_{v,\bullet} \|^2 ,$$

 where v is any element of $\mathcal{A}$.
 Next change-point: $\hat{u} \leftarrow \operatorname{argmin}_{j \in [1, n-1]} \| \hat{c}_{j,\bullet} \|$, $\mathcal{A} \leftarrow \mathcal{A} \cup \{\hat{u}\}$.
 Update $\hat{c} \leftarrow \hat{c} - \alpha_{\hat{u}} a$.
end for

5 Theoretical analysis

In this section, we study theoretically to what extent the estimator (4) recovers correct change-points. The vast majority of existing theoretical results for offline segmentation and change-point detection con-

sider the setting where p is fixed (usually $p = 1$), and n increases (e.g., [2]). This typically corresponds to cases where we can sample a continuous signal with increasing density, and wish to locate more precisely the underlying change-points as the density increases.

We propose a radically different analysis, motivated notably by applications in genomics. Here, the length of profiles n is fixed for a given technology, but the number of profiles p can increase when more samples or patients are collected. The property we would like to study is then, for a given change-point detection method, to what extent increasing p for fixed n allows us to locate more precisely the change-points. While this simply translates our intuition that increasing the number of profiles should increase the statistical power of change-point detection, and while this property was empirically observed in [7], we are not aware of previous theoretical results in this setting. In particular we are interested in the consistency of our method, in the sense that it should correctly detect the true change-points if enough samples are available.

5.1 Consistent estimation of a single change-point

As a first step towards the analysis of this "fixed n increasing p" setting, let us assume that the observed centered profiles $\bar{Y}$ are obtained by adding noise to a set of profiles with a *single* shared change-point between positions u and $u + 1$, for some $u \in [1, n-1]$. In other words, we assume that

$$\bar{Y} = \bar{X}\beta^* + W\,,$$

where β^* is an $(n-1) \times p$ matrix of zeros except for the u-th row $\beta^*_{u,\bullet}$, and W is a noise matrix whose entries are assumed to be independent and identically distributed with respect to a centered Gaussian distribution with variance σ^2. In this section we study the probability that the first change-point found by our procedure is the correct one, when p increases. We therefore consider an infinite sequence of jumps $\left(\beta^*_{u,i}\right)_{i\geq 1}$, and letting $\bar{\beta}_p^2 = \frac{1}{p}\sum_{i=1}^{p}(\beta^*_{u,i})^2$, we assume that $\bar{\beta}^2 = \lim_{p\to\infty}\bar{\beta}_p^2$ exists and is finite. We first characterize the first selected change-point as p increases.

Lemma 1. *Assume, without loss of generality, that $u \geq n/2$, and let, for $i \in [1, n-1]$,*

$$G_i = d_i^2\frac{i(n-i)}{n}\sigma^2 + \frac{\bar{\beta}^2 d_i^2 d_u^2}{n^2} \times \begin{cases} i^2\,(n-u)^2 & \text{if } i \leq u\,, \\ u^2\,(n-i)^2 & \text{otherwise.} \end{cases} \tag{11}$$

When $p \to +\infty$, the first change-point selected by the group fused Lasso (4) is in $\operatorname{argmax}_{i\in[1,n-1]} G_i$ with probability tending to 1.

Proof of this result is given in Annex B. From it we easily deduce conditions under which the first change-point is correctly found with increasing probability as p increases. Let us first focus on the unweighted group fused Lasso (3), corresponding to the setting $d_i = 1$ for $i = 1, \ldots, n-1$.

Theorem 2. *Let $\alpha = u/n$ be the position of the change-point scaled in the interval $[0, 1]$, and*

$$\tilde{\sigma}_\alpha^2 = n\bar{\beta}^2\frac{(1-\alpha)^2(\alpha - \frac{1}{2n})}{\alpha - \frac{1}{2} - \frac{1}{2n}}\,. \tag{12}$$

If $\sigma^2 < \tilde{\sigma}_\alpha^2$, the probability that the first change-point selected by the unweighted group fused Lasso (3) is the correct one tends to 1 as $p \to +\infty$. When $\sigma^2 > \tilde{\sigma}_\alpha^2$, it is not the correct one with probability tending to 1.

This theorem, the proof of which can be found in Annex C, deserves several comments.

- To detect a change-point at position $u = \alpha n$, the noise level σ^2 must not be larger than the critical value $\tilde{\sigma}_\alpha^2$ given by (12), hence the method is not consistent for all positions. $\tilde{\sigma}_\alpha^2$ decreases monotonically from $\alpha = 1/2$ to 1, meaning that change-points near the boundary are more difficult to

detect correctly than change-points near the center. The most difficult change-point is the last one ($u = n - 1$) which can only be detected consistently if σ^2 is smaller than

$$\bar{\sigma}^2_{1-1/n} = \frac{2\bar{\beta}^2}{n} + o(n^{-1}).$$

- For a given level of noise σ^2, change-point detection is asymptotically correct for any $\alpha \in [\epsilon, 1-\epsilon]$, where ϵ satisfies $\sigma^2 = \tilde{\sigma}^2_{1-\epsilon}$, i.e.,

$$\epsilon = \sqrt{\frac{\sigma^2}{2n\bar{\beta}^2}} + o(n^{-1/2})\,.$$

  This shows in particular that increasing the profile length n increases the relative interval (as a fraction of n) where change-points are correctly identified, and that we can get as close as we want to the boundary for n large enough.

- When $\sigma^2 < \tilde{\sigma}^2_\alpha$, the correct change-point is found consistently when p increases, showing the benefit of the accumulation of many profiles.

Theorem 2 shows that the unweighted group fused Lasso (3) suffers from boundary effects, since it may not correctly identify a single change-points near the boundary is the noise is too large. In fact, Lemma 1 tells us that if we miss the correct change-point position, it is because we estimate it more towards the middle of the interval (see proof of Theorem 2 for details). The larger the noise, the more biased the procedure is. We now show that this issue can be fixed when we consider the weighted group fused Lasso (4) with well-chosen weights.

Theorem 3. *The weighted group fused Lasso (4) with weights given by (5) correctly finds the first change-point at any position with probability tending to* 1 *as* $p \to +\infty$.

The proof of Theorem 3 is postponed to Annex D. It shows that the weighting scheme (5) cancels the effect of the noise and allows us to consistently estimate any change-point, independently of its position in the signal, as the number of signals increases.

5.2 Consistent estimation of a single change-point with fluctuating position

An interesting variant of the problem of detecting a change-point common to many profiles is that of detecting a change-point with similar location in many profiles, allowing fluctuations in the precise location of the change-point. This can be modeled by assuming that the profiles are random, and that the i-th profile has a single change-point of value β_i at position U_i, where $(\beta_i, U_i)_{i=1,\dots,p}$ are independent and identically distributed according to a distribution $P = P_\beta \otimes P_U$ (i.e., we assume β_i independent from U_i). We denote $\bar{\beta}^2 = E_{P_\beta}\beta^2$ and $p_i = P_U(U = i)$ for $i \in [1, n-1]$. Assuming that the support of P_U is $[a, b]$ with $1 \le a \le b \le n-1$, the following result extends Theorem 2 by showing that the first change-point discovered by the unweighted group fused Lasso is in the support of P_U under some condition on the noise level, while the weighted group fused Lasso correctly identifies a change-point in the support of P_U asymptotically without conditions on the noise.

Theorem 4. *1. Let* $\alpha = U/n$ *be the random position of the change-point on* $[0,1]$ *and* $\alpha_m = a/n$ *and* $\alpha_M = b/n$ *the position of the left and right boundaries of the support of* P_U *scaled to* $[0,1]$. *If* $1/2 \in (\alpha_m, \alpha_M)$, *then for any noise level* σ^2, *the probability that the first change-point selected by the unweighted group fused Lasso (3) is in the support of* P_U *tends to* 1 *as* $p \to +\infty$. *If* $1/2 < \alpha_m$ *or* $\alpha_M < 1/2$, *let*

$$\tilde{\sigma}^2_{P_U} = n\bar{\beta}^2\left[(1-E\alpha)^2 + \mathit{var}(\alpha)^2\right] \times \begin{cases} \frac{\alpha_m - \frac{1}{2n}}{\alpha_m - \frac{1}{2} - \frac{1}{2n}} & \text{if } \alpha_m > \frac{1}{2}\,, \\ \frac{1 - \frac{1}{2n} - \alpha_M}{\frac{1}{2} - \alpha_M - \frac{1}{2n}} & \text{if } \alpha_M < \frac{1}{2}\,. \end{cases} \tag{13}$$

The probability that the first selected change-point is in the support of P_U tends to 1 *when* $\sigma^2 < \tilde{\sigma}^2_{P_U}$. *When* $\sigma^2 > \tilde{\sigma}^2_{P_U}$, *it is outside of the support of* P_U *with probability tending to* 1.

2. *The weighted group fused Lasso (4) with weights given by (5) finds the first change-point in the support of* P_U *with probability tending to* 1 *as* $p \to +\infty$, *independently of* σ^2 *and of the support of* P_U.

This theorem, the proof of which is postponed to Annex E, illustrates the robustness of the method to fluctuations in the precise position of the change-point shared between profiles. Although this situation rarely occurs when we are considering classical multidimensional signals such as financial time series or video signals, it is likely to be the rule when we consider profiles coming from different biological samples, where for example we can expect frequent genomic alterations at the vicinity of important oncogenes or tumor suppressor genes. Although the theorem only gives a condition on the noise level to ensure that the selected change-point lies in the support of the distribution of change-point locations, a precise estimate of the location of the selected change-point as a function of P_U, which generalizes Lemma 1, is given in the proof.

5.3 The case of multiple change-points

While the theoretical results presented above focus on the detection of a single change-point, the real interest of the method is to estimate multiple change-points. The extension of Theorem 2 to this setting is, however, not straightforward and we postpone it for future efforts. We conjecture that the group fused Lasso estimator can, under certain conditions, consistently estimate multiple change-points. More precisely, in order to generalize the proof of Theorem 2, we must analyze the path of the vectors $(\hat{c}_{i,\bullet})$, and check that, for some λ in (3) or (4), they reach their maximum norm precisely at the true change-points. The situation is more complicated than in the single change-point case since, in order to fulfill the KKT optimality conditions, the vectors $(\hat{c}_{i,\bullet})$ must hit a hypersphere at each correct change-point, and must remain strictly within the hypersphere between consecutive change-points. This can probably be ensured if the noise level is not too high (like in the single change-point case), and if the positions corresponding to successive change-points on the hypersphere are far enough from each other, which could be ensured if two successive change-points are not too close to each other, and are in sufficiently different directions. Although the weighting scheme (5) ensures consistent estimation of the first change-point independently of the noise level, it may however not be sufficient to ensure consistent estimation of subsequent change-points.

Although we propose no theoretical results besides these conjectures for the case of multiple change-points, we provide experimental results below that confirm that, when the noise is not too large, we can indeed correctly identify several change-points, with probability of success increasing to 1 as p increases.

5.4 Estimating the number of change-points

The number of change-points detected by the group fused Lasso in the multidimensional signal depends on the choice of λ in (3) and (4). In practice, we propose the following scheme in order to estimate a segmentation and the number of change-points. We try to select a λ that over-segments the multidimensional signal, that is, finds more change-points that we would normally expect for the given type of signal or application. Then, on the set of k change-points found, we perform post-processing using a simple least-squares criteria. Briefly, for each given subset of $k' \leq k$ change-points, we approximate each signal between successive change-points with the mean value of the points in that interval; then, we calculate the total sum of squared errors (SSE) between the set of real signals and these piecewise-constant approximations to them. Though it may appear computationally intensive or even impossible to do this for all subsets of $k' \leq k$ change-points, a dynamic programming strategy (e.g., [6]) means that the best subset of $k' \leq k$ change-points can be calculated for all $k' \in \{1, \ldots, k\}$ in $O(k^3)$.

It then remains to choose the "best" $k' \in \{1, \ldots, k\}$ using, for example, a model-selection strategy. The optimal SSE for $k'+1$ (which we may call $SSE(k'+1)$ to ease notation), will be smaller than

$SSE(k')$ but at a certain point, adding a further change-point will have no physical reality, it only improves the SSE due to random noise. Here, we implemented a method proposed in [26, 6] where we first normalize the SSE for $k' = 1, \ldots, k$ into a score $J(k')$ such that $J(1) = k$ and $J(k) = 1$, in such a way the $J(k')$ has an average slope of -1 between 1 and k; we then try to detect a kink in the curve by calculating the discrete second derivative of $J(k')$, and selecting the k' after which this second derivative no longer rises above a fixed threshold (typically 0.5).

6 Experiments

In this section we test the group fused Lasso on several simulated and real data sets. All experiments were run under Linux on a machine with two 4-core Intel Xeon 3.16GHz processors and a total of 16Gb of RAM. We have implemented the group fused Lasso in MATLAB; the package `GFLseg` is available for download[1].

6.1 Speed trials

In a first series of experiments, we tested the behavior of the group fused Lasso in terms of computational efficiency. We simulated multidimensional profiles with various lengths n between 2^4 and 2^{23}, various dimensions p between 1 and 2^{15}, and various number of shared change-points k between 1 and 2^7. In each case, we first ran the iterative weighted group fused LARS (Section 4.2) to detect successive change-points, and recorded the corresponding λ values. We then ran the exact group fused Lasso implementation by block coordinate descent (Section 4.1) on the same λ values. Figure 2 shows speed with respect to increasing one of p, n and k while keeping the other two variables fixed, for both implementations. The axes are $\log$-$\log$, so the slope gives the exponent of the complexity (resp. n, p and k). For the weighted group fused LARS, linearity is clearest for k, whereas for n and p, the curves are initially sub-linear, then slightly super-linear for extremely large values of n and p. As these time trials reach out to the practical limits of current technology, we see that this is not critical - on average, even the longest trials here took less than 200 seconds. The weighted fused group Lasso results are perhaps more interesting, as it is harder to predict in advance the practical time performance of the algorithm. Surprisingly, when increasing n (p and k fixed) or increasing p (n and k fixed), the group fused Lasso eventually becomes as fast the iterative, deterministic group fused LARS. This suggests that at the limits of current technology, if k is small (say, less than 10), the potentially superior performance of the Lasso version (see later) may not even be punished by a slower run-time with respect to the LARS version. We suggest that this may be due to the Lasso optimization problem becoming relatively "easier" to solve when n or p increases, as we observed that the Lasso algorithm converged quickly to its final set of change-points. The main difference between the Lasso and LARS performance appears when the number of change-points increases: with respective empirical complexities cubic and linear in k, as predicted by the theoretical analysis, Lasso is already 1,000 times slower than LARS when we seek 100 change-points.

6.2 Accuracy for detection of a single change-point

Next, we tested empirically the accuracy the group fused Lasso for detecting a single change-point. We first generated multidimensional profiles of dimension p, with a single jump of height 1 at a position u, for different values of p and u. We added to the signals an i.i.d. Gaussian noise with variance $\tilde{\sigma}_\alpha^2 = 10.78$, the critical value corresponding to $\alpha = 0.8$ in Theorem 2. We ran 1000 trials for each value of u and p, and recorded how often the group fused Lasso with or without weights correctly identified the change-point. According to Theorem 2, we expect that, for the unweighted group fused Lasso, for $50 \leq u < 80$ there is convergence in accuracy to 1 when p increases, and for $u > 80$, convergence in accuracy to zero. This is indeed what is seen in Figure 3 (left panel), with $u = 80$ the limit case between

[1] Available at `http://cbio.ensmp.fr/GFLseg`

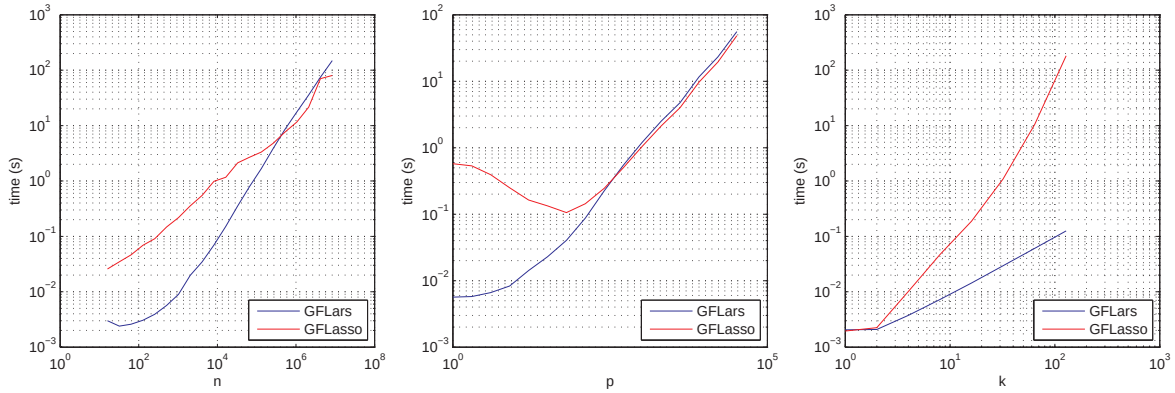

Figure 2: **Speed trials for group fused LARS (top row) and Lasso (bottom row).** *Left column:* varying n, with fixed $p = 10$ and $k = 10$; *center column:* varying p, with fixed $n = 1000$ and $k = 10$; *right column:* varying k, with fixed $n = 1000$ and $p = 10$. Figure axes are log-log. Results are averaged over 100 trials.

the two different modes of convergence. The center panel of Figure 3 shows that when the default weights (5) are added, convergence in accuracy to 1 occurs across all u, as predicted by Theorem 3. In

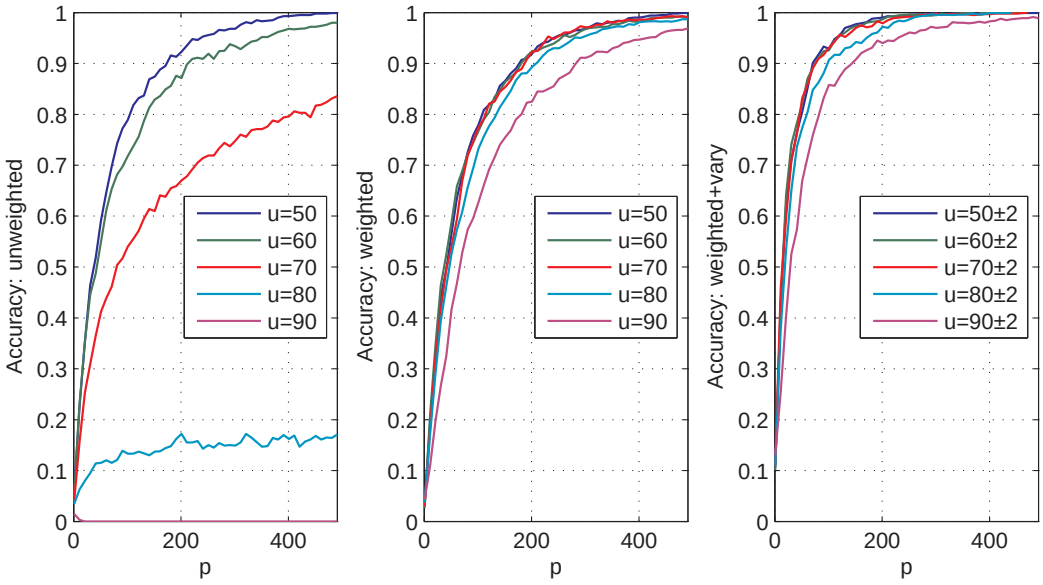

Figure 3: **Single change-point accuracy for the group fused Lasso.** Accuracy as a function of the number of profiles p when the change-point is placed in a variety of positions $u = 50$ to $u = 90$ (left and centre plots, resp. unweighted and weighted group fused Lasso), or: $u = 50 \pm 2$ to $u = 90 \pm 2$ (right plot, weighted with varying change-point location), for a signal of length 100.

addition, the right-hand-side panel of Figure 3 shows results for the same trials except that change-point locations can vary uniformly in the interval $u \pm 2$. We see that, as predicted by Theorem 4, the accuracy of the weighted group fused Lasso remains robust against fluctuations in the exact change-point location.

6.3 Accuracy for detecting multiple change-points

To investigate the potential for extending the results to the case of many shared change-points, we further simulated profiles of length $n = 100$ with a change-point at *all* of positions $10, 20, \ldots, 90$. We

consider dimensions p between 1 and 500. Jumps at each change-point of each profile were drawn from a Gaussian with mean 0 and variance 1; we then added centered Gaussian noise with $\sigma^2 \in \{0.05, 0.2, 1\}$ to each position in each profile. For each value of p and σ^2, we ran one hundred trials of both implementations, with or without weights, and recorded the accuracy of each method, defined as the percentage of trials where the first 9 change-points detected by the method are exactly the 9 true change-points. Results are presented in Figure 4 (from left to right, resp. $\sigma^2 = 0.05, 0.2, 1$). Clearly, the group fused Lasso outperforms the group fused LARS, and the weighted version of each algorithm outperforms the unweighted version. Although the group LARS is usually considered a reliable alternative to the exact group Lasso [21], this experiment shows that the exact optimization by block coordinate descent may be worth the computational burden if one is interested in accurate group selection. It also demonstrates that, as we conjectured in Section 5.3, the group fused Lasso can consistently estimate multiple change-points as the number of profiles increases.

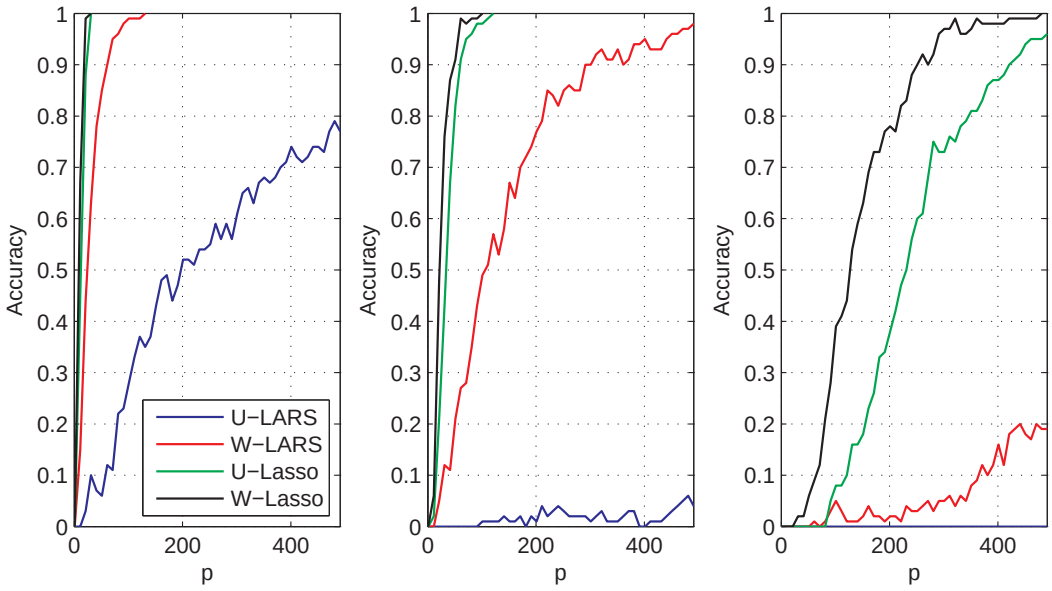

Figure 4: **Multiple change-point accuracy.** Accuracy as a function of the number of profiles p when change-points are placed at the nine positions $\{10, 20, \ldots, 90\}$ and the variance σ^2 of the centered Gaussian noise is either 0.05 (left), 0.2 (center) and 1 (right). The profile length is 100.

6.4 Application to gain and loss detection

We now consider a possible application of our method for the detection of regions with frequent gains (positive values) and losses (negative values) among a set of DNA copy number profiles, measured by array comparative genomic hybridization (aCGH) technology [27]. We propose a two-step strategy for this purpose: first, find an adequate joint segmentation of the signals; then, check the presence of gain or loss on each interval of the segmentation by summarizing each profile by its average value on the interval. Note that we do not assume that all profiles share exactly the same change-points, but merely see the joint segmentation as an adaptive way to reduce the dimension and remove noise from data.

In practice, we used group fused LARS on each chromosome to identify a set of 100 candidate change-points, and selected a subset of them by post-processing as described in Section 5.4. Then, in each piecewise-constant interval between successive shared change-points, we calculate the mean of the positive segments (shown in green in Figures 5(a) and 6(c)) and the mean of the negative segments (shown in red). The larger the mean of the positive segments, the more likely we are to believe that a region harbors an important common gain; the reasoning is analogous for important common losses and

the mean of the negative segments. Obviously, many other statistical tests could be carried out to detect frequent gains and losses on each segment, once the joint segmentation is performed.

We compare this method for detecting regions of gain and loss with the state-of-the-art H-HMM method [27], which has been shown to outperform several other methods in this setting. As [27] have provided their algorithm online with several of their data sets tested in their article, we implemented our method and theirs (H-HMM) on their benchmark data sets.

In the first data set in [27], the goal is to recover two regions – one amplified, one deleted, that are shared in 8 short profiles, though only 6 of the profiles exhibit each of the amplified or deleted regions. Performance is measured by area under ROC curve (AUC), following [27]. Running H-HMM with the default parameters, we obtained an AUC (averaged over 10 trials) of $0.96 \pm .01$, taking on average 60.20 seconds. The weighted group fused LARS, asked to select 100 breakpoints and followed by dynamic programming, took 0.06 seconds and had an AUC of 0.97. Thus, the performance of both methods was similar, though weighted group fused LARS was around 1000 times faster.

The second data set was a cohort of lung cancer cell lines originally published in [28, 29]. As in [27], we concentrated on the 18 NSCLC adenocarcinoma (NA) cell lines. Figure 5 shows the score statistics obtained on Chromosome 8 when using either weighted group fused LARS or H-HMM. Weighted group fused LARS first selected 100 candidate change-points per chromosome, then followed optimization of the number of change-points by dynamic programming, took in total 4.7 seconds and finally selected 260 change-points. In contrast, H-HMM took 38 minutes (100 iterations, as given in the code provided by the authors). The H-HMM scores should look like those shown in Figure 4 (top panel) of [27]; the difference is either due to the stochastic nature of the algorithm or using a different number of iterations than given in the sample code by the authors. In any case, at the MYC locus (near 13×10^7 bp), both methods strongly suggest a common gained region. However, the supposed advantage of H-HMM to very sparsely predict common gains and losses is not clear here; for example, it gives high common gain confidence to several fairly large genomic regions between 9 and 14 $\times 10^7$ bp.

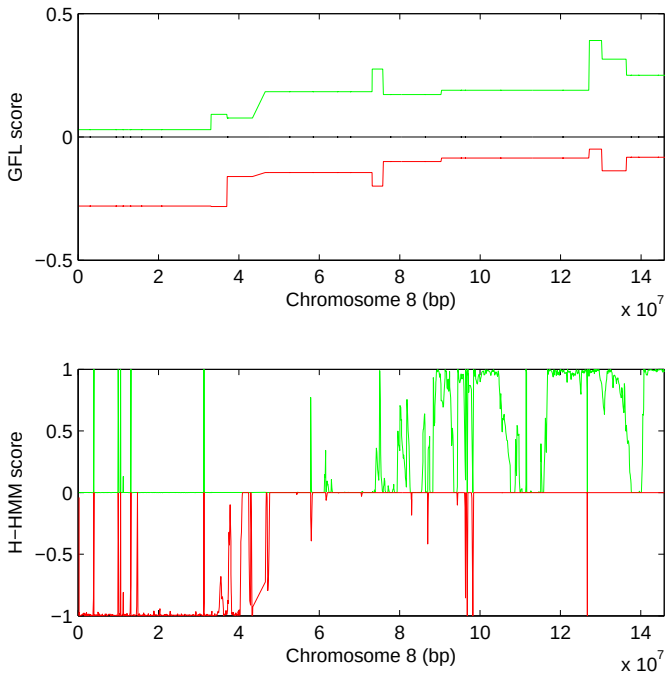

Figure 5: **Joint scores for a set of 18 NSCLC adenocarcinoma cell lines.** 5(a) using weighted group fused LARS; 5(b) using H-HMM with the actual code provided by [27].

6.5 Application to bladder tumor aCGH profiles

We further considered a publicly available aCGH data set of 57 bladder tumor samples [30]. Each aCGH profile gave the relative quantity of DNA for 2215 probes. We removed the probes corresponding to sex chromosomes because the sex mismatch between some patients and the reference made the computation of copy number less reliable, giving us a final list of 2143 probes.

Results are shown in Figure 6. 97 change-points were selected by the weighted group fused LARS; this took 1.1 seconds (Figure 6(c)). The H-HMM method (Figure 6(d)) took 13 minutes for 200 iterations (after 100 iterations convergence had not occured). We used the comprehensive catalogue of common

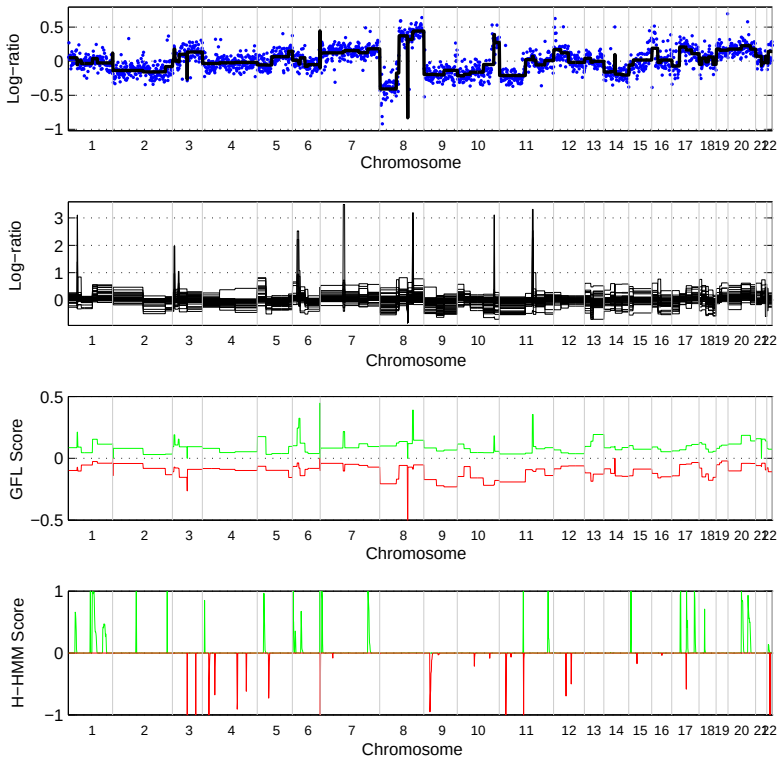

Figure 6: **Bladder cancer profiles.** 6(a) shows one of the original 57 profiles and its associated smoothed version. 6(b) shows the result of superimposing the smoothed versions of the 57 bladder tumor aCGH profiles obtained using weighted group fused LARS followed by dimension-selection. 6(c) shows the result of transforming the set of smoothed outputs into "scores" for amplification/deletion (see Section 6.4) and 6(d) the corresponding output for the H-HMM method [27]. Vertical black lines indicate chromosome boundaries.

genomic alterations in bladder cancer provided in Table 2 in [31] to validate the method and compare with H-HMM. Our method (Figure 6(c)) concurred with the known frequently-amplified chromosome arms 20q, 8q, 19q, 1q, 20p, 17q, 19p, 5p, 2p, 10p, 3q and 7p, and frequently-lost 9p, 9q, 11p, 10q, 13q, 8p, 17p, 18q, 2q, 5q, 18p, 14q and 16q. The only known commonly-lost region which showed unconvincing common loss here was 6q. As for the H-HMM method (Figure 6(d)), it selects a small number of very small regions of gain and loss, which are difficult to verify with respect to the well-known frequently amplified arms in [31]. As is suggested, the method may therefore be useful for selecting the precise location of important genes. However, as can be seen in Figure 6(a)-(b), many, but not all, alterations are much larger than those found with H-HMM, and where for example there

are clearly several localized gains and losses in chromosome 8, H-HMM finds nothing at all. Perhaps the complexity of rearrangements in chromosome 8 is not easily taken into account by the H-HMM algorithm. Note finally that the weighted group fused LARS was more than 700 times faster than H-HMM.

7 Conclusion

We have proposed a framework that extends total-variation based approximation to the multidimensional setting, and developed two algorithms to solve the resulting convex optimization problem either exactly or approximately. We have shown theoretically and empirically that the group fused Lasso can consistently estimate a single change-points, and observed experimentally that this property is likely to hold also when several change-points are present. In particular, we observed both theoretically and empirically that increasing the number of profiles is highly beneficial to detect approximatively shared change-points, an encouraging property for biological applications where the accumulation of data measured on cohorts of patients promises to help in the detection of common genomic alterations.

Although we do not assume that all profiles have the same change-points, we estimate only shared change-points. In other words, we try to estimate the union of the change-points present in the profiles. This can be useful by itself, eg, for dimension reduction. If we wanted to detect change-points of individual profiles, we may either post-process the results of the group fused Lasso, or modify the formulation by, e.g., adding a TV penalty to each profile in addition to the group lasso penalty. Similarly, for some applications, we may want to add a ℓ_1/ℓ_2 norm to the group fused Lasso objective function in order to constrain some or all signals to be frequently null. Finally, from a computational point of view, we have proposed efficient algorithms to solve an optimization problem (4) which is the proximal operator of more general optimization problems where a smooth convex functional of U is minimized with a constraint on the multidimensional TV penalty; this paves the way to the efficient minimization of such functionals using, e.g., accelerated gradient methods [32].

Annex A: Computational lemmas

In this Annex we collect a few results useful to carry out the fast implementations claimed in Section 4. Remember that the $n \times (n-1)$ matrix X defined in (6) is defined by $X_{i,j} = d_j$ for $i > j$, 0 otherwise. Since the design matrix $\bar{X}$ of the group Lasso problem (8) is obtained by centering each column of X to zero mean, its columns are given by:

$$\forall i = 1, \ldots, n-1, \quad \bar{X}_{\bullet,i} = \left(\underbrace{\left(\frac{i}{n} - 1\right) d_i, \ldots, \left(\frac{i}{n} - 1\right) d_i}_{i}, \underbrace{\frac{i}{n} d_i, \ldots, \frac{i}{n} d_i}_{n-i} \right)^\top . \tag{14}$$

We first show how to compute efficiently $\bar{X}^\top R$ for any matrix R:

Lemma 5. *For any $R \in \mathbb{R}^{n \times p}$, we can compute $C = \bar{X}^\top R$ in $O(np)$ operations and memory as follows:*

1. *Compute the $n \times p$ matrix r of cumulative sums $r_{i,\bullet} = \sum_{j=1}^{i} R_{j\bullet}$ by the induction:*
   - $r_{1,\bullet} = R_{1,\bullet}$.
   - *For* $i = 2, \ldots, n$, $r_{i,\bullet} = r_{i-1,\bullet} + R_{i,\bullet}$.
2. *For* $i = 1, \ldots, n-1$, *compute* $C_{i,\bullet} = d_i \left(i r_{n,\bullet}/n - r_{i,\bullet} \right)$.

Proof. Using (14) we obtain the i-th row of $C = \bar{X}^\top R$, for $i = 1, \dots, n-1$, as follows:

$$\begin{aligned} C_{i,\bullet} &= \bar{X}_{\bullet,i}^\top R \\ &= \left(\frac{i}{n} - 1\right) d_i \left(\sum_{j=1}^{i} R_{j,\bullet}\right) + \frac{i}{n} d_i \left(\sum_{j=i+1}^{n} R_{j,\bullet}\right) \\ &= d_i \left(\frac{i}{n} r_{n,\bullet} - r_{i,\bullet}\right) . \end{aligned}$$

□

Next, we show how to compute efficiently submatrices of the $(n-1) \times (n-1)$ matrix $\bar{X}^\top \bar{X}$.

Lemma 6. *For any two subsets of indices $A = \left(a_1, \dots, a_{|A|}\right)$ and $B = \left(b_1, \dots, b_{|B|}\right)$ in $[1, n-1]$, the matrix $\bar{X}_{\bullet,A}^\top \bar{X}_{\bullet,B}$ can be computed in $O\left(|A||B|\right)$ in time and memory with the formula:*

$$\forall (i,j) \in [1, |A|] \times [1, |B|] \ , \quad \left[\bar{X}_{\bullet,A}^\top \bar{X}_{\bullet,B}\right]_{i,j} = d_{a_i} d_{b_j} \frac{\min(a_i, b_j)\left[n - \max(a_i, b_j)\right]}{n} . \tag{15}$$

Proof. Let us denote $V = \bar{X}_{\bullet,A}^\top \bar{X}_{\bullet,B}$. For any $(i,j) \in [1, |A|] \times [1, |B|]$, denoting $u = \min(a_i, b_j)$ and $v = \max(a_i, b_j)$, we easily get from (14) an explicit formula for $V_{i,j}$, namely,

$$\begin{aligned} V_{i,j} &= \bar{X}_{\bullet,a_i}^\top \bar{X}_{\bullet,b_j} \\ &= d_u d_v \left[u \left(\frac{u}{n} - 1\right)\left(\frac{v}{n} - 1\right) + (v-u)\frac{v}{n}\left(\frac{u}{n} - 1\right) + (n-v)\frac{u}{n}\frac{v}{n}\right] \\ &= d_u d_v \frac{u(n-v)}{n} . \end{aligned}$$

□

The next lemma provides another useful computational trick to compute efficiently $\bar{X}^\top \bar{X} R$ for any matrix R:

Lemma 7. *For any $R \in \mathbb{R}^{(n-1)\times p}$, we can compute $C = \bar{X}^\top \bar{X} R$ in $O(np)$ by*

1. *Compute, for $i = 1, \dots, n-1$, $\tilde{R}_{i,\bullet} = d_i R_{i,\bullet}$.*
2. *Compute the $1 \times p$ vector $S = \left(\sum_{i=1}^{n-1} i \tilde{R}_{i,\bullet}\right)/n$.*
3. *Compute the $(n-1) \times p$ matrix T defined by $T_{i,\bullet} = \sum_{j=i}^{n-1} \tilde{R}_{j,\bullet}$ by the induction:*
   - $T_{n-1,\bullet} = \tilde{R}_{n-1,\bullet}$.
   - *for $i = n-2, \dots, 1$, $T_{i,\bullet} = T_{i+1,\bullet} + \tilde{R}_{i,\bullet}$.*
4. *Compute the $(n-1) \times p$ matrix U defined by $U_{i,\bullet} = \sum_{j=1}^{i} \left(S - T_{j,\bullet}\right)$ by the induction:*
   - $U_{1,\bullet} = S - T_{1,\bullet}$.
   - *for $i = 2, \dots, n-1$, $U_{i,\bullet} = U_{i-1,\bullet} + S - T_{i,\bullet}$.*
5. *Compute, for $i = 1, \dots, n-1$, $C_{i,\bullet} = d_i U_{i,\bullet}$*

Each step in Lemma 7 has complexity $O(np)$ in memory and time, leading to an overall complexity in $O(np)$ to compute $\bar{X}^\top \bar{X} R$. We note that if R is row-sparse, i.e., is several rows of R are null, then the first two steps have complexity $O(sp)$, where s is the number of non-zero rows in R. Although this does not change the overall complexity to compute $\bar{X}^\top \bar{X} R$, this leads to a significant speed-up in practice when $s \ll n$.

Proof. Let us denote D the $(n-1)\times(n-1)$ diagonal matrix with entries $D_{i,i}=d_i$. By Lemma 6, we know that $\bar{X}^\top\bar{X}=DVD$, with $V_{i,j}=\min(i,j)\left[n-\max(i,j)\right]/n$, for $1\le i,j\le n-1$. Since step 1 computes $\tilde{R}=DR$ and step 5 computes $C=DU$, we just need to show that the U computed in step 4 satisfies $U=V\tilde{R}$ to conclude that $C=DV\tilde{R}=DVDR=\bar{X}^\top\bar{X}R$. By step 4, U is defined by the relation $U_{i,\bullet}-U_{i-1,\bullet}=S-T_{i,\bullet}$ for $i=1,\dots,n-1$ (with the convention $U_{0,\bullet}=0$), therefore we just need to show that $\left(V_{i,\bullet}-V_{i-1,\bullet}\right)\tilde{R}=S-T_{i,\bullet}$ for $i=1,\dots,n-1$ to conclude. For $0\le j<i\le n-1$, we note that $V_{i,j}=j(n-i)/n$ (with the convention $V_{0,\bullet}=0$) and $V_{i-1,j}=j(n-i+1)/n$, and therefore $V_{i,j}-V_{i-1,j}=-j/n$. For $1\le i\le j\le n-1$, we have $V_{i,j}=i(n-j)/n$ and $V_{i-1,j}=(i-1)(n-j)/n$ and therefore $V_{i,j}-V_{i-1,j}=1-j/n$. Combining these expressions we get, for $i=1,\dots,n-1$:

$$\left(V_{i,\bullet}-V_{i-1,\bullet}\right)\tilde{R}=-\sum_{j=1}^{n-1}\frac{j\tilde{R}_{j,\bullet}}{n}+\sum_{j=i}^{n-1}\tilde{R}_{j,\bullet}=S-T_{i,\bullet}\,,$$

where S and T are defined in steps 2 and 3. This concludes the proof that $C=\bar{X}^\top\bar{X}R$. □

Next we show that $\left(\bar{X}^\top\bar{X}\right)^{-1}$ has a tridiagonal structure, resulting in fast matrix multiplication.

Lemma 8. *For any set $A=\left(a_1,\dots,a_{|A|}\right)$ of distinct indices with $1\le a_1<\ldots<a_{|A|}\le n-1$, the matrix $\left(\bar{X}_{\bullet,A}^\top\bar{X}_{\bullet,A}\right)$ is invertible, and for any $|A|\times p$ matrix R, the matrix*

$$C=\left(\bar{X}_{\bullet,A}^\top\bar{X}_{\bullet,A}\right)^{-1}R$$

can be computed in $O(|A|p)$ in time and memory by

1. *For $i=1,\dots,|A|-1$, compute*

$$\Delta_i=\frac{d_{a_{i+1}}^{-1}R_{i+1,\bullet}-d_{a_i}^{-1}R_{i,\bullet}}{a_{i+1}-a_i}\,.$$

2. *Compute the successive rows of C according to:*

$$\begin{aligned}C_{1,\bullet}&=d_{a_1}^{-1}\left(\frac{R_{1,\bullet}}{a_1}-\Delta_1\right)\,,\\ C_{i,\bullet}&=d_{a_i}^{-1}\left(\Delta_{i-1}-\Delta_i\right)\quad \text{for } i=2,\dots,|A|-1\,,\\ C_{|A|,\bullet}&=d_{a_{|A|}}^{-1}\left(\Delta_{|A|-1}+\frac{R_{|A|,\bullet}}{n-a_{|A|}}\right)\,.\end{aligned}\tag{16}$$

Proof. Let us denote $V=\bar{X}_{\bullet,A}^\top\bar{X}_{\bullet,A}$. By Lemma 6 we know that, for $1\le i\le j\le|A|$,

$$V_{i,j}=d_{a_i}d_{a_j}\frac{a_i(n-a_j)}{n}\,.$$

V being symmetric semi-separable, one can easily check that V is invertible and admits as inverse a tridiagonal matrix with the following entries [33]:

$$\begin{aligned}V_{i,i}^{-1}&=d_{a_i}^{-2}\left(\frac{1}{a_i-a_{i-1}}+\frac{1}{a_{i+1}-a_i}\right)\quad\text{for } i=1,\dots,|A|,\\ V_{i,i+1}^{-1}=V_{i+1,i}^{-1}&=-\frac{d_{a_i}^{-1}d_{a_{i+1}}^{-1}}{a_{i+1}-a_i}\quad\text{for } i=1,\dots,|A|-1,\end{aligned}\tag{17}$$

where by convention we define $a_0 = 0$ and $a_{|A|+1} = n$. This tri-diagonal structure allows successive rows of C to be expressed as a sum of just a few terms. More precisely, for $1 < i < |A|$, we obtain:

$$\begin{aligned} C_{i,\bullet} &= -\frac{d_{a_{i-1}}^{-1} d_{a_i}^{-1} R_{i-1,\bullet}}{a_i - a_{i-1}} + d_{a_i}^{-2} R_{i,\bullet} \left(\frac{1}{a_i - a_{i-1}} + \frac{1}{a_{i+1} - a_i} \right) - \frac{d_{a_i}^{-1} d_{a_{i+1}}^{-1} R_{i+1,\bullet}}{a_{i+1} - a_i} \\ &= d_{a_i}^{-1} \left(\frac{d_{a_i}^{-1} R_{i,\bullet} - d_{a_{i-1}}^{-1} R_{i-1,\bullet}}{a_i - a_{i-1}} + \frac{d_{a_i}^{-1} R_{i,\bullet} - d_{a_{i+1}}^{-1} R_{i+1,\bullet}}{a_{i+1} - a_i} \right) \\ &= d_{a_i}^{-1} \left(\Delta_{i-1} - \Delta_i \right) . \end{aligned}$$

Similarly, for $i = 1$ and $i = |A|$ we easily recover (16). □

Annex B: Proof of Lemma 1

The solution of (4) is constant, i.e., corresponds to $\beta = 0$ (no change-point), as long as the KKT conditions (10) are satisfied for $\beta = 0$. This translates to $\| \bar{X}_{\bullet,i}^\top \bar{Y} \| \leq \lambda$ for all i. The first change-point occurs when $\lambda = \max_i \| \bar{X}_{\bullet,i}^\top \bar{Y} \|$, and the change-point is precisely located in the position i that reaches the maximum. Therefore the first change-point is the row with the largest Euclidean norm of the matrix:

$$\hat{c} = \bar{X}^\top \bar{Y} = \bar{X}^\top \bar{X} \beta^* + \bar{X}^\top W .$$

The entries of the matrix $\hat{c}$ are therefore jointly Gaussian. Since only the u-th row $\beta_{u,\bullet}$ of β is non-zero, we get

$$E(\hat{c}) = \bar{X}^\top \bar{X} \beta^* = \bar{X}^\top \bar{X}_{\bullet,u} \beta^*_{u,\bullet} .$$

Using Lemma 6 we compute:

$$E(\hat{c}_{i,\bullet}) = \left[\bar{X}^\top \bar{X} \beta^* \right]_{i,\bullet} = \begin{cases} d_i d_u \frac{i(n-u)}{n} \beta^*_{u,\bullet} & \text{for } 1 \leq i \leq u , \\ d_i d_u \frac{u(n-i)}{n} \beta^*_{u,\bullet} & \text{for } u \leq i \leq n-1 . \end{cases} \tag{18}$$

On the other hand, by (14) we have for any $i \in [1, n-1]$,

$$\left[\bar{X}^\top W \right]_{i,\bullet} = d_i \left[\sum_{j=1}^{i} \left(\frac{i}{n} - 1 \right) W_{j,\bullet} + \sum_{j=i+1}^{n} \frac{i}{n} W_{j,\bullet} \right] .$$

Since

$$E \left(W_{i,\bullet}^\top W_{j,\bullet} \right) = \delta_{i,j} \sigma^2 \mathbf{I}_p ,$$

where $\delta_{i,j}$ is the Dirac function, we have for $1 \leq i \leq j \leq n-1$:

$$\begin{aligned} & E \left(\left[\bar{X}^\top W \right]_{i,\bullet}^\top \left[\bar{X}^\top W \right]_{j,\bullet} \right) \\ &= d_i d_j \left[i \left(\frac{i}{n} - 1 \right) \left(\frac{j}{n} - 1 \right) + (j-i) \frac{i}{n} \left(\frac{j}{n} - 1 \right) + (n-j) \frac{i}{n} \frac{j}{n} \right] \sigma^2 \mathbf{I}_p \\ &= d_i d_j \frac{i\,(n-j)}{n} \sigma^2 \mathbf{I}_p . \end{aligned} \tag{19}$$

In summary, we have shown that $\hat{c}$ is jointly Gaussian with $E\left(\hat{c}_{i,\bullet}\right)$ given by (18) and covariance between $\hat{c}_{i,\bullet}$ and $\hat{c}_{j,\bullet}$ given by (19).

In particular, if we denote $F_i = \| \hat{c}_{i,\bullet} \|^2$, then, for $i \leq u$, $F_i n / \left(d_i^2 i(n-i)\sigma^2 \right)$ follows a non-central χ^2 distribution with p degrees of freedom and non-centrality parameter $p \bar{\beta}_p^2 d_u^2 i(n-u)^2 / \left[n(n-i)\sigma^2 \right]$. In particular,

$$E F_i = p {\bar{\beta}_p}^2 d_i^2 d_u^2 \frac{i^2\,(n-u)^2}{n^2} + p d_i^2 \frac{i(n-i)}{n} \sigma^2 ,$$

and since $\lim_{p\to+\infty} \bar{\beta}_p^{\,2} = \bar{\beta}^2$, we get that F_i/p converges in probability to

$$G_i = \frac{EF_i}{p} = \bar{\beta}^2 d_i^2 d_u^2 \frac{i^2\,(n-u)^2}{n^2} + d_i^2 \frac{i(n-i)}{n}\sigma^2\,. \tag{20}$$

A similar computation shows that for $i \geq u$, F_i/p converges in probability to

$$G_i = \bar{\beta}^2 d_i^2 d_u^2 \frac{u^2\,(n-i)^2}{n^2} + d_i^2 \frac{i(n-i)}{n}\sigma^2\,. \tag{21}$$

Note that (20) and (21) are equivalently defined in (11). Now, let $V = \operatorname{argmax}_{i\in[1,n-1]} G_i$. For any $v \in V$ and $j \notin V$, the probability of the event $F_v > F_j$ tends to 1, because $G_v > G_j$. By the union bound the probability of the event $\max_{j\notin V} F_i < \max_{v\in V} F_v$ also converges to 1, showing that the probability to select a change-point in V converges to 1 as $p \to +\infty$. □

Annex C: Proof of Theorem 2

By Lemma 1, we know that the first change-point selected by (4) is in $\operatorname{argmax}_{i\in[1,n]} G_i$ with probability tending to 1 as p increases, where G_i is defined in (11). We will therefore asymptotically select the correct change-point u if and only if $G_u = \max_{i\in[1,n-1]} G_i$. Remember we assume, without lack of generality, that $u \geq n/2$. For $u \leq i \leq n-1$, we observe that G_i given by (21) is a decreasing function of i as a sum of two decreasing functions. Therefore, it always holds that $G_u = \max_{i\in[u,n-1]} G_i$, and we just need to check whether or not $G_u = \max_{i\in[1,u]} G_i$ holds.

For $i \in [1,u]$, G_i given by (20) is a second-order polynomial of i, which is equal to 0 at $i = 0$ and strictly positive for $i = u$. Therefore $G_u = \max_{i\in[1,u]} G_i$ if and only if $G_u > G_{u-1}$. Let us therefore compute:

$$\begin{aligned}
G_u - G_{u-1} &= \bar{\beta}^2 \frac{(n-u)^2}{n^2}\left[u^2 - (u-1)^2\right] + \frac{\sigma^2}{n}\left[u(n-u) - (u-1)(n-u+1)\right] \\
&= \frac{\bar{\beta}^2(2u-1)(n-u)^2}{n^2} + \frac{\sigma^2(n-2u+1)}{n} \\
&= 2\left[\bar{\beta}^2 n\,(1-\alpha)^2\left(\alpha - \frac{1}{2n}\right) + \sigma^2\left(\frac{1}{2} - \alpha + \frac{1}{2n}\right)\right] \\
&= 2\left(\tilde{\sigma}^2 - \sigma^2\right)\left(\alpha - \frac{1}{2} - \frac{1}{2n}\right)\,,
\end{aligned} \tag{22}$$

where $\alpha = u/n$ and

$$\tilde{\sigma}^2 = n\bar{\beta}^2 \frac{(1-\alpha)^2(\alpha - \frac{1}{2n})}{\alpha - \frac{1}{2} - \frac{1}{2n}}\,.$$

This shows that, when $\alpha > 1/2 + 1/(2n)$, $G_u > G_{u-1}$ if and only if $\sigma < \tilde{\sigma}$. On the other hand, when $\alpha = 1/2$ or $1/2 + 1/(2n)$, we have always that $G_u > G_{u-1}$. □

Annex D: Proof of Theorem 3

As for the proof of Theorem 2, we need to check whether or not $G_u = \max_{i\in[1,n-1]} G_i$, where G_i is defined in (11), to deduce whether the method selects the correct change-point u or a different position with probability tending to 1 when p increases. Substituting weights d_i defined in (5) into G_i, we obtain:

$$G_i = \sigma^2 + \bar{\beta}^2 \times \begin{cases} i\,(n-u)\,/u\,(n-i) & \text{if } i \leq u\,, \\ u\,(n-i)\,/i\,(n-u) & \text{otherwise.} \end{cases} \tag{23}$$

It is then easy to see that (23) is increasing on $[1,u]$, and decreasing on $[u,n-1]$, showing that we always have $\operatorname{argmax}_{i\in[1,n-1]} G_i = u$. The result then follows from Lemma 1. □

Annex E: Proof of Theorem 4

Following the proof of Lemma 1, let us estimate $F_i = \| \hat{c}_{i,\bullet} \|^2$ for $i \in [1, n-1]$. For any $j \in [1, p]$, we first observe by (18) that

$$\left[\bar{X}^\top \bar{X} \beta\right]_{i,j} = \begin{cases} d_i d_{U_j} \frac{i(n-U_j)}{n} \beta_j & \text{if } i \leq U_j \,, \\ d_i d_{U_j} \frac{U_j(n-i)}{n} \beta_j & \text{otherwise} \,. \end{cases} \tag{24}$$

Therefore,

$$\sum_{j=1}^{p} \left[\bar{X}^\top \bar{X} \beta\right]_{i,j}^2 = \frac{d_i^2}{n^2} \sum_{j=1}^{p} d_{U_j}^2 \beta_j^2 \left[i^2 (n - U_j)^2 \mathbf{1}(i \leq U_j) + (n-i)^2 U_j^2 \mathbf{1}(i > U_j) \right] , \tag{25}$$

and by independence of β_i and U_i:

$$\frac{1}{p} E \sum_{j=1}^{p} \left[\bar{X}^\top \bar{X} \beta\right]_{i,j}^2 = \bar{\beta}^2 \frac{d_i^2}{n^2} \left[\sum_{u=1}^{i} p_u d_u^2 u^2 (n-i)^2 + \sum_{u=i+1}^{n-1} p_u d_u^2 (n-u)^2 i^2 \right] .$$

Since $(\beta_i, U_i)_{i=1,\ldots,p}$ are independent of the noise, we obtain that F_i/p converges in probability to

$$G_i = \bar{\beta}^2 \frac{d_i^2}{n^2} \left[\sum_{u=1}^{i} p_u d_u^2 u^2 (n-i)^2 + \sum_{u=i+1}^{n-1} p_u d_u^2 (n-u)^2 i^2 \right] + d_i^2 \frac{i(n-i)}{n} \sigma^2 \,. \tag{26}$$

As in Lemma 1 we can conclude that the method will select the position

$$\hat{u} = \operatorname*{argmax}_{u \in [1, n-1]} G_i$$

with probability tending to 1 as p increases.

Let us now assume that the support of P_U is an interval $[a, b]$ (corresponding to a possible range of fluctuation of a change-point). Then, we observe that for $i \leq a$, G_i in (26) reduces to

$$\begin{aligned} G_i &= \bar{\beta}^2 \frac{d_i^2}{n^2} \left[0 + \sum_{u=a}^{b} p_u d_u^2 (n-u)^2 i^2 \right] + d_i^2 \frac{i(n-i)}{n} \sigma^2 \\ &= \bar{\beta}^2 \frac{i^2 d_i^2}{n^2} E \left[d_U (n - U) \right]^2 + d_i^2 \frac{i(n-i)}{n} \sigma^2 \,. \end{aligned} \tag{27}$$

Let us now consider the two possible weighting schemes.

- In the unweighted case $d_i = 1$ for $i = 1, \ldots, n-1$, we obtain from (27) that for $i \leq a$:

$$G_i = \bar{\beta}^2 \frac{i^2 E (n-U)^2}{n^2} + \frac{i(n-i)}{n} \sigma^2 \,. \tag{28}$$

  While the first term in (28) is strictly increasing on $[0, a]$, the second term moves the maximum of G_i towards $n/2$. This shows that the maximum of G_i is always at least a when $a \leq n/2$. By symmetry, it is also always smaller or equal to b when $b \geq n/2$. When $n/2 \in [a, b]$, we deduce that for any $\sigma^2 > 0$, $\hat{u} \in [a, b]$. Otherwise, let us suppose without lack of generality that $n/2 < a \leq b$. Then, G_i being quadratic on $[0, a]$ and equal to 0 at 0, the maximum of G_i will not occur before a if and only if $G_{a-1} < G_a$. A computation similar to the one in the proof of Theorem 2 shows that

$$G_a - G_{a-1} = 2 \left(\tilde{\sigma}_m^2 - \sigma^2 \right) \left(\alpha_m - \frac{1}{2} + \frac{1}{2n} \right) ,$$

where

$$\tilde{\sigma}_m^2 = n\bar{\beta}^2 \frac{E(1-\alpha)^2(\alpha_m - \frac{1}{2n})}{\alpha_m - \frac{1}{2} - \frac{1}{2n}} .$$

This shows that $G_a > G_{a-1}$ if and only if $\sigma^2 < \tilde{\sigma}_m^2$. Since $b > n/2$, we also know that $\hat{u} \leq b$, i.e., $\hat{u} \in [a, b]$ in that case. The case $1 \leq a \leq b < n/2$ can be treated similarly. To conclude the proof it suffices to observe that

$$E(1-\alpha)^2 = (1 - E\alpha)^2 + \text{var}(\alpha) .$$

- In the weighted case $d_i = \sqrt{\frac{n}{i(n-i)}}$ for $i = 1, \ldots, n-1$, we obtain from (26) and (27) that for $i \leq a$:

$$G_i = \bar{\beta}^2 \frac{i}{n-i} E\left[\frac{n-U}{U}\right] + \sigma^2 . \tag{29}$$

This is always an increasing function of i on $[1, a]$, showing that the maximum of G_i can not be strictly smaller than a. By symmetry, it can also never be larger than b, from which we conclude that it is always between a and b, i.e., in the support of P_U.

□

References

[1] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE T. Signal. Proces.*, 53(8):2961–2974, 2005.

[2] Z. Harchaoui, F. Vallet, A. Lung-Yut-Fong, and O. Cappe. A regularized kernel-based approach to unsupervised audio segmentation. In *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1665–1668, Washington, DC, USA, 2009. IEEE Computer Society.

[3] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek, and Hongjoong Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE T. Signal. Proces.*, 54(9):3372–3382, 2006.

[4] C. Lévy-Leduc and F. Roueff. Detection and localization of change-points in high-dimensional network traffic data. *Ann. Appl. Stat.*, 3(2):637–662, 2009.

[5] M. Talih and N. Hengartner. Structural learning with time-varying components: tracking the cross-section of financial time series. *J. R. Stat. Soc. Ser. B*, 67(3):321–341, 2005.

[6] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 6:27, 2005.

[7] N. R. Zhang, D. O. Siegmund, H. Ji, and J. Li. Detecting simultaneous change-points in multiple sequences. *Biometrika*, 97(3):631–645, 2010.

[8] M. Basseville and N. Nikiforov. *Detection of abrupt changes: theory and application*. Information and System Sciences Series. Prentice Hall Information, 1993.

[9] B. Brodsky and B. Darkhovsky. *Nonparametric Methods in Change-Point Problems*. Kluwer Academic Publishers, 1993.

[10] Y. C. Yao. Estimating the number of change-points via schwarz criterion. *Stat. Probab. Lett.*, 6:181–189, 1988.

[11] L. Birgé and P. Massart. Gaussian model selection. *J. Eur. Math. Soc.*, 3:203–268, 2001.

[12] M. Lavielle and G. Teyssière. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, 46(3):287–306, 2006.

[13] L. J. Vostrikova. Detection of disorder in multidimensional stochastic processes. *Soviet Mathematics Doklady*, 24:55–59, 1981.

[14] M. Lavielle and Teyssière. Adaptive detection of multiple change-points in asset price volatility. In G. Teyssière and A. Kirman, editors, *Long-Memory in Economics*, pages 129–156. Springer Verlag, Berlin, 2005.

[15] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[16] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(1):91–108, 2005.

[17] Z. Harchaoui and C. Levy-Leduc. Catching change-points with lasso. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Adv. Neural. Inform. Process Syst.*, volume 20, pages 617–624. MIT Press, Cambridge, MA, 2008.

[18] A. Rinaldo. Properties and refinements of the fused lasso. *Ann. Stat.*, 37(5B):2922–2952, 2009.

[19] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Statist.*, 1(1):302–332, 2007.

[20] H. Hoefling. A path algorithm for the Fused Lasso Signal Approximator. Technical Report 0910.0526v1, arXiv, Oct. 2009.

[21] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B*, 68(1):49–67, 2006.

[22] J-P. Vert and K. Bleakley. Fast detection of multiple change-points shared by many signals using group LARS. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Adv. Neural. Inform. Process Syst.*, volume 22, pages 2343–2352, 2010.

[23] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, 2010.

[24] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[25] W. Fu. Penalized regressions: the bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7:397–416, 1998.

[26] M. Lavielle. Using penalized contrasts for the change-point problem. *Signal Process.*, 85(8):1501–1510, 2005.

[27] S.P. Shah, W.L. Lam, R.T. Ng, and K.P. Murphy. Modeling recurrent DNA copy number alterations in array CGH data. *Bioinformatics*, 23(13):i450–i458, 2007.

[28] B. P. Coe, W. W. Lockwood, L. Girard, R. Chari, C. MacAulay, S. Lam, A. F. Gazdar, J. D. Minna, and W. L. Lam. Differential disruption of cell cycle pathways in small cell and non-small cell lung cancer. *Br. J. Cancer*, 94:1927–1935, 2006.

[29] C. Garnis, W. W. Lockwood, E. Vucic, Y. Ge, L. Girard, J. D. Minna, A. F. Gazdar, S. Lam, C. MacAulay, and W. L. Lam. High resolution analysis of non-small cell lung cancer cell lines by whole genome tiling path array CGH. *Int. J. Cancer*, 118(6):1556–1564, 2006.

[30] N. Stransky, C. Vallot, F. Reyal, I. Bernard-Pierrot, S. G. Diez de Medina, R. Segraves, Y. de Rycke, P. Elvin, A. Cassidy, C. Spraggon, A. Graham, J. Southgate, B. Asselain, Y. Allory, C. C. Abbou, D. G. Albertson, J.-P. Thiery, D. K. Chopin, D. Pinkel, and F. Radvanyi. Regional copy number-independent deregulation of transcription in cancer. *Nat. Genet.*, 38(12):1386–1396, Dec 2006.

[31] E. Blaveri, J. L. Brewer, R. Roydasgupta, J. Fridlyand, S. DeVries, T. Koppie, S. Pejavar, K. Mehta, P. Carroll, J. P. Simko, and F. M. Waldman. Bladder cancer stage and outcome by array-based comparative genomic hybridization. *Clin. Cancer Res.*, 11(19 Pt 1):7012–7022, Oct 2005.

[32] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

[33] J. Baranger and M. Duc-Jacquet. Matrices tridiagonales symétriques et matrices factorisables. *Revue française d'informatique et de recherche opérationnelle, série rouge*, 5(3):61–66, 1971.