Journal of Multivariate Analysis 167 (2018) 319–330

Contents lists available at ScienceDirect

Journal of Multivariate Analysis

journal homepage: www.elsevier.com/locate/jmva

A model selection approach for multiple sequence segmentation and dimensionality reduction

Bruno M. Castro [a,*], Renan B. Lemes [b], Jonatas Cesar [b], Tábita Hünemeier [b], Florencia Leonardi [c]

[a] Departamento de Estatística, Universidade Federal do Rio Grande do Norte, Brazil
[b] Instituto de Biociências, Universidade de São Paulo, Brazil
[c] Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil

ARTICLE INFO

Article history:
Received 30 April 2018
Available online 22 May 2018

AMS 2010 subject classifications:
62G05
62G20

ABSTRACT

In this paper we consider the problem of segmenting n aligned random sequences of equal length m into a finite number of independent blocks. We propose a penalized maximum likelihood criterion to infer simultaneously the number of points of independence as well as the position of each point. We show how to compute exactly the estimator by means of a dynamic programming algorithm with time complexity $O(m^2n)$. We also propose another method, called hierarchical algorithm, that provides an approximation to the estimator when the sample size increases and runs in time $O\{m\ln(m)n\}$. Our main theoretical results are the strong consistency of both estimators when the sample size n grows to infinity. We illustrate the convergence of these algorithms through some simulation examples and we apply the method to identify recombination hotspots in real SNPs data.

1. Introduction

The problem of multiple sequence segmentation and dimensionality reduction is of crucial importance for many applied areas, including the analysis of multiple alignments of DNA/RNA and Amino Acid (AA) sequences. In these cases, one of the main goals is to investigate some aspects of the genetic variation, for example, inferring which genomic regions can be considered putative hotspots of genetic recombination. Another application on practical ground is to look for small subregions in the sequences that are related to a phenotypic variable. One example of this is the genome-wide association studies (GWAS) of Single Nucleotide Polymorphisms (SNPs), where the interest is to find positions in the genome associated with a given phenotypic trait. Traditionally, this task is performed by making a simultaneous hypotheses test on each individual position or on small sub-windows of fixed length, as in the PLINK suite [6,20]. But considering all variables as mutually independent does not translate the intrinsic relations present in genomic data and can result in weak or spurious discoveries. This fact has led the community to develop methods that take into account the dependence between adjacent or even non-adjacent variables; see, e.g., [16].

Many other authors have also considered the problem of inferring local dependencies in data, using a wide range of probabilistic models. In a recent paper, Algama and Keith [1] present a detailed review about the most well-known sequence segmentation techniques and the models assumed in each case. Their list contains sliding window analysis [22], hidden Markov models [3,10], recursive segmentation algorithms [8,18] and multiple change-point analysis [9,21]. They also refer to other methods for sequence segmentation and pattern identification based on least squares estimation [13] or on wavelet

[*] Corresponding author.
E-mail address: bruno.monte19@gmail.com (B.M. Castro).

https://doi.org/10.1016/j.jmva.2018.05.006
0047-259X/© 2018 Elsevier Inc. All rights reserved.

analysis [23]. We refer the reader to the work [1] where a brief explanation of these methods is presented, and also other references for the problem of sequence segmentation are given.

Our main goal in this paper is to introduce a new approach for the problem of multiple sequence segmentation into independent blocks. We are interested in inferring the maximal set of points of independence, when the number of such points is unknown. To do this we propose a penalized maximum likelihood criterion to infer simultaneously the number of points of independence and their positions, for n aligned random sequences of equal length m. We show how to compute exactly this estimator by means of a dynamic programming algorithm and we prove its almost sure convergence to the true set of points of independence when the sample size n increases. In cases where the size m of the sequences is large, we propose a suboptimal but more efficient algorithm that also converges almost surely to the set of points of independence when the sample size n increases. The main advantage of our procedure is that we do not need to assume a fixed number of segments and the optimal number of points of independence can be learned from the data. Our method can be used to reduce drastically the dimensionality of the joint probability distributions from exponential to linear functions of the length of the sequences, given by m, somehow sharing the same objectives of correspondence analysis, a principal component method for nominal categorical data.

A related approach is considered by Gwadera et al. [12], who present a method to determine the optimal number of segments in a sequence using a Variable Length Markov Chain (VLMC) model on each segment. They propose to use the Bayesian Information Criterion (BIC) and a variant of the Minimum Description Length (MDL) Principle to select the number of segments for the given sequence. Their method consists in estimating change points on a unique stationary sequence while ours looks for points of independence on non-stationary aligned sequences. In stark contrast to their approach, we do not need to assume a specific probabilistic model on each segment and we can estimate a general multivariate distribution on each segment. Moreover, Gwadera et al. [12] do not present a formal proof that their method succeeds to detect the number and position of the change-points.

This paper is organized as follows. In Section 2 we present background material, show how to compute the estimators, and state the main theoretical results. In Section 3 we report the results of simulations illustrating the performance of the segmentation method, and in Section 4 we show a practical application on real data. In Section 5 we discuss the results and in the Appendix we include the proofs of the theoretical results presented in Section 2.

2. Likelihood function and model selection

2.1. Notation and definitions

Let $\mathbf{X} = (X_1, \ldots, X_m)$ be a random vector taking values in $A_1 \times \cdots \times A_m$, where A_i is a finite alphabet for all $i \in \{1, \ldots, m\}$. The cardinal of the finite set A_i will be denoted by $|A_i|$. We say that $j \in \{1, \ldots, m-1\}$ is a point of independence for $\mathbf{X}$ if the random vectors $(X_1, \ldots, X_j)$ and $(X_{j+1}, \ldots, X_m)$ are independent.

Given two integers $r \le s$, denote by $r:s$ the integer interval $r, \ldots, s$. We say $U_{r:s} \subset r:(s-1)$ is a maximal set of points of independence for the interval $r:s$ if no $v \in r:(s-1) \setminus U_{r:s}$ is a point of independence for $\mathbf{X}$. For each random vector $\mathbf{X}$ and each interval $r:s$ there is only one maximal set of points of independence; from now on this special set will be denoted by $U^*_{r:s}$. In the special case $r = 1, s = m$ we will simply write U^*.

Without loss of generality we will also suppose that the set $U_{r:s}$ is ordered; in this case $U_{r:s} = (u_1, \ldots, u_k)$ with $u_i < u_j$ if $i < j$. From $U_{r:s}$ it is possible to obtain the set of blocks of independent variables as the set $B(U_{r:s}) = \{I_1, \ldots, I_{k+1}\}$ of integer intervals given by $I_1 = r:u_1$, $I_i = (u_{i-1}+1):u_i$ for all $i \in \{2, \ldots, k\}$, and $I_{k+1} = (u_k+1):s$.

Given an integer interval $I = r:s$ denote by A^I the set of finite strings on $A_r \times \cdots \times A_s$ with positive probability, viz.

$$A^I = \{w \in A_r \times \cdots \times A_s : \Pr(w) > 0\}.$$

Assume we observe an iid sample $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ of $\mathbf{X}$, denoted by $\mathbf{x}$. Then, the likelihood function for the set U can be written as

$$L(U; \mathbf{x}) = \prod_{i=1}^{n} \prod_{I \in B(U)} \Pr(X_j = x_j^{(i)} : j \in I). \tag{1}$$

Denote by $\mathbf{x}_{r:s}$ the iid sample $\mathbf{x}_{r:s}^{(1)}, \ldots, \mathbf{x}_{r:s}^{(n)}$. Given a finite string $a_{r:s} \in A^{r:s}$, define

$$N(a_{r:s}) = \sum_{i=1}^{n} \mathbf{1}\{x_{r:s}^{(i)} = a_{r:s}\}.$$

Then $L(U; \mathbf{x})$ can be rewritten as

$$L(U; \mathbf{x}) = \prod_{I \in B(U)} \prod_{a_I \in A^I} \Pr(X_I = a_I)^{N(a_I)}. \tag{2}$$

Denote by $\widehat{\Pr}(a_I)$ the maximum likelihood estimators for the probabilities $\Pr(a_I)$, i.e., the values maximizing (2). It can be proved that for any interval I and any $a_I \in A^I$, the estimator $\widehat{\Pr}(a_I)$ is given, for all $a_I \in A^I$, by

$$\widehat{\Pr}(a_I) = N(a_I)/n.$$

Then we can plug-in these estimators in (2) obtaining the value of the maximum likelihood, given by

$$\hat{L}(U;\mathbf{x}) = \prod_{I\in B(U)} \prod_{a_I\in A^I} \widehat{\Pr}(a_I)^{N(a_I)}.$$

For simplicity in the sequel we will work with the logarithm of the estimated likelihood function. For this reason, we define

$$\hat{\ell}(U;\mathbf{x}) = \sum_{I\in B(U)} Q(I,\mathbf{x}),$$

where

$$Q(I,\mathbf{x}) = \sum_{a_I\in A^I} N(a_I)\ln \widehat{\Pr}(a_I).$$

Now we introduce the model selection criterion based on the maximization of the penalized log-likelihood.

Definition 1. Given a sample $\mathbf{x}$ and a constant $c > 0$, define

$$\mathrm{PML}(U,\mathbf{x}) = \hat{\ell}(U,\mathbf{x}) + c(|U|+1)\sqrt{n}$$

with $|U|$ the number of points in U and let

$$\hat{U}(\mathbf{x}) = \arg\max_{U\subseteq 1:(m-1)} \{\mathrm{PML}(U,\mathbf{x})\}. \tag{3}$$

The estimator $\hat{U}(\mathbf{x})$ is a penalized maximum log-likelihood estimator for the true set of points of independence U^*. The penalizing factor $|U|+1$ represents a measure of the complexity of the model when the set of points of independence is U. Observe that in the case of the multivariate distribution whose likelihood function is given in (2), the number of parameters decreases as U increases. Therefore, by adding the term $c(|U|+1)\sqrt{n}$ the estimator (3) favors models with less parameters (or more blocks). In Definition 1 and in the proof of the main theoretical results, we use this specific penalizing term in order to maintain the simplicity of the approach, but many other functions are possible. In particular, the estimators remain strongly consistent if we use as penalty term the total number of parameters of the multivariate distribution with independent blocks $B(U)$, as is usual in the BIC penalty term, that is given by $\sum_{I\in B(U)}(|A|^{|I|}-1)$.

Remark 1. Even though the approach presented in this paper is totally nonparametric, many other definitions for the likelihood function in (2) are possible taking into account some parametric model. For example we can assume a Markov chain structure of any given order or a Variable Length Markov Chain for each block. This is overall appealing for large m and $|U|$ small relative to m where the large number of parameters of the multivariate distribution can lead to substantial overestimation errors.

Remark 2. In the general case of the multivariate distribution without any independence structure, we would need to estimate $|A|^m-1$ parameters to compute the likelihood function (1). In contrast, by using the information contained in the independence set U^* the number of parameters decreases to $\sum_{I\in B(U^*)}(|A|^{|I|}-1)$, that can be as small as $(|A|-1)m$ when all m variables are independent. For this reason, the estimator $\hat{U}(\mathbf{x})$ can reduce considerably the dimensionality of the estimated probability distribution for $\mathbf{X}$, overall in cases where the sample size n is not very large related to the sequences length m.

2.2. Computation of the independence set estimator

In this section we show how to compute efficiently the penalized maximum likelihood estimator given in Definition 1. The first part, mostly inspired in [15], presents a dynamic programming algorithm that computes exactly the optimal argument of (3), performing $O(m^2n)$ operations; see also [2] and references therein. In the second part we propose a divide and conquer approximation for the optimum in (3), at a more efficient computing time. We show that this second algorithm also retrieves the true set of points of independence with probability 1 when the sample size grows.

2.2.1. Dynamic programming algorithm

Let $F_{k+1}(m)$ denote the maximum value of the function in (3) corresponding to a k-dimensional vector U for the sample $\mathbf{x}$, i.e.,

$$F_{k+1}(m) = \max_{U,|U|=k}\{\hat{\ell}(U,\mathbf{x}) + c(|U|+1)\sqrt{n}\}.$$

It is easy to see that the optimal k-dimensional vector U leading to $F_{k+1}(m)$ consists of $k-1$ independence points over $1:i$ and a single block $(i+1):m$, where i is the rightmost point of independence. Moreover, the k blocks over $1:i$ must maximize the function (3) for the sample $\mathbf{x}_{1:i}$, attaining $F_k(i)$. In this way, the dynamic programming recursion is

$$F_1(m) = \widetilde{Q}(1:m,\mathbf{x}), \quad F_{k+1}(m) = \max_{i\in\{k,\ldots,m-1\}}\{F_k(i) + \widetilde{Q}((i+1):m,\mathbf{x})\},$$

where $\widetilde{Q}$ is given by

$$\widetilde{Q}(I,\mathbf{x}) = Q(I,\mathbf{x}) + c\sqrt{n}.$$

The estimator $\hat{U}(\mathbf{x})$ in Definition 1 is computed by tabulating $F_1(i)$ for all i up to m, and then by computing $F_2(i)$ for all i and so on up to $F_m(m)$. The optimal value of k is obtained by the equation

$$\hat{k} = \underset{k\in\{1,\ldots,m\}}{\arg\max}\{F_k(m)\} - 1.$$

and the vector $\hat{U}(\mathbf{x}) = (\hat{u}_1,\ldots,\hat{u}_{\hat{k}})$ is given by

$$\hat{u}_{\hat{k}} = \underset{i\in\{\hat{k},\ldots,m-1\}}{\arg\max}\ \{F_{\hat{k}}(i) + \widetilde{Q}((i+1):m,\mathbf{x})\},$$

$$\hat{u}_i = \underset{j\in\{i\ldots,\hat{u}_{i+1}-1\}}{\arg\max}\ \{F_i(j) + \widetilde{Q}((j+1):\hat{u}_{i+1},\mathbf{x})\},\quad i\in\{1,\ldots,\hat{k}-1\}.$$

2.2.2. Hierarchical algorithm

Here we present a more efficient divide-and-conquer algorithm to approximate the estimator given by Definition 1, with computational cost $O\{m\ln(m)n\}$. Let $I = r:s$ be an integer interval (with the convention that $I = \emptyset$ if $s < r$). Define

$$h(I,\mathbf{x}) = \underset{i\in I}{\arg\max}\{\widetilde{Q}(r:(i-1),\mathbf{x}) + \widetilde{Q}(i:s,\mathbf{x})\} - 1, \tag{4}$$

where also by convention $\widetilde{Q}(\emptyset,\mathbf{x}) = 0$.

The idea of this algorithm is to compute the best point of independence for the interval I (if there is one point in the interval leading to a maximum of the penalized likelihood or a point outside I otherwise) and then to iterate this criterion on both segments separated by this point, until no more points are detected. We describe the different steps of the algorithm with the following procedure:

1. Initialize $\hat{U}_{ha}(\mathbf{x}) = \emptyset$ and $I = 1:m$, with m the number of columns of $\mathbf{x}$.
2. Compute $h(I,\mathbf{x})$; if $h(I,\mathbf{x}) \in I$, add $h(I,\mathbf{x})$ to $\hat{U}_{ha}(\mathbf{x})$.
3. Repeat Step 2 for the intervals $I_1 = I\cap\{i : i \le h(I,\mathbf{x})\}$ and $I_2 = I\cap\{i : i > h(I,\mathbf{x})\}$, until no more points can be added to $\hat{U}_{ha}(\mathbf{x})$.

When m is large in relation to the sample size n, as in many real applications, both the exact dynamic programming algorithm and the hierarchical algorithm as described above are not very efficient in terms of computing time to estimate the set of points of independence. In these cases, and when a big set U with small segments' sizes is expected, we can modify the hierarchical algorithm to identify its first change point near the variable with index $m/2$, and taking into account at most $3L$ variables around this point, where L is an upper bound for the block size $|I|$, with $I \in B(U^*)$. This procedure splits the original dataset into two matrices of almost equal length, and therefore the recursive procedure is more efficient. When each split part has size smaller than $3L$, the original hierarchical algorithm is applied instead.

The modification of the hierarchical algorithm described above leads to the estimator $\hat{U}_{fha}(\mathbf{x})$ (fast hierarchical algorithm) that can be described by the following procedure. First, for $I = r:s$ with $s-r+1 > 3L$, define $i_1 = \lfloor (r+s)/2 - 3L/2\rfloor$, $i_2 = \lceil (r+s)/2 + 3L/2\rceil$, $\tilde{I} = (i_1+L):(i_2-L)$ and

$$\tilde{h}(I,\mathbf{x}) = \underset{i\in\tilde{I}}{\arg\max}\{\widetilde{Q}(i_1:i,\mathbf{x}) + \widetilde{Q}((i+1):i_2,\mathbf{x})\}.$$

1. Initialize $\hat{U}_{fha}(\mathbf{x}) = \emptyset$ and $I = 1:m$, with m the number of columns of $\mathbf{x}$.
2. If $m > 3L$ compute $\tilde{h}(I,\mathbf{x})$ and add it to $\hat{U}_{fha}(\mathbf{x})$; if $m \le 3L$ compute $h(I,\mathbf{x})$ and proceed as in the hierarchical algorithm.
3. Repeat Step 2 for the intervals $I_1 = I\cap\{i : i \le \tilde{h}(I,\mathbf{x})\}$ and $I_2 = I\cap\{i : i > \tilde{h}(I,\mathbf{x})\}$, until no more points can be added to $\hat{U}_{fha}(\mathbf{x})$.

2.3. Theoretical results

We present in this section the strong consistency of the estimator defined by (3) and computed by the dynamic programming algorithm presented in Section 2.2. We also show that the approximation given by the hierarchical algorithm is also strongly consistent when the sample size $n \to \infty$. That is, we show that for m fixed, each estimator is equal to the true set U^* eventually almost surely when n is large enough. In other words, with probability 1, there exists n_0 (depending on the infinite sample $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$) such that for all $n \ge n_0$ we have $\hat{U}(\mathbf{x}) = U^*$ (respectively $\hat{U}_{ha}(\mathbf{x}) = U^*$).

Theorem 1. *For m fixed, the estimator $\hat{U}(\mathbf{x})$ given by (3) is strongly consistent, i.e., $\hat{U}(\mathbf{x}) = U^*$ eventually almost surely when $n \to \infty$.*

The proof of Theorem 1 is postponed to the Appendix.

Although the hierarchical algorithm gives an approximate solution to the maximum in (3), we can also prove that it is a consistent estimator of U^*.

Theorem 2. *For m fixed, the estimator $\hat{U}_{ha}(\mathbf{x})$ given by the hierarchical algorithm is strongly consistent, i.e., $\hat{U}_{ha}(\mathbf{x}) = U^*$ eventually almost surely as $n \to \infty$.*

As a corollary we obtain that when the sizes of the blocks $I \in B(U^*)$ are bounded by L, the fast hierarchical algorithm is also strongly consistent.

Corollary 1. *Suppose*

$$\max_{I \in B(U^*)} (|I|) \le L.$$

Then for m fixed, the estimator $\hat{U}_{fha}(\mathbf{x})$ is strongly consistent, i.e., $\hat{U}_{fha}(\mathbf{x}) = U^*$ eventually almost surely as $n \to \infty$.

The proofs of Theorem 2 and Corollary 1 are also postponed to the Appendix.

3. Simulations

In this section we show the results of different simulation experiments to test the exact penalized maximum likelihood estimator of the set of points of independence and the hierarchical estimator, introduced in Section 2.2. We consider two different models for sequences of length $m = 15$, with two and one point of independence, respectively.

For Model 1 we consider a random vector $\mathbf{X} = (X_1, \dots, X_{15})$, where each X_i assumes values in the set $A = \{0, 1\}$, composed by three independent blocks of length 5. That is we define

$$Y_1, Y_3 \sim \mathcal{U}(A) \text{ (independent)},$$
$$Y_2 = Y_1 - Y_3 \text{ (mod) } |A|,\ Y_4 = Y_1 + Y_3 \text{ (mod) } |A|,\ Y_5 = Y_1$$

and we take $(X_1, \dots, X_5), (X_6, \dots, X_{10})$ and $(X_{11}, \dots, X_{15})$ to be independent and identically distributed, with the same distribution as $(Y_1, \dots, Y_5)$. It is clear that for this distribution the set of points of independence is $U^* = (5, 10)$.

For Model 2 we take $(X_1, \dots, X_5)$ and $(X_{11}, \dots, X_{15})$ as before, but we redefine $(X_6, \dots, X_{10})$ as $X_6 = X_{10} = X_1$, $X_7 = X_1 - X_3$ (mod) $|A|$, $X_8 = X_3$ and $X_9 = X_1 + X_3$ (mod) $|A|$. With this modification, it is then immediate that the set of points of independence becomes $U^* = (10)$. The idea of this example is to have segments with different lengths, to see the performance of the algorithms in these cases.

For each estimator we simulated n independent realizations of the vector $\mathbf{X} = (X_1, \dots, X_{15})$ for each one of the models presented above. To measure the performance of each estimator to detect the true set of points of independence we used the Hausdorff distance defined on sets, given, for all $U, V \subseteq \mathbb{R}$, by

$$d(U, V) = \max\left[\max_{u \in U}\{\min_{v \in V}(|u - v|)\}, \max_{v \in V}\{\min_{u \in U}(|u - v|)\}\right].$$

In each run we computed this measure between the estimated set of points $\hat{U}(\mathbf{x})$ (or $\hat{U}_{ha}(\mathbf{x})$ for the hierarchical algorithm) and the true set of points U^*, each point divided by the number of variables m in order to normalize the measure in some way. Moreover, without losing any information on the measurement, we added to these sets the "extreme" points 0 and 1 in order to avoid empty sets and a non-defined measure. For each algorithm we performed 100 replications and we computed the mean Hausdorff distance. The results for both models and for the alphabets $A = \{0, 1\}$ and $A = \{0, 1, 2\}$ are given in Fig. 1.

In order to evaluate the computing time of both methods on larger sequences, we implemented another simulation taking multiple alignments of different column size. The results are shown in Fig. 2, where we have a clear idea of the time requirements of both algorithms when the size of the sequences grows. The algorithms were implemented in the open-source software R and are available upon request. The time reported in these experiments was estimated on a single run of the algorithms, in a MacBook Air (11-inch, Mid 2013) with 1,3 GHz Intel Core i5 processor and running OS X 10.11.6.

4. Identifying recombination hotspots

A possible application of our method is to identify recombination hotspots in a given population. In a simplified view, linkage disequilibrium corresponds to a non-random association among consecutive loci of a short DNA sequence (haplotype block). In a population, the few common haplotypes were separated by genotype recombination hotspots across which little association remains [14,19]. Therefore, if we consider chromosome data as being completely independent, the edges of the segments should, in general, overlap with recombination hotspots.

We applied our segmentation method in the data obtained from 157 European individuals from Human Genome Diversity Project (HGDP) database, which were genotyped for a set of around 600,000 SNP markers from Illumina HuHap 650k platform [17] (HGDP dataset 2 available at ftp://ftp.cephb.fr/hgdp_supp1/). The accurate haplotype inference was made by

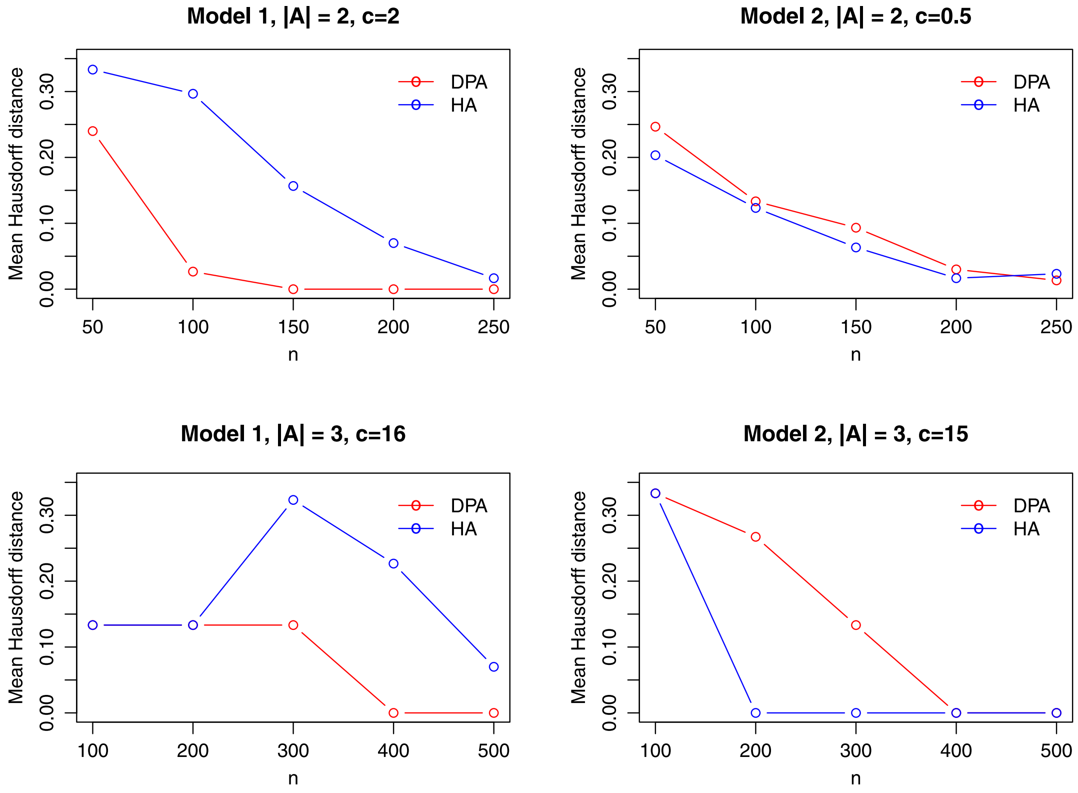

Fig. 1. Comparison between the exact dynamic programming algorithm (DPA) and the hierarchical algorithm (HA) for samples of different sizes of Model 1 (left) and Model 2 (right), with $|A| = \{0, 1\}$ (top) and $|A| = \{0, 1, 2\}$ (bottom). The figures show the mean Hausdorff distance between the true vector of points of independence and the estimated vector, for each method, on 100 independent replications, with different penalizing constants.

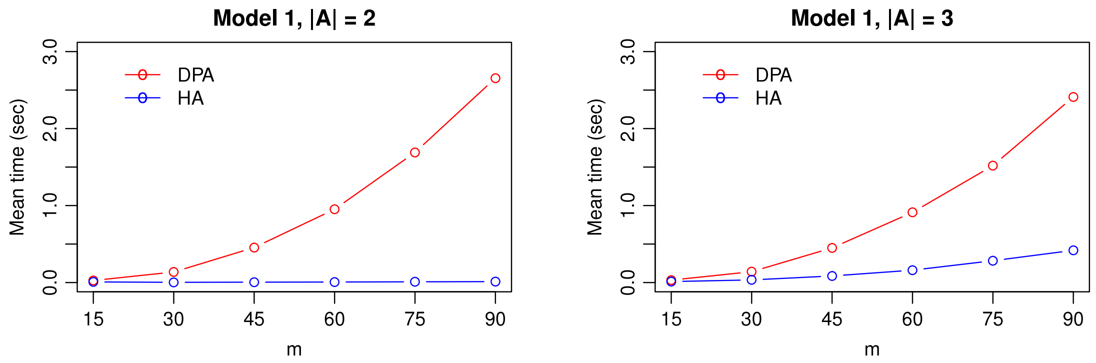

Fig. 2. Computation time of the exact dynamic programming algorithm (DPA) and the hierarchical algorithm (HA) for samples of size $n = 100$ and different sizes of the multiple alignment. The sequences were obtained by concatenating a different number of blocks from the vector $\mathbf{X}$ in Model 1, with $|A| = \{0, 1\}$ (left) and $|A| = \{0, 1, 2\}$ (right). The time was estimated in a single run of the algorithm, in a Mac Book Pro (13-inch, 2017) with 2,3 GHz Intel Core i5 processor and running OS X 10.12.6.

phasing these samples through the software Beagle v.4.1 [5]. In order to identify population recombination hotspots, the resulting phased chromosomes were considered completely independent.

Recombination rates estimated from HapMap populations [14] obtained from ftp://ftp.ncbi.nlm.nih.gov/hapmap/recombination/ were used as the "truth" to be compared to the independent segments estimated using the fast hierarchical algorithm described in Section 2.2, considering different penalizing constant values. Fig. 3 shows the values of recombination rates by locus (y axis), highlighting the independent segments obtained from HGDP European population represented by horizontal bars. To make it easier to visualize the data, we chose to show an arbitrary stretch of chromosome 1, but this pattern can also be observed throughout the genome.

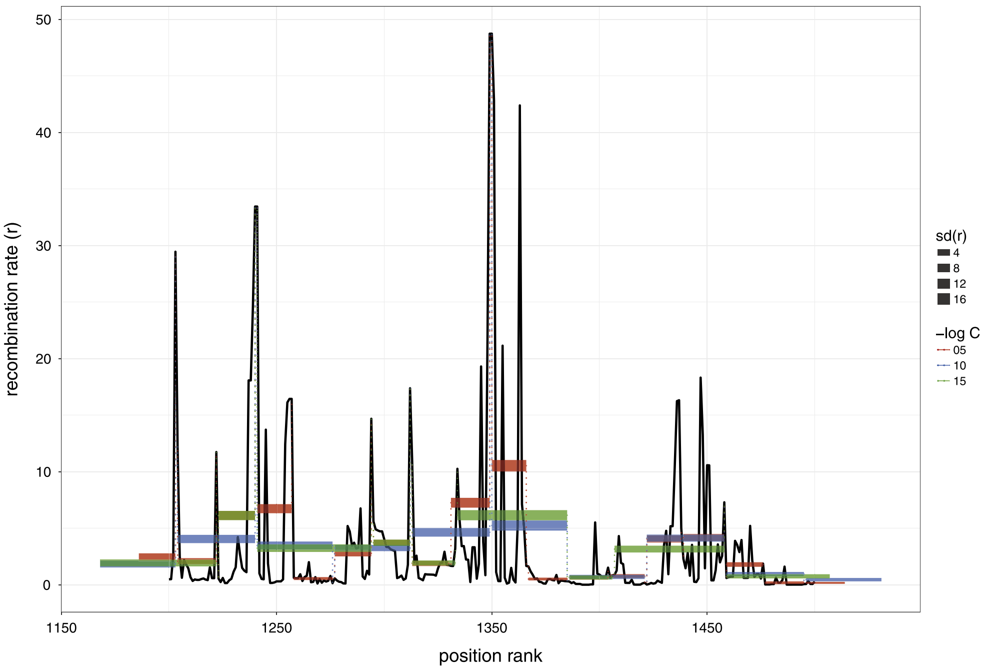

Fig. 3. Recombination rates (r) per locus (y-axis), according to its order in genomic physical position. Horizontal bars represent the putative independent segments. The y-axis position and the thickness of the bars are, respectively, the average and the standard deviation of the rates within the segments. Different bar colors denote segments created considering different penalizing constants c. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Observed recombination rate mean, bootstrap mean and p-value for different penalizing constants c. The bootstrap values were obtained by randomly selecting sets of SNPs in the entire dataset of the same size as the corresponding estimated sets of points of independence.

Penalizing constant c	Observed mean	Bootstrap mean	Bootstrap p-value
0.1	3.779764	2.498357	0.0
0.01	3.547373	2.498901	0.0
0.001	4.348381	2.498227	0.0
1e−04	4.528738	2.498667	0.0
1e−05	3.447744	2.497623	0.0
1e−06	4.575773	2.498265	0.0
1e−07	5.255874	2.497928	0.0
1e−08	5.481127	2.498848	0.0
1e−09	4.790805	2.499632	0.0
1e−10	3.274655	2.498607	2e−04
1e−11	4.085479	2.499059	0.0
1e−12	4.515567	2.498170	0.0
1e−13	4.757720	2.498965	0.0
1e−14	4.897905	2.497790	0.0
1e−15	4.321468	2.497402	0.0

From Fig. 3 one can see that most segment edges rely on SNPs with high observed recombination rates, suggesting that our method could be a good predictor of these regions. The sizes of the windows are highly dependent on the values of the penalyzing constant c, since lower c values create larger segments. We tested the hypothesis of a significantly different recombination rate mean at the edges of the estimated blocks for constants c ranging from 10^{-1} to 10^{-15}. We did this by a usual bootstrap test with 100,000 random samples extracted from the entire chromosome and with the same length as the estimated vector. The estimated recombination rates and the corresponding p-values for each constant are summarized in Table 1. We suggest that the selection of the optimal c value should be more deeply investigated in further studies.

5. Discussion

In this paper we presented a novel and simple method to identify independent blocks in multiple aligned sequences. One of the main advantages of our approach is its generality, in the sense that we do not assume a maximal number of segments and we do not impose a specific model on each segment; the method estimates the nonparametric multivariate distribution on each segment and the optimal number of such segments. But the use of specific models on each segment as in [3,10] is also possible and should be more explored in the future, enabling the generalization of the method to a high dimensional setting, in which the dimension of sequences could grow with the sample size. Moreover, the method and the theoretical results could also be extended to the case considered in [12], where a unique stationary sequence is modeled by a concatenation of Variable Length Markov chains. The use of other penalizing functions is also possible because the main argument in the proof of the consistency results are the asymptotic bounds given by Lemmas 3 and 4. Any penalizing term lying between these two bounds could also result in consistent estimators.

By the computational complexity, the exact algorithm seems to be only appropriate for sequences of relatively small length. In contrast, the approximate hierarchical algorithm and its faster version assuming an upper bound in the block's size can be applied to very big sequences, as in the case of the SNPs data analyzed in Section 4. Nevertheless the algorithms used in our analyses were coded in the R language to illustrate the performance of the estimators on simulated data and to show the potential applicability of the method, but the computational efficiency was not the primarily goal of our contribution. Faster algorithms could certainly be developed in the future, by using more sophisticated programming techniques.

The exploratory application of our method showed that the average of recombination rate in the edges of the segments obtained considering different penalizing constant values are significantly different than the one obtained over the genome. This is consistent with what is expected for recombination hotspots. The lengths of the resulting segments are highly dependent on the value of the constant c, and there is no standard methodology to optimize this value. While in the experiments presented in this article we chose rather arbitrary constants, the search for a "constant-free" method to estimate the vector of points of independence is desirable. This could be addressed, e.g., by using a similar approach as in [11]. This constitutes a goal for future research in this area.

Acknowledgments

Florencia Leonardi is partially supported by a CNPq-Brazil fellowship (304836/2012-5). This article was produced as part of the activities of FAPESP Research, Innovation and Dissemination Center for Neuromathematics, grant 2013/07699-0, and FAPESP's project "Structure selection for stochastic processes in high dimensions", grant 2016/17394-0, São Paulo Research Foundation.

Appendix. Proof of the theoretical results

Let $I \subset 1:m$ be an integer interval. Given two probability distributions P_1 and P_2 over A^I, let $D(P_1 \parallel P_2)$ denote the Kullback–Leibler divergence between P_1 and P_2, i.e.,

$$D(P_1 \parallel P_2) = \sum_{a_I : P_2(a_I) > 0} P_1(a_I) \ln \{P_1(a_I)/P_2(a_I)\},$$

where, by convention, $0 \ln 0/p = 0$ for all $p \in (0, 1]$. A well-known property about the Kullback–Leibler divergence between two probability distributions states that $D = 0$ if and only if $P_1 = P_2$.

For any $j \in 1:m$ denote by $\widetilde{\Pr}_j$ the probability distribution given, for all $a_{1:m} \in A^{1:m}$, by

$$\widetilde{\Pr}_j(a_{1:m}) = \Pr(a_{1:j}) \Pr(a_{(j+1):m}),$$

and let

$$\alpha = \min\{D(\Pr \parallel \widetilde{\Pr}_j) : j \notin U^*\}. \tag{A.1}$$

By the definition of $\widetilde{\Pr}_j$ and the basic property of the Kullback–Leibler divergence we must have $\alpha > 0$.

We state without proof a basic lemma from [7] (see Lemma 6.3 therein) that will be useful later.

Lemma 1. *For probability distributions P_1 and P_2 on A^I,*

$$D(P_1 \parallel P_2) \le \sum_{a_I : P_2(a_I) > 0} \frac{|P_1(a_I) - P_2(a_I)|^2}{P_2(a_I)}.$$

Now we prove a lemma that provides us the rate of convergence of $D(\widehat{\Pr} \parallel \Pr)$ to 0 when $n \to \infty$.

Lemma 2. *For any interval $I \subset 1:m$ we have $D(\widehat{\Pr} \parallel \Pr) < 8|A^I| \ln\ln(n)/n$, eventually almost surely as $n \to \infty$.*

Proof. Define, for a fixed $a_I \in A^I$, the random variables given, for $i \in \{1, \dots, n\}$, by

$$Y_i(a_I) = \mathbf{1}\{\mathbf{x}_I^{(i)} = a_I\} - \Pr(a_I),$$

and

$$Z_n(a_I) = \sum_{i=1}^{n} Y_i(a_I) = N(a_I) - n\Pr(a_I).$$

The variables $Y_1(a_I), \ldots, Y_n(a_I)$ are independent and identically distributed, with $\mathrm{E}\{Y_i(a_I)\} = 0$ and $\mathrm{E}\{Y_i(a_I)^2\} = \Pr(a_I)\{1 - \Pr(a_I)\}$. Then, by the Law of the Iterated Logarithm (see Theorem 3.52 in [4]) we have, for all $\epsilon > 0$,

$$|Z_n(a_I)| < (1+\epsilon)\Pr(a_I)\{1-\Pr(a_I)\}\sqrt{2n\ln\ln n}$$

eventually almost surely as $n \to \infty$. Dividing both sides of the last inequality by $n\sqrt{\Pr(a_I)}$ and taking $\epsilon = 1$ we obtain

$$\frac{|Z_n(a_I)/n|}{\sqrt{\Pr(a_I)}} = \frac{|\widehat{\Pr}(a_I) - \Pr(a_I)|}{\sqrt{\Pr(a_I)}} < 2\sqrt{\frac{2\ln\ln(n)}{n}} \tag{A.2}$$

eventually almost surely as $n \to \infty$. Now by Lemma 1 and (A.2) we have

$$D(\widehat{\Pr}|\Pr) \le \sum_{a_I:\Pr(a_I)>0} \frac{|\widehat{\Pr}(a_I) - \Pr(a_I)|^2}{\Pr(a_I)} \le |A^I| \max_{a_I} \frac{|\widehat{\Pr}(a_I) - \Pr(a_I)|^2}{\Pr(a_I)} \le \frac{8|A^I|\ln\ln(n)}{n}$$

eventually almost surely as $n \to \infty$ and this completes the proof. □

Lemma 3. *Let $I = r:s \subset 1:m$ and suppose $(I \setminus \{s\}) \cap U^* = \emptyset$, i.e., there is no point of independence belonging to $I \setminus \{s\}$. Then*

$$\min_{i \in I\setminus\{s\}} \{Q(I;\mathbf{x}) - Q(r:i;\mathbf{x}) - Q((i+1):s;\mathbf{x})\} > \alpha n$$

eventually almost surely when $n \to \infty$, where α is given by (A.1).

Proof. Note that for any $a_{r:i} \in A^{r:i}$ we have $N(a_{r:i}) = \sum_{a_{(i+1):s}} N(a_{r:i}a_{(i+1):s})$ and analogously for any $a_{(i+1):s} \in A^{(i+1):s}$, $N(a_{(i+1):s}) = \sum_{a_{r:i}} N(a_{r:i}a_{(i+1):s})$. Then we can write

$$Q(I;\mathbf{x}) = \sum_{a_{r:s}\in A^I} N(a_{r:s})\ln\widehat{\Pr}(a_{r:s})$$

and

$$Q(r:i;\mathbf{x}) + Q((i+1):s;\mathbf{x}) = \sum_{a_{r:s}\in A^I} N(a_{r:s})\ln\widehat{\Pr}(a_{r:i})\widehat{\Pr}(a_{(i+1):s}),$$

therefore

$$Q(I;\mathbf{x}) - Q(r:i;\mathbf{x}) - Q((i+1):s;\mathbf{x}) = \sum_{a_{r:s}\in A^I} N(a_{r:s})\ln\left\{\frac{\widehat{\Pr}(a_{r:s})}{\widehat{\Pr}(a_{r:i})\widehat{\Pr}(a_{(i+1):s})}\right\}.$$

Dividing by n and taking limit when $n \to \infty$ we have that the expression above converges almost surely to

$$\sum_{a_{r:s}\in A^I} \Pr(a_{r:s})\ln\left\{\frac{\Pr(a_{r:s})}{\Pr(a_{r:i})\Pr(a_{(i+1):s})}\right\} = D(\Pr \parallel \widetilde{\Pr}_i) \ge \alpha > 0. \quad \square$$

Lemma 4. *Let $I = r:s \subset 1:m$ and suppose there exists $i \in (I \setminus \{s\}) \cap U^*$, i.e., there is a point of independence in the interval I. Then we have $Q(I;\mathbf{x}) - Q(r:i;\mathbf{x}) - Q((i+1):s;\mathbf{x}) < 8|A^I|\ln\ln(n)$ eventually almost surely when $n \to \infty$.*

Proof. As in the proof of Lemma 3 we can write

$$Q(I;\mathbf{x}) = \sum_{a_{r:s}\in A^I} N(a_{r:s})\ln\widehat{\Pr}(a_{r:s})$$

and

$$Q(r:i;\mathbf{x}) + Q((i+1):s;\mathbf{x}) = \sum_{a_{r:s}\in A^I} N(a_{r:s})\ln\widehat{\Pr}(a_{r:i})\widehat{\Pr}(a_{(i+1):s}).$$

As $\widehat{\Pr}(\cdot)$ is the maximum likelihood estimator of $\Pr(\cdot)$ and i is a point of independence, we have

$$Q(r:i;\mathbf{x}) + Q((i+1):s;\mathbf{x}) \ge \sum_{a_{r:s\in A^I}} N(a_{r:s})\ln\Pr(a_{r:i})\Pr(a_{(i+1):s}) = \sum_{a_{r:s}\in A^I} N(a_{r:s})\ln\Pr(a_{r:s}).$$

Then by combining this last inequality and Lemma 2 we have

$$Q(I;\mathbf{x}) - Q(r:i;\mathbf{x}) - Q((i+1):s;\mathbf{x}) \le \sum_{a_{r:s}\in A^I} N(a_{r:s})\ln\frac{\widehat{\Pr}(a_{r:s})}{\Pr(a_{r:s})} = nD(\widehat{\Pr} \parallel \Pr) < 8|A^I|\ln\ln(n)$$

eventually almost surely as $n \to \infty$. □

Proof of Theorem 1. We will show that eventually almost surely the maximizer of (3) is U^*. Define the sets $E_1 = \{U : U^* \not\subseteq U\}$ and $E_2 = \{U : U^* \subseteq U\}$. First we will prove that eventually almost surely $\hat{U}(\mathbf{x})$ must belong to E_2. Let $U \in E_1$ and assume $U^* \neq \emptyset$, because if $U^* = \emptyset$ then $E_1 = \emptyset$ and the assertion is trivial. Assume $U^* = (u_1^*, \ldots, u_K^*)$, with $K \geq 1$, $U = (u_1, \ldots, u_k)$, with $k \geq 0$, and let i be the first index such that $u_i^* \notin U$. Define $U' = U \cup \{u_i^*\} = (u_1, \ldots, u_{j-1}, u_i^*, u_j, \ldots, u_k)$; see Fig. 4. Note that $B(U')$ has the two blocks $I_1 = u_{j-1} : u_i^*$ and $I_2 = (u_i^* + 1) : u_j$ replacing the single block $I = u_{j-1} : u_j$ in $B(U)$. Therefore,

$$\hat{\ell}(U'; \mathbf{x}) - \hat{\ell}(U; \mathbf{x}) = Q(I_1, \mathbf{x}) + Q(I_2, \mathbf{x}) - Q(I, \mathbf{x})$$

and by Lemma 4 we have

$$\hat{\ell}(U'; \mathbf{x}) - \hat{\ell}(U; \mathbf{x}) > -8|A^I| \ln\ln(n)$$

eventually almost surely as $n \to \infty$. Moreover,

$$c(|U'| + 1)\sqrt{n} - c(|U| + 1)\sqrt{n} = c\sqrt{n}$$

then

$$\mathrm{PML}(U'; \mathbf{x}) - \mathrm{PML}(U; \mathbf{x}) > c\sqrt{n} - 8|A^I| \ln\ln(n) > 0$$

eventually almost surely as $n \to \infty$. Therefore the penalized log-likelihood increases when we add the missing points in U^* to any set $U \in E_2$, showing that the global maximizer of (3), given by $\hat{U}(\mathbf{x})$, must belong to E_2 eventually almost surely when $n \to \infty$. Now we will show that in E_2 there is a global maximizer of the penalized log-likelihood given by U^*, and this will imply the result. To show this assume $U^* = (u_1^*, \ldots, u_K^*)$, with $K \geq 0$, and let $U = (u_1, \ldots, u_k) \in E_2$, with $k \geq 1$. As before let i_1 be the first index such that $u_{i_1} \notin U^*$. Define $U^{(0)} = U$ and $U^{(1)} = U \setminus \{u_{i_1}\} = (u_1, \ldots, u_{i_1-1}, u_{i_1+1}, \ldots, u_k)$; see Fig. 5. In this case we have that $B(U^{(1)})$ has the single block $I = (u_{i_1-1} + 1) : u_{i_1+1}$ replacing the two blocks $I_1 = (u_{i_1-1} + 1) : u_{i_1}$ and $I_2 = (u_{i_1} + 1) : u_{i_1+1}$ in $B(U^{(0)})$. Then

$$\hat{\ell}(U^{(1)}; \mathbf{x}) - \hat{\ell}(U^{(0)}; \mathbf{x}) = Q(I, \mathbf{x}) - Q(I_1, \mathbf{x}) - Q(I_2, \mathbf{x}).$$

By (A.1) and Lemma 3, as $u_{i_1} \notin U^*$ we have that eventually almost surely

$$\hat{\ell}(U^{(1)}; \mathbf{x}) - \hat{\ell}(U^{(0)}; \mathbf{x}) > \alpha n.$$

Moreover we have

$$c(|U^{(1)}| + 1)\sqrt{n} - c(|U^{(0)}| + 1)\sqrt{n} = -c\sqrt{n}$$

then

$$\mathrm{PML}(U^{(1)}; \mathbf{x}) - \mathrm{PML}(U^{(0)}; \mathbf{x}) > \alpha n - c\sqrt{n}.$$

By iterating this procedure and removing all the points $u \in U \setminus U^*$ we obtain the sequence of vectors $U^{(0)}, \ldots, U^{(s)}$, with $s \leq m$, satisfying $U^{(0)} = U$, $U^{(m)} = U^*$ and

$$\mathrm{PML}(U^*; \mathbf{x}) - \mathrm{PML}(U; \mathbf{x}) = \sum_{j=0}^{s-1} \mathrm{PML}(U^{(j+1)}; \mathbf{x}) - \mathrm{PML}(U^{(j)}; \mathbf{x}) > \alpha n - c\sqrt{n} > 0$$

eventually almost surely as $n \to \infty$. Therefore we have

$$\mathrm{PML}(U^*; \mathbf{x}) > \max_{U \in E_2 \setminus \{U^*\}} \mathrm{PML}(U; \mathbf{x})$$

eventually almost surely as $n \to \infty$, and then $\hat{U} = U^*$. This concludes the proof of Theorem 1. □

Proof of Theorem 2. First consider the case $U^* = \emptyset$. By Lemma 3 we have that eventually almost surely as $n \to \infty$ the value of i maximizing (4) will be $i = 1$, giving $h(1 : m, \mathbf{x}) = 0$. Therefore $\hat{U}_{ha}(\mathbf{x}) = \emptyset = U^*$ eventually almost surely as $n \to \infty$. Now suppose there is a point of independence $u \in U^*$. We will prove that eventually almost surely $u \in \hat{U}_{ha}(\mathbf{x})$. As $u \in U^*$, for any integer $r < u$ and any integer $s > u$ we have by Lemma 4 that

$$Q(r : u, \mathbf{x}) + Q((u + 1) : s, \mathbf{x}) - Q(r : s, \mathbf{x}) > -8|A^I| \ln\ln(n)$$

eventually almost surely as $n \to \infty$. Then eventually we will have

$$Q(r : u, \mathbf{x}) + Q((u + 1) : s, \mathbf{x}) - Q(r : s, \mathbf{x}) > -c\sqrt{n}$$

and therefore $h(r : s, \mathbf{x}) = u$, or equivalently $u \in \hat{U}_{ha}(\mathbf{x})$. As in any iteration of the algorithm there is an interval that contains u and the inequality above is true for any $r < u$ and $s > u$, then $u \in \hat{U}_{ha}(\mathbf{x})$ eventually almost surely as $n \to \infty$. □

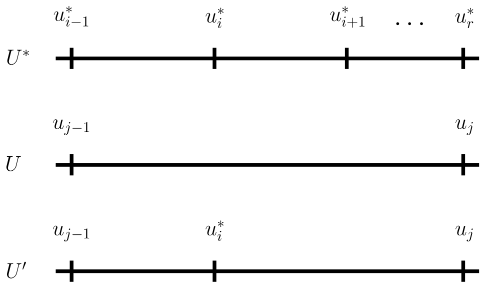

Fig. 4. A generic set $U \in E_1$, with $U^* \not\subseteq U$, and the corresponding modification $U' = U \cap \{u^*_i\}$.

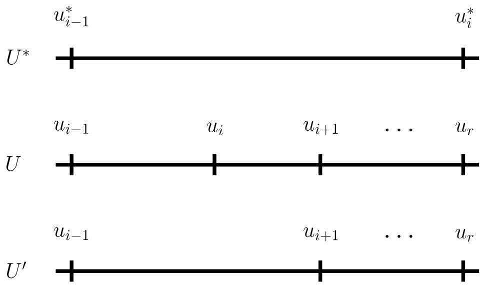

Fig. 5. A generic set $U \in E_2$, with $U^* \subsetneq U$, and the corresponding modification $U' = U \setminus \{u_i\}$.

Proof of Corollary 1. Follows directly by an adaptation of the proof of Theorem 2, by observing that when $m > 3L$ and

$$\max_{I \in B(U^*)} (|I|) \leq L$$

then there must exist at least one point $u \in U \cap \tilde{I}$. For n big enough this point is detected by the algorithm and the consistency follows by the iteration of this argument on the split samples. □

References

[1] M. Algama, J.M. Keith, Investigating genomic structure using changept: A Bayesian segmentation model, Comput. Struct. Biotechnol. J. 10 (2014) 107–115.
[2] J. Bai, P. Perron, Computation and analysis of multiple structural change models, J. Appl. Econometrics 18 (2003) 1–22.
[3] R.J. Boys, D.A. Henderson, A Bayesian approach to DNA sequence segmentation, Biometrics 60 (2004) 573–581.
[4] L. Breiman, Probability, SIAM, Philadelphia, PA, 1992.
[5] S.R. Browning, B.L. Browning, Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering, Am. J. Hum. Genet. 81 (2007) 1084–1097.
[6] C.C. Chang, C.C. Chow, L.C. Tellier, S. Vattikuti, S.M. Purcell, J.J. Lee, Second-generation PLINK: Rising to the challenge of larger and richer datasets, GigaScience 4 (2015) 7.
[7] I. Csiszar, Z. Talata, Context tree estimation for not necessarily finite memory processes, via BIC and MDL, IEEE Trans. Inform. Theory 52 (2006) 1007–1016.
[8] S. Deng, Y. Shi, L. Yuan, Y. Li, G. Ding, Detecting the borders between coding and non-coding DNA regions in prokaryotes based on recursive segmentation and nucleotide doublets statistics, BMC Genomics 13 (2012) S19.
[9] A. Finkelstein, M. Roytberg, Computation of biopolymers: A general approach to different problems, Biosystems 30 (1993) 1–19.
[10] J. Fridlyand, A.M. Snijders, D. Pinkel, D.G. Albertson, A.N. Jain, Hidden Markov models approach to the analysis of array CGH data, J. Multivariate Anal. 90 (2004) 132–153.
[11] A. Galves, C. Galves, J.E. García, N.L. Garcia, F. Leonardi, Context tree selection and linguistic rhythm retrieval from written texts, Ann, Appl. Stat. 6 (2012) 186–209.
[12] R. Gwadera, A. Gionis, H. Mannila, Optimal segmentation using tree models, Knowl. Inf. Syst. 15 (2008) 259–283.
[13] N. Haiminen, H. Mannila, Discovering isochores by least-squares optimal segmentation, Gen 394 (2007) 53–60.
[14] International HapMap Consortium, A second generation human haplotype map of over 3.1 million SNPs, Nature 449 (2007) 851–861.
[15] D.M. Hawkins, Point estimation of the parameters of piecewise regression models, J. Roy. Statist. Soc. Ser. C Appl. Statist. 25 (1976) 51–57.
[16] C.J. Hoggart, J.C. Whittaker, M. De Iorio, D.J. Balding, Simultaneous analysis of all SNPs in genome-wide and re-sequencing association studies, PLoS Genet. 4 (2008) 1–8.
[17] J.Z. Li, D.M. Absher, H. Tang, A.M. Southwick, A.M. Casto, S. Ramachandran, H.M. Cann, G.S. Barsh, M. Feldman, L.L. Cavalli-Sforza, R.M. Myers, Worldwide human relationships inferred from genome-wide patterns of variation, Science 319 (2008) 1100–1104.
[18] W. Li, P. Bernaola-Galván, F. Haghighi, I. Grosse, Applications of recursive segmentation to the analysis of DNA sequences, Comput. Chem. 26 (2002) 491–510.

[19] S. Pääbo, The mosaic that is our genome, Nature 421 (2003) 409.
[20] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M.A. Ferreira, D. Bender, J. Maller, P. Sklar, P.I. de Bakker, M.J. Daly, P.C. Sham, PLINK: A tool set for whole-genome association and population-based linkage analyses, Am. J. Hum. Genet. 81 (2007) 559–575.
[21] V.E. Ramensky, V.J. Makeev, M.A. Roytberg, V.G. Tumanyan, DNA segmentation through the Bayesian approach, J. Comput. Biol. 7 (2000) 215–231.
[22] F. Tajima, Determination of window size for analyzing DNA sequences, J. Mol. Evol. 33 (1991) 470–473.
[23] S.-Y. Wen, C.-T. Zhang, Identification of isochore boundaries in the human genome using the technique of wavelet multiresolution analysis, Biochem. Biophys. Res. Commun. 311 (2003) 215–222.