A parallelizable framework for segmenting piecewise signals

Junbo Duan, Charles Soussen, David Brie, Jérôme Idier, Yu-Ping Wang, Mingxi Wan

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 24, 2018, accepted December 14, 2018, date of publication December 28, 2018, date of current version February 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2890077

A Parallelizable Framework for Segmenting Piecewise Signals

JUNBO DUAN[1], (Member, IEEE), CHARLES SOUSSEN[2],(Member, IEEE), DAVID BRIE[3], (Member, IEEE), JÉRÔME IDIER[4], (Member, IEEE), YU-PING WANG[5], (Senior Member, IEEE), AND MINGXI WAN[1], (Member, IEEE)

[1]Department of Biomedical Engineering, Xi'an Jiaotong University, Xi'an, China
[2]Laboratoire des Signaux et Systèmes, CentraleSupélec-CNRS-Université Paris-Sud, Université Paris-Saclay, 91192 Gif-sur-Yvette, France
[3]Centre de Recherche en Automatique de Nancy, Université de Lorraine, CNRS, France
[4]Laboratoire des Sciences du Numériques de Nantes, Ecole Centrale de Nantes, Nantes, France
[5]Department of Biomedical Engineering, Tulane University, New Orleans, LA, USA

Corresponding author: Junbo Duan (junbo.duan@mail.xjtu.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61771381 and Grant 61401352, in part by the China Postdoctoral Science Foundation under Grant 2014M560786, in part by the National Institutes of Health under Grant R01GM109068, Grant R01MH104680, and Grant R01MH107354, and in part by the National Science Foundation under Grant 1539067.

ABSTRACT Piecewise signals appear in many application fields. Here, we propose a framework for segmenting such signals based on the modeling of each piece using a parametric probability distribution. The proposed framework first models the segmentation as an optimization problem with sparsity regularization. Then, an algorithm based on dynamic programming is utilized for finding the optimal solution. However, dynamic programming often suffers from a heavy computational burden. Therefore, we further show that the proposed framework is parallelizable and propose using GPU-based parallel computing to accelerate the computation. This approach is highly desirable for the analysis of large volumes of data that are ubiquitous. The experiments on both the simulated and real genomic datasets from the next-generation sequencing demonstrate an improved performance in terms of both segmentation quality and computational speed.

INDEX TERMS Parallel computing, piecewise distribution, segmentation algorithm, dynamic programming, next generation sequencing.

I. INTRODUCTION

Piecewise signals appear in many fields of applications, and a piecewise signal consists of several pieces, and each piece obeys a distinct parameterized distribution. Specifically, the parameters of a distribution vary with each piece.

In communication, a random telegraph signal can be modeled as a piecewise constant signal that is contaminated with i.i.d. zero mean Gaussian noise. It is actually a piecewise signal, whose pieces follow a Gaussian distribution with the same variance but different means. In genetics, microarray data are also modeled as piecewise constant signals contaminated with Gaussian noise [1]. The read depth signal, which is generated by next generation sequencing (NGS) techniques, was modeled as a piecewise Poisson distributed signal in our previous work [2]. The segmentation of both microarray data and read depth signals helps us detect the so-called copy number variation (CNV) [3], [4], which is an important biomarker frequently observed in human genomes and associated with complex diseases.

The segmentation of piecewise signals has been studied for decades, and the key problem is to detect the breakpoints between consecutive pieces. Circular binary segmentation (CBS) [5], which was originally designed for array-based DNA CNV detection, is one of the most commonly used algorithms and is based on hypothesis testing. High noise levels significantly degenerate the specificity of detection. Thus, signals must be smoothed in preprocessing, but at the cost of reduced detection sensitivity. As a result, several more advanced methods have been proposed for smoothing signals and detecting breakpoints simultaneously. The total variation method [6], which uses a total variation term for smoothing, has been found to be robust in detecting breakpoints. Nikolova [7] showed that when the penalty potential is nonsmooth at the origin, *e.g.*, ℓ_1 norm, then breakpoints can be preserved while smoothing a locally homogenous signal. Therefore, several methods that involve an ℓ_1 norm penalty have been proposed. Harchaoui and Lévy-Leduc [8] proposed detecting breakpoints with LASSO; Tibshirani and Wang [9]

proposed the fused LASSO for smoothing piecewise constant signals; and Kim *et al.* [10] proposed the ℓ_1 trend filter for smoothing piecewise linear signals. Since an ℓ_1-norm-penalty-based solution is biased [11], the ℓ_0-norm penalty is used as an alternative [12], [13]. Such problem can be viewed as model complexity problems. Thus, model selection criteria such as the Akaike information criterion (AIC) [14] and the Bayesian information criterion (BIC) [15] can be employed.

Here, we propose a framework for segmenting piecewise signals. The proposed framework first models each piece using a parametric probability distribution. Then, the maximum likelihood estimator with sparsity regularization is derived as the optimization criterion for segmentation. An algorithm based on dynamic programming is proposed for finding the optimal solution. However, dynamic programming often suffers from a heavy computational burden, and this issue is exacerbated for high-dimensional data such as genomic data arising from NGS. For example, chromosome 22, which is the shortest one in humans, consists of 47 million base pairs. To accelerate computation, we further show that the proposed framework is parallelizable, and we employ GPU-based parallel computing.

The rest of the paper is organized as follows: in Sec. II, we present the general optimization criterion (Subsec. II-A), and then apply the general criterion to multiple specific problems (Subsec. II-B). In Sec. III, we present the general algorithm (Subsec. III-B) and an accelerated version with GPU-based parallel computing (Subsec. III-C) and discuss related issues (Subsecs. III-E, III-D, and III-F). In Sec. IV, the segmentation (Subsec. IV-A) and computational performances (Subsec. IV-B) are tested on both simulated and real data. The conclusions of the study are presented in the last section, followed by an appendix (A) describing the computation of a specific example.

II. PROPOSED FRAMEWORK

A. DEFINITION OF PIECEWISE DISTRIBUTED SIGNALS

A signal $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T$ is assumed to be piecewise distributed if the following holds:

$$p(\boldsymbol{y}|\boldsymbol{\Theta}, \mathcal{I}) = \prod_{k=1}^{K} f_k(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k),$$

where $k = 1, 2, \ldots, K$ denotes the segment index, K is the number of total segments, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$, $\boldsymbol{\theta}_k$ consists of the parameter(s) of the kth segment, segmentation $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_K\}$, $\mathcal{I}_k$ consists of the indices i that belong to the kth segment, and $\boldsymbol{y}_{\mathcal{I}_k} = \{y_i | i \in \mathcal{I}_k\}$.

If the segmentation $\mathcal{I}$ is known *a priori*, *i.e.,* the locations of the breakpoints are known in advance, then for a specified segment k the maximal likelihood estimator of $\boldsymbol{\theta}_k$ is expressed as

$$\boldsymbol{\theta}_k^* = \arg\max_{\boldsymbol{\theta}_k} f_k(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k).$$

By taking the negative natural logarithm, this estimator is equivalent to

$$\boldsymbol{\theta}_k^* = \arg\min_{\boldsymbol{\theta}_k} \varepsilon(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k), \tag{1}$$

where

$$\varepsilon(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k) \triangleq -\ln\left(f_k(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k)\right).$$

As a result, the minimum at the kth segment is

$$\varepsilon_k^* \triangleq \varepsilon(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k^*)$$

However, in practice, K is usually unknown, not to say $\mathcal{I}$, which determines the locations of the breakpoints. If we assume that $\mathcal{I}$ is known, then K is the cardinality of set $\mathcal{I}$, which is denoted by $|\mathcal{I}| = K$ (note that $\mathcal{I}$ is a set whose elements are also sets ($\mathcal{I}_k$)). Following the law of *Occam's razor*, which is a widely used approach, each segment is penalized by λ to reduce the number of segments. A large penalty discourages segmentation. Therefore, the estimator of $\boldsymbol{\Theta}$ and $\mathcal{I}$ is

$$(\boldsymbol{\Theta}^*, \mathcal{I}^*) = \arg\min_{\boldsymbol{\Theta}, \mathcal{I}} \left\{ \sum_{k=1}^{K} \varepsilon(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k) + \lambda K \right\}$$

In the piecewise-distributed model, each segment can be estimated separately if $\mathcal{I}$ is known. Therefore, the joint estimation problem can be decomposed into K separate subproblems, with each yielding an estimator as defined in Eq. (1). Consequently,

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} \left\{ \sum_{k=1}^{K} \varepsilon_k^* + \lambda K \right\}. \tag{2}$$

In Sec. III, an algorithm that is based on dynamic programming is presented for solving Eq. (2) after all possible ε_k^* values have been precomputed. However, before proceeding, certain examples are discussed.

B. EXAMPLES

Several real application signals can be solved by this model. In the following, we discuss three of them.

1) PIECEWISE POISSON PROCESS

In the NGS techniques, the read depth signal [16] is typically derived, based on which the CNV can be detected [3]. The read depth signal is assumed to be a piecewise Poisson process [17], [18], *i.e.,* $y_i \sim Poisson(\tau_k)$, $i \in \mathcal{I}_k$:

$$f_k(y_i|\boldsymbol{\theta}_k) = \frac{\tau_k^{y_i} e^{-\tau_k}}{y_i!},$$

where τ_k is the Poisson parameter of the kth segment.

In this example, since the Poisson distribution has only one parameter, $\boldsymbol{\theta}_k = \tau_k$ and the negative natural logarithm likelihood function is expressed as

$$\varepsilon(\boldsymbol{y}_{\mathcal{I}_k}|\boldsymbol{\theta}_k) = \sum_{i \in \mathcal{I}_k} \left(-y_i \ln \tau_k + \tau_k + \ln(y_i!)\right),$$

and the maximal likelihood estimator is

$$\boldsymbol{\theta}_k^* = \tau_k^* = \overline{\boldsymbol{y}_{\mathcal{I}_k}},$$

where

$$\overline{\boldsymbol{y}_{\mathcal{I}_k}} \triangleq \frac{\sum_{i\in\mathcal{I}_k} y_i}{|\mathcal{I}_k|},$$

and $|\mathcal{I}_k|$ is the cardinality of set $\mathcal{I}_k$. Thus,

$$\varepsilon_k^* = |\mathcal{I}_k|\overline{\boldsymbol{y}_{\mathcal{I}_k}}(1 - \ln \overline{\boldsymbol{y}_{\mathcal{I}_k}}) + \sum_{i\in\mathcal{I}_k} \ln(y_i!), \tag{3}$$

and the incarnation of criterion Eq. (2) in this example is

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} \left\{ \sum_{k=1}^{K} \left(-|\mathcal{I}_k|\overline{\boldsymbol{y}_{\mathcal{I}_k}} \ln \overline{\boldsymbol{y}_{\mathcal{I}_k}}\right) + \lambda K + cst_1 \right\},$$

where the constant $cst_1 = \sum_{i=1}^{N}(y_i + \ln y_i!)$ does not depend on $\mathcal{I}$ or $\boldsymbol{\Theta}$.

2) PIECEWISE CONSTANT SIGNALS CONTAMINATED BY GAUSSIAN NOISE WITH DISTINCT VARIANCES

Suppose that $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^T$ is a piecewise constant signal, *i.e.*, $x_i = \mu_k, i \in \mathcal{I}_k$, and is contaminated by Gaussian noise n_i, which is also piecewise distributed with zero mean and variance σ_k^2, *i.e.*, $n_i \sim \mathcal{N}(0, \sigma_k^2), i \in \mathcal{I}_k$. As a result, the observation $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ follows $y_i \sim \mathcal{N}(\mu_k, \sigma_k^2), i \in \mathcal{I}$.

If $\mathcal{I}$ is known, then each segment parameter $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$ can be estimated as

$$\mu_k^* = \overline{\boldsymbol{y}_{\mathcal{I}_k}}, {\sigma_k^2}^* = \tilde{\sigma}^2_{\boldsymbol{y}_{\mathcal{I}_k}} \triangleq \frac{1}{|\mathcal{I}_k|}\sum_{i\in\mathcal{I}_k}(y_i - \overline{\boldsymbol{y}_{\mathcal{I}_k}})^2.$$

Therefore,

$$\varepsilon_k^* = |\mathcal{I}_k| \ln(\sqrt{2\pi e}\tilde{\sigma}_{\boldsymbol{y}_{\mathcal{I}_k}}), \tag{4}$$

and

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} \left\{ \sum_{k=1}^{K} |\mathcal{I}_k| \ln(\sqrt{2\pi e}\tilde{\sigma}_{\boldsymbol{y}_{\mathcal{I}_k}}) + \lambda K \right\}.$$

3) PIECEWISE CONSTANT SIGNALS CONTAMINATED BY GAUSSIAN NOISE WITH THE SAME VARIANCE

If all segments share the same variance σ, *i.e.*, $\sigma_1 = \sigma_2 = \ldots = \sigma_K = \sigma$, then we have

$$\varepsilon_k^* = \frac{|\mathcal{I}_k|\tilde{\sigma}^2_{\boldsymbol{y}_{\mathcal{I}_k}}}{2\sigma^2} + |\mathcal{I}_k| \ln(\sqrt{2\pi}\sigma), \tag{5}$$

and the segmentation problem becomes

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} \left\{ \sum_{i=1}^{K} \frac{|\mathcal{I}_k|\tilde{\sigma}^2_{\boldsymbol{y}_{\mathcal{I}_k}}}{2\sigma^2} + N \ln(\sqrt{2\pi}\sigma) + \lambda K \right\}. \tag{6}$$

Multiplying the above criterion by $2\sigma^2$ yields

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} \left\{ \sum_{i=1}^{K} |\mathcal{I}_k|\tilde{\sigma}^2_{\boldsymbol{y}_{\mathcal{I}_k}} + \lambda_1 K + cst_2 \right\}, \tag{7}$$

where $\lambda_1 = 2\sigma^2\lambda$ and $cst_2 = 2\sigma^2 N \ln(\sqrt{2\pi}\sigma)$.

The recovery of $\boldsymbol{x}$ from $\boldsymbol{y}$ is typically formulated as a total variation penalized least-square minimization problem [6]:

$$\min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{N}(y_i - x_i)^2 + \lambda_1 \sum_{i=1}^{N-1} |x_{i+1} - x_i|_0 \right\}, \tag{8}$$

where $|x|_0$ is the scalar version of the ℓ_0 quasi-norm, which equals to 0 if and only if x is 0, and 1 otherwise.

Problem (7) and (8) are equivalent up to a constant. The penalty term of (8) forces $\boldsymbol{x}$ to be a piecewise constant signal. Therefore if the segmentation profile $\mathcal{I}$ is known, then $x_i^* = \overline{\boldsymbol{y}_{\mathcal{I}_k}}, i \in \mathcal{I}_k$. As a result, the first term in Problem (7) and (8) is equal. The second term in (8) is equal to $\lambda_1(K-1)$ since there are $K-1$ breakpoints between K segments.

4) WEAK STRING MODEL

To reconstruct a visual signal from noisy data, Blake proposed the weak string model [19]:

$$\min_{x_i, l_i} \left\{ \sum_{i=1}^{N}(y_i - x_i)^2 + \alpha \sum_{i=1}^{N-1}(x_{i+1} - x_i)^2(1 - l_i) + \beta \sum_{i=1}^{N-1} l_i \right\}, \tag{9}$$

where the first term defines the data distortion, the second term reflects the smoothness of reconstruction with non-negative scaling factor α, and the third term penalizes each breakpoint with non-negative β. l_i is introduced to distinguish the continuity:

$$l_i = \begin{cases} 1, & (y_i, y_{i+1}) \text{ is discontinuous;} \\ 0, & \text{otherwise.} \end{cases}$$

Note that (9) can be rewritten as:

$$\begin{aligned} &\min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{N}(y_i - x_i)^2 + \min_{\boldsymbol{l}} \sum_{i=1}^{N-1} \left[\alpha(x_{i+1} - x_i)^2(1 - l_i) + \beta l_i\right] \right\} \\ &= \min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{N}(y_i - x_i)^2 + \right. \\ &\qquad \left. \min_{\boldsymbol{l}} \sum_{i=1}^{N-1} h_{\alpha,\beta}(x_{i+1} - x_i, l_i) \right\} \\ &= \min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{N}(y_i - x_i)^2 + \sum_{i=1}^{N-1} h^*_{\alpha,\beta}(x_{i+1} - x_i) \right\}, \end{aligned}$$

where

$$h_{\alpha,\beta}(x, l) = \alpha x^2(1 - l) + \beta l,$$

and

$$h^*_{\alpha,\beta}(x) = \min_{l\in\{0,1\}} h_{\alpha,\beta}(x, l) = \begin{cases} \alpha x^2, & |x| < \sqrt{\beta/\alpha}; \\ \beta, & \text{otherwise.} \end{cases}$$

Note that when $\beta = 1$ and α tends to infinity, $h^*_{\alpha,\beta}(x)$ tends to $|x|_0$ (see Fig. 1). Therefore, when $\beta = \lambda_1$ and α tends to infinity, Problem (9) approaches Problem (8) and is thus covered by (2).

TABLE 1. The computation of ε for four examples.

Example problem	ε_k^*	$\varepsilon^*(u,v)$
piecewise Poisson process	Eq. (3): $\lvert\mathcal{I}_k\rvert\overline{\boldsymbol{y}_{\mathcal{I}_k}}(1-\ln\overline{\boldsymbol{y}_{\mathcal{I}_k}})+\sum_{i\in\mathcal{I}_k}\ln(y_i!)$	$-(v-u+1)\overline{\boldsymbol{y}_{u:v}}\ln\overline{\boldsymbol{y}_{u:v}}$
Gaussian 1	Eq. (4): $\lvert\mathcal{I}_k\rvert\ln(\sqrt{2\pi e}\tilde{\sigma}_{\boldsymbol{y}_{\mathcal{I}_k}})$	$\frac{v-u+1}{2}\ln\tilde{\sigma}^2_{\boldsymbol{y}_{u:v}}$
Gaussian 2	Eq. (5): $\lvert\mathcal{I}_k\rvert\tilde{\sigma}^2_{\boldsymbol{y}_{\mathcal{I}_k}}/(2\sigma^2)+\lvert\mathcal{I}_k\rvert\ln(\sqrt{2\pi}\sigma)$	$(v-u+1)\tilde{\sigma}^2_{\boldsymbol{y}_{u:v}}$
weak string model	Eq. (12): $\boldsymbol{z}^T\boldsymbol{z}-\boldsymbol{z}^T\boldsymbol{A}_M^{-1}\boldsymbol{z}$	$-\boldsymbol{z}^T\boldsymbol{A}_M^{-1}\boldsymbol{z}$

Gaussian 1 and 2 represent piecewise constant signal contaminated by Gaussian noise with distinct and same variance, respectively.

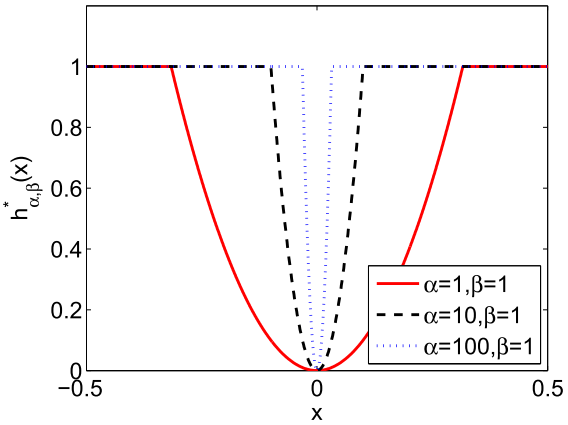


FIGURE 1. The penalty function $h^*_{\alpha,\beta}(x)$ in the weak string model with $\beta = 1$. Note that when α tends to infinity, this function tends to $|x|_0$.

III. OPTIMIZATION ALGORITHM

In this section, we utilize a dynamic programming algorithm for solving (2) when all possible ε_k^* are known (or precomputed). However, before proceeding, we discuss the computation of ε.

A. PRECOMPUTATION OF ε_K^*

As shown in Example 1 of Subsec. II-B, cst_1 does not depend on segmentation $\mathcal{I}$. Therefore, the second term in (3) can be discarded to reduce the number of computations. In addition, in Example 2 of Subsec. II-B, when all σ_k^2 are assumed to equal σ^2, factor $2\sigma^2$ is implied by λ_1. Therefore, (7) is more advantageous than (6) from a computational point of view. In summary, segmentation does not require us to compute ε_k^* exactly.

Therefore, without loss of generality, for the kth segment $\mathcal{I}_k = \{u_k, u_k+1, \ldots, v_k\}$, which ranges from y_{u_k} to y_{v_k}, we introduce $\varepsilon^*(u_k, v_k)$, $1 \leqslant u_k \leqslant v_k \leqslant N$ to denote the portion of ε_k^* that is indispensable for segmentation. Note that $u_1 = 1$, $v_K = N$, and that $u_{k+1} = v_k + 1$, $1 \leqslant k \leqslant K-1$. Therefore, the independent variables of segmentation $\mathcal{I}$ are $v_1, v_2, \ldots, v_{K-1}$.

Tab. 1 lists the computations of $\varepsilon^*(u,v)$ for the examples in Subsec. II-B.

B. GENERAL ALGORITHM

When all possible values of $\varepsilon^*(u,v)$ have been precomputed, the following problem can be solved by dynamic programming:

$$E^* = \min_{v_1,\ldots,v_{K-1}} \left\{ \sum_{k=1}^{K} \varepsilon^*(u_k, v_k) + \lambda K \right\} \tag{10}$$

We introduce $\phi_k(v)$ to denote the minimal cost from the first data point y_1 to y_v with a maximum of k segments.

Based on the reduction principle [20], the general algorithm is expressed as

$$\begin{aligned} \phi_1(v) &= \varepsilon^*(1,v) + \lambda, \quad (2 \leqslant v \leqslant N) \\ \phi_k(v) &= \min \left\{ \min_{1\leqslant u\leqslant v-1} \left[\phi_{k-1}(u) + \varepsilon^*(u+1,v) + \lambda\right], \right. \\ &\qquad \left. \varepsilon^*(1,v) + \lambda \right\}, \quad (2 \leqslant k \leqslant K). \end{aligned} \tag{11}$$

Here we note that in the case $\varepsilon^* \geqslant 0$ (*e.g.,* Gaussian 2 model), we can introduce $\phi_k(0) = 0$ and $\phi_k(1) = \lambda$, therefore, the outer minimum in Eq. (11) can be avoided. However, in general cases this is not true.

Once all $\phi_k(v)$ have been evaluated, the break points can be called backwards recursively as follows:

$$\begin{aligned} v_K &= N, \\ v_{k-1} &= p_{k-1}(v_k), \quad (2 \leqslant k \leqslant K), \end{aligned}$$

where $p_{k-1}(v)$ is a backwards pointer that stores the minimizer of the inner optimization problem in Eq. (11), or equivalently (omitting λ)

$$p_{k-1}(v) = \arg\min_{1\leqslant u\leqslant v-1} \left[\phi_{k-1}(u) + \varepsilon^*(u+1,v)\right].$$

C. ACCELERATION WITH PARALLEL COMPUTING

The computation burden consists of two parts: the computation of ε^* and the computation of ϕ.

To cache ε^* and ϕ, an upper-triangular matrix $\boldsymbol{E} \in \mathbb{R}^{N\times N}$ and a matrix $\boldsymbol{\Phi} \in \mathbb{R}^{K\times N}$ are needed, with elements $[\boldsymbol{E}]_{u,v} = \varepsilon^*(u,v)$, $u < v$ and $[\boldsymbol{\Phi}]_{k,v} = \phi_k(v)$.

Note that the computational complexity of ε^* depends on the model (see Tab. 1), but that of ϕ does not (see (11)).

The aforementioned examples are considered again in the following.

1) COMPUTATION OF ε^*

For the piecewise Poisson process, the main computation is the calculation the mean value of segment $\overline{y_{u:v}}$; while for the Gaussian 1 and 2 models (distinct variances and the same variance), the main computation is the calculation of $\tilde{\sigma}^2_{y_{u:v}}$, which also requires the computation of $\overline{y_{u:v}}$. As a results, an upper-triangular matrix $\overline{\boldsymbol{Y}} \in \mathbb{R}^{N\times N}$ is used to cache those mean values, with element $[\overline{\boldsymbol{Y}}]_{u,v} = \overline{y_{u:v}}$.

To compute each diagonal element of $\overline{\boldsymbol{Y}}$, $\mathcal{O}(N)$ operations (summations and divisions) are needed. Therefore, the computational complexity is $\mathcal{O}(N^2)$ for $\overline{\boldsymbol{Y}}$. Moreover, to compute each element in $\boldsymbol{E}$, 3 additional operations (1 natural logarithm and 2 multiplications) are needed once $\overline{\boldsymbol{Y}}$ is known. Therefore, the overall computational complexity is $\mathcal{O}(\frac{5}{2}N^2)$ for $\boldsymbol{E}$ of the piecewise Poisson process.

Note that the elements in $\boldsymbol{E}$ are computationally independent of one another. Therefore, the elements of the diagonal (or row or column) can be computed in parallel, thereby accelerating the speed by a factor of N. Furthermore, all elements in $\boldsymbol{E}$ can be computed in parallel when sufficiently many processers are provided. Thus, the minimal computational time is bounded by $\mathcal{O}(\log_2 N)$, which corresponds to the longest summation (from y_1 to y_N) when parallel prefix summation [21] is used.

For the Gaussian 1 and 2 models, to compute $\tilde{\sigma}^2_{y_{u:v}}$ of each diagonal of $\boldsymbol{E}$, $3N$ additional operations (N subtractions, N squares, N summations or divisions) are needed once $\overline{\boldsymbol{Y}}$ is known. To compute $\boldsymbol{E}$, 1 or 2 additional operations (1 natural logarithm and 1 multiplication) are needed for each element. Therefore, the overall computational complexity is $\mathcal{O}(5N^2)$. Parallel computing can also be employed for acceleration.

For the weak string model, Appendix A provides an acceleration strategy for the computation of ε.

2) COMPUTATION OF ϕ

For a given value of k, the computational complexity of $\phi_k(v)$ (see Eq. (11)) is $\mathcal{O}(2(v-1))$ (including $v-1$ summations and $v-1$ logical comparisons, omitting λ), if ε is known; when v increases from 1 to N, the computational complexity is approximately $\mathcal{O}(N^2)$. As a result, the computational complexity of $\boldsymbol{\Phi}$ is $\mathcal{O}(KN^2)$.

Note that the computation of $\phi_k(v)$ depends on $\phi_{k-1}(i)$, but not on $\phi_k(i)$ for all $i < v$, *i.e.,* each element of the kth row of $\boldsymbol{\Phi}$ can be computed in parallel when the $k-1$th row of $\boldsymbol{\Phi}$ is known. Therefore, computation can be accelerated by a factor of N. When the concept of parallel prefix summation [21] is extended to the logical comparison in Eq. (11), the minimal computational time is bounded by $\mathcal{O}(\log_2 N)$ for each row of $\boldsymbol{\Phi}$. Thus, $\mathcal{O}(K\log_2 N)$ for $\boldsymbol{\Phi}$.

3) OVERALL COMPUTATION AND IMPLEMENTATION ISSUES

In summary, taking the piecewise Poisson process for example, the computational complexities of ε are ϕ is $\mathcal{O}(\frac{5}{2}N^2)$ and $\mathcal{O}(KN^2)$, respectively. Therefore, when $K \gg 2.5$, then the overall computational complexity is $\mathcal{O}(KN^2)$. When the elements of rows of $\boldsymbol{E}$ and $\boldsymbol{\Phi}$ are computed in parallel, the computation time can be reduced by a factor of N, *i.e.,* $\mathcal{O}(KN)$; when parallel prefix summation is used, the computation time can reach $\mathcal{O}(K\log_2 N)$; and when the pruning rules in pruned exact linear time (PELT) [22] are equipped, computation time can be further reduced.

D. UPPER BOUND OF K

Since the proposed algorithm searches values of segmentation number k from 1 to K (the maximal segmentation number), the computational burden is linearly dependents on K (see Subsec. III-C). Therefore, the following provides an upper bound for K.

Since K is between 1 and N, the minimum of Eq. (10) *i.e.,* E^*, is no larger than both $\varepsilon^*(1,N)+\lambda$ and $\sum_{u=1}^{N}\varepsilon^*(u,u)+\lambda N$, corresponding to $K=1$ and N, respectively.

In contrast, E^* is greater than $K(\varepsilon_0+\lambda)$, where ε_0 is the minimum of $\varepsilon^*(u,v)$. Together, we have

$$K(\varepsilon_0+\lambda) \leqslant E^* \leqslant \min\{\varepsilon^*(1,N)+\lambda, \sum_{u=1}^{N}\varepsilon^*(u,u)+\lambda N\}.$$

Therefore, if $\varepsilon_0+\lambda > 0$, K is upper bounded by

$$K \leqslant \frac{\min\{\varepsilon^*(1,N)+\lambda, \sum_{u=1}^{N}\varepsilon^*(u,u)+\lambda N\}}{\varepsilon_0+\lambda}.$$

As an example, for the Gaussian 2 model, since $\varepsilon_0 = 0$ and $\varepsilon^*(u,u) = 0$, K is upper bounded by $\min\{\frac{\varepsilon^*(1,N)}{\lambda}+1, N\}$, which indicates that a large values of λ reduces the computational burden.

E. SELECTION OF PENALTY PARAMETERS

Since the penalty parameter λ controls the trade-off between the data fitting fidelity and the number of segments [23], it is necessary to tune this parameter with caution.

From the model selection point of view, λ can be determined using information criteria, such as the Akaike information criterion (AIC) [14], the Bayesian information criterion (BIC) [15], which is also known as the Schwarz information criterion (SIC), the Hannan and Quinn criterion (HQC) [24], the Draper information criterion (DIC) [25], and other variants [26]. The minimum description length (MDL) [27], which is popular for model selection, can be approximated by the BIC or DIC [26].

Tab. 2 lists the settings of λ on the basis of these information criteria.

F. RESTRICTION ON THE MINIMAL SEGMENT LENGTH

As an option, one can easily impose a minimal segment length restriction on the proposed model. For example, one can achieve the goal of having the length of all segments to be larger than L by setting $\varepsilon^*(u,v) = +\infty, \forall 0 < v-u < L$ or, equivalently, setting the first L diagonals above the main diagonal in $\boldsymbol{E}$ to positive infinity.

TABLE 2. The setting of λ with popular information criteria.

Information criterion	λ
Akaike information criterion (AIC)	1
Bayesian/Schwarz information criterion (BIC/SIC)	$\frac{\ln(N)}{2}$
Draper information criterion (DIC)	$\frac{\ln(N/2\pi)}{2}$
Hannan and Quinn criterion (HQC)	$\ln(\ln(N))$

IV. EXPERIMENTS AND RESULTS

To test the performance of the proposed framework in terms of segmentation accuracy and computation speed, we consider both simulated and real-world datasets. In the *in silico* studies, two types of data are simulated: one follows the Poisson distribution and the other follows the Gaussian distribution. These datasets are fed to the proposed method and circular binary segmentation (CBS) [5], which is one of the most famous segmentation algorithms. In the analysis of the real-world dataset, the detection of CNV in NGS data from a breast cancer cell line was conducted as an example. CNV detection in NGS data is an important problem in genomics.

A. SEGMENTATION PERFORMANCE

1) PIECEWISE POISSON DISTRIBUTED SIGNALS

A signal $\boldsymbol{y}$ of length $N = 1000$ is first simulated with Poisson parameter $\tau = 40$. Then, 9 segments with various lengths and Poisson parameters are implanted into $\boldsymbol{y}$. The parameters are listed in Tab. 3.

This signal $\boldsymbol{y}$ is segmented under the proposed framework, which includes the piecewise distributed Poisson model, the Gaussian model with distinct variance (Gaussian1), and the Gaussian model with the same variance (Gaussian2). In addition, CBS is also included for comparison. The penalty parameter λ (or λ_1) of the proposed frame work is set over a wide range, from 2^{-10} to 2^{10} with a common ratio of 2. CBS has a similar hyperparameter α_{CBS}(*alpha* in the MATLAB function *cghcbs*) which tunes the significance level for the statistical tests of breakpoint detection. We also implement CBS with several α_{CBS} values from 2^{-18} to 2^{-1} with a common ratio 2, and from 0.6 to 0.9 with a common difference of 0.1.

To evaluate the average performance, 100 Monte Carlo replicates are conducted, *i.e.,* 100 signals $\boldsymbol{y}$ are simulated. For each signal $\boldsymbol{y}$ and hyperparameter (λ for the piecewise Poisson process, 'Gaussian1' and 'Gaussian2', or α_{CBS} for CBS), we implement these segmentation algorithms and calculate the statistics.

Fig. 2(a) shows the normalized ℓ_2 reconstruction error, *i.e.,* $\frac{\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_2}{\|\boldsymbol{y}\|_2}$, with respect to the hyperparameters, where $\hat{\boldsymbol{y}}$ is the reconstructed signal of $\boldsymbol{y}$ after segmentation. The proposed framework outperforms CBS when proper penalty parameter is provided. Markers 'diamond', 'square', and 'triangle' correspond to the values of the penalty parameter λ that are determined by information criteria AIC, HQC and BIC/SIC, respectively; marker 'cross' corresponds to the recommended value of α_{CBS} (0.01). The figure suggests that BIC/SIC provides the best solution among the three alternative information criteria, since the triangle markers are always the closest to the valley bottom of the error curves.

Fig. 2(b) shows the best segmentations obtained by each method (which correspond to the lowest points in Fig. 2(a)) as examples. CBS fails to detect the two pulses (the trench near locus 300 and the peak near locus 400), which are challenging to detect since both the amplitude and size are small (20 and 10, respectively).

Fig. 2(c) shows a more rigid test, in which the hit rate is recorded with respect to the hyperparameters. A hit is defined as a case in which a method finds the ground truth segmentation and the hit rate is the percentage of hits that a method achieves among 100 Monte Carlo replicates. The Poisson model and the Gaussian model with the same variance (Gaussian2) achieve higher hit rates than CBS. However, for Gaussian2, a wider range of values of the hyperparameter yield high hit rates than for the Poisson model.

It is also shown in Fig. 2(b) that the Gaussian model with distinct variances (Gaussian1) prefers segments of small size (the red curve), thereby yielding many false breakpoints, and zero hit rate in Fig. 2(c). To avoid this situation, the minimal segmentation length restriction was applied in subsequent experiments. Since the minimal segment is of length 10 in the ground truth, $L = 8$ is applied.

Fig. 2(d),(e) and (f) shows the normalized ℓ_2 reconstruction error, best segmentation, and hit rate with minimal segmentation length restriction, respectively. The performance of the Gaussian model with distinct variance (Gaussian1) improves substantially. Note that since all methods find the same segmentations, the reconstruction curves are overlapping

TABLE 3. The settings for the simulated piecewise Poisson process.

starting locus	100	200	300	400	500	600	700	800	900
length	20	40	10	10	10	10	10	20	40
Poisson parameter	0	20	20	60	80	100	120	100	80

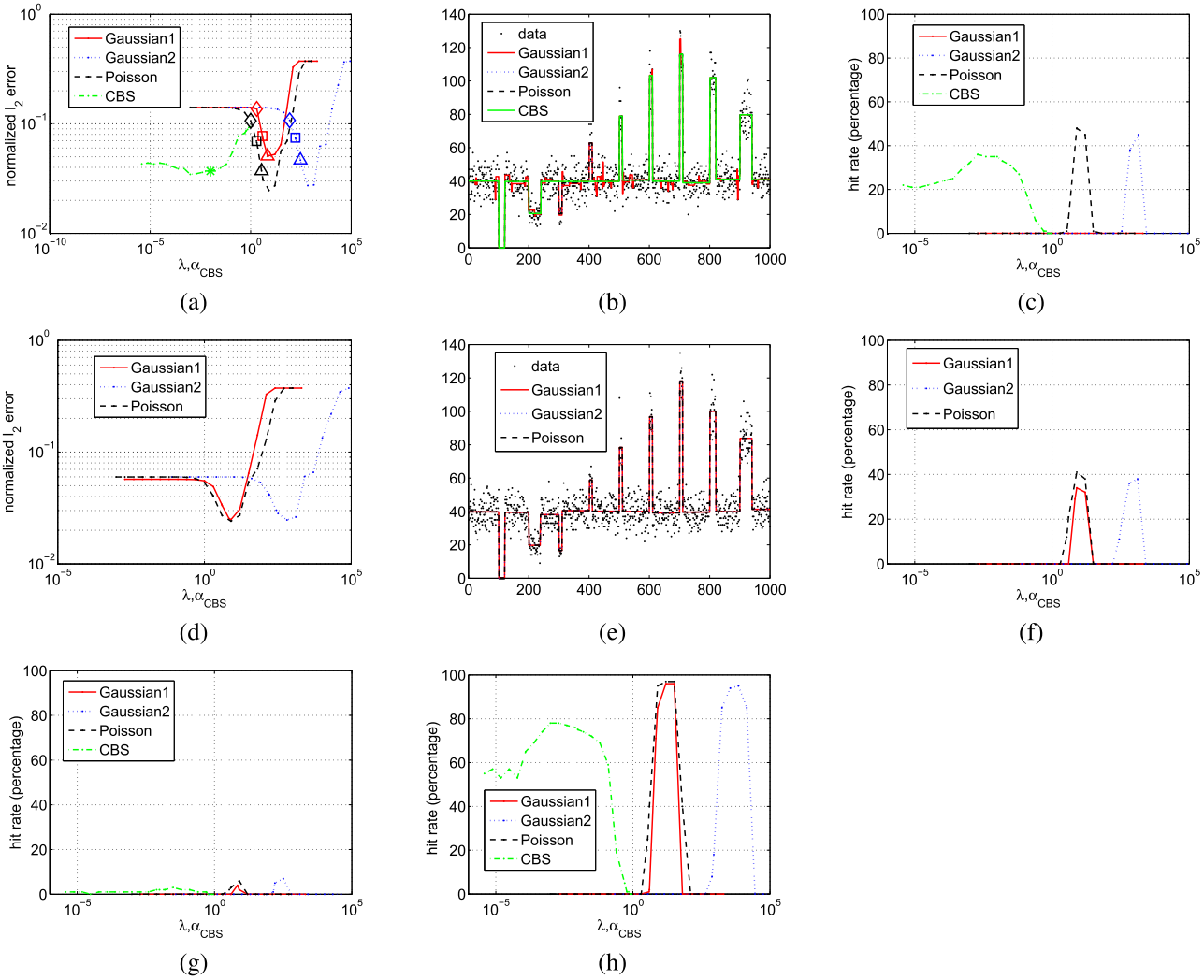


FIGURE 2. The segmentation of piecewise Poisson process. (a) shows the normalized ℓ_2 reconstruction error with respect to hyperparameters (λ for the piecewise Poisson process, 'Gaussian1', 'Gaussian2', and α_{CBS} for CBS). The markers 'diamond', 'square', 'triangle' correspond to λ determined by information criteria AIC, HQC and BIC/SIC, respectively; the marker 'cross' corresponds to the recommended value of α_{CBS} (0.01). 'Gaussian1' and 'Gaussian2' represent the Gaussian model with distinct and same variance, respectively. (b) shows a typical best segmentations of each method (corresponding to the lowest points in (a)). (c) shows the hit rate (in percentage) with respect to hyperparameters. (d) (e) and (f) show the normalized ℓ_2 reconstruction error, best segmentation, and hit rate respectively, when the minimal segmentation length restriction ($L = 8$) is applied. (g) and (h) show the hit rate when the Poisson parameter of ground-true signal is halved and tripled respectively.

in Fig. 2(e), and only the top curve (the black curve, which corresponds to Poisson model) is visible.

To simulate more and less challenging scenarios, the Poisson parameters of the ground-true signal $\boldsymbol{y}$ are halved and tripled and the hit rates are shown in Fig. 2(g) and (h), respectively. The hit rate decreases to 5%, or increases to 95%, respectively. Nevertheless, in both scenarios, the proposed framework outperforms CBS.

2) PIECEWISE GAUSSIAN DISTRIBUTED SIGNALS

This simulation further evaluates the performance of the proposed framework when the data follow a Gaussian distribution. Since negative data points may occur, which yield an undefined likelihood for the Poisson distribution (see Tab. 1), the Poisson model is excluded from this simulation.

The ground-truth signal $\boldsymbol{y}$ is a piecewise-constant signals, *i.e.,* the expectation signal of the piecewise Poisson process in the previous simulation (the black signal in Fig. 2(e)). To mimic various noise level scenarios, i.i.d. Gaussian noise with mean zero is added to $\boldsymbol{y}$, and the signal-to-noise ratio (SNR) is calculated to indicate the noise level.

Fig. 3(a) shows the best hit rates that are obtained by employing the Gaussian model with distinct variances (Gaussian1) and the same variances (Gaussian2) by employing CBS with respect to SNR. This panel shows that with the increase of SNR, the hit rates of all methods increase monotonically from 0 to 100%, and both Gaussian1 and Gaussian2 achieve higher hit rates than CBS.

When SNR drops below 10 dB, neither method offers a single hit. As a result, the previously used normalized

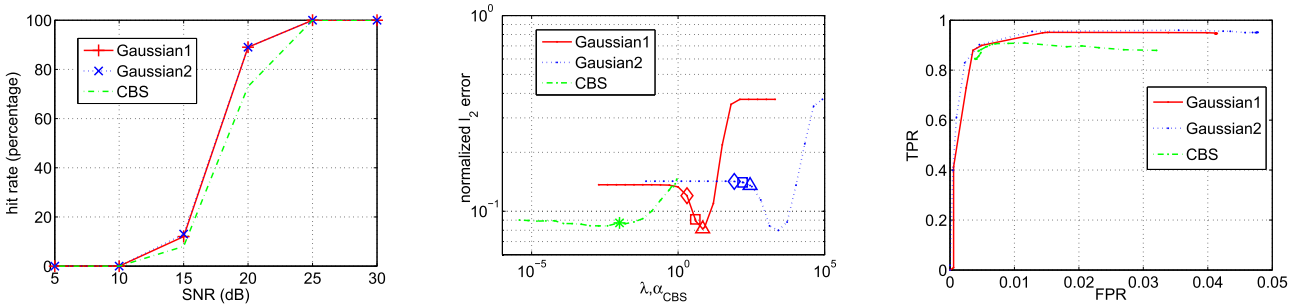


FIGURE 3. The segmentation of piecewise Gaussian distributed signal. (a) shows the hit rates with respect to the SNR. (b) shows the normalized ℓ_2 reconstruction error with respect to hyperparameter at a low SNR scenario (SNR = 10), and (c) shows the corresponding ROC at this scenario.

ℓ_2 reconstruction error and the receiver operating characteristic (ROC) curve are employed instead of the hit rate. The ROC displays the true-positive rate (TPR or sensitivity/statistical power/recall) versus the false-positive rate (FPR or 1-specificity). In the ground-truth signal $\boldsymbol{y}$, segments with amplitude 40 are assumed to be negatives, while other segments (peaks and trenches) are assumed to be positives.

Fig. 3(b) presents the normalized ℓ_2 reconstruction error. Relative to CBS, the proposed framework yields a smaller error when a proper penalty parameter is provided.

Fig. 3(c) shows the ROC of each method with varying hyperparameters. The proposed framework achieves higher sensitivity and specificity than CBS, which suggests better detection performance.

3) CNV DETECTION

The CNV is a type of structural variation (SV) that occurs frequently in mammalian genomes including human, which is associated with genetic diseases and cancers [28], [29]. The NGS technique enables us to study CNV in a much faster and more informative way and has been widely adopted in the study of genomics.

The detection of CNV in NGS data has been proven to be both effective and efficient [17], [30], yet the underlying signal processing problem has not been fully explored. The majority of detection methods make use of the piecewise features of the read depth signal, which is derived from NGS data by counting the number of sequencing reads that are aligned with fixed-size non-overlapping bins or a sliding window along the chromosomes. Therefore, a data point in the read depth signal follows the Poisson distribution. Furthermore, the Poisson parameter is proportional to the copy number locally. Therefore the read depth signal can be modeled as a piecewise Poisson process. As a result, CNVs can be detected by segmenting the read depth signal [3], [4].

In this study, we used the proposed piecewise Poisson model to process a real read depth signal that was extracted from the whole-genome sequencing data of cell line HCC1143, which was generated from a 52 year old Caucasian woman with breast cancer [31]. This cell line has been studied extensively as a breast cancer model and sequenced with the Illumina platform.

A condensed data after mapping, *i.e.,* a BAM file [32] was downloaded. This file includes several types of alignment information, such as the chromosome index, short-read mapping coordinates in the reference genome (homo sapiens NCBI37/hg19), and orientation flag of each mate pair. There are approximately 10 million short reads with a read length of 36 base pairs (bps) or, equivalently, a coverage of 0.17 of the whole human genome.

Then, a bin size of 10 kbp was used to calculate the read depth signal, which is the number (only considering the first base of the 3' end for each read) of aligned short reads in each bin. The resultant read depth is approximately 47 on average.

Subsequently, the read depth signal was processed with both the proposed piecewise Poisson model and CBS, and Fig. 4 shows the results of chromosome 17 at genomic coordinates 56.5 to 61.5 Mbp as a showcase. The two algorithms produce similar segmentation results, except the Poisson model detected a CNV loss at the genomic coordinate from

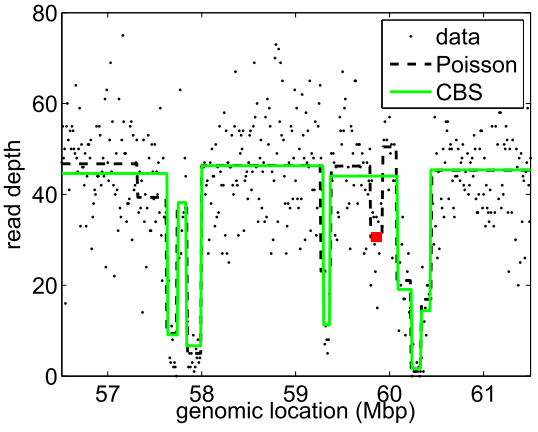


FIGURE 4. The segmentation result (genomic coordinate from 56.5 to 61.5 Mbp) of the read depth signal (black dots) of chromosome 17 of HCC1143, which was generated by next generation sequencing from a Caucasian woman with breast cancer. The red thick line (genomic coordinate from 59.80 to 59.92 Mbp) covers the breast cancer gene ***BRIP1*** (OMIM entry number: 605882).

59.80 to 59.92 Mbp, which is highlighted as the red thick line. Further investigation shows that this region covers a susceptible gene, namely, *breast cancer interacting protein 1 (BRIP1)* [33], which has OMIM entry number 605882 [34].

B. COMPUTATIONAL PERFORMANCE

We tested the computational performance of the proposed framework in terms of execution time. The MATLAB stopwatch timer function *tic* and *toc* were used to record the elapsed time on a desktop with an Intel Core i7 processor and 32 GB memory.

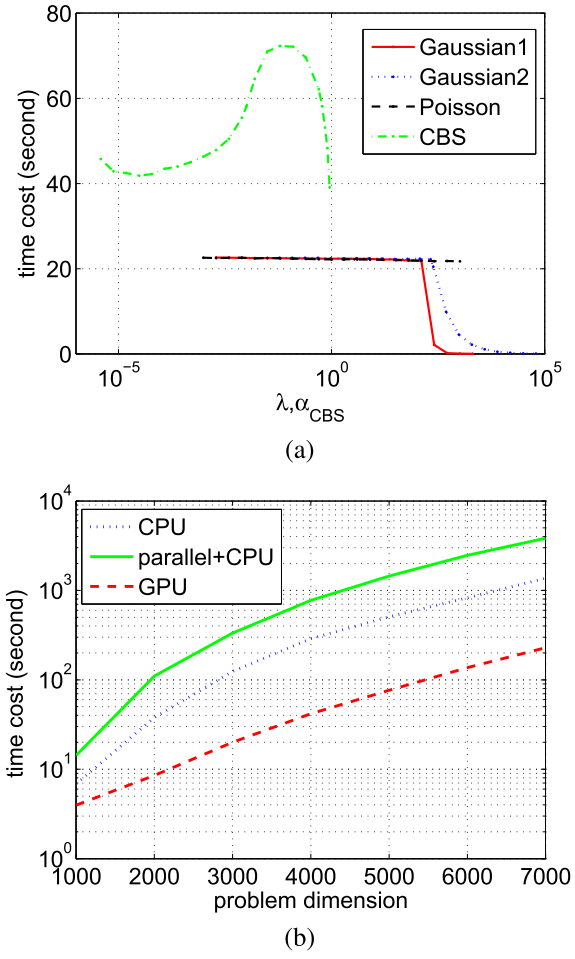


FIGURE 5. The computational performance. (a) shows the computation time of tested methods with respect to the hyperparameters, while (b) shows the computational time with respect to the problem dimension *N*.

Fig. 5(a) shows the average time cost of 100 Monte Carlo replicates with respect to the hyperparameters. The test signal follows a piecewise Poisson process and the problem dimension N is 1000. The time cost of CBS varies from 40 to 70 seconds. For the proposed method, the computation time is at most 22.5 seconds, and with increasing penalty parameter λ, the computation time decreases steeply to zero. Overall, the proposed method spends half as much time as the CBS.

We implemented the GPU-based parallel computing version code (see Sec. III-C) of the proposed framework with the MATLAB GPU computing toolbox and tested the performance on a workstation with an nVIDIA Tesla C2050 GPU. Fig. 5(b) shows the computation time with respect to the problem dimension N. In this panel, 'CPU' and 'GPU' represent the serial computing version code and a parallel alternative on a CPU and GPU, respectively, while 'CPU+parallel' represents the parallel computing version code on a CPU, *i.e.,* data matrices $\boldsymbol{E}$ and $\boldsymbol{\Phi}$ are stored in normal memory instead of *gpuArray* to disable the GPU-based parallel computing.

The GPU-based parallel computing version code increases the computational speed by approximately one order of magnitude relative to its CPU-based serial computing alternative. The time cost increases with respect to the problem dimension N quadratically (from 4 seconds at $N = 1000$ to 200 seconds at $N = 7000$), which is consistent with the computational complexity analysis in Sec. III-C. Note that 'CPU+parallel' costs more time than 'CPU' since there are additional expenses involved in parallelizing the proposed framework from its serial alternative.

V. CONCLUSIONS AND DISCUSSION

We have proposed a parallel-computing-based framework for segmenting piecewise signals. The main contributions are three-fold:

- The proposed framework is applicable not only to signals that follow the distributions that are mentioned in the current paper but also to other distributions, such as the negative binomial distribution, which is used in the NGS data analysis [35], the gamma distribution; and the Nakagami distribution, which is used in ultrasound tissue characterization [36], [37].
- The proposed framework is capable of incorporating parallel computing to accelerate the computing speed, which is of great importance for many applications that demand high-throughput analysis methods, such as large-volume NGS data. i
- In the proposed framework, a restriction on the minimal segmentation length (*i.e.,* the minimal distance between adjacent breakpoints) can be applied easily, which is useful for suppressing false-positive detection.

We tested the proposed framework on both simulated and real genomic data; the results shows that the proposed framework outperforms existing segmentation algorithms, such as the representative CBS, in terms of both segmentation accuracy and computational cost.

The selection of a proper model distribution remains an open problem. When the distribution is known *a priori* for a specific application, a precise model can be built, *e.g.,* the piecewise Poisson process addressed here. However, in most applications this knowledge is missing. In such cases, based on our experiences, the use of the Gaussian model with the same variance is a safe choice. As shown in Sec. IV-A.1, at small τ (20), the Poisson model outperforms the Gaussian models. However, when the Poisson parameter τ is large (*e.g.,* greater than 50), the Poisson distribution is close to the

Gaussian distribution with mean τ and variance τ, and there is little difference between the two criteria.

The selection of penalty parameters is another open problem. As noted in Sec. III-E, several information criteria have been proposed for tackling this problem. Our experiments show that for the segmentation problem, the BIC/SIC provides more realistic parameter settings than the other two criteria, namely, AIC and HQC. For further investigation on this topic, Markon and Krueger [26] provided a comprehensive comparison.

One main limitation of dynamic programming is the heavy computational burden, which increases sharply with the dimension. As mentioned previously, one of the three main contributions of this work is the use of parallel computing to accelerate the execution. At present, the parallel computing was implemented in the GPU. Because the memory and cores in our GPU are limited (3 GB and 448, respectively, in the nVIDIA Tesla C2050), the maximal problem dimension is 7000 and the computing is not fully parallelized. For higher dimensionality, implementation on other platforms is necessary, *e.g.,* on a high-performance computer, to further explore the potential of the proposed framework.

APPENDIX
COMPUTATION OF ε^* FOR THE WEAK STRING MODEL

This appendix describes the computation of ε^* for the weak string model (9), which is much more complex than the examples discussed previously. Subsec. A-A presents a direct method for computing this quantity. However, since this direct method involves a matrix inversion (see Eq. (12)), the computational burden is high. Therefore, in Subsec. A-B we show an indirect method that avoids matrix inversion by updating ε^* (see Eq. (26)). Based on this indirect method, a recursive updating algorithm is shown in Subsec. A-C.

A. DIRECT COMPUTATION

For the weak string model (9), without loss of generality, suppose that a segment starts at 1 ($u = 1$) and ends at v, ($v > 1$), *i.e.,* $l_i = \begin{cases} 1, & i = v; \\ 0, & i = 1, \ldots, v-1. \end{cases}$. Denote $\boldsymbol{z} = [y_1, \ldots, y_v]^T \in \mathbb{R}^v$, $\boldsymbol{w} = [x_1, \ldots, x_v]^T \in \mathbb{R}^v$, and $\boldsymbol{e} = [0, \ldots, 0, 1]^T \in \mathbb{R}^v$. The sum of the first two terms in Eq. (9) is

$$\varepsilon(\boldsymbol{w}) = \|\boldsymbol{z} - \boldsymbol{w}\|^2 + \alpha\|\boldsymbol{D}_v\boldsymbol{w}\|^2,$$

where

$$\boldsymbol{D}_v = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(v-1)\times v}.$$

$\varepsilon(\boldsymbol{w})$ is quadratic with respect to $\boldsymbol{w}$. Therefore,

$$\begin{aligned} &\frac{\partial\varepsilon(\boldsymbol{w})}{\partial\boldsymbol{w}} = 2(\boldsymbol{w} - \boldsymbol{z}) + 2\alpha\boldsymbol{D}_v^T\boldsymbol{D}_v\boldsymbol{w} = 0 \\ &\Rightarrow \boldsymbol{w}^* = (\boldsymbol{I}_v + \alpha\boldsymbol{D}_v^T\boldsymbol{D}_v)^{-1}\boldsymbol{z} = \boldsymbol{A}_v^{-1}\boldsymbol{z}, \end{aligned}$$

where $\boldsymbol{I}_v \in \mathbb{R}^{v\times v}$ is an identity matrix and

$$\boldsymbol{A}_v = \boldsymbol{I}_v + \alpha\boldsymbol{D}_v^T\boldsymbol{D}_v \in \mathbb{R}^{v\times v}$$

is symmetric, positive definite, and nonsingular with $\alpha > 0$. Therefore, it is invertible:

$$\begin{aligned} \varepsilon^*(1, v) &= \varepsilon(\boldsymbol{w}^*) \\ &= (\boldsymbol{z} - \boldsymbol{A}_v^{-1}\boldsymbol{z})^T(\boldsymbol{z} - \boldsymbol{A}_v^{-1}\boldsymbol{z}) \\ &\quad + \alpha(\boldsymbol{D}_v\boldsymbol{A}_v^{-1}\boldsymbol{z})^T(\boldsymbol{D}_v\boldsymbol{A}_v^{-1}\boldsymbol{z}) \\ &= \boldsymbol{z}^T[(\boldsymbol{I}_v - \boldsymbol{A}_v^{-1})^2 + \alpha\boldsymbol{A}_v^{-1}\boldsymbol{D}_v^T\boldsymbol{D}_v\boldsymbol{A}_v^{-1}]\boldsymbol{z} \\ &= \boldsymbol{z}^T[(\boldsymbol{I}_v - \boldsymbol{A}_v^{-1})^2 + \boldsymbol{A}_v^{-1}(\boldsymbol{A}_v - \boldsymbol{I}_v)\boldsymbol{A}_v^{-1}]\boldsymbol{z} \\ &= \boldsymbol{z}^T[(\boldsymbol{I}_v - \boldsymbol{A}_v^{-1})^2 + (\boldsymbol{I}_v - \boldsymbol{A}_v^{-1})\boldsymbol{A}_v^{-1}]\boldsymbol{z} \\ &= \boldsymbol{z}^T(\boldsymbol{I}_v - \boldsymbol{A}_v^{-1})\boldsymbol{z} \\ &= \boldsymbol{z}^T\boldsymbol{z} - \boldsymbol{z}^T\boldsymbol{A}_v^{-1}\boldsymbol{z}. \end{aligned} \tag{12}$$

B. INDIRECT COMPUTATION

Furthermore, denoting $\boldsymbol{z}_+ = [y_1, \ldots, y_v, y_{v+1}]^T \in \mathbb{R}^{v+1}$ and $\boldsymbol{w}_+ = [x_1, \ldots, x_v, x_{v+1}]^T \in \mathbb{R}^{v+1}$, from Eq. (12) we obtain

$$\varepsilon^*(1, v+1) = \boldsymbol{z}_+^T\boldsymbol{z}_+ - \boldsymbol{z}_+^T\boldsymbol{A}_{v+1}^{-1}\boldsymbol{z}_+, \tag{13}$$

where

$$\begin{aligned} \boldsymbol{A}_{v+1} &= \boldsymbol{I}_{v+1} + \alpha\boldsymbol{D}_{v+1}^T\boldsymbol{D}_{v+1} \\ &= \boldsymbol{I}_{v+1} + \alpha\left[\begin{array}{c|c} \boldsymbol{D}_v^T\boldsymbol{D}_v + \boldsymbol{e}\boldsymbol{e}^T & -\boldsymbol{e} \\ \hline -\boldsymbol{e}^T & 1 \end{array}\right] \\ &= \left[\begin{array}{c|c} \boldsymbol{I}_v + \alpha\boldsymbol{D}_v^T\boldsymbol{D}_v + \alpha\boldsymbol{e}\boldsymbol{e}^T & -\alpha\boldsymbol{e} \\ \hline -\alpha\boldsymbol{e}^T & 1+\alpha \end{array}\right] \\ &= \left[\begin{array}{c|c} \boldsymbol{A}_v + \alpha\boldsymbol{e}\boldsymbol{e}^T & -\alpha\boldsymbol{e} \\ \hline -\alpha\boldsymbol{e}^T & 1+\alpha \end{array}\right] \\ &= \left[\begin{array}{c|c} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \hline \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{array}\right], \end{aligned}$$

and

$$\boldsymbol{B}_{11} = \boldsymbol{A}_v + \alpha\boldsymbol{e}\boldsymbol{e}^T \tag{14}$$
$$\boldsymbol{B}_{12} = -\alpha\boldsymbol{e} \tag{15}$$
$$\boldsymbol{B}_{21} = -\alpha\boldsymbol{e}^T \tag{16}$$
$$\boldsymbol{B}_{22} = 1 + \alpha.$$

From the block matrix inversion lemma [38], we have

$$\boldsymbol{A}_{v+1}^{-1} = \left[\begin{array}{c|c} \boldsymbol{S}_1^{-1} & -\boldsymbol{B}_{11}^{-1}\boldsymbol{B}_{12}\boldsymbol{S}_2^{-1} \\ \hline -\boldsymbol{S}_2^{-1}\boldsymbol{B}_{21}\boldsymbol{B}_{11}^{-1} & \boldsymbol{S}_2^{-1} \end{array}\right], \tag{17}$$

where

$$\begin{aligned} \boldsymbol{S}_1 &= \boldsymbol{B}_{11} - \boldsymbol{B}_{12}\boldsymbol{B}_{22}^{-1}\boldsymbol{B}_{21} \\ &= \boldsymbol{A}_v + \alpha\boldsymbol{e}\boldsymbol{e}^T - (-\alpha\boldsymbol{e})\frac{1}{1+\alpha}(-\alpha\boldsymbol{e})^T \\ &= \boldsymbol{A}_v + \alpha\boldsymbol{e}\boldsymbol{e}^T - \frac{\alpha^2}{1+\alpha}\boldsymbol{e}\boldsymbol{e}^T \\ &= \boldsymbol{A}_v + \frac{\alpha}{1+\alpha}\boldsymbol{e}\boldsymbol{e}^T \end{aligned} \tag{18}$$

$$\begin{aligned} \boldsymbol{S}_2 &= \boldsymbol{B}_{22} - \boldsymbol{B}_{21}\boldsymbol{B}_{11}^{-1}\boldsymbol{B}_{12} \\ &= 1 + \alpha - (-\alpha\boldsymbol{e}^T)\boldsymbol{B}_{11}^{-1}(-\alpha\boldsymbol{e}) \\ &= 1 + \alpha - \alpha^2\boldsymbol{e}^T\boldsymbol{B}_{11}^{-1}\boldsymbol{e} \end{aligned} \tag{19}$$

Since $\boldsymbol{A}_v$ is non-singular, we can designate the last main diagonal element and the last column of $\boldsymbol{A}_v^{-1}$ as $[\boldsymbol{A}_v^{-1}]_{v,v} = \rho$ and $[\boldsymbol{A}_v^{-1}]_{\cdot,v} = \boldsymbol{a}$, respectively. Therefore $\boldsymbol{A}_v^{-1}\boldsymbol{e} = \boldsymbol{a}$ and $\boldsymbol{e}^T\boldsymbol{A}_v^{-1}\boldsymbol{e} = \boldsymbol{e}^T\boldsymbol{a} = \rho = \boldsymbol{a}^T\boldsymbol{e}$. Since $\boldsymbol{A}_v$ is positive definite, we have $\rho > 0$. Therefore, $1 + \alpha\rho > 0$.

From the Woodburg matrix identity, we have

$$\begin{aligned}\boldsymbol{B}_{11}^{-1} &= (\boldsymbol{A}_v + \alpha\boldsymbol{e}\boldsymbol{e}^T)^{-1}\\ &= \boldsymbol{A}_v^{-1} - \boldsymbol{A}_v^{-1}\boldsymbol{e}(\frac{1}{\alpha} + \boldsymbol{e}^T\boldsymbol{A}_v^{-1}\boldsymbol{e})^{-1}\boldsymbol{e}^T\boldsymbol{A}_v^{-1}\\ &= \boldsymbol{A}_v^{-1} - \boldsymbol{a}(\frac{1}{\alpha} + \rho)^{-1}\boldsymbol{a}^T\\ &= \boldsymbol{A}_v^{-1} - \frac{\alpha}{1+\alpha\rho}\boldsymbol{a}\boldsymbol{a}^T \end{aligned} \tag{20}$$

Comparing Eq. (14) with Eq. (18) and replacing α in Eq. (20) with $\frac{\alpha}{1+\alpha}$, we obtain

$$\begin{aligned}\boldsymbol{S}_1^{-1} &= \boldsymbol{A}_v^{-1} - \frac{\frac{\alpha}{1+\alpha}}{1+\frac{\alpha}{1+\alpha}\rho}\boldsymbol{a}\boldsymbol{a}^T\\ &= \boldsymbol{A}_v^{-1} - \frac{\alpha}{1+\alpha+\alpha\rho}\boldsymbol{a}\boldsymbol{a}^T\\ &= \boldsymbol{A}_v^{-1} - \rho_1\boldsymbol{a}\boldsymbol{a}^T \end{aligned} \tag{21}$$

where

$$\rho_1 = \frac{\alpha}{1+\alpha+\alpha\rho}.$$

Substituting Eq. (20) into Eq. (19), we obtain

$$\begin{aligned}\boldsymbol{S}_2 &= 1 + \alpha - \alpha^2\boldsymbol{e}^T\boldsymbol{B}_{11}^{-1}\boldsymbol{e}\\ &= 1 + \alpha - \alpha^2\boldsymbol{e}^T(\boldsymbol{A}_v^{-1} - \frac{\alpha}{1+\alpha\rho}\boldsymbol{a}\boldsymbol{a}^T)\boldsymbol{e}\\ &= 1 + \alpha - \alpha^2(\boldsymbol{e}^T\boldsymbol{A}_v^{-1}\boldsymbol{e} - \frac{\alpha}{1+\alpha\rho}\boldsymbol{e}^T\boldsymbol{a}\boldsymbol{a}^T\boldsymbol{e})\\ &= 1 + \alpha - \alpha^2(\rho - \frac{\alpha}{1+\alpha\rho}\rho^2)\\ &= \frac{1+\alpha+\alpha\rho}{1+\alpha\rho} = \frac{1}{\rho_2}, \end{aligned} \tag{22}$$

where

$$\rho_2 = \frac{1+\alpha\rho}{1+\alpha+\alpha\rho}, \tag{23}$$

and

$$\rho_1 + \rho_2 = 1.$$

Substituting Eqs. (21), (22), (14), and (15) into Eq. (17) yields

$$\boldsymbol{A}_{v+1}^{-1} = \left[\begin{array}{c|c} \boldsymbol{A}_v^{-1} - \rho_1\boldsymbol{a}\boldsymbol{a}^T & \rho_1\boldsymbol{a} \\ \hline \rho_1\boldsymbol{a}^T & \rho_2 \end{array}\right]. \tag{24}$$

Assigning

$$\varphi = \boldsymbol{z}^T\boldsymbol{a}, \tag{25}$$

and substituting Eq. (24) into Eq. (13), we obtain

$$\begin{aligned}&\varepsilon^*(1, v+1)\\ &= \boldsymbol{z}^T\boldsymbol{z} + y_{v+1}^2 - [\boldsymbol{z}^T(\boldsymbol{A}_v^{-1} - \rho_1\boldsymbol{a}\boldsymbol{a}^T)\boldsymbol{z}\\ &\quad + 2\boldsymbol{z}^T\rho_1\boldsymbol{a}y_{v+1} + \rho_2 y_{v+1}^2]\\ &= \boldsymbol{z}^T\boldsymbol{z} + y_{v+1}^2 - \boldsymbol{z}^T\boldsymbol{A}_v^{-1}\boldsymbol{z} + \rho_1\varphi^2\\ &\quad - 2\rho_1 y_{v+1}\varphi - \rho_2 y_{v+1}^2\\ &= \varepsilon^*(1, v) + \rho_1(y_{v+1}^2 + \varphi^2 - 2y_{v+1}\varphi)\\ &= \varepsilon^*(1, v) + (1-\rho_2)(y_{v+1} - \varphi)^2 \end{aligned} \tag{26}$$

Based on the above equation, $\varepsilon^*(1, v)$, $(1 \leqslant v \leqslant N)$ can be computed recursively.

C. RECURSIVE RELATIONSHIPS

Note that $[\boldsymbol{A}_v^{-1}]_{v,v} = \rho$ and that $[\boldsymbol{A}_{v+1}^{-1}]_{v+1,v+1} = \boldsymbol{S}_2^{-1} = \rho_2$. From Eq. (23), the recursive relationship of ρ_2 is expressed as

$$\begin{aligned}\rho_2^{(i+1)} &= \frac{1+\alpha\rho_2^{(i)}}{1+\alpha+\alpha\rho_2^{(i)}}, \quad (i \geqslant 0)\\ \rho_2^{(0)} &= 1. \end{aligned}$$

From Eq. (25) and Eq. (24), we define $\varphi_{u:v} = [y_u, \ldots, y_v]\boldsymbol{a}$ and the recursive relationship of φ is expressed as

$$\begin{aligned}\varphi_{u:v+1} &= \left[\,[y_u, \ldots, y_v] \;\middle|\; y_{v+1}\,\right]\left[\begin{array}{c}\rho_1\boldsymbol{a}\\ \hline \rho_2\end{array}\right]\\ &= \rho_1[y_u, \ldots, y_v]\boldsymbol{a} + \rho_2 y_{v+1}\\ &= \rho_1\varphi_{u:v} + \rho_2 y_{v+1}\\ &= (1 - \rho_2^{(v-u+1)})\varphi_{u:v} + \rho_2^{(v-u+1)}y_{v+1}, \quad (u \leqslant v < N)\\ \varphi_{u:u} &= y_u, \quad (1 \leqslant u \leqslant N). \end{aligned}$$

Finally, from Eq. (26), for any u, the recursive relationship of ε^* is expressed as

$$\begin{aligned}\varepsilon^*(u, v+1) &= (1 - \rho_2^{(v-u+1)})(y_{v+1} - \varphi_{u:v})^2\\ &\quad + \varepsilon^*(u, v), \quad (u \leqslant v < N)\\ \varepsilon^*(u, u) &= 0, \quad (1 \leqslant u \leqslant N). \end{aligned}$$

ACKNOWLEDGMENT

The MATLAB code of the proposed framework can be downloaded at https://www.mathworks.com/matlabcentral/fileexchange/65947-a-parallelizable-framework-for-segmenting-piecewise-signals.

REFERENCES

[1] J. Chen and Y.-P. Wang, "A statistical change point model approach for the detection of dna copy number variations in array CGH data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 6, no. 4, pp. 529–541, Oct./Dec. 2009.

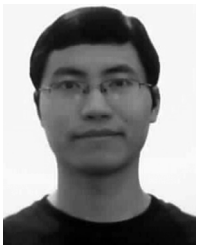
[2] J. Duan, C. Soussen, D. Brie, J. Idier, Y.-P. Wang, and M. Wan, "An optimal method to segment piecewise poisson distributed signals with application to sequencing data," in *Proc. IEEE 37th Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBC)*, Milan, Italy, Aug. 2015, pp. 6465–6468.

[3] J. Duan, J.-G. Zhang, H.-W. Deng, and Y.-P. Wang, "Comparative studies of copy number variation detection methods for next-generation sequencing technologies," *PLoS One*, vol. 8, no. 3, p. e59128, 2013.

[4] J. Duan, J.-G. Zhang, H.-W. Deng, and Y.-P. Wang, "CNV-TV: A robust method to discover copy number variation from short sequencing reads," *BMC Bioinf.*, vol. 14, no. 1, p. 150, 2013.

[5] A. B. Olshen, E. S. Venkatraman, R. Lucito, and M. Wigler, "Circular binary segmentation for the analysis of array-based DNA copy number data," *Biostatistics*, vol. 5, no. 4, pp. 557–572, 2004.

[6] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numer. Math.*, vol. 76, no. 2, pp. 167–188, 1997.
[7] M. Nikolova, "Local strong homogeneity of a regularized estimator," *SIAM J. Appl. Math.*, vol. 61, pp. 633–658, Feb. 2000.
[8] C. Lévy-leduc and Z. Harchaoui, "Catching change-points with lasso," in *Proc. NIPS*, 2007, pp. 617–624.
[9] R. Tibshirani and P. Wang, "Spatial smoothing and hot spot detection for CGH data using the fused lasso," *Biostatistics*, vol. 9, no. 1, pp. 18–29, 2008.
[10] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 trend filtering," *SIAM Rev.*, vol. 51, no. 2, pp. 339–360, 2009.
[11] H. Zou and R. Li, "Discussion: One-step sparse estimates in nonconcave penalized likelihood models," *Ann. Statist.*, vol. 36, no. 4, pp. 1509–1533, 2008.
[12] F. Friedrich, A. Kempe, V. Liebscher, and G. Winkler, "Complexity penalised m-estimation: Fast computation," Inst. Biomath. Biometry, GSF-Nat. Res. Center Environ. Health, Tech. Rep., Nov. 2005.
[13] M. Storath, A. Weinmann, and L. Demaret, "Jump-sparse and sparse recovery using Potts functionals," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3654–3666, Jul. 2014.
[14] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.
[15] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.
[16] A. Magi, M. Benelli, A. Gozzini, F. Girolami, F. Torricelli, and M. L. Brandi, "Bioinformatics for next generation sequencing data," *Genes*, vol. 1, no. 2, pp. 294–307, 2010.
[17] D. Y. Chiang *et al.*, "High-resolution mapping of copy-number alterations with massively parallel sequencing," *Nature Methods*, vol. 6, pp. 99–103, Jan. 2009.
[18] G. Klambauer *et al.*, "cn.MOPS: Mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate," *Nucleic Acids Res.*, vol. 40, no. 9, p. e69, 2012.
[19] A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-11, no. 1, pp. 2–12, Jan. 1989.
[20] F. Friedrich, A. Kempe, V. Liebscher, and G. Winkler, "Complexity penalized M-estimation: Fast computation," *J. Comput. Graph. Statist.*, vol. 17, no. 1, pp. 201–224, 2008.
[21] R. E. Ladner and M. J. Fischer, "Parallel prefix computation," *J. ACM JACM*, vol. 27, no. 4, pp. 831–838, 1980.
[22] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *J. Amer. Statist. Assoc.*, vol. 107, no. 500, pp. 1590–1598, 2012.
[23] Y. C. Eldar and G. Kutyniok, Eds., *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[24] E. J. Hannan and B. G. Quinn, "The determination of the order of an autoregression," *J. Roy. Stat. Soc. B*, vol. 41, no. 2, pp. 190–195, 1979.
[25] D. Draper, "Assessment and propagation of model uncertainty," *J. Roy. Stat. Soc. B*, vol. 57, no. 1, pp. 45–97, 1995.
[26] K. E. Markon and R. F. Krueger, "An empirical comparison of information-theoretic selection criteria for multivariate behavior genetic models," *Behav. Genet.*, vol. 34, no. 6, pp. 593–610, 2004.
[27] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Ann. Statist.*, vol. 11, no. 2, pp. 416–431, 1983.
[28] J. L. Freeman *et al.*, "Copy number variation: New insights in genome diversity," *Genome Res.*, vol. 16, pp. 949–961, Aug. 2006.
[29] P. Stankiewicz and J. R. Lupski, "Structural variation in the human genome and its role in disease," *Annu. Rev. Med.*, vol. 61, pp. 437–455, Feb. 2010.
[30] P. Medvedev, M. Stanciu, and M. Brudno, "Computational methods for discovering structural variation with next-generation sequencing," *Nature Methods*, vol. 6, pp. S13–S20, Oct. 2009.
[31] A. F. Gazdar *et al.*, "Characterization of paired tumor and non-tumor cell lines established from patients with breast cancer," *Int. J. Cancer*, vol. 78, no. 6, pp. 766–774, 1998.
[32] V. Boeva *et al.*, "Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization," *Bioinformatics*, vol. 27, no. 2, pp. 268–269, Jan. 2011.
[33] S. B. Cantor *et al.*, "BACH1, a novel helicase-like protein, interacts directly with BRCA1 and contributes to its DNA repair function," *Cell*, vol. 105, no. 1, pp. 149–160, 2001.
[34] *Online Mendelian Inheritance in Man (OMIM:605882)*. [Online]. Available: http://www.omim.org/entry/605882
[35] C. A. Miller, O. Hampton, C. Coarfa, and A. Milosavljevic, "ReadDepth: A parallel R package for detecting copy number alterations from short sequencing reads," *PLoS ONE*, vol. 6, no. 1, p. 16327, 2011.
[36] P. M. Shankar, "A general statistical model for ultrasonic backscattering from tissues," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 47, no. 3, pp. 727–736, May 2000.
[37] P.-H. Tsui and C.-C. Chang, "Imaging local scatterer concentrations by the Nakagami statistical model," *Ultrasound Med. Biol.*, vol. 33, no. 4, pp. 608–619, 2007.
[38] D. Bernstein, *Matrix Mathematics*. Princeton, NJ, USA: Princeton Univ. Press, 2009.



JUNBO DUAN (M'15) received the B.S. degree in information engineering and the M.S. degree in communication and information system from Xi'an Jiaotong University, Xi'an, China, in 2004 and 2007, respectively, and the Ph.D. degree in signal processing from Université Henry Poincaré, Nancy, France, in 2010. He was a Postdoctoral Fellow with the Department of Biomedical Engineering and Biostatistics and Bioinformatics, Tulane University, USA, until 2013. He is currently an Associate Professor with the Department of Biomedical Engineering, Xi'an Jiaotong University. His major research interests include probabilistic approaches to inverse problems in biomedical engineering and bioinformatics.

CHARLES SOUSSEN (M'12) was born in Paris, France, in 1972. He graduated from the École Nationale Supérieure en Informatique et Mathématiques Appliquées, Grenoble, France, in 1996. He received the Ph.D. degree in physics from the Université de Paris-Sud, Orsay, France, in 2000, and the Habilitation à Diriger des Recherches degree in signal processing from the Université de Lorraine, France, in 2013. From 2001 to 2005, he was an Associate Professor with Paris-Sud University. From 2005 to 2017, he was with the University of Lorraine. He has been a Full Professor with CentraleSupélec and the Laboratoire des Signaux et Systèmes, since 2017. His research interests include inverse problems and sparse approximation.

DAVID BRIE (M'13) received the Ph.D. and Habilitation à Diriger des Recherches degrees from the Université de Lorraine, France, in 1992 and 2000, respectively. Since 1990, he has been with the Centre de Recherche en Automatique de Nancy, Université de Lorraine. He has been a Full Professor with the Université de Lorraine, since 2001. In 2013, he was a Visiting Researcher with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include statistical signal processing, inverse problems, and multidimensional signal processing. Since 2014, he has been serving as the Editor-in-Chief for the *Traitement du Signal*, French journal.

JÉRÔME IDIER (M'09) was born in France, in 1966. He received the Diploma degree in electrical engineering from the École Supéieure d'Électricité, Gif-sur-Yvette, France, in 1988, and the Ph.D. degree in physics from the University of Paris-Sud, Orsay, France, in 1991. In 1991, he joined the Centre National de la Recherche Scientifique. He is currently a Senior Researcher with the Institut de Recherche en Communications et Cybernétique, Nantes, France. His major scientific interests include probabilistic approaches to inverse problems for signal and image processing. He is an elected member of the French National Committee for Scientific Research.

YU-PING WANG (SM'06) received the B.S. degree in applied mathematics from Tianjin University, Tianjin, China, in 1990, and the M.S. degree in computational mathematics and the Ph.D. degree in communications and electronic systems from Xi'an Jiaotong University, Xi'an, China, in 1993 and 1996, respectively. After graduation, he had visiting positions with the Center for Wavelets, Approximation and Information Processing, National University of Singapore, and with the Washington University School of Medicine, St. Louis. From 2000 to 2003, he was a Senior Research Engineer with Perceptive Scientific Instruments, Inc., and with Advanced Digital Imaging Research, LLC, Houston, TX, USA. In 2003, he returned to academia as an Assistant Professor of computer science and electrical engineering with the University of Missouri-Kansas City. He is currently a Professor of biomedical engineering and biostatistics and bioinformatics with the School of Science and Engineering, Tulane University, New Orleans, LA, USA, and with the School of Public Health and Tropical Medicine, New Orleans, LA, USA. He is also a member of the Tulane Center of Bioinformatics and Genomics, Tulane Cancer Center, and the Tulane Neuroscience Program. His research interests include computer vision, signal processing, and machine learning with applications to biomedical imaging and bioinformatics. He has published 150 peer-reviewed articles in these areas. He has served on numerous program committees and NSF/NIH review panels. He served as an Editor for several journals, such as the *Journal of Neuroscience Methods*.

MINGXI WAN (M'01) was born in Hubei, China, in 1962. He received the B.S. degree in geophysical prospecting from the Jianghan Petroleum Institute, Jingzhou, China, in 1982, and the M.S. and Ph.D. degrees in biomedical engineering from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1989, respectively. From 1995 to 1996, he was a Visiting Scholar and an Adjunct Professor with Drexel University, Philadelphia, PA, USA, and with Pennsylvania State University, University Park, PA, USA. From 2000 to 2001, he was a Visiting Scholar with the Department of Biomedical Engineering, University of California at Davis, Davis, CA, USA. From 2000 to 2010, he was the Dean of the School of Life Science and Technology, Xi'an Jiaotong University. He is currently a Full Professor with the Department of Biomedical Engineering, Xi'an Jiaotong University. He has authored or co-authored more than 100 peer-reviewed publications in international journals and five books about medical ultrasound. His current research interests include voice science, ultrasonic imaging, especially in tissue elasticity imaging, contrast and tissue perfusion evaluation, therapeutic ultrasound, and theranostics. He has received several important awards from the Chinese Government and university.