

Research Article

biomvRhsmm: Genomic Segmentation with Hidden Semi-Markov Model

Yang Du,¹ Eduard Murani,¹ Siriluck Ponsuksili,² and Klaus Wimmers¹

¹ *Institute for Genome Biology, Leibniz Institute for Farm Animal Biology, 18196 Dummerstorf, Germany*

² *Research Group Functional Genomics, Leibniz Institute for Farm Animal Biology, 18196 Dummerstorf, Germany*

Correspondence should be addressed to Klaus Wimmers; wimmers@fbn-dummerstorf.de

Received 19 November 2013; Revised 3 March 2014; Accepted 21 March 2014; Published 3 June 2014

Academic Editor: Stanley Brul

Copyright © 2014 Yang Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

High-throughput technologies like tiling array and next-generation sequencing (NGS) generate continuous homogeneous segments or signal peaks in the genome that represent transcripts and transcript variants (transcript mapping and quantification), regions of deletion and amplification (copy number variation), or regions characterized by particular common features like chromatin state or DNA methylation ratio (epigenetic modifications). However, the volume and output of data produced by these technologies present challenges in analysis. Here, a hidden semi-Markov model (HSMM) is implemented and tailored to handle multiple genomic profile, to better facilitate genome annotation by assisting in the detection of transcripts, regulatory regions, and copy number variation by holistic microarray or NGS. With support for various data distributions, instead of limiting itself to one specific application, the proposed hidden semi-Markov model is designed to allow modeling options to accommodate different types of genomic data and to serve as a general segmentation engine. By incorporating genomic positions into the sojourn distribution of HSMM, with optional prior learning using annotation or previous studies, the modeling output is more biologically sensible. The proposed model has been compared with several other state-of-the-art segmentation models through simulation benchmarking, which shows that our efficient implementation achieves comparable or better sensitivity and specificity in genomic segmentation.

1. Introduction

The advent of high-throughput technologies like tiling array and massively parallel sequencing has produced a windfall of large-scale genomic data. Analysis of genome-wide data from these experiments generally requires researchers to search for continuous homogeneous segments or signal peaks. These features can represent regulatory regions [1, 2], transcripts [3–6], or regions of deletion or amplification [7, 8]. The objective of these investigations is, in general, the segmentation or partitioning of the genome into nonoverlapping homogeneous segments and the assignation of a biologically sensible class to each segment.

Various models and computational tools have been developed to handle either the general segmentation problem or particular types of partitioning. Most commonly, the approaches address the detection of chromosomal alterations with array-based comparative genomic hybridization (aCGH) [9–18] or SNP array [19–23], transcript [24, 25] and

protein-binding site detection [26, 27] with tiling array, and the identification of gene expression domains [28, 29]. In recent years, more effort has been devoted to the development of computational tools to deal with read-count data generated from next-generation sequencing (NGS) [30–35].

Many of these computational tools utilize hidden Markov model (HMM) [9, 13, 19, 20, 22, 23, 26, 32, 33] because of its inherent capability of resolving segmentation tasks. However, a standard HMM cannot easily account for one basic property of genomic data—the physical position of the feature. To our knowledge, there have been few limited attempts to incorporate this positional information into HMM [13, 22, 23, 26] or to adopt more complex dynamic Bayesian network models [34]. On the other hand, a hidden semi-Markov model (HSMM), a more generalized form of HMM, could be applied to utilize positional information. Indeed, HSMM was proposed for modeling aCGH data [18], but the tool did not actually utilize positional information and the implementation is no longer publicly available.

Here, we implement in the R/Bioconductor [36, 37] package *biomvRCNS* a novel hidden semi-Markov model, *biomvRhsmm*. This model is specially designed to handle genomic data and tailored to serve as a general segmentation tool for various types of genomic profiles arising from both traditional microarray-based experiments and the recent NGS platform, with native support for modeling spatial patterns carried by genomic position. We also compare the proposed model with several other state-of-the-art segmentation models through simulation benchmarking, which shows that our efficient implementation achieves comparable or better sensitivity and specificity in genomic segmentation.

2. Materials and Methods

2.1. Hidden Semi-Markov Model. A brief summary of the concepts involved and a definition of the hidden semi-Markov model follow. For some experimental data X , we have a vector of observations $x_t = (x_t^1, \dots, x_t^N)$ made for N samples at each time or position t , $t = 1, \dots, T$. At each t , there is an underlying unobserved state $S_t \in S = \{1, \dots, J\}$, which depends only on the previous state at $t-1$, thus forming a length T discrete Markov chain with a finite number J of possible states. The initial state probability is determined by vector π , $\pi_j = P(S_1 = j)$, $j = 1, \dots, J$, with $\sum_{j=1}^J \pi_j = 1$ and $\pi_j \geq 0$. The conditional probability of the observed variable x_t given the unobserved (or hidden) state j is referred to as the emission density B , $b_{tj} = P(X_t = x_t | S_t = j)$. The transition matrix A , giving the probabilities of moving from one state to another, is formulated as $a_{ij} = P(S_{t+1} = j | S_t = i)$, with $\sum_{j=1}^J a_{ij} = 1$ and $a_{ij} \geq 0$. Thus an HMM can be defined by $\theta = (\pi, A, B)$.

A semi-Markov chain can be considered as a two-layer mixture, an embedded first-order Markov chain representing the transitions between distinct states—which follows the standard definition of HMM—and an occupancy distribution attached to each nonabsorbing state of the embedded first-order Markov chain.

The discrete state occupancy distribution or the sojourn distribution, D , is defined as the probability of spending u consecutive time steps in state j , which is geometrically distributed for a normal HMM, $d_j(u) = a_{jj}^{u-1} (1 - a_{jj})$. The hidden semi-Markov model, with the sojourn distribution explicitly specified using a common distribution, can be defined by $\theta = (\pi, A, B, D)$. The explicit modeling of sojourn time immediately enables the full inclusion of genomic distance in the segmentation process. A complete likelihood (1) of the HSMM is given in [38] with survivor function $D_i(u) = \sum_{v \geq u} d_i(v)$ representing the sojourn time spent in the last state.

Consider

$$L(\theta) = \pi_{S_1} d_{S_1}(u_1) \left\{ \prod_{r=2}^R P(S_r | S_{r-1}) d_{S_r}(u_r) \right\} \times P(S_R | S_{R-1}) D_{S_R}(u_R) \prod_{t=1}^T P(X_t | S_t). \quad (1)$$

With likelihood function defined, the optimal model parameters could then be estimated using the expectation-maximization (EM) algorithm. A forward-backward algorithm for the estimation step and a Viterbi algorithm to derive the most likely state sequence are explained in [38], where the author also shows the possibility of replacing the nonparametric M-step of the EM algorithm in sojourn distribution parameters reestimation with a parametric M-step in practice, to simplify the model and prevent overfitting. The E-step of the forward-backward EM procedure and the Viterbi algorithm have been implemented as C library in the package. We also attempted to provide support for parametric reestimations of the M-step based on continuous and discrete distributions like Gamma, Poisson, and negative binomial distribution, which is done *ad hoc* using point estimation methods. More detail about the estimation of HSMM can be found in the Supplementary Material available online at <http://dx.doi.org/10.1155/2014/910390>.

2.2. Implementation. The batch function *biomvRhsmm* accepts both the basic R data matrix and the more encapsulated *GenomicRanges*-like object as input, for better interfacing with other Bioconductor classes and methods [39]. The function will sequentially process each region identified by the distinctive sequence names in the positional input. A second layer of stratification is introduced by a grouping argument, assigning each profile to a group, which could be used to reflect experimental design. Sample columns within the same group could be treated simultaneously in the modeling process as well as iteratively. The assumption is that profiles from the same group could be considered homogeneous and, thus, processed together in a multivariate fashion. Simultaneous treatment of multiple profiles is currently available for emission type set to multivariate normal distribution or multivariate t distribution. Additionally, there is a built-in automatic grouping method by hierarchical clustering.

The prior distribution of the sojourn density will be initialized as flat or be estimated from another related data source by calling the function *sojournAnno*. State number could be either assigned explicitly or inferred during the sojourn learning. The model complexity is limited by a constant M , denoting the upper bound to the time spent in a state, which is quite similar to the approach adapted in the segmentation model in *tilingArray* [24]. The constant could be explicitly given by the argument *maxk* or inferred by another constant *maxbp* together with positional information. The modeling of sojourn time is done using positional information like genomic distance between markers and regresses to a rank-based position setting, like the original design in [38], when positional information is not available. Starting state probabilities will be initialized as a flat vector. Initial parameters for the emission distribution could be estimated using different levels of quantile of the input or via a clustering process, assuming different states tend to have different levels of emitted signals.

The function will then call the C library to compute the smoothed-state probability profile in the E-step, after which model parameters will be reestimated in an M-step.

TABLE 1: List of algorithms compared in this paper.

Name	Reference	Method	R package (version)
<i>bcp</i>	Erdman and Emerson (2008) [16]	Product Partition Model	<i>bcp</i> _3.0.1
<i>bioHMM</i>	Marioni et al. (2006) [13]	Heterogeneous HMM	<i>snapCGH</i> _1.30.0
<i>CBS</i>	Venkatraman and Olshen (2007) [14]	Modified Circular Binary Segmentation	<i>DNAcopy</i> _1.34.0
<i>cghseg</i>	Picard et al. (2011) [17]	Joint CGH Segmentation	<i>cghseg</i> _1.0.1
<i>GLAD</i>	Hupé et al. (2004) [10]	Adaptive Weights Smoothing	<i>GLAD</i> _2.24.0
<i>HaarSeg</i>	Ben-Yaacov and Eldar (2008) [15]	Wavelet Decomposition and Thresholding	<i>HaarSeg</i> _0.0.3
<i>HMM</i>	Fridlyand et al. (2004) [9]	Homogeneous HMM	<i>aCGH</i> _1.38.0

Eventually, the most likely state sequence could be inferred from the smoothed-state probability profile or estimated with the Viterbi algorithm. The complexity of the forward-backward algorithm used in the E-step and the Viterbi algorithm is $O(JT(J+T))$ time in the worst case and $O(JTM)$ space. To relax the high memory burden from NGS data of base-pair resolution, we attempt to use run-length encoding (RLE) for the storage and handling of sequencing count data, since the feature distribution is normally sparse across the genome. Also, to speed up computation, parallel processing of multiple chromosomes or contigs could be enabled, to take advantage of the multicore infrastructure of modern PC.

After the batch run, results are combined and returned together with input data plus model parameters as a *biomvR*-CNS class object, for which a plot method has been implemented to provide integrative visualization of the segmentation results with optional annotation.

2.3. Performance Comparison with Other Segmentation Methods. To show the reliability and relative performance of the proposed model, we compared our implementation with several other state-of-the-art segmentation algorithms (Table 1), using a similar approach as in [38], by calculating the receiver operating characteristic (ROC) curves on simulated data.

Some of the models reviewed in [40] have evolved over the years. Venkatraman and Olshen [14] present a faster, modified version of circular binary segmentation (CBS) [11] in R/Bioconductor package *DNAcopy*. Picard et al. [17] extend the univariate dynamic programming procedure [12] to joint analysis of multiple CGH profiles in R package *cghseg* and adopt the modified Bayesian information criterion [41] for model selection. We also included the unsupervised hidden Markov model described in R package *aCGH* [9] (labeled *HMM* hereafter) and the local adaptive weights smoothing procedure in R package *GLAD* [10] in our comparison; these are considered to be early efforts in the field, thus they can serve as baselines to show advances in the approaches.

In recent years, several new methods and computational tools have also been introduced. In R package *bcp* [16], Erdman and Emerson implement an efficient Bayesian change point model described by Barry and Hartigan [42]. Ben-Yaacov and Eldar suggest an ultrafast segmentation model based on wavelet decomposition and thresholding in R package *HaarSeg* [15]. Marioni et al. implement a heterogeneous hidden Markov model *bioHMM* [13] in R package *snapCGH*, which can utilize positional information or clone quality in the modeling process and, thus, could be considered as an extension of the *HMM* in package *aCGH*.

Among these models, there has been no comparison study between *bcp*, *bioHMM*, and *HaarSeg* in recent literature. We did not include implementations that are specific to SNP data in our comparison, mainly due to the unique nature of the platform, which is less general in terms of segmentation and may require more inputs like B Allele Frequency (BAF) or genotype call, in addition to the copy number profile in the form of Log R Ratio (LRR).

2.4. Data Simulation for Algorithm Comparison. For the data simulation, we attempt to make it conceptually similar to the scenario one may encounter in real experiments. For copy number studies using CGH or using sequencing with matched case-control sample, three states are commonly assumed, and regions of copy gain and loss are of major interest when sizes range from about 1kb to several megabases [43]. For this purpose, we first create pools of segments for each state; lengths of the segments are sampled from three Poisson distributions, with lambda equal to 20, 270, and 10, respectively. The distance between data points is assumed to be regular and equal to 1. Signal intensities are sampled from three normal distributions, $N_1(r, 1)$, $N_2(2 \times r, 1)$, and $N_3(3 \times r, 1)$ for each state, respectively, with state mean controlled via a ratio factor r varying from 1 to 3 at a step of 1. Segments from different states are then randomly sampled and joined together to form one data sequence.

For sequencing data, to check for splicing and novel transcripts or detect peaks for transcript factor binding sites, one would be interested in distinguishing the true expression signal from the background. Normally, annotated coding or noncoding transcripts are relatively much shorter compared to intergenic regions. In this case, we also first create pools of segments for three virtual states, intergenic, short, and relatively lowly-expressed gene and protein coding sequence with high abundance; lengths of the segments are sampled from three Poisson distributions, with lambda equal to 285, 5, and 10, respectively. Signal intensities for each segment are then sampled from three pools of Poisson distribution, $P_1(1)$, $P_2(r)$, and $P_3(r^2)$, with mean controlled via a ratio parameter r varying from 1.5 to 2 at a step of 0.25 for each pool of segments. Segments from different states are then randomly sampled and joined together to form one data sequence, representing one targeted region.

2.5. Performance Comparison Using ROC Curves. In this work, we compare our model with several well-tested segmentation algorithms, all of which are available as R packages. Since different algorithms tend to be tuned differently

TABLE 2: Area under the ROC curves of simulation 1 data.

	$r = 1$		$r = 2$		$r = 3$		Weighted avg. rank
	AUC _g	AUC _l	AUC _g	AUC _l	AUC _g	AUC _l	
<i>hsmm</i>	0.619487	0.729668	0.982176	0.988689	0.999141	0.998519	3.528400
<i>bcp</i>	0.675283	0.758042	0.912009	0.956031	0.921090	0.963401	6.416511
<i>bioHMM</i>	0.685575	0.875202	0.977079	0.990215	0.995062	0.997551	3.408419
<i>CBS</i>	0.633272	0.795941	0.974008	0.985508	0.996065	0.995409	4.491084
<i>CGHseg</i>	0.586505	0.696329	0.960183	0.991765	0.996045	0.998051	4.621081
<i>GLAD</i>	0.506588	0.548763	0.833011	0.962862	0.986098	0.996577	6.907573
<i>HaarSeg</i>	0.649687	0.763682	0.923416	0.993653	0.995908	0.998408	3.886978
<i>HMM</i>	0.717595	0.854783	0.749492	0.887209	0.526358	0.573611	6.589848

AUC_g and AUC_l are area under the receiver operating characteristic (ROC) curves for simulated gain and loss segments, respectively, for each r .

Weighted avg. rank is calculated as $n + 1 - \sum_{j=1}^{j=c} AUC_i \times \text{rank}^j(AUC_i)/c$ for each model i , where c is the number of AUC columns and n is the number of competing models.

to suit their own methodologies for better sensitivity, here we do not attempt to alter their default settings and feed only the simulated signals without other information to the models, thus achieving an essentially fair comparison and mimicking a common-use case for normal users.

We use simulated data with varying levels of interstate ratio r , which is conceptually similar to signal-to-noise ratio (SNR); since, for both simulations, states with extreme values are of interest, the differences in mean between the extreme states and the intermediate states could be considered as signal, while the variation associated with the intermediate state could be considered as noise. We calculate the true-positive rates (TPR) and the false-positive rates (FPR) over 10000 iterations (100 simulations for each of the 100 random segments formation) of simulation for each level of r .

The TPR is defined as the number of points that are from the states of interest and fall into the predicted states of interest, divided by the total number of points from the states of interest. The FPR is defined as the number of points that are not from the states of interest but fall into the predicted states of interest, divided by the total number of points not from the states of interest. The true states of interest depend on the type of simulation; for normal data in simulation 1, this is assigned to the first and the third states, namely the gain and loss states, respectively. For count data in simulation 2, this is assigned to the third state, which is used to represent signal peak. The prediction is done by comparing the estimated segment mean with a threshold (t) varying from the maximum to the minimum of the simulated value. For abnormal state of gain in simulation 1 and peak in simulation 2, the segment with estimated value above the threshold is considered as positive; for state of loss in simulation 1, the segment with estimated value below the threshold is considered as positive. Definitions of TPR and FPR are formulated as follows:

$$\text{TPR}_{\text{loss}} = \frac{N(x < t \mid s = 1)}{N(s = 1)},$$

$$\text{FPR}_{\text{loss}} = \frac{N(x < t \mid s \neq 1)}{N(s \neq 1)},$$

$$\text{TPR}_{\text{gain}|s2} = \frac{N(x > t \mid s = 3)}{N(s = 3)},$$

$$\text{FPR}_{\text{gain}|s2} = \frac{N(x > t \mid s \neq 3)}{N(s \neq 3)}.$$

(2)

All calculations were carried out in the statistical language R (version 3.0.1). Area under the curve (AUC) was estimated using Bioconductor package ROC (version 1.36.0). The system used for benchmarking is a standard 64 bits Linux desktop with Intel Core i7 with 3.07 GHz and 6 GB DDR3 memory.

3. Results

3.1. Performance Comparison with Simulated Data. After two extensive simulation runs, we show the resulting ROC curves under different signal-to-noise ratios for all compared models in Figure 1. In Figure 2, two sets of randomly simulated data (chosen from the 50th random grid formation and the 50th iteration of that formation), one from each simulation run (using the intermediate r level, 2 for simulation 1 and 1.75 for simulation 2), have been illustrated as an example together with estimated segments from competing models.

In simulation 1 (Table 2), most algorithms—except for *HMM*—perform comparably well at intermediate- and low-noise scenarios. The difference in detecting gain and loss is consistent with our simulation setup, where we intentionally set the loss region to be relatively longer, making it easier to detect. In general, the competing algorithms can be categorized into three performance groups: our model, *bioHMM*, and *HaarSeg* perform best, followed by *CBS* and *cghseg*, and the last three algorithms perform less satisfactorily. Notably, in simulation 1, *bioHMM* has surprisingly high power in a high-noise setting. However, the advantage essentially disappears when signals get stronger. This phenomenon could result from its model selection process, where it attempts to assign a higher number of states, thus more segments, to compensate for the random noise. Additionally, *HaarSeg* has difficulty detecting short gain segments, which could be

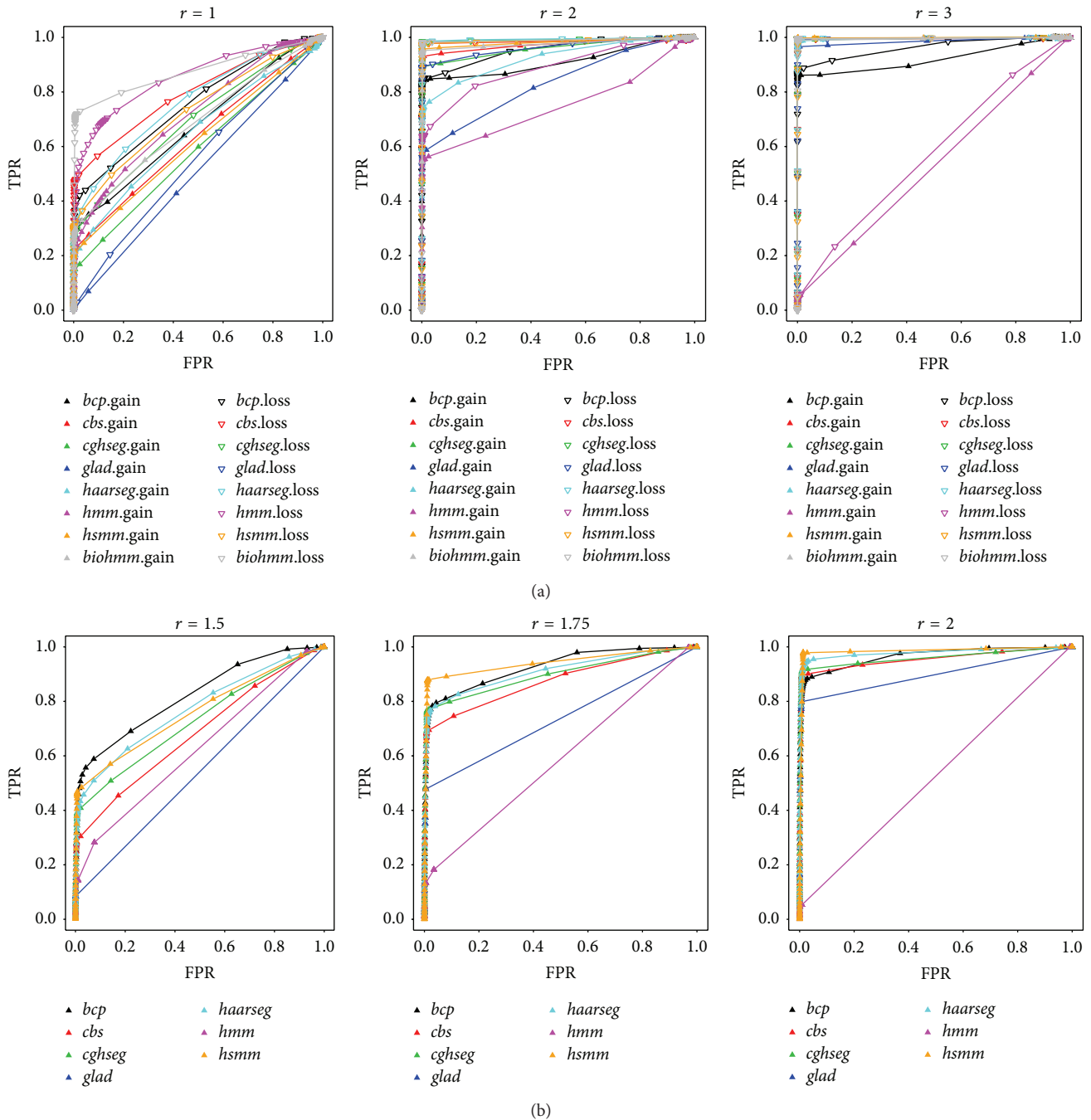


FIGURE 1: ROC curves for performance comparison. Receiver operating characteristic (ROC) curves for segmentation algorithm comparison under different signal-to-noise settings (r). Curves were generated by measuring the sensitivity and specificity at different threshold levels. The x -axis and y -axis show the false-positive rate (FPR) and true-positive rate (TPR), respectively. The upper panel (a) shows simulation 1, similar to an aCGH analysis, and the lower panel shows simulation 2, similar to peak identification using NGS. Compared algorithms are color-coded as indicated in the figure legend, while the up-triangle represents segment of gain in simulation 1 and peak in simulation 2, and hollow down-triangle represents segment of loss in simulation 1. Models are labeled using lowercase letters of their name. Our proposed model is coded as “*hsmm*” for simplicity and the hidden Markov model in package *aCGH* is labeled as “*hmm*.”

related to the default model setting that is not well adapted to short aberrations [15].

A smoothing algorithm like *GLAD* only operates well under higher signal-to-noise ratio. Smoothing results in less accurate segment boundaries. Further, as mentioned in [40],

GLAD is sensitive to single outliers, which explains the minor deficiency of sensitivity in detecting gain regions even for low-noise cases. The behavior of *bcp* indicates that to achieve higher power specificity must be lost, even with a high signal-to-noise setting. *HMM* achieves a high area

TABLE 3: Area under the ROC curves of simulation 2 data.

	$AUC_{r=1.5}$	$AUC_{r=1.75}$	$AUC_{r=2}$	Weighted avg. rank
<i>hsmm</i>	0.7623442	0.9423881	0.9849165	2.232382
<i>bcp</i>	0.8219388	0.9280808	0.9656302	2.616598
<i>CBS</i>	0.6828102	0.874411	0.954873	5.487906
<i>CGHseg</i>	0.7292456	0.8982693	0.9594827	4.550670
<i>GLAD</i>	0.5418485	0.7366826	0.8971196	6.730182
<i>HaarSeg</i>	0.7728192	0.9114189	0.9786983	2.977933
<i>HMM</i>	0.6243222	0.5872855	0.5259849	7.212695

Weighted avg. rank is calculated as $n + 1 - \sum_{j=1}^{j=c} AUC_i \times \text{rank}^j(AUC_i)/c$ for each model i , where c is the number of AUC columns and n is the number of competing models.

TABLE 4: Processing time and error estimate of the compared models.

	avg. <i>t</i>	Simulation 1			Simulation 2		
		avg.cp	MAE	RMSE	avg.cp	MAE	RMSE
<i>hsmm</i>	0.25645	12/11.16	5.756	3.476	6/6.50	18.73	6.31
<i>bcp</i>	1.46298	NA	NA	NA	NA	NA	NA
<i>bioHMM</i>	6.96811	14/14.71	8.655	7.376	NA	NA	NA
<i>CBS</i>	0.12168	12/11.02	7.444	4.178	5/4.93	20.762	7.068
<i>CGHseg</i>	0.28938	12/9.89	9.059	4.783	7/7.00	14.83	5.533
<i>GLAD</i>	0.23725	10/8.22	13.128	6.071	3/4.15	22.139	7.668
<i>HaarSeg</i>	0.00268	17/15.57	12.896	4.984	10/10.65	10.117	4.018
<i>HMM</i>	0.28008	7/38.97	94.666	80.792	1/60.05	144.83	97.178

avg.*t* is calculated as the mean run time of 20000 simulation iterations.

avg.cp is the median/mean number of segments estimated across 3 SNR settings.

MAE is calculated as the mean absolute error $\sum |\text{no.seg} - \text{true.no.seg}|/n$.

RMSE is the rooted mean squared error $\sqrt{\sum (\text{no.seg} - \text{true.no.seg})^2/n}$.

NA indicates that the measurement is not applicable for this algorithm. For *bcp*, the model output posterior means for each position that does not tend to form segments with constant mean. For *bioHMM*, the model cannot be run, thus no results were collected.

under the curve (AUC) when high noise exists ($r = 1$) in simulation 1 and performs comparably worse when signals are stronger, eventually failing to identify most segments. This is in accordance with [40], where *HMM* failed to identify any region in Glioblastoma Multiforme (GBM) data. It also fails to make any meaningful segmentation in simulation 2.

In simulation 2 (Table 3), when data contain a mixture of Poisson distributions, we failed to run *bioHMM* due to an error in a foreign function call to the C library. We have to assume that the implementation cannot work on discrete count data. However, all other implementations are still operable and achieve similar performance as in simulation 1. Though the mean parameter for Poisson data simulation is not considerably large, the normal approximation could still achieve reasonably good power. Nonetheless, our explicit modeling of count data remains advantageous for segmenting count data, which has the highest weighted average rank (Table 4), followed by *bcp* and *HaarSeg*. Compared to *HaarSeg*, the power boost for *bcp* essentially occurs under higher FPR. It is possible that algorithms like *bcp* perform better when a stronger signal exists, which could be due to the normal error assumption in the model.

For both simulations we could confirm that, as has been shown in [40], *cghseg* and *CBS* perform consistently well under various scenarios. Two of the newly introduced methods, *bioHMM* and *HaarSeg*, also exhibit comparable

or better performance; in contrast, our model consistently ranks among the top 3 performing algorithms. Across the two simulations, *HMM* and *GLAD* are considered to possess lowest power, while for *bcp* an associated high error rate is observed.

Concerning computation time, *HaarSeg* is the fastest algorithm among all implementations, by a factor of 50–100; *bioHMM* is the slowest due to its internal model selection process. *bcp* is the second slowest, as a result of long Markov chain Monte Carlo (MCMC) run. The processing time of our model is similar to *cghseg* and is slower than *CBS*, which is about two times faster.

We also took a closer look at the overall accuracy of estimated segment number, in Table 4. For both simulations, we joined, on average, 14 segments into one sequence. Occam's razor states that the best model should be the simplest yet still retain the same power. In simulation 1, our model achieves the lowest rooted mean-squared error (RMSE) and mean absolute error (MAE); in simulation 2, our model finds fewer segments: the median number of detected segments was only 6 across three noise levels. Taking into account the power advantage of our method in the performance comparison, this finding indicates that the estimated segment boundaries are more accurate in our model. Simulated aberrant segments are sampled from the same distribution, and the sojourn modeling in our method

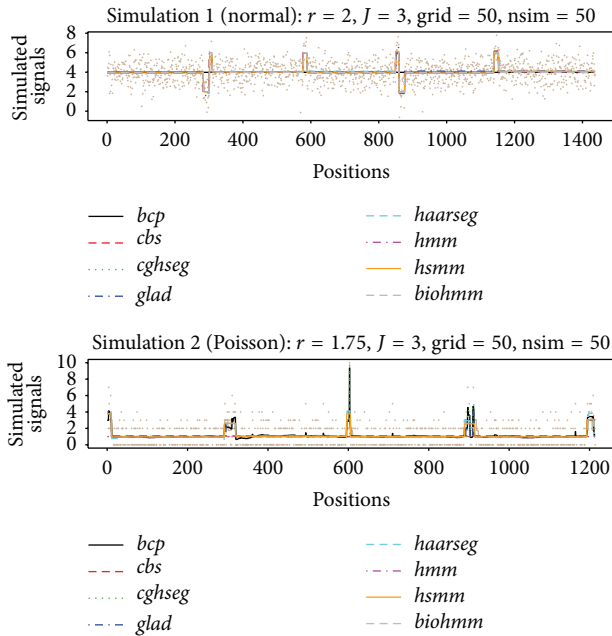


FIGURE 2: Examples of stimulated data and estimated segments. Two sets of randomly simulated data (chosen from the 50th random grid formation and the 50th iteration of that formation), one for each simulation run (using the intermediate r level, 2 for simulation 1 and 1.75 for simulation 2), are illustrated as an example with estimated segments from competing models. Segments are represented using the estimated segment averages. The true underlying grid used for data simulation is shown as the solid line in beige.

takes advantage of this property. In the second simulation, *HaarSeg* achieves the lowest error estimates for both RMSE and MAE. *cghseg* gives error estimates similar to our model. For *HaarSeg* and *cghseg*, this is essentially achieved by fitting more segments. As has been pointed out in [12], assumptions of the mean-variance relationship imposed on the model may lead to more segments to satisfy such requirements.

4. Example of Differentially Methylated Region (DMR) Detection

Differentially methylated regions (DMRs) are genomic regions with different methylation status, that is, variable degrees of DNA methylation between different samples, which has been considered to have regulatory functions for gene transcription [44] and is associated with cell differentiation and proliferation [45, 46]. Such regions can be surveyed using high-throughput technologies like tiling array [47] and sequencing [48].

As an example, we include a set of data extracted from *BiSeq* [49], which contains a small subset of a published study [50], comprising intermediate differential methylation results before DMR detection. We first load the “variosm” data,

```
> library (biomvRCNS)
> data (variosm)
```

The data contains a *GRanges* object *variosm* with two meta columns: “*meth.diff*,” methylation difference between the two

sample groups and “*p.val*,” significance level from the Wald test. Our model could be applied on data from other pipelines as well, using similar data input.

In the *BiSeq* workflow, they use an approach similar to the max-gap-min-run algorithm to define DMR boundaries, by prior filtering and comparing the differential test statistics with a user-specified significance level in the candidate regions. The positional information for methylation sites is accounted for by locating and testing highly correlated cluster regions in the filtering process.

With *biomvRhsmm*, we utilize both types of information to detect DMRs: (1) the difference in the methylation ratio and (2) the significance level from the differential test. The methylation difference gives information about the directionality of the change as well as the size, and the significance level gives the confidence in claiming differential events. We implicitly ask the model to give 3 states, since J is default to 3. Regarding the methylation ratio “*meth.diff*,” these levels may be hypomethylated regions, undefined null regions, or hypermethylated regions, respectively. When modeling significance levels “*p.val*,” these states would represent high confidence regions, low confidence regions, or null results. For both scenarios, we are more interested in extreme states, where we have consistent direction of differences and low P values. However, the distributions of observation values in “*meth.diff*” and “*p.val*” are both nonuniform and asymmetric around 0 (for “*meth.diff*”) and 0.5 (for “*p.val*”), thus we enable the cluster mode for emission prior to initialization by setting *prior.m* = “cluster.” The “cluster” mode will employ the method described in [51] to divide data into clusters and then use the centroid of each cluster to represent the mean parameter; further, the variance structure or other distributional parameters can be estimated using the corresponding clusters.

Due to the nonuniformly located CpG sites, one may split inter-spreading long segments with parameter *max gap* = 100 (see code chunk in Box 1).

After the model fitting, by intersecting regions with extreme “*meth.diff*” and regions with low “*p.val*,” we can locate those detected DMRs, returned with their average “*meth.diff*” and “*p.val*.” Compared to the regions detected in the *BiSeq* vignette, the two sets of regions are largely similar except for two regions. First chr1: 872335, 872386 had highly asymmetric distribution of “*meth.diff*.” Another region, chr2: 46915, 46937, resides in the tail of chromosome 2 and has only 2 methylation sites; this was sorted into the intermediate state due to the lack of support from both the emission level and the sojourn time. However, due to the filtering applied in *BiSeq* workflow, they built wider regions out of a smaller set of more significant sites; in contrast, our approach has more refined regions, and we identified two hypomethylated regions (chr1: 876807, 876958 and chr1: 877684, 877738). The two segmented profiles are depicted in Figure 3, using the default plot method.

5. Discussion

The segmentation problem, in general, occurs in many types of biological experiments and can naturally fit into the hidden

```

# model run
> rhsmm <- biomvRhsmm(x=variosm, maxbp=100, prior.m='cluster', maxgap=100)
> hiDiffgr <- rhsmm@res[mcols(rhsmm@res), 'STATE']!=2
  & mcols(rhsmm@res), 'SAMPLE']=='meth.diff']
# check for direction of changes
> dirNo <- mcols(hiDiffgr), 'STATE']=='1' & mcols(hiDiffgr), 'AVG']>0 |
  mcols(hiDiffgr), 'STATE']=='3' & mcols(hiDiffgr), 'AVG']<0
> hiDiffgr <- hiDiffgr[!dirNo]
# locating low p.val regions
> loPgr <- rhsmm@res[mcols(rhsmm@res), 'STATE']==1
  & mcols(rhsmm@res), 'SAMPLE']=='p.val']
# find common high difference and low p.val regions
> DMRs <- intersect(hiDiffgr, loPgr)
> idx <- findOverlaps(variosm, DMRs, type='within')
> mcols(DMRs) <- DataFrame(cbind(TYPE='DMR', aggregate(as.data.frame(mcols(variosm[idx@queryHits])),
  by=list(DMR=idx@subjectHits), FUN=median)[-1 ]))
> names(DMRs) <- paste0('DMRs', seq_along(DMRs))
> DMRs
GRanges with 5 ranges and 3 metadata columns:
Seqnames      Ranges Strand      TYPE      meth.diff      p.val
<Rle>         <IRanges>  <Rle>      <factor>  <numeric>     <numeric>
DMRs1  chr1 [875227, 875470] * | DMR      0.31947418    6.677193e-06
DMRs2  chr1 [876807, 876958] * | DMR     -0.06108219    6.500328e-02
DMRs3  chr1 [877684, 877738] * | DMR     -0.06123008    2.844639e-02
DMRs4  chr2 [46126, 46280]   * | DMR      0.41008524    1.818530e-07
DMRs5  chr2 [46389, 46558]  * | DMR      0.44823172    1.890819e-06
---
Seqlengths:
chr1 chr2
NA   NA
> plot (rhsmm, gmgr=DMRs)

```

Box 1: Code chunk.

Markov model framework with segment boundaries modeled as transitions between hidden states.

As a generalization of the hidden Markov model, HSMM allows the sojourn distribution to be specified other than the Geometric distribution implicitly used in common HMM. Given the complexity of the genome, such an implicit assumption could be easily violated. Though the true underlying sojourn distributions involving various genomic features remains unknown, our implementation gives more flexible options in the modeling and, thus, might provide more insight.

In this package, several types of sojourn distribution are implemented. For example, with gamma-distributed sojourn, the neighboring position will tend to stay in the same state and transit to other states if far apart. Differing from the original design in [38], our implementation utilizes the positional information naturally associated with most genomic features for the sojourn density estimation. Such an integrative approach is advantageous over simply using the rank of feature positions, since mapping positions are not always uniformly distributed and the spatial patterns may be of interest in experiments like DMR detection. Further, HSMM differs from those models that embed positions in a nonparametric fashion, like *BioHMM* [13] and *QuantiSNP* [19], or as in the “instability-selection” model for LOH analysis [22, 23]; these all employ variations of exponential

function to account for feature position. Our HSMM is closer to the DBN model employed in Segway [34] but is less experiment-specific and easier to interpret and has convenient communication with other analytical and visualization tools within the Bioconductor community.

The explosion of data availability provides another possibility of learning from previous studies. Other than the flat prior commonly used in Bayesian inference, prior information for the sojourn density could be estimated from annotation or previous studies, thus it can be effectively utilized together with positional information of features to guide the estimation of the most likely state sequence.

With its full probabilistic model, various emission densities are provided, enabling the model to handle normally distributed data from traditional array platforms as well as counting data from sequencing experiments. The proposed model has also been applied on a well-studied aCGH dataset from Coriell cell lines [7] and from RNA-seq data generated by the ENCODE project [52, 53] to illustrate its other functionalities in the package vignette.

6. Conclusions

In this work, we present a novel hidden semi-Markov model designed specifically for genomic data analysis. The proposed

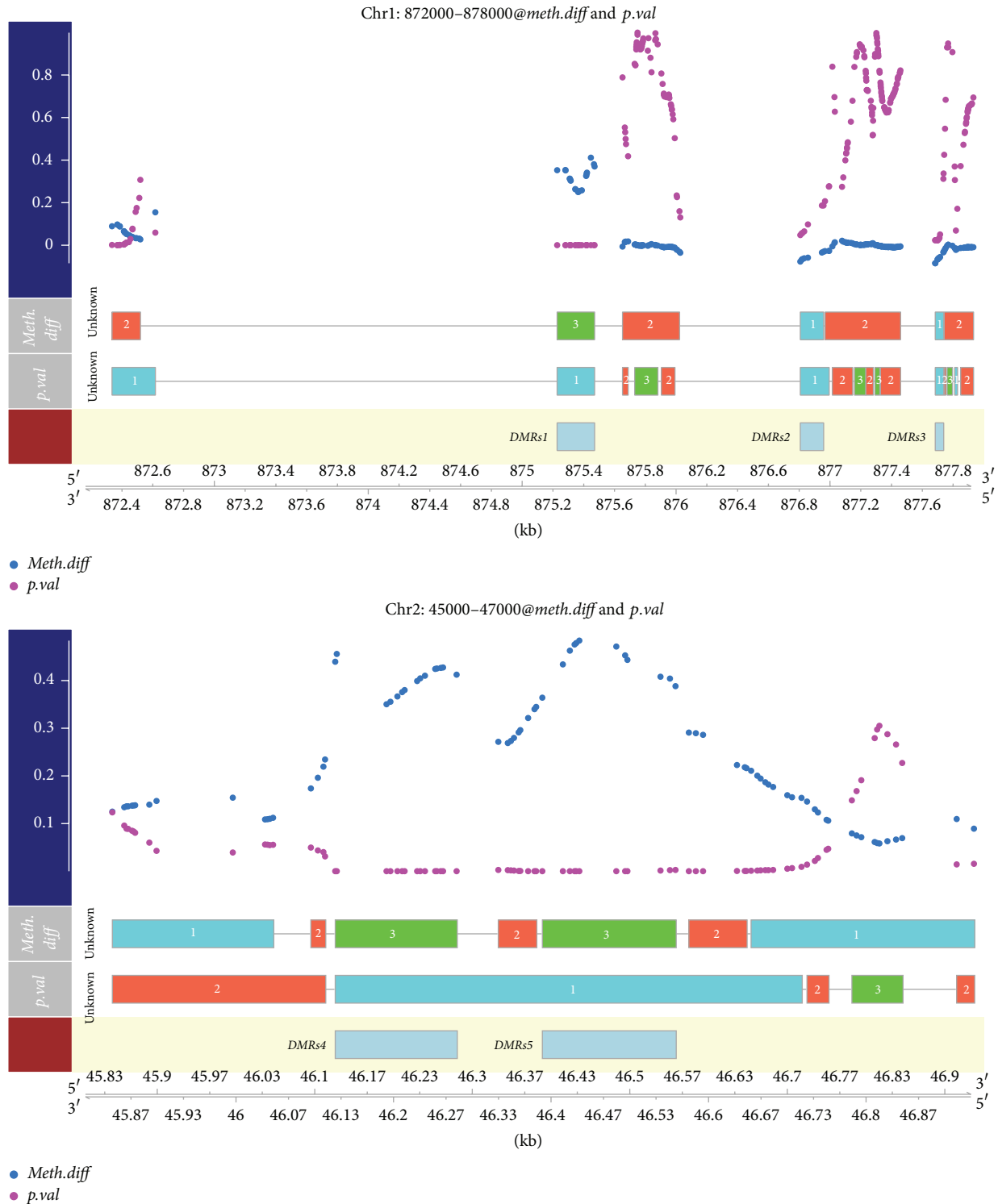


FIGURE 3: Detected differentially methylated regions (DMRs) in the example data, together with estimated segmentation profiles. DMRs could be located by intersecting resulting states “1” or “3” in “*meth.diff*” and segment “1” in “*p.val*,” like has been shown in the code chunk, indicated by boxes in the third row.

model has been compared with several other state-of-the-art segmentation methods; our implementation is efficient and achieves comparable or better sensitivity and specificity in genomic segmentation. Further, our model has flexible

data distribution assumption, enabling a unified interface for segmenting data generated from different experimental platforms. By incorporating genomic positions into the sojourn distribution of HSMM, with optional prior learning

using annotation or previous studies, model output is more biologically sensible. To this end we would like to present our model as a general segmentation engine to serve in a wide range of genomic research.

7. Availability and Requirements

Function *biomvRhsmm* is implemented as part of the R/Bioconductor package *biomvRCNS*, which is available from the Bioconductor project.

Function name: *biomvRhsmm*

Package name: *biomvRCNS*

Project home page:

<http://bioconductor.org/packages/devel/bioc/html/biomvRCNS.html>

Operating system(s): Linux, Mac OS X, Windows

Programming language: R, C

Other requirements: R ($\geq 3.0.0$)

License: GPL (≥ 2).

Conflict of Interests

The authors declare that they have no conflict of interests.

Authors' Contribution

Yang Du implemented the package, performed data analysis, and wrote the paper. Eduard Murani, Siriluck Ponsuksili, and Klaus Wimmers tested the application and provided scientific advice. Klaus Wimmers conceived this study and critically revised the paper. All authors contributed to the discussions for the improvement of the original draft and approved the final paper.

Acknowledgment

This research was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG Wi 1754/14-1).

References

- [1] T. I. Lee, N. J. Rinaldi, F. Robert et al., "Transcriptional regulatory networks in *Saccharomyces cerevisiae*," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.
- [2] A. Doi, I.-H. Park, B. Wen et al., "Differential methylation of tissue- and cancer-specific CpG island shores distinguishes human induced pluripotent stem cells, embryonic stem cells and fibroblasts," *Nature Genetics*, vol. 41, no. 12, pp. 1350–1353, 2009.
- [3] P. Bertone, V. Stolc, T. E. Royce et al., "Global identification of human transcribed sequences with genome tiling arrays," *Science*, vol. 306, no. 5705, pp. 2242–2246, 2004.
- [4] V. Stolc, M. P. Samanta, W. Tongprasit et al., "Identification of transcribed sequences in *Arabidopsis thaliana* by using high-resolution genome tiling arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 12, pp. 4453–4458, 2005.
- [5] L. David, W. Huber, M. Granovskaia et al., "A high-resolution map of transcription in the yeast genome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 14, pp. 5320–5325, 2006.
- [6] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," *Nature Reviews Genetics*, vol. 10, no. 1, pp. 57–63, 2009.
- [7] A. M. Snijders, N. Nowak, R. Segraves et al., "Assembly of microarrays for genome-wide measurement of DNA copy number," *Nature Genetics*, vol. 29, no. 3, pp. 263–264, 2001.
- [8] P. H. Sudmant, J. O. Kitzman, F. Antonacci et al., "Diversity of human copy number variation and multicopy genes," *Science*, vol. 330, no. 6004, pp. 641–646, 2010.
- [9] J. Fridlyand, A. M. Snijders, D. Pinkel, D. G. Albertson, and A. N. Jain, "Hidden Markov models approach to the analysis of array CGH data," *Journal of Multivariate Analysis*, vol. 90, no. 1, pp. 132–153, 2004.
- [10] P. Hupé, N. Stransky, J.-P. Thiery, F. Radvanyi, and E. Barillot, "Analysis of array CGH data: from signal ratio to gain and loss of DNA regions," *Bioinformatics*, vol. 20, no. 18, pp. 3413–3422, 2004.
- [11] 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.
- [12] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin, "A statistical approach for array CGH data analysis," *BMC Bioinformatics*, vol. 6, article 27, 2005.
- [13] J. C. Marioni, N. P. Thorne, and S. Tavaré, "BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data," *Bioinformatics*, vol. 22, no. 9, pp. 1144–1146, 2006.
- [14] E. S. Venkatraman and A. B. Olshen, "A faster circular binary segmentation algorithm for the analysis of array CGH data," *Bioinformatics*, vol. 23, no. 6, pp. 657–663, 2007.
- [15] E. Ben-Yaacov and Y. C. Eldar, "A fast and flexible method for the segmentation of aCGH data," *Bioinformatics*, vol. 24, no. 16, pp. i139–i145, 2008.
- [16] C. Erdman and J. W. Emerson, "A fast Bayesian change point analysis for the segmentation of microarray data," *Bioinformatics*, vol. 24, no. 19, pp. 2143–2148, 2008.
- [17] F. Picard, E. Lebarbier, M. Hoebeke, G. Rigaiil, B. Thiam, and S. Robin, "Joint segmentation, calling, and normalization of multiple CGH profiles," *Biostatistics*, vol. 12, no. 3, pp. 413–428, 2011.
- [18] J. Ding and S. P. Shah, "Robust hidden semi-Markov modeling of array CGH data," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM '10)*, pp. 603–608, Hong Kong, China, December 2010.
- [19] S. Colella, C. Yau, J. M. Taylor et al., "QuantiSNP: an objective bayes hidden-markov model to detect and accurately map copy number variation using SNP genotyping data," *Nucleic Acids Research*, vol. 35, no. 6, pp. 2013–2025, 2007.
- [20] K. Wang, M. Li, D. Hadley et al., "PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data," *Genome Research*, vol. 17, no. 11, pp. 1665–1674, 2007.
- [21] T. Popova, E. Manié, D. Stoppa-Lyonnet, G. Rigaiil, E. Barillot, and M. H. Stern, "Genome Alteration Print (GAP): a tool to visualize and mine complex cancer genomic profiles obtained by SNP arrays," *Genome Biology*, vol. 10, no. 11, article R128, 2009.

- [22] R. Beroukhim, M. Lin, Y. Park et al., “Inferring loss-of-heterozygosity from unpaired tumors using high-density oligonucleotide SNP arrays,” *PLoS Computational Biology*, vol. 2, no. 5, article e40, 2006.
- [23] R. B. Scharpf, G. Parmigiani, J. Pevsner, and I. Ruczinski, “Hidden Markov models for the assessment of chromosomal alterations using high-throughput SNP arrays,” *The Annals of Applied Statistics*, vol. 2, no. 2, article 687, 2008.
- [24] W. Huber, J. Toedling, and L. M. Steinmetz, “Transcript mapping with high-density oligonucleotide tiling arrays,” *Bioinformatics*, vol. 22, no. 16, pp. 1963–1970, 2006.
- [25] A. Piccolboni, “Multivariate segmentation in the analysis of transcription tiling array data,” *Journal of Computational Biology*, vol. 15, no. 7, pp. 845–856, 2008.
- [26] J. Du, J. S. Rozowsky, J. O. Korbelt et al., “A supervised hidden markov model framework for efficiently segmenting tiling array data in transcriptional and ChIP-chip experiments: systematically incorporating validated biological knowledge,” *Bioinformatics*, vol. 22, no. 24, pp. 3016–3024, 2006.
- [27] J. Toedling, O. Sklyar, and W. Huber, “Ringo—an R/Bioconductor package for analyzing ChIP-chip readouts,” *BMC Bioinformatics*, vol. 8, article 221, 2007.
- [28] M. J. Lercher, A. O. Urrutia, and L. D. Hurst, “Clustering of housekeeping genes provides a unified model of gene order in the human genome,” *Nature Genetics*, vol. 31, no. 2, pp. 180–183, 2002.
- [29] H. Caron, B. Van Schaik, M. Van der Mee et al., “The human transcriptome map: clustering of highly expressed genes in chromosomal domains,” *Science*, vol. 291, no. 5507, pp. 1289–1292, 2001.
- [30] Y. Zhang, T. Liu, C. A. Meyer et al., “Model-based analysis of ChIP-Seq (MACS),” *Genome Biology*, vol. 9, no. 9, article R137, 2008.
- [31] C. Spyrou, R. Stark, A. G. Lynch, and S. Tavaré, “BayesPeak: bayesian analysis of chip-seq data,” *BMC Bioinformatics*, vol. 10, article 299, 2009.
- [32] Z. S. Qin, J. Yu, J. Shen et al., “HPeak: an HMM-based algorithm for defining read-enriched regions in ChIP-Seq data,” *BMC Bioinformatics*, vol. 11, article 369, 2010.
- [33] J. Ernst and M. Kellis, “ChromHMM: automating chromatin-state discovery and characterization,” *Nature Methods*, vol. 9, no. 3, pp. 215–216, 2012.
- [34] M. M. Hoffman, O. J. Buske, J. Wang, Z. Weng, J. A. Bilmes, and W. S. Noble, “Unsupervised pattern discovery in human chromatin structure through genomic segmentation,” *Nature Methods*, vol. 9, no. 5, pp. 473–476, 2012.
- [35] G. Klambauer, K. Schwarzbauer, A. Mayr 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 Research*, vol. 40, no. 9, article e69, 2012.
- [36] R Core Team, *R: A Language and Environment For Statistical Computing*, Vienna, Austria, 2013.
- [37] R. C. Gentleman, V. J. Carey, D. M. Bates et al., “Bioconductor: open software development for computational biology and bioinformatics,” *Genome Biology*, vol. 5, no. 10, p. R80, 2004.
- [38] Y. Guédon, “Estimating hidden semi-markov chains from discrete sequences,” *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, pp. 604–639, 2003.
- [39] M. Lawrence, W. Huber, H. Pagès et al., “Software for computing and annotating genomic ranges,” *PLoS Computational Biology*, vol. 9, no. 8, Article ID e1003118, 2013.
- [40] W. R. Lai, M. D. Johnson, R. Kucherlapati, and P. J. Park, “Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data,” *Bioinformatics*, vol. 21, no. 19, pp. 3763–3770, 2005.
- [41] N. R. Zhang and D. O. Siegmund, “A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data,” *Biometrics*, vol. 63, no. 1, pp. 22–32, 2007.
- [42] D. Barry and J. A. Hartigan, “A Bayesian analysis for change point problems,” *Journal of the American Statistical Association*, vol. 88, no. 421, pp. 309–319, 1993.
- [43] P. Stankiewicz and J. R. Lupski, “Structural variation in the human genome and its role in disease,” *Annual Review of Medicine*, vol. 61, pp. 437–455, 2010.
- [44] E. Li, T. H. Bestor, and R. Jaenisch, “Targeted mutation of the DNA methyltransferase gene results in embryonic lethality,” *Cell*, vol. 69, no. 6, pp. 915–926, 1992.
- [45] W. Reik, W. Dean, and J. Walter, “Epigenetic reprogramming in mammalian development,” *Science*, vol. 293, no. 5532, pp. 1089–1093, 2001.
- [46] R. A. Irizarry, C. Ladd-Acosta, B. Wen et al., “The human colon cancer methylome shows similar hypo- and hypermethylation at conserved tissue-specific CpG island shores,” *Nature Genetics*, vol. 41, no. 2, pp. 178–186, 2009.
- [47] X. Zhang, S. Shiu, A. Cal, and J. O. Borevitz, “Global analysis of genetic, epigenetic and transcriptional polymorphisms in *Arabidopsis thaliana* using whole genome tiling arrays,” *PLoS Genetics*, vol. 4, no. 3, Article ID e1000032, 2008.
- [48] S. J. Cokus, S. Feng, X. Zhang et al., “Shotgun bisulphite sequencing of the Arabidopsis genome reveals DNA methylation patterning,” *Nature*, vol. 452, no. 7184, pp. 215–219, 2008.
- [49] K. Hebestreit, M. Dugas, and H.-U. Klein, “Detection of significantly differentially methylated regions in targeted bisulfite sequencing data,” *Bioinformatics*, vol. 29, no. 13, pp. 1647–1653, 2013.
- [50] T. Schoofs, C. Rohde, K. Hebestreit et al. et al., “DNA methylation changes are a late event in acute promyelocytic leukemia and coincide with loss of transcription factor binding,” *Blood*, vol. 121, no. 1, pp. 178–187, 2013.
- [51] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344, John Wiley & Sons, 2009.
- [52] Consortium TER, “The ENCODE (ENCyclopedia of DNA elements) project,” *Science*, vol. 306, no. 5696, pp. 636–640, 2004.
- [53] E. P. C. ENCODE Project Consortium, R. M. Myers, J. Stamatoyannopoulos et al., “A user’s guide to the encyclopedia of DNA elements (ENCODE),” *PLoS Biology*, vol. 9, no. 4, Article ID e1001046, 2011.