IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 8, 2020, accepted February 2, 2020, date of publication February 5, 2020, date of current version February 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2971863

dpGMM: A Dirichlet Process Gaussian Mixture Model for Copy Number Variation Detection in Low-Coverage Whole-Genome Sequencing Data

YAOYAO LI, JUNYING ZHANG, (Member, IEEE), XIGUO YUAN, AND JUNPING LI
School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Corresponding authors: Junying Zhang (jyzhang@mail.xidian.edu.cn) and Xiguo Yuan (xiguoyuan@mail.xidian.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61571341, and in part by the Natural Science Foundation of Shaanxi Province in China under Grant 2017JM6036.

ABSTRACT Comprehensive identification and cataloging of copy number variation (CNVs) are essential to providing a complete view of human genetic variation and to finding diseased genes. Due to the large-scale sequencing and cost control whole-genome sequencing (WGS) data, low-coverage data is favorably disposed towards CNV identification. However, such low-coverage data is sensitive to noise and sequencing biases, which results in low resolution of CNV detection in past experimental designs for WGS datasets. In this paper, we present a control-free Dirichlet process Gaussian mixture model (dpGMM) based approach, to analyze the read depth (RD) of low-coverage WGS datasets for CNV discovery. First, noise and biases of the RD signals are corrected through the preprocessing step of dpGMM. Then we assume that RD signals across genomic regions follow a Gaussian mixture model (GMM) in which each Gaussian distribution is followed by a copy number state. Without requiring the number of Gaussian distributions, dpGMM builds a Dirichlet process (DP) GMM for RD signals and further uses a DP prior to infer the number of Gaussian models. After that, we apply dpGMM to simulation datasets with different coverages and individual datasets, and compare ours to three widely used RD-based pipelines, CNVnator, GROM-RD, and BIC-seq2. Simulation results demonstrate that our approach, dpGMM, has a high F1 score in both low- and high-coverage sequences. Also, the number of overlaps between CNVs detected in real data by ours and the standard benchmark is twice as much as that detected by other tools such as CNVnator and GROM-RD.

INDEX TERMS Copy number variation, Dirichlet process, Gaussian mixture model, read depth, low coverage.

I. INTRODUCTION

Copy number variations (CNVs), as an important form of structural variations, have gained considerable interest in genetic and functional analysis of human genome variation. Several large-scale studies have reported CNV participates in phenotypic variation and adaptation by disrupting genes and altering gene dosage [1], [2]. Some CNVs are remained by normal individuals while others are implicated in many diseases including Parkinson [3], diabetes mellitus [4], Autism [5], Alzheimer [6], and cancer [7]. Thus, comprehensive identification and cataloging of CNVs from whole-genome sequencing (WGS) data contributes significantly to research on human diversity and disease.

The rapid development of next-generation sequencing (NGS) technology has provided an unprecedented opportunity for genome-wide analysis of CNVs on the scale of whole-genome. Due to the cost control, low-coverage data is often favored in genome-wide variation analysis. However, read depth (RD) signals from low-coverage data are sensitive to systematic noises, and sequencing biases, which may cause false CNV calls using RD-based methods.

Several types of biases are found in the generation and pretreatment of NGS data. In the NGS process, the first deviation introduced into raw reads is sequencing bias. During the library preparation, PCR duplicates, and sequencing, NGS platforms generally generate short reads or read pairs

The associate editor coordinating the review of this manuscript and approving it for publication was Liangtian Wan.

with sequencing bias [8]. The sequencing machine makes an erroneous call, or due to properties of the sequenced DNA (e.g., homopolymers) [9]. Then, some raw reads from high SNP densities are aligned to wrong locations in the reference by current mapping algorithms, where mapping bias arises [10]. Moreover, extreme GC composition (GC-rich, or GC-poor) in genome regions results in uneven read coverage or no coverage produced by NGS platform [11], which is the so-called GC bias. Besides, genomic regions with high sequence degeneracy show lower mapped read coverage than unique regions, creating systematic bias [12]. Gaps and Indels in genomic sequences also influence read coverage, resulting in a low number of false-positive calls.

For the biases, numerous RD-based CNV detection methods have been proposed in recent years, with a common assumption that the number of reads aligned to a genomic region is proportional to the copy number of this region [13]–[16]. Based on this assumption, these RD-based methods find different strategies for RD signals normalization to resolve sequencing biases, uneven GC content bias, and other noises. Although some methods perform well on CNV detection, many studies show there are still several limitations of discovering CNV events with specific structural characteristics [17]. For example, a popular CNV tool, CNVnator [18] combines the established mean-shift approach with additional refinements (multiple-bandwidth partitioning and GC bias correction) to discover broad CNVs from low- or high- coverage data, which is suitable for large size of CNVs detection. For 'N' bases in genome that are not sequenced or incompletely identified, CNVnator regards them as losses, which results in a high false positive rate and a low precision. Another control-free RD-based method, GROM-RD [19], analyzes multiple genomic biases in read coverage and divides the genome into size-varying overlapping segments to improve the breakpoint accuracy of CNVs. Although GROM-RD considering multiple biases has a high specificity, it cannot discovery some CNVs, especially at low coverage datasets. In addition, a common RD-based method BIC-seq2 [20] normalizes the sequencing data by considering the GC content, the nucleotide composition of the short reads and the mappability. BIC-seq2 performs normalization at a nucleotide level, resulting in its high sensitivity of detection for small CNVs, but with a little bit low precision.

With careful consideration of the issues above, we develop dpGMM, a control-free low-coverage sequences RD-based method with the Dirichlet process (DP) Gaussian mixture model (GMM) algorithm for CNV detection. In this study, we first filter out the reads with sequencing bias and mapping bias and use nonoverlapping sliding windows to divide the genomic sequences into segments. Since CNV is a form of structural variations, we count the RD signal of a segment (window) not the reads of a genomic locus. Ideally, the genomic loci among a window share the same copy number, except for the breakpoint of a CNV. If the breakpoint of a CNV locates in the window, the genomic loci among the window do not share the same copy number. Then, the average number of reads of each segment is calculated as the RD value of that segment based on several studies [21], [22]. GC-correction and smoothing RD signals of all segments are implemented for further removing biases. There is an assumption that the observed sequencing RD signals of all segments follow a GMM where a Gaussian model represents a copy number state and the number of Gaussian components is unknown. Without requiring the number of Gaussian distributions, dpGMM takes the DP as the prior distribution to establish a DP GMM for RD signals based on Jordan's study [23]. In this way, we discriminate the state of copy number distinctly with high sensitivity and specificity, as evidenced by comparisons of dpGMM to three most commonly used control-free RD tools, GROM-RD, CNVnator, and BIC-seq2. dpGMM shows improved predictive abilities for CNVs and excellent scalability for different simulated and real NGS datasets.

II. METHODOLOGY

The pipeline of dpGMM consists of three parts: (1) Biases correction and Segmentation; (2) Integrated dpGMM; (3) Estimation of copy numbers. We now detail the three parts.

A. BIAS CORRECTION AND SEGMENTATION

To remove the sequencing bias and mapping bias, we map raw reads to a reference genome and discard raw reads with quality scores <Q20 by Burrow-Wheeler Aligner (BWA) [24]. Then we divide the entire genome into nonoverlapping segments by sliding windows of equal size, which is a binning process, not a segmentation that combines adjacent RDs (bins) that share the same copy number. A segment is the result of a sliding window. We use the average of read counts in the segment as its raw RD signal on the basis of other studies [21], [22]. For GC-content bias, we employ the median algorithm similar to another study [18] to correct this bias:

$$R_i' = \frac{R_{all}^{mean}}{R_{gc}^{mean}} \times R_i \tag{1}$$

where $R_i^{'}$ is the corrected RD signal of the i-th segment, R_i is the raw RD signal of the i-th segment, R_{all}^{mean} is average RD signal over all segments, and R_{gc}^{mean} is the average RD signal over all segments with the same GC content as the segment.

For furthermore rectifying the systematic bias, all GC-corrected RD signals ($R_i^{'}$) are smoothed using cghFLasso implemented in the R package [25]. The cghFLasso uses a fused lasso regression, which seeks coefficients minimizing a loss function consisting of three terms: the sum of square error, the sum of absolute value of regression coefficients (the lasso penalty), and the sum of the absolute difference between contiguous coefficients (the fused penalty) [26], [27]. The fused lasso is developed for situations when predictors in the regression model have some kinds of natural ordering. The lasso penalty controls the total number of nonzero coefficients in the model [27]. Fig. 1 displays the RD signals

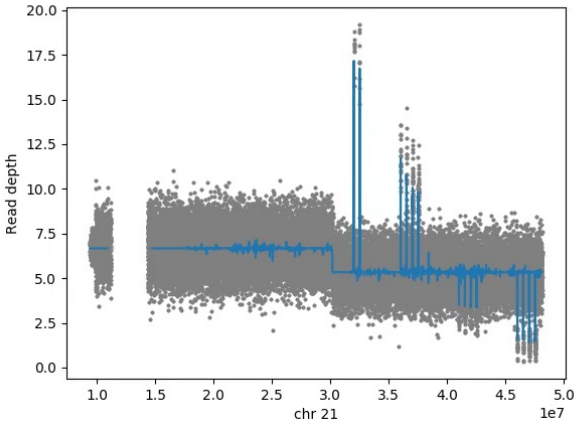

FIGURE 1. cghFLasso is applied to some RD signals data in a simulated sample. The grey points present the GC-corrected RD signals, and the solid blue line shows the smoothed RD signals using cghFLasso.

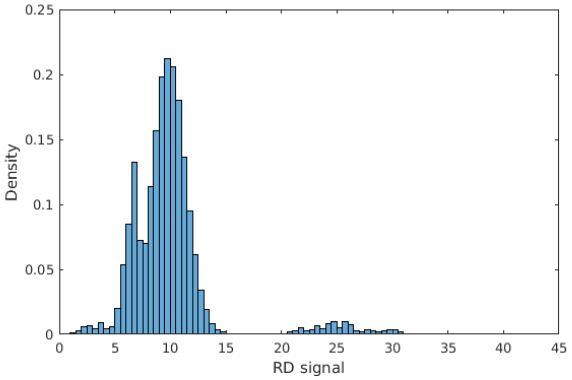

FIGURE 2. The frequency histogram of RD signals from an ovarian cancer sample (chr21, EGAD00001000084).

before and after the smoothing process. By using the cghFLasso algorithm, the locally adjacent segments with similar constancy of RD signals are merged into a partition, where some systematic noises are balanced out.

Considering the importance of genomic positions, we combine smoothed RD signals and their corresponding genomic positions to transform the smoothed RD signals in one-dimensional space into a two-dimensional profile. The details of this transformation are described in our previous study [28]. In this way, we can observe RD signals from both horizontal and vertical levels, which reflect copy number amplitude and positional space, respectively.

B. INTEGRATED DPGMM

We use $X = \{x_1, x_2 \ldots, x_N\}$ represents the above two-dimensional RD signals of the observed sample, where N denotes the number of partitions in this sample and x_i is a vector, which denotes the smoothed RD signal and the corresponding genomic position of the i-th partition in this sample. Here we assume that smoothed RD signals across the entire genome follow a GMM ideally, where a Gaussian distribution model represents a state of copy number [17]. Here we display the frequency histogram of RD signals from an ovarian cancer sample (chr21, EGAD00001000084) in Fig. 2. We find that the RD signals maybe follow a GMM, in which there may be four Gaussian models. By calculating the maximum probability of the RD signal of a partition generated by Gaussian distribution models, we determine the specific copy number of that partition. Suppose X is a mixture of K Gaussian distributions (K is unknown). To solve K, the implicit variable $s = \{s_1, s_2, \ldots, s_N\}(s_i \in \{1, \ldots, K\})$ is defined to represent the Gaussian distribution's label of the partition in this paper. Let $s_i = k$ denotes the i-th partition belongs to the k-th Gaussian component, and let $p(\cdot)$ represents a Gaussian component with a parameter of θ_k, $\theta_k = \{\mu_k, \sigma_k^2\}$ in the GMM, where μ_k, σ_k^2 are the mean and variance of the k-th component separately. π_k is defined to represent the weight of the k-th component, where $\pi_k \geq 0, k = \{1, \ldots, K\}$, and $\sum_{k=1}^{K} \pi_k = 1$. At this point, let a parameter set $\Theta = \{\pi_1, \ldots \pi_K; \theta_1, \ldots, \theta_K\}$ denotes all parameters of the GMM to be determined. RD signals of partitions can be established as:

$$\begin{cases} p(x_i|\Theta) = \sum_{k=1}^{K} \pi_k p(x_i|\theta_k) \\ p\left(x_i \mid \theta_k\right) = N(x_i; \mu_k, \sigma_k^2) \end{cases} \tag{2}$$

The i-th partition is labeled by the following (3):

$$\tau_i = arg \max_{k \in K} \left(p\left(\theta_k \mid x_i\right)\right) \tag{3}$$

where $p\left(\theta_k \mid x_i\right)$ represents the probability of the k-th Gaussian distribution that x_i belongs to, and τ_i is the label of the i-th partition. Thus, our goal is to estimate the parameter set Θ of this GMM and calculate τ_i.

At present, there are two kinds of methods for the hyper-parameter estimate: Expectation-maximization (EM) algorithm, and nonparametric Bayesian method. The EM algorithm is mainly used to estimate parameters of a finite mixed model by adopting the maximum likelihood criterion. GMM based on EM algorithm requires a prior of the number of Gaussian models in advance and is easy to fall into local optimum. Once the number of Gaussian models is roughly set as three because of three copy number states (loss, neutral, and gain), there are many coarse CNV calls with a high false positive rate [17]. The second method employs the DP as the prior distribution, establishes an infinite DP GMM of RD signals, and constructs the posterior distribution of parameter set Θ by Monter-Carlo Markov chain (MCMC) sampling methods or by variational methods. The methodology of MCMC sampling provides a systematic approach to compute likelihoods and posterior distributions, but it can be slow to converge and their convergence can be difficult to diagnose [29], [30]. One class of alternative variational inference methods seeks an approximate distribution (variational distribution) of a posterior distribution and converts the inference of the posterior distribution into an optimization problem for minimum distance between these two distributions. Compared with MCMC sampling methods, the

variational inference method is prohibitively faster and its convergence time is independent of dimensionality for the range of data [29]. In this study, we regard the DP as the prior distribution and adopt the mean-field variational inference algorithm [29] to tackle the parameter set Θ of the posterior distribution.

DP is generated from Dirichlet distribution, the extension of Dirichlet distribution in continuous space, and is a measure on measure [23]. The DP comprises of a base distribution G_0 and a positive scaler α, which is shown as (4):

$$\mathrm{G} \sim \mathrm{DP}(G_0, \alpha) \tag{4}$$

where G is a random distribution produced from a DP with the base distribution G_0 and concentration scaler α. The larger α is, the closer G is to G_0. Therefore, we present the infinite DP GMM for RD signals of a sample:

$$\begin{cases} x_i \sim p(x_i|\theta_{s_i})(i = 1, \ldots, N) \\ \theta_{s_i} \sim G \\ G \sim DP(G_0, \alpha) \end{cases} \tag{5}$$

where $p\left(x_i \mid \theta_{s_i}\right)$ is the probability of a Gaussian distribution with parameter θ_{s_i}. θ_{s_i} is the parameter θ_k of the Gaussian model that the i-th partition belongs to, where $s_i = k$.

According to Bayesian theory:

$$p(\Theta|X, \alpha, G_0) \propto p(X|\Theta, \alpha, G_0) \times p(\Theta) \tag{6}$$

we estimate parameters (Θ, α, G_0) of (6) by computing its prior distribution $p(\Theta)$ and likelihood function $p(X|\Theta, \alpha, G_0)$. $p\left(X \mid \Theta, \alpha, G_0\right)$ is also the posterior distribution of data. Here we use DP to determine the prior distribution for every parameter in Θ, and use the variational inference algorithm to approximate the posterior probability $p\left(X \mid \Theta, \alpha, G_0\right)$.

For computing $p\left(X \mid \Theta, \alpha, G_0\right)$, we first introduce latent variables W and integrate W for $p\left(X \mid \Theta, \alpha, G_0\right)$. Considering the DP GMM with parameter set (Θ, α, G_0), latent variables W, and observations X. The posterior distribution of the latent variables W is:

$$\begin{aligned} p\left(W \mid X, \Theta, \alpha, G_0\right) = exp\{&logp\left(X, W \mid \Theta, \alpha, G_0\right) \\ &- logp(X|\Theta, \alpha, G_0)\} \end{aligned} \tag{7}$$

Next, the problem of computing the posterior distribution $p\left(W \mid X, \Theta, \alpha, G_0\right)$ is transformed as an optimization problem [29]. In this work, we find a variational distribution q to approximate the posterior distribution p, and minimize the Kullback-Leibler (KL) divergence between these two distributions:

$$\begin{aligned} &\mathrm{D}(q_v\left(\mathrm{W}\right)|p(W|X, \Theta, \alpha, G_0)) \\ &= E_q\left[logq_v\left(W\right)\right] \\ &\quad - E_q\left[logp\left(W, X|\Theta, \alpha, G_0\right)\right] + logp(W, X|\Theta, \alpha, G_0) \end{aligned} \tag{8}$$

where let $q_v\left(W\right)$ is the variational distributions with free variational parameters ν.

The minimization in (8) can be converted as the maximization of a lower bound on the log marginal likelihood [23]:

$$\begin{aligned} logp\left(X \mid \Theta, \alpha, G_0\right) \geq &E_q\left[logp\left(W, X|\Theta, \alpha, G_0\right)\right] \\ &- E_q\left[logq_v\left(W\right)\right] \end{aligned} \tag{9}$$

Here the posterior $p\left(X \mid \Theta, \alpha, G_0\right)$ is calculated by the mean-field framework based on the stick-breaking representation of the DP mixture proposed by Jordan [23], [31]. We provide a detailed explanation of the equation from (7) to (9) in Appendix file 1. The specific details are described in Jordan's research [23] and Pedregosa's scikit-learn [32]. Thus, we obtain the optimal parameter set of the DP GMM for the observed RD signals.

In the process of solving the parameters of the DP GMM model, some parameters need to be introduced to help initialize the model, such as the initialization of α and K. Among these parameters, α is represented by the "weight_concentration_prior", and K is represented by "n_component" in the DP GMM. We will discuss these parameters in the "*RESULT*" Section. Also, the window size of each segment when we compute RD signals will also be discussed in the "*PARAMETER SETTINGS*" subsection of the "*RESULTS*" section.

C. ESTIMATION OF COPY NUMBERS

After acquiring the DP GMM for RD signals, we assign the copy number for each Gaussian component using the formula mentioned in previous work [8] for the RD signal of each segment as follows:

$$\frac{CN_k}{2} = \frac{\mu_k}{RD_{mean}} \tag{10}$$

where CN_k, μ_k represent the rounded integer copy number and the mean value of k-th Gaussian component, respectively. Here we suppose the sequenced normal human genome is diploid. RD_{mean} is the mean value of all RD signals in the DP GMM and is calculated by the following (11):

$$\begin{aligned} RD_{mean} &= \int xp\left(x\right) d\left(x\right) = \int x \sum \pi_k p\left(x \mid \theta_k\right) dx \\ &= \sum \pi_k \int xp\left(x \mid \theta_k\right) dx = \sum \pi_k \mu_k \end{aligned} \tag{11}$$

To quantify the copy number of a partition i, we just calculate τ_i using (3) mentioned in Section "Integrated dpGMM" to determine which Gaussian component the partition belongs to, whose copy number is the copy number of the partition. Especially, partitions whose copy number are larger than 2, are regarded as CNV gains. Otherwise, partitions with 0- or 1- copy are treated as CNV losses. Aiming at continuous partitions, we merge the partitions with the same copy number into a CNV call.

III. RESULT

To evaluate the performance of dpGMM, we apply it three simulation datasets and two real datasets. Meanwhile, we compare the performance of dpGMM with CNVnator

TABLE 1. Summary of simulated and real datasets.

Dataset	Read length	Variation size	Coverage	Reference
Simulation1	100	gain == loss	4x, 6x, 10x, and 20x	hg19-chr21
Simulation2	100	gain >> loss	4x, 6x, 10x, and 20x	hg19-chr21
Simulation3	100	gain>>loss	4x and 10x	hg19-chr11
6 individuals	101	unknown	5x	hg19
EGAD00001000084	unknown	unknown	low coverage	hg19

GROM-RD, and BIC-seq2 on these datasets. The reason why we choose CNVnator and GROM-RD for the comparative study is explained as follows: Firstly, CNVnator and GROM-RD are two widely used RD-based models for CNV discovery in the low depth of coverage WGS data. Our proposed dpGMM is also based on RD to detect CNV from low-coverage WGS data. Secondly, CNVnator and GROM-RD do not require a control sample or multiple samples when analyzing CNVs. Our proposed dpGMM is also a control-free CNV detection method. Thirdly, CNVnator, GROM-RD, and our method all consider the biases of NGS technologies, but use different strategy to normalize the RD signals and build different algorithms to discover CNVs. In addition, we also add comparisons with another popular RD-based method BIC-seq2 for CNV detection in this study, since BIC-seq2 is also popular and normalizes RD signals by considering GC contents, the nucleotide composition of the short reads and the mappability to reduce biases of NGS technologies.

In this section, there are mainly six subsections: (1) *DATASETS*, (2) *EVALUATION MEASUREMENTS* (3) *PARAMETER SETTINGS*, (4) *SIMULATION DATASETS*, (5) *REAL DATASETS*, and (6) *ALGORITHM METRICS*.

A. DATASETS

To test the performance of dpGMM for predicting CNV regions, we use both three simulated datasets (with known CNV labels), six individual genomes (NA12878, NA12891, NA12892, NA19238, NA19239, and NA19240) real data (with a great many of validated CNVs) and 22 ovarian cancer samples (Table 1). We first compare our method with three commonly used RD-based approaches, GROM-RD, CNVnator, and BIC-seq2 on simulated datasets.

We adopt the recently developed software IntSIM [33] to simulate two groups of simulation datasets under various configurations, which considering the sequencing bias, GC-content bias, and complex variations. There are several studies using chromosome 21 (chr21) to generate simulated data and using 50 replicated samples of chr21 for the statistical analysis of the performance [21], [22]. Firstly, the public chr21 of human reference genome hg19 is taken as the reference input of the IntSIM. Considering normal tissue cells may be involved in the sequenced pathological tissue samples, we set the proportion of normal cells as 0.2. Given the CNV detection performance of an algorithm on low-coverage sequencing data, we set low sequence coverages vary from 4x to 6x and high sequence coverages vary from 10x to 20x to configure the simulations. Moreover, taking the influence of CNV length on CNV identification into account, we simulate CNV lengths in our simulation varying from 10kb to 50kb. Furthermore, due to the imbalanced RD signals, such as containing a large number of amplified fragments and a small number of deletion regions, the performance of CNV detection methods is also influenced. On consideration of this imbalance, we use this simulator to generate two simulation datasets: each sample in simulation1 embeds 6 gains and 8 losses (gain == loss, balance simulated data); each sample in simulation2 is comprised of 26 gains and 10 losses (gain $\gg$ loss, unbalanced simulated data). Gains range from 3 to 8 copy numbers and Losses vary from 0 to 1 copy number (homozygous and heterozygous). In addition, we use a larger chromosome 11 (chr11) with a different CNV pattern (12 gains and 10 losses) to generate samples at 4x coverage and 10x coverage, separately as the simulation3 datasets. We simulate 50 replicated samples for each coverage level in each simulation dataset for estimating the average performance of the algorithm and avoiding the contingency of the algorithm. All simulated samples are aligned to hg19 using BWA.

Next, we compared GROM-RD, CNVnator, BIC-seq2, and dpGMM on chr17 of individual genome NA12878 [34] and on chr21 of six individual genomes with 5x coverage depth downloaded from The International Genome Sample Resource (IGSR). The standard benchmark of the real dataset is referred to as the collection of Mill's study [35] and Altshuler *et al.*'s study [34], embedding some experimentally validated and high confidence CNVs. We test the performance of the four algorithms using this standard benchmark.

In addition, we also apply the dpGMM to the genome-wide (22 autosome chromosome) of 22 ovarian cancer samples from the EGA archive (https://www.ebi.ac.uk/ega/home) under accession EGAD00001000084. These samples are aligned and formatted with the BAM files. Meanwhile, we also compare the performance of dpGMM with CNVnator, and BIC-seq2 on these ovarian cancer samples' sequencing data.

B. EVALUATION MEASUREMENTS

To better access the performance of the proposed approach and compare it to peer popular methods (GROM-RD, BIC-seq2, and CNVnator) on current NGS data, we employ the following commonly used measurements: recall, precision, specificity, F1-score, and G-mean.

$$F1 - score = \frac{2 \times recall \times precision}{recall + precision} \tag{12}$$

$$G - mean = \sqrt{recall + specificity} \tag{13}$$

Here recall (also called the true positive rate, TPR) reflects the ability of a model to discover CNVs in the dataset, while

the precision represents the proportion of CNVs detected by the model to real CNVs of ground truth. However, when the recall and precision of a model are unbalanced, it is difficult to compare the performance of the model by using these two measurements. F1 score takes into account both the precision and recall of a model and is considered as the harmonic average of the two indicators, ranging from 0 to 1. Nevertheless, some methods have high true positive rates on CNV discovery, but there are also many false positive calls. At this time, the F1 score cannot adequately describe this appearance. In this work, we choose G-mean combining recall and specificity (1-false positive rate) to comprehensively display these two critical aspects of the method's performance on imbalanced datasets. Especially, the performance of the algorithm is the average value of a measurement (TPR, FPR, F1 score, and G-means) of this algorithm over 50 replicated simulations with the same level coverage.

C. PARAMETER SETTINGS

We discuss parameters of dpGMM, and one of the most important of these parameters is the number of Gaussian components (''n_component''). ''n_component'' is the K parameter mentioned in ''*Integrated dpGMM*'' section. Users need to set a value as ''n_component'' in dpGMM, even though the n_component does not match the true generative distribution of the dataset. Additionally, the appropriate window size in Section ''*BIAS CORRECTION AND SEGMENTATION*'' can also influence the detection performance of CNV in dpGMM. Many existed RD-based CNV detection methods regard the window size as 1000 bp or other fixed size, since the definition of CNV is a duplication or deletion of DNA segment of size more than 1000 bp [18], [36], [37]. However, the window size is a critical parameter and is supposed to depend on the RD, as it adjusts the trade-off between detection resolution and robustness to noise. To calibrate the parameters for ''n_component'' and ''window size'', we first use 50 samples from the simulation1 dataset with coverage of 6x and window sizes ranging from 0.5kbp to 5kbp to compute the performance of our method (Fig. 3 and Appendix file 1: Fig. 1S). It should be noted that 0.5kbp is not the up-bound resolution of the proposed method theoretically. Users can set the ''window_size'' parameter by themselves. Fig. 3 shows dpGMM performs best when the ''n_component'' is 4 and the ''window size'' is 1kbp. When plotting the TPR and FPR in Figure 3, we set the value of TPR of FPR as null instead of the average of TPRs or FPRs of all samples with the same coverage, if dpGMM cannot detect CNVs on some conditions in a certain sample with coverage of 6x. Therefore, there are no results for some conditions. In this way, we find 1kbp is the optimal window size on dataset with coverage of 6x. In addition, we also apply the dpGMM with window size ranging from 0.5kbp to 5kbp to analysis of simulation1 dataset with coverage of 4x, 10x, and 20x and simulation2 datasets. The results are shown in Appendix file 2, which demonstrates dpGMM has a good performance with window size of 1kbp. In addition, 1kbp also confirm to the definition of length of

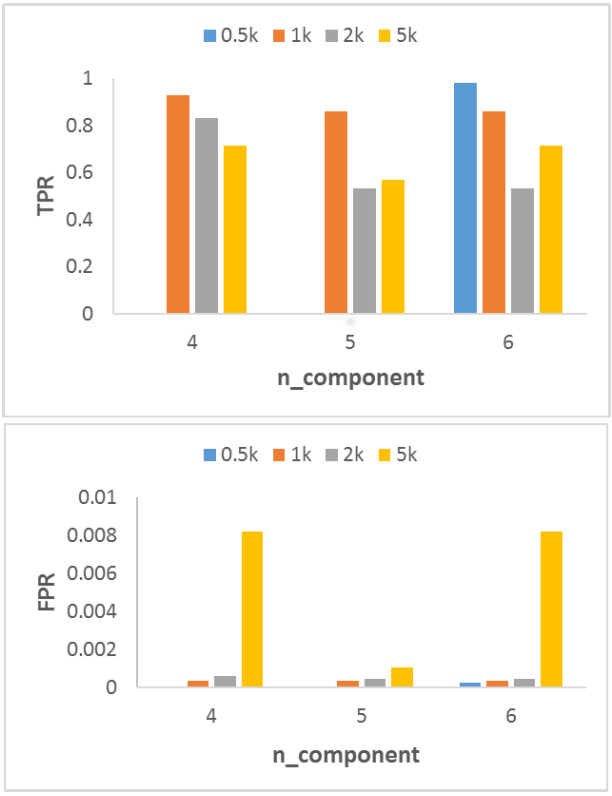

FIGURE 3. Performance of dpGMM on simulation1 dataset (6x coverage depth) with different parameter settings (''n_component'', ''window size'').

CNV mentioned above. When the coverage of sequencing data is uncertain, especially in real dataset, we can use 1kbp as a priority of window size. Here, we also set ''covariance_type'' is ''full'' (Appendix file 1: Fig. 2S), and the concentration parameter α (''weight_concentration_prior'') of dpGMM is None. Users can also set these parameters as needed. Taken together, we use the optimal parameter set (''n_componet'': 4, ''window size'': 1kbp, ''covariance_type'': full, and ''weight_concentration_prior'': None) on all simulation datasets. For GROM-RD and CNVnator, 100 base-window [19] and the default 1k base-window are suitable for all datasets [18], separately. For BIC-seq2, we choose 1kbp as the expected window size. To ensure the fairness of comparison, the aforementioned optimal parameter settings of all algorithms are adopted in both simulated and real datasets.

D. SIMULATION RESULTS

The major motivation of applying dpGMM is to automatically determine the number of mixture components from data. To clarify the flexibility of dpGMM model for CNV detection task, we perform a comparison of the dpGMM with basic GMM of pre-determined k. Here we set k to 3 and 7, respectively. In the first case, k equals 3, which means there are three copy number states: loss, neutral, and gain. In the latter case, we regard k as 7, which represents CNs of diploid, heterozygous deletion, homozygous deletion and 3, 4, 5,

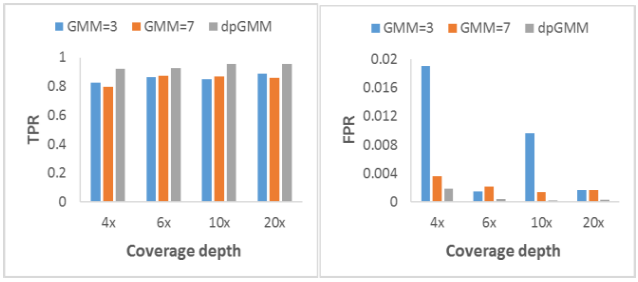

FIGURE 4. Performance of dpGMM, basic GMM of pre-determined 3 (GMM = 3), and basic GMM of pre-determined 7 (GMM = 7) on simulation1 datasets. The three bars (cyan, orange and grey) respectively represent the TPR (left panel) and the FPR (right panel) of the three CNV detection tools (dpGMM, GMM = 3, and GMM = 7).

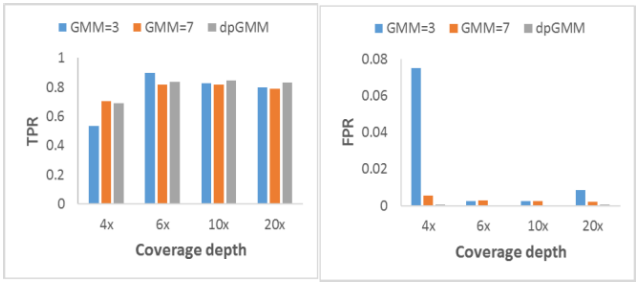

FIGURE 5. Performance of dpGMM, basic GMM of pre-determined 3 (GMM = 3), and basic GMM of pre-determined 7 (GMM = 7) on simulation2 datasets. The three bars (cyan, orange and grey) respectively represent the TPR (left panel) and the FPR (right panel) of the three CNV detection tools (dpGMM, GMM = 3, and GMM = 7).

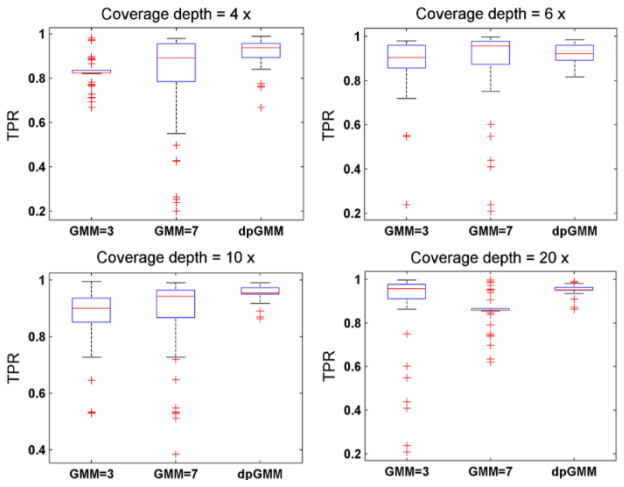

FIGURE 6. Performance of dpGMM, basic GMM of pre-determined 3 (GMM = 3), and basic GMM of pre-determined 7 (GMM = 7) on simulation1 datasets.

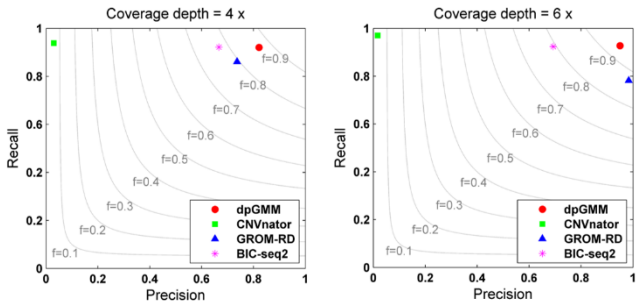

FIGURE 7. F1 scores of dpGMM, CNVnator, GROM-RD, and BIC-seq2 on simulation1 datasets with low coverage (4x: left panel and 6x: right panel). The four marks (red circle, green square, blue triangle, and purple star) respectively represent the F1 scores of the four CNV detection tools (dpGMM, CNVnator, GROM-RD, and BIC-seq2). The grey lines indicate different levels of F1 score between 0.1 and 0.9.

6 copies (Even larger copy gains might be very rare). We have computed TPR and FPR of results of the basic GMM with the two cases on both simulation 1 datasets and simulation 2 datasets in Fig. 4 and Fig. 5, respectively. The comparison results show that the dpGMM model performs better than the basic GMM of pre-determined k (3, and 7). We also display the boxplots of results on simulation1 datasets, shown as Fig.6, which demonstrates that the proposed dpGMM remains stable.

To have a fair comparison, we also compare the performance of the proposed method dpGMM with CGHcall [38] that employs the GMM to detect CNVs. The comparison results are displayed in Appendix file 1 and demonstrate that dpGMM still has a comprehensive performance.

Next, the analysis of GROM-RD, CNVnator, BIC-seq2, and dpGMM on the simulation1 dataset with low coverage (4x and 6x) are shown in Fig. 7 (left panel and right panel). When the reciprocal overlap of detected CNV calls with true calls of simulation is more than 10%, the detected CNV calls are regarded as true positive calls. Each mark in Fig. 7 represents an average performance of 50 samples from simulatation1 datasets. The overall F1 score of our method is between 0.87 - 0.94, the recall between 0.92 - 0.927, and the precision between 0.82 - 0.95 when coverage depth is low. Although CNVnator correctly identifies more known variations than dpGMM and GROM-RD, it has the least true positives. GROM-RD can detect more true positives than ours, but its sensitivity is inferior to that of our method on 6x coverage sequences. BIC-seq2 has a higher recall with a lower precision compared with our method. Particularly, the precision of dpGMM is the best among the four tools on 4x coverage data. Taken together, our proposed method has the best trade-off between sensitivity and precision to reliable calls on low coverage simulation1 datasets with different variations.

Applying these four methods to CNV detection of high-coverage simulation1 datasets (Fig. 8), we find the performance of CNVnator and BIC-seq2 are similar to those of low-coverage datasets. GROM-RD runs CNV calls with higher precision than the other three methods on both 10x and 20x coverage datasets. Compared with GROM-RD, our dpGMM can discover more known CNVs more accurately.

Although the F1 score can accurately reflect the algorithm's performance in terms of recall and precision, it cannot report the false discovery positives. For this reason, we calculate the G-means of all four methods depicted in Fig. 9.

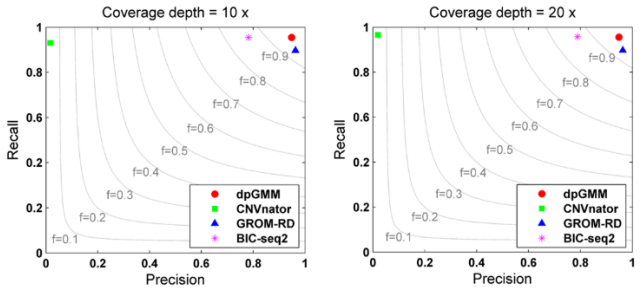

FIGURE 8. F1 scores of dpGMM, CNVnator, GROM-RD, and BIC-seq2 on simulation1 datasets with high coverage (10x: left panel and 20x: right panel). The four marks (red circle, green square, blue triangle, and purple star) respectively represent the F1 scores of the four CNV detection tools (dpGMM, CNVnator, GROM-RD, and BIC-seq2). The grey lines indicate different levels of F1 score between 0.1 and 0.9.

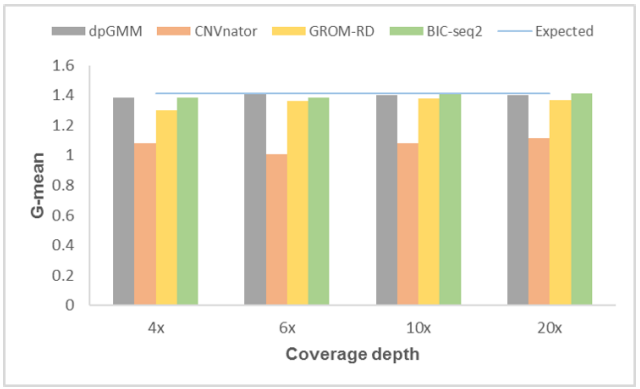

FIGURE 9. G-means of dpGMM, CNVnator, GROM-RD, and BIC-seq2 on simulation1 datasets with different coverage. The light blue horizontal line represents the optimal G-mean value. The bar charts (grey, orange, dark yellow, and green) are the G-means of the four methods (dpGMM, CNVnator, GROM-RD, and BIC-seq2) separately.

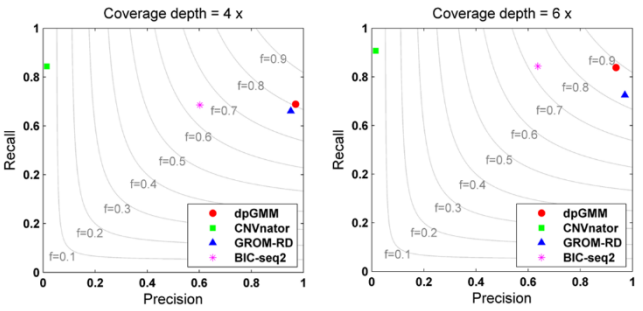

FIGURE 10. F1 scores of dpGMM, CNVnator, GROM-RD, and BIC-seq2 on simulation2 datasets with low coverage (4x: left panel and 6x: right panel). The four marks (red circle, green square, blue triangle, and purple star) respectively represent the F1 scores of the four CNV detection tools (dpGMM, CNVnator, GROM-RD, and BIC-seq2). The grey lines indicate different levels of F1 score between 0.1 and 0.9.

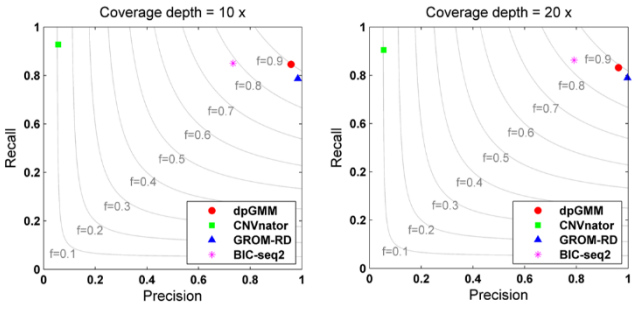

FIGURE 11. F1 scores of dpGMM, CNVnator, GROM-RD, and BIC-seq2 on simulation2 datasets with high coverage (10x: left panel and 20x: right panel). The four marks (red circle, green square, blue triangle, and purple star) respectively represent the F1 scores of the four CNV detection tools (dpGMM, CNVnator, GROM-RD, and BIC-seq2). The grey lines indicate different levels of F1 score between 0.1 and 0.9.

Since CNVnator takes the 'N' bases in the genome sequence as losses, there are many false positives in this approach, resulting in a low G-mean of CNVnator. On low-coverage datasets, our method and BIC-seq2 have slightly higher G-means than GROM-RD. From this point of view, dpGMM is also a good choice for CNV detection with few false positives.

We apply the four approaches to simulation2, the unbalanced dataset embedding 26 gains and 10 losses, respectively. F1 scores of the four algorithms on simulation2 with low coverage (4x and 6x) and high coverage (10x, and 20x) are displayed in Fig. 10 and Fig. 11 respectively. Next, their G-means are exhibited as Fig. 3S in Appendix file 1. From Fig. 10 and Fig. 11, we find CNVnator has an excellent capability of recall but with a poor ability to call true positives as with the results on simulation1 datasets, which is also illustrated by the G-means in supplementary Fig. 3S. BIC-seq2 has a similar recall but with a lower precision, compared with dpGMM and GROM-RD. dpGMM and GROM-RD have outstanding accuracy rates for both low and high coverage data although they have slightly lower recall rates than CNVnator. CNVnator regards the "N" bases in genomic sequences as deletions, which results in very low precision. The accuracy of variation calls detected by GROM-RD is marvelous on all coverage datasets, except in the case of 4x coverage. Compared with GROM-RD, our method performs well with higher F1 score on 4x coverage datasets. In a word, all simulated experimental results show that the proposed approach has a good comprehensive performance of recall, precision, and specificity.

We also apply the four approaches to simulation3, embedding 12 gains and 10 losses, respectively. F1 scores of the four algorithms on simulation3 with low coverage (4x) and high coverage (10x) are depicted in Fig. 12. In Fig. 12, our proposed dpGMM has the highest F1 score among these four methods. G-means of these methods on simulation3 are computed in Fig. 4S. These results demonstrate that dpGMM has a comparable performance in simulation3, which also give us a similar conclusion as before. Finally, we also provide the variance of all simulation results in Table 1S-3S in Appendix file 1, which shows that our method has the similar robustness to other comparative methods.

E. REAL-DATA RESULTS

To extend dpGMM on real data, firstly we analyze chr17 of an individual genome NA12878 with 5x coverage. We regard

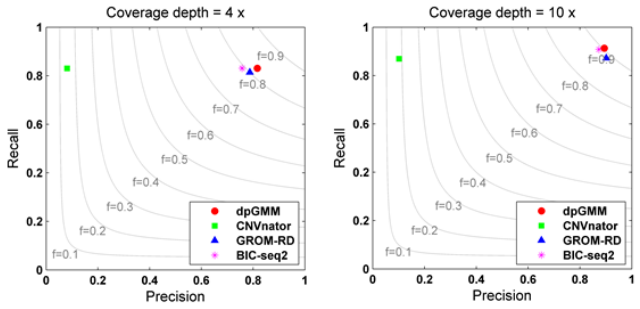

FIGURE 12. F1 scores of dpGMM, CNVnator, GROM-RD, and BIC-seq2 on simulation3 datasets with low coverage (4x, left panel) and high coverage (10x, right panel). The four marks (red circle, green square, blue triangle, and purple star) respectively represent the F1 scores of the four CNV detection tools (dpGMM, CNVnator, GROM-RD, and BIC-seq2). The grey lines indicate different levels of F1 score between 0.1 and 0.9.

TABLE 2. Comparison of methods with the standard benchmark (21 CNVs) on NA12878 (chr17).

Algorithm	Recall	True positives	others
GROM-RD	3/21	3	60
CNVnator	3/21	3	30
dpGMM	6/21	6	26
BIC-seq2	5/21	5	106

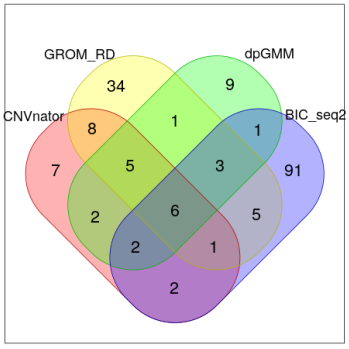

FIGURE 13. CNV calls of the four methods on NA12878 (chr17). The different color circles (yellow, red, green, and purple) represent the results of four different methods (GROM-RD, CNVnator, dpGMM, and BIC-seq2), separately. Shadow parts are overlapping CNV collections of different methods.

at least a 10% overlap between a predicted CNV and the standard benchmark as an overlap. We also compare the quality of CNVs dpGMM detected with those by other mentioned tools on this real data, as shown in Table 2. ''others'' in Table 2 are the CNVs detected by a method but not overlapped with the standard benchmark. To ensure fair comparisons, default parameter settings are used for all algorithms, except for the window size in GROM-RD. Using the window size described in the previous GROM-RD paper, we identify 100-base windows for chr17 of the NA12878 genome sequence. Table 2 displays that dpGMM finds six overlaps, while BIC-seq2, CNVnator, and GROM-RD discover five, three, and three overlaps, separately. Again, dpGMM has the highest recall rate for CNVs in low-coverage data.

Since some new CNVs may not be found in the standard benchmark, we calculate overlaps of the CNV calls detected by the four tools in Fig. 13. BIC-seq2 detects the most CNVs (111 CNVs) and GROM-RD detects the second most CNVs (63 CNVs) from this real data compared with the remaining two methods. Although dpGMM has fewer CNV calls than GROM-RD and BIC-seq2, dpGMM is the most consistent with the standard benchmark. For further quantification of overlaps, we also employ the Overlap Density Score (ODS) measurement proposed by Yuan *et al.* [21] to estimate each method. The ODS is defined as (the mean number of CNVs of one method overlapped with other methods) $\times$ (the mean overlapping times for the total number of detected CNVs). As for dpGMM in analysis of NA12878, the ODS is $[(15+15+12)/3] \times [(15+15+12)/29] = 20.27$. Similarly, the ODS of GROM-RD, the ODS of BIC-seq2, and that of CNVnator are calculated as 21.37, 4.313 and 13.23, respectively. Among the four algorithms, dpGMM achieves a moderate ODS and CNVnator implements the highest ODS. Additionally, to analyze CNVs detected by our method but not by other methods, we annotate all detected CNV regions via the annotation software ANNOVAR [39] and the GeneCard human gene database (Appendix file 3). Some CNV calls are confirmed to be related to some pathologic processes. For example, the copy number loss region (chr17: 39420001-39429000), only detected by our algorithm, is a location of the SYNRG gene, which is a protein-coding gene. The GeneCard database reports SYNRG-related diseases including chromosome 17Q12 Deletion Syndrome. In NA12878, we find a copy number loss in the SYNRG region, suggesting that this individual may be carrying disease-causing genes.

Next, we analyze chr21 of six samples (NA12878, NA12890, NA12891, NA19238, NA19239, and NA19240) downloaded from the 1000 Genome Project as complement real datasets. The quality of CNVs dpGMM detected with those detected by other mentioned tools on these real datasets, are shown in Table 3 (NA12878, chr21), Table 4 (NA19239, chr21), and Table 4S-7S in Appendix file 1. In Table 3 and Table 4, BIC-seq2 calls the greatest number of true positives (9 CNVs and 38 CNVs), and GROM-RD calls the second number of true positives. Although BIC-seq2 and GROM-RD detect more true positives than our method and CNVnator, they also detect more others. In Table 3, dpGMM detects only one true positive less than GROM-RD, but has the highest precision among these four methods. In Table 4 and Table 4S-7S, dpGMM and CNVnator also have higher precision than the other two methods. In addition, we compute the ODS for each method's results on these six samples, which are shown as Table 8S in Appendix file 1. In most cases, our method dpGMM has the highest ODS value. We have a similar conclusion as before.

Additionally, we provide the analysis of CNVs on the genome-wide (22 autosome chromosome) of 22 ovarian cancer samples by using our proposed dpGMM, CNVnator, and BIC-seq2. The overlaps of CNVs detected by three methods

TABLE 3. Comparison of methods with the standard benchmark (20 CNVs) on NA12878 (chr21).

Algorithm	Recall	True positives	others
GROM-RD	7/20	7	110
CNVnator	4/20	4	9
dpGMM	6/20	6	13
BIC-seq2	9/20	9	749

TABLE 4. Comparison of methods with the standard benchmark (65 CNVs) on NA19239 (chr21).

Algorithm	Recall	True positives	others
GROM-RD	24/65	24	77
CNVnator	7/65	7	9
dpGMM	7/65	7	18
BIC-seq2	38/65	38	429

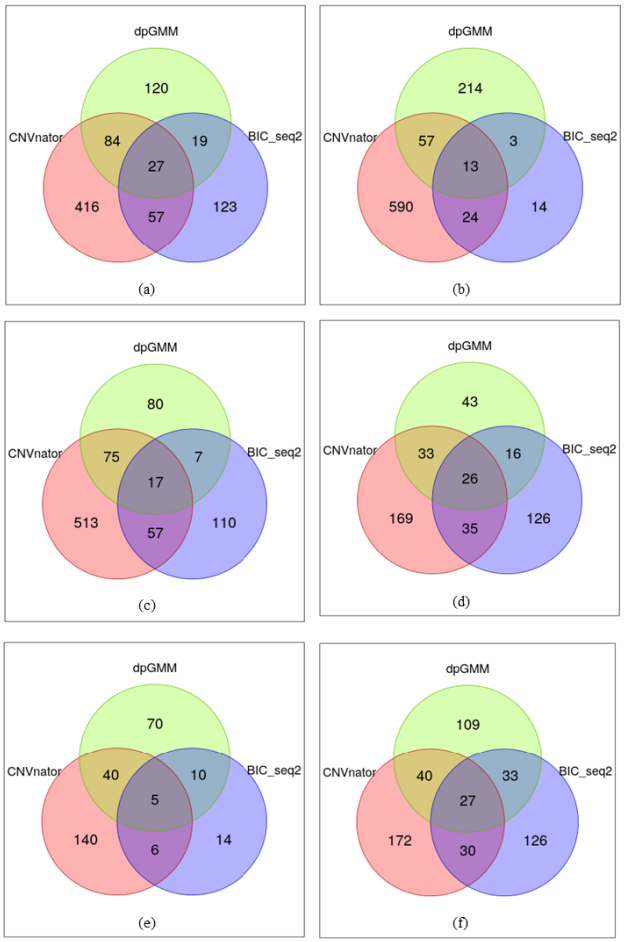

FIGURE 14. CNV calls of the three methods on whole-genome (22 autosome chromosomes) data of ovarian cancer samples. (a)∼(f) represent the results on samples with suffix of id: 2053_1, 2561_1, 2815_3, 1743_8, 1752_2, and 1752_8, respectively. The different color circles (red, green, and purple) represent the results of three different methods (CNVnator, dpGMM, and BIC-seq2), separately. Shadow parts are overlapping CNV collections of different methods.

are depicted in Fig. 14 and Fig. 5S-6S in Appendix file 1. The results show that dpGMM identifies a modest number of CNVs and displays a relatively higher overlapping density than others. Since there are no ground truth for these samples, we also compute the ODS for each method's results on different sample data, which are shown as Table 5. In most cases, our method dpGMM has the highest ODS value.

TABLE 5. Comparison of methods on whole-genome (22 autosome chromosomes) data of 22 ovarian cancer samples.

Sample_id	CNVnator	BIC-seq2	dpGMM
EGAR00001004837_2044_2	31.77	14.83	37.16
EGAR00001004802_2053_1	32.55	37.39	49.3
EGAR00001004836_2561_1	8.38	36.75	16.06
EGAR00001004837_2561_2	20.61	16.73	32.23
EGAR00001004838_2561_3	34.72	19.44	22.89
EGAR00001004839_2561_5	26.84	18.69	25.87
EGAR00001004851_2815_1	16.38	18.05	7.56
EGAR00001004852_2815_2	48.86	28.99	70.31
EGAR00001004853_2815_3	20.81	25.14	37.58
EGAR00001004854_2815_5	13.5	14.08	24.01
EGAR00001004857_2884_2	27.4	13	25.04
EGAR00001004895_3705_2	37.07	31.07	45.3
EGAR00001004896_3705_3	23.87	19.8	32.7
EGAR00001004897_3865_3	46.98	27.77	49.56
EGAR00001005411_1743_8	27.38	26.13	43.2
EGAR00001005450_1752_1	69.42	35.23	57.2
EGAR00001005451_1752_2	8.21	9.66	14.4
EGAR00001005452_1752_3	32.17	27.06	51.69
EGAR00001005453_1752_5	24.4	14.5	31.4
EGAR00001005454_1752_6	34.51	16.01	40.5
EGAR00001005455_1752_7	64.5	55.5	71.4
EGAR00001005456_1752_8	28.58	31.69	38.58

In summary, dpGMM identifies a modest number of CNVs in real data and displays a relatively higher overlapping density than others.

IV. ALGORITHM METRICS

Both simulation datasets and real datasets are conducted on a PC with an Inter (R) Celeron (R) CPU G1840 @ 2.80GHz processor and 8GB of memory. The input is a sorted BAM file. Run times for the four methods on chr17 of NA12878 sequences are listed as 1219s (GROM-RD), 621s (dpGMM), 632s (CNVnator), and 474s (BIC-seq2). Therefore, dpGMM is a comparable faster tool to analyze CNVs, except for BIC-seq2. The source code is written in python and is freely available at https://github.com/tudui123/dpGMM/.

V. DISCUSSION AND CONCLUSION

In this paper, we propose dpGMM, an alternative RD-based pipeline for CNV detection from low-coverage WGS data. dpGMM comprehensively normalizes RD signals, builds a GMM for RD signals to discover CNVs. The key idea of the dpGMM method is that it analyzes RD signals and establishes a DP GMM for the RD signals, which adopts a DP as prior to solve the DP GMM, instead of giving a prior of the number of Gaussian components. By this means, the effective number of Gaussian components can be inferred from the RD signal data, which allows us to recognize specific copy number states for RD signals of genomic regions. Besides, comprehensive biases are considered in the "*BIAS CORRECTION*

AND NORMALIZATION'' step, making RD signals more normal, which helps improve the accuracy of dpGMM in the discovery of homo-deletions and losses.

The novelties of our model are presented as follows: (1) our model assumes that RD signals across genomic regions from a single sample follow a GMM in which each Gaussian distribution represents a copy number state; (2) our model does not require the given number of Gaussian distributions but uses a DP prior to infer the number of Gaussian models.

The new and differences in the proposed method compared to other CNV detection methods that use GMM approaches are listed below: (1) Many existed CNV detection methods using GMM approaches are based on array CGH data or Affymetrix 6.0 SNP array data [38], [27], [40]–[51], while our proposed method analyzes NGS data; (2) Several existed CNV detection methods build a GMM for multiple samples' data [38], [40], [47], [49], while our method builds a GMM for RD signals from a single sample; (3) Many existed CNV detection methods using GMM approaches need to give the number of Gaussian components in a GMM, such as 3 or 6 [41], [43], [45]–[47], [50], [51], while the proposed method does not require that.

We demonstrate the performance of the proposed method by using both simulation and real sequencing datasets. The simulation results show that the proposed method performs better than three popular peer methods in terms of F1 score and G-mean. Moreover, real data results show that dpGMM discovers a modest number of CNV calls and has a higher consistency with the standard benchmark than the other three methods. Thus, our method is an effective and powerful bioinformatics tool, which can be used for the identification of genomic structure variations on WGS data.

Five points of the dpGMM algorithm should be mentioned here. First, we initialize the parameter of ''weight_concentration_prior'' of dpGMM to None, due to without the prior of this parameter. However, specifying different values for the concentration prior will make the dpGMM model put different weights on Gaussian components at the beginning of the algorithm. Users can choose the parameter settings by themselves. Second, since the complexity of tumor samples, such as tumor aneuploidy and tumor heterogeneity, dpGMM needs to be further improved when applied to the analysis of tumor samples. Third, because of gaps between exons, it is not suitable for the proposed method to analyze exome-sequencing data, as dpGMM is designed to cope with WGS data. For the future work, we plan to seek the optimal concentration prior and analyze other important hyper-parameters in this model on sequencing datasets with complex variations for the improvement of dpGMM. Moreover, dpGMM is based on a fixed-window to normalized RD signals for CNV detection. However, for RD-based CNV detection methods, especially for low-coverage data, the bin size (window size) is a critical parameter, as it adjusts the trade-off between detection resolution and robustness to noise. Also, mappability bias is also related to the choice of window size and read-length. For mappability bias, we just filter out these reads with mapping quality scores below 20 for simplicity [52]. In the future work, we will also design a RD normalization strategy combining automatic window size and mappability bias reduction to discover CNVs.

VI. APPENDIX

Appendix file 1: Supplementary figures. Fig. 1S. G-means of dpGMM on simulation1 datasets (6x coverage depth) with different parameter settings (''n_component'', ''window size''). **Fig. 2S.** BIC scores of dpGMM on simulate1 datasets (6x coverage depth) with the different parameters of ''covariance_type''. **Fig. 3S.** G-means of dpGMM, CNVnator, GROM-RD, BIC-seq2 on simulation2 datasets with different coverage. **Fig. 4S.** G-means of dpGMM, CNVnator, GROM-RD, and BIC-seq2 on simulation3 datasets with different coverage. **Fig.5S. and Fig.6S.** CNV calls of three methods on whole-genome (22 autosome chromosomes) data of ovarian cancer samples. **Table 1S-3S.** The variances of results on simulation1 datasets, simulation2 datasets, and simulation3 datasets, respectively. **Table 4S-7S.** Comparison of methods with the standard benchmark on chr21 of NA12891, NA12892, NA19238, and NA19240, respectively. **Table 8S.** Comparison of methods on six samples. **Appendix file 2:** Performances of dpGMM with different parameter settings on all simulation datasets. **Appendix file 3:** Annotations of CNV calls detected by CNVnator, dpGMM, GROM-RD, and BIC-seq2.

ACKNOWLEDGMENT

Y. Li thanks the joint Editor and referees for their insightful comments, which greatly improve the presentation of the article. Y. Li also thanks the Prof. W. Wang of North Carolina State University to help edit the English writing of this manuscript.

REFERENCES

[1] The Wellcome Trust Case Control Consortium, N. Craddock, M. E. Hurles, N. Cardin, R. D. Pearson, and V. Plagnol, ''Genome-wide association study of CNVs in 16,000 cases of eight common diseases and 3,000 shared controls,'' *Nature*, vol. 464, pp. 713–720, Apr. 2010.

[2] C. Preuss and G. Andelfinger, ''Genetics of heart failure in congenital heart disease,'' *Can. J. Cardiol.*, vol. 29, no. 7, pp. 803–810, Jul. 2013.

[3] Y.-X. Gui, Z.-P. Xu, W. Lv, J.-J. Zhao, and X.-Y. Hu, ''Evidence for polymerase gamma, POLG1 variation in reduced mitochondrial DNA copy number in Parkinson's disease,'' *Parkinsonism Rel. Disorders*, vol. 21, no. 3, pp. 282–286, Mar. 2015.

[4] M. Prabhanjan, R. V. Suresh, M. N. Murthy, and N. B. Ramachandra, ''Type 2 diabetes mellitus disease risk genes identified by genome wide copy number variation scan in normal populations,'' *Diabetes Res. Clin. Pract.*, vol. 113, pp. 160–170, Mar. 2016.

[5] C. Piochon, A. D. Kloth, G. Grasselli, H. K. Titley, H. Nakayama, V. Wan, D. H. Simmons, T. Eissa, J. Nakatani, A. Cherskov, and K. Hashimoto, ''Cerebellar plasticity and motor learning deficits in a copy-number variation mouse model of autism,'' *Nature Commun.*, vol. 5, no. 1, 2015, Art. no. 5586.

[6] X. Zheng, F. Y. Demirci, M. M. Barmada, G. A. Richardson, O. L. Lopez, M. I. Kamboh, E. Feingold, and R. A. Sweet, ''Genome-wide copy-number variation study of psychosis in Alzheimer's disease,'' *Transl. Psychiatry*, vol. 5, no. 6, p. e574, 2015.

[7] J. Xi, A. Li, and M. Wang, ''A novel unsupervised learning model for detecting driver genes from pan-cancer data through matrix tri-factorization framework with pairwise similarities constraints,'' *Neurocomputing*, vol. 296, pp. 64–73, Jun. 2018.

[8] Y. Li, X. Yuan, J. Zhang, L. Yang, J. Bai, and S. Jiang, "SM-RCNV: A statistical method to detect recurrent copy number variations in sequenced samples," *Genes Genomics*, vol. 41, no. 5, pp. 529–536, May 2019.
[9] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the solexa/illumina FASTQ variants," *Nucleic Acids Res.*, vol. 38, no. 6, pp. 1767–1771, Apr. 2010.
[10] D. Y. Brandt, V. R. Aguiar, B. D. Bitarello, K. Nunes, J. Goudet, and D. Meyer, "Mapping bias overestimates reference allele frequencies at the HLA genes in the 1000 genomes project phase I data," *G3, Genes, Genomes, Genet.*, vol. 5, no. 5, pp. 931–941, 2015.
[11] M. Teng and R. A. Irizarry, "Accounting for GC-content bias reduces systematic errors and batch effects in ChIP-seq data," *Genome Res.*, vol. 27, no. 11, pp. 1930–1938, Nov. 2017.
[12] T. Blomquist, E. L. Crawford, and J. C. Willey, "Abstract 4150: Quantitative sequencing following PCR-driven library preparation with internal standard mixtures has improved analytical performance and lower cost," *Cancer Res.*, vol. 73, p. 4150, Apr. 2013.
[13] M. Zhao, Q. Wang, Q. Wang, P. Jia, and Z. Zhao, "Computational tools for copy number variation (CNV) detection using next-generation sequencing data: Features and perspectives," *BMC Bioinf.*, vol. 14, no. 11, p. S1, 2013.
[14] Y. Cun, T.-P. Yang, V. Achter, U. Lang, and M. Peifer, "Copy-number analysis and inference of subclonal populations in cancer genomes using Sclust," *Nature Protocols*, vol. 13, no. 6, pp. 1488–1501, Jun. 2018.
[15] A. Li, M. Wang, Z. Yu, and C. Guo, "ExomeHMM: A hidden Markov model for detecting copy number variation using whole-exome sequencing data," *Current Bioinf.*, vol. 12, no. 2, pp. 147–155, Mar. 2017.
[16] 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, Mar. 2013, Art. no. e59128.
[17] Y. Li, J. Zhang, and X. Yuan, "BagGMM: Calling copy number variation by bagging multiple Gaussian mixture models from tumor and matched normal next-generation sequencing data," *Digit. Signal Process.*, vol. 88, pp. 90–100, May 2019.
[18] A. Abyzov, A. E. Urban, M. Snyder, and M. Gerstein, "CNVnator: An approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing," *Genome Res.*, vol. 21, no. 6, pp. 974–984, Jun. 2011.
[19] S. D. Smith, J. K. Kawash, and A. Grigoriev, "GROM-RD: Resolving genomic biases to improve read depth detection of copy number variants," *PeerJ*, vol. 3, p. e836, Mar. 2015.
[20] R. Xi, S. Lee, Y. Xia, T.-M. Kim, and P. J. Park, "Copy number analysis of whole-genome data using BIC-seq2 and its application to detection of cancer susceptibility variants," *Nucleic Acids Res*, vol. 44, no. 13, pp. 6274–6286, Jul. 2016.
[21] X. Yuan, J. Bai, J. Zhang, L. Yang, J. Duan, Y. Li, and M. Gao, "CONDEL: Detecting copy number variation and genotyping deletion zygosity from single tumor samples using sequence data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published.
[22] X. Yuan, J. Zhang, L. Yang, J. Bai, and P. Fan, "Detection of significant copy number variations from multiple samples in next-generation sequencing data," *IEEE Trans. Nanobiosci.*, vol. 17, no. 1, pp. 12–20, Jan. 2018.
[23] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Anal.*, vol. 1, no. 1, pp. 121–143, Mar. 2006.
[24] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows–Wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, Jul. 2009.
[25] N. A. Johnson, R. Tibshirani, and W. Pei, *cghFLasso: Detecting Hot Spot on CGH Array Data With Fused Lasso Regression*, document R package version 0.2-1, 2009.
[26] 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, Jan. 2008.
[27] P. Wang, "Algorithms for calling gains and losses in array CGH data," in *Microarray Analysis of the Physical Genome: Methods and Protocols*. Totowa, NJ, USA: Humana Press, 2009.
[28] X. Yuan, J. Li, J. Bai, and J. Xi, "A local outlier factor-based detection of copy number variations from NGS data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published.
[29] J. D. Mcauliffe, D. M. Blei, and M. I. Jordan, "Nonparametric empirical Bayes for the Dirichlet process mixture model," *Statist. Comput.*, vol. 16, no. 1, pp. 5–14, Jan. 2006.
[30] O. Zobay, "Mean field inference for the Dirichlet process mixture model," *Electron. J. Statist.*, vol. 3, pp. 507–545, 2009.
[31] E. P. Xing, M. I. Jordan, and S. Russell, "A generalized mean field algorithm for variational inference in exponential families," in *Proc. 19th Conf. Uncertainty Artif. Intell.*, 2002, pp. 583–591.
[32] F. Pedregosa, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[33] X. Yuan, J. Zhang, and L. Yang, "IntSIM: An integrated simulator of next-generation sequencing data," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 2, pp. 441–451, Feb. 2017.
[34] D. Altshuler and E. Lander, "An integrated map of genetic variation from 1,092 human genomes," *Nature*, vol. 491, pp. 56–65, Oct. 2014.
[35] R. E. Mills *et al.*, "Mapping copy number variation by population-scale genome sequencing," *Nature*, vol. 470, pp. 59–65, Feb. 2011.
[36] S. Yoon, Z. Xuan, V. Makarov, K. Ye, and J. Sebat, "Sensitive and accurate detection of copy number variants using read depth of coverage," *Genome Res.*, vol. 19, no. 9, pp. 1586–1592, Sep. 2009.
[37] V. Boeva, T. Popova, K. Bleakley, P. Chiche, J. Cappo, G. Schleiermacher, I. Janoueix-Lerosey, O. Delattre, and E. Barillot, "Control-FREEC: A tool for assessing copy number and allelic content using next-generation sequencing data," *Bioinformatics*, vol. 28, no. 3, pp. 423–425, Feb. 2012.
[38] M. A. Van De Wiel, K. I. Kim, S. J. Vosse, W. N. Van Wieringen, S. M. Wilting, and B. Ylstra, "CGHcall: Calling aberrations for array CGH tumor profiles," *Bioinformatics*, vol. 23, no. 7, pp. 892–894, Apr. 2007.
[39] H. Yang and K. Wang, "Genomic variant annotation and prioritization with ANNOVAR and wANNOVAR," *Nature Protocols*, vol. 10, no. 10, pp. 1556–1566, Oct. 2015.
[40] C. Y. Lin, Y. Lo, and K. Q. Ye, "Genotype copy number variations using Gaussian mixture models: Theory and algorithms," *Stat. Appl. Genet. Mol. Biol.*, vol. 11, no. 5, pp. 1–26, 2012.
[41] N. Kumasaka, H. Fujisawa, N. Hosono, Y. Okada, A. Takahashi, Y. Nakamura, M. Kubo, and N. Kamatani, "PlatinumCNV: A Bayesian Gaussian mixture model for genotyping copy number polymorphisms using SNP array signal intensity data," *Genet. Epidemiol.*, vol. 35, no. 8, pp. 831–844, Dec. 2011.
[42] R. Pique-Regi, J. Monso-Varona, A. Ortega, R. C. Seeger, T. J. Triche, and S. Asgharzadeh, "Sparse representation and Bayesian detection of genome copy number alterations from microarray data," *Bioinformatics*, vol. 24, no. 3, pp. 309–318, Feb. 2008.
[43] P. Broët and S. Richardson, "Detection of gene copy number changes in CGH microarrays using a spatially correlated mixture model," *Bioinformatics*, vol. 22, no. 8, pp. 911–918, Apr. 2006.
[44] R.-S. Daruwala, A. Rudra, H. Ostrer, R. Lucito, M. Wigler, and B. Mishra, "A versatile statistical analysis algorithm to detect genome copy number variation," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 46, pp. 16292–16297, Nov. 2004.
[45] D. A. Engler, "A pseudolikelihood approach for simultaneous analysis of array comparative genomic hybridizations," *Biostatistics*, vol. 7, no. 3, pp. 399–421, Jan. 2006.
[46] G. Hodgson, J. H. Hager, S. Volik, S. Hariono, M. Wernick, D. Moore, D. G. Albertson, D. Pinkel, C. Collins, D. Hanahan, and J. W. Gray, "Genome scanning with array CGH delineates regional alterations in mouse islet carcinomas," *Nature Genet.*, vol. 29, no. 4, pp. 459–464, Dec. 2001.
[47] A. Valsesia, B. J. Stevenson, D. Waterworth, V. Mooser, P. Vollenweider, G. Waeber, C. Jongeneel, J. S. Beckmann, Z. Kutalik, and S. Bergmann, "Identification and validation of copy number variants using SNP genotyping arrays from a large clinical cohort," *BMC Genomics*, vol. 13, no. 1, p. 241, 2012.
[48] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin, "A statistical approach for array CGH data analysis," *BMC Bioinf.*, vol. 6, no. 1, p. 27, 2005.
[49] J. C. Marioni, N. P. Thorne, A. Valsesia, T. Fitzgerald, R. Redon, H. Fiegler, T. D. Andrews, B. E. Stranger, A. G. Lynch, E. T. Dermitzakis, N. P. Carter, S. Tavaré, and M. E. Hurles, "Breaking the waves: Improved detection of copy number variation from microarray-based comparative genomic hybridization," *Genome BiolGenome Biol.*, vol. 8, no. 10, p. R228, 2007.
[50] M. A. Sheha, M. S. Mabrouk, and M. Elhefnawi, "Detecting and analyzing copy number alternations in array-based CGH data," *Biomed. Eng. Appl. Basis Commun.*, vol. 28, no. 06, Dec. 2016, Art. no. 1650044.
[51] C. L. Myers, M. J. Dunham, S. Y. Kung, and O. G. Troyanskaya, "Accurate detection of aneuploidies in array CGH and gene expression microarray data," *Bioinformatics*, vol. 20, no. 18, pp. 3533–3543, Dec. 2004.
[52] S. M. Teo, Y. Pawitan, C. S. Ku, K. S. Chia, and A. Salim, "Statistical challenges associated with detecting copy number variations with next-generation sequencing," *Bioinformatics*, vol. 28, no. 21, pp. 2711–2718, Nov. 2012.

YAOYAO LI received the bachelor's degree in management from Harbin Medical University, in 2015. She is currently pursuing the integrated M.D. and Ph.D. degrees with the School of Computer Science and Technology, Xidian University, China. Her researches focused on detecting copy number variations and other bio-models from next-generation sequencing data by statistics and machine learning algorithms.

JUNYING ZHANG (Member, IEEE) received the Ph.D. degree in signal and information processing from Xidian University, Xi'an, China, in 1998.

She is currently a Professor with the School of Computer Science and Engineering, Xidian University. Her research interests include from intelligent information processing, machine learning and causation learning, and their applications to genome-wide association study, cancer related bioinformatics, medical image processing, and pattern recognition.

Dr. Zhang is a Senior Member of the China Electronics Society, the China Computer Society, Emergency Management Expert of Shaanxi Province, Evaluation Expert of Overseas Study in Shaanxi Province, National Study Fund Project Evaluation Expert, National Natural Science Foundation Project Evaluation Expert, Shaanxi Provincial Education Department Nature Scientific Fund Project Review Expert, Ningbo Science and Technology Plan Project Review Expert, and Beijing Natural Science Foundation Project Review Expert, *Chinese Science*, *Automation Journal*, *Electronic Journal*, *Neurocomputing*, *Digital Signal Processing*, *BMC Bioinformatics*, *ACM Computing Surveys*, and other publications expert.

XIGUO YUAN received the B.S. and M.S. degrees in computer applications from the Wuhan University of Science and Technology, in 2005 and 2008, respectively, and the Ph.D. degree in computer applications from Xidian University, in 2011.

He is currently an Associate Professor with the School of Computer Science and Technology, Xidian University. His research interest includes analyzing biomolecular data by computer technology to reveal the connotation of biomolecular data. Specifically: he is good at using comprehensive (design) machine learning algorithms, probability theory methods, and statistical test methods to detect or identify variant sites or fragments in the DNA genome to discover patterns with biological functions. His research group focuses on the analysis of single nucleotide polymorphism, copy number variation, methylation, and other data. Related articles have been published in the important international publications *BMC Genomics*, the Journal of Computational Biology, the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, the IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, and so on.

JUNPING LI received the B.S. degree with the Xi'an University of Technology, in 2017. She is currently pursuing the master's degree with Xidian University. She is also working on copy number variations. Her main research direction is computer bioinformatics.