frontiers in Cell and Developmental Biology

METHODS
published: 22 December 2021
doi: 10.3389/fcell.2021.796249

KNNCNV: A K-Nearest Neighbor Based Method for Detection of Copy Number Variations Using NGS Data

Kun Xie[1,2†], *Kang Liu*[1†], *Haque A K Alvi*[1], *Yuehui Chen*[3], *Shuzhen Wang*[1] *and Xiguo Yuan*[1,2*]

[1]School of Computer Science and Technology, Xidian University, Xi'an, China, [2]Hangzhou Institute of Technology, Xidian University, Hangzhou, China, [3]Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan, China

Copy number variation (CNV) is a well-known type of genomic mutation that is associated with the development of human cancer diseases. Detection of CNVs from the human genome is a crucial step for the pipeline of starting from mutation analysis to cancer disease diagnosis and treatment. Next-generation sequencing (NGS) data provides an unprecedented opportunity for CNVs detection at the base-level resolution, and currently, many methods have been developed for CNVs detection using NGS data. However, due to the intrinsic complexity of CNVs structures and NGS data itself, accurate detection of CNVs still faces many challenges. In this paper, we present an alternative method, called KNNCNV (K-Nearest Neighbor based CNV detection), for the detection of CNVs using NGS data. Compared to current methods, KNNCNV has several distinctive features: 1) it assigns an outlier score to each genome segment based solely on its first *k* nearest-neighbor distances, which is not only easy to extend to other data types but also improves the power of discovering CNVs, especially the local CNVs that are likely to be masked by their surrounding regions; 2) it employs the variational Bayesian Gaussian mixture model (VBGMM) to transform these scores into a series of binary labels without a user-defined threshold. To evaluate the performance of KNNCNV, we conduct both simulation and real sequencing data experiments and make comparisons with peer methods. The experimental results show that KNNCNV could derive better performance than others in terms of F1-score.

Keywords: k-nearest neighbor, copy number variation, next-generation sequencing, variational Bayesian Gaussian mixture model, tumor genome

OPEN ACCESS

Edited by:
Shibiao Wan,
St. Jude Children's Research Hospital, United States

Reviewed by:
Ruifeng Hu,
Harvard Medical School, United States
Xiaoli Lin,
Wuhan University of Science and Technology, China

***Correspondence:**
Xiguo Yuan
xiguoyuan@mail.xidian.edu.cn

[†]These authors have contributed equally to this work and share first authorship

Specialty section:
This article was submitted to Molecular and Cellular Oncology, a section of the journal Frontiers in Cell and Developmental Biology

Received: *16 October 2021*
Accepted: *23 November 2021*
Published: *22 December 2021*

Citation:
Xie K, Liu K, Alvi HAK, Chen Y, Wang S and Yuan X (2021) KNNCNV: A K-Nearest Neighbor Based Method for Detection of Copy Number Variations Using NGS Data. Front. Cell Dev. Biol. 9:796249. doi: 10.3389/fcell.2021.796249

INTRODUCTION

Copy number variations (CNVs) of DNA sequences are accountable for functional phenotypic diversity in many species and play an important role in human genomic variation and cancer initiation (Schrider et al., 2013; Unckless et al., 2016). CNV is a commonly reported variation from the diploid state caused by amplification or deletion of genomic regions ranging from one kilo-base to several mega-bases (Redon et al., 2006; Li et al., 2020). In cancer, tumor-derived CNVs are one of the most significant genomic anomalies, alongside somatic mutations and structural variations (SVs). Tumor suppressor gene inactivation or oncogene activation are frequently ascribed to copy number loss or gain, respectively (Yuan et al., 2012). Specifically, Gains may contain oncogenes, and losses may include tumor-suppressor genes (Xie et al., 2021). Consequently, detecting cancer-

associated copy number occurrences is crucial in identifying patient subtypes, as well as providing insights into prognosis and prospective treatment options. Fortunately, next-generation sequencing (NGS) technology has accelerated the development of the detection of CNVs (Teo et al., 2012), which provides greater scope to discover novel CNVs and has a greater resolution to forecast both breakpoints and shorter CNVs. However, owing to the intrinsic complexity of CNVs structure and the huge scale of NGS data, accurate detection of CNVs remains challenging.

Numerous bioinformatics tools for the detection of CNVs from NGS data have been developed, and these algorithms can be classified into four main categories: read-pair (RP), split-read (SR), read-depth (RD), and *de novo* assembly (DA). The above four approaches have their strengths, shortcomings, and scope of implementation, and their details can be referred to (Zhao et al., 2013; K. Ye et al., 2016). Among these approaches, the RD-based strategy is most frequently used to detect CNVs, since the strategy is theoretically more likely to detect CNVs with different sizes (Zare et al., 2017). A great number of methods under the RD-based strategy have been developed based on the characteristics of NGS data. FREEC (Boeva et al., 2010; Boeva et al., 2012) considers the RD profile from a global context and exploits the variance in RD values to discover CNVs. When normal matched samples are not present, FREEC can use GC-content to normalize the RD values and accurately identify CNVs from tumor samples. ReadDepth (Miller et al., 2011) and iCopyDAV (Dharanipragada et al., 2018) are similar approaches. The *m*-HMM (Wang et al., 2014) method considers the entire RD profile as a Markov model and forecasts copy number states. CNVnator (Abyzov et al., 2011) leverages the multiple-bandwidth partitioning technique and mean-shift approach to detect broad CNVs. GROM-RD (Smith et al., 2015) can analyze multiple biases such as GC-bias and repeat bias and use sliding windows with variable size to improve breakpoint resolution.

The above methods take different perspectives on the features of CNVs, and such methods have the advantage of detecting broad CNVs. However, the focal (*i.e.*, local) CNVs may be ignored. To address the above limitation, the CNV-LOF (Yuan et al., 2021a) takes a local view on the RD values, so the method avoids some local CNVs being masked by the surrounding regions. To consider the correlation of copy numbers in adjacent positions, CNV_IFTV (Yuan et al., 2021b) calculates outlier scores based upon the isolation forest algorithm and leverages the total variation model to smooth these scores, and similar methods include CNV-RF (Onsongo et al., 2016) and CONDEL (Yuan et al., 2020). In addition, IhbyCNV (Xie et al., 2021) takes a comprehensive viewpoint on the characteristics of CNVs, that is, the method treats CNVs detection as outlier events from five perspectives on the RD profile to be addressed. Although these methods exhibit their own characteristics and advantages in different scenarios, it is still necessary to design a simple and effective method to deal with the intrinsic complexity of CNVs structure and NGS data itself.

With careful consideration of the challenges above, in this paper, we propose an alternative method used for whole genome sequencing, coined KNNCNV (K-Nearest Neighbor based CNV detection), which can identify CNVs using NGS data. The core module of the KNNCNV is that the outlier scores for all genome segments are calculated solely by their *k*th nearest-neighbor distances, and then these scores are converted into a succession of binary labels through the VBGMM (Corduneanu and Bishop, 2001; Tzikas et al., 2008). In this work, we make two key contributions as follows.

1) The outlier score for any genome segment can be defined based solely on its first k nearest-neighbor distances. More specifically, the average value of these distances is regarded as the outlier score of the genome segment, which is not only easy to extend to other data types but also boosts the power of detection CNVs, especially the local CNVs that are likely to be masked by their surrounding regions.
2) This paper leverages the VBGMM to convert the outlier scores for all genome segments into a series of binary labels that can indicate which genome segments are CNVs. The VBGMM can approximate the posterior distribution of these scores, so it can also be considered as a soft clustering method, and these binary labels are obtained without a pre-specified threshold.

MATERIALS AND METHODS

Overview of KNNCNV

The workflow of the KNNCNV method is shown in **Figure 1**, which consists mainly of three steps. In the first step, a sequenced sample and a reference genome are taken as the input data. The second step is preprocessing, including the read alignment, read count (RC) preprocessing, and read depth (RD) profile generation and segmentation. In the third step, the outlier score for each genome segment is calculated by the *k*-nearest neighbor (KNN) (Ramaswamy et al., 2000; Angiulli and Pizzuti, 2002), and these scores are converted into binary labels via the VBGMM (Corduneanu and Bishop, 2001; Tzikas et al., 2008). In addition, the KNNCNV is implemented in Python and R language, which is freely available at https://github.com/BDanalysis/KNNCNV.

Preprocessing

After obtaining a sequenced sample (*i.e.*, a Fastq file) and a reference genome (*i.e.*, a Fasta file), the sequenced sample is aligned to the reference genome with the BWA algorithm (Li and Durbin, 2010). Then the alignment result is extracted by the SAMtools software (Li et al., 2009), and the RC profile, which is SAM or BAM format, is obtained. The preprocessing of the RC profile includes the preprocessing of the reference genome, generating the genome bins, and correcting the GC-bias. The reference genome has some problems with missing positions and 'N' positions. In this paper, the missing positions are filled with zeros, and the 'N' positions are removed. As for generating the genome bins, the RC profile is partitioned into continuous and disjoint genome bins with the same length L_b (*i.e.*, the bin size L_b equals 1,000 bp). The average RC value for each genome bin is regarded as its RD value, and simultaneously the fraction of GC-content can be obtained. In terms of the GC-bias, it is corrected by the prior work (Yuan et al., 2021a). Owing to the correlations

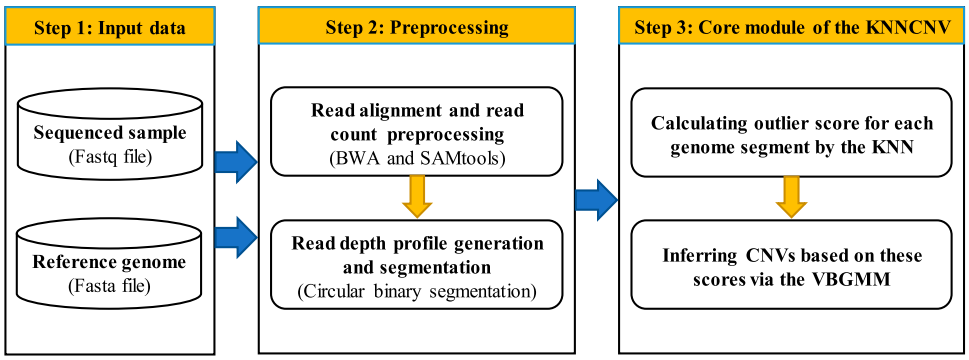

FIGURE 1 | Workflow of the KNNCNV method.

between adjacent bins (Yuan et al., 2018; Yuan et al., 2021a), segment-based units have some advantages over bin-based ones in both computational cost and reliable results. Therefore, the whole genome is divided into continuous and non-overlapping regions with the same length L_r (*i.e.*, the region size L_r is equal to 50,000 bp), and then each region is segmented by the circular binary segmentation (CBS) algorithm (Venkatraman and Olshen, 2007). Thus, a family of genome segments that are different in size are generated in each region, and the number of genome segments relies on the fluctuation of the RD values. Let the number of all genome segments generated by all regions be N, hence all genome segments are denoted as $R = [r_1, r_2, \cdots, r_N]^T \in \mathcal{R}^{N\times 1}$, where r_i represents average RD values for all genome bins in the ith genome segment, and $[\cdot]^T$ is a transposed matrix.

Calculating Outlier Score for Each Genome Segment by the KNN

After the above preprocessing, the RD values for all genome segments (*i.e.*, R) can be obtained. Next, to estimate the degree of abnormality (*i.e.*, outlier score) of each genome segment, we resort to the k-nearest neighbor (KNN) method (Ramaswamy et al., 2000; Angiulli and Pizzuti, 2002), which is a distance-based method and naturally assumes that the k-nearest neighbor distance of outliers (*i.e.*, CNVs) is much larger than that of normal points. To simplify the representation, we use k to denote the number of nearest neighbors for any object. Before the introduction of calculating the outlier score by the KNN, we first describe two definitions. Note that Definitions 1 refers to prior work (Breunig et al., 2000; Yuan et al., 2021a), and similarly, Definition 2 refers to (Ramaswamy et al., 2000; Angiulli and Pizzuti, 2002; Aggarwal, 2017).

Definition 1 (*k-distance* and k-nearest neighborhood for any object r) Given the RD values for all genome segments $R = [r_1, r_2, \cdots, r_N]^T \in \mathcal{R}^{N\times 1}$ and a positive integer k, the *k-distance* for any $r \in R$ to the remaining ones can be defined as $k - distance(r) = distance(r, e)$, where $e \in R$, and *distance* (r, e) denotes the Euclidean distance between objects r and e. Moreover, among all objects in R, e is an object that is the kth nearest neighbor to r. The k-nearest neighborhood for any object $r \in R$ can be formulated as $N_k(r) = \{t | distance(r,t) \le k - distance(r), t \in R, t \ne r\}$. Thus, the first k nearest-neighbor distances between the object r and the rest can be expressed as $\{distance(r,t) | t \in N_k(r)\}$.

Definition 2 (outlier score for any object r) Knowing the RD values for all genome segments $R = [r_1, r_2, \cdots, r_N]^T \in \mathcal{R}^{N\times 1}$ and a positive integer k, the outlier score for any object $r \in R$ can be defined as $^1/_{|N_k(r)|} \sum_{t\in N_k(r)} distance(r,t)$, where $|N_k(r)|$ denotes the cardinality of the set $N_k(r)$, and $0 \le |N_k(r)| \le k$. Furthermore, see Definitions 1 for more information on $distance(r,t)$ and $N_k(r)$. Note that the above scheme for calculating outlier scores is referred to as the average outlier score scheme.

From the above definitions, it is obvious that Euclidean distances between all pairwise objects must be calculated to obtain the k-nearest neighborhood for all objects in R. The computational overhead $O(N^2)$ increases significantly with the increase of N, where N denotes the number of genome segments. To partially circumvent this problem, a space-partitioning tree data structure, k-dimensional tree (KDTree) (Ramasubramanian and Paliwal, 1992), is adopted to search the k-nearest neighborhood for any objects in R. On utilizing the KDTree, its computational cost is $O(N \log N)$. Next, this paper introduces how to estimate the outlier score for each genome segment via the KNN method. Given the RD values for all genome segments $R = [r_1, r_2, \cdots, r_N]^T \in \mathcal{R}^{N\times 1}$ and a positive integer k, the outlier score s_r for any object $r \in R$ can be obtained by Definition 2. More exactly, the score s_r is defined as the average value among its first k nearest-neighbor distances. Additionally, there are two simple variations of the scoring mechanism corresponding to the largest outlier score scheme and the median outlier score scheme (Aggarwal, 2017). Precisely, for any object $r \in R$, the two simple variations treat the largest and median value among the first k nearest-neighbor distances as its outlier score, respectively. Nevertheless, the two simple variations neglect or hardly consider the information of other nearest neighbors, which may yield unstable performance when the k value is not reasonable. Therefore, this paper adopts a more robust average outlier score scheme to estimate the outlier score for any object in R. It is noteworthy that, among these three outlier score schemes, a 'correct' k value should be specified in advance (Aggarwal,

2017). However, in the detection of CNVs, it is difficult to search for a 'correct' k value due to the lack of ground truth. To partially bypass this issue, one specifies a range of values of k and then leverages random strategy to determine a final k value. More specifically, the integer k is randomly selected in the range of $[0.2N, 0.35N]$, and k is rounded down. Accordingly, outlier scores for all genome segments (*i.e.*, $S = [s_1, s_2, \cdots, s_N]^T \in \mathcal{R}^{N\times 1}$) are obtained through the average outlier score scheme, as shown in Definition 2.

Inferring CNVs Based on the Scores via the VBGMM

Although outlier scores for all genome segments are obtained, these scores cannot be directly used to determine which genome segments are CNVs. A simple solution is that after the outlier scores are ranked in descending order, the solution treats the genome segments corresponding to the first n scores as CNVs, and the solution is called a simple threshold scheme in this paper. Although converting these scores into binary labels is feasible by the simple threshold, it is difficult to define a reasonable n value owing to the absence of ground truth. To address this issue, the variational Bayesian Gaussian mixture model (VBGMM) (Corduneanu and Bishop, 2001; Tzikas et al., 2008) is adopted to convert the outlier scores into a series of binary labels. The VBGMM can approximate the posterior distribution of these scores, so the method can also be considered as a soft clustering method, and these labels are obtained without a user-defined threshold.

This paper first introduces the Gaussian mixture model (GMM) (Bishop, 2007), which assumes that the distribution of S can be represented by the linear superposition of M Gaussian distributions (*i.e.*, component). Let α_m and $\mathcal{N}(S|\mu_m, \sigma_m^2)$ be the mixing coefficient and the probability density of the mth component, respectively, where μ_m and σ_m^2 represent the mean and variance. Note that we use μ to denote the set $\{\mu_1, \mu_2, \cdots, \mu_M\}$, and similarly for $\sigma^2 = \{\sigma_1^2, \sigma_2^2, \cdots, \sigma_M^2\}$ and $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_M\}$. Thus, the mixture distribution of the ith outlier score s_i can be formulated as **Eq. 1**.

$$p(s_i|\alpha, \mu, \sigma^2) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(s_i|\mu_m, \sigma_m^2). \tag{1}$$

The left and right sides of **Eq. 1** integrate s_i at the same time. $0 \le \alpha_m \le 1$ and $\sum_{m=1}^{M} \alpha_m = 1$ are obtained due to $p(s_i|\alpha, \mu, \sigma^2) \ge 0$ and $\mathcal{N}(s_i|\mu_m, \sigma_m^2) \ge 0$. To calculate the parameters of μ, σ^2, and α, binary latent variables are introduced, which can be defined as $Z = \{z_{im}|1 \le i \le N, 1 \le m \le M, z_{im} \in \{0, 1\}\}$, where $\sum_{m=1}^{M} z_{im} = 1$, and $z_{im} = 1$ means that s_i is sampled from the mth component. Therefore, the marginal distribution of Z can be formulated as **Eq. 2**.

$$p(Z|\alpha) = \prod_{i=1}^{N} \prod_{m=1}^{M} \alpha_m^{z_{im}}. \tag{2}$$

To simplify some representations, let θ be $\{Z, \mu, \sigma^2\}$. Given the parameters θ, the conditional probability of S can be formulated as **Eq. 3**.

$$p(S|\theta) = p(S|Z, \mu, \sigma^2) = \prod_{i=1}^{N} \prod_{m=1}^{M} \mathcal{N}(s_i|\mu_m, \sigma_m^2)^{z_{im}}. \tag{3}$$

According to Bayes' theorem, after the s_i is observed, the posterior distribution $p(z_{im} = 1|s_i)$ from the mth component can be formulated as **Eq. 4**.

$$p(z_{im} = 1|s_i) = \frac{p(s_i|z_{im} = 1)p(z_{im} = 1)}{p(s_i)} = \frac{\alpha_m \mathcal{N}(s_i|\mu_m, \sigma_m^2)}{\sum_{m=1}^{M} \alpha_m \mathcal{N}(s_i|\mu_m, \sigma_m^2)}, \tag{4}$$

where $p(z_{im} = 1|s_i)$ is also referred to as the responsibility $\gamma(z_{im})$ of the mth component to s_i, that is, $\gamma(z_{im}) = p(z_{im} = 1|s_i)$. Given the mixing coefficients and the parameters of components, the likelihood function can be formulated as **Eq. 5**.

$$p(S|\alpha, \mu, \sigma^2) = \prod_{i=1}^{N} \left[\sum_{m=1}^{M} \alpha_m \mathcal{N}(s_i|\mu_m, \sigma_m^2) \right]. \tag{5}$$

On obtaining the likelihood function, the parameters of the GMM can be estimated by using the maximum likelihood framework of expectation maximization (EM) algorithm (Zandi et al., 2013). However, the likelihood function may lead to singularities, that is, one or more component density collapses onto specific data (Bishop, 2007). Therefore, this paper utilizes the VBGMM to infer CNVs based on the outlier scores for all genome segments. Precisely, the VBGMM uses a simpler distribution $q(\theta)$ to estimate the true posterior distribution $p(\theta|S, \alpha)$ and then maximizes the evidence lower bound (ELOB) on $\ln p(S|\alpha)$.

Next, the details of the VBGMM are described in the following. By Bayes' theorem, we have:

$$\begin{aligned} \ln p(S|\alpha) &= \ln p(S, \theta|\alpha) - \ln p(\theta|S, \alpha) \\ &= [\ln p(S, \theta|\alpha) - \ln q(\theta)] - [\ln p(\theta|S, \alpha) - \ln q(\theta)] \\ &= \ln \frac{p(S, \theta|\alpha)}{q(\theta)} - \ln \frac{p(\theta|S, \alpha)}{q(\theta)}, \end{aligned} \tag{6}$$

the left and right of **Eq. 6** calculate the expectation to $q(\theta)$ at the same time, thus **Eq. 7** is obtained.

$$\begin{aligned} \ln p(S|\alpha) &= \underbrace{\int q(\theta) \ln \frac{p(S, \theta|\alpha)}{q(\theta)} d\theta}_{\mathcal{L}(q)} \underbrace{- \int q(\theta) \ln \frac{p(\theta|S, \alpha)}{q(\theta)} d\theta}_{\mathbb{KL}(q|p)} \\ &= \mathcal{L}(q) + \mathbb{KL}(q|p), \end{aligned} \tag{7}$$

where $\mathcal{L}(q)$ denotes the ELOB on $\ln p(S|\alpha)$, and $\mathbb{KL}(q|p)$ denotes the Kullback-Leibler divergence between $q(\theta)$ and $p(\theta|S, \alpha)$. Since $\mathbb{KL}(q|p) \ge 0$, the ELOB $\mathcal{L}(q)$ is less than or equal to $\ln p(S|\alpha)$. The goal of the VBGMM is to select a reasonable $q(\theta)$ to approximate the true posterior distribution $p(\theta|S, \alpha)$, that is, minimization $\mathbb{KL}(q|p)$. Of course, the ideal state is $\mathbb{KL}(q|p) = 0$, in other words, $q(\theta) = p(\theta|S, \alpha)$ and $\ln p(S|\alpha) = \mathcal{L}(q)$. Additionally, the $\ln p(S|\alpha)$ is fixed relative to the selection of $q(\theta)$, so minimizing the $\mathbb{KL}(q|p)$ is equivalent to

maximizing the ELOB $\mathcal{L}(q)$. To simplify this problem, it is assumed that the $q(\theta)$ follows the mean field theory (Bishop, 2007). Accordingly, the $q(\theta)$ can be formulated as $q(\theta) = \prod_j q_j(\theta_j) = q_Z(Z)q_\mu(\mu)q_{\sigma^2}(\sigma^2)$. By maximizing the ELOB on $\ln p(S|\alpha)$, the solution of variational posterior $q_j(\theta_j)$ can be formulated as **Eq. 8**.

$$q_j(\theta_j) = \frac{\exp \mathbb{E}_{l\neq j}[\ln p(S,\theta|\alpha)]}{\int \exp \mathbb{E}_{l\neq j}[\ln p(S,\theta|\alpha)]d\theta_j}, \tag{8}$$

where $\mathbb{E}_{l\neq j}[\cdot]$ represents the expectations with respect to $q_l(\theta_l)$ for all $l \neq j$. Refer to prior works (Corduneanu and Bishop, 2001; Tzikas et al., 2008) for the specific derivation process. In addition, according to previous work (Corduneanu and Bishop, 2001), the joint distribution $p(S,\theta|\alpha)$ can be formulated as **Eq. 9**.

$$p(S,\theta|\alpha) = p(S,Z,\mu,\sigma^2|\alpha) = p(S|Z,\mu,\sigma^2)p(Z|\alpha)p(\mu)p(\sigma^2), \tag{9}$$

where $p(\mu)$ and $p(\sigma^2)$ follow the Gaussian distribution and the Wishart distribution, respectively. Their specific forms refer to (Corduneanu and Bishop, 2001). Thus, considering **Eqs. 2**, **3**, **8**, **9** jointly, the iterative formula of the variational posterior $q_j(\theta_j)$ can be formulated as **Eq. 10**.

$$\begin{aligned} q_Z(Z) &= \prod_{i=1}^{N}\prod_{m=1}^{M} h_{im}^{z_{im}} \\ q_\mu(\mu) &= \prod_{m=1}^{M} \mathcal{N}\left(\mu_m \middle| b_\mu^m, \sigma_m^2(\mu)\right) \\ q_{\sigma^2}(\sigma^2) &= \prod_{m=1}^{M} \mathcal{W}\left(\sigma_m^2 \middle| v_{\sigma^2}^{(m)}, V_{\sigma^2}^{(m)}\right), \end{aligned} \tag{10}$$

where $\mathcal{W}$ represents the Wishart distributions. For more information on h_{im}, b_μ^m, $\sigma_m^2(\mu)$, $v_{\sigma^2}^{(m)}$, and $V_{\sigma^2}^{(m)}$, please refer to (Corduneanu and Bishop, 2001). After obtaining these variational posteriors, the ELOB $\mathcal{L}(q)$ is also obtained. Next, let the partial derivative of ELOB $\mathcal{L}(q)$ with respect to α be zero, so the iterative formula of α can be formulated as **Eq. 11**.

$$\alpha_m = \frac{1}{N}\sum_{i=1}^{N} h_{im}, \tag{11}$$

note that details on h_{im} can be obtained by referring to previous work (Corduneanu and Bishop, 2001). The maximum likelihood framework of the EM algorithm is summarized as the following two steps. In the expectation step, the solutions of variational posterior $q_j(\theta_j)$ are calculated by **Eq. 10**. In the maximization step, the iterative formula of α is obtained by maximizing the ELOB $L(q)$ with respect to α. Repeat the above expectation and maximization steps until the stop condition is met (*e.g.*, the maximum number of iterations is reached).

On the basis of the above introduction, one can find a $q(\theta)$ to approximate the true posterior distribution $p(\theta|S,\alpha)$. Thus, the outlier scores S can be composed of M clusters, and each cluster corresponds to a component. The cluster index λ_i for any score s_i can be defined as **Eq. 12**.

$$\lambda_i = \arg\max\{\gamma_{im} | 1 \le m \le M\}, \tag{12}$$

where $\arg\max\{\cdot\}$ denotes the index corresponding to the maximum value in the set $\{\cdot\}$. Consequently, the VBGMM can be regarded as a soft clustering method (Tzikas et al., 2008). Since the outlier scores indicate the anomaly degree of each genome segment, each segment is either a CNV or a normal one. Thus, let M equal two, that is, it is assumed that the distribution of S can be represented by the linear superposition of two Gaussian distributions. Note that the VBGMM is implemented by scikit-learn (Pedregosa et al., 2011), and the detailed architecture of the VBGMM is described in **Algorithm 1**.

Algorithm 1: Converting outlier scores into binary labels by the VBGMM

input:	the outlier scores S for all genome segments; the number of components M=2
output:	binary labels indicate whether genome segments are CNVs
	// estimate the parameters θ and α of the VBGMM using the EM algorithm
1:	initialize the components parameters and mixing coefficients $\{(\mu_m,\sigma_m^2,\alpha_m)\|1\le m\le 2\}$
2:	**repeat**
3:	**for** $i = 1,2,\cdots,N$ **do**
4:	calculate the responsibility for outlier score s_i by Eq. (4)
5:	**end for**
	// in expectation step
6:	update the solutions of the variational posterior $q_j(\theta_j)$ by Eq. (10)
	// in maximization step
7:	update the mixing coefficients α by Eq. (11)
8:	**until** the stop condition is satisfied
	// determine the cluster index for all outlier scores
9:	$C_m = \varnothing, 1 \le m \le 2$
10:	**for** $i = 1,2,\cdots,N$ **do**
11:	determine the cluster index λ_i for s_i by Eq. (12)
12:	put the score s_i into the corresponding cluster C_{λ_i}, that is, $C_{\lambda_i} = C_{\lambda_i} \cup \{s_i\}$
13:	**end for**
	// determine the cluster in which CNVs are located
14:	$m_1, m_2 \leftarrow$ the mean of the C_1 and C_2
15:	**if** $m_1 \ge m_2$
16:	**then** label $\leftarrow$ genome segments corresponding to the scores in cluster C_1 are regarded as CNVs, otherwise, ones are normal
17:	**else** label $\leftarrow$ genome segments corresponding to the scores in cluster C_2 are regarded as CNVs, otherwise, ones are normal
18:	**end if**
19:	**return** label

RESULTS

To evaluate the performance of KNNCNV, we conduct experiments on simulated and real datasets. As for the experiments on the simulated datasets and real blood datasets, we first make comparisons between the proposed method and peer methods and then discuss the influence of the hyperparameter k on the result of KNNCNV. Finally, we explore the effectiveness of each part of the KNNCNV. In addition, the performance of the above methods is quantified by precision, sensitivity, and F1-score, where $precision = TP/PP$, and $sensitivity = TP/P$, and F1-score is the harmonic mean between the precision and sensitivity. Here TP denotes the number of duplicate genomic positions between the declared CNVs and confirmed CNVs, and PP represents the total number of genomic positions in the declared CNVs, and similarly, P is the total number of positions in the confirmed CNVs. In terms of the real cancer datasets, the comparison of our method with peer methods is made in terms of the overlapping density score (ODS)

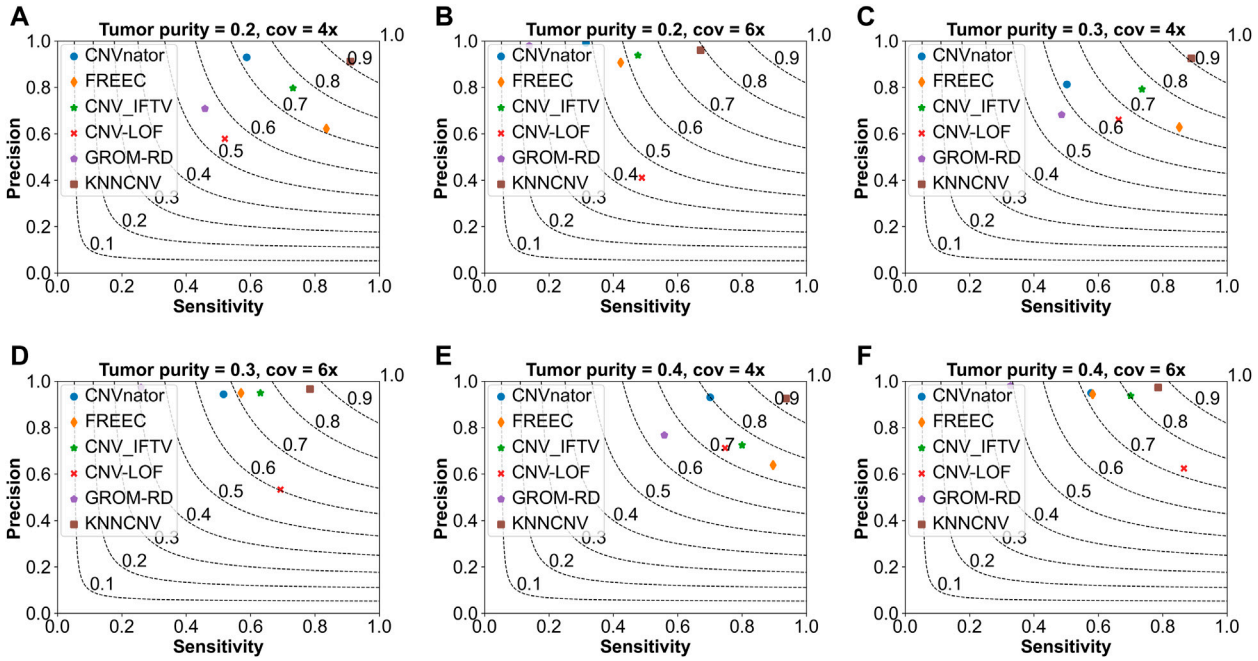

FIGURE 2 | Comparison of the KNNCNV with five peer methods on the simulated datasets in terms of precision, sensitivity, and F1-score. The F1-score is shown in black dashed lines ranging from 0.1 to 0.9 with an increment of 0.1. **(A–F)** They show the performance of the above six methods on different simulated configurations, respectively.

(Yuan et al., 2020). To fairly compare our method with existing ones, their default parameters are used. Note that the performance of the KNNCNV on a third-generation sequencing sample is shown in **Supplementary Table 1**.

Simulation Studies

The simulated datasets were generated by the IntSIM (Yuan et al., 2017), and two key parameters (*i.e.*, tumor purity and coverage depth) should be specified. In each simulated configuration, the tumor purity ranged from 0.2 to 0.4 in increments of 0.1, and the coverage depth belonged to the set {4x, 6x}. In addition, chromosome 21 of hg19 was selected as the reference genome. To simplify the representations, we use (*p*, *cov*) to represent the tumor purity and coverage depth, respectively. Note that each simulated configuration was repeated fifty times to reduce the randomness of the experiments, and their average performance was reported.

To show the effectiveness of the KNNCNV, the comparisons of the KNNCNV with five existing methods are shown in **Figure 2**, and these existing methods include CNVnator (Abyzov et al., 2011), FREEC (Boeva et al., 2010; Boeva et al., 2012), CNV_IFTV (Yuan et al., 2021b), CNV-LOF (Yuan et al., 2021a), and GROM-RD (Smith et al., 2015). One can observe that the sensitivity of our method outperforms other methods except for **Figure 2F**. Furthermore, although the KNNCNV is not very prominent in precision and sensitivity, it achieves a surprised F1-score compared to these existing methods. More precisely, in terms of F1-score, the KNNCNV is about 14.33%, 14.28%, 13.84%, 9.07%, 12.64%, and 5.28% higher than the highest existing methods, respectively.

The KNNCNV involves some hyperparameters including bin size, region size, and the number of nearest neighbors (*i.e.*, *k*). Among them, only the *k* value has been carefully researched, so **Figures 3A–C** show the variation of different outlier score schemes generated by the KNN with the number of nearest neighbors, respectively. The results indicate that the median outlier score scheme and the largest scheme may yield unstable performance when the *k* value is not reasonable, and the average outlier score scheme is relatively insensitive to the *k* value when it reaches a certain value. Additionally, we study the effectiveness of the VBGMM (Corduneanu and Bishop, 2001; Tzikas et al., 2008). The comparison of the VBGMM with other threshold selection strategies is shown in **Figure 3D**, and these threshold selection strategies consist of some simple threshold schemes, boxplot (Sim et al., 2005), and GMM (Bishop, 2007; Aggarwal, 2017). Note that the boxplot scheme treats the upper fence of the boxplot as CNVs, and its whisker is 0.75. It can be seen that the GMM and the VBGMM outperform other strategies for *cov* = 4x. Although the boxplot scheme ranks first for *cov* = 6x, the one is less stable than the GMM and the VBGMM. Furthermore, a simple threshold scheme is also desirable when a suitable threshold is found, but it is difficult to find such a threshold in real-world applications. To further discuss the complexity of the proposed method, the computational cost and performance of five methods vary with the number of BAM files, as shown in **Figures 4A,B**, and these five methods include CNVnator, FREEC, CNV_IFTV, CNV-LOF, and KNNCNV. The results show that the KNNCNV not only has promising performance, but also its computing overhead is acceptable.

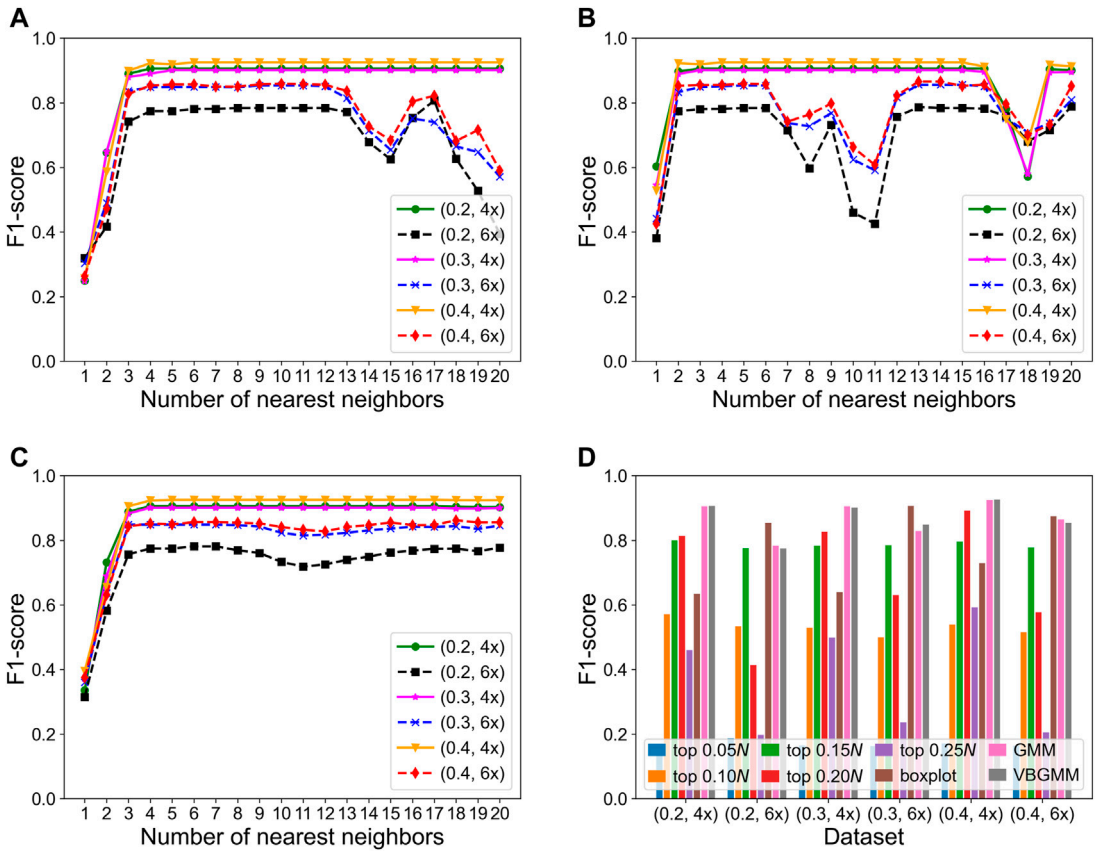

FIGURE 3 | F1-score on the simulated datasets. **(A-C)** Variation of different outlier score schemes generated by the KNN with the number of nearest neighbors, where 1–20 denote 0.05, *0.10*, 0.15*N*, . . . , 0.95*N*, and 1.00*N*, respectively. The different outlier score schemes are the median outlier score scheme **(A)**, the largest outlier score scheme **(B)**, and the average outlier score scheme **(C)**. **(D)** Result with different threshold selection strategies under the framework of our method, where the 'top *n*' denotes a simple threshold, and *n* belongs to the set {0.05, *0.10, 0.15, 0.20*, 0.25*N*}. Specifically, after the outlier scores are ranked in descending order, the scheme treats the genome segments corresponding to the first *n* scores as CNVs. Note that 1–20 and *n* are rounded down.

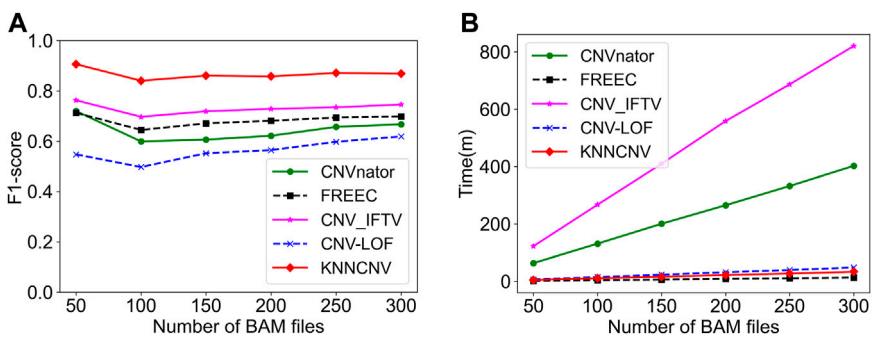

FIGURE 4 | Comparison of the KNNCNV with four peer methods on the simulated datasets in terms of F1-score and computational cost. **(A)** Performance of the five methods with different the number of BAM files. **(B)** Variation of running time of the five methods with the number of BAM files.

Application to Real Datasets

Analysis of Blood Samples from the 1,000 Genomes Project

The real blood samples consist of NA12878, NA12891, NA12892, NA19238, NA19239, and NA19240, where the first three samples come from the CEU trio of European ancestry and the remaining three from the YRI trio of Yoruba Nigerian ethnicity. Note that each trio includes two parents and one daughter, and the above six samples can be obtained from the 1,000 Genomes Project (http://www.1000genomes.org). Each real sequencing sample was

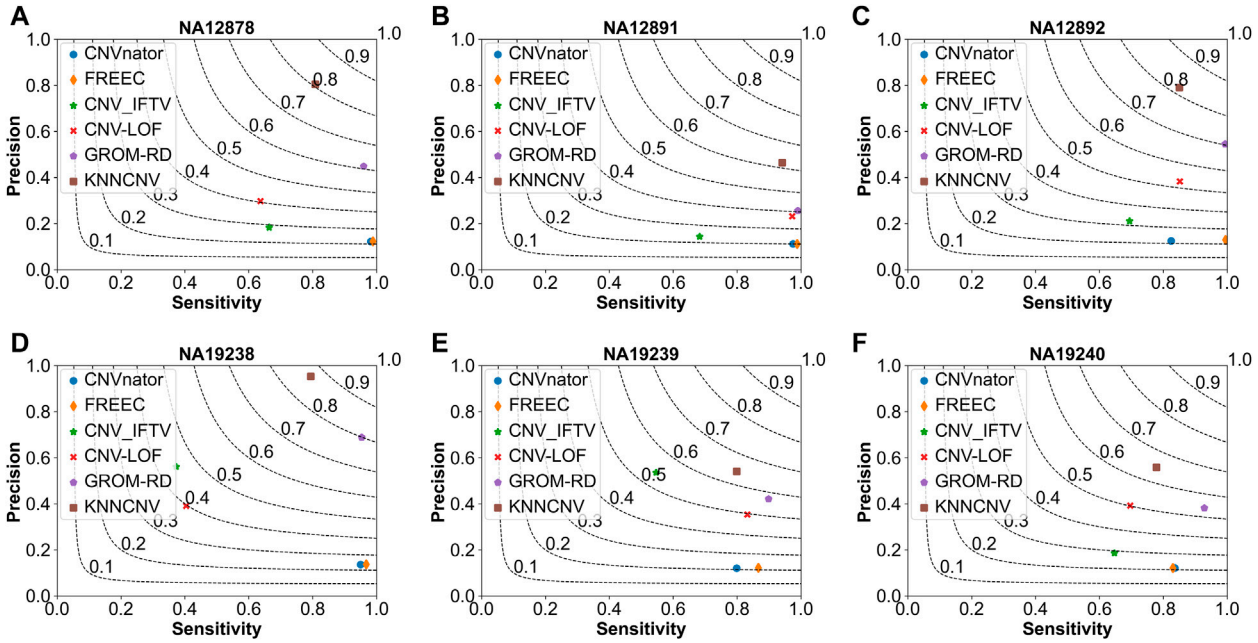

FIGURE 5 | Comparison of the KNNCNV with five peer methods on the real datasets in terms of precision, sensitivity, and F1-score. The F1-score is shown in black dashed lines ranging from 0.1 to 0.9 with an increment of 0.1. **(A-F)** They show the performance of the above six methods on six real datasets, respectively.

repeated twenty times on the 21st chromosome, and their average performance was reported. The confirmed CNVs of these samples can be obtained from the database of genomic variants (http://dgv.tcag.ca/dgv/app/home), which can help us calculate some performance metrics, such as precision, sensitivity, and F1-score.

As shown in **Figure 5**, we make comparisons between the KNNCNV and five peer methods on the six real datasets. It can be observed that our method achieves the best F1-score, outperforming the highest existing method by 19.57%, 21.45%, 11.50%, 6.61%, 7.17%, and 11.00%, respectively. Furthermore, the precision of our method also significantly outperforms these peer methods. Additionally, the precision of many methods is unsatisfactory compared to **Figure 2** since there is a certain deviation between the simulated and real datasets. Specifically, due to the complexity of realistic cancer genomes, the simulated datasets cannot accurately reflect the variant distributions and correlations of the real datasets and do not take into account insertion and deletion errors.

To verify the effectiveness of each part of the KNNCNV, the hyperparameter k, other threshold selection strategies, different outlier score schemes generated by KNN, and other detector schemes are discussed. The variation of the KNNCNV with the number of nearest neighbors is shown in **Figure 6A**. The result illustrates that the performance of the KNNCNV is less sensitive to the k value when it reaches a certain value. Furthermore, our method has slight fluctuations in performance for $k = 1.00N$, as some local CNVs may be ignored. **Figure 6B** shows the result with different threshold selection strategies under the framework of our method, and these threshold selection strategies contain some simple threshold schemes, boxplot, GMM, and VBGMM. One can observe that in addition to a single simple threshold scheme, the VBGMM is significantly better than other threshold selection strategies. Additionally, although the simple threshold scheme (*i.e.*, top n) is promising when a suitable n value is found, such as 'top $0.10N$' on NA12878 and 'top $0.05N$' on NA19239, it is a challenge to determine the n value in real-world applications. To prove the effectiveness of the KNN detector, the comparison of the KNN with the LOF (Breunig et al., 2000) and the IF (Liu et al., 2012) detector is shown in **Table 1**. 'Detector + VBGMM' denotes that Detector calculates the outlier scores for all genome segments, and these scores are transformed into binary labels by the VBGMM. Note that the input of Detector is the RD values for all genome segments (*i.e.*, R), and * represents the KNNCNV. Here the highest value in each column is highlighted. The result indicates that KNN outperforms LOF and IF except for NA12891 and ranks first in the average performance among the six real datasets.

Analysis of Cancer Samples from the European Genome-Phenome Archive

The cancer samples involve a lung cancer sample (*i.e.*, EGAD00001000144_LC) and two ovarian cancer samples (*i.e.*, EGAR00001004802_2053_1 and EGAR00001004836_2561_1), and they can be obtained from the European Genome-Phenome Archive (https://ega-archive.org/). These samples are genome-wide samples (22 autosome chromosomes) and have no confirmed CNVs (*i.e.*, ground truth). Thus, the performance of methods cannot be quantified by the precision, sensitivity, and F1-score. As a remedy, the ODS is adopted to quantify the performance of methods, and the ODS for the jth method can be formulated as **Eq. 13**.

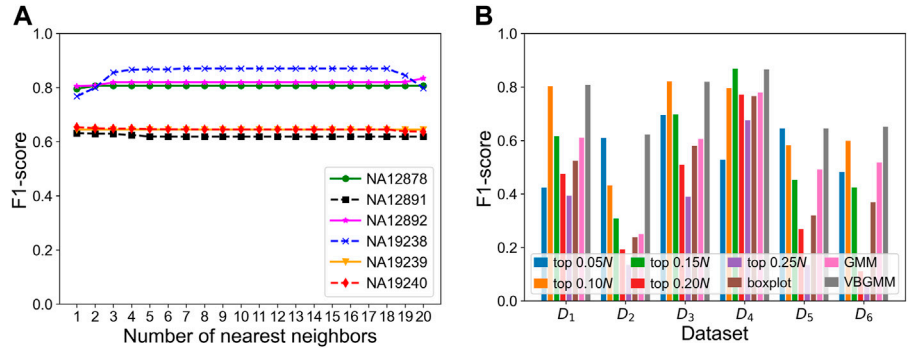

FIGURE 6 | Performance of the KNNCNV in terms of F1-score on the six real datasets. **(A)** Performance of our method with different the number of nearest neighbors, where 1–20 denote 0.05, *0.10*, 0.15*N*, . . . , 0.95*N*, and 1.00*N*, respectively. **(B)** Result with different threshold selection strategies under the framework of our method, where D_1-D_6 denote NA12878, NA12891, NA12892, NA19238, NA19239, and NA19240, respectively. Additionally, the 'top *n*' represents a simple threshold scheme, and *n* belongs to the set {0.05, *0.10, 0.15, 0.20*, 0.25*N*}. Specifically, after the outlier scores are ranked in descending order, the scheme treats the genome segments corresponding to the first *n* scores as CNVs. Note that 1–20 and *n* are rounded down, and *N* is the number of genome segments.

TABLE 1 | F1-score on six real blood datasets.

Methods	NA12878	NA12891	NA12892	NA19238	NA19239	NA19240	Average
IF + VBGMM	0.3790	0.1553	0.4247	0.5455	0.3764	0.3015	0.3637
LOF + VBGMM	0.6424	**0.6453**	0.6813	0.6906	0.6290	0.5968	0.6476
KNN + VBGMM*	**0.8068**	0.6325	**0.8194**	**0.8658**	**0.6449**	**0.6444**	**0.7356**

TABLE 2 | ODS on three genome-wide samples (22 autosome chromosomes).

Sample	CNVnator	FREEC	CNV_IFTV	CNV-LOF	KNNCNV
EGAD00001000144_LC	0.0062	0.0026	0.0212	0.1452	**1.1204**
EGAR00001004802_2053_1	0.1176	0.1488	3.4559	**10.5594**	8.6538
EGAR00001004836_2561_1	0.6182	2.4948	0.9263	2.8583	**5.6250**
Average	0.2473	0.8821	1.4678	4.5210	**5.1331**

$$ODS(j) = m(j)_{cnv} \cdot m'(j)_{cnv}, \tag{13}$$

where the definitions of $m(j)_{cnv}$ and $m'(j)_{cnv}$ refer to the prior work (Yuan et al., 2020). The comparison of the KNNCNV with peer methods on the three genome-wide samples (22 autosome chromosomes) is shown in **Table 2**, and these peer methods consist of CNVnator, FREEC, CNV_IFTV, and CNV-LOF. Here the highest value in each row is shown in bold. The result illustrates that in samples EGAD00001000144_LC and EGAR00001004836_2561_1, the KNNCNV outperforms peer methods and ranks second in the remaining sample. In addition, our method achieves the highest average ODS among the three genome-wide samples.

DISCUSSION

This paper proposes a new method used for whole genome sequencing, called KNNCNV, which can detect CNVs using NGS data. The KNNCNV first calculates the outlier score for any genome segment based solely on its first *k* nearest-neighbor distances. Specifically, the average value of these distances is considered as the outlier score for the genome segments. Finally, based on the VBGMM, these scores for all genome segments are converted into a succession of binary labels to indicate which genome segments are CNVs. Note that the outlier score calculation schemes for KNNCNV and CNV-LOF (Yuan et al., 2021a) are all based on the first *k* nearest-neighbor distances between a genome segment and the remaining ones. The difference between these two types of scores is that the KNNCNV treats solely the average value of the first *k* nearest-neighbor distances as its scores, while the scores of CNV-LOF require the further calculation of reachability distance, local reachability density, and local outlier factor. Thus, in contrast to CNV-LOF, KNNCNV is not only simpler but also has less computing overhead. Compared to the existing methods, the KNNCNV has two key features: 1) the outlier score for any genome segment can be obtained by the average outlier score scheme, which is not only easy to extend to other data types but also improves the power of detection CNVs, especially the local

CNVs that are likely to be masked by their surrounding regions; 2) the posterior distribution of these scores is approximated by the VBGMM, which can obtain a series of binary labels without a pre-determined threshold.

We conduct experiments on simulated and real datasets to show the effectiveness of the KNNCNV. The comparisons of our method with peer methods are made, and the results show that the KNNCNV achieves encouraging performance in terms of F1-score. In addition, we verify the effectiveness of each part of the KNNCNV. The results indicate that the VBGMM is an effective threshold selection strategy, and the KNN is a simple and effective detector. Therefore, the KNNCNV might become a promising tool for the detection of CNVs.

As for the potential disadvantages of our method, when calculating the outlier scores for all genome segments, there is a natural assumption that the *k*-nearest neighbor distance of outliers (*i.e.*, CNVs) is much larger than that of normal points. In other words, it assumes that the CNVs regions only account for a small fraction of the whole genome. However, the CNVs regions may cover a large fraction of the whole genome in some cancers, so the KNNCNV may not detect CNVs accurately in that case. In future work, we would be dedicated to solving the case that the CNVs regions account for a large proportion of the whole genome.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

KX, KL, and XY contributed to conception and design of the study. XY organized the datasets. KX, KL, HA, and XY wrote and revised the first draft of the manuscript. KL designed the computer programs. YC and SW provided financial support for this publication. XY guided the whole work. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported by the Guangxi Key Laboratory of Trusted Software (No. KX202041) and the Opening Fund of Shandong Provincial Key Laboratory of Network based Intelligent Computing (University of Jinan) and University Innovation Team Project of Jinan (2019GXRC015).

ACKNOWLEDGMENTS

This is a short text to acknowledge the contributions of specific colleagues, institutions, or agencies that aided the efforts of the authors.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcell.2021.796249/full#supplementary-material.

REFERENCES

Abyzov, A., Urban, A. E., Snyder, M., and Gerstein, M. (2011). CNVnator: an Approach to Discover, Genotype, and Characterize Typical and Atypical CNVs from Family and Population Genome Sequencing. *Genome Res.* 21 (6), 974–984. doi:10.1101/gr.114876.110

Aggarwal, C. (2017). *Outlier Analysis*. Cham: Springer.

Angiulli, F., and Pizzuti, C. (2002). "Fast Outlier Detection in High Dimensional Spaces," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Editors T. Elomaa, H. Mannila, and H. Toivonen (Helsinki, Finland: Springer). doi:10.1007/3-540-45681-3_2

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. 5th Edition. Springer.

Boeva, V., Popova, T., Bleakley, K., Chiche, P., Cappo, J., Schleiermacher, G., et al. (2012). Control-FREEC: a Tool for Assessing Copy Number and Allelic Content Using Next-Generation Sequencing Data. *Bioinformatics* 28 (3), 423–425. doi:10.1093/bioinformatics/btr670

Boeva, V., Zinovyev, A., Bleakley, K., Vert, J.-P., Janoueix-Lerosey, I., Delattre, O., et al. (2010). Control-free Calling of Copy Number Alterations in Deep-Sequencing Data Using GC-Content Normalization. *Bioinformatics* 27 (2), 268–269. doi:10.1093/bioinformatics/btq635

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). "LOF: Identifying Density-Based Local Outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (Dallas, Texas: ACM).

Corduneanu, A., and Bishop, C. (2001). Variational Bayesian Model Selection for Mixture Distribution. *Artif. Intelligence Stat.* 18, 27–34.

Dharanipragada, P., Vogeti, S., and Parekh, N. (2018). iCopyDAV: Integrated Platform for Copy Number Variations-Detection, Annotation and Visualization. *PLoS One* 13 (4), e0195334. doi:10.1371/journal.pone.0195334

K, Y., G, H., and Ning, Z. (2016). Structural Variation Detection from Next Generation Sequencing. *Next Generat Sequenc & Applic* 01, 0007. doi:10.4172/2469-9853.S1-007

Li, H., and Durbin, R. (2010). Fast and Accurate Long-Read Alignment with Burrows-Wheeler Transform. *Bioinformatics* 26 (5), 589–595. doi:10.1093/bioinformatics/btp698

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., et al. (2009). The Sequence Alignment/map Format and SAMtools. *Bioinformatics* 25 (16), 2078–2079. doi:10.1093/bioinformatics/btp352

Li, J., Fan, Z., Shen, F., Pendleton, A. L., Song, Y., Xing, J., et al. (2020). Genomic Copy Number Variation Study of Nine Macaca Species Provides New Insights into Their Genetic Divergence, Adaptation, and Biomedical Application. *Genome Biol. Evol.* 12 (12), 2211–2230. doi:10.1093/gbe/evaa200

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based Anomaly Detection. *ACM Trans. Knowl. Discov. Data* 6, 1–39. doi:10.1145/2133360.2133363

Miller, C. A., Hampton, O., Coarfa, C., and Milosavljevic, A. (2011). ReadDepth: a Parallel R Package for Detecting Copy Number Alterations from Short Sequencing Reads. *PLoS One* 6 (1), e16327. doi:10.1371/journal.pone.0016327

Onsongo, G., Baughn, L. B., Bower, M., Henzler, C., Schomaker, M., Silverstein, K. A. T., et al. (2016). CNV-RF Is a Random forest-based Copy Number Variation Detection Method Using Next-Generation Sequencing. *J. Mol. Diagn.* 18 (6), 872–881. doi:10.1016/j.jmoldx.2016.07.001

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine Learning in Python. *J. Machine Learn. Res.* 12, 2825–2830.

Ramasubramanian, V., and Paliwal, K. K. (1992). Fast K-Dimensional Tree Algorithms for Nearest Neighbor Search with Application to Vector

Quantization Encoding. *IEEE Trans. Signal. Process.* 40 (3), 518–531. doi:10.1109/78.120795

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). "Efficient Algorithms for Mining Outliers from Large Data Sets," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (Dallas, Texas: ACM). doi:10.1145/342009.335437

Redon, R., Ishikawa, S., Fitch, K. R., Feuk, L., Perry, G. H., Andrews, T. D., et al. (2006). Global Variation in Copy Number in the Human Genome. *Nature* 444 (7118), 444–454. doi:10.1038/nature05329

Schrider, D. R., Houle, D., Lynch, M., and Hahn, M. W. (2013). Rates and Genomic Consequences of Spontaneous Mutational Events in *Drosophila melanogaster*. *Genetics* 194 (4), 937–954. doi:10.1534/genetics.113.151670

Shahidi Zandi, A., Tafreshi, R., Javidan, M., and Dumont, G. A. (2013). Predicting Epileptic Seizures in Scalp EEG Based on a Variational Bayesian Gaussian Mixture Model of Zero-Crossing Intervals. *IEEE Trans. Biomed. Eng.* 60 (5), 1401–1413. doi:10.1109/TBME.2012.2237399

Sim, C. H., Gan, F. F., and Chang, T. C. (2005). Outlier Labeling with Boxplot Procedures. *J. Am. Stat. Assoc.* 100 (470), 642–652. doi:10.1198/016214504000001466

Smith, S. D., Kawash, J. K., and Grigoriev, A. (2015). GROM-RD: Resolving Genomic Biases to Improve Read Depth Detection of Copy Number Variants. *PeerJ* 3, e836. doi:10.7717/peerj.836

Teo, S. M., Pawitan, Y., Ku, C. S., Chia, K. S., and Salim, A. (2012). Statistical Challenges Associated with Detecting Copy Number Variations with Next-Generation Sequencing. *Bioinformatics* 28 (21), 2711–2718. doi:10.1093/bioinformatics/bts535

Tzikas, D. G., Likas, A. C., and Galatsanos, N. P. (2008). The Variational Approximation for Bayesian Inference. *IEEE Signal. Process. Mag.* 25, 131–146. doi:10.1109/MSP.2008.929620

Unckless, R. L., Howick, V. M., and Lazzaro, B. P. (2016). Convergent Balancing Selection on an Antimicrobial Peptide in Drosophila. *Curr. Biol.* 26 (2), 257–262. doi:10.1016/j.cub.2015.11.063

Venkatraman, E. S., and Olshen, A. B. (2007). A Faster Circular Binary Segmentation Algorithm for the Analysis of Array CGH Data. *Bioinformatics* 23 (6), 657–663. doi:10.1093/bioinformatics/btl646

Wang, H., Nettleton, D., and Ying, K. (2014). Copy Number Variation Detection Using Next Generation Sequencing Read Counts. *BMC Bioinformatics* 15 (1), 109. doi:10.1186/1471-2105-15-109

Xie, K., Liu, K., Alvi, H. A. K., Ji, W., Wang, S., Chang, L., et al. (2021). IhybCNV: an Intra-hybrid Approach for CNV Detection from Next-Generation Sequencing Data. *Digital Signal. Process.* 121, 103304. doi:10.1016/j.dsp.2021.103304

Yuan, X., Bai, J., Zhang, J., Yang, L., Duan, J., Li, Y., et al. (2018). CONDEL: Detecting Copy Number Variation and Genotyping Deletion Zygosity from Single Tumor Samples Using Sequence Data. *Ieee/acm Trans. Comput. Biol. Bioinf.* 17 (4), 1. doi:10.1109/TCBB.2018.2883333

Yuan, X., Li, J., Bai, J., and Xi, J. (2021a). A Local Outlier Factor-Based Detection of Copy Number Variations from NGS Data. *Ieee/acm Trans. Comput. Biol. Bioinf.* 18 (5), 1811–1820. doi:10.1109/TCBB.2019.2961886

Yuan, X., Yu, G., Hou, X., Shih, I.-M., Clarke, R., Zhang, J., et al. (2012). Genome-wide Identification of Significant Aberrations in Cancer Genome. *BMC Genomics* 13, 342. doi:10.1186/1471-2164-13-342

Yuan, X., Yu, J., Xi, J., Yang, L., Shang, J., Li, Z., et al. (2021b). CNV_IFTV: an Isolation forest and Total Variation-Based Detection of CNVs from Short-Read Sequencing Data. *Ieee/acm Trans. Comput. Biol. Bioinf.* 18 (2), 539–549. doi:10.1109/TCBB.2019.2920889

Yuan, X., Zhang, J., Yang, L., Bai, J., and Fan, P. (2018). Detection of Significant Copy Number Variations from Multiple Samples in Next-Generation Sequencing Data. *IEEE Trans.on Nanobioscience* 17 (1), 12–20. doi:10.1109/TNB.2017.2783910

Yuan, X., Zhang, J., and Yang, L. (2017). IntSIM: an Integrated Simulator of Next-Generation Sequencing Data. *IEEE Trans. Biomed. Eng.* 64 (2), 441–451. doi:10.1109/TBME.2016.2560939

Zare, F., Dow, M., Monteleone, N., Hosny, A., and Nabavi, S. (2017). An Evaluation of Copy Number Variation Detection Tools for Cancer Using Whole Exome Sequencing Data. *BMC Bioinformatics* 18 (1), 286. doi:10.1186/s12859-017-1705-x

Zhao, M., Wang, Q., Wang, Q., Jia, P., and Zhao, Z. (2013). Computational Tools for Copy Number Variation (CNV) Detection Using Next-Generation Sequencing Data: Features and Perspectives. *BMC Bioinformatics* 14 (11), S1. doi:10.1186/1471-2105-14-S11-S1

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.