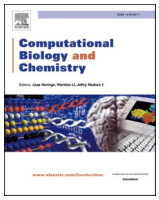
Computational Biology and Chemistry 99 (2022) 107720

Contents lists available at ScienceDirect

Computational Biology and Chemistry

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



Research article

CNVABNN: An AdaBoost algorithm and neural networks-based detection of copy number variations from NGS data

Xuan Wang [a], Junqing Li [a,b,*], Tihao Huang [a]

[a] School of Computer Science and Technology, Liaocheng University, Liaocheng, China
[b] School of Information Engineering, HengXing University, Qingdao, China

ARTICLE INFO

Keywords:
Copy number variation
AdaBoost algorithm
Next-generation sequencing
Neural network
Read depth

ABSTRACT

Copy number variation (CNV) is a non-negligible structural variation on the genome. And next-generation sequencing (NGS) technology is widely used to detect CNVs due to the feature of high throughput and low cost on the whole genome. Based on the original MFCNV method, this paper proposes an improved CNV detection method, which is called CNVABNN. In comparison to the MFCNV method, CNVABNN has three advantages: (1) It adds detectable categories, and refines the categories of loss into hemi_loss and homo_loss. (2) It utilizes the idea of integrated learning. The AdaBoost algorithm is used as the core framework and neural networks are used as weak classifiers, then CNVABNN combines all of the weak classifiers into a strong classifier. The overall performance of CNV detection is improved by using the strong classifier. (3) The detection is optimized by predicting CNVs twice through neural networks and voting mechanisms. To evaluate the performance of CNVABNN, six existing detection methods are used for comparison. The experimental results show that CNVABNN achieves better results in terms of precision, sensitivity, and F1-score for both simulated and real samples.

1. Introduction

CNV is an important genetic structural variation, which mainly refers to chromosomal aberrations, including copy number deletion, duplication, inversion, and other copy number changes in genome segments. It can be divided into microscopic and submicroscopic levels according to size, where the submicroscopic level is the main type of CNV studied so far, and the size of submicroscopic CNV is usually between 1 KB and 3 MB (Alkan et al., 2011). Copy number variants (CNVs) in humans are associated with nonallelic genes on homologous chromosomes. And the nonallelic genes paired incorrectly lead to the emergence of CNVs during meiosis or mitosis (Freeman et al., 2006). There is a certain number of CNVs in the human body, most of which are benign and have no bad impact. A few CNVs can cause diseases, such as Alzheimer's disease (Byman et al., 2020), epilepsy (Hirabayashi et al., 2019), rheumatoid arthritis (Aslam et al., 2020), and other diseases caused by CNV. Therefore, analysis of different categories of copy numbers has an important impact on the treatment of human diseases and the application of targeted drugs (Fanciulli et al., 2010).

Next-generation sequencing (NGS) is the common sequencing technology used for CNV detection (van Dijk et al., 2018; Ambardar et al., 2016; Van Dijk et al., 2014). Based on NGS data, a lot of research on CNV has been conducted: PSE-HMM investigates the problem of CNV detection under NGS data, and solves it by using a Hidden Markov Model (HMM), which uses position-specific emission probabilities to predict copy number deletion and duplication (Malekpour et al., 2017). PattRec is an automated CNV detection tool developed by Roca, which can detect both point mutations and CNVs (Roca et al., 2020). ADM-CNV considers CNV detection as a quadratic optimization problem and accurately extracts the signal from NGS data by adding sparse and smoothing constraints (Zhang et al., 2016). CNV-RF detects CNVs in four steps: preprocessing, segmentation, filtering, and classification, and CNVs are detected through a random forest approach (Onsongo et al., 2016). Faced with the complexity of processing due to the oversized NGS data, Sinha developed GenSeg, a CNV segmentation algorithm that can accurately identify breakpoints based on specific data onto the DNA genome (R. Sinha et al., 2020). IcopyDAV builds a modular framework based on NGS data, which includes modules for preprocessing, segmentation, variable calling, annotation, and visualization, to detect CNVs in an integrated manner (Dharanipragada et al., 2018). Based on

* Corresponding author at: School of Computer Science and Technology, Liaocheng University, Liaocheng, China.
E-mail address: lijunqing@lcu-cs.com (J. Li).

https://doi.org/10.1016/j.compbiolchem.2022.107720
Received 28 January 2022; Received in revised form 22 June 2022; Accepted 23 June 2022
Available online 25 June 2022

NGS data, four basic strategies for CNV detection have been gradually developed (Zhao et al., 2013): read depth (RD) (Yoon et al., 2009), paired-end mapping (Korbel et al., 2007), split-read (Ye et al., 2009), and sequence assembly (Nijkamp et al., 2012), among which the RD-based method is a popular and preferred method for CNV detection.

Currently, researchers have proposed several detection methods through RD-based strategies. These methods can be classified into two categories based on the type of approach: methods based on probability statistics and methods based on machine learning. For probability statistics-based methods, BagGMM uses the average coverage in genome segments for rough detecting CNVs, followed by a Gaussian mixture model (GMM) to detect the remaining CNVs (Li et al., 2019). In the literature (Xu et al., 2016), the identification of CNV is described as a quadratic optimization problem based on single-cell sequencing data, and then an alternating direction minimization method (ADMM) is used to solve the problem. CNVeM takes into account the uncertainty inherent in read mapping and uses a maximum likelihood estimation method to estimate CNVs on nucleotides. Then, CNVeM uses an expectation-maximization (EM) algorithm to find the optimal set of feasible solutions (Wang et al., 2013). Based on exon sequencing data, CODEX constructs a potential factor model associated with Poisson distribution. In addition, this method uses a normalization model to eliminate the bias of GC fraction, exon target, and the CNVs are estimated by a Poisson probability-based segmentation algorithm (Jiang et al., 2015). Based on CODEX, CODEX2 improves the breadth of detection through denoising operations (Jiang et al., 2018). CONDEL combines a peel-off scheme to evaluate the significance of bins and then establishes a mixed statistical model to detect CNVs (Yuan et al., 2018). For exon-level CNVs, DeviCNV constructs a linear regression model for each probe and identifies candidate regions by comparing the RD ratios of individual probes and all probes, combined with a circular binary segmentation algorithm (CBS). Therefore, different RD ratios reflect the different reliability of CNV candidate regions (Kang et al., 2018). By analyzing SNP sequencing data, PenCNN utilizes an HMM-based approach to detect CNVs (Wang et al., 2007). In addition to methods based on probability statistics, machine learning-based CNV detection has also been a popular topic in recent years. CNV_IFTV detects CNVs by a random forest algorithm and performs denoising by a full variation method, then numerous experiments have proved the effectiveness of this method (X. Yuan et al., 2019). RKDOSCNV is designed as a method based on kernel density estimation, and this method can estimate the local kernel density distribution of the sequencing data, then use relative kernel density outlier scores (RKDOS) to detect potential CNVs (G. Liu et al., 2020). CNV_LOF uses the CBS algorithm to segment the sequencing data, then assigns an outlier factor to each segment, followed by a box line plot to determine CNVs (X. Yuan et al., 2019,). Based on CNV_LOF, CIRCNV transforms the dimensions of the RD signal, and experiments showed that this method optimizes the results of CNV detection (Zhao et al., 2021). CONY obtains RD signals through a Bayesian hierarchical model and reversible-jump Markov chain Monte Carlo (RJMCMC) (Wei and Huang, 2020). The idea of deep learning developed on machine learning methods was also being applied to the detection of CNV. cnnCNV proposes a framework based on convolutional neural networks. In the cnnCNV method, the outputs of multiple existing CNV detection tools are combined on the basis of the multiple detection theories, and images of candidate regions are generated. Then, the trained framework can determine the authenticity of the candidate regions (Ding et al., 2018). The neural network can be built using CNV-MEANN, and the neural network's weights and thresholds are improved using the Mind Evolutionary Algorithm (MEA), therefore the optimized neural network model is a good tool to detect CNVs (T. Huang et al., 2021). DL-CNV compares the performance of CNN and logistic regression algorithms for CNV detection, which concludes that CNN has better generalization capabilities. It is verified that deep learning techniques for CNV detection are possible (Zhang et al., 2020).

The above-mentioned RD-based methods have yielded good results.

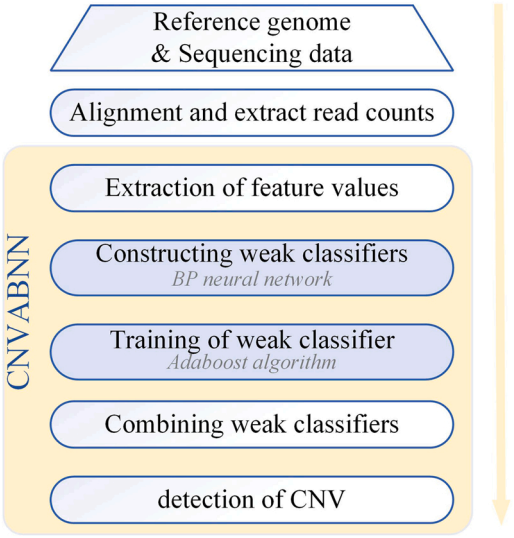


Fig. 1. The workflow of CNVABNN.

However, these methods tend to rely too much on RD and are still difficult to detect inconspicuous CNVs. MFCNV has used neural works and multiple features to optimize the results of the detection to some extent, but some problems remain in MFCNV and other methods: (1) Few methods achieve excellent detection results from low purity, low coverage sequencing data. (2) The instability of the single algorithmic model can lead to occasional situations in the detection of CNV. Therefore, it is still very important to propose a stable detection method of strong generalization power. Based on the above considerations, we propose the following method as an improvement: CNVABNN, which uses the backpropagation (BP) neural network as the weak classifier and the AdaBoost algorithm as the framework. By forming multiple weak classifiers into a strong classifier, the categories of copy numbers can be determined accurately. And the idea of integrated learning improves the efficiency of the classification and increases the stability of the method. CNVABNN has been tested on simulated and real samples, respectively, and compared with some classical CNV detection methods. The results show that CNVABNN can detect the vast majority of CNVs in samples. The following are CNVABNN's significant contributions: (1) A detailed classification of copy numbers is provided, containing four main categories: gain, normal, hemi_loss, and homo_loss. (2) Combined with integrated learning. With the AdaBoost algorithm as a framework and the neural networks as weak classifiers, all the weak classifiers form a strong classifier to improve the performance and stability of predicting CNVs. (3) The neural network is used to predict the CNV during the training of weak classifiers, and the voting mechanism is used to determine the final categories of copy numbers. Prediction in two stages improves the overall performance of the method.

The remainder of the article is organized as follows: The workflow of CNVABNN is explained in Section 2, which includes the preprocessing of sequencing data and related calculations; Section 3 compares CNVABNN with several peer methods on simulated and real datasets, and makes an analysis in terms of precision, sensitivity, F1-score, and stability; Section 4 summarizes CNVABNN and analyzes the current shortcomings of this method, as well as presenting an outlook for future improvements.

2. Method

CNVABNN is a detection method for single samples and uses the RD-based strategy to detect CNVs in sequencing data. The workflow of this method is shown in Fig. 1. Firstly, preprocessing is performed on the reference sequence and the input data, on the one hand, the positions marked with 'N' in the reference sequence are removed to ensure the

plausibility of the reference data. The sequencing data obtained directly are unordered short reads, so on the other hand, unordered reads need to be contrasted and sorted by specific tools. The CNVs are then predicted in four steps using CNVABNN: (1) Extraction of feature values. And four features are extracted and normalized. (2) Training of classifiers based on BP neural network and AdaBoost algorithm, which mainly contains two parts: constructing neural networks as weak classifiers and training for weak classifiers. Under the framework of the AdaBoost algorithm, the combined weights of weak classifiers are assigned according to the number of misclassified bins by weak classifiers. And the sample weights are also updated to enhance the attention of the next weak classifier for misclassified bins. (3) Combination of weak classifiers into a strong classifier. All weak classifiers are assigned combined weights according to the number of misclassified bins, and the strong classifier is formed according to these weights. (4) Detection of CNV. In the first stage of detection, CNVs of samples are predicted by each weak classifier to calculate the error, and for the second stage, a voting mechanism is used to predict CNV. By the two-stage prediction, CNVABNN improves the performance of detecting CNVs.

The code for CNVABNN is available for free on the website (https://github.com/1010392946/CNVABNN.git). And the following is a detailed explanation of the above steps.

2.1. Preprocessing

The sequencing data obtained by NGS are generally stored in fastq format, where fastq files contain many sequencing reads. Sequencing data were compared with reference sequences by BWA software (Li and Durbin, 2010) to sort the unordered genomes generated during NGS, then the read count profile can be extracted by samtools (Li et al., 2009). The normal human genome sequence is an arrangement of 4 bases, 'A', 'C', 'G', 'T'. In addition to these four types of bases, 'N' markers in the position of the reference sequence represent incorrect bases in the sequencing process or invalid sequencing positions. In order to make the read count profile more logical and obtain the processed sequencing data, we need to remove the positions with 'N' markers from the reference sequence. According to the read count profile, the processed sequencing data can be divided into consecutive and non-repeating bins, and the size of each bin is set to 1000 bp in this paper.

2.2. Extraction of feature values

In this paper, feature values are extracted from the reads. A total of four features (R_i, G_i, C_i, q_i) are extracted by CNVABNN. The read count profile can be obtained by preprocessing, and the RD of each bin is calculated from the average read count, which is the first feature R_i. In addition, the GC fraction is defined as the content of guanine and cytosine, and the degree of randomness in sequencing can be reflected in the GC fraction. Therefore, the GC fraction is an important factor for detecting CNV. In fact, sequence coverage is also influenced by GC fraction (Bentley et al., 2008; Dohm et al., 2008). According to formula (1), the GC fractions of the reads are extracted as the second feature.

$$G_i = n_i \tag{1}$$

Where G_i represents the GC fraction of the i-th bin, and n_i represents the total number of guanine and cytosine in the i-th bin. Also, the mapping quality and the relationship between adjacent positions are factors that cannot be ignored. q_i represents the mapping quality of the i-th bin, and the higher the q_i, the better the quality of the i-th bin. The extraction of mapping quality is similar to that of the GC fraction and will not be repeated here. The method of calculating the relationship between adjacent positions is referenced from the literature (Zhao et al., 2020), which can be described as formula (2).

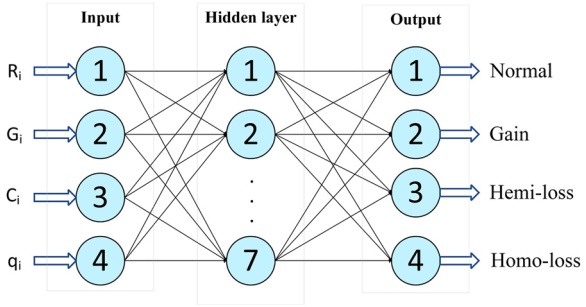


Fig. 2. The structure of the BP neural network.

$$C_i = \left| R_i - \frac{1}{Wl + Wr} \sum_{i-Wl, j \neq i}^{i+Wr} R_j \right| \tag{2}$$

Where C_i represents the quantification of the relationship between the i-th bin and its neighboring bins, R_i represents the RD of the i-th bin, Wlrepresents the number of neighboring bins from the left of the i-th bin, and Wr represents the number of neighboring bins from the right of the i-th bin. The correlation between positions is quantified by calculating the difference in RD between the current bin and neighboring bins. To ensure that the input data is reasonable, a normalization operation is performed on the obtained features, and the normalization process is shown in the formula (3).

$$y = \frac{(y_{\max} - y_{\min}) * (x - x_{\min})}{x_{\max} - x_{\min}} + y_{\min} \tag{3}$$

Where y represents the normalized feature, $y_{\max}$ and $y_{\min}$ represent the maximum and minimum values of the normalized feature; x represents the feature before normalization, $x_{\max}$ and $x_{\min}$ represent the maximum and minimum values of the feature to be normalized, respectively.

During the calculation of RD, the GC fraction may create a certain amount of noise, thus affecting the distribution of the true RD signal. The traditional approach to this problem is to correct GC bias caused by unequally distributed reads (Yoon et al., 2009), but this approach has not achieved significant results. Therefore, in this paper, multiple features are extracted to avoid the correction process.

2.3. Constructing BP neural networks as weak classifiers

For classification issues, neural networks are effective tools. The literature (Gong et al., 2020) uses the probabilistic neural network (PNN), learning vector quantization neural network (LVQ), and Elman neural network (ENN) to analyze the intracranial electroencephalography (iEEG) signals respectively. The classification accuracy of all three neural networks is above 70%, where PNN performs the best. According to the AdaBoost algorithm, the weak classifiers are required to have a certain level of classification ability, and therefore, the classical BP neural network is adopted as the weak classifier in this paper. The BP neural network is a multilayer feedback neural network based on error backpropagation that has a strong nonlinear mapping capability. However, the learning speed of BP neural networks is slow, and the networks often require numerous iterations to reach convergence. The literature (Hecht-Nielsen, 1992) contains a more extensive explanation of the BP neural network. Therefore, the reasons for the choice of the BP neural network in this paper are as follows: (1) The BP neural network can be used for classification problems and is effective in classifying unbalanced data. The literature (Chen et al., 2020) identified four conditions for pulses through an optimized three-layer BP network, whose accuracy of identification exceeded 90 %. (2) There is no perfect way to determine

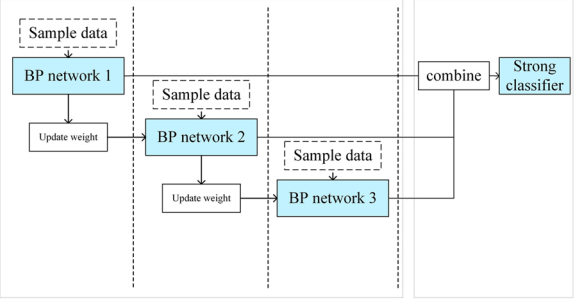


Fig. 3. Training of weak classifiers.

the number of nodes in the hidden layer of a BP neural network, and this uncertainty can be compensated for by the AdaBoost algorithm. Since the AdaBoost algorithm can train multiple BP networks, and the final classification result is determined by all networks together, so a weak classifier with high performance is not necessary for AdaBoost. For a single BP network, it is also not necessary to determine an exact number of hidden layer nodes because the structure of the network does not need to be optimal. In the following, the structure of the BP neural network used in this paper is shown in Fig. 2.

The neural network used in this paper has four input nodes, seven hidden layer nodes, and four output nodes. As shown in Fig. 2, the leftmost end of the figure refers to the four features (R_i, G_i, C_i, q_i) illustrated above. The features are used as the four input nodes to the neural network. Similarly, the four output nodes refer to the four categories of copy numbers, normal, gain, hemi_loss, and homo_loss. And the four categories are generally defined as follows: (1) Normal state, where the copy number is 2; (2) Gain state, where the copy number is greater than 2; (3) Hemi_loss state, where the copy number is 1; (4) Homo_loss state, where the copy number is 0. The neural network is fully connected, and between the hidden and input layers, there is a sigmoid activation function, which can be described using the formula (4):

$$S(n) = \frac{1}{1+e^{-n}} \tag{4}$$

$$n = wx + b \tag{5}$$

Where n in the formula (4) can be described by the formula (5). In formula (5), x represents the input of the current node, w represents the weights of the nodes between the different layers, and b represents the bias of the corresponding node.

2.4. Training of weak classifier

The AdaBoost algorithm is a classic representative of integrated learning and is well-adapted to different datasets (Freund and Schapire, 1997). In the field of machine learning, building a perfect model is difficult, and a predictive model will often inevitably get the wrong prediction. The advantage of integrated learning is that when one classifier is wrong, another classifier can correct it back. In this paper, a neural network is utilized as a weak classifier. And the process of training weak classifiers by the AdaBoost algorithm can be described in Fig. 3. In Fig. 3, the block on the left shows the iterative process of the weak classifier, and each weak classifier processes the sample data serially. There is a dependency on each weak classifier, and the next weak classifier updates the weights based on the results predicted by the previous weak classifier. Based on the framework of the AdaBoost algorithm, the weak classifier is trained in 4 steps:

Step 1: Initialization of the data, which includes the construction of the neural network, and the assignment of the initial sample weight D. The CNVABNN involves two types of weights: the first is the combined weight a_k, based on the combined weight, a strong classifier can be formed by the weak classifiers; the second is the sample weight D. D reflects the importance that the weak classifier places on different bins, and the process of initializing D is shown in the formula (6).

$$D_k(i) = \{(1/m)|i = 1, 2, 3.m\} \tag{6}$$

Where k represents the number of weak classifiers, and the number of weak classifiers utilized in this paper is 3; m represents the total number of bins.

Step 2: Weak classifiers predict the categories of copy numbers. For each weak classifier, predictions are made on the sample data and the error is evaluated. The error is calculated as shown in the formula (7).

$$E_k = \left\{ \left(\sum_i D_k(i) \right) \middle| x_k(i) \neq y(i), i = 1, 2, ., m \right\} \tag{7}$$

Where k represents the number of weak classifiers, x_k represents the predicted outcome of the k-th weak classifier, and y represents the expected outcome of the bin. The sum of D for bins where the expectation and the prediction do not match is calculated as the error of the current weak classifier.

Step 3: Calculate the combined weight of the current weak classifier, based on the error obtained in step 2. The calculation method refers to AdaBoost SAMME in the literature (Hastie et al., 2009), and the calculation of combined weight is shown in formula (8).

$$a_k = \log \frac{1 - E_k}{E_k} + \log\left(class - 1\right) \tag{8}$$

Where *class* represents the number of data categories, and the number of categories about copy number is equal to 4. When E_k is larger, and the a_k is smaller, which means that the strong classifier assigns high weight to weak classifiers with small errors.

Step 4: The sample weight D of the next weak classifier is updated by using the combined weight a_k of the previous weak classifier. The update process is shown in the formula (9).

$$D_{k+1}(i) = \frac{D_k(i)}{B_k} * \begin{cases} \exp(-a_k), x_k(i) = y(i) \\ \exp(a_k), x_k(i) \neq y(i) \end{cases} \tag{9}$$

Where B_k is the normalization factor, whose main function is to normalize the updated sample weights, x_k represents the predicted outcome of the k-th weak classifier and y represents the expected outcome of the bin. In the formula (9), when $x_k(i) \neq y(i)$, it means that the weak classifier k gets wrong prediction results at the i-th bin. Thus, the i-th bin will be given a larger sample weight D in the next weak classifier $k + 1$. The larger the sample weight, the more attention the weak classifier pays to the bin. Therefore, the weak classifier assigns larger sample weights to misclassified bins, causing the next classifier to focus on these misclassified bins. When a weak classifier has trained all the bins, the algorithm will return to step 2 to predict the sequencing samples with the next weak classifier. And a strong classifier will be formed until all the weak classifiers have finished predicting.

2.5. Combining weak classifiers

According to the combined weights, all weak classifiers produce a strong classifier, and the construction of the strong classifier is based on a voting mechanism. For each bin, the combined weights of multiple weak classifiers with the same prediction result will be accumulated. Then the values obtained will be compared, and the category corresponding to the largest value represents the category predicted by the strong classifier. The process of combining the three weak classifiers used in this paper can be shown in the formula (10):

$$S(x) = \arg\max_{class} \sum_{k=1}^{3} a(k), x_k(i) = y(i) \tag{10}$$

Where $S(x)$ represents the result of the strong classifier for each bin. For

Table 1
The detection result of CNVABNN for simulated datasets.

	Normal	Gain	Hemi_loss	Homo_loss
(0.2, 4x)	34,637	164	143	144
(0.2, 6x)	34,637	163	138	130
(0.3, 4x)	34,637	164	143	139
(0.3, 6x)	34,637	164	141	136
(0.4, 4x)	34,637	163	143	141
(0.4, 6x)	34,637	164	142	137

each possible category of the bin, the sum of the combined weights accounted for in all weak classifiers is calculated, and finally, the category with the largest sum of combined weights is selected as the result of the strong classifier for the current bin.

2.6. The detection of CNV

Two stages of detection are performed by CNVABNN: (1) In the first stage of prediction, the CNVs are predicted by BP networks; (2) In the second stage of prediction, the CNVs are predicted through a voting

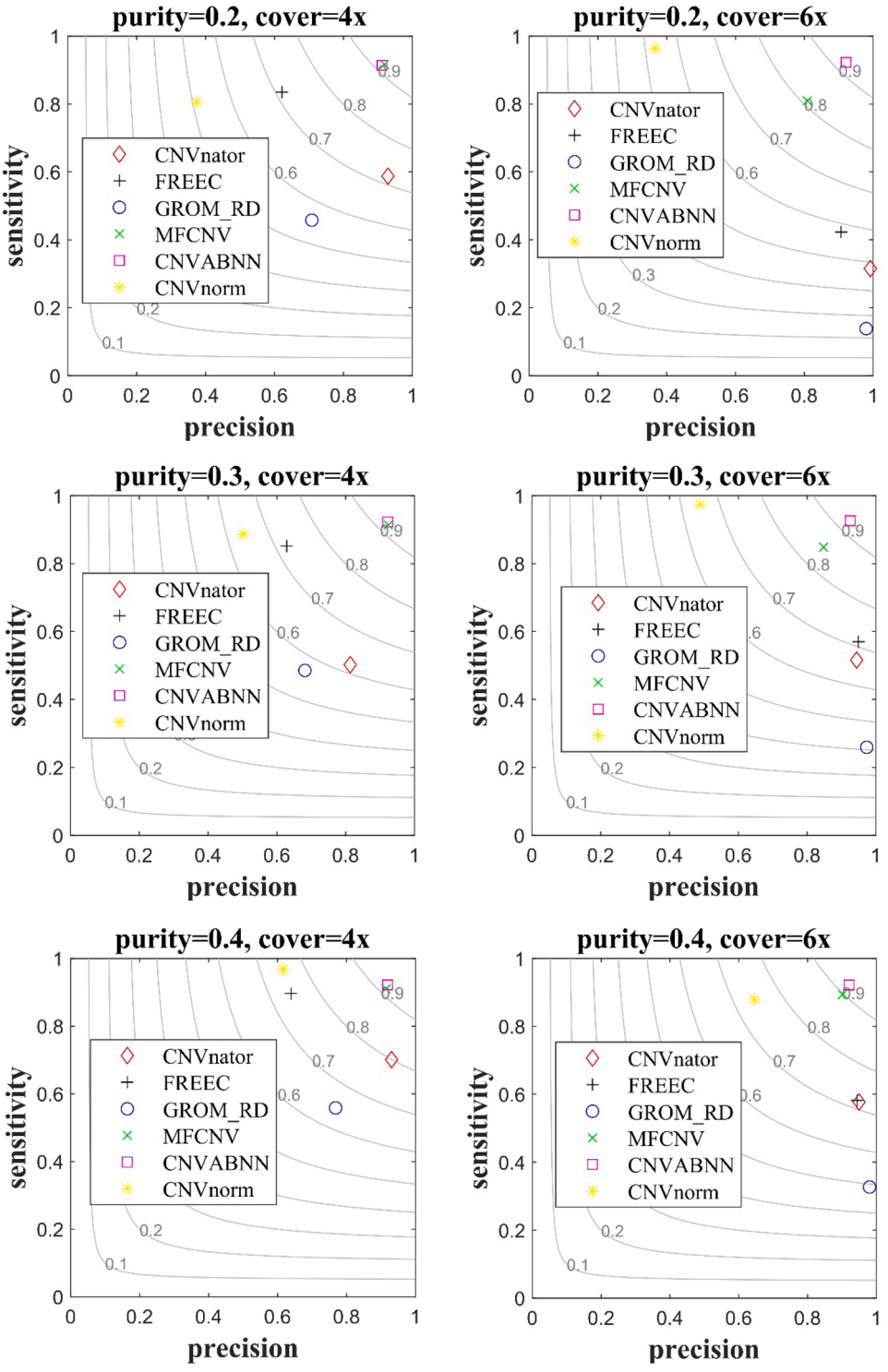


Fig. 4. Performance comparisons between the six methods in terms of sensitivity, precision and F1-score on simulated datasets.

mechanism.

The purpose of predicting CNVs by using BP networks is to calculate the error. Based on the error, the next classifier can be updated and the weights can be adjusted. In the second stage of prediction, the strong classifier combines the results from weak classifiers mainly through a voting mechanism. The voting mechanism weighs the results of the first stage prediction to obtain a comprehensive result, and the voting mechanism has been described in the process of combining weak classifiers. The voting mechanism also reduces the impact on weak classifiers due to misclassification. Therefore, this paper combines BP neural networks with AdaBoost algorithms to enhance the effectiveness of classification. The effectiveness of CNVABNN can be verified through the following experiments.

3. Results

In order to evaluate the performance of CNVABNN, we conduct experiments with real and simulated datasets and combined the CNVnator (Abyzov et al., 2011), FRREEC (Boeva et al., 2012), GROM_RD (Smith et al., 2015), CNVnorm (Gusnanto et al., 2012), ReadDepth (Miller et al., 2011), and MFCNV (Zhao et al., 2020) for comparison. The following is a specific description of the experiment.

3.1. Simulation studies

The samples used for the simulation experiments in this paper were obtained from Intsim (Yuan et al., 2016), and IntSim references chromosome 21 to produce low purity samples. Samples of simulation studies have six different simulation configurations, with tumor purity of 0.2, 0.3, 0.4, and coverage depth of 4x and 6x. 50 samples are set up in each configuration, and a total of 300 samples are generated for the simulation experiments in this paper. Firstly, we show the classification results of CNVABNN in Table 1.

As seen in Table 1, CNVABNN clearly distinguishes between four categories of copy numbers. In the simulated dataset, most of the bins are normal and only a few bins fall into the categories of gain, hemi_loss, and homo_loss. To better demonstrate the detection capabilities of CNVABNN, we analyze the precision, sensitivity, F1-score, boundary bias, and stability of CNVABNN, where the definition of precision, sensitivity and F1-score can be expressed as formulas (11−13).

$$precision = \frac{TP}{TP + FP} \tag{11}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{12}$$

$$F1 - score = \frac{2 \cdot precision \cdot sensitivity}{precision + sensitivity} \tag{13}$$

In the formulas (11−13), TP represents the number of CNVs correctly predicted, and $TP+FP$ represents the total number of CNVs predicted by CNVABNN, $TP+FN$ represents the total number of actual CNVs. Fig. 4 shows the performance of CNVABNN compared to other detection methods with simulated samples. The x-axis represents the precision of methods, the y-axis represents the sensitivity of methods, and the points on the same curve in Fig. 4 have the same F1-score. In addition, the closer to the top right corner of the figure, the better the method's performance. As shown in Fig. 4, when coverage depth is equal to 4x, the points of MFCNV and CNVABNN almost overlap. When coverage is equal to 6x, the precision and sensitivity of MFCNV are not as good as that of CNVABNN, which is due to the fact that the voting mechanism reduces the impact of misclassification on the result. Other methods also show good retrieval performance in some simulation configurations, e.g., CNVnator is slightly more precise than CNVABNN at purity of 0.2 and coverage of 4x. However, in this configuration, the sensitivity of the CNVnator is only 0.6, and the F1-score of the CNVnator is almost 0.72,

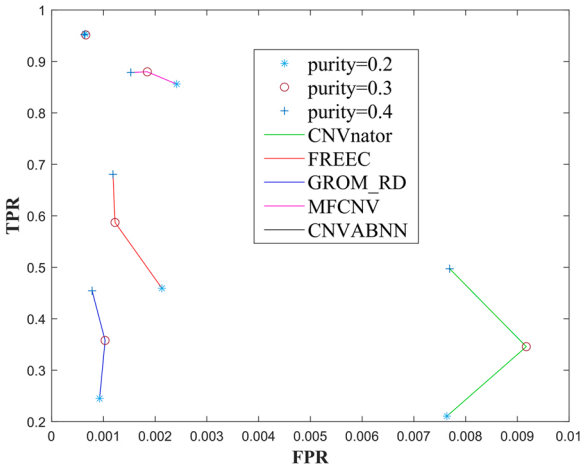


Fig. 5. ROC curves between CNVABNN and five peer methods on simulated datasets.

while the F1-score of the CNVABNN can reach 0.9. Similarly, for purity of 0.2 and coverage of 6x, FREEC has high precision, but at a sensitivity below 0.4. With purity of 0.4 and coverage of 6x, GROM_RD has a precision of nearly 100% but has a sensitivity of less than 0.4. For the above methods, the F1-score of CNVABNN is higher and the performance of CNVABNN is better.

The ROC curve is also an important tool for evaluating the performance of the method and is mainly used to analyze the true positive rate (TPR) and false positive rate (FPR) of the method. TPR is defined as the proportion of correctly predicted CNVs to all actual CNVs, and in this paper, TPR is equal to sensitivity. FPR is defined as the proportion of normal bins that are incorrectly predicted to all normal bins. TPR and FPR can be expressed by formulas (14−15):

$$TPR = sensitivity \tag{14}$$

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

In the formula (15), FP represents the number of normal bins that are incorrectly predicted, and$FP+TN$ represents the total number of actual normal bins. Fig. 5 shows the ROC curves of CNVABNN and four classical methods: CNVnator, FREEC, GROM_RD, and MFCNV. For the ROC curve, having a high TPR and a low FPR means that the detection method has good performance, i.e., the closer to the top left corner, the better the performance of the method, and the closer to the bottom right corner of Fig. 5, the worse the performance of the method. By analyzing the TPR and FPR of several methods for purity of 0.2, 0.3, and 0.4 respectively, it can be seen that CNVABNN is closest to the upper right-hand corner. When purity is 0.2, CNVABNN has the highest TPR, followed by MFCNV. The FPR of CNVABNN and GROM_RD are similar, both less than 0.001. When the purity is 0.3, CNVABNN has the highest TPR. In this case, though both FREEC and GROM_RD achieve low FPRs, they are still slightly worse than the CNVABNN. When purity is 0.4, CNVABNN still has the highest TPR, followed by MFCNV. The FPRs of MFCNV, FREEC, and GROM_RD are all less than 0.002 but the FPR of CNVABNN is less than 0.001. Overall, CNVABNN has a better performance compared to other methods when the tumor purity of the sample is low.

Since CNVABNN uses BP neural networks as weak classifiers, the boundary bias and stability of this method are compared with that of MFCNV to verify the advantages of CNVABNN. The method's boundary bias is also a critical indicator to consider, and the definition of the boundary bias was introduced in the literature (X. Yuan et al., 2019). Fig. 6 shows the comparisons of the boundary bias between CNVABNN and MFCNV. As shown in Fig. 6, the boundary bias of CNVABNN is

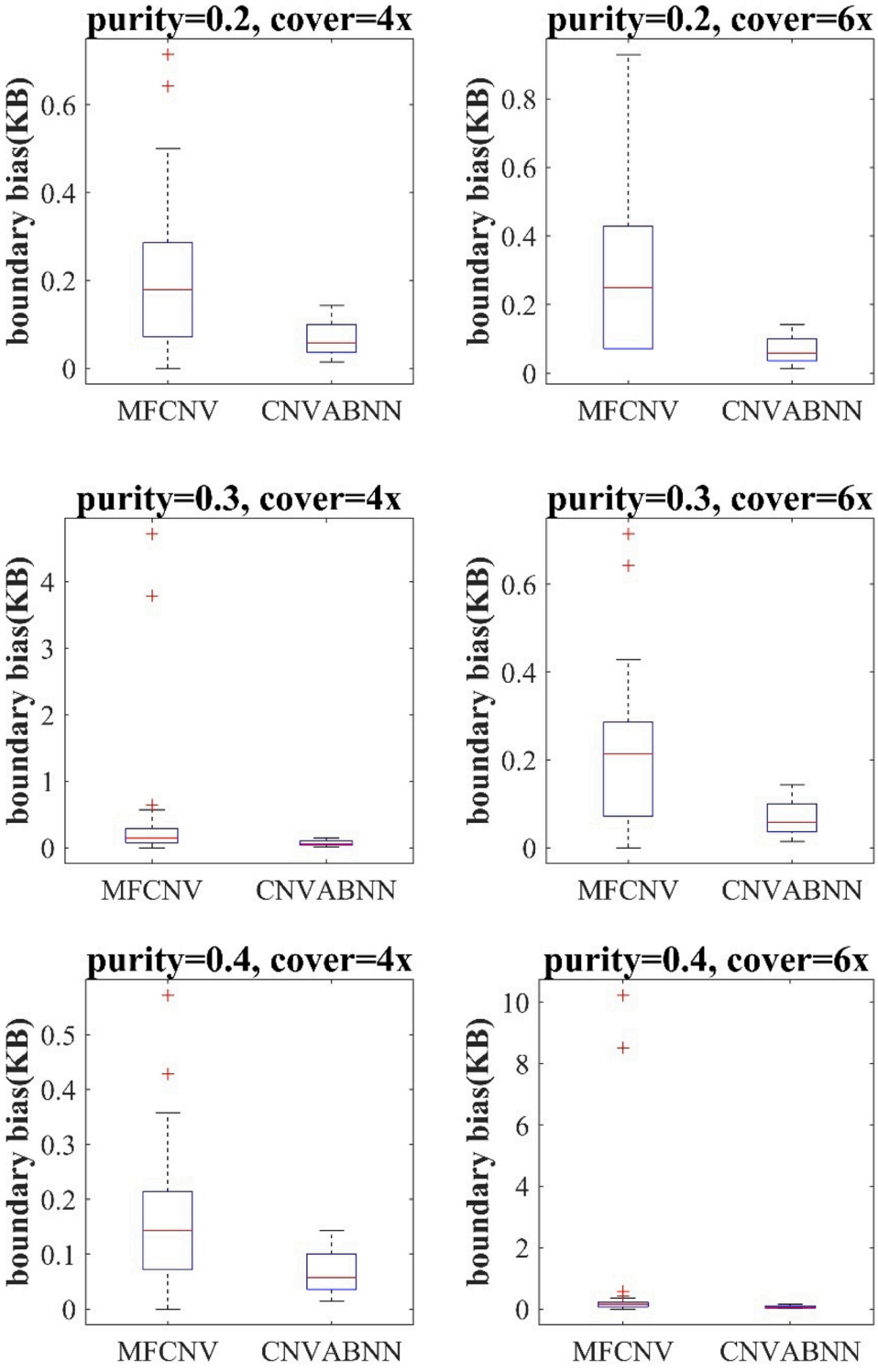


Fig. 6. Comparisons of the boundary bias between CNVABNN and MFCNV.

smaller than that of MFCNV in all six simulation configurations. This suggests that integrative learning facilitates the correction of biases, and in general CNVABNN is more practical than MFCNV for detecting CNV. To analyze the stability of CNVABNN, we calculated the relative deviations in F1-score for each of the two methods on simulated datasets. And the results are shown in Fig. 7. For each coverage depth, the average of the F1-scores measured in 50 samples is used as a criterion to judge the stability of each method. The relative deviation can be expressed in the formula (16):

$$d_i = \frac{x_i - \bar{x}}{\bar{x}} \times 100\% \tag{16}$$

Where $d_i$represents the relative deviation of the i-th sample, x_i represents the F1-score of the i-th sample, and $\bar{x}$ represents the average of the F1-score for 50 samples. As shown in Fig. 7, the relative deviation of CNVABNN is between [− 5 %, 5 %] at purity of 0.2 and 0.3, while the relative deviation of MFCNV oscillates significantly and there are samples with relative deviation below − 10 %. Similarly, at a purity of 0.4,

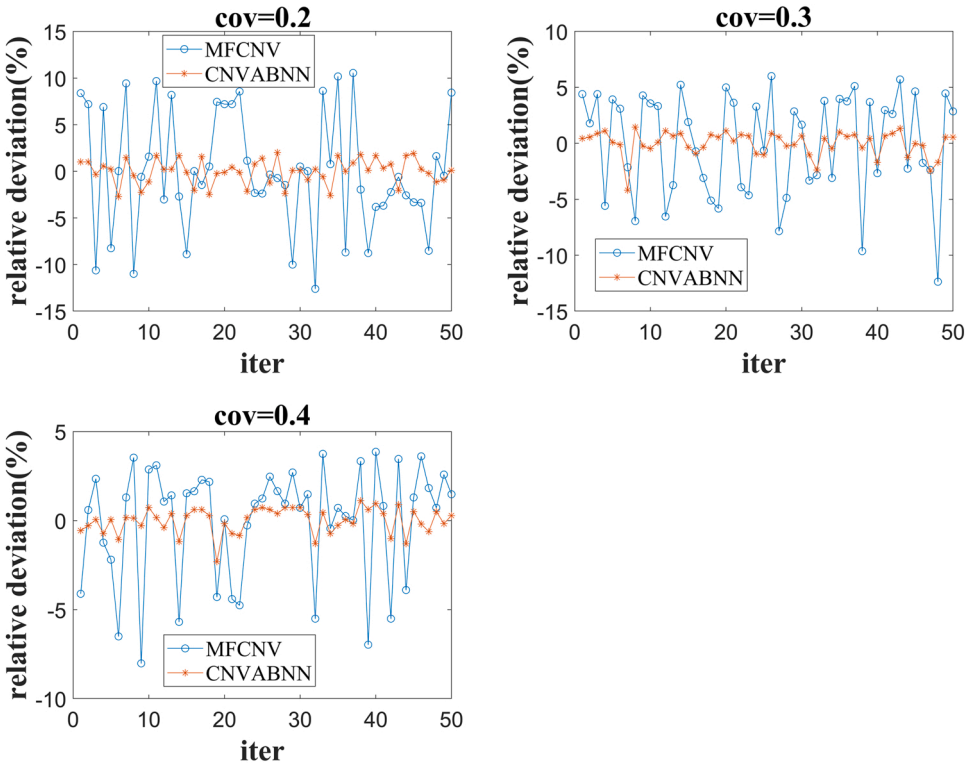


Fig. 7. Comparisons of the relative deviation between CNVABNN and MFCNV.

the relative deviation of CNVABNN is between [− 2 %, 2 %], while the relative deviation of MFCNV is between [− 8 %, 4 %]. This indicates that the F1-score of CNVABNN deviates less from the mean than those of MFCNV, and CNVABNN can detect CNVs more steadily than MFCNV for samples of different purity. However, CNVABNN will also have some samples with large relative deviation, for example, the relative deviation of the 18th sample is less than − 2% at a purity of 0.4. This is because there may be some CNVs difficult to identify in the sample, in which case the stability of CNVABNN is still better than that of MFCNV.

3.2. Detection of CNV from real datasets

In order to verify that this method is valid for real datasets, this paper obtained real sequencing samples (NA19238, NA19239, NA19240) from the 1000 Genomes Project (http://www.internationalgome.org/), the three samples came from the CEU family. Fig. 8 shows the precision and sensitivity of CNVABNN, CNVnator, FREEC, GROM_RD, MFCNV, and Read Depth under three sets of real sequencing samples. As seen in Fig. 8, MFCNV has the highest precision among the three samples but the sensitivity is not high, while CNVABNN has the highest sensitivity among the three samples and the precision is only second to MFCNV. The point of CNVABNN is located in the upper right corner of Fig. 8, therefore, CNVABNN has the highest F1-score and achieves a good balance of sensitivity and precision. The number of CNVs discovered by CNVABNN and other methods respectively under three real sequencing samples is shown in Table 2.

In Table 2, the number of CNVs represents the TP value of real samples. As seen in Table 2, CNVABNN detects the highest number of CNVs, which means that the TP of CNVABNN is higher than that of other methods. On the one hand, MFCNV has the highest precision but does not identify as many CNVs as CNVABNN. Therefore, MFCNV may have ignored some CNVs. On the other hand, CNVABNN has the highest sensitivity. Although this means that CNVABNN may cause some misjudgment, i.e., generate more FPs than MFCNV, CNVABNN can identify more CNVs than MFCNV and can increase the likelihood of identifying more diseases. By weighing sensitivity and precision, CNVABNN achieves the highest F1-score, this is due to the fact that each weak classifier focuses on the bins with the wrong prediction. Each weak classifier has dependencies, therefore, the next weak classifier can be trained and optimized based on the previous classifier. In addition, the voting mechanism of the strong classifier can effectively correct for misclassified bins, which leads to a better improvement in the precision and sensitivity of this method. In summary, the CNVABNN proposed in this paper also has good adaptability for real samples.

4. Discussion and conclusion

Accurate detection of CNV is of great importance for the analysis of the human genome. In response to the instability of some current detection methods, this paper proposes a detection method combining integrated learning to detect CNVs based on MFCNV. By using the BP neural networks as weak classifiers, the sample weights of the incorrectly predicted bins are used to obtain the combined weight occupied by the current classifier. In addition, a voting mechanism is used for the combination of weak classifiers, where for each bin, the combined weights accounted for by weak classifiers with the same prediction result are accumulated separately. And the category with the largest values is used as the final category for the bin. The mainstream method in preprocessing is to correct the GC fraction of the reads before detecting, whereas, in this paper, the GC fraction is considered directly as a feature of samples. On the one hand, this method is simple to put into practice; and on the other hand, it takes into account the influence of the GC fraction on the sequencing sample rather than the RD signal alone. In simulated and real samples, the performance of CNVABNN was verified by precision, sensitivity, F1-score, ROC curve, and stability, with a total of six comparison methods used. Existing methods rarely analyze the stability of the method, while CNVABNN was compared with MFCNV and the results showed that CNVABNN has good stability. In addition, three real sequencing samples were used to verify the performance of the method, and the CNVs appeared in the real samples are

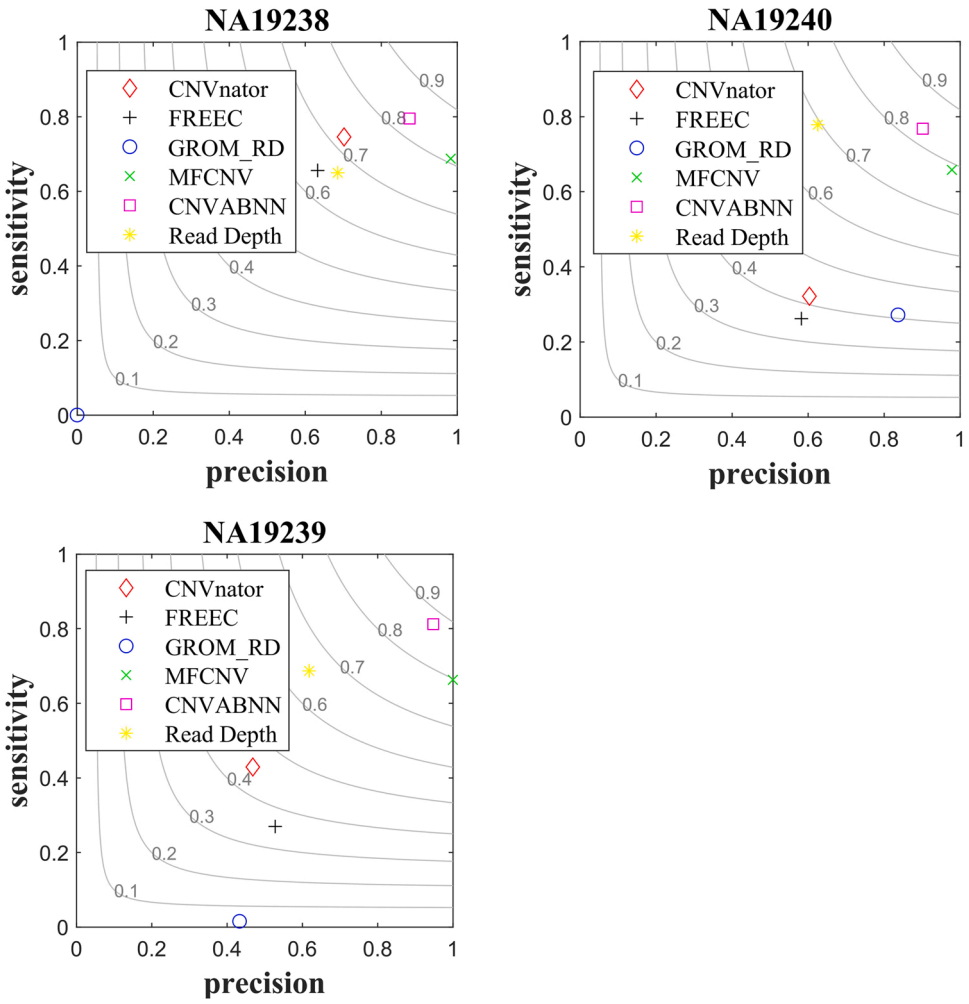


Fig. 8. Performance comparisons between the six methods in terms of sensitivity, precision and F1-score on real datasets.

Table 2
Number of CNVs detected by CNVABNN and other detection methods on real sequencing samples.

samples	CNVnator	FREEC	GROM_RD	MFCNV	CNVABNN	Read Depth
NA19238	252	222	0	233	283	220
NA19239	145	91	5	181	217	187
NA19240	109	88	9	183	225	211

available in the database of genomic variants (DGV). Experiments on both simulated and real samples demonstrate the effectiveness of CNVABNN.

CNVABNN currently has some shortcomings, which can be divided into three main points: the first one is that the different strategies used to update the weights may cause differences in the performance of the method. In this method, the sample weights and combined weights are updated according to the number of bins misclassified by the weak classifier. Therefore, different weight updated strategies lead to different training results of weak classifiers, which affects the performance of the method. The second one is that the training process of the neural network has a certain degree of randomness. CNVABNN uses neural networks as weak classifiers, and the classification results will vary each time. Therefore, the results of detection will inevitably be random. The third one is that the running speed of CNVABNN is slow. This is due to the fact that the AdaBoost algorithm requires the combination of multiple weak classifiers. The running time is positively related to the number of weak classifiers, and the more the number of classifiers, the more precision of the results. Therefore, the number of classifiers is set uncertainly. The above issues are also directions for improvement. In future work, some optimization algorithms will be considered to optimize the detection process of CNV (Li et al., 2021; Li et al., 2021; Du et al., 2022), and we need to explore new strategies to enhance the performance of CNVABNN for better detection of CNV in the clinical setting.

Funding

This study received no specific support from the public, commercial, or non-profit funding bodies.

CRediT authorship contribution statement

Xuan Wang: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Junqing Li:** Conceptualization, Resources, Supervision, Data curation, Writing – original draft, Writing – review & editing. **Tihao Huang**: Conceptualization, Methodology, Investigation, Data curation, Validation, Writing – review & editing.

Declaration of Competing Interest

None declared.

Data Availability

The real datasets presented in this study can be found in 1000 Genomes Project. And the identified CNVs can be found in the Genomic Variant Database (http://DGV.tcag.ca/).

References

Abyzov, A., Urban, A.E., Snyder, M., 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. https://doi.org/10.1101/gr.114876.110.

Alkan, C., Coe, B.P., Eichler, E.E., 2011. Genome structural variation discovery and genotyping. Nat. Rev. Genet. 12 (5), 363–376. https://doi.org/10.1038/nrg2958.

Ambardar, S., Gupta, R., Trakroo, D., Lal, R., Vakhlu, J., 2016. High throughput sequencing: an overview of sequencing chemistry. Indian J. Microbiol. 56 (4), 394–404. https://doi.org/10.1007/s12088-016-0606-4.

Aslam, M.M., John, P., Fan, K.-H., Bhatti, A., Feingold, E., Demirci, F.Y., Kamboh, M.I., 2020. Association of vpreb1 gene copy number variation and rheumatoid arthritis susceptibility. Dis. Mark. https://doi.org/10.1155/2020/7189626.

Bentley, D.R., Balasubramanian, S., Swerdlow, H.P., Smith, G.P., Milton, J., Brown, C.G., Hall, K.P., Evers, D.J., Barnes, C.L., Bignell, H.R., et al., 2008. Accurate whole human genome sequencing using reversible terminator chemistry. Nature 456 (7218), 53–59. https://doi.org/10.1038/nature07517.

Boeva, V., Popova, T., Bleakley, K., Chiche, P., Cappo, J., Schleiermacher, G., Janoueix-Lerosey, I., Delattre, O., Barillot, E., 2012. Control-freec: a tool for assessing copy number and allelic content using next-generation sequencing data. Bioinformatics 28 (3), 423–425. https://doi.org/10.1093/bioinformatics/btr670.

Byman, E., Nägga, K., Gustavsson, A.-M., Andersson-Assarsson, J., Hansson, O., Sonestedt, E., Wennström, M., 2020. Alpha-amylase 1a copy number variants and the association with memory performance and Alzheimer's dementia. Alzheimer's Res. Ther. 12 (1), 1–10. https://doi.org/10.1186/s13195-020-00726-y.

Chen, Z., Huang, A., Qiang, X., 2020. Improved neural networks based on genetic algorithm for pulse recognition. Comput. Biol. Chem. 88, 107315 https://doi.org/10.1016/j.compbiolchem.2020.107315.

Dharanipragada, P., Vogeti, S., Parekh, N., 2018. icopydav: integrated platform for copy number variations-detection, annotation and visualization. PLoS One 13 (4), e0195334. https://doi.org/10.1371/journal.pone.0195334.

van Dijk, E.L., Jaszczyszyn, Y., Naquin, D., Thermes, C., 2018. The third revolution in sequencing technology. Trends Genet. 34 (9), 666–681. https://doi.org/10.1016/j.tig.2018.05.008.

M. Ding, J. Gao, C. Ling, L. Gao, cnncnv: A sensitive and efficient method for detecting copy number variation based on convolutional neural networks, in: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2018, pp. 2744–2746, doi: http://dx.doi.org/10.1109/BIBM.2018.8621321.

Dohm, J.C., Lottaz, C., Borodina, T., Himmelbauer, H., 2008. Substantial biases in ultra-short read data sets from high-throughput dna sequencing. Nucleic Acids Res. 36 (16), e105 https://doi.org/10.1093/nar/gkn425.

Fanciulli, M., Petretto, E., Aitman, T., 2010. Gene copy number variation and common human disease. Clin. Genet. 77 (3), 201–213. https://doi.org/10.1111/j.1399-0004.2009.01342.x.

Freeman, J.L., Perry, G.H., Feuk, L., Redon, R., McCarroll, S.A., Altshuler, D.M., Aburatani, H., Jones, K.W., Tyler-Smith, C., Hurles, M.E., et al., 2006. Copy number variation: new insights in genome diversity. Genome Res. 16 (8), 949–961. https://doi.org/10.1101/gr.3677206.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55 (1), 119–139. https://doi.org/10.1006/jcss.1997.1504.

G. Liu, J. Zhang, X. Yuan, C. Wei, 2020. Rkdoscnv: A local kernel density-based approach to the detection of copy number variations by using next-generation sequencing data, Frontiers in genetics 11, doi: http://dx.doi.org/10.3389/fgene.2020.569227.

Gong, C., Zhang, X., Niu, Y., 2020. Identification of epilepsy from intracranial eeg signals by using different neural network models. Comput. Biol. Chem. 87, 107310 https://doi.org/10.1016/j.compbiolchem.2020.107310.

Gusnanto, A., Wood, H.M., Pawitan, Y., Rabbitts, P., Berri, S., 2012. Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. Bioinformatics 28 (1), 40–47. https://doi.org/10.1093/bioinformatics/btr593.

Hastie, T., Rosset, S., Zhu, J., Zou, H., 2009. Multi-class adaboost Stat. its Interface, 2, 3, pp. 349–360 doi: 10.4310/SII.2009.v2.n3.a8.

Hirabayashi, K., Uehara, D.T., Abe, H., Ishii, A., Moriyama, K., Hirose, S., Inazawa, J., 2019. Copy number variation analysis in 83 children with early-onset developmental and epileptic encephalopathy after targeted resequencing of a 109-epilepsy gene panel. J. Hum. Genet. 64 (11), 1097–1106. https://doi.org/10.1038/s10038-019-0661-x.

T. Huang, J. Li, B. Jia, H. Sang, 2021. Cnv-meann: A neural network and mind evolutionary algorithm-based detection of copy number variations from next-generation sequencing data, Frontiers in Genetics 12, doi: http://dx.doi.org/10.3389/fgene.2021.700874.

J.-Q. Li, X.-l. Chen, P.-Y. Duan, J.-h. Mou, 2021. Kmoea: A knowledge-based multi-objective algorithm for distributed hybrid flow shop in a prefabricated system, IEEE Transactions on Industrial Informatics, doi: http://dx.doi.org/10.1109/TII.2021.3128405.

Jiang, Y., Oldridge, D.A., Diskin, S.J., Zhang, N.R., 2015. Codex: a normalization and copy number variation detection method for whole exome sequencing. e39–e39 Nucleic Acids Res. 43 (6). https://doi.org/10.1093/nar/gku1363.

Jiang, Y., Wang, R., Urrutia, E., Anastopoulos, I.N., Nathanson, K.L., Zhang, N.R., 2018. Codex2: full-spectrum copy number variation detection by high-throughput dna sequencing. Genome Biol. 19 (1), 1–13. https://doi.org/10.1186/s13059-018-1578-y.

Kang, Y., Nam, S.-H., Park, K.S., Kim, Y., Kim, J.-W., Lee, E., Ko, J.M., Lee, K.-A., Park, I., 2018. Devicnv: detection and visualization of exon-level copy number variants in targeted next-generation sequencing data. BMC Bioinform. 19 (1), 1–13. https://doi.org/10.1186/s12859-018-2409-6.

Korbel, J.O., Urban, A.E., Affourtit, J.P., Godwin, B., Grubert, F., Simons, J.F., Kim, P.M., Palejev, D., Carriero, N.J., Du, L., et al., 2007. Paired-end mapping reveals extensive structural variation in the human genome. Science 318 (5849), 420–426. https://doi.org/10.1126/science.1149504.

Li, H., Durbin, R., 2010. Fast and accurate long-read alignment with burrows–wheeler transform. Bioinformatics 26 (5), 589–595. https://doi.org/10.1093/bioinformatics/btp698.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., 2009. The sequence alignment/map format and samtools. Bioinformatics 25 (16), 2078–2079. https://doi.org/10.1093/bioinformatics/btp352.

J.-Q. Li, Y. Du, K.-Z. Gao, P.-Y. Duan, D.-W. Gong, Q.-K. Pan, P.N. Suganthan, 2021. A hybrid iterated greedy algorithm for a crane transportation flexible job shop problem, IEEE Transactions on Automation Science and Engineering, doi: http://dx.doi.org/10.1109/TASE.2021.3062979.

Li, Y., Zhang, J., Yuan, X., 2019. Baggmm: calling copy number variation by bagging multiple gaussian mixture models from tumor and matched normal next-generation sequencing data. Digit. Signal Process. 88, 90–100. https://doi.org/10.1016/j.dsp.2019.01.025.

Malekpour, S.A., Pezeshk, H., Sadeghi, M., 2017. Pse-hmm: genome-wide cnv detection from ngs data using an hmm with position-specific emission probabilities. BMC Bioinform. 18 (1), 1–11. https://doi.org/10.1186/s12859-016-1296-y.

Miller, C.A., Hampton, O., Coarfa, C., Milosavljevic, A., 2011. Readdepth: a parallel r package for detecting copy number alterations from short sequencing reads. PLoS One 6 (1), e16327. https://doi.org/10.1371/journal.pone.0016327.

Nijkamp, J.F., van den Broek, M.A., Geertman, J.-M.A., Reinders, M.J., Daran, J.-M.G., de Ridder, D., 2012. De novo detection of copy number variation by co-assembly. Bioinformatics 28 (24), 3195–3202. https://doi.org/10.1093/bioinformatics/bts601.

Onsongo, G., Baughn, L.B., Bower, M., Henzler, C., Schomaker, M., Silverstein, K.A., Thyagarajan, B., 2016. Cnv-rf is a random forest–based copy number variation detection method using next-generation sequencing. J. Mol. Diagn. 18 (6), 872–881. https://doi.org/10.1016/j.jmoldx.2016.07.001.

R. Hecht-Nielsen, Theory of the backpropagation neural network, in: Neural networks for perception, Elsevier, 1992, pp. 65–93, doi: http://dx.doi.org/10.1109/IJCNN.1989.118638.

R. Sinha, R.K. Pal, R.K, De, Genseg and mr-genseg : A novel segmentation algorithm and its parallel mapreduce based approach for identifying genomic regions with copy number variations, IEEE/ACM Transactions on Computational Biology and BioinformaticsDoi: http://dx.doi.org/10.1109/TCBB.2020.3000661.

Roca, I., González-Castro, L., Maynou, J., Palacios, L., Fernández, H., Couce, M.L., Fernández-Marmiesse, A., 2020. Pattrec: An easy-to-use cnv detection tool optimized for targeted ngs assays with diagnostic purposes. Genomics 112 (2), 1245–1256. https://doi.org/10.1016/j.ygeno.2019.07.011.

Smith, S.D., Kawash, J.K., Grigoriev, A., 2015. Grom-rd: resolving genomic biases to improve read depth detection of copy number variants. PeerJ 3, e836. https://doi.org/10.7717/peerj.836.

Van Dijk, E.L., Auger, H., Jaszczyszyn, Y., Thermes, C., 2014. Ten years of next-generation sequencing technology. Trends Genet. 30 (9), 418–426. https://doi.org/10.1016/j.tig.2014.07.001.

Wang, K., Li, M., Hadley, D., Liu, R., Glessner, J., Grant, S.F., Hakonarson, H., Bucan, M., 2007. Penncnv: an integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome snp genotyping data. Genome Res. 17 (11), 1665–1674. https://doi.org/10.1101/gr.6861907.

Wang, Z., Hormozdiari, F., Yang, W.-Y., Halperin, E., Eskin, E., 2013. Cnvem: copy number variation detection using uncertainty of read mapping. J. Comput. Biol. 20 (3), 224–236. https://doi.org/10.1089/cmb.2012.0258.

Wei, Y.-C., Huang, G.-H., 2020. Cony: a bayesian procedure for detecting copy number variations from sequencing read depths. Sci. Rep. 10 (1), 1–14. https://doi.org/10.1038/s41598-020-64353-1.

X. Yuan, J. Li, J. Bai, J. Xi, 2019 A local outlier factor-based detection of copy number variations from ngs data, IEEE/ACM transactions on computational biology and bioinformaticsDoi: http://dx.doi.org/10.1109/TCBB.2019.2961886.

X. Yuan, J. Yu, J. Xi, L. Yang, J. Shang, Z. Li, J. Duan, 2019. Cnv_iftv: an isolation forest and total variation-based detection of cnvs from short-read sequencing data, IEEE/ACM transactions on computational biology and bioinformaticsDoi: http://dx.doi.org/10.1109/TCBB.2019.2920889.

Xu, B., Cai, H., Zhang, C., Yang, X., Han, G., 2016. Copy number variants calling for single cell sequencing data by multi-constrained optimization. Comput. Biol. Chem. 63, 15–20. https://doi.org/10.1016/j.compbiolchem.2016.02.007.

Y. Du, J.-Q. Li, X.-L. Chen, P.-Y. Duan, Q.-K. Pan, 2022. Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem, IEEE Transactions on Emerging Topics in Computational Intelligence, doi: http://dx.doi.org/10.1109/TETCI.2022.3145706.

Ye, K., Schulz, M.H., Long, Q., Apweiler, R., Ning, Z., 2009. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. Bioinformatics 25 (21), 2865–2871. https://doi.org/10.1093/bioinformatics/btp394.

Yoon, S., Xuan, Z., Makarov, V., Ye, K., Sebat, J., 2009. Sensitive and accurate detection of copy number variants using read depth of coverage. Genome Res. 19 (9), 1586–1592. https://doi.org/10.1101/gr.092981.109.

Yuan, X., Zhang, J., Yang, L., 2016. Intsim: an integrated simulator of next-generation sequencing data. IEEE Trans. Biomed. Eng. 64 (2), 441–451. https://doi.org/10.1109/TBME.2016.2560939.

Yuan, X., Bai, J., Zhang, J., Yang, L., Duan, J., Li, Y., Gao, M., 2018. Condel: detecting copy number variation and genotyping deletion zygosity from single tumor samples using sequence data. IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (4), 1141–1153. https://doi.org/10.1186/s13059-018-1578-y.

Zhang, Y., Cheung, Y.-M., Xu, B., Su, W., 2016. Detection copy number variants from ngs with sparse and smooth constraints. IEEE/ACM Trans. Comput. Biol. Bioinform. 14 (4), 856–867. https://doi.org/10.1109/TCBB.2016.2561933.

Zhang, Y.X., Jin, L.C., Wang, B., Hu, D., Wang, L., Li, P., Zhang, J., Han, K., Tian, G., Yuan, D., et al., 2020. Dl-cnv: a deep learning method for identifying copy number variations based on next generation target sequencing. Math. Biosci. Eng. 17, 202–215. https://doi.org/10.3934/mbe.2020011.

Zhao, H., Huang, T., Li, J., Liu, G., Yuan, X., 2020. Mfcnv: a new method to detect copy number variations from next-generation sequencing data. Front. Genet. 11, 434. https://doi.org/10.3389/fgene.2020.00434.

Zhao, H.-Y., Li, Q., Tian, Y., Chen, Y.-H., Alvi, H.A., Yuan, X.-G., 2021. Circnv: detection of cnvs based on a circular profile of read depth from sequencing data. Biology 10 (7), 584. https://doi.org/10.3390/biology10070584.

Zhao, M., Wang, Q., Wang, Q., Jia, P., Zhao, Z., 2013. Computational tools for copy number variation (cnv) detection using next-generation sequencing data: features and perspectives. BMC Bioinform. 14 (11), 1–16. https://doi.org/10.1186/1471-2105-14-S11-S1.