MaxDEL: Accurate and Efficient Calling Genomic Deletions From Single Molecular Real-Time Sequencing Using Integrated Method

Yaoxian Lv
Beijing University of Chemical Technology https://orcid.org/0000-0002-2414-4389
Lei Cai
Beijing University of Chemical Technology
Jingyang Gao (gaojy@mail.buct.edu.cn)
Beijing University of Chemical Technology https://orcid.org/0000-0003-1270-6257

Methodology

Keywords: single-molecule real-time, structural variations, machine learning, sequencing visualization, convolutional neural network.

Posted Date: March 11th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-247574/v1

MaxDEL: accurate and efficient calling genomic deletions from single molecular real-time sequencing using integrated method

Yaoxian Lv, Lei Cai*, Jingyang Gao*
College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, P. R. China
*Corresponding author: csuperlei@outlook.com; gaojy@mail.buct.edu.cn

Abstract
Background: Single-molecule real-time (SMRT) sequencing data are characterized by long reads and high read depth. Compared with next-generation sequencing (NGS), SMRT sequencing data can present more structural variations (SVs) and has greater advantages in calling variation. However, there are high sequencing errors and noises in SMRT sequencing data, which brings inaccurately on calling SVs from sequencing data. Most existing tools are unable to overcome the sequencing errors and detect genomic deletions.
Methods and results: In this investigation, we propose a new method for calling deletions from SMRT sequencing data, called MaxDEL. MaxDEL can effectively overcome the noise of SMRT sequencing data and integrates new machine learning and deep learning technologies. Firstly, it uses machine learning method to calibrate the deletions regions from variant call format (VCF) file. Secondly, MaxDEL develops a novel feature visualization method to convert the variant features to images and uses these images to accurately call the deletions based on convolutional neural network (CNN). The result shows that MaxDEL performs better in terms of accuracy and recall for calling variants when compared with existing methods in both real data and simulative data.
Conclusions: We propose a method (MAXDEL) for calling deletion variations, which effectively utilizes both machine learning and deep learning methods. We tested it with different SMRT data and evaluated its effectiveness. The research result shows that the use of machine learning and deep learning methods has great potential in calling deletion variations.

Keywords: single-molecule real-time, structural variations, machine learning, sequencing visualization, convolutional neural network.

1.Introduction
Single molecular real-time (SMRT) sequencing technology highlights a change from the short reads to the long reads. There are two main SMRT sequencing strategies: Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT). Traditional next-generation sequencing (NGS) provides a low read depth and short reads, which restrict its ability to call more variants [1-5]. SMRT sequencing data coverage is uniform and SMRT sequencing data avoids the errors caused by PCR amplification, for example, GC bias [6,7]. Moreover, studies have shown that SMRT sequencing data can find more deletions by 48% and insertions by 83% than NGS data [8]. While there have been substantial

advances in SMRT sequencing data, most calling methods have been conducted to estimate single nucleotide polymorphism (SNP) and indels. There are no new methods to call structural variations (SVs) as much and as accurately as possible on SMRT sequencing data. Therefore, how to accurately and effectively identify the SVs has become a focus on the SMRT sequencing data.

When calling genomic deletions on SMRT sequencing data, there are three main situations (called signatures) of the sequence reads mapped onto the given reference genome near the deletion site. (i) Split Reads (SR). When the read overlaps the breakpoints of a deletion border, the read consists of two parts that are not contiguous. Such a read is called split read. There are some calling methods based on it, such as PbHoney [9], pbsv, Sniffles [10], Picky [11], SVIM [12]. (ii) Read Depth (RD). Read depth within a deletion is lower than non-deletion region. If the deletion is homozygous, the read depth is closed to zero; If the deletion is heterozygous, the read depth should still be lower than expected. Read depth is the most commonly signature for calling deletions. Sniffles also uses this strategy. (iii) Local Assembly (LA). Local assembly signature is different from other two signatures. LA detects the variant region by assembling reads, such as SMRT-SV [13]. In addition, there are some other methods. For example, Next-SV [14] cleans the genome files and uses Sniffles to call variants.

Recently, deep learning methods have been used successfully to call genomic variations on NGS data. Google's DeepVariant [15], which uses deep learning methods to call SNP and Indel. It uses the base sequence as the main information and generates the image of the base, and treating variant calling as a special kind of image classification. Moreover, Cai et al. 's DeepSV [16], which is a tool to call deletions from NGS data, also uses base sequence as the main information. It generates RGB stack images from NGS data, and finally uses Convolutional Neural Networks (CNN) for image classification. DeepSV is a novel method in SVs calling. These cases of successful application of deep learning to NGS data convey a message to us: applying deep learning to SMRT sequencing data may also have a good performance in calling deletions.

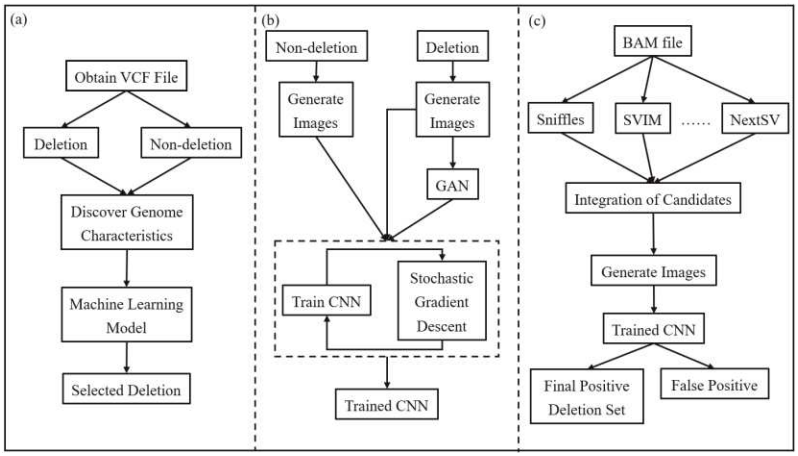
In this investigation, we propose MaxDEL, which is accurate and efficient for calling genomic deletions with integrated method. MaxDEL addresses two aspects of calling problems for SMRT sequencing data. First, that VCF file is inaccurate of the deletion regions. Second, that most of the existing tools use an empirical model to call SVs. We know that the empirical model is based on extracting features manually. However, the deep learning model can learn the features independently. We propose a distinct detection method based on CNN [17] model that can convert the variant features to images and call the genomic detections from these images.

To investigate how well MaxDEL calls the variants, we compared MaxDEL and five state-of-art methods, including NextSV, SVIM, Sniffles, SMRT-SV and Picky. The performance is measured in terms of three metrics: precision, recall, and F1-score. We also counted the false positives (FP), true positives (TP), and false negatives (FN). The results showed that the MaxDEL achieved better performance compared to those methods in simulative data and real data.

2.Method

2.1 The high-level approach

MaxDEL is a deep learning structural method that is based on the calling of SVs. Because of the SMRT sequencing data has higher sequencing noise (~15%) [18] than NGS data, MaxDEL is designed to be used with the noise data. It uses a novel method to convert sequence characteristics to the images and uses images to identify the variants. There are two main advantages to MaxDEL. The first is that MaxDEL can correct the errors in VCF and find the accurate deletion region, which can help to generate precise variant images. The second is that MaxDEL can use the strong learning ability and robustness of CNN to learn and overcome the noise from the sequencing data. In addition, another advantage of MaxDEL is that it solves the problem of data imbalance. As we know, the CNN model is sensitive when the data were unbalanced. However, the number of non-SVs was about 100 times greater than that of SVs. If we direct generate the images to call deletions, which led to reduce accuracy. MaxDEL uses GAN [19] to augment data to solve the problem of data imbalance. GAN can generate fake images with original images distribution so that the images have greater randomness and authenticity. After obtaining enough genomic images, MaxDEL uses these images data to train CNN. In order to implement the end-to-end model, MaxDEL has also complete one procedure from BAM file to calling results. First, MaxDEL collects the candidate variant from other tools



and use the random forest to find accurate breakpoints of deletions. Second, MaxDEL extracts the variant features to generate images and use GAN to augment these images. Finally, MaxDEL uses CNN to call the deletions and filter false positives. The details are shown in Figure 1.

Figure 1. The overall framework of MaxDEL. (a). Obtaining features from the deletion and non- deletion regions, training a machine learning model, and using the trained machine learning model to filter breakpoint offsets to obtain the accurate deletion regions. (b) The deletion and non-deletion regions are generated into images. Due to the small number of deletions, we can obtain enough deletion images by GAN and using these two genetic images to train CNN to get a trained CNN model. (c): Using existing tools to call variants to obtain a candidate deletion set. The trained CNN model is used to discriminate the candidate set and get the final deletion result. The trained CNN model will distinguish the candidate set and get the final deletion result.

2.2 Obtain accurate deletion regions

The accuracy of the deletion regions is related to the accuracy of calling the variants. After analyzing the BAM file of real data, it is found that the VCF has existed off-set of the breakpoint positions, which contain a part of the non-deletion region. Therefore, ac-

curately determining the deletion region is significant. In this paper, we use random forest to filter the VCF file errors and correct the breakpoint offset of variants.

(i). Genome feature extraction. Through observing the BAM file, it can be found that the deletion region has unique characteristics different from the non-deletion region. We use the sliding window to scan the reads aligned the reference genome, and adopt the RD strategy to extract the useful features. The features are including: window information, mapping type of reads, mapping quality and alignment type. The features description is shown in Table 1. Among these features, window information and mapping quality can provide us with the basic information of the reads in the target window, which is the basis for distinguishing the non-deletion region and the deletion region. Mapping type of reads can record the main matching pattern of the target window. Alignment type describes the number of primary alignments in the target window. Moreover, the average depth is obtained by formula (1). The window complexity is obtained by formulas (2) and (3).

Table 1. Feature description

Features	Description of numerical features
Window information	Average depth
	Window complexity
	Number of reads
Mapping type of reads	Number of soft-clip
	Number of deletions
Mapping quality	Average mapping quality of reads
Alignment type	Number of primary alignments

$$\mathrm{D}_E = \frac{\sum_{i=1}^{Length} d_i}{Length} \quad (1)$$

$$C = \frac{\sum_{i=1}^{Length} (D_E - d_i)^2}{Length - 1} \quad (2)$$

$$\frac{e^z - 1}{e^z}, \text{ while } z = \sqrt{\frac{C}{20}} \quad (3)$$

(ii). Calibration deletion regions. The deletion information provided by the VCF file may exist the errors. By statistic chromosomes 1 to 6 and selecting 125 deletion regions for each chromosome, we can obtain the accuracy of VCF in deletion regions (Table 2). In order to get the accurate deletion regions, we use random forest method in machine learning to identify the real deletion regions. Random forest method uses ensemble algorithm, which has high accuracy. In addition, due to the introduction of randomness, it is not easy to overfit and has anti-noise ability. When training the model, first, we shrink the deletion regions recorded in the VCF to get small deletion regions, which can greatly remove the false positive part and keep the true positive part. Then, we extract variant features from these regions to construct a multi-feature positive sample. Secondly, we extract non-variant features from wild regions to construct multi-feature negative samples. Finally, positive samples and negative samples are used to construct a training data set and train a random forest model. We use the trained model to validate the new multi-

feature samples which are needed to be distinguished. The new samples are formed by expanding both sides of the deletion regions recorded in the VCF. An example is shown in Figure 2. The results in table 2 show that the accuracy of the deletion regions is improved after the processing of the random forest model, and the overall accuracy reaches 88.16%, which proves that the random forest can correct the breakpoint offset.

Table 2. Comparison of the accuracy of the deletion regions before and after using the random forest method.

chromosomes	before	after
chr1	79.17%	82.40%
chr2	70.23%	86.61%
chr3	88.86%	87.16%
chr4	81.03%	87.96%
chr5	84.64%	89.21%
chr6	82.78%	89.27%
total	81.42%	88.16%

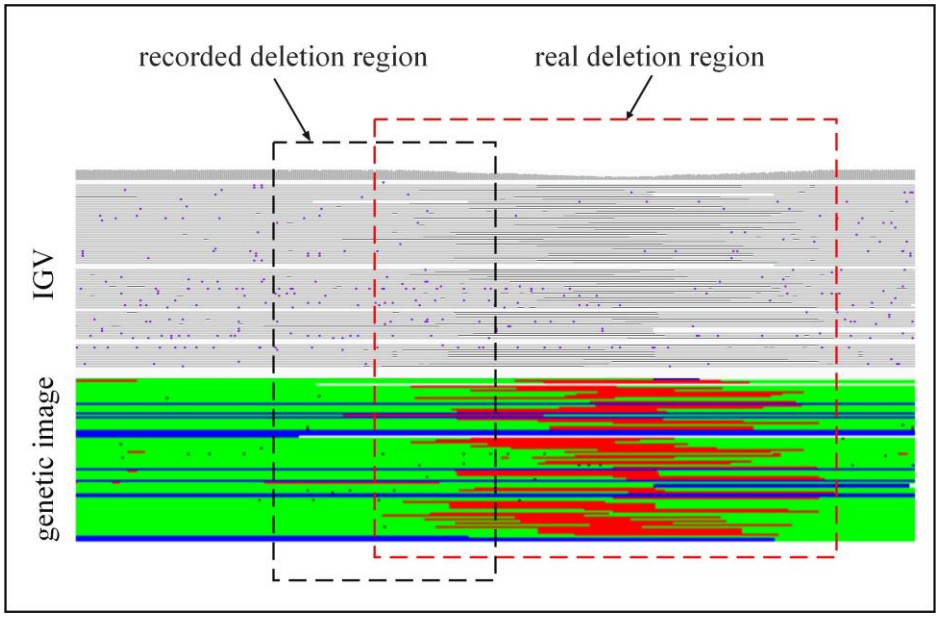


Figure 2. Visualization of genomic region and calibration result (chromosome3,33799438-33799858). The upper part is the genomic region in IGV. The lower part is the image of the corresponding region with the image strategy of MaxDEL. The black part is the deletion region recorded in VCF, and the red part is the deletion region corrected by the random forest method.

2.3 Image Generation

As we all know, BAM files contain a lot of sequence information. Converting the sequence characteristics into images can reflect more three-dimensional information between sequences. It means that it can show the positional relationship between reads in the same region and display the regional characteristics between reads. By studying real data, it is found that most of the deletions are hundreds to thousands of bp in length, and it is difficult to express the entire deletion region in a RGB image. Therefore, the ge-

nomic image generation method will be solved from two aspects: interval division and image design.

(i). Interval division. Small regions were obtained by dividing the deletion region and the non-deletion region respectively. We divide the long genomic region into small regions and generate images instead of compressing the entire genomic region. The image process can not only maintain the integrity of the read information in each small region but also improve the sensitivity of MaxDEL. When dividing the interval, the deletion region and non-deletion region are divided one by one according to the principle of fixed-length and fixed-step. The interval length here is 100bp and the step size is 75bp.

(ii). Image design. How to filling content and arranging colors of the image is significant. When generating an image, we use the CIGAR string in reads as the main information. Compared with the base sequence of reads, using CIGAR string information can not only reduce the redundancy of the image content but also keep the difference between the deletion region and non-deletion region. The CIGAR string contains four matching modes of D, M, S, and I in the selected window. They represent the four mapping types: Delete, Match, Soft-clip, and Insertion. The four mapping types are represented by RGB values (Figure 3a) as: (255, 0, 0), (0, 255, 0), (0, 0, 255), (0, 0, 0). The blank area is filled with (255, 255, 255). The corresponding CIGAR string and color are following (D, red), (M, green), (S, blue), (I, black).

Through interval division and image design, the deletion regions and the non-deletion regions are generated into images (Figure 3b and Figure 3c). It could be found that there were obvious depth differences between them, which indicated that the image generation strategy was effective and could better show their differences.

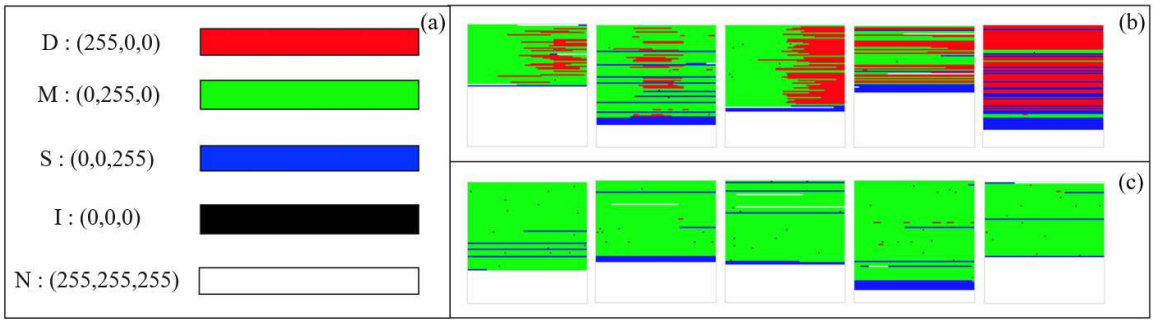


Figure 3. Image color distribution and image generation results. (a): Color assignment of genomic image. (b): Deletion region images. (c): Non-deletion region images.

2.4 Augment the image

Deep learning methods have a wide range of applications in processing image data, one of which is to augment image data. When training data is unbalanced, the CNN model will learn more knowledge on the positive samples, which will lead to not convergence. In genomic data, the number of deletion regions is extremely small compared with non-deletion regions, and there is also an imbalance in the amount of data between them. Therefore, the amount of data is too small and the sample imbalance is another obstacle to calling deletions. With the development of deep learning, Ian J. Goodfellow et al. proposed GAN in 2014 (Figure 4a). It is a powerful generation model, which can use a small amount of data to generate a large number of simulative data. MaxDEL uses GAN to augment genomic images to solve these two problems mentioned.

(i). Generation and comparison of fake images. MaxDEL uses GAN to augment deletions to overcome the imbalance of positive and negative samples. In genomic image, the direction of it is deterministic and the content is regular, which can greatly reduce the complexity of the image and maintain sufficient information. Therefore, GAN can obtain amplified images with higher quality and distribution closer to real data generated images through less training. We use GAN to augment the gene image, and set the epoch to 100,000 to get sufficient quality and number of genomic images. In the experiment, two different GAN methods are compared to find a suitable one for MaxDEL. Figure 4b and figure 4c show the images generated by two different GAN methods. Among them, figure 4b is the result of traditional GAN method, and figure 4 is the result of DCGAN [20]. It can be found that compared with Figure 4b, Figure 4 is closer to the original deletion images in color distribution, and there are fewer noises in the generated image generated by DCGAN method. Therefore, DCGAN was selected to augment the gene data.

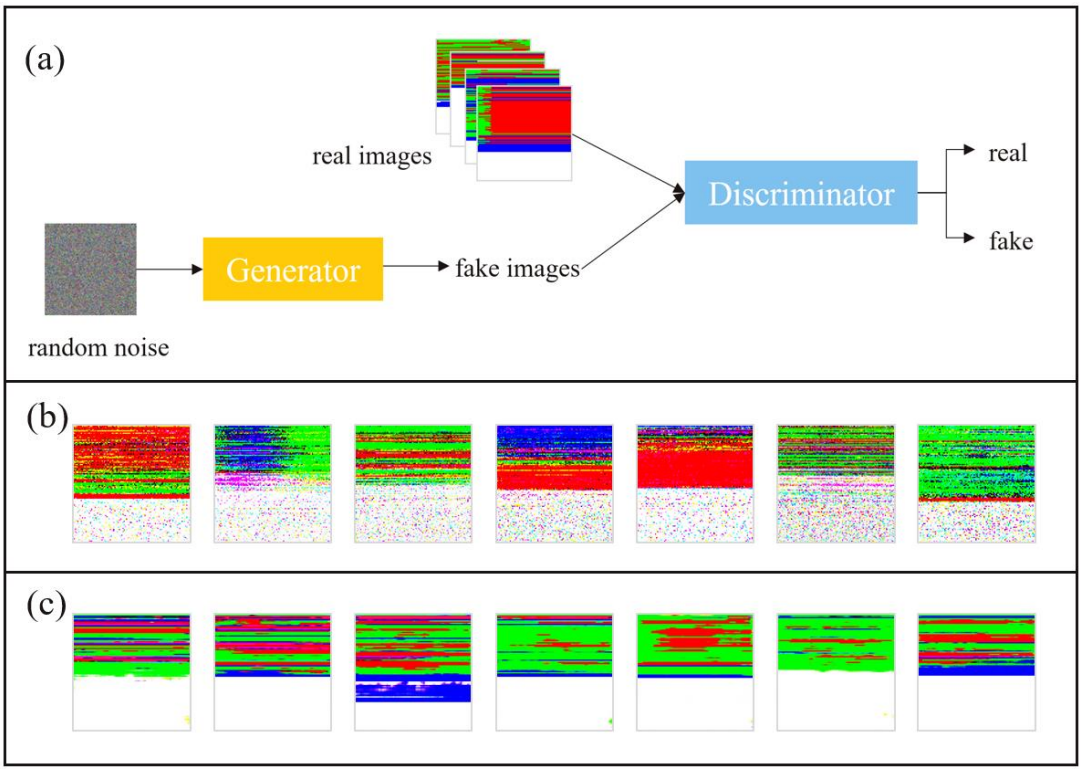


Figure 4. The structure of GAN and generated images of GAN methods. (a): General structure of GAN. (b) and (c): The generated images of original GAN and DCGAN.

(ii). Selection and comparison of activation functions. In the generator of GAN, different activation functions used in the hidden layer will also affect the final results. ReLU is a commonly used activation function, while SELU is a novel activation function, which can automatically normalize the input sample data to zero mean and unit variance. As shown in Figure 5, we compare their impact on the accuracy of the generated data and take the accuracy of fake images of each 25 rounds in the first 5,000 rounds and calculated the average result. It can be seen from Figure 5 that when we used ReLU, the accuracy of the results varied greatly. While using SELU to train data, when the epoch is close to 3000, the accuracy of the result is closer to 100%. Therefore, compared with using ReLU, SELU can better improve the accuracy of generated images and obtain high-quality generated images. SELU is used as the activation function of DCGAN to generate images.

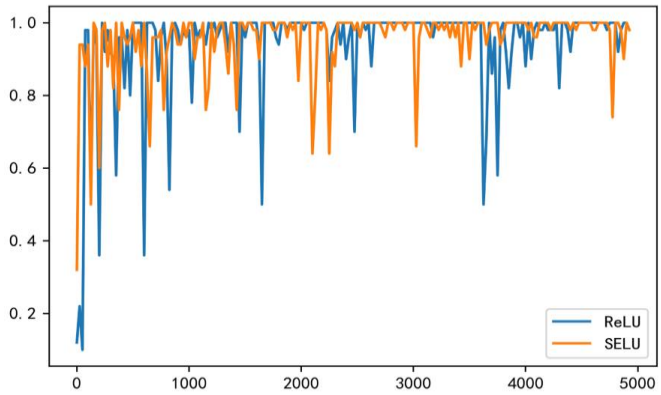


Figure 5. Accuracy comparison by using ReLU and SELU.

3. Result

3.1 Dataset

This paper uses GRCh37 as the reference genome. The real data comes from Genome in a Bottle (GIAB), which is a benchmark by Zook et al(ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/)[21]. This dataset has high-quality and it is provided through calculation and experimental verification [22]. BAM files coverage are HG002 (69x), HG003 (32X) and HG004 (30X) respectively. The first 6 chromosomes of HG002 were used as the training set, the remaining chromosomes and samples are testing sets. In addition, SURVIVOR [23], PaSS [24] and NGMLR [9] are used to generate simulative BAM data, which can increase the amount of data and verify the validity of MaxDEL. The VCF for HG002 comes from GIAB (ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/), while HG003 and HG004 use the output of existing tools as benchmark. The simulative data uses the VCF given by the simulative tool as benchmark.

3.2 Training and validating the CNN model

After image amplification, there are enough image samples, which are used as CNN input to train a CNN model. CNN has made great achievements in classification problems and computer vision. Firstly, when CNN learns image samples, it can transform the original data into abstract representation layer through representation learning. Secondly, CNN can also find some inherent features in the image through convolution processing that we have not discovered. Finally, in SMRT sequencing data, there are a large number of sequencing errors, but CNN has strong robustness to noise. While overcoming the shortcomings of the high error rate of SMRT sequencing data, it automatically learns the features of the genomic image and obtains the trained CNN model. In the following discussion, we will explore the composition of the data to get a trained CNN model.

(i). The impact of before and after sample-balance on the CNN model. The imbalance of positive and negative samples will cause many problems, which may eventually lead to a decrease in the credibility of the training model. Figure 6a, 6b shows the accuracy results of CNN for image classification when balancing the positive and negative samples. Among them, the amount of negative sample data is about 80000. The number of positive samples before balance is about 16000, and the ratio of positive and negative

samples is 1:5; After using DCGAN to augment the positive samples, the ratio of positive and negative samples is about 1:1. When unbalanced, although the accuracy and loss are the best on the training set, the results on the validation set are the opposite. Finally, on the test set, the accuracy of CNN model with unbalanced positive and negative samples is only 93.39%, and the loss is 0.2089. After the balance of positive and negative samples, the accuracy is 97.225%, and the loss is 0.1097. Therefore, it is proved that after balanced positive and negative samples, CNN model can have better discriminability to unknown samples.

(ii). The influence of different data preprocessing on the accuracy of CNN model. The data composition of the genomic image needs to be compared in several experiments. In order to explore the influence of different data composition on the accuracy of CNN, we conducted three sets of experiments for comparison: (1) image samples are generated without any processing; (2) image samples are generated only by machine learning processing; (3) image samples are generated after random forest processing and data amplification of GAN. The batch size is 128 and epoch is 50 for each comparative experiment, and the proportion of positive samples and negative samples in the training set is roughly the same. Figure 6c, 7d shows the loss and accuracy on the training set and the verification set. According to statistics, in experiment 1, the accuracy of the training set is 89.538% on average and 95.85% at the highest. The average accuracy on the validation set was 87.617% and the highest was 88%. The loss on the verification set has increased significantly, which indicates that the network has already been over-fitting at this time. Therefore, this result shows the necessity of machine learning to process deletion regions. Secondly, the accuracy and loss of the generated image only after random forest processing are in the middle position. Finally, after random forest processing and GAN image amplification, the training results and verification results of CNN model are the best. Therefore, we use the trained CNN model obtained from (3) and use this model to discriminate candidate images.

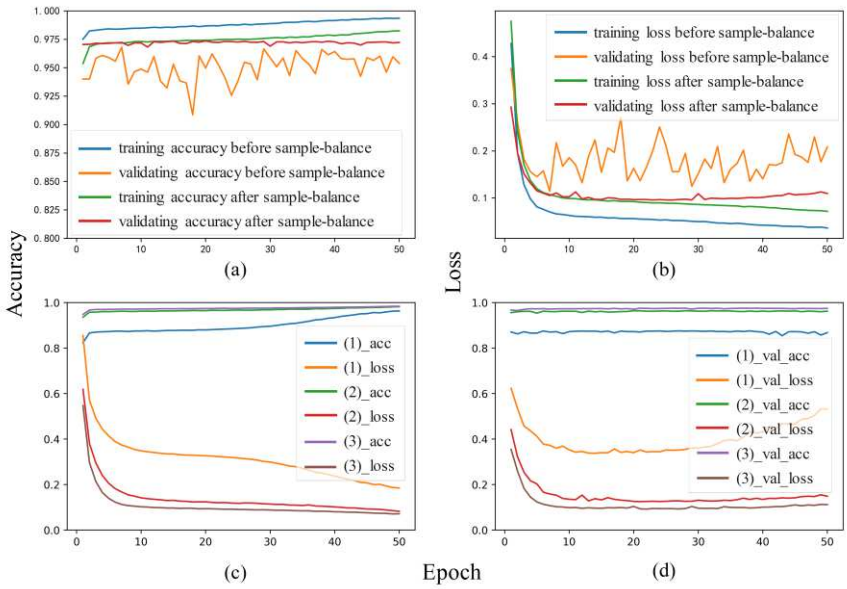


Figure 6. CNN's results in different data sets. (a): The difference of accuracy before and after sample-balance. (b): The difference of loss before and after sample-balance. (c) and (d): Accuracy and loss of different training data in the three groups. The three sets of data are described above.

3.3 Performance of the Simulative Data

Testing on simulative data is one of the methods to verify the effectiveness of the calling tool. First, we use SURVIVOR to simulate the reference genome file with variations, then generate the simulative FASTQ file through PaSS, and finally use NGMLR to align the simulative FASTQ file with the simulative reference genome to obtain the BAM file.

When generating simulative data, according to the length of the variation, the simulative data is divided into two categories: shorter variations (50bp~1000bp, category A) and longer variations (1000bp~5000bp, category B). In each category, we validate the results of homozygous variants and heterozygous variants, respectively.

(i). Calling result of category A. MaxDEL uses the trained CNN above to distinguish the candidate sets. Through Table 3 and Table 4, we can conclude that the results of MaxDEL performed well in calling homozygous deletions and heterozygous deletions. Its recall has reached more than 0.9527, which is better than existing tools. And F1-score is also the highest among all the test tools, especially when calling heterozygous variants.

Table 3. Category A. Homozygous variants

Sample	Benchmark	Tools	TP	FN	FP	R	P	F1
sim (32X)	4971	NextSV	4892	79	18	0.9841	0.9963	0.9902
		SVIM	4919	57	174	0.9885	0.9658	0.9770
		Sniffles	4792	179	5	0.9640	0.9989	0.9812
		Picky	4828	143	26	0.9712	0.9946	0.9828
		MaxDEL	**4928**	**43**	**53**	**0.9913**	**0.9894**	**0.9904**
sim (24X)		NextSV	4883	88	25	0.9823	0.9949	0.9886
		SVIM	4909	62	129	0.9875	0.9744	0.9809
		Sniffles	4534	437	3	0.9121	0.9993	0.9537
		Picky	4795	176	23	0.9646	0.9952	0.9797
		MaxDEL	**4913**	**58**	**81**	**0.9883**	**0.9838**	**0.9861**
sim (16X)		NextSV	4845	126	47	0.9747	0.9904	0.9825
		SVIM	4885	86	113	0.9827	0.9774	0.9800
		Sniffles	3230	1741	3	0.6498	0.9991	0.7874
		Picky	4779	192	21	0.9614	0.9956	0.9782
		MaxDEL	**4901**	**70**	**111**	**0.9859**	**0.9779**	**0.9819**
sim (8X)		NextSV	4698	273	2	0.9451	0.9994	0.9715
		SVIM	4875	96	67	0.9807	0.9864	0.9836
		Sniffles	472	4499	0	0.0950	1.0000	0.1734
		Picky	4692	279	13	0.9439	0.9972	0.9698
		MaxDEL	**4897**	**74**	**67**	**0.9851**	**0.9865**	**0.9858**

Table 4. Category A. Heterozygous variants

Sample	Benchmark	Tools	TP	FN	FP	R	P	F1
sim (32X)	4945	NextSV	4841	104	13	0.9790	0.9973	0.9881
		SVIM	4882	63	98	0.9873	0.9803	0.9838
		Sniffles	3704	1538	3	0.6890	0.9991	0.8156
		Picky	4748	197	19	0.9602	0.9960	0.9778
		MaxDEL	**4891**	**54**	**44**	**0.9891**	**0.9911**	**0.9901**

sim (24X)		NextSV	4811	134	10	0.9729	0.9979	0.9853
		SVIM	4876	69	84	0.9860	0.9831	0.9846
		Sniffles	1777	3168	1	0.3594	0.9994	0.5286
		Picky	4724	221	18	0.9553	0.9962	0.9753
		MaxDEL	**4884**	**61**	**81**	**0.9877**	**0.9837**	**0.9857**
sim (16X)		NextSV	4646	299	7	0.9395	0.9985	0.9681
		SVIM	4855	90	72	0.9818	0.9854	0.9836
		Sniffles	454	4491	0	0.0918	1.0000	0.1682
		Picky	4653	292	14	0.9410	0.9970	0.9682
		MaxDEL	**4875**	**70**	**74**	**0.9858**	**0.9850**	**0.9854**
sim (8X)		NextSV	3699	1246	4	0.7480	0.9989	0.8555
		SVIM	4661	284	38	0.9426	0.9919	0.9666
		Sniffles	11	4934	2	0.0022	0.8462	0.0044
		Picky	4138	807	11	0.8368	0.9973	0.9101
		MaxDEL	**4711**	**234**	**40**	**0.9527**	**0.9916**	**0.9717**

(ii). Calling result of category B. In this category, because the longer variants are complicated, the accuracy of all tools for calling deletions is reduced. From the results in Table 5 and Table 6, it can be concluded that MaxDEL still performs well. In the high coverage simulative data, the results were somewhat lower, but in the low coverage data, the results are the best. It proves the stability of MaxDEL at different coverage depths.

Table 5. Category B. Homozygous variants

Sample	Benchmark	Tools	TP	FN	FP	R	P	F1
sim (32X)	3065	NextSV	2187	878	35	0.7135	0.9872	0.8273
		SVIM	3016	49	237	0.9840	0.9271	0.9547
		Sniffles	2164	901	4	0.7060	0.9982	0.8271
		Picky	3002	63	239	0.9794	0.9263	0.9521
		MaxDEL	**3034**	**31**	**271**	**0.9899**	**0.9180**	**0.9526**
sim (24X)		NextSV	2099	966	30	0.6848	0.9859	0.8082
		SVIM	2976	89	213	0.9710	0.9332	0.9517
		Sniffles	2012	1053	2	0.6564	0.9990	0.7923
		Picky	2899	166	151	0.9458	0.9505	0.9482
		MaxDEL	**3006**	**59**	**225**	**0.9808**	**0.9304**	**0.9549**
sim (16X)		NextSV	2006	1059	22	0.6545	0.9892	0.7877
		SVIM	2974	91	169	0.9703	0.9462	0.9581
		Sniffles	1625	1440	0	0.5302	1.0000	0.6930
		Picky	2889	176	147	0.9426	0.9516	0.9471
		MaxDEL	**3004**	**61**	**221**	**0.9801**	**0.9315**	**0.9552**
sim (8X)		NextSV	1746	1319	10	0.5967	0.9943	0.7243
		SVIM	2933	132	114	0.9569	0.9626	0.9598
		Sniffles	344	2721	0	0.1122	1.0000	0.2018
		Picky	2879	186	159	0.9393	0.9477	0.9435
		MaxDEL	**2982**	**83**	**123**	**0.9729**	**0.9604**	**0.9666**

Table 6. Category B. Heterozygous variants

Sample	Benchmark	Tools	TP	FN	FP	R	P	F1
sim (32X)	3006	NextSV	2027	979	22	0.6743	0.9893	0.8020
		SVIM	2937	69	139	0.9770	0.9548	0.9658
		Sniffles	1659	1347	1	0.5519	0.9994	0.7111
		Picky	2882	124	99	0.9587	0.9668	0.9628
		MaxDEL	**2966**	**40**	**108**	**0.9867**	**0.9649**	**0.9757**
sim (24X)		NextSV	1927	1079	14	0.6411	0.9928	0.7791
		SVIM	2923	83	113	0.9724	0.9628	0.9676
		Sniffles	1084	1922	0	0.3606	1.0000	0.5301
		Picky	2910	96	54	0.9681	0.9818	0.9749
		MaxDEL	**2955**	**51**	**127**	**0.9830**	**0.9588**	**0.9708**
sim (16X)		NextSV	1750	1256	9	0.5822	0.9949	0.7345
		SVIM	2877	129	79	0.9571	0.9733	0.9651
		Sniffles	344	2662	0	0.1144	1.0000	0.2054
		Picky	2859	147	33	0.9511	0.9886	0.9695
		MaxDEL	**2940**	**66**	**119**	**0.9780**	**0.9611**	**0.9695**
sim (8X)		NextSV	1262	1744	3	0.4198	0.9976	0.5910
		SVIM	2612	394	51	0.8689	0.9808	0.9215
		Sniffles	8	2298	2	0.0027	0.8000	0.0053
		Picky	2446	560	8	0.8137	0.9967	0.8960
		MaxDEL	**2759**	**242**	**66**	**0.9178**	**0.9766**	**0.9463**

Figure 7 displays the performance of the recall rate and F1-score of each tool and MaxDEL. The testing on the low coverage data shows that MaxDEL performed the best with an overall 99.13% recall of homozygous calling results and 98.91% recall of heterozygous calling results. The testing on the high coverage data shows that MaxDEL also achieved the best no matter homozygous and heterozygous. Furthermore, we validate the performance of MaxDEL with other four tools on F1-score. We found that the results of F1-score fluctuate strongly, but the results of MaxDEL maintain the high level.

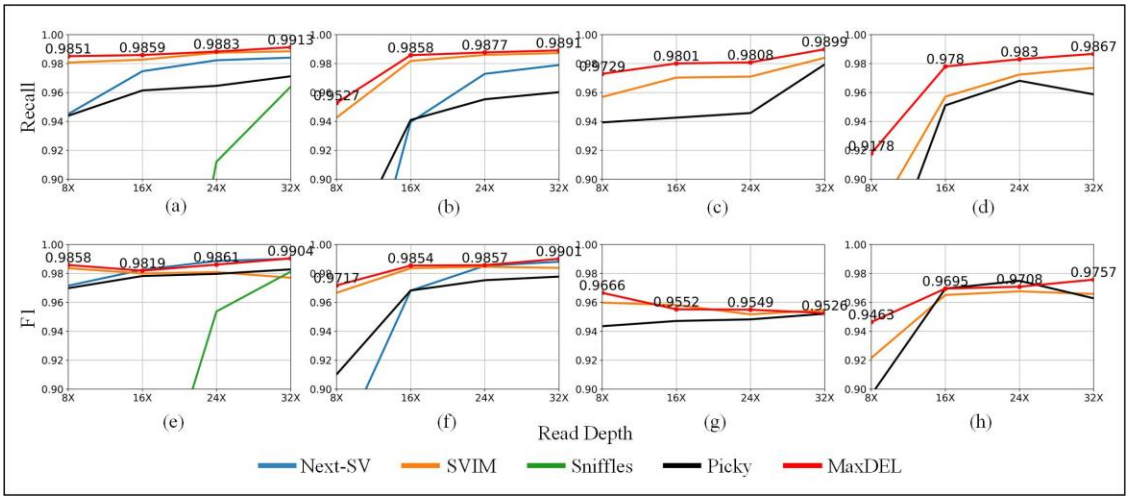


Figure 7. Comparison of existing tools and MaxDEL's recall and F1-score in different coverage depth data. The upper part is the recall and the lower part is the F1-score. The statistical range is (0.9,1). (a), (b) and (e), (f): category A, calling results of homozygous deletions and heterozygous deletions. (c), (d) and (g), (h): category B, calling results of homozygous deletions and heterozygous deletions.

3.4 Performance of the Real Data

Here we use HG002 real data to validate the MaxDEL performance in comparison with the NextSV, SMRT-SV, SVIM and Sniffles. From table 7, we found MaxDEL also can get more true positives and fewer false negatives, as well as the highest recall and F1-score. The results show that MaxDEL has the best comprehensive performance in calling deletions in real data.

Table 7. Comparison of HG002 calling results

Sample	Benchmark	Tools	TP	FN	FP	R	P	F1
HG002 (69X)	7723	NextSV	6876	847	178	0.8903	0.9748	0.9306
		SMRT-SV	2995	4728	551	0.3878	0.8446	0.5315
		SVIM	4619	3104	22	0.5981	0.9956	0.7472
		Sniffles	4366	3357	23	0.5653	0.9948	0.7209
		MaxDEL	**7364**	**359**	**387**	**0.9535**	**0.9501**	**0.9518**

Furthermore, to verify whether read depth affects the metrics of calling deletions, we use HG003 and HG005 low coverage data to test on each tool. From table 8, the MaxDEL performed the best precision of about 99.10% and recall about 96.24% on the HG003. On the HG004, the Sniffles and SMRT-SV have the lower precision, and the NextSV and SVIM results are similar. The MaxDEL has also performed the best including precision, recall, and F1-score, which confirms that the MaxDEL is reliable on calling deletions.

Table 8. Comparison of HG003 and HG004 calling results

Sample	Benchmark	Tools	TP	FN	FP	R	P	F1
HG003 (32X)	9817	NextSV	8736	1081	93	0.8899	0.9895	0.9370
		SMRT-SV	4899	4918	409	0.4990	0.9229	0.6478
		SVIM	7306	2511	53	0.7442	0.9928	0.8507
		Sniffles	4426	5391	29	0.4509	0.9935	0.6202
		MaxDEL	**9729**	**88**	**380**	**0.9910**	**0.9624**	**0.9765**
HG004 (30X)	9812	NextSV	8642	1170	82	0.8806	0.9906	0.9325
		SMRT-SV	4610	5202	11	0.4698	0.9976	0.6388
		SVIM	7172	2640	45	0.7309	0.9938	0.8423
		Sniffles	3624	6188	15	0.3693	0.9959	0.5388
		MaxDEL	**9750**	**62**	**134**	**0.9937**	**0.9864**	**0.9900**

4. Discussion and conclusion

In this paper, we define and elaborate a novel method, MaxDEL, for calling genetic deletions, using integrated framework with SMRT sequencing data. The MaxDEL method can capture the variant features of the deletions and establish the learning model between images and gene data. In our experiment, the MaxDEL method is superior to NextSV, SVIM, Sniffles, Picky and SMRT-SV, especially in recall, and F1-score.

The MaxDEL focused on three strengths on calling the deletions. The first was that MaxDEL solve the breakpoint offset problem. Especially, in the VCF file, most deletion information has errors, which makes it difficult to generate accurate images. However, MaxDEL corrects the VCF file and give the precise breakpoint and deletion region.

The second strength was that the MaxDEL provide a new gene image augmentation method, which address the sample imbalance problem. MaxDEL use GAN to generate gene image to expand the deletion samples and automatically analyze the image feature to adjust the model parameters.

The third strength was that MaxDEL provide a significant method to use image to call the SVs on the SMRT sequencing data with different coverage, which prove the deep learning model can capable of accurate and efficient calling variants.

When we explore the limits of MaxDEL, we find that it cannot infer the genotype. The MaxDEL can recognize the deletion region, but the genotype of deletion depends on the different coverage. At present, we need to one extra step to calculate the genotype when given the deletion region. Future investigations will focus on how to use MaxDEL to infer genotype.

Abbreviations

SMRT: Single-molecule real-time; NGS: Next-generation sequencing; SVs: Structural variations; VCF: Variant call format; CNN: Convolutional neural network; PacBio: Pacific Biosciences; ONT: Oxford Nanopore Technologies; SNP: Single nucleotide polymorphism; SR: Split Reads; RD: Read Depth; LA: Local Assembly.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

The real data is available online. The source code in this paper is available at https://github.com/superKSG/MaxDEL.

Competing interests

The authors declare that they have no competing interests.

Funding

Project supported by Beijing Natural Science Foundation (5182018).

Authors' contributions

Conceived and designed the experiments: YL, JG, Performed the experiments: YL, LC, Analyzed the data: YL, LC. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

References

[1] Sudmant P H, Rausch T, Gardner E J, et al. An integrated map of structural variation in 2,504 human genomes[J]. Nature, 2015, 526(7571): 75-81.

[2] Sudmant P H, Kitzman J O, Antonacci F, et al. Diversity of human copy number variation and multicopy genes[J]. Science, 2010, 330(6004): 641-646.

[3] Korbel J O, Urban A E, Affourtit J P, et al. Paired-end mapping reveals extensive structural variation in the human genome[J]. Science, 2007, 318(5849): 420-426.
[4] Handsaker R E, Van Doren V, Berman J R, et al. Large multiallelic copy number variations in humans[J]. Nature genetics, 2015, 47(3): 296.
[5] Schneider V A, Graves-Lindsay T, Howe K, et al. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly[J]. Genome research, 2017, 27(5): 849-864.
[6] Loomis E W, Eid J S, Peluso P, et al. Sequencing the unsequenceable: expanded CGG-repeat alleles of the fragile X gene[J]. Genome research, 2013, 23(1): 121-128.
[7] Rasko D A, Webster D R, Sahl J W, et al. Origins of the E. coli strain causing an outbreak of hemolytic–uremic syndrome in Germany[J]. New England Journal of Medicine, 2011, 365(8): 709-717.
[8] Chaisson M J P, Sanders A D, Zhao X, et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes[J]. Nature communications, 2019, 10(1): 1-16.
[9] English A C , Salerno W J , Reid J G . PBHoney: identifying genomic variants via long-read discordance and interrupted mapping[J]. Bmc Bioinformatics, 2014, 15(1):180.
[10] Sedlazeck F J, Rescheneder P, Smolka M, et al. Accurate detection of complex structural variations using single-molecule sequencing[J]. Nature methods, 2018, 15(6): 461-468.
[11] Gong, Liang, Wong, et al. Picky comprehensively detects high-resolution structural variants in nanopore long reads[J]. Nature Methods, 2018.
[12] Heller D, Vingron M. SVIM: structural variant identification using mapped long reads[J]. Bioinformatics, 2019, 35(17): 2907-2915.
[13] Huddleston J , Chaisson M J , Steinberg K M , et al. Discovery and genotyping of structural variation from long-read haploid genome sequence data[J]. Genome Research, 2017, 27(5):677-685.
[14] Li F , Jiang H , Depeng W , et al. NextSV: a meta-caller for structural variants from low-coverage SMRT data[J]. Bmc Bioinformatics, 2018, 19(1):180-.
[15] Poplin R, Chang P C, Alexander D, et al. A universal SNP and small-indel variant caller using deep neural networks[J]. Nature biotechnology, 2018, 36(10): 983-987.
[16] Cai L, Wu Y, Gao J. DeepSV: accurate calling of genomic deletions from high-throughput sequencing data using deep convolutional neural network[J]. BMC bioinformatics, 2019, 20(1): 665.
[17] Lecun Y , Bottou L . Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
[18] Chaisson M J, Tesler G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory[J]. BMC bioinformatics, 2012, 13(1): 238.
[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.
[20] Radford A , Metz L , Chintala S . Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J]. Computer ence, 2015.
[21] Zook J M, Hansen N F, Olson N D, et al. A robust benchmark for germline structural variant detection[J]. BioRxiv, 2019: 664623.

[22] Ho S S, Urban A E, Mills R E. Structural variation in the sequencing era[J]. Nature Reviews Genetics, 2019: 1-19.
[23] Jeffares D C , Jolly C , Hoti M , et al. Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast[J]. Nature Communications, 2017.
[24] Zhang W , Jia B , Wei C . PaSS: a sequencing simulator for PacBio sequencing[J]. BMC Bioinformatics, 2019, 20(1).

Figures

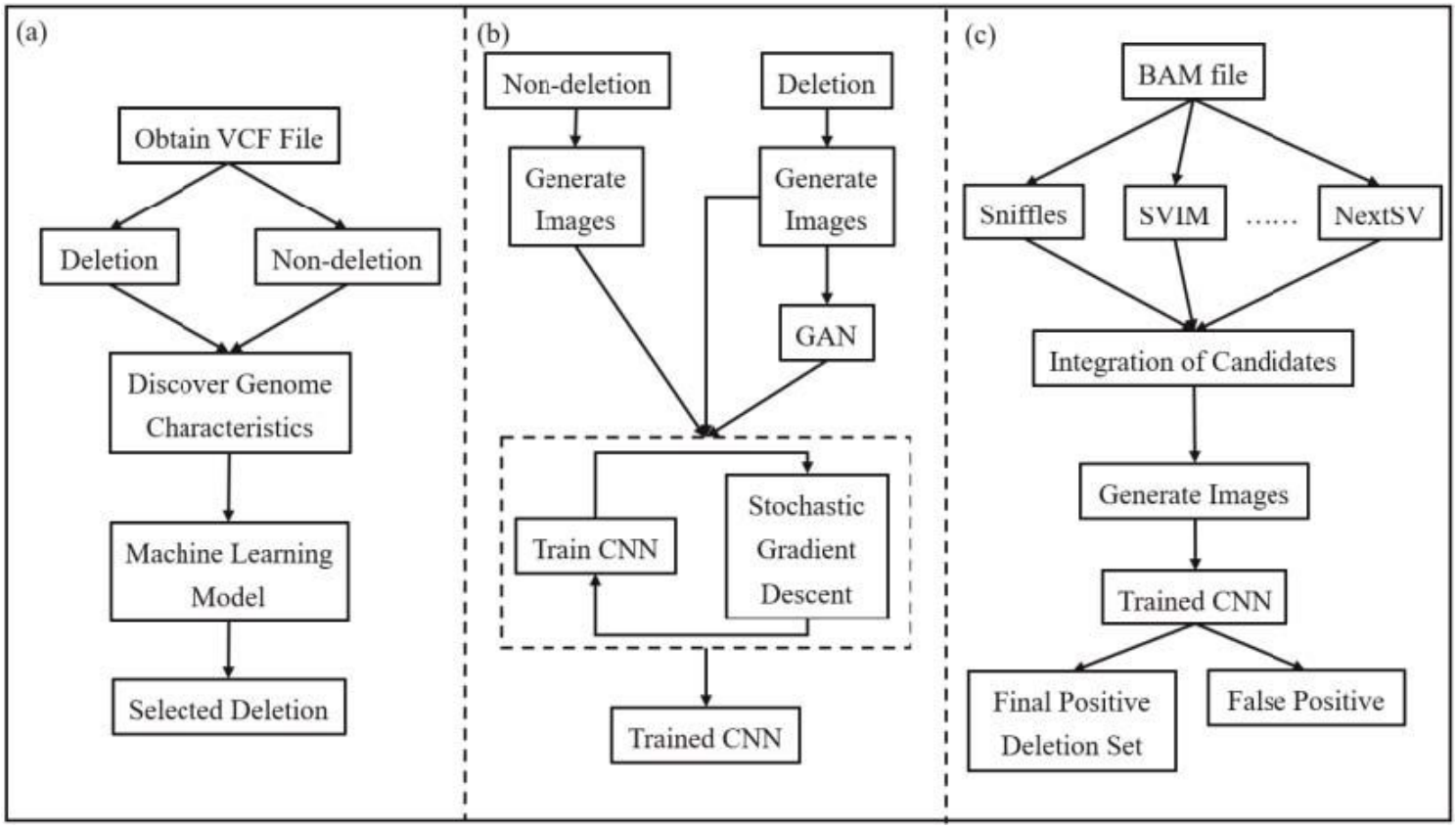


Figure 1

The overall framework of MaxDEL. (a). Obtaining features from the deletion and non- dele-tion regions, training a machine learning model, and using the trained machine learning model to filter breakpoint offsets to obtain the accurate deletion regions. (b) The deletion and non-deletion regions are generated into images. Due to the small number of deletions, we can obtain enough deletion imag-es by GAN and using these two genetic images to train CNN to get a trained CNN model. (c): Using existing tools to call variants to obtain a candidate deletion set. The trained CNN model is used to dis-criminate the candidate set and get the final deletion result. The trained CNN model will distinguish the candidate set and get the final deletion result.

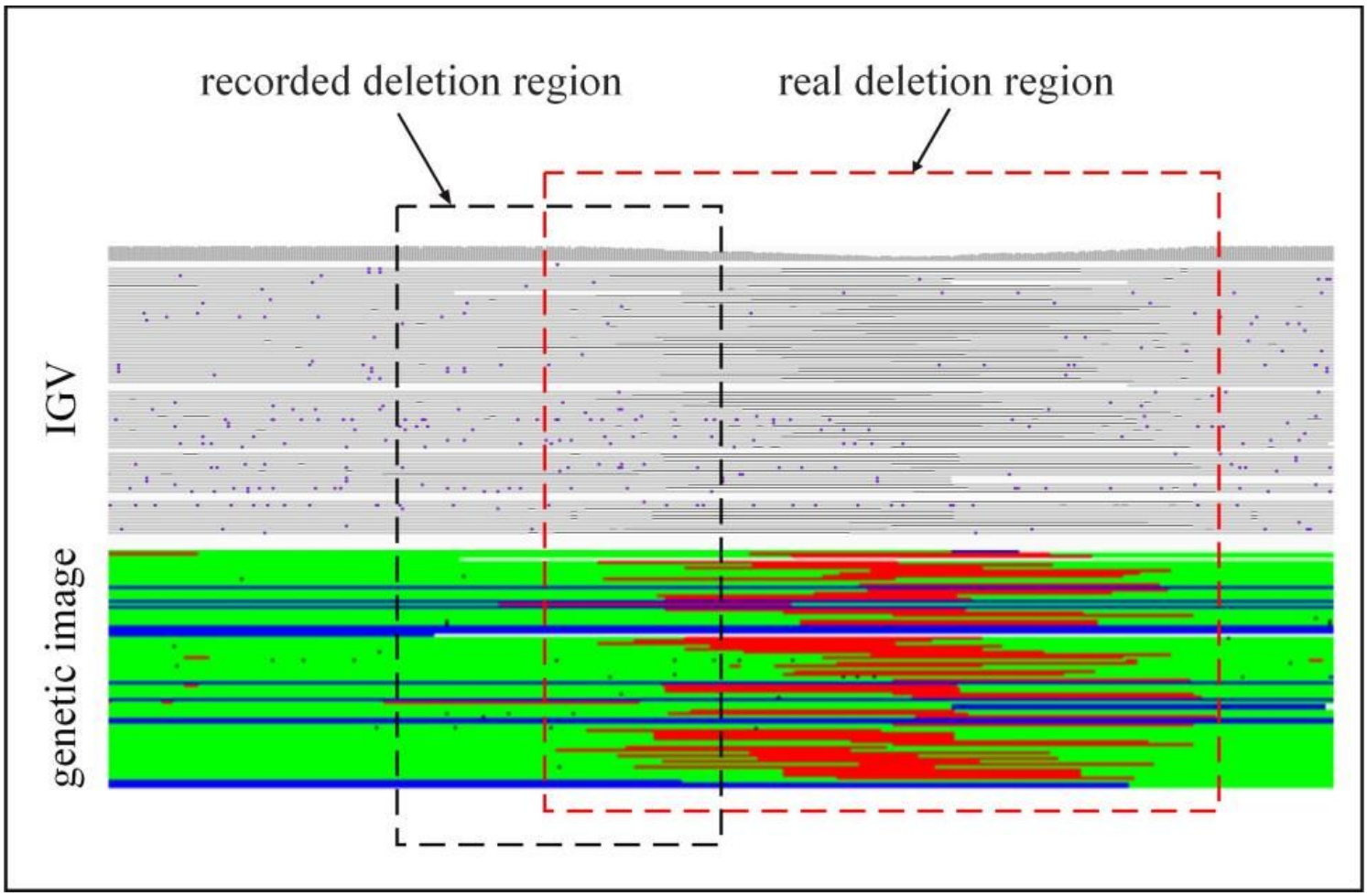


Figure 2

Visualization of genomic region and calibration result (chromo-some3,33799438-33799858). The upper part is the genomic region in IGV. The lower part is the image of the corresponding region with the image strategy of MaxDEL. The black part is the deletion region recorded in VCF, and the red part is the deletion region corrected by the random forest method.

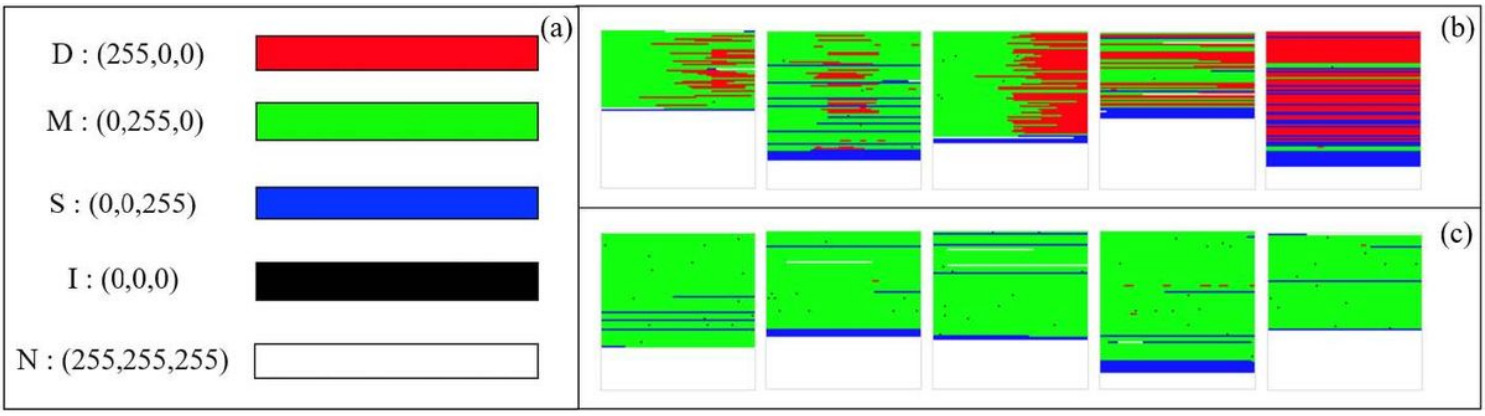


Figure 3

Image color distribution and image generation results. (a): Color assignment of genomic image. (b): Deletion region images. (c): Non-deletion region images.

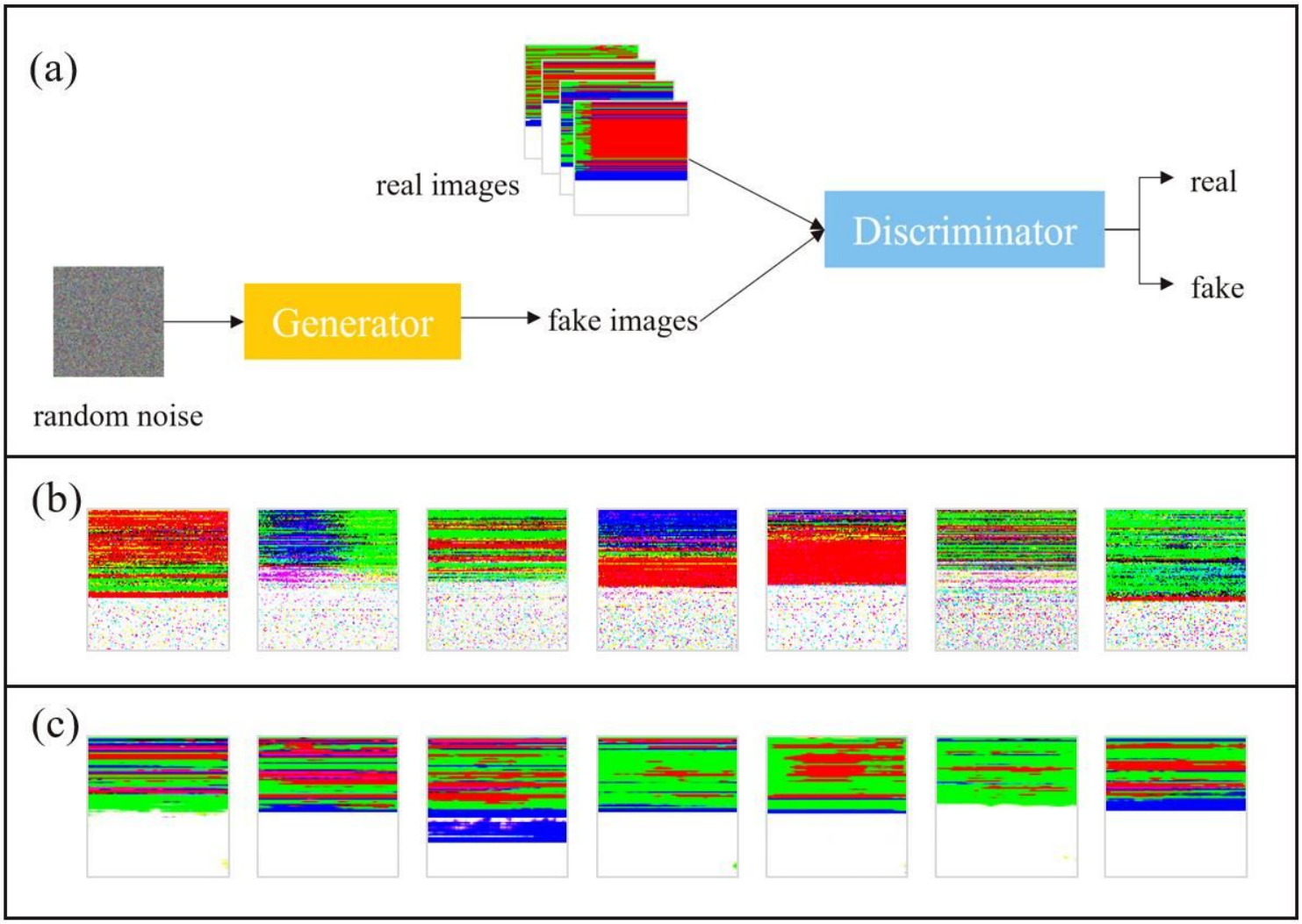


Figure 4

The structure of GAN and generated images of GAN methods. (a): General structure of GAN. (b) and (c): The generated images of original GAN and DCGAN.

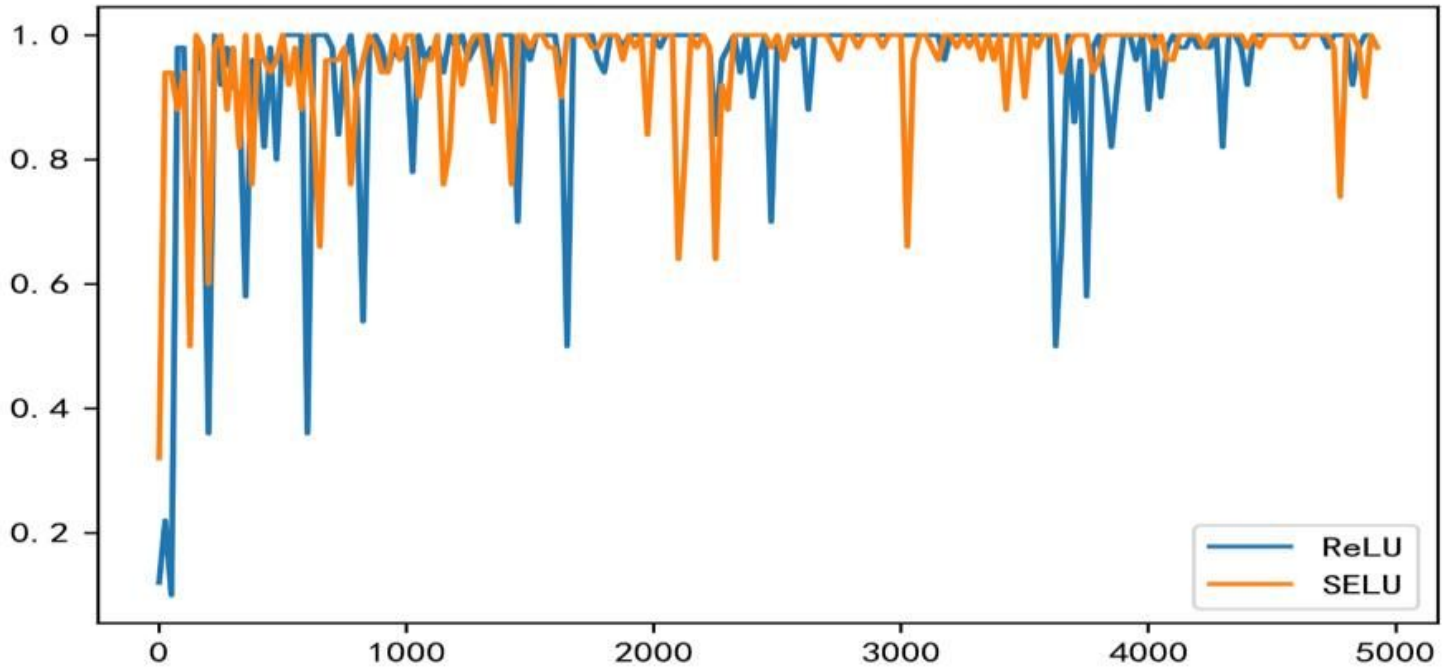


Figure 5

Accuracy comparison by using ReLU and SELU.

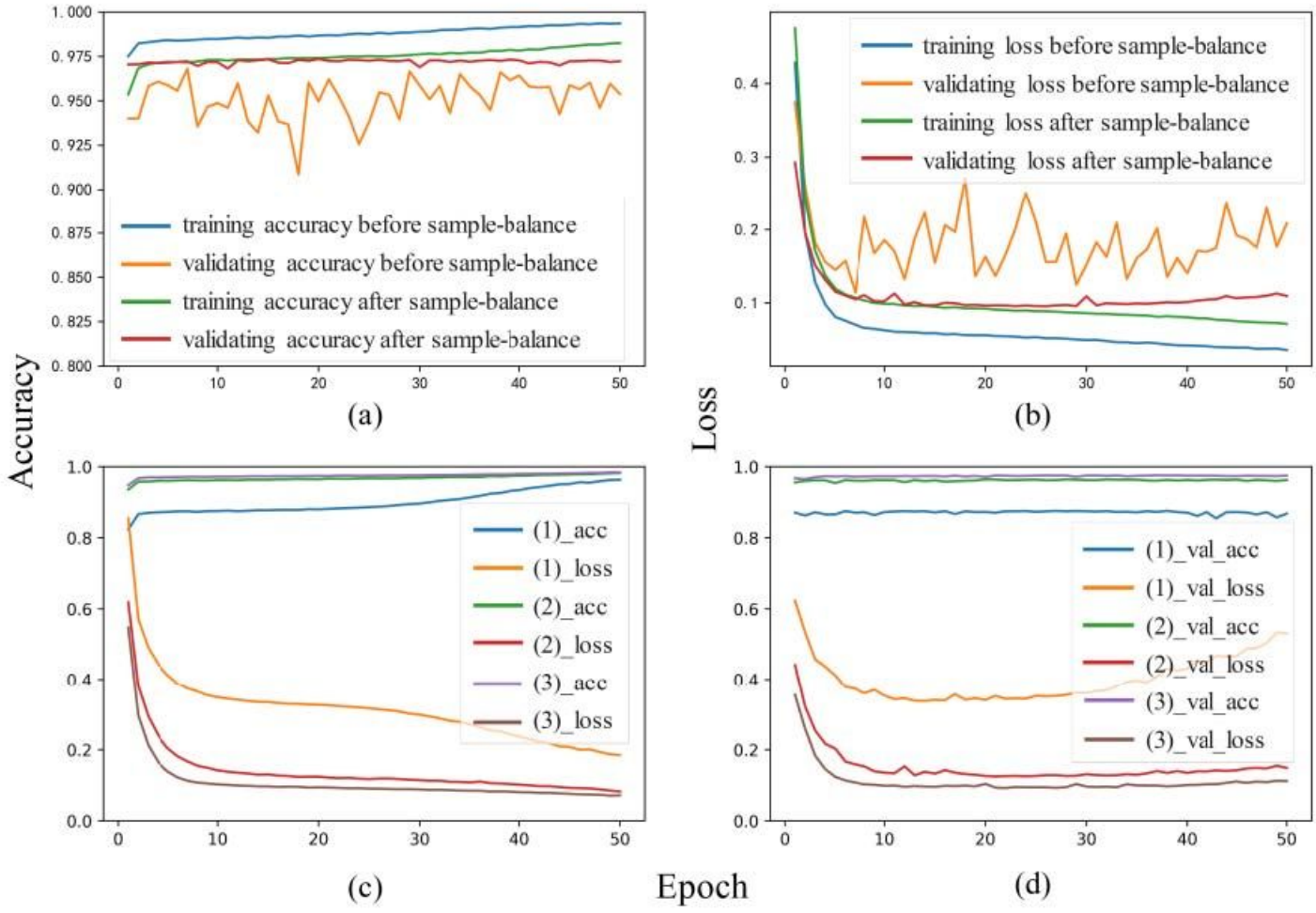


Figure 6

CNN's results in different data sets. (a): The difference of accuracy before and after sample-balance. (b): The difference of loss before and after sample-balance. (c) and (d): Accuracy and loss of different training data in the three groups. The three sets of data are described above.

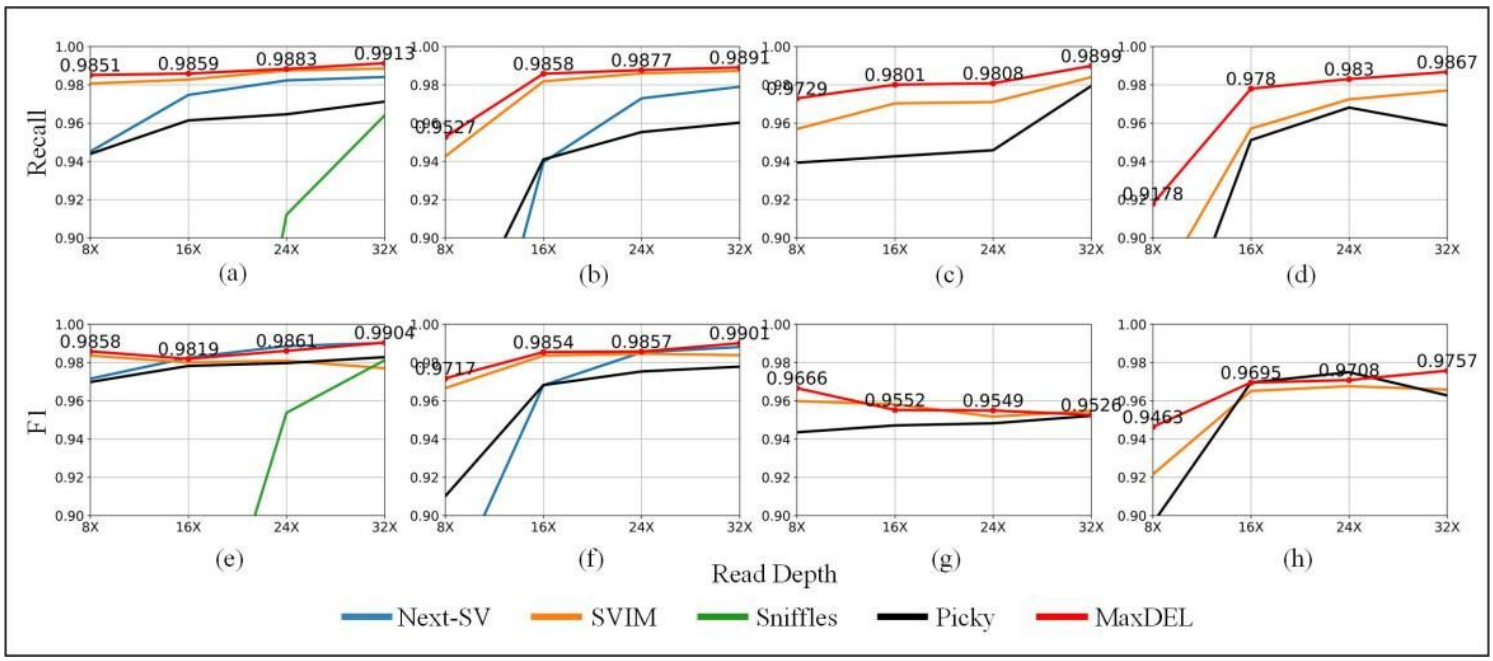


Figure 7

Comparison of existing tools and MaxDEL's recall and F1-score in different coverage depth data. The upper part is the recall and the lower part is the F1-score. The statistical range is (0.9,1). (a), (b) and (e), (f): category A, calling results of homozygous deletions and heterozygous deletions. (c), (d) and (g), (h): category B, calling results of homozygous deletions and heterozygous deletions.