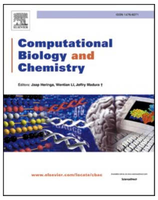
Computational Biology and Chemistry 94 (2021) 107417

Contents lists available at ScienceDirect

Computational Biology and Chemistry

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



Research Article

Cnngeno: A high-precision deep learning based strategy for the calling of structural variation genotype

Ruofei Bai, Cheng Ling, Lei Cai *, Jingyang Gao *

Department of Computer Science and Technology, Beijing University of Chemical Technology, Beijing, China

ARTICLE INFO

Keywords:
Genotype calling
Structural variations
Next-generation data
Convolutional neural network
Bootstrapping strategy

ABSTRACT

Genotype plays a significant role in determining characteristics in an organism and genotype calling has been greatly accelerated by sequencing technologies. Furthermore, most parametric statistical models are unable to effectively call genotype, which is influenced by the size of structural variations and the coverage fluctuations of sequencing data. In this study, we propose a new method for calling deletions' genotypes from the next-generation data, called Cnngeno. Cnngeno can convert sequencing data into images and classifies the genotypes from these images using the convolutional neural network(CNN). Moreover, Cnngeno adopted the convolutional bootstrapping strategy to improve the anti-noisy label's ability. The results show that Cnngeno performs better in terms of precision for calling genotype when compared with other existing methods. The Cnngeno is an open-source method, available at https://github.com/BRF123/Cnngeno.

1. Introduction

Emerging high throughput sequencing data highlight a change in focus from the small-scale sequencing to large-scale sequencing. A number of approaches for discovering structural variation (SV) from sequence reads have been developed in recent years. However, traditional calling tools cannot reached high accuracy on the SV genotype. Variant genotype detection is widely applied in disease diagnosis, such as loss of heterozygosity (LOH), breast and cervical cancer (Geng et al., 2017; Abyzov et al., 2011). Therefore, accurate calling of genotypes is tough problem especially on low coverage data with sequencing noise.

There are several state-of-the-art detection tools for calling deletion genotypes. For example, the CNVnator (Abyzov et al., 2011) detects genotypes by counting the read depth distribution of the samples on the reference genome. Pindel (Ye et al. (2009)) was originally a tool for variation detection, which was improved by adding depth analysis to confirm the genotypes. Dindel (Albers et al., 2011) can detect deletion genotype as well, but it focuses on the deletions of very small length, i.e. 30bp or less. LUMPY (Layer et al., 2014) is only a variation detection tool, thus the genotype results need to be further confirmed by SVTyper (Chiang et al., 2015). Genome STRiP (Handsaker et al., 2011) is mainly used to detect genotypes of long deletions, which has a poor effect on short deletions. In addition, Genome STRiP typically requires more than 20 whole-genome data to run, which consumes long time and is quite critical about the memory of the devices. Delly (Rausch et al., 2012) has poor performs on low coverage data. GINDEL (Chu et al., 2014), Concod (Cai et al., 2017) and the dudeML (Wang et al., 2017) are machine learning approaches with hand-crafted extracting features. CNNdel (Wang et al., 2017) is also based on the convolutional neural network (CNN). DeepVariant (Li and Durbin, 2009) transforms the mapped sequencing data into RGB images and uses CNN to train and classify these images. Its precision is higher than the mainstream indel calling software GATK (Cai et al., 2017), reaching 99.9%. But DeepVariant is only suitable for SNP and small-indel. DeepSV (Deep, 2020) is also based on image datas, but it piles up bases longitudinally to generate images, and it is unable to call deletion genotypes.

In this investigation, we propose Cnngeno, a deep learning based approach of deletion genotype calling. There are three main innovations to Cnngeno in comparison to traditional tools. First, Cnngeno can convert the sequencing data into the corresponding image. Second, Cnngeno adopts a novel method to compress the original images into a uniform images with fixed size. Finally, the Cnngeno construct the CNN model to test genotype using normalized images. In particular, these image datas can capture multiple information and integrate various signals in the sequencing data that are closely bound up with the deletion genotypes. Cnngeno's performance can not only surpasses other existing methods on high and low coverage data, but also has good ability to identify deletion genotypes of various lengths on complex real

* Corresponding authors.
E-mail addresses: cailei1512@qq.com (L. Cai), gaojy@mail.buct.edu.cn (J. Gao).

https://doi.org/10.1016/j.compbiolchem.2020.107417
Received 30 May 2020; Received in revised form 31 October 2020; Accepted 16 November 2020
Available online 21 November 2020
1476-9271/© 2020 Published by Elsevier Ltd.

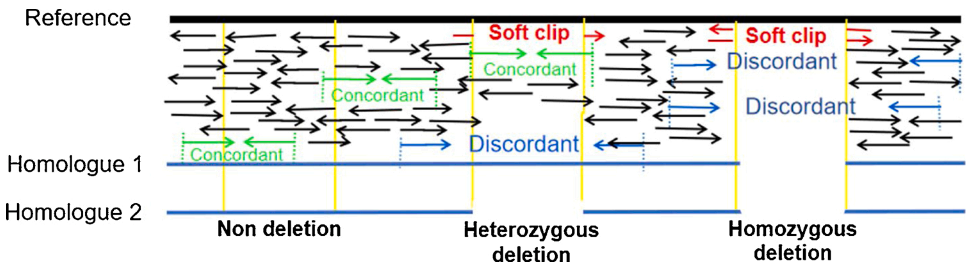


Fig. 1. Signals of deletion genotype calling.

data. Cnngeno has the following characteristics.

- Proposing a new visualization methods that can conver the genotype signals into feature images.
- Using the bootstrapping strategy to solve the noise problem in the real data and improve the model performance.

The experimental results suggest that Cnngeno surpasses Pindel (Albers et al., 2011), LUMPY + SVTyper (Chiang et al., 2015; Handsaker et al., 2011), Delly (Chu et al., 2014), CNVnator (Ye et al., 2009) and DINDEL (Cai et al., 2017) on real data in both precision and sensitivity, respectively. Meanwhile, Cnngeno can accurately call the structural variation genotypes of real data in comparison with the CNN method which is only applicable to simulation data.

2. Background

The human individual gene of diploid has three cases of alleles: the two alleles of homologous chromosomes are identical with the reference genome; only one allele in the homologous chromosome is different from the reference genome; both alleles of the homologous chromosomes have distinction with the reference genome. These cases are called non variation, heterozygous and homozygous variation respectively. Fig. 1 illustrates the non variation, heterozygous and homozygous examples of the deletion variations. Breakpoints refer to the coordinates on the genome, which can indicate where the deletion begin or end.

There are three signatures of genomic deletions when the sequences mapped onto the given reference genome near the deletion site. (i) Read depth. The depth of the sequences in the deletion region is much lower than that in the neighborhood without deletion variations. If a deletion is homozygous and the read is correctly mapped, the read depth in deletion region should be close to zero. If a deletion is heterozygote, the read depth in the deletion region should still be lower than expected. (ii) Discordant read pairs. Using tools such as BWA tools (Li and Durbin (2009)), paired-end sequences from an individual genome can be mapped to a given reference genome. If the mapping distance is greater than the acceptable upper limit of the insertion size of the library, there will be deletion variation in the genome region. (iii) Soft-clip read (Cai et al. (2017)). When a read spans a complete deletion, it will contain two fragments: one from the region before the deletion and the other from the region after the deletion, called split-read. When one read aligning the reference genome exists the split-read situation, it turns out that the variation occurs on this region.

3. Methods

Cnngeno calls genotypes of deletion mainly in three steps: converting to image data, image compression and genotype calling based on bootstrapping strategy. In the first step, Cnngeno converts sequencing texts of candidate variations to their candidate image datas. The sequencing data features are extracted from BAM files and transformed these features into image datas according to the conversion rules. In the second step, these transformed images are compressed to a uniform size, which satisfied the requirement of the subsequent deep learning networks. In the last step, these compressed images are used to train the CNN based on bootstrapping strategy. Then we use the final trained CNN model to classify the genotypes of the image datas.

3.1. Converting sequencing data features into images

How to convert the sequencing data features into the images is the key point of calling the structural variation. The process from sequencing to the images involves two aspects about read design and image design.

3.1.1. Read design

The background of the overall picture was set to white. The BAM file header had 11 fields, and each field represented distinct variant features. For one alignment read, we concern the CIGAR field (the sixth filed in the BAM file), alignment quality of the base (the fifth field, eleventh column field) and the distance of paired-end read ISPE (the ninth field) in its BAM file. The conversion rules is as following:

- When the CIGAR field is "S", it indicates a soft clip read and the read is designed red with RGB value (255, 0, 0).
- When ISPE is extremely large, the read is designed blue and RGB value is (0, 0, 255).
- When the CIGAR field is "M" and ISPE is normal, the read is designed green, and RGB value is (0, 255, 0).

Cnngeno firstly filters out the reads with low alignment quality, and then converts the reads to image datas according to the conversion rules. When the alignment quality is low, the two values with 0 in the tuple of RGB value are duly increased, which makes the image color become visually shallow.

3.1.2. Image design

In this section, we introduced the layout of the feature images. In the feature image, the horizontal coordinate represents the reference gene and the vertical coordinate represents the read depth. The height of the image are set to be equal the average depth of the BAM files. Each pixel represents one base of a read. We adopted the pileup strategy to arrange the reads. Cnngeno fetched all reads from the BAM file and sorts them by the starting coordinates. Each read was arranged the in the image from top to bottom according the starting coordinates. Figure 2 shows the feature image arrangement.

3.2. Image compression

Because different lengths of human deletions vary from 50bp to over thousands of bases, the imageimage compression is used to ensure that the deletion images are uniform in size. The down-sampling process is necessary because the subsequent deep learning model requires the size of input images to be consistent.

The generated images are named as original images, and the images

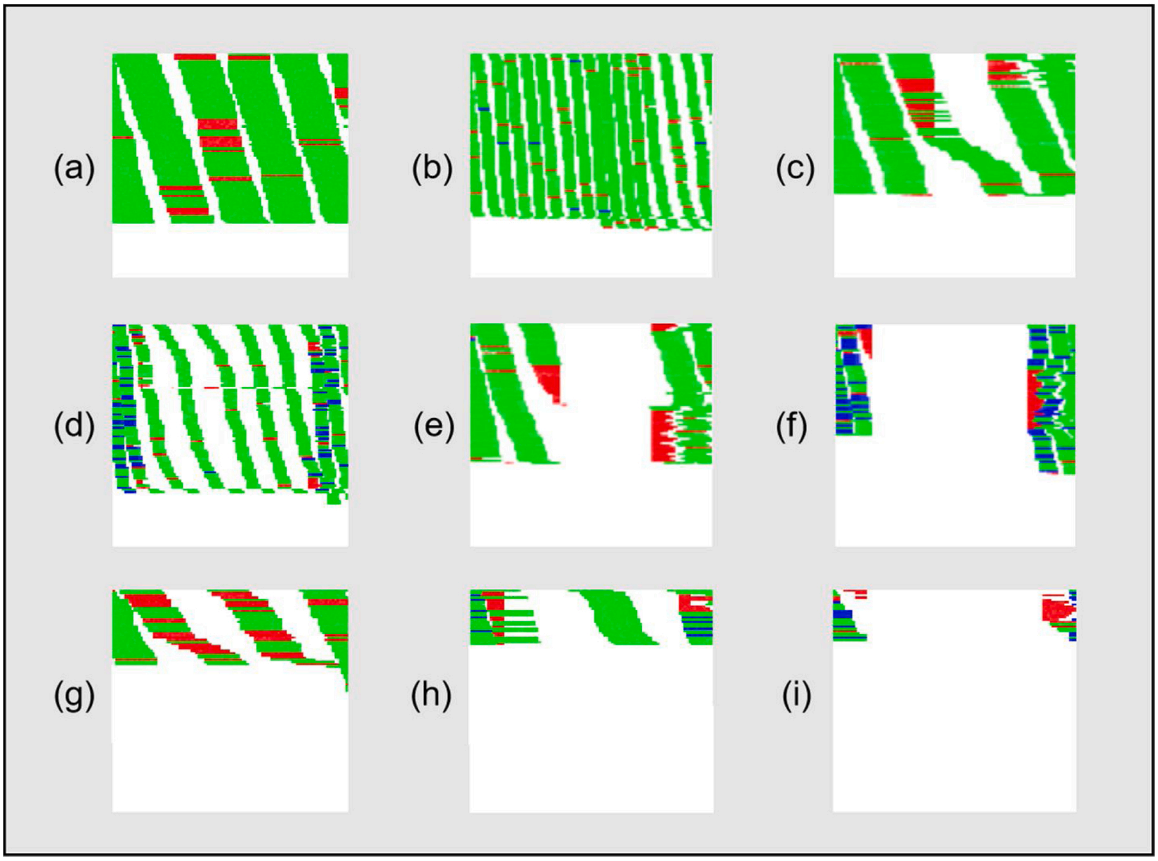


Fig. 2. Visualization of deletion regions: (a) shorter non deletion region of higher read depth, labeled as 0; (b) longer non deletion region of higher depth, labeled as 0; (c) shorter heterozygous deletion region of higher depth, labeled as 1; (d) longer heterozygous deletion region of higher depth, labeled as 1; (e) shorter homozygous deletion region of higher depth, labeled as 2; (f) longer homozygous deletion region of higher depth, labeled as 2; (g) shorter non deletion region of lower read depth, labeled as 0; (h) shorter heterozygous deletion region of lower depth, labeled as 1; (i) shorter homozygous deletion region of lower depth, labeled as 2.

after down-sampling compression are named as destination images. The candidate deletion regions are put at the center of the images, and the images also contain the reads outside the two candidate breakpoints. In the process of down-sampling compression, the number of horizontal pixels is unchanged, while vertical pixels are compressed.

We assume that the number of vertical pixels of the original images is N, and that of the destination images is n. The size of the down-sampling window is 1* (N/n), and N is set as the number that can be divisible by n. The pixels in each window are compressed according to the compression rules. The rules make the pixels are compressed into a white pixel if all pixels in the same window are white. Otherwise the pixels of each color (red, green, blue, white) are counted separately. The images after compression are shown in Fig. 2.

3.3. Genotype calling based on bootstrapping strategy

Cnngeno is a novel genotype calling approach based on deep learning method. In supervised learning, the training dataset and the labels are significant for deep learning model. In this section, we construct the training dataset and use the anti-noise label to train the CNN model.

3.3.1. Construction of Benchmark using bootstrapping strategy

CNNs are trained on three types of images: (i) images that contain the whole regions of homozygous variants are labeled as 2; (ii) images that contain the whole regions of heterozygous variants are labeled as 1; (iii) images that only contain normal regions without variants are labeled as 0. The deletions and their genotypes are gathered from the BAM files according to the VCF flies.

However, it is a biggest challenge that there is no benchmark in the development of genotype calling tools. For simulation data, the deletion region and breakpoints is accurte.. However, the variant information of real data have a high proportion of mistaken using IGV[20] to verify, which is because the current VCF files of the 1000 genome project (1000 Genomes Project Consortium, 2010) are merely a combination of the several existing tools results . Due to the inaccurate genotype labels of real data, the classifier network cannot train real data with VCF flies directly.

3.3.2. Convolution-bootstrapping algorithm anti noisy label

Cnngeno uses the bootstrapping algorithm to speculate the noise in genotype of the real data. First of all, the absolutely correct seeds are selected artificially as training set, which are trained until the model's convergence or maximum number of iterations. In this paper, the initial seeds are selected by hand based on images. Then the remaining samples are classified and these samples with higher classification confidence are selected into the seed set. In this paper, the classification confidence is set at 0.8. The process of expanding the seed set is repeated until no new seeds are added to the training. The final trained model is used to classify the genotypes. For samples with a large number of noisy labels, since the training of the initial seed set is a supervised learning process, the subsequent sample labels are assigned to a higher confidence classification in comparison with the labels specified in the original set of data. The statistic of this bootstrapping method is effective on reducing the noise disturbances to classifiers (Freedman, 1981).

3.3.3. CNN model architecture

The proposed CNN architecture of Cnngeno contains eight layers,

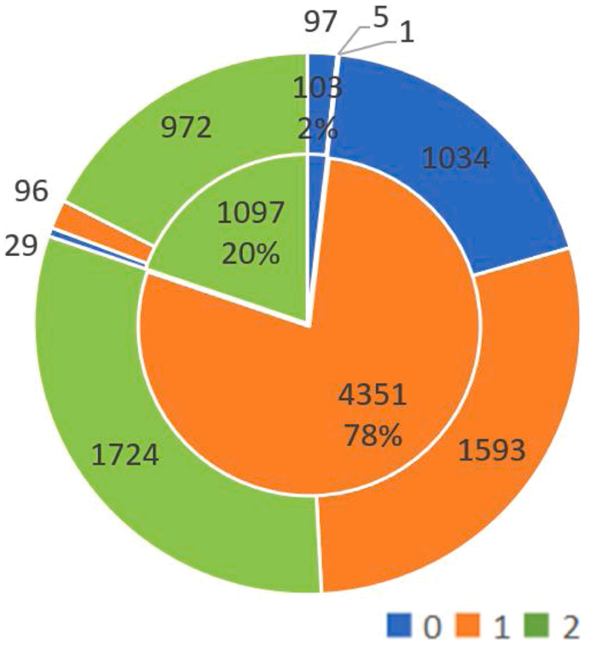


Fig. 3. The predicted results of dataset 1 after training CNN with dataset 3.

and it has three convolutional layers, two max pooling layers, one fully connected layer, one input layer and one output layer. The filter sizes of the three convolutional layers are 3 × 3 and the max pooling layers are 2 × 2. The output sizes of the three convolutional layers, two max pooling layers, one fully connected layer, one input layer and one output layer are 98 × 98, 96 × 96, 46 × 46, 48 × 48, 23 × 23, 1 × 16, 100 × 100 and 1 × 3 respectively. Through the experiments, the activation function is selected as RReLU and the dropout rate and learning rate are set 0.25 and 0.001 respectively. Compared with Sigmoid and tanh activation function, RReLU has a less computational cost and can accelerate the model convergence.

3.3.4. Genotype calling

The trained model are used to call deletion genotypes on the test data. The calling process were divided three steps. First, Cnngeno extracted the sequencing data featrures from BAM file and converted these features into images. Then these images are compressed into the fixed size using down-sampling strategy. Finally, Cnngeno use these images to train the CNN model, and verify the performance of calling deletion genotype. The Cnngeno's output has three states of homozygous deletion, heterozygous deletion and non-deletion.

4. Results

Cnngeno is evaluated on simulative data and real data for calling deletion gentoypes compared with the existing tools, including Pindel, LUMPY + SVTyper, Delly, CNVnator and DINDEL.

4.1. Experiment data

4.1.1 Simulative dataset Simulative datasets with high coverage are used, which contains deletions of various length ranges. The simulative data is 80× coverage. There are 2196 homozygous deletions, heterozygous deletion and non-deletion regions, respectively. 4.1.2 Real dataset *DatasetI:* are from 74 autosomal VCF files of 1000 genomes project, the records labeled as heterozygous and homozygous deletion are 4351 and 1097 respectively, the non-deletion records are picked from non-deletion regions adjacent the deletion region. *DatasetII:* is hand-selected (make sure the labels accurate) 300 regions including 103 non deletion, 100 heterozygous and 97 homozygous deletions, which are chosen from*datasetI*, and regarded as correct genotypes on the IGV by artificial judgment, the non deletion data is derived from *datasetI* totally. *DatasetIII* is obtained by *datasetII* through data enhancement. Data enhancement is usually applied in situations where the training dataset is limited in image size or the classifications are unbalanced. The *datasetII* are amplified by 10 times by means of translation transformation and pixel noise perturbation. The records of non deletion, heterozygous and homozygous deletion are respectively 1133, 1100 and 1067.

4.2. Performance on Different ReLU Functions 4.3 Experiments on Different Dropout Value 4.4 Experimental Results and Analysis of Anti Noise Label on CNN 4.5 Estimation of genotype noises for 1000 genomes project

- To evaluate the calling results about bootstrapping strategy, we used *Dataset III* (without anti-noise label) and *Dataset I*(with anti-nosie labels) as the test set. Test on the *Dataset III*: An ordinary convolution neural network is used to train 3300 images from *datasetIII* approximately without noisy labels. The proportion of three labels is about 1:1:1. The results of the test are shown in Fig. 3. Fig. 3 inferences that there are approximately 24 % non deletions and 40 % homozygous deletions, 9% heterozygous deletions Due to the high proportion of noisy labels and the factor of class imbalance, the training cannot reach convergenceon real data. The true distribution of genotype is estimated to be 1160:1694:2697 (approximately 1:1.5:2.3).
- Test on *dataset I*: The convolution-Bootstrapping strategy is used to train the *datasetI*. The initial seed was set 300 and the remaining 5251 images of *dtaset I* were tested. There are 5551 candidate variants containing seed sets are detected, as shown in Fig. 7(b).

Fig. 7(a) The predicted results of *dataset I* after training CNN. (b) The predicted results of *dataset II*after training CNN based on bootstrapping strategy. The prediction results in Fig. 7(a) are similar with Fig. 7(b). On the basis of the two experimental results on 1000 genomes project, it is speculated that the bootstrapping strategy can improve the anti-noise performance.

4.3. Performance of genotype calling on high coverage trio data

In order to evaluate the performance of Cnngeno on real data, experiments are carried out on a set of high coverage data published by 1000 genomes project. The three individuals used in this experiment are NA19238 (mother), NA19239 (father) and NA19240 (daughter), and the average coverage of each individual is about 70 × . We compared the Cnngeno with state-of-the-art methods, including CNVnator, Pindel, LUMPY+SVTyper and Delly. We choose 50 called deletions to further analyze the genotypes and verify the false positives on each tool. To evaluate the performance of these tools on different coverage data, we down-sample 70× raw data and generate two datasets with average coverage of 20× and 10×, respectively. Three criteria defined below are used to evaluate the performance of the six tools.

- The number of deletions violating Mendel's law of inheritance or not. For example, if both parents have a deletion genotype of 2, their children should have the same deletion and genotype. There are eight cases that do not follow Mendel's law of inheritance: (0, 0, 1), (0, 0, 2), (0, 1, 2), (0, 2, 0), (0, 2, 2), (1, 2, 0), (2, 2, 0), and (2, 2, 1). The first two elements represent the parents' genotypes and the third one is the children's genotypes. These conditions indicate that there is a high probability of error detection for at least one of the three genotypes.

Table 1
Comparisons of CNVnator, Pindel, LUMPY + SVTyper, Delly, GINDEL and Cnngeno on high coverage data of three individuals with down-sampling.

Coverage	Criteria	CNVnator	Pindel	LUMPY + SVTyper	Delly	GINDEL	**Cnngeno**
100 % (70×)	#inconsistent deletions	3(50)	1(50)	1(50)	2(50)	1(50)	**1(50)**
	#deletions with all-0 genotype	2(50)	2(50)	1(50)	3(50)	2(50)	**0(50)**
30 % (20×)	#inconsistent deletions	5(27)	14(44)	7(43)	5(41)	5(47)	**0(50)**
	#deletions with all-0 genotype	2(27)	7(44)	10(43)	10(41)	7(47)	**1(50)**
	#changed genotypes in down-sampling	21(81)	46(132)	21(129)	33(123)	17(141)	**11(150)**
15 % (10×)	#inconsistent deletions	4(17)	6(37)	11(37)	6(34)	11(43)	**3(50)**
	#deletions with all-0 genotype	3(17)	23(37)	19(37)	9(34)	9(43)	**1(50)**
	#changed genotypes in down-sampling	29(51)	82(111)	39(111)	69(102)	29(129)	**16(150)**

Table 2
Comparisons of CNVnator, Pindel, LUMPY + SVTyper, Delly, GINDEL and Cnngeno on CNV data.

Tools	500−1000bp			1000bp–5000bp			5000bp–10000bp			> 10000bp		
	P	S	F	P	S	F	P	S	F	P	S	F
CNVnator	53.3	50.0	51.6	59.2	60.0	59.6	63.6	66.7	65.1	63.6	70.0	66.7
Pindel	91.7	68.8	78.6	89.2	89.2	89.2	88.9	76.2	82.1	88.9	70.0	78.3
LUMPY + SVTyper	92.9	81.3	86.7	95.5	84.0	89.4	100.0	90.5	95.0	90.0	90.0	90.0
Delly	83.3	62.5	71.4	94.0	62.7	75.2	94.1	76.2	84.2	88.9	80.0	84.2
GINDEL	93.8	93.8	93.8	89.3	100.0	94.3	95.0	90.5	92.7	100.0	90.0	94.7
Cnngeno	**100.0**	**100.0**	**100.0**	**98.7**	**100.0**	**99.3**	**95.5**	**100.0**	**97.7**	**100.0**	**90.0**	**94.7**

- A deletion with all genotypes 0 indicates genotype error at least one of a family members, when the deletions are called by four tools on high coverage data.
- The number of genotypes that changes when data coverage is reduced. In comparison to the performance on low coverage data, all tools should perform better on high coverage data. Hence, these genotypes are likely to be wrong when the variant genotypes are undetectable after down-sampling.

Table 1 shows the results of CNVnator, Pindel, LUMPY + SVTyper, Delly, GINDEL and Cnngeno on high coverage data. The performance of these tools are similar when the coverage is 70 × . There are no more than three deletions that do not conform to Mendel’s law of inheritance . In other words, all the six tools perform well in high coverage data. However, the performance of almost all tools decreases with the coverage decreases. For example, there are 14(44) deletions detected by Pindel violate Mendel’s law of inheritance, and 10(41) deletions detected by Delly are all 0 genotypes on coverage 20 × . In contrast, Cnngeno performs better when coverage reduces. In the worst case, only three deletions violate Mendel’s law of inheritance. In addition, the genotype detected by Cnngeno is more consistent when the coverage is decreased from 70 to 10×: the change rate of genotype detected by Cnngeno is 10.67 %. The corresponding value of CNVnator, Pindel, LUMPY + SVTyper, Delly and GINDEL is 56.86 %, 73.87 %, 35.14 %, 67.65 % and 22.48 %, respectively.

4.4. Performance on the CNV dataset

The experiment uses the dataset released by the 1000 genome project. The data consists of 96 BAM files from 48 individuals (chromosomes 11 and 20 for each individual). Among the 48 individuals, 10 are from CEU population, 35 are from YRI population and 3 are from CHB + JPT population. These data are mapped to NCBI37 using BWA tools to generate BAM data. These CNVs were validated by array-CGH using a set of NimbleGen, so these genotypes are accuracy as benchmarks (Conrad et al., 2010). Most of these deletions are quite long, only 16 are ranging from 450 to 960bp, and others are longer than 1kbp. The smallest deletion is 450bp, and the largest is 88,384bp. The average size of the deletion is 2461bp. Table 2 shows the results of each tool on different deletion sizes, with an average data coverage of 60 × . It is obvious that Cnngeno is superior to other tools on these long deletions, with precision from 95.5–100% and sensitivity from 90 to 100 %.

5. Conclusions

In this paper, we define and elaborate a novel method, Cnngeno, for calling deletion genotype using deep learning network from next-generation sequencing data. The experimental results illustrate that Cnngeno, performs better than the current state-of-the-art methods on complex real data with high and low coverage. The proposal of converting the sequencing data features into image datas and integrating genotype signals are the key contributions of this work. Besides, down-sampling compression strategy is adopted, which normalized the image inputs as fixed size for subsequent networks. Especially, the bootstrapping strategy were used to improve the recognized ability of anti-nosisy labels. The experimental results also indicate that Cnngeno is able to call a wider length range of deletion genotypes. However, the sensitivity of longer deletion calling still needs to be improved due to the limitation of the length of sequencing fragments of the next-generation data. Future investigations will focus on how to improve the sensitivity on longer deletion.

CRediT authorship contribution statement

Ruofei Bai: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Cheng Ling:** Writing - review & editing, Visualization. **Lei Cai:** Project administration, Writing - review & editing. **Jingyang Gao:** Resources, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Project supported by Beijing Natural Science Foundation (5182018).

References

Abyzov, A., Urban, A.E., Snyder, M., et al., 2011. CNVnator: an approach to discover, genotype and characterize typical and atypical CNVs from family and population genome sequencing. Genome Res. 21 (6), 974.

Albers, C.A., Lunter, G., MacArthur, D.G., McVean, G., Ouwehand, W.H., et al., 2011. Dindel: accurate indel calls from short-read data. Genome Res. 21, 961–973.

Cai, Lei, Chu, Chong, Zhang, Xiaodong, Wu, Yufeng, Gao, Jingyang, 2017. Concod: an effective integration framework of consensus-based calling deletions from next-generation sequencing data. Int. J. Data Min. Bioinform. 17 (2).

Chiang, C., Layer, R.M., Faust, G.G., et al., 2015. SpeedSeq: Ultra-fast personal genome analysis and interpretation. Nat. Methods 12 (10), 966.

Chu, C., Zhang, J., Wu, Y., 2014. GINDEL: accurate genotype calling of insertions and deletions from low coverage population sequence reads. PLoS One 9, e113324.

Deep, S.V., 2020. Accurate Calling of Genomic Deletions from High-throughput Sequencing Data Using Deep Convolutional Neural Network, bioRxiv Preprint. https://doi.org/10.1101/561357.

Freedman, D.A., 1981. Bootstrapping regression models. Ann. Stat. 9 (6), 1218–1228.

Handsaker, R.E., Korn, J.M., Nemesh, J., et al., 2011. Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. Nat. Genet. 43 (3), 269–276.

Layer, R.M., Chiang, C., Quinlan, A.R., et al., 2014. LUMPY: a probabilistic framework for structural variant discovery. Genome Biol. 15 (6), R84.

Li, H., Durbin, R., 2009. Fast and accurate short read alignment with burrows–wheeler transform. Bioinformatics 25, 1754–1760.

Rausch, T., Zichner, T., Schlattl, A., et al., 2012. DELLY: structural variant discovery by integrated pairedend and split-read analysis. Bioinformatics 28 (18), i333.

Wang, Jing, Ling, Cheng, Gao, Jingyang, 2017. CNNdel: calling structural variations on low coverage data based on convolutional neural networks. Comput. Biomed. Res. 2017, 6375059.

Ye, K., Schulz, M.H., Long, Q., et al., 2009. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertionsfrom paired-end short reads. Bioinformatics 25 (21), 2865.

1000 Genomes Project Consortium, 2010. A map of human genome variation from population-scale sequencing. Nature 467 (7319), 1061–1073.