Mathematical Biosciences and Engineering

MBE, 17(1): 202–215.
DOI: 10.3934/mbe.2020011
Received: 28 April 2019
Accepted: 17 September 2019
Published: 30 September 2019

http://www.aimspress.com/journal/MBE

Research article

DL-CNV: A deep learning method for identifying copy number variations based on next generation target sequencing

Yunxiang Zhang[1], Lvcheng Jin[2], Bo Wang[3], Dehong Hu[1], Leqiang Wang[1],Pan Li[3], Junling Zhang[3],Kai Han[3], Geng Tian[3], Dawei Yuan[3,*], Jialiang Yang[3,*], Wei Tan[1,*], Xiaoming Xing[4,*] and Jidong Lang[3,*]

[1] Weifang People's Hospital, Guang Wen Road, Weifang 261000, China
[2] Weifang Medical University, Bao Tong West Street, Weifang 261053, China
[3] Geneis Beijing Limited Company, Beijing 100102, China
[4] The Affiliated Hospital of Qingdao University, Jiang Su Road, Qingdao 266071, China

* **Correspondence:** Email: yuandw@geneis.cn, yangjl@geneis.cn, tanwei67@126.com, edithxing@126.com, langjd@geneis.cn.

Abstract: Copy number variations (CNVs) play an important role in many types of cancer. With the rapid development of next generation sequencing (NGS) techniques, many methods for detecting CNVs of a single sample have emerged: (i) require genome-wide data of both case and control samples, (ii) depend on sequencing depth and GC content correction algorithm, (iii) rely on statistical models built on CNV positive and negative sample datasets. These make them costly in the data analysis and ineffective in the targeted sequencing data. In this study, we developed a novel alignment-free method called DL-CNV to call CNV from the target sequencing data of a single sample. Specifically, we collected two sets of samples. The first set consists of 1301 samples, in which 272 have CNVs in *ERBB2* and the second set is composed of 1148 samples with 63 samples containing CNVs in *MET*. Finally, we found that a testing AUC of 0.9454 for *ERBB2* and 0.9220 for *MET.* Furthermore, we hope to make the CNV detection could be more accurate with clinical "gold standard" (e.g. FISH) information and provide a new research direction, which can be used as the supplement to the existing NGS methods.

Keywords: copy number variation; next generation sequencing; deep learning; convolutional neural network; target sequencing

1. Introduction

Copy number variations (CNVs) are commonly repeated regions on human genomes that vary between individuals [1]. CNVs may be caused by non-allelic homologous recombination (NAHR) at highly similar sequences [2]. Genomics research has shown that approximately two-thirds of the whole human genome is composed of repeats [3] and 4.8–9.5% of the whole genome can be classified as CNV [4]. CNV plays an important role in generating necessary variations. Thus, CNV detection is integral to the detection of many diseases [1].

Currently, the "gold standard" method for detecting CNVs in clinical settings is fluorescent in situ hybridization (FISH) [5]. Comparative genomic hybridization is also commonly used [5]. One major drawback of these techniques is that the genomic resolution can be as low as 40 kb [6], meaning that only large repeats such as whole gene repeats can be detected. For the detection of small CNVs, next-generation sequencing has been applied over the last 10 years [7,8]. The most popular software used in the detection of CNVs, including CNVkit [9], CNVnator [10] and Control-FREEC [11] are based on whole genome sequencing, whole exome sequencing or target sequencing to detect CNVs. For CNV detection in single samples, sequencing depth and GC content have been used to rectify the results. As the amount of targeted sequenced samples has increased, CNV detection methods mainly concerning the target area, such as CNV-RF [12], PatternCNV [13] and Ioncopy [14] have emerged. For example, CNV-RF [12] utilizes the NGS to detect deletions as small as 180 bp and duplications as small as 300 bp.

However, the existing methods all require alignment and heavily rely on the sequencing depth and GC content, with most works concentrating on the detection of CNVs at the level of the whole genome. Additionally, these methods are restricted to the alignment parameters and their hand-crafted parameters, which lead to complicated procedures and high costs in the detection of CNVs for a single sample. Moreover, for the targeted sequencing of data, some of those methods are not very effective.

Our study demonstrated that the convolutional neural network (CNN) could be used to detect CNVs from the matrix through a basic operation based on raw data. In the fields of artificial intelligence, CNN first proved useful in image recognition [15], and it was afterward used in life science [16]. Recently, a group from Google implemented an algorithm called "deepvariant" to form images through sequence-alignment results. By feeding the image into a well-designed CNN, the group managed to detect a single nucleotide polymorphism (SNP) and a small indel variant [17]. However, due to the window size and the implementation of deepvariant, it is very expensive to use the same algorithm for the detection of large indels, also known as CNVs. To address the need for quick CNV detection, we wanted to develop a novel, alignment-free algorithm that could detect patients carrying CNVs at a high accuracy for a non-paired sample.

2. Results

2.1. An alignment-free deep learning pipeline to call CNV from single sample

In this study, we developed a novel method called DL-CNV to call CNV from the sequencing data of a single sample. Specifically, we first collected 1301 sample dataset with 272 *ERBB2*-amplification samples and 1148 sample dataset with 63 *MET*-amplification samples for the

subsequent analysis. *ERBB2*, also known as HER2, is the target of the monoclonal antibody trastuzumab (marketed as Herceptin) [18]. Detecting the amplification of *ERBB2* is vital to targeted therapy [19]. The frequency of the CNV of *ERBB2* was around 16% [20]. *MET* is a receptor tyrosine kinase (RTK) considered to be a druggable target in non-small cell lung cancer (NSCLC) [21]. The frequency of the CNV of *MET* was around 2% [22]. Two independent sample sets for two genes were separately trained and tested. The results showed the possible application of deep learning methods in detecting CNVs (Figure 1).

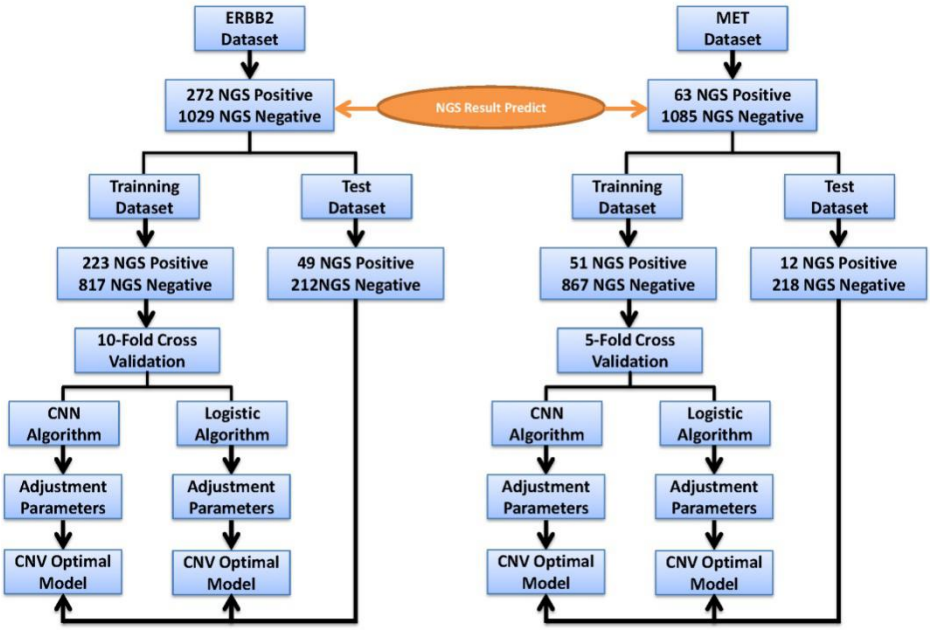


Figure 1. A schematic view of the procedures for model generation for *ERBB2* and *MET*.

A conceptual overview of DL-CNV is shown in Figure 1. As for the labels, the widely used software Ioncopy was fine-tuned using previously tested data (data not shown) to make sure the result will be highly similar to the result generated by FISH. And then, the fine-tuned Ioncopy was used to output the labels of each sample. We generated the windows of the reference exon sequence split by 50 bp with a stride of 40 bp, and the windows of all exons of a specific gene were then piled up. Afterward, we generated matrices from different raw sequencing data for different samples that represented the window-wise read depths; the read depths were in the same order as the reference windows. Normalization was done to balance the factor of the whole genome sequencing depth. The learned CNN model could intuit the local features from a matrix and generalize them across different matrices from different samples (Figure 2). The training set and the test set were split according to the year of the sequenced samples.

Figure 2a showed the procedures used to prepare the coverage matrices. First, exons from one gene were extracted. Then, each exon was separated by 50-bp windows with 10-bp overlaps. The read number that covered the window was counted and then used to generate a list of read numbers for the current exon. After counting each exon, lists of read numbers were piled up to form a matrix. The elements corresponding to shorter exons were filled with zeroes. The matrix was used for the

models as a training or test sample. The CNN model was used for the classification of CNV of the matrix. Figure 2b showed the CNN architecture. The heatmap on the left is an example for the sample with brightness, which indicates the read number after the normalization. The CNN contained one convolutional layer, one max pooling layer, one dropout layer, and two fully connected layers. The kernel used in the convolutional layer was set at 5 × 5. The kernel used in the max pooling layer was set at 2 × 2. The fully connected layers were of the sizes 1024 and 2, in which the second layer was the output layer that classified whether the sample was CNV positive. The dropout layer whose dropout value was set 0.5 was between the two fully connected layers.

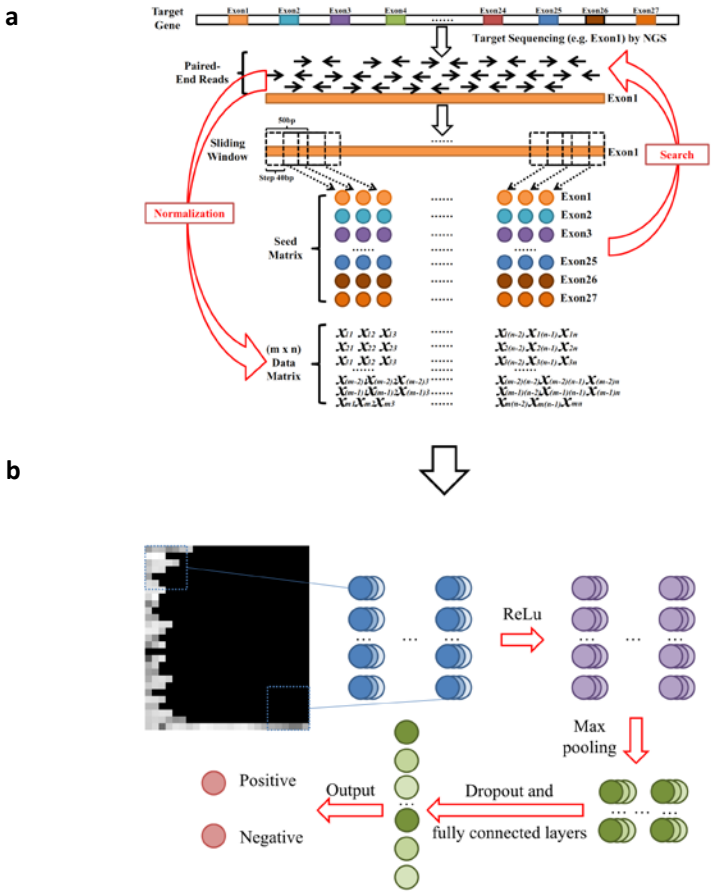


Figure 2. A schematic view of the procedure to detect CNV.

2.2. CNN could outperform the logistic regression model in the training process

The matrices were generated using the methods mentioned in the Materials and Methods section. Two algorithms, logistic regression and CNN, were applied separately to classify the matrices representing positive/negative samples. To make up for the low number of the samples, 10-fold cross validation and 5-fold cross validation were applied to the data of *ERBB2* and *MET*, respectively. Cross-validations were conducted for each gene, generating 10 models for *ERBB2* and 5 models for *MET*. The validation results were merged for analysis. The best hyperparameter were chosen by

AUC using grid search among the parameters mentioned in the materials and methods section. The best accuracy, sensitivity and specificity of those models are shown in Table 1 (Figure 3).

Table 1. The performance of the models given the gene and the algorithm during the training phase.

Gene	*ERBB2*		*MET*	
Algorithm	CNN	Logistic	CNN	Logistic
AUC	0.9304	0.9293	0.8884	0.8497
Accuracy (threshold set to 0.5)	88.27%	86.78%	90.00%	86.09%
Sensitivity (threshold set to 0.5)	83.41%	81.17%	58.33%	60.00%
Specificity (threshold set to 0.5)	89.60%	88.31%	91.74%	87.91%

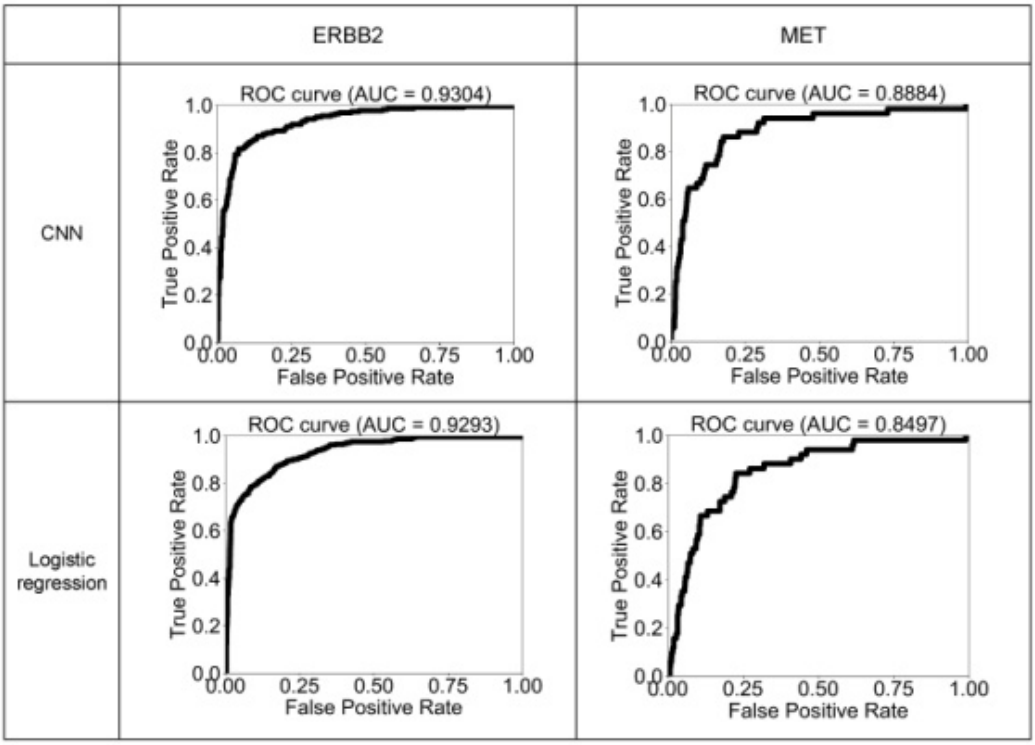


Figure 3. The plotted ROC using the predicted values from the validation dataset during the cross validation.

The CNN model for *ERBB2* outperformed the logistic regression model as expected. The sensitivity for CNN model for the *MET* is lower than that of the logistic regression model. This might have been due to the small number of samples of *MET*.

2.3. CNN had a better generalization ability

Next, the best hyperparameters were applied to train all samples in the training dataset. Afterwards, the trained model was used to classify all samples in the test dataset. The AUC for the

test dataset of *ERBB2* by CNN was 0.9454, similar to that of logistic regression which is 0.9477. The AUC for the test dataset of *MET* by CNN was 0.9220, outperforming that of logistic regression which is 0.8666 (Figure 4). It showed that the CNN method may have a better generalization ability than that of the logistic regression method.

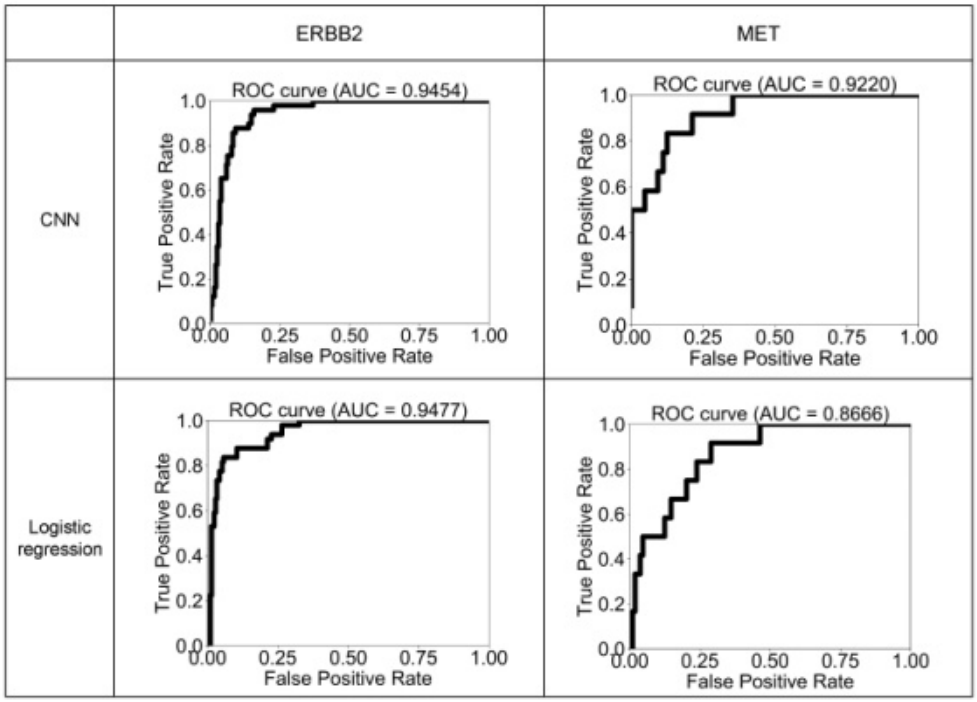


Figure 4. The plotted ROC of the test dataset using the model trained from the whole training dataset using the best hyperparameters.

2.4. Optimization

Since the CNV classification of *MET* was much worse than that of *ERBB2*, we made further efforts to analyze the source of the error.

(a) Firstly, we tried to merged the paired-end reads to avoid some possible repeat counts in the split windows, but the results were not significantly improved or even worse.

(b) Then, we tried to add a GC content correction algorithm to improve our method. The procedure for calculating the GC content of a sample was as follows: (1) The sequences of the previously mentioned windows split from the exons were used as the seed sequence. (2) The reads that contained the seed sequence were extracted from the raw fastq. (3) For each window, a GC content percentage was calculated from the matched reads. (4) The mean of all GC content percentages of all windows was calculated as the divisor for each sample.

We analyzed the GC content difference between the correctly classified samples and the incorrectly classified samples using the training dataset (Figure 5a) and the test dataset (Figure 5b), respectively. We found that in the training/test dataset, the median GC content of the reads that mapped to the gene of the incorrectly classified samples was significantly different from that of the correctly classified samples (two-tailed heteroscedasticity *t* test; *ERBB2*: $p = 0.0015/p = 0.0015$;

MET: $p = 0.032/p = 6.90e{-}05$). Thus, we attempted to reduce the influence of the GC content by dividing the normalized matrices by the mean GC content of the reads covering the gene. To test whether we can utilize the GC content to gain better performance, 78 samples of *MET* were chosen for cross validation. The GC content-corrected matrices were used again on the same sample sets for CNN cross-validation. The accuracy before GC content correction was 89.74%, and the accuracy after GC content correction dropped to 88.46%, showing the above-mentioned method did not boost the CNN performance.

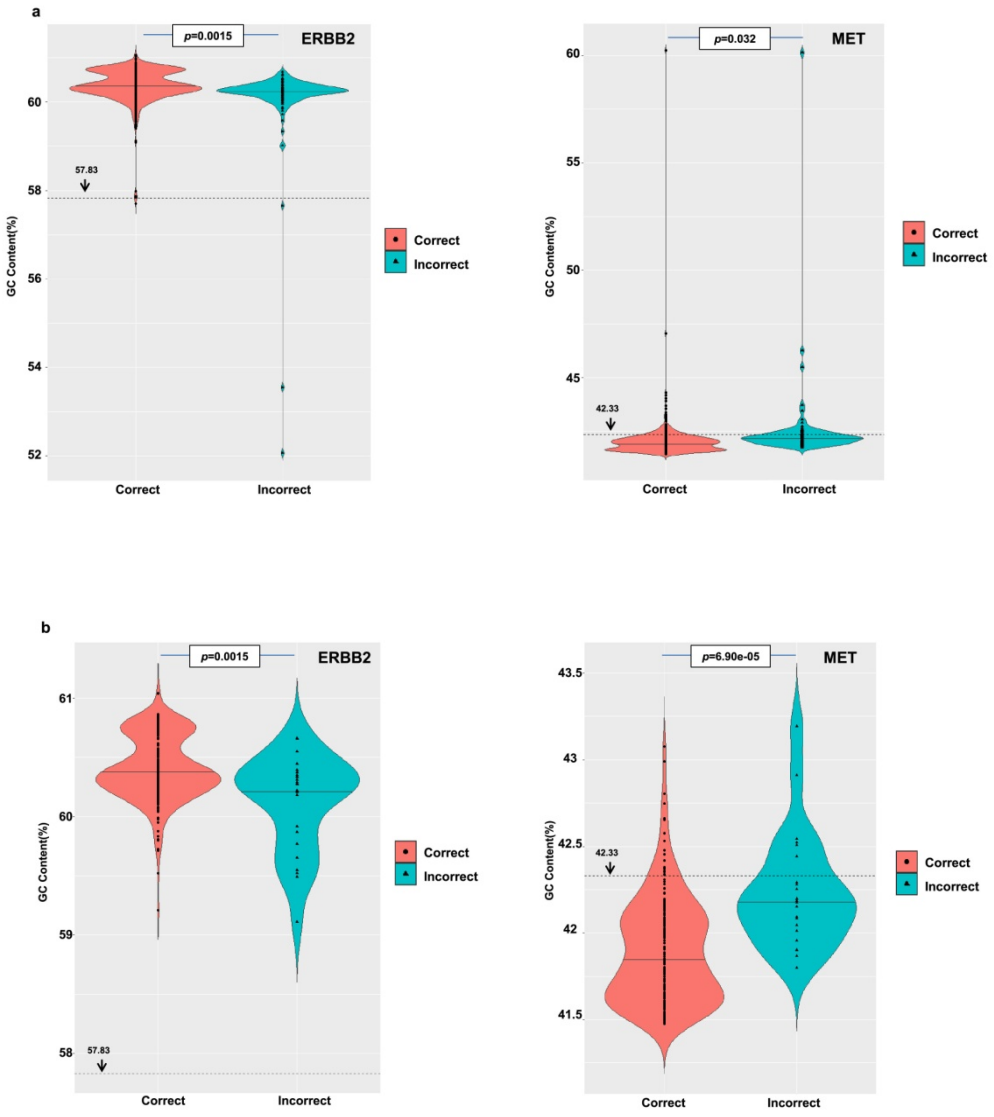


Figure 5. (a) GC content differences between the correctly classified samples and the incorrectly classified samples from the training dataset; (b) GC content difference between the correctly classified samples and the incorrectly classified samples from the test dataset. Dotted line means *ERBB2*/*MET*'s GC content on the reference genome.

2.5. Comparison to other methods

We chose 2 softwares, Control-FREEC [11] and CNVnator [10] as candidates to make a comparison with our method. First, we randomly selected 221 samples (94 CNV-positive samples and 127 CNV-negative samples) for *ERBB2* gene set and 237 samples (14 CNV-positive samples and 223 CNV-negative samples) for *MET* for further tests. The two methods were separately fine-tuned and predicted the CNV of the random selected samples. The results were shown in Table 2. The results indicate that the DL-CNV outperformed the Control-FREEC and the CNVnator, while considering both precisions and recalls.

Table 2. The comparison between the CNVnator, Control-FREEC and DL-CNV on two datasets.

	CNVnator			DL-CNV			FreeC		
ERBB2 dataset	Precision	Recall	Specificity	Precision	Recall	Specificity	Precision	Recall	Specificity
Negative	57.47%	100.00%	0.00%	92.80%	91.34%	90.43%	97.96%	37.80%	98.94%
Positive	0.00%	0.00%	100.00%	88.54%	90.43%	91.34%	54.07%	98.94%	37.80%
Accuracy		57.47%			90.95%			63.80%	
MET dataset	Precision	Recall	Specificity	Precision	Recall	Specificity	Precision	Recall	Specificity
Negative	94.09%	100.00%	0.00%	95.48%	94.62%	28.57%	100.00%	48.43%	100.00%
Positive	0.00%	0.00%	100.00%	25.00%	28.57%	94.62%	10.85%	100.00%	48.43%
Accuracy		94.09%			90.72%			51.48%	

3. Discussion

We developed an alignment-free and no-control method for the detection of CNV in which all that we needed were the reads that covered the gene of interest. Combine with clinical “gold standard” (e.g. FISH) information, the CNV detection could be more accurate since they could complement each other. As the concordance rate between our method and the current NGS method was high, which indicated that the reads covering the gene of interest contained almost enough information to determine the copy number abnormality, there were still contradictory results and optimization was needed. Our method could be a better complement to the existing NGS methods to detect copy number variations. This tool was made freely available at https://github.com/wangbo00129/DL-CNV.

We also inferred that the size of the training set greatly influenced testing performance. Even when the sample number of *ERBB2* and *MET* were similar, the large imbalance of positive/negative samples made the available positive training sample of *MET* relatively lacking.

In this study, we applied a CNN to the detection of the CNV problem. We chose 2 genes, *ERBB2* and *MET*, as candidates, and used the coverage depth throughout the gene region as inputs in order to determine whether a sample had a CNV at the specific gene. To make up for the low number of the samples, cross-validations were performed.

Among each model family generated through cross-validation, the accuracy varied greatly, indicating that the training set was crucial to the CNN model.

Though the concordance rate between the CNN model and the NGS result was high, it remains unclear whether the CNN model has a high concordance rate with the true CNV because

the gold standard method was not used [23]. This method could be viewed as an auxiliary method for the NGS method.

Furthermore, as shown in the results from the *ERBB2* models and the *MET* models, the accuracy, sensitivity and specificity was greatly influenced by the size of the training set. While the training set was too small (this was caused by lower frequency of *MET* CNV, leading to only 76 samples for the *MET* dataset), the CNN model may not outperform the logistic regression model. However, even the CNN model output exhibited lower accuracy than the logistic regression model in the cross-validation for *MET*. The CNN model performed better in the test set, showing its strong generalization ability.

We recommend the positive sample number should be similar to the *ERBB2* dataset, which is about 200 if a high precision and recall rate are expected. Since the negative samples are more than the positive samples in most cases, the negative samples were not discussed here.

We further attempted to refine the CNN model by adjusting the stride size, joining the raw fastq format files. Furthermore, we calculated the median GC content of all the reads that matches each window of the specific gene, and divide the read number matrices by dividing by the GC content median for GC content correction. After adjusting the stride size, the models did not seem to improve, indicating that the stride size did not affect the performance of the method. After joining the fastqs, the CNN's performance did not change significantly, showing that the fused fastqs neither abolished nor gained any information. After the GC content correction, the specificity of the models improved, but the gain was insignificant, which showed that the GC content might not have been the source of error.

We can further optimize the current method by (1) collecting more training sets, (2) taking the case history into account, (3) working with the base qualities while calculating coverage depth and (4) re-considering the way to utilize the GC content for each sample, (5) coming up with a method for lowering the amount of the sample size, which will make researches about rare diseases possible. The method is expected to exhibit better performance on the detection problem when utilizing information other than the read coverage depth.

Two commonly used CNV detecting tools, Control-FREEC and CNVnator was also used for about 200 samples for each gene dataset. Apparently, our method detects the CNVs better. It is worth noting that the CNVnator makes all-0 predictions, which might be due to the CNVnator was developed for whole-genome sequencing but not for target sequencing. As for Control-FREEC, the samples tend to be predicted as CNV-positive samples, which raised the false positive rate. One possible reason might be that that the two tools both work at a relatively wide sequencing intervals, which leads to deficiency at narrow sequencing intervals as the scope was shrunk to the single-gene-wide scope. Thus, at small intervals, the DL-CNV method performed better than the traditional tools.

4. Methods

4.1 Sample source

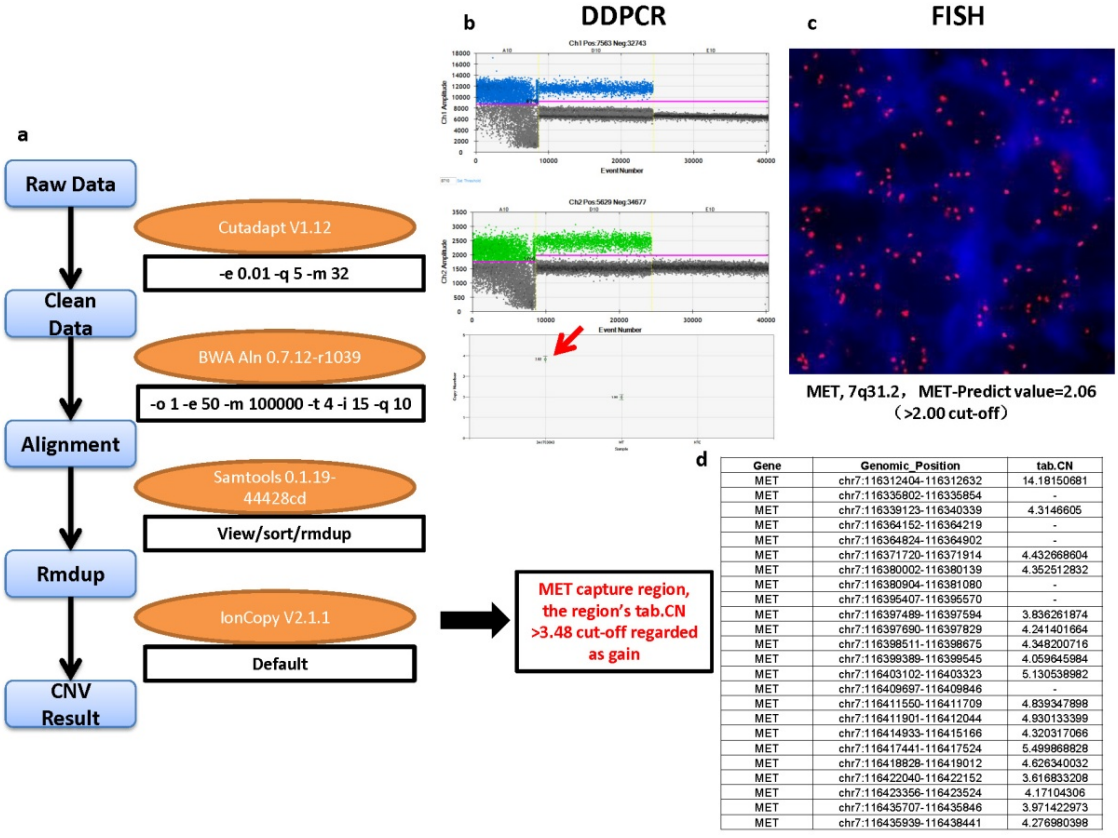
We collected 2449 (*ERBB2*: 1301; *MET*: 1148) non-paired samples data through *ERBB2+MET* whole-exon-designed NGS panel sequencing with Illumina Nextseq500 platform (Illumina, San Diego, CA, US). All the data were produced from *Geneis Lung Cancer 41 Gene Detecting Panel.*

4.2 CNV classification procedure by NGS

Ioncopy [14] was used to label the CNV samples as positive or negative. The pipeline of CNV detection through an NGS panel was (1) use Cutadapt V1.12 to cut the adapters with parameters of "–e 0.01 –q 5 –m32"; (2) use bwa-aln [24] for alignment with the parameters of "–o 1 –e50 –m 100000 –t 4 –i 15 –q 10"; (3) use samtools [25] to remove duplicates; (4) use Ioncopy with the default cut-off value of 3.48 for tab.CN to classify the samples (Figure 6a). Ioncopy is used for to generate the labels of the samples for indicating CNV.

4.3 Preparation of datasets combined with FISH results

Based on the FISH results, the CNV-positive sample's target region almost always had the gain signal, so we defined this to calibrate our predicted CNV-positive samples (Figure 6b–d). Finally, we collected 272 *ERBB2* CNV-positive samples and 1029 CNV-negative samples, and we collected 63 *MET* CNV-positive samples and 1085 CNV-negative samples.



Gene	Genomic_Position	tab.CN
MET	chr7:116312404-116312632	14.18150681
MET	chr7:116335802-116335854	-
MET	chr7:116339123-116340339	4.3146605
MET	chr7:116364152-116364219	-
MET	chr7:116364824-116364902	-
MET	chr7:116371720-116371914	4.432668604
MET	chr7:116380002-116380139	4.352512832
MET	chr7:116380904-116381080	-
MET	chr7:116395407-116395570	-
MET	chr7:116397489-116397594	3.836261874
MET	chr7:116397690-116397829	4.241401664
MET	chr7:116398511-116398675	4.348200716
MET	chr7:116399389-116399545	4.059645984
MET	chr7:116403102-116403323	5.130538982
MET	chr7:116409697-116409846	-
MET	chr7:116411550-116411709	4.839347898
MET	chr7:116411901-116412044	4.930133399
MET	chr7:116414933-116415166	4.320317066
MET	chr7:116417441-116417524	5.499868828
MET	chr7:116418828-116419012	4.626340032
MET	chr7:116422040-116422152	3.616833208
MET	chr7:116423356-116423524	4.17104306
MET	chr7:116435707-116435846	3.971422973
MET	chr7:116435939-116438441	4.276980398

Figure 6. The procedures to infer the label. (a) A pipeline for CNV detection by an NGS panel using Ioncopy; (b) the ddPCR validation for a *MET* CNV-positive sample, which showed high accordance with FISH result; (c) the FISH validation for a *MET* CNV-positive sample; (d) the analysis results of a *MET* CNV-positive sample by an NGS panel.

4.4 Coverage depth calculation

The first step of our algorithm was the preparation of the read number matrix of the specific gene. Our approach involved no alignment. Instead, we split the reference exons into multiple 50-bp bins with a stride of 40-bp (or 25-bp), with a 10-bp (or 25-bp) overlap between every pair of adjacent windows. We counted the reads that matched the whole bin to calculate the read number at each bin. The rows representing the exons consisting of the read numbers were then piled up to form up the input matrix for the next step's training (Figure 1). Since the exons were of different lengths, the max length was used as the width of the matrix, in which the blank elements were filled by zeroes.

4.5 Coverage normalization

The original matrices were first used as the dataset for the prediction of CNV. However, when the CNN were fed with the training set, the algorithm did not converge after 20,000 iterations. We inspected the prediction accuracy of the training set and found that the algorithm predicted whole ones or whole zeroes for all samples, leading the training accuracy to remain at ~50%. The predictions for the test set also consisted of whole ones or whole zeros. We presumed that the read numbers were so imbalanced in different samples, which induced a high volume of noise in the algorithm. As a result, we attempted to normalize the matrices to lower the impact of the read number noise. For normalizing purposes, we chose the mean value of the non-zero elements of each matrix as the divisor for the elements from the corresponding matrix. If necessary, the GC content percentage was used for the divisor of the normalized matrix to perform the GC content correction.

4.6 Network design

The LeNet-5 model for classifying MNIST [15] was used to classify the CNV samples. The deep learning framework TensorFlow [26] was used to implement the CNV detection algorithm. The algorithm structure is shown in Figure 2. It contains one convolutional layer, one max pooling layer, one dropout layer and two fully connected layers. The kernel used in the convolutional layer was set to $n \times n$ where $n \in \{3,5,7,9,11,13,15,17,19,21\}$. The kernel used in the max pooling layer was set to 2×2. The fully connected layers were of the size 1024 and 2, in which the second layer was the output layer that classified whether the sample was CNV-positive. The dropout layer, whose dropout value was set to $lr \in \{0.3,0.5,0.7,0.9,1.0\}$, was between the two fully connected layers. The Adam algorithm was used as the gradient-decent method. The learning rate was set as 1×10^{-n} where $n \in \{4,5,6,7\}$. The model was trained for 10,000 iterations with a batch size of 30 samples.

For the implementation of the logistic regression model, the sigmoid function was used as the activation function and the batch-gradient descent was used. The learning rate was set to 1×10^{-n} where $n \in \{4,5,6,7,8,9\}$. The model was trained for 10,000 epochs.

4.7 Training and cross validation

272 NGS-positive and 1029 NGS-negative samples for *ERBB2*, and 63 NGS-positive and 1085 NGS-negative samples for *MET* were used as the dataset for the evaluation of our method.

For the balance of positive and negative samples, 223 NGS-positive and 817 NGS-negative *ERBB2* samples were used as the training set of the *ERBB2* CNN model. 51 NGS-positive and 867 NGS-negative *MET* samples were used as the training set of the *MET* CNN model. The rest of the *ERBB2* and *MET* samples were used as the test set for the two genes, respectively. The training set was randomly split into 10 parts for the *ERBB2* samples and 5 parts for the *MET* samples for further cross-validation.

4.8 Evaluation of classifier

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TP}{TP+FP}$$

where: TP = True positive; FP = False positive; TN = True negative; FN = False negative.

4.9 Other methods for comparison

The other methods, Control-FREEC and CNVnator was chosen as candidates for the comparison with our method. The parameters of Control-FREEC and CNVnator were separately adjusted according to their instructions. Specifically, the parameters of Control-FREEC were "breakPointType = 4; window = 0, breakPointThreshold = 1.2, readCountThreshold = 50" and the parameters of CNVnator were "-his 30;-stat 30;-partition 30;-call 30".

Acknowledgements

Jidong Lang, Bo Wang, Yunxiang Zhang and Lvcheng Jin implemented the algorithms and wrote the paper. Dehong Hu, Leqiang Wang, Kai Han, Pan Li and Junling Zhang collected the samples. Wei Tan, Xiaoming Xing, Dawei Yuan, Geng Tian, Jialiang Yang and Jidong Lang reviewed the manuscript. Jidong Lang developed the idea, analyzed the data and coordinated the study.

The authors thank Siwen Zhang, Yingmin Han, Yuebin Liang and Huixin Lin for helpful discussions and comments. We thank Xu Chu for help adjusting the figures.

This work was supported by the Scientific and Technological Development Project of Shandong Province (2015GSF118168).

Conflict of Interest

The authors have no conflicts of interest to declare.

References

1. S. A. McCarroll and D. M. Altshuler, Copy-number variation and association studies of human disease, *Nat. Genet.*, **39** (2007), S37–42.
2. P. Liu, C. M. Carvalho, P. J. Hastings, et al., Mechanisms for recurrent and complex human genomic rearrangements, *Curr. Opin. Genet. Dev.*, **22** (2012), 211–220.
3. A. P. de Koning, W. Gu, T. A. Castoe, et al., Repetitive elements may comprise over two-thirds of the human genome, *Plos Genet.*, **7** (2011), e1002384.
4. M. Zarrei, J. R. MacDonald, D. Merico, et al., A copy number variation map of the human genome, *Nat. Rev. Genet.*, **16** (2015), 172–183.
5. J. L. Freeman, G. H. Perry, L. Feuk, et al., Copy number variation: new insights in genome diversity, *Genome Res.*, **16** (2006), 949–961.
6. S. F. Chin, A. E. Teschendorff, J. C. Marioni, et al., High-resolution aCGH and expression profiling identifies a novel genomic subtype of ER negative breast cancer, *Genome Biol.*, **8** (2007), R215.
7. D. He, N. Furlotte and E. Eskin, Detection and reconstruction of tandemly organized de novo copy number variations, *BMC Bioinf.*, **11** (2010), S12.
8. G. Klambauer, K. Schwarzbauer, A. Mayr, et al., cn.MOPS: Mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate, *Nucleic Acids Res.*, **40** (2012), e69.
9. E. Talevich, A. H. Shain, T. Botton, et al., CNVkit: Genome-Wide Copy Number Detection and Visualization from Targeted DNA Sequencing, *PLoS Comput. Biol.*, **12** (2016), e1004873.
10. A. Abyzov, A. E. Urban, M. Snyder, et al., CNVnator: An approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing, *Genome Res.*, **21** (2011), 974–984.
11. V. Boeva, T. Popova, K. Bleakley, et al., Control-FREEC: A tool for assessing copy number and allelic content using next-generation sequencing data, *Bioinf.*, **28** (2012), 423–425.
12. G. Onsongo, L. B. Baughn, M. Bower, et al., CNV-RF Is a Random Forest-Based Copy Number Variation Detection Method Using Next-Generation Sequencing, *J. Mol. Diagn.*, **18** (2016), 872–881.
13. C. Wang, J. M. Evans, A. V. Bhagwate, et al., PatternCNV: A versatile tool for detecting copy number changes from exome sequencing data, *Bioinf.*, **30** (2014), 2678–2680.
14. J. Budczies, N. Pfarr, A. Stenzinger, et al., Ioncopy: A novel method for calling copy number alterations in amplicon sequencing data including significance assessment, *Oncotarget*, **7** (2016), 13236–13247.
15. Y. L. Cun, L. Bottou, Y. Bengio, et al., Gradient-Based Learning Applied to Document Recognition, *Proc. IEEE.*, **86** (1998), 2278–2324.
16. J. Zhou and O. G. Troyanskaya, Predicting effects of noncoding variants with deep learning-based sequence model, *Nat. Methods*, **12** (2015), 931–934.
17. R. Poplin, D. Newburger, J. Dijamco, et al., Creating a universal SNP and small indel variant caller with deep neural networks, *BioRxiv*, (2016), 092890.
18. M. X. Sliwkowski, J. A. Lofgren, G. D. Lewis, et al., Nonclinical studies addressing the mechanism of action of trastuzumab (Herceptin), *Semin Oncol.*, **26** (1999), 60–70.

19. S. Ahn, M. Hong, M. Van Vrancken, et al., A nCounter CNV Assay to Detect HER2 Amplification: A Correlation Study with Immunohistochemistry and In Situ Hybridization in Advanced Gastric Cancer, *Mol. Diagn. Ther.*, **20** (2016), 375–383.
20. F. Sircoulomb, I. Bekhouche, P. Finetti, et al., Genome profiling of *ERBB2*-amplified breast cancers, *BMC Cancer*, **10** (2010), 539.
21. S. Kim, T. M. Kim, D. W. Kim, et al., Acquired Resistance of *MET*-Amplified Non-small Cell Lung Cancer Cells to the *MET* Inhibitor Capmatinib, *Cancer Res. Treat.*, **5** (2019), 951–962.
22. N. Pfarr, R. Penzel, F. Klauschen, et al., Copy number changes of clinically actionable genes in melanoma, non-small cell lung cancer and colorectal cancer-A survey across 822 routine diagnostic cases, *Genes Chromosomes Cancer*, **55** (2016), 821–833.
23. F. Zare, M. Dow, N. Monteleone, et al., An evaluation of copy number variation detection tools for cancer using whole exome sequencing data, *BMC Bioinf.*, **18** (2017), 286.
24. H. Li and R. Durbin, Fast and accurate short read alignment with Burrows-Wheeler transform, *Bioinf.*, **25** (2009), 1754–1760.
25. H. Li, B. Handsaker, A. Wysoker, et al., The Sequence Alignment/Map format and SAMtools, *Bioinf.*, **25** (2009), 2078–2079.
26. M. Abadi, P. Barham, J. Chen, et al., *TensorFlow: A system for large-scale machine learning*, 12th Symposium on Operating Systems Design and Implementation (OSDI), 2016, 265–283. Available from: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

AIMS Press