Spatiotemporal attention boosts calling of complicated variations from long reads' alignment data

Ying Shi
y.shi@siat.ac.cn
School of Computer and Information Technology, Shanxi University
Taiyuan, Shanxi, China Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology
Shenzhen, Guangdong, China

Shifu Luo
yc37182@connect.um.edu.mo
Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology
Shenzhen, Guangdong, China Faculty of Health Sciences, University of Macau
Taipa, Macau SAR, China

Yi Pan
yi.pan@siat.ac.cn
Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology
Shenzhen, Guangdong, China
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
Shenzhen, Guangdong, China

Hao Wu
wuhao@siat.ac.cn
Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology
Shenzhen, Guangdong, China
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
Shenzhen, Guangdong, China

Wenjian Wang*
wjwang@sxu.edu.cn
School of Computer and Information Technology, Shanxi University
Taiyuan, Shanxi, China

Jinyan Li*
lijinyan@suat-sz.edu.cn
Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology
Shenzhen, Guangdong, China
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
Shenzhen, Guangdong, China

Abstract

With the latest Q20 technology, the base error rate in ONT long-read sequencing has been down to 1%. However, such rates of sequencing errors (base insertions, deletions or substitutions) still lag behind the 0.1% base error rate in NGS short reads, resulting in many complicated variation regions in the full alignment data of deeply sequenced long reads and posing a big challenge to germline variant calling. For example, current deep learning methods could misidentify 20,000 to 30,000 variants from the ONT long reads basecalled by Q20 on a single chromosome, or could misidentify more than 30,000 at the complicated variation regions when the reads basecalled by Guppy v5.0.14. We proposed a spatiotemporal attention deep learning method (Attdeepcaller) to boost the performance of variation calling on these complicated variation regions. The novel use of spatiotemporal attention is to modulate the confusion between genuine sequencing errors and the true germline variations in the alignment data so that the identification by the algorithms becomes clear at most cases. As tested on the complicated regions in the alignment data basecalled by Q20 on chr1 of HG002, Attdeepcaller made only 22,739 misidentifications, reduced by 12.69% from current 26,043 misidentifications. Similarly, the misidentification number is reduced by 16.49% on HG003 and by 23.58% on HG004 compared with the current best. When tested on the Guppy 5 alignment data, Attdeepcaller improved the precision by 3 percent and the recall by 1 percent on the complicated variation regions. We also conducted comparative analysis of these methods on old versions of guppy data. Specifically on the Guppy v3.4.5 datasets, Attdeepcaller boosted the precision by a jump of 16 percent and improved the recall by 10 percent. This result suggests that Attdeepcaller can still work and can work substantially better when the reads alignment data becomes more complicated (the older the version of basecalling, the higher the base error rate of the sequencing data, and the more complicated the alignment data is).

Keywords

Spatiotemporal attention, long reads' alignment data, complicated variation regions, germline variant calling

*Both authors contributed equally to this research.

This work is licensed under a Creative Commons Attribution International 4.0 License.

BCB '24, November 22–25, 2024, Shenzhen, China

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1302-6/24/11
https://doi.org/10.1145/3698587.3701335

ACM Reference Format:
Ying Shi, Shifu Luo, Yi Pan, Hao Wu, Wenjian Wang, and Jinyan Li. 2024. Spatiotemporal attention boosts calling of complicated variations from long reads' alignment data. In *15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '24), November 22–25, 2024, Shenzhen, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3698587.3701335

1 Introduction

Germline variant calling or detection of gene variations such as SNPs and small Indels directly from DNA/RNA sequencing alignment data is helpful in many applications [Marx,2023, Logsdon *et al.*,2020], including population genetics, precision medicine, cancer therapy, and phylogeny evolution. The detection is very accurate when PacBio CCS HiFi reads or Illumina short-read sequencing data are given because the data is of extraordinary quality with almost no sequencing errors.

However, the detection becomes challenging when we are given the increasingly adopted third generation Nanopore long-read sequencing data. This is because the raw data may contain up to 20% base errors caused by sequencing mistaken incidents [Shendure *et al.*,2017]. Even basecalled by the Guppy tools [Wick *et al.*,2019], some of these errors can still pass to the reads. Although the errors contained in the reads are occasionally located at some preferred regions exhibiting patterns, these sequencing errors rarely happen at the same base positions in deep sequencing processes. Thus, the true SNP or Indel regions in the alignment data of deeply sequenced long reads when mapped to a reference genome can be much ruined by the sequencing errors. Meanwhile, those alignment regions without any variation can be significantly messed by these sequencing errors as well. Thus, the variation detection problem becomes very complicated because of the confusion between the sequencing errors and the true variations in these data regions.

Examples of these complicated variation regions (exact definition is given in the next section) from the Guppy 3 alignment data are depicted in Fig.1, where the first four complicated regions contain true SNPs or true Indels, while the last two complicated regions contain no germline variations. Although there are huge leaps in advancing new basecalling approaches (such as the newest Q20 and Guppy 5 and above) which have massively removed the base errors in the ONT long reads, there still exist tens of thousands of complicated variation regions in these long reads' alignment data.

Machine learning algorithms were proposed recently to decide whether a true variation exists or not in each of these complicated variation regions. However, current best learning method Clair3 (the latest version) [Zheng *et al.*,2022] had misidentified 20,000 to 30,000 variants on a single chromosome from the alignment data of the ONT long reads basecalled by Q20; On the alignment data of ONT long reads basecalled by Guppy v5.0.14, Clair3 made more than 30,000 misidentified variants. On the old versions of the data such as Guppy v3.2.5, the misclassification rate is much higher. An earlier method DeepVariant [Poplin *et al.*,2018] has outstanding performance (99.94% accuracy) only on short reads with no effective applications in ONT long-read sequencing data. Other variant detection methods, such as Clairvoyante [Luo *et al.*,2019], Clair [Luo *et al.*,2020], and Nanocaller [Ahsan *et al.*,2021], use *pileup summaries* of the long reads as algorithm input, making performance improvements over DeepVariant but still incompetent to deal with the calling on the complicated variation regions. Method PEPPER [Shafin *et al.*,2021] further improved the performance by incorporating spatial information of the read alignments in a form called *full-alignment*, which is dozen times larger in size than pileup data. Nevertheless, PEPPER is reported to be inferior in precision to Clair3 [Zheng *et al.*,2022], the current best method exploiting the strong potential of deep ResNet learning [He *et al.*,2016], when tested on the same set of full-alignment data.

A full alignment is a 3-dimensional data cube (denoted as $Cube_{C\times H\times W}$ here), that contains interweaved channel and spatial information of the deeply sequenced long reads after aligned to a reference genome. In this work, a full alignment consists of 23496 integer numbers allocated into 8 channels (i.e., $_C$, the number of sequencing channels such as reference base, alternative base, strand information, mapping quality, base quality, candidate proportion, insertion base and phasing information) at 33 gene positions ($_W$) with a maximum read coverage of 89 ($_H$). As such a big data cube also contains many random sequencing errors confusing the detection of true positions of SNPs or Indels, *removal or modulation of these heavy noise* is crucially important for the subsequent variation detection. However, there is no pre-processing steps for Clair3 to extract confusion-modulated essential features from these full alignment data cubes before the deep learning method ResNet is taken for the germline variant calling.

We propose two novel steps to overcome the limits of Clair3' workflow for a more accurate detection of germline variations from the complicated variation regions in the alignment data. One novelty is that we use ResNeXt [Xie *et al.*,2017] instead of ResNet because ResNeXt updates blocks and uses Group Convolution with a smaller number of parameters than ResNet for a better classification clarity. More importantly, we propose to use *spatiotemporal attention* to extract confusion-modulated features from the full alignment data cubes before the newly introduced ResNeXt is applied for the germline variant calling. Spatiotemporal attention is a double-attention mechanism which specializes in feature transformation of 3-dimensional data input and extracts weighted essential features from the full alignment data cubes. The key idea of this attention is originated in CBAM (Convolutional Block Attention Module [Woo *et al.*,2018]) that combines both spatial and channel information to weigh the raw input features and then place greater weights on attention features.

Extensive tests on ONT long reads' alignment datasets basecalled by various Guppy versions demonstrate that Attdeepcaller can reduce lots of variant misidentifications in comparison to Clair3 or other long-read variation detection methods such as PEPPER.

2 Methods and Benchmark Datasets

We present novelties of our method and dataset details in this section.

2.1 Detailed workflow of Attdeepcaller

Our germline variant calling method is a deep aggregated residual neural network that integrates the reads' sequence feature extraction data and the variant classification into an end-to-end network. The network structure consists of two BiLSTM (Bi-directional Long Short-Term Memory) layers [Zhang *et al.*,2015], four ResNeXt+CBAM Blocks, one PyramidPooling layer, and three feedback layers. Shortcut links are set at the input and output terminals of each ResNeXt module. A dropout strategy [Ghiasi *et al.*,2018] is added before each fully connected layer to increase the robustness of the model. Fig.2 shows the overall network architecture of our variation detection method Attdeepcaller.

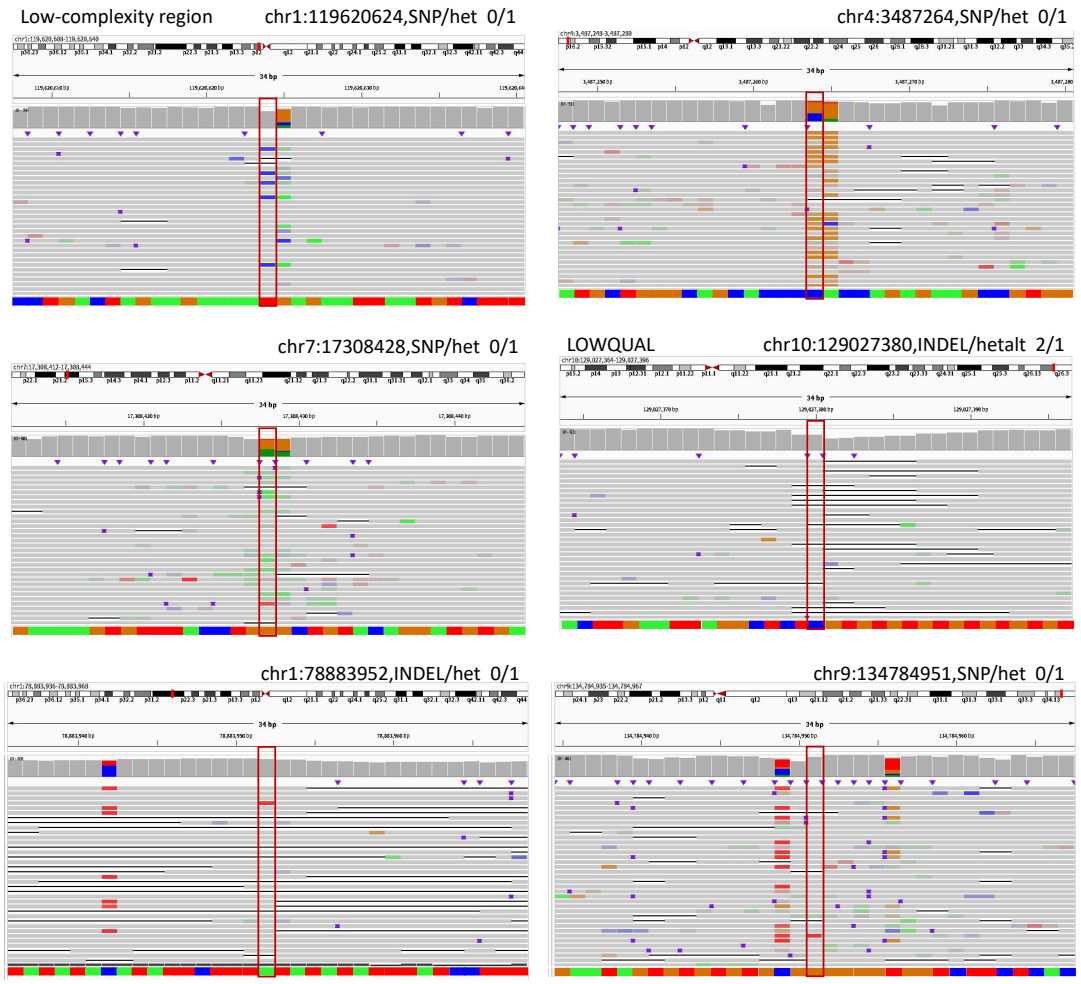

Figure 1: Complicated variation regions in the alignment data of ONT long reads (basecalled by Guppy v3.4.5) as visualized through the IGV plots of the Nanopore reads of the HG002 genome at different base positions. In these screen captures, bases A, G, C, and T are in green, orange, blue, and red, respectively. Different shades of the colors indicate different base coverage. Insertions are represented by those purple dots and deletions are denoted by long black lines. Here 'hom' represents a homozygous reference, 'het' represents heterozygous with 1/2 alternative alleles, and 'homalt' represents a homozygous variant, 'hetalt' represents heterozygous with multiple alleles.

Input files to the network for model training include the BAM file of the reads, a reference sequence file (in .fa format), and the ground-truth VCF file of the variations (see the detailed data workflow of Attdeepcaller at Supplementary Figure 1). The workflow first chooses the pileup data of the candidate variant site as input data and then uses the pileup network for calling. The Pileup network classifies each input into four sets of predictions: (1) a homozygous reference (0/0); (2) heterozygous with 1/2 alternative alleles (0/1); (3) heterozygous with multiple alleles (1/2); and (4) a homozygous variant (1/1). All the variant candidates are ranked by the quality of variation (QUAL). The Pileup network directly classifies the high-quality variants, and also generates a phased alignment through WhatsHap [Patterson *et al.*,2016] for the improperly classified low-quality candidate variants, which are input into the full-alignment network for a further detection. The high-quality heterozygous SNP calls (the top 70% of 0/1 calls) are also included as input to the full-alignment network to ensure variant detection more accurate.

When the workflow goes for the pileup low-quality candidates, it has four prediction tasks: (1) the 21-genotype probabilistic labels; (2)zygosity; (3) the length of the first indel allele; (4) the length of the second indel allele. The 21-genotype comprises all of the possible genotypes of a deploid sample at a genome position, including $'AA'$, $'AC'$, $'AG'$, $'AT'$, $'CC'$, $'CG'$, $'CT'$, $'GG'$, $'GT'$, $'TT'$, $'AI'$, $'CI'$, $'GI'$, $'TI'$, $'AD'$, $'CD'$, $'GD'$, $'TD'$, $'II'$, $'DD'$, and $'ID'$, where $'A'$, $'C'$, $'G'$, $'T'$, $'I'$ (insertion), $'D'$ (deletion) denote the six possible alleles. The two indel-length prediction tasks tell an exact indel length from -15 to 15bp, or below -15bp / above 15bp. These four tasks are bound to each other, and we have added cross-validation to the code to improve the accuracy of variant detection. For example, the zygosity prediction is a coarse-grained version of task one and

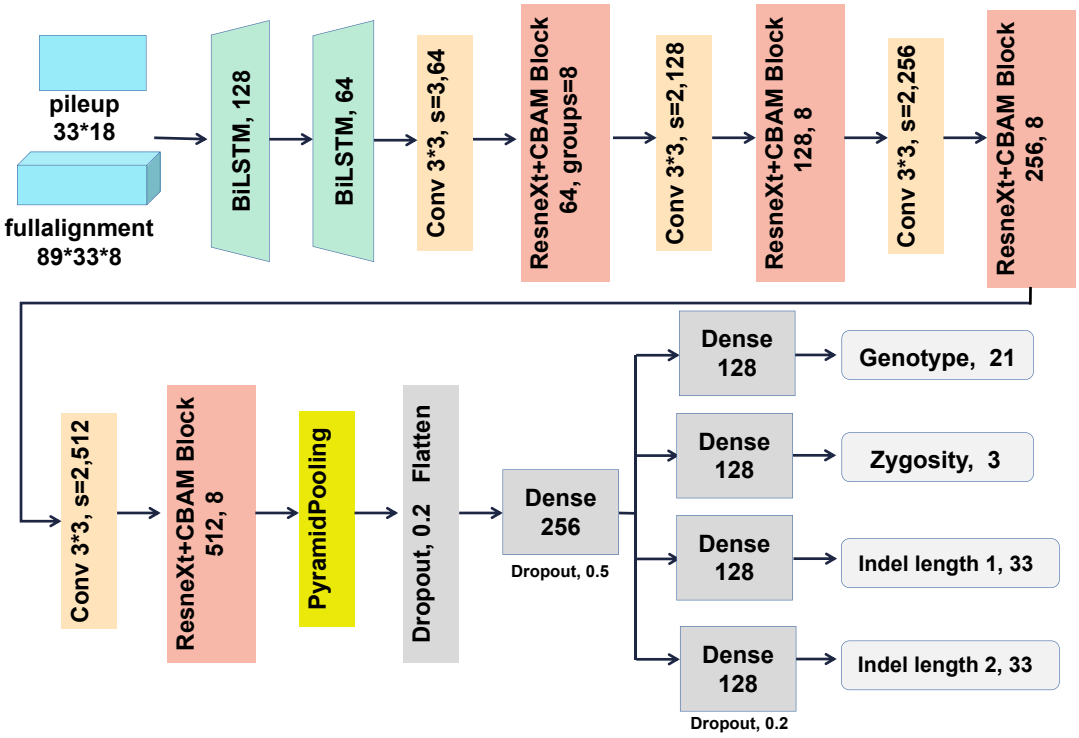

Figure 2: Attdeepcaller network architecture.

can veto the decision made by task one. Tasks 3 and 4 will tell a 0-bp indel length if an SNP variant is decided by task one. The key component in this brunch of workflow is the ResNeXt+CBAM block, where spatiotemporal attention is used for feature extraction and noise modulation/removal.

As other input data details for the workflow, the pileup input contains a matrix of 594 integers: 33 genome positions wide with 18 features at each position (A+,C+, G+, T+, I_S+, I_S^1+, D_S+, D_S^1+, D_R+, A−, C−, G−, T−, I_S−, I_S^1−, DS−, D_S^1−, D_R−). Symbols A, C, G, T, I, D, +, and -, respectively, stand for the count of read support of the four nucleotides: insertion, deletion, positive strand, and negative strand. Superscript $'1'$ means that only the indel with the highest read support is counted (i.e., all indels are counted if without $'1'$). Subscript $'S'/'R'$ means the starting/non-starting position of an indel. For example, a 3bp deletion with the most reads support will have the first deleted base counted in either D_S^1+ or D_S^1−, and the second and third deleted bases counted in either D_R+ or D_R−. The full-alignment input is a cube of 23,496 integers: 8 channels of 33 genome positions and 89 maximum representable reads. The 8 channels are referred to as Reference base, Alternative base, Strand information, Mapping quality, Base quality, Candidate proportion, Insertion base and Phasing information.

2.2 Complicated variation regions in the 3D alignment data: definition and examples

As introduced, not all SNP or Indel regions in ONT reads' alignment data contain heavy confusion noise caused by the sequencing errors. If the *pileup summary data* of a candidate region (well defined and exploited by Clairvoyante [Luo *et al.*,2019], Clair [Luo *et al.*,2020], and Nanocaller [Ahsan *et al.*,2021]) is of high-quality and clarity, Attdeepcaller does not use its full-alignment data cube as input to predict whether the variation region is a true variation or not. Otherwise, Attdeepcaller needs the full-alignment cube as input data and takes the spatiotemporal-attention deep learning to make the prediction. Namely, the sequences of the long reads are first aligned in the pileup mode, and a variant detection is performed; only those low-quality candidates which cannot be confidently detected in the pileup alignment step are diverted into the process of using full-alignment for the algorithm to make a decision.

We call those alignment regions having a high-quality score of pileup summary *simple variation regions* in the ONT germline variation detection. By visualization, simple variation regions are clear pictures of variations or clear pictures of non-variations having little confusion noise inside them (Fig.3).

On the other hand, we call those alignment data regions having a low-quality score and high-quality heterozygous SNP calls of pileup summary *complicated variation regions*. By visualization, complicated variation regions are such blurred pictures of variations or blurred pictures of non-variations those that are indistinguishable vividly or by linear prediction algorithms.

Fig.1 shows some of the complicated variation regions which can be correctly identified. Of them, Fig1.1 and Fig1.2 are two cases that can be detected by both Attdeepcaller and Clair3, while the remaining two cases (Fig1.3 to Fig1.4) are complicated variations

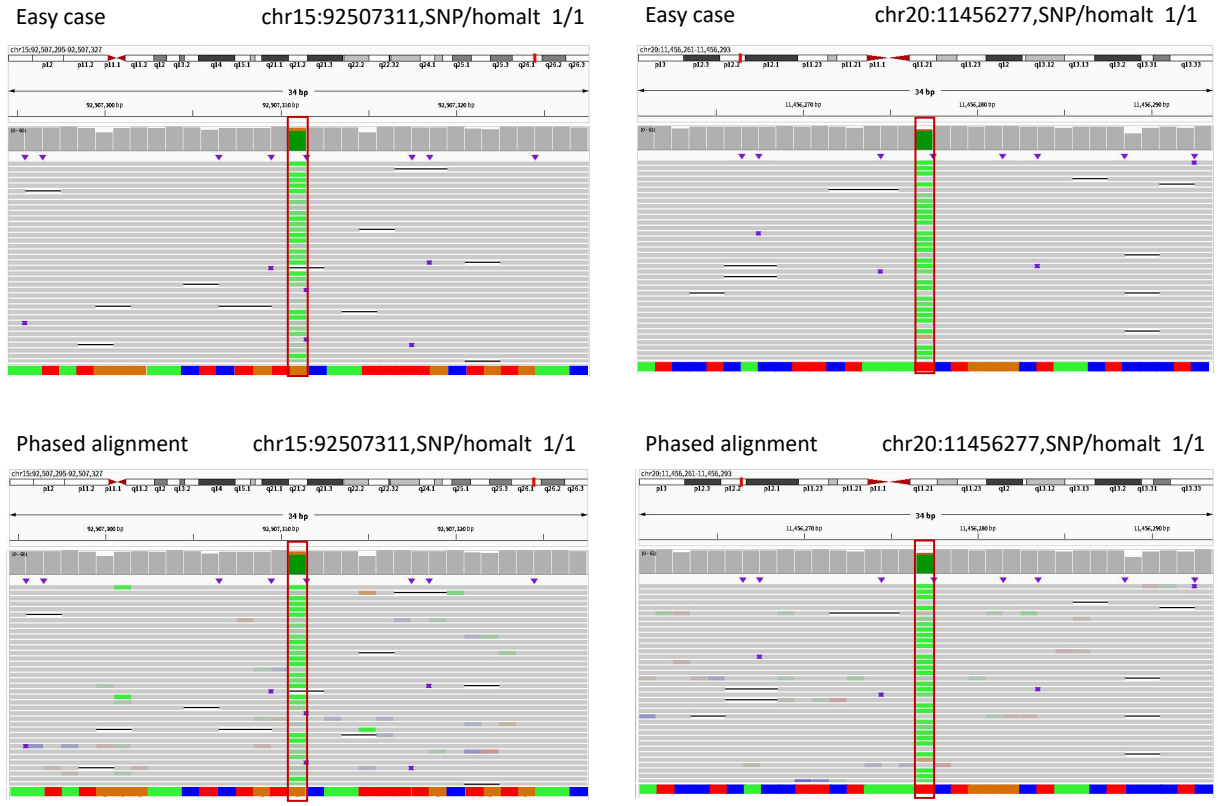

Figure 3: Visualization of simple variation regions in the alignment data of ONT long reads basecalled by Guppy v3.4.5 specially through the IGV plots of Nanopore reads of the HG002 genome at different positions. In the screen captures, bases A, G, C, and T are in green, orange, blue, and red, respectively. Different shades of the colors indicate different base coverage. Insertions are represented by those purple dots and deletions are denoted by long black lines. Here 'hom' represents a homozygous reference, 'het' represents heterozygous with 1/2 alternative alleles, and 'homalt' represents a homozygous variant, 'hetalt' represents heterozygous with multiple alleles.

that can be detected by Attdeepcaller only. Fig.3.1 and Fig.3.2 display two simple variation regions which have high-quality pileup scores; and Fig.3.3 and Fig.3.4 show the diagrams of the two simple variation regions in the phased alignment, where the features are sharp and easy to be distinguished.

2.3 Attention-based ResNeXt: novel use of spatiotemporal attention for feature extraction from 3-dimensional full-alignment data cubes

As seen at the ResNeXt+CBAM modules in the architecture of Attdeepcaller (the orange color part of Fig.2), we propose to use spatiotemporal attention, a double-attention mechanism which specializes in feature transformation of 3-dimensional data input, to extract confusion-modulated essential features from the full alignment data cubes. Specifically by this attention (see Fig. 4), the input feature F is first modeled for channel attention, and varied weights are assigned to these channels to get F'. Then, the spatial attention of feature F' is modeled, and each model places more attention to the region of interest of each feature space (the gene-coverage space) to obtain F''. We then multiply the feature vector F with the weight coefficient F'' to get the final subset of important features. This double-attention mechanism is denoted as

$$F_{output} = F \otimes F'' \tag{1}$$

where, input features $F \in R^{C\times H\times W}$, C is the number of channels in the full alignment data cubes, H represents the height of the cube, and W stands for the width of the cube; the symbol $\otimes$ represents an element-wise multiplication.

F' and F'' are defined as

$$F^{'} = M_C(F) \otimes F \tag{2}$$

$$F^{''} = M_S(F^{'}) \otimes F^{'} \tag{3}$$

where, $M_c \in R^{C\times 1\times 1}$ is a one-dimensional channel attention graph inferred by the CBAM module along the feature channel; $M_S \in R^{1\times H\times W}$ is a two-dimensional space attention graph inferred by the CBAM module along the feature space; F'' is the output of adaptive feature optimization by multiplying the attention map and

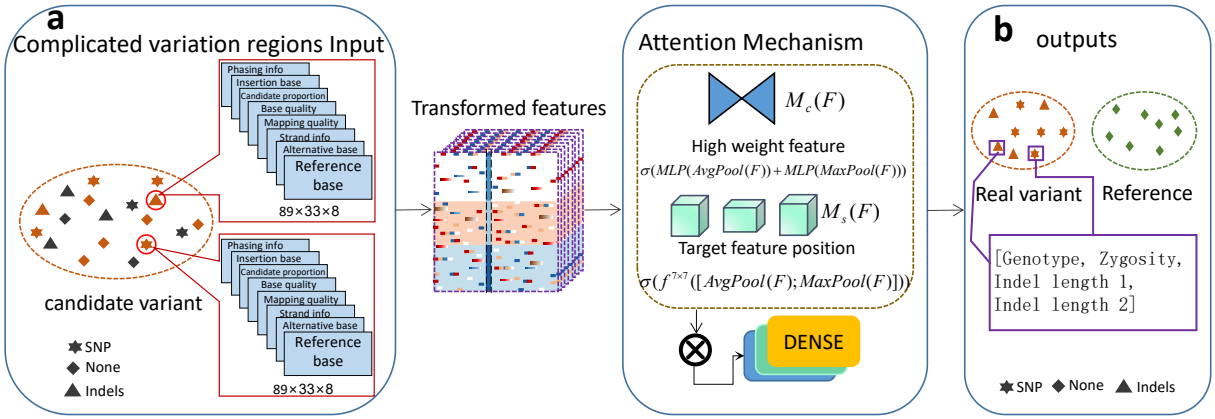

Figure 4: Germline variant calling empowered by spatiotemporal attention.

input feature graph. $M_C(F)$ represents the channel attention, given by

$$\begin{aligned} M_C(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \end{aligned} \tag{4}$$

where σ represents the Sigmoid function, $W_0 \in R^{C/r\times C}$, $W_1 \in R^{C\times C/r}$. The MLP (multi-layer perceptron) is a shared network that shares the weights of W_0 and W_1. $M_S(F)$ represents the spatial attention, and its definition is given by

$$\begin{aligned} M_S(F) &= \sigma(f^{7\times7}([AvgPool(F);MaxPool(F)])) \\ &= \sigma(f^{7\times7}([F^S_{avg};F^S_{max}]) \end{aligned} \tag{5}$$

where 7×7 represents the size of the convolution kernel.

ResNeXt in our ResNeXt+CBAM module was proposed by [Xie *et al.*,2017] which adds an inception idea on top of ResNet [He *et al.*,2016] to widen the network for improving the performance of the network. Data flow by ResNeXt is as follows:

$$y = x + \sum_{i=1}^{C} T_i(x) \tag{6}$$

where x represents the input feature, T_i represents a transformation, such as a series of convolution operations, etc. C is the cardinality of Inception, which represents the network input width.

Detailed architecture of our attention-based ResNeXt is shown in Supplementary Figure 3.

2.4 Multi-scale spatial pyramid pooling

A multi-scale spatial pyramid pooling layer is introduced to remove the restriction on the input size in the network. MSPP is a simple improvement of SPP (Spatial Pyramid Pooling) [He *et al.*,2015]. The input data is passed through four pooling modules, and the specific feature maps in each module are extracted and fused to form a fixed-size matrix, so as to achieve the function of fixed parameters. In this study, the feature map obtained from the ResNeXt-CBAM network needs to be input into the MSPP layer for pooling, and a fixed-length vector is obtained, and then the vector is input into the fully connected layer. After the pooling of MSPP, a fixed $n\times M$-dimensional vector is obtained, where M is the number of blocks after the multi-layer spatial pyramid is merged, and n is the number of convolution kernels in the last layer of the network model. The MSPP network structure is shown in Supplementary Figure 4.

2.5 Model availability and training data

We trained the pileup model and the full-alignment model separately and saved the two models. In the process of testing, one of the two models is called according to the workflow options for the prediction of all the germline variants. The two trained models are provided in Attdeepcaller's github installation website.

All experiments were carried out on a server running Ubuntu 18.04.6 (64-bit) with a 2.40GHz Intel(R) Xeon(R) Silver 4210R (10-core), NVIDIA Tesla P100 PCIe 16GB, 100 GB RAM, and 16 TB disk space. To verify the effectiveness of our model, we used the ONT datasets, PacBio CCS HIFI datasets, and Illumina datasets in the experiment. The experimental network was implemented based on Tensorflow.

The links to the reference genomes, truth variants, benchmarking materials and ONT data are provided in Table 1. More details are provided in Supplementary Table 29 to Supplementary Table 31 in supplementary files. The commands and parameters used in this study are also available in Supplementary. All analysis output, including the VCFs and running logs, are available at https://github.com/shiying-sxu/Attdeepcaller. Source data and code are provided as well.

3 Results

We mainly use misidentification number, together with Precision, Recall and F1-score metrics to evaluate the variant-calling performance. The Precision, Recall and F1-score are computed via hap.py (v0.3.12) [Krusche *et al.*,2019]. For individual evaluation, the benchmark ground truth was constrained in the high-confidence regions provided in GIAB's v3.3.2 or v4.2.1 small variant benchmark.

Table 1: ONT long reads datasets used in the training and testing.

Sample	Reference	BAM	Benchmark VCF	Benchmark BED	Aligner	Coverage	Basecaller
HG001 (Guppy v2.3.8)	GRCh38_no_alt	Guppy v2.3.8	NISTv3.3.2	NISTv3.3.2	minimap2	49.83	Guppy v2.3.8
HG002 (Guppy v3.4.5)	GRCh38_no_alt	Guppy v3.4.5	NISTv3.3.2	NISTv3.3.2	minimap2	52.25	Guppy v3.4.5
HG003 (Guppy v3.2.5)	GRCh38_no_alt	Guppy v3.2.5	NISTv3.3.2	NISTv3.3.2	minimap2	76.85	Guppy v3.2.5
HG004 (Guppy v3.2.5)	GRCh38_no_alt	Guppy v3.2.5	NISTv4.2.1	NIST v4.2.1	minimap2	78.89	Guppy v3.2.5
HG002 (Guppy v5.0.14)	GRCh38_no_alt	Guppy v5.0.14	NISTv4.2.1	NIST v4.2.1	minimap2	117.37	Guppy v5.0.14 (dna_r9.4.1_450bps_sup_prom.cfg)
HG003 (Guppy v5.0.14)	GRCh38_no_alt	Guppy v5.0.14	NISTv4.2.1	NIST v4.2.1	minimap2	78.79	Guppy v5.0.14 (dna_r9.4.1_450bps_hac_prom.cfg)
HG004 (Guppy v5.0.14)	GRCh38_no_alt	Guppy v5.0.14	NISTv4.2.1	NIST v4.2.1	minimap2	79.04	Guppy v5.0.14 (dna_r9.4.1_450bps_sup_prom.cfg)
HG002 (Q20)	GRCh38_no_alt	Q20	NISTv4.2.1	NIST v4.2.1	minimap2	91.18	Q20 (Dorado v4.0.0 SUP_R10.4.1 E8.2)
HG003 (Q20)	GRCh38_no_alt	Q20	NISTv4.2.1	NIST v4.2.1	minimap2	72.51	Q20 (Dorado v4.0.0 SUP_R10.4.1 E8.2)

3.1 Significant reduction of variation misidentifications by Attdeepcaller on ONT Q20 and Guppy 5 datasets

Newer basecalling tools such as Guppy version 5 and Q20 have massively reduced sequencing errors in ONT long reads. However, there are still tens of thousands of complicated variation regions in the alignment data of these quality-improved ONT long reads after aligned to a reference genome. To understand the tremendous roles of spatiotemporal attention played in the accurate detection of gene variations, we trained Attdeepcaller on chr20 from ONT Guppy v5.0.14 dataset $117.37 \times HG002 + 78.79 \times HG003 + 79.04 \times HG004$, and tested the model on chr1 from Q20 dataset and Guppy v5.0.14 dataset. To demonstrate the effect of Attdeepcaller on more complicated datasets (earlier Guppy versions of the data), we also trained Attdeepcaller on ONT dataset $HG001(Guppyv2.3.8) + 60 \times HG001 + HG002(Guppyv3.4.5)$ [Jain *et al.*,2018], and tested the model on benchmark datasets HG003 (Guppy v3.4.5)
[Shafin *et al.*,2015] and HG004(Guppy v3.4.5) [Shafin *et al.*,2015].

For the ONT long reads' alignment data basecalled by Q20, the number of variation misidentification which Attdeepcaller made on chr1 of HG002 was only 22,739, which is 12.69% lower than Clair3's 26,043 misidentifications from the same set of complicated variation regions. Similarly, the number of misidentifications is reduced by 16.49% on chr1 of HG003 and reduced by 23.58% on chr1 of HG004 (see Table 2). Namely for the same set of complicated variation regions, Attdeepcaller outperformed Clair3 on precision, recall, and F1-score by 3%, 1%, and 2%, respectively for the Guppy v5.0.14 test dataset (from chr1). For the Q20 test datasets (from chr1), Attdeepcaller is slightly inferior to Clair3 on precision (less than 1%), but better than Clair3 on recall and F1-score by 2% and 1%, respectively (see Table 3). We note that Clair3 was re-trained on the same dataset as our Attdeepcaller for a fair performance comparison. The total numbers for the complicated variation regions and simple variation regions on chr1 of ONT HG002 are show in Table 4.

On the simple variation regions, Attdeepcaller has reached an almost perfect prediction performance (up to 99.94% precision and 99.99% recall), and Clair3 made similar performance. These results once again illustrate that the challenge in the area of germline variation detection is how to reduce misidentification rates for the complicated variation regions because detection of the simple cases has reached nearly 100% precision and recall. Merging the performance from the two situations as one measurement may hide the challenge of gene variation detection because the number of simple variation regions is much bigger than complicated variation regions.

We trained and tested on the same datasets using the newest PEPPER (v0.8), the result shows that on the complicated regions of ONT long reads' alignment data basecalled by Guppy 5 on chr1 of HG002, our Attdeepcaller made only 23,186 misidentifications, 3.98% lower than PEPPER's 24,147 misidentifications; similarly, Attdeepcaller made 23,122 misidentifications, 6.19% lower than PEPPER's 24,649 misidentifications on HG003. Namely, Attdeepcaller outperforms the newest version of PEPPER as well. Detailed comparisons are presented in Supplementary Table 1 to Table 3 (HG002) and Table 5 to 7(HG003). More details are provided at other tables from Supplementary Table 1 to Supplementary Table 10.

As another example, we applied the trained models to the chr20 data sets of Guppy v5.0.14 and to those of the Q20 versions of HG002, HG003, and HG004. The performance of our Attdeepcaller is consistently good as those on chr1. More details of the performance are provided in Supplementary Table 11 to Supplementary Table 16 in the supplementary files. More performance comparison between Attdeepcaller and Clair3 are also presented at Supplementary Table 17 to Supplementary Table 19.

The above achieved results are all those where the trained models were tested on only Chr1, thus we tested these models on the other chromosomes' long reads alignment data for a more comprehensive understanding of their performance (not on chromosome 20 because it is the training data). As shown in Fig.5, Attdeepcaller outperforms both Clair3 and PEPPER in recall and F1, further demonstrating the effectiveness of the Attdeepcaller method.

Table 2: The numbers of variation misidentifications on the complicated variation regions. (FP: False Positive indicates the number of false positive samples. FN: False Negative: the number of positive samples missed.)

Basecalling	Data	Training data	Trained model	Test Region	Overrall FP+FN	SNPs FP+FN	INDELs FP+FN
Q20	HG002_chr1 HG003_chr1 HG004_chr1	Chr20 from 117.37× HG002+78.79× HG003 +79.04× HG004(Guppy v5.0.14)	Attdeepcaller	complicated variation regions	22739	2349	20390
			Clair3	complicated variation regions	26043	4005	22038
			Attdeepcaller	complicated variation regions	23511	2944	20567
			Clair3	complicated variation regions	28154	4655	23499
			Attdeepcaller	complicated variation regions	27152	4883	22269
			Clair3	complicated variation regions	35528	6861	28667
Guppy v5.0.14	HG002_chr1 HG003_chr1 HG004_chr1	Chr20 from 117.37× HG002+78.79× HG003 +79.04× HG004(Guppy v5.0.14)	Attdeepcaller	complicated variation regions	23186	2898	20288
			Clair3	complicated variation regions	36480	6280	30200
			Attdeepcaller	complicated variation regions	23122	3012	20110
			Clair3	complicated variation regions	34211	5781	28430
			Attdeepcaller	complicated variation regions	23569	2996	20573
			Clair3	complicated variation regions	34364	5548	28816
Guppy v3.2.5	HG003 HG004	HG001(Guppy v2.3.8)+60× HG001 +HG002(Guppy v3.4.5)	Attdeepcaller	complicated variation regions	395346	169778	225568
			Clair3	complicated variation regions	649556	384676	264880
			Attdeepcaller	complicated variation regions	868568	354350	514218
			Clair3	complicated variation regions	1542439	917947	624492

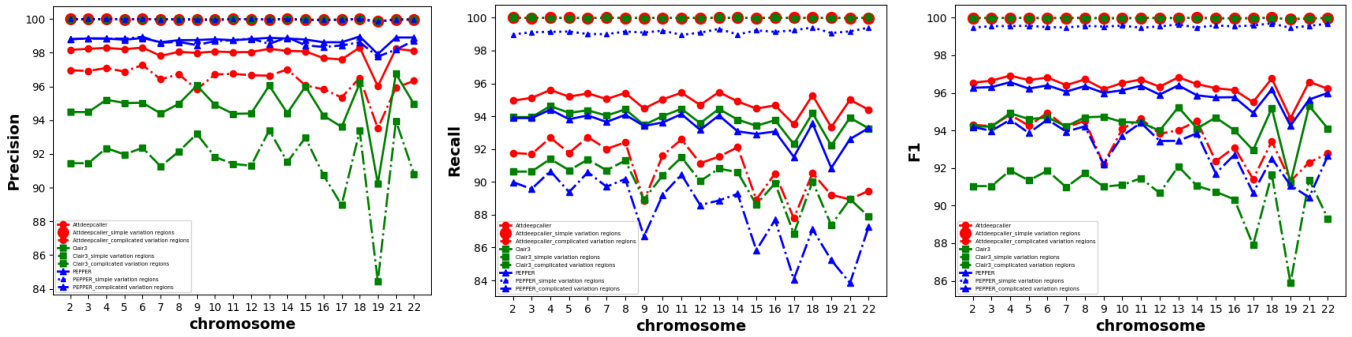

Figure 5: Calling performance on all other chromosomes (HG002 child).

Table 3: Calling performance comparisons on complicated variation regions and on simple variation regions (Chr1 from HG004). The 'precision', 'recall', 'F1-score' all in percentage are written in order, separated by slashes. Training data: Chr20 from 117.37× **HG002**+78.79× **HG003**+79.04× **HG004(Guppy v5.0.14).**

Trained model	Test	Performance (%)					
		on complicated variation regions			on simple variation regions		
		overall	on SNPs	on INDELs	overall	on SNPs	on INDELs
Attdeepcaller	HG004_chr1 Guppy v5.0.14	95.91/89.21/92.44	98.90/98.60/98.75	84.49/62.19/71.65	99.96/99.99/99.97	99.96/100/99.98	100/97.53/98.75
	HG004_chr1 Q20	97.83/85.23/91.10	99.74/96.26/97.97	88.52/51.84/65.39	99.98/99.99/99.99	99.98/100/99.99	99.51/99.30/99.41
Clair3	HG004_chr1 Guppy v5.0.14	92.85/88.61/90.68	97.58/98.70/98.14	69.78/51.95/59.56	99.98/100/99.99	99.98/100/99.99	99.92/99.75/99.83
	HG004_chr1 Q20	98.89/82.43/89.91	98.98/96.52/97.73	97.56/26.63/41.84	99.97/99.99/99.98	99.98/100/99.99	99.68/99.81/99.75

3.2 Significant reduction of variation misidentifications by Attdeepcaller on Guppy 3 datasets (more complicated datasets)

We conducted further comparative analysis between Attdeepcaller and Clair3 on the old versions of guppy data which has much higher degrees of sequencing errors and confusion noise than the Q20 and Guppy 5 datasets. The overall performance of Attdeepcaller and Clair3 on benchmark datasets HG001 (NA12878) [Jain *et al.*,2018] and HG002 (NA24385) [Jain *et al.*,2018], where the performance on HG003 and HG004 are included for more comprehensive comparison (Supplementary Table 18). Again, Attdeepcaller has made significant numbers of variation misidentifications reduced from Clair3's detection on these Guppy v3 data sets, namely Attdeepcaller made a jump of about 16% precision and 10% recall performance improvements. This also indicates that Attdeepcaller can work better when the data becomes more complicated (the older

the version of basecalling, the higher the base error rate of the sequencing data and the more complicated the alignment data).

3.3 Variation detection performance on PacBio CCS HiFi sequencing data

Moreover, Attdeepcaller was tested on PacBio CCS HIFI sequencing data, and made an outstanding performance similarly as the best algorithm Clair3 and DeepVariant did for SNP detection, but made slightly inferior performance on Indels detection. On Illumina short-read sequencing datasets, our method is slightly better than Clair3, and both of them have made exceptionally good performance. Details of these comparison results are presented provided in Supplementary Table 20 to Supplementary Table 25 in supplementary files including speed performance.

3.4 Verification of our detected germline variations

We specially analyzed those variant calls made by Attdeepcaller on HG002 (ONT reads basecalled by Guppy 2.3.4) which are excluded by the GIAB benchmark regions released in 2017 (version3.3.2) [Zook *et al.*,2016] and we validated 17 data regions of those variant calls by Sanger sequencing before v4 benchmark for HG002 was made available. Ahsan et al. [Ahsan *et al.*,2021] have deciphered Sanger sequencing results, identified 41 novel variants (25 SNPs, 10 insertions, and 6 deletions), as shown in Supplementary Table 26. We used multiple methods to detect variations on different versions of ONT HG002 reads, and evaluated the 41 novel variants. On the older version of ONT HG002 reads (version 2.3.4), Medaka [medaka,2019] correctly identified 8 SNPs, 3 insertions, and 1 deletion; and Clair identified 8 SNPs, 2 insertions, and 1 deletion, whereas Longshot [Edge *et al.*,2019] correctly identified 8 SNPs. With much improvement, Attdeepcaller was able to correctly identify 15 SNPs, 4 insertions, and 2 deletions. In particular, 6 of these 15 SNPs, 1 of these 4 insertions and 1 of these 2 deletions were not called correctly by the other variant callers. On a newly produced ONT HG002 reads (version 3.3.2), Attdeepcaller correctly identified 17 SNPs, 6 insertions, and 4 deletions; and Clair3 identified 18 SNPs, 4 insertions, and 3 deletions. In more details, Attdeepcaller correctly detected a deletion at chr3:5336477 (GCA→G), an insertion at chr20:11064574 (A→ATTTTCAAGACTATTGTGACTATGAC) and an insertion at chr12: 100940063(A→AT). These two insertions are correctly identified by Attdeepcaller only but missed by the other variant callers. This performance improvement is mainly attributed to the spatiotemporal double-attention mechanism that enables Attdeepcaller to detect full-alignment sub-cubes of strong resilience to the confusion effect from the sequencing errors.

Supplementary Table 26 also shows some novel variants in the *HG*002 genome, missing in v3.3.2 benchmark variants, as discovered by Sanger sequencing together with the prediction information by Attdeepcaller and other variant callers using ONT reads basecall with Guppy 2.3.4. Attdeepcaller was trained on the ONT $HG001(Guppyv2.3.8) + 60 \times HG001 + HG002$ (Guppy 3.4.5) basecalled reads.

4 Discussion on different types of spatiotemporal attention

In the theory of neural networks, the attention mechanism is to assign various weights for the feature map using some network layers and carries out the attention mechanism on the feature map. Spatiotemporal attention is a double attention mechanism which specializes in the transformation of 3-dimensional data cubes for relevant feature extraction.

There are other choices of attention mechanisms. For example, SAM (Spatial Attention Module) [Woo *et al.*,2018] generates the spatial attention feature map by analyzing the relationships within the feature map space. SAM focuses on the "where" of the useful information on the feature map. CAM (channel attention module) [Woo *et al.*,2018] generates channel attention feature maps through the recognition of relationships between the features. Each channel in the feature map is treated as a feature detector, so the channel feature focuses on the "what" of the useful information in the image. SE (Squeeze-and-excitation networks) [Hu *et al.*,2018] has key operations including squeeze and excitation. Using automatic learning, an extra neural network is used to obtain the importance of each channel in the feature graph and then assign a weight to each feature, so that the neural network focuses on only some feature channels. ECA (Efficient channel attention networks) [Wang *et al.*,2020] is an improved version of SE by using the 1×1 convolution layer directly after the global averaging pooling layer, removing the fully connected layer. This module avoids dimension reduction and can effectively capture cross-channel interactions. ECA works well with only a few parameters.

These attention mechanisms integrated with convolutional neural networks focus more on the separate analysis of channel domains or spatial domains. CBAM introduces spatial attention and channel attention to realize a sequential attention structure from channel to space. The spatial attention can make the neural network pay more attention to the pixel regions that play a decisive role in the classification of the image while ignoring the unimportant regions. The channel attention is used to deal with the distribution relationship of the channels in the feature map. At the same time, the attention allocation of the two dimensions enhances the effect of the attention mechanism on the model performance.

To understand the performance of different attention mechanisms, we applied each of them to the Attdeepcaller model as five different architectures: CBAM, SAM, CAM, SE, ECA. We trained a model for each of the architectures on ONT $HG001(Guppyv2.3.8) + 600 \times HG001 + HG002$ (Guppy 3.4.5) basecalled reads, and tested these models on ONT datasets HG003 and HG004. The performance is shown in Supplementary Figure 5. CBAM has the best effect on the germline variation detection. More details are provided in Supplementary Table 27 to Supplementary Table 28 in the supplementary file.

5 Conclusion

We introduced Attdeepcaller, a deep learning method based on a spatiotemporal attention mechanism, which can effectively improve the accuracy of variation detection, especially for the accurate variation detection from complicated regions. Attdeepcaller outperformed state-of-the-art tools for germline variant calling under

Table 4: The total numbers for the complicated variation regions and simple variation regions on chr1 from ONT HG002. The 'FP', 'FN', 'TP' are written in order, separated by slashes. Training data: Chr20 from 117.37× **HG002+**78.79× **HG003+**79.04× **HG004(Guppy v5.0.14). (TP: True Positive indicates the number of positive samples correctly identified. FP: False Positive indicates the number of false positive samples. FN: False Negative: the number of positive samples missed.)**

Data	Trained model	Test Region	Overrall FP/FN/TP	SNPs FP/FN/TP	INDELs FP/FN/TP
HG002_chr1 Guppy v5.0.14	Attdeepcaller	ALL	6441/16832/290000	1471/1517/262626	4970/15315/27374
		complicated variation regions	6354/16832/141732	1381/1517/114726	4973/15315/27006
		simple variation regions	96/5/148263	96/0/147900	0/5/363
	Clair3	ALL	16337/20185/286647	4506/1818/262325	11831/18367/24322
		complicated variation regions	16295/20185/164337	4462/1818/141351	11833/18367/22986
		simple variation regions	55/8/122302	54/1/120973	1/7/1329
HG002_chr1 Q20	Attdeepcaller	ALL	3810/18943/287889	702/1669/262474	3108/17274/25415
		complicated variation regions	3796/18943/142538	680/1669/118298	3116/17274/24240
		simple variation regions	44/11/145340	42/0/144176	2/11/1164
	Clair3	ALL	3203/22869/283963	1714/2318/261825	1489/20551/22138
		complicated variation regions	3174/22869/165416	1687/2318/145941	1487/20551/19475
		simple variation regions	52/9/118538	43/0/115884	9/9/2654

all of the recall, precision and F1 metrics. We conducted further comparative analysis for variant calling on the old guppy versions. For the Guppy v3.4.5 datasets, Attdeepcaller boosted the precision by a jump of 16 percent and improved the recall by 10 percent. This suggests that Attdeepcaller can still work and can work substantially better when the data becomes more complicated such as the older the versions of basecalling Guppy datasets, where the base erring rate of the sequencing data is higher.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (No.U21A20513, No.62476157, No.62076154), the Key R&D Program of Shanxi Province (202202020101003), and the National Innovation Fellowship Program of the MOST of China (E327130001).

References

[Marx,2023] Marx V (2023). Method of the year: long-read sequencing. *Nature Methods*, **20(1)**, 6-11.

[Logsdon *et al.*,2020] Logsdon G A, Vollger M R, Eichler E E (2020). Long-read human genome sequencing and its applications.*Nature Reviews Genetics*,**21(10)**, 597-614.

[Shendure *et al.*,2017] Shendure J, Balasubramanian S, Church G M, et al (2017). DNA sequencing at 40: past, present and future. *Nature*, **550(7676)**, 345-353.

[Wick *et al.*,2019] Wick R R, Judd L M, Holt K E (2017). Performance of neural network basecalling tools for Oxford Nanopore sequencing.*Genome biology*, **20**,1-10.

[Zheng *et al.*,2022] Zheng Z, Li S, Su J, et al (2022). Symphonizing pileup and full-alignment for deep learning-based long-read variant calling. *Nature Computational Science*, **2(12)**, 797-803.

[Poplin *et al.*,2018] Poplin R, Chang P C, Alexander D, et al (2018). A universal SNP and small-Indel variant caller using deep neural networks. *Nature Biotechnology*, **36(10)**, 983-987.

[Luo *et al.*,2019] Luo R, Sedlazeck F J, Lam T W, et al (2019). A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nature Communications*, **10(1)**, 998.

[Luo *et al.*,2020] Luo R, Wong C L, Wong Y S, et al (2020). Exploring the limit of using a deep neural network on pileup data for germline variant calling. *Nature Machine Intelligence*, **2(4)**, 220-227.

[Ahsan *et al.*,2021] Ahsan M U, Liu Q, Fang L, et al (2021). NanoCaller for accurate detection of SNPs and Indels in difficult-to-map regions from long-read sequencing by haplotype-aware deep neural networks. *Genome biology*, **22**, 1-33.

[Shafin *et al.*,2021] Shafin K, Pesout T, Chang P C, et al (2021). Haplotype-aware variant calling with PEPPER-Margin-DeepVariant enables high accuracy in nanopore long-reads. *Nature Methods*,**18(11)**, 1322-1332.

[He *et al.*,2016] He K, Zhang X, Ren S, et al (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*. **2016**, 770-778.

[Xie *et al.*,2017] Xie S, Girshick R, Dollár P, et al (2017). Aggregated residual transformations for deep neural networks, *Proceedings of the IEEE conference on computer vision and pattern recognition*.**2017**, 1492-1500.

[Woo *et al.*,2018] Woo S, Park J, Lee J Y, et al (2018). Cbam: Convolutional block attention module. *Proceedings of the European conference on computer vision (ECCV)*. **2018**, 3-19.

[Zhang *et al.*,2015] Zhang S, Zheng D, Hu X, et al (2015). Bidirectional long short-term memory networks for relation classification. *Proceedings of the 29th Pacific Asia conference on language, information and computation*. **2015**, 73-78.

[Shafin *et al.*,2015] Shafin K, Pesout T, Lorig-Roach R, et al (2015). Efficient de novo assembly of eleven human genomes using PromethION sequencing and a novel nanopore toolkit. *BioRxiv*, **2019**, 715722.

[Jain *et al.*,2018] . Jain M, Koren S, Miga K H, et al (2018). Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature biotechnology*, **36(4)**, 338-345.

[Zook *et al.*,2016] Zook J M, Catoe D, McDaniel J, et al (2016). Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data*, **3(1)**, 1-26.

[medaka,2019] . medaka (2019). https://github.com/nanoporetech/medaka.**2019**, 11.

[Edge *et al.*,2019] Edge P, Bansal V (2019). Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing. *Nature communications*, **10(1)**, 4660.

[Patterson *et al.*,2016] Patterson M, Marschall T, Pisanti N, et al (2015). WhatsHap: weighted haplotype assembly for future-generation sequencing reads. *Journal of Computational Biology*, **22(6)**, 498-509.

[Krusche *et al.*,2019] Krusche P, Trigg L, Boutros P C, et al (2019). Best practices for benchmarking germline small-variant calls in human genomes. *Nature Biotechnology*, **37(5)**, 555-560.

[Ghiasi *et al.*,2018] Ghiasi G, Lin T Y, Le Q V (2018). Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, **2018**,31.

[He *et al.*,2015] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, **37(9)**, 1904-1916.

[Hu *et al.*,2018] Hu J, Shen L, Sun G (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*. **2018**, 7132-7141.

[Wang *et al.*,2020] Wang Q, Wu B, Zhu P, et al (2020).ECA-Net: Efficient channel attention for deep convolutional neural networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. **2020**,11534-11542.

Received 26 June 2024; revised 12 July 2024; accepted 25 September 2024