NanoCaller for accurate detection of SNPs and small indels from long-read sequencing by deep neural networks

Umair Ahsan[1,#], Qian Liu[1,#], Kai Wang[1,2*]

[1] Raymond G. Perelman Center for Cellular and Molecular Therapeutics, Children's Hospital of Philadelphia, Philadelphia, PA 19104, USA

[2] Department of Pathology and Laboratory Medicine, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA

[#] These authors contributed equally to this work.

[*] To whom correspondence should be addressed. Email: wangk@email.chop.edu

Abstract

Background: Variant detection from high-throughput sequencing data remains an important, unresolved yet often overlooked problem. Long-read sequencing technologies, such as Oxford Nanopore and PacBio sequencing, present unique advantages to detect SNPs and small indels in genomic regions that short-read sequencing cannot reliably examine (for example, only ~80% of genomic regions are marked as "high-confidence region" to have SNP/indel calls in the Genome In A Bottle project). However, existing software tools for short-read data perform poorly on long-read data; instead, several recent studies showed promising results in variant detection on long-read data by deep learning.

Methods: Here we present NanoCaller, a computational method that integrates haplotype structure in deep convolutional neural network for the detection of SNPs/indels from long-read sequencing data. NanoCaller uses long-range information to generate predictions for each candidate variant site by considering pileup information of other candidate sites sharing reads. Subsequently, it performs read phasing and carries out local realignment on each set of phased reads to call indels.

Results: We evaluate NanoCaller on multiple human genomes (NA12878/HG001, NA24385/HG002, NA24149/HG003, NA24143/HG004 and HX1), by cross-genome, cross-chromosome, cross-reference genome, and cross-platform benchmarking tests. Our results demonstrate that NanoCaller performs competitively against other long-read variant callers. In particular, NanoCaller can generate SNP/indel calls in complex genomic regions that are removed from variant calling by other software tools.

Conclusions: In summary, NanoCaller enables the detection of genetic variants from genomic regions that are previously inaccessible to genome sequencing, and may facilitate the use of long-read sequencing in finding disease variants in human genetic studies.

Introduction

Single-nucleotide polymorphisms (SNPs) and small insertions/deletions (indels) are two common types of genetic variants in human genomes. They contribute to genetic diversity and critically influence phenotypic differences, including susceptibility to human diseases. The detection (i.e. "calling") of SNPs and indels are thus a fundamentally important problem in using the new generations of high-throughput sequencing data to study genome variation and genome function. A number of methods have been designed to call SNPs and small indels on Illumina short-read sequencing data. Short reads are usually 100-150 bp long and have per-base error rate less than 1%. Variant calling methods on short reads, such as GATK [1] and FreeBayes [2], achieved excellent performance to detect SNPs and small indels in genomic regions marked as "high-confidence regions" in various benchmarking tests[3-5]. However, since these methods were developed for short-read sequencing data with low per-base error rates and low insertion/deletion errors, they do not work well with long-read sequencing data. Additionally, due to inherent technical limitations of short-read sequencing, the data cannot be used to call SNPs and indels in complex or repetitive genomic regions; for example, only ~80% of genomic regions are marked as "high-confidence region" to have SNP/indel calls in the Genome In A Bottle (GIAB) project, suggesting that ~20% of the human genome is inaccessible to conventional short-read sequencing technologies to find variants reliably.

Oxford Nanopore [6] and Pacific Biosciences (PacBio) [7] technologies are two leading long-read sequencing platforms, which have been rapidly developed in recent years with continuously decreased costs and continuously improved read length, in comparison to Illumina short-read sequencing technologies. Long-read sequencing techniques can overcome several challenging issues that cannot be solved using short-read sequencing, such as calling long-range haplotypes, identifying variants in complex genomic regions, identifying variants in coding regions for genes with many pseudogenes, sequencing across repetitive regions, phasing of distant alleles and distinguishing highly homologous regions [8]. To date, long-read sequencing techniques have been successfully used to sequence genomes for many species to powerfully resolve various challenging biological problems such as de novo genome assembly [9-13] and SV detection [14-19]. However, the per-base accuracy of long reads is much lower with base calling errors of 3-15% [20] compared with short-read data. The high error rate challenges widely-used variant calling methods (such as GATK [1] and FreeBayes [2]), which were previously designed for Illumina short reads and cannot handle reads with higher error rates. As more

and more long-read sequencing data become available, there is an urgent need to detect SNPs and small indels to take the most advantage of long-read data.

Several recent works aimed to design accurate SNP/indel callers on long-read sequencing data using machine learning methods, especially deep learning-based algorithms. DeepVariant [21] is among the first successful endeavors to develop a deep learning variant caller for SNPs and indels across different sequencing platforms (i.e. Illumina, PacBio and Nanopore sequencing platforms). In DeepVariant, local regions of reads aligned against a variant candidate site were transformed into an image representation, and then a deep learning framework was trained to distinguish true variants from false variants that were generated due to noisy base calls. DeepVariant achieved excellent performance on short reads as previous variant calling methods did. Later on, Clairvoyante [22] and its successor Clair [23] implemented variant calling methods using deep learning, where the summary of aligned reads around putative candidate sites were used as input of deep learning framework. The three deep learning-based methods can work well on both short-read and long-read data, but haplotype structure is not incorporated in variant calling yet, and the published version of DeepVariant and Clairvoyante have limited ability to identify multiallelic variants where two different alternative alleles are present simultaneously. Recently, LongShot [24] was published where pair-Hidden Markov Model (pair-HMM) was used to call and phase SNPs on long-read data, with excellent performance in the benchmarking tests. However, Longshot cannot identify indels. It is also worth noting that (1) HiFi reads after circular consensus sequencing on PacBio long-read sequencing [25] or similar methods on the Nanopore platform can potentially improve the detection of SNPs/indels by adapting existing short-read variant callers, due to its much lower per-base error rates. However, HiFi reads would substantially increase sequencing cost given the same base output, so it is more suitable for specific scenarios such as capture-based sequencing; (2) the Oxford Nanopore company also recently released a SNP/indel caller, i.e. medaka [26], using deep learning on long-read data; however, the details on medaka are not yet available. In summary, although several methods for variant detection on long-read sequencing data have become available, there may be room in further improving these approaches. We believe that improved SNP/indel detection on long read data will enable widespread research and clinical applications of long-read sequencing techniques.

In this study, we propose a deep learning framework, NanoCaller, which integrates haplotype structure in deep convolutional neural network to improve the detection of SNPs and small indels on long-read sequencing data. In NanoCaller, candidate SNP sites are defined according to observed reference and

alternative alleles as well as allele frequency, and then pileup of a candidate site is built with its tens of neighboring heterogeneous candidate sites with shared long reads, and then fed into a deep convolutional neural network for SNP calling. For indel calling, multiple sequencing alignment of a group of long reads aligned against a candidate indel site is performed to categorize long reads into two groups, and then indel calling is performed from the consensus of long reads from each group. We evaluate NanoCaller on several human genomes, such as NA12878 (HG001), NA24385 (HG002), NA24149 (HG003), NA24143 (HG004) and HX1 with both Nanopore and PacBio long-read data. Our preliminary evaluation demonstrated competitive performance of NanoCaller against existing tools, especially in generating SNP/indel calls in complex genomic regions that are removed from variant calling by other software tools.

Methods

Datasets

Long-read data

Five long-read data sets for human genomes are used for the evaluation of NanoCaller. The first genome is NA12878, whose Oxford Nanopore Technology (ONT) rel6 FASTQ files were downloaded from WGS consortium database[9], and aligned to the GRCh38 reference genome using minimap2 [27]; PacBio alignment files for NA12878 were downloaded from the GIAB database [28, 29]. The second genome is NA24385 (HG002) whose alignment files for both ONT and PacBio data were downloaded from the GIAB database [28, 29]. The third and fourth genome are NA24143 (HG004) and NA24149 (HG003), mother and father of NA24385, whose ONT FASTQ files were downloaded from the GIAB database [28, 29] and aligned to the GRCh38 reference genome using minimap2. PacBio alignments files for these two genomes were also downloaded from the GIAB database [28, 29]. The fifth genome is HX1 which was sequenced by us using PacBio [10] and Nanopore sequencing [30]. The long-read data was aligned to the GRCh38 reference genome using minimap2. Table 1 shows the statistics of mapped reads in the five genomes. Coverage shown is defined as number of mapped bases divided by the reference genome length.

Benchmark variant calls

The benchmark set of SNPs and indels (version 3.3.2) for the first four genomes (NA12878, NA24385, NA24149 and NA24143) were download from the Genome in a Bottle (GIAB) Consortium [28] together with high-confidence regions for each genome. There are 3,004,071, 3,079,462, 2,927,639, and 2,953,590 SNPs for NA12878, NA24385, NA24149 and NA24143 respectively, and 483,630, 475,332, 471,156, and 481,114 indels for them, as shown in Table 2. Benchmark variant calls for HX1 were generated by using GATK on Illumina 300X reads sequenced by us [10] (Table 2).

Variant Calling

The procedures of NanoCaller for variant calling is shown in Figure 1 where candidate sites of SNPs and indels are defined according to the input alignment file and reference file. Then, pileups are generated for each candidate site and fed into a convolutional neural network with haplotype structure for SNP calling. To call indels, local multiple sequence alignment of phased reads from called SNPs is used to generate consensus sequence for indels. The details are described below.

Candidate site selection

Candidate sites of SNPs and indels are defined according to the depth and alternative allele frequency for a specific genomic position. In NanoCaller, SAMtools mpileup [31] is used to generate aligned bases against each genomic positions. The alternative allele frequency for a specific genomic position is calculated as the fraction of reads supporting the base. A genomic position is considered as a candidate site if the total read depth, and the allele frequency for some alternative allele are both above certain thresholds. Both minimum alternative allele frequency and minimum read depth can vary for different datasets and can be specified by the user depending on coverage and base calling error rate of the genome sequencing data. By default, candidate sites with alternative allele frequency between 30% and 70%, denoted by V, are designated as highly likely heterozygous SNP sites, and are used to create input images for candidate sites.

Image input of candidate site for convolutional neural network

The image of each candidate site is generated using the procedures below as shown in Figure 2. For a candidate site b:

1. We select sites from the set V that share at least one read with b and are at most 20,000bp away from b.

2. In each direction, upstream or downstream, of the site b, we choose the closest 20 sites to the candidate site b. If there are less than 20 such sites, we just append the final image with zeros. We denote the set of these potential heterozygous SNP sites nearby b (including b) by Z. An example is shown in Figure 2 (a).
3. The set of reads covering b is divided into four groups, $\mathrm{R}_B = \{\text{reads that support base B at b}\}$, $\mathrm{B} \in \{\mathrm{A}, \mathrm{G}, \mathrm{T}, \mathrm{C}\}$. Reads that do not support any base at b are not used.
4. For each read group in R_B with supporting base B, we count the number ($\mathrm{C}^{\mathrm{t}}_{\mathrm{BD}}$) of supporting reads for site $t \in$ Z with base $\mathrm{D} \in \{\mathrm{A}, \mathrm{G}, \mathrm{T}, \mathrm{C}\}$ (As shown in Figure 2(b)). Then, we normalize the number of supporting reads using

   $\mathrm{F}^{\mathrm{t}}_{\mathrm{BD}} = \frac{\mathrm{C}^{\mathrm{t}}_{\mathrm{BD}}}{\mathrm{C}^{\mathrm{t}}_{\mathrm{B}}} \cdot g(\mathrm{D})$ where g(D)=1 if D is not the reference base at site t, and -1 otherwise, and

   $$\mathrm{C}^{\mathrm{t}}_{\mathrm{B}} = \sum_{\mathrm{D} \in \{\mathrm{A,G,T,C}\}} \mathrm{C}^{\mathrm{t}}_{\mathrm{BD}} = \textit{reads suporting base B at site b and some base at site t}$$

   An example is shown in Figure 2(c).
5. We obtain a 4x41x4 matrix M with entries $[\mathrm{F}^{\mathrm{t}}_{\mathrm{BD}}]_{\mathrm{B,t,D}}$ (as shown Figure 2(d)) where the first dimension corresponds to base B at site b, second dimension corresponds to site t, and the third dimension corresponds to base D at site t. Our image has read groups as rows, columns as various base positions, and has 4 channels, each recording frequencies of different bases in the given read group at the given site.
6. We add another channel to our image which is a 4x41 matrix $[\mathrm{Q}^{\mathrm{t}}_{\mathrm{B}}]_{\mathrm{B,t}}$ where $\mathrm{Q}^{\mathrm{t}}_{\mathrm{B}} = 1$ if B is the reference base at site b and 0 otherwise (as shown in (as shown Figure 2(d)). In this channel, we have a row of ones for reference base at b and rows of zeroes for other bases.
7. We add another row to the image which encodes reference bases of site in Z, and the final image is illustrated in as shown Figure 2(e).

Convolutional neural network architecture

In NanoCaller, a convolutional neural network [32] takes pileup images as input and estimates four independent probability estimates for the presence of each base type at a specific reference position. NanoCaller uses three convolutional layers with Scaled Exponential Linear Unit (SELU) activation units followed by two different full connection layers for SNP calling as shown in Figure 3. In NanoCaller, the first layer uses kernels of three different dimensions and combines the convolved features into one a single output: one capture local information from a row, another from a column and the other from a

2D local region. The second and third layers use kernels of size 2x3. The output from third layer is flattened and used as input of fully connected layer of 48 nodes with dropout (using 0.5 drop date). After this layer, on one hand, we use another fully connected layer with 16 nodes to estimate probability estimates for each base type at a specific position; on the other hand, we use another fully connected layer feeds into 16 nodes together with the estimated probability of each base type to make probability estimates for zygosity, which is used only in the training phase to propagate errors backwards for incorrect zygosity predictions.

This model has 107,122 parameters, a significantly lower number than Clairvoyante[23] (1,631,496) and DeepVariant (23,885,392), all of which are initiated by Xavier's method [33]. We also apply L2-norm regularization, with coefficient 0.001, to prevent overfitting of the model.

Generating variant calls

In NanoCaller, four base probabilities P(A), P(G), P(T) and P(C) are predicted to determine zygosity of the candidate sites. If a candidate site has two bases with probabilities exceeding 0.5, it is considered to be heterozygous. For heterozygous sites, two bases with highest probabilities are chosen with a heterozygous variant call. For homozygous sites, only the highest probability base is chosen: if that base is not the reference allele, a homozygous variant call is made, otherwise, it is homozygous reference. Each of called variants is also assigned with a quality score which is calculated by $-10\log_{10} Probability\ (no\ variant)$ and recorded as a float in QUAL field of the VCF file to indicate the chance of false positive prediction: the larger the score is, the less likely that the prediction is wrong.

Haplotyping and SNP refinement

To refine our SNP calls, if the number of SNP calls in a 2,000bp region centered at a given variant call exceeds a fixed threshold we discard that variant call (while the default value is 25, these parameters were chosen arbitrarily but are user-adjustable parameters based on the characteristics of the long-read sequencing data set). This step eliminates clusters of false positive calls arising from poorly aligned reads: a cluster of variant calls that has a high correlation between variants and the reads supporting them but may not be true variants. Afterwards we use Whatshap [34] to phase variants, allowing the algorithm to re-genotype variants. In the phasing step, all variant calls made by NanoCaller, both in high-confidence regions and outside confidence regions are used. Any candidate site whose genotype

changes to homozygous reference is considered as non-variant call. As a final step we remove variants with quality score less than or equal to a certain threshold (default is 5).

Local realignment and indel calling

To call indels, we use Whatshap to phase reads using the predicted SNPs in NanoCaller, and then infer indels. In this phasing step, we use all variant calls made by NanoCaller, both in high-confidence regions and outside confidence regions. Long reads aligned against an indel candidate site usually are grouped into two clusters, each corresponding to a haplotype. For each haplotype, we call indels using the steps below, and then merge haploid indel calls from both haplotypes to issue a diploid call.

Given an indel candidate site b, (i) if there is a mononucleotide sequence of length four in reference sequence in the 11 bp window centered at *b* then pass; Otherwise, calculate insertion frequency at *b* as the proportion of reads with an insertion beginning right after *b*, and calculate deletion frequency at *b* as the proportion of reads with a deletion beginning right after *b* or a deletion at *b*. (ii) If insertion frequency or deletion frequency are above certain thresholds, we consider *b* as a candidate site. We use a higher threshold for deletion to account for high deletion errors in Nanopore reads. (iii) After extracting sequences that span a window of fixed size around *b* from each read in the haplotype, we use MUSCLE [35] multiple sequence alignment algorithm for DNA sequences on the extracted sequences. (iv) We then obtain consensus sequence of the realigned sequences by choosing the symbol from the set with highest frequency at each base position. We subtract 0.2 from frequency of deletion, which is the random noise found when we calculate frequency of deletion after multiple sequence alignment at sites with no indels. (v) Then, we use BioPython's pairwise2 local alignment algorithm with affine gap penalty to compare the consensus sequence with reference sequence around the site *b*. We obtain the alternative allele by observing any sequence insertion in consensus sequence or deletion in the reference sequence after pairwise alignment. (vi) if an indel is called at *b*, no other variant calls will be made in the region less than 10bp ahead of *b*.

Performance measurement

The performance of SNP/indel calling by a variant caller is evaluted against the benchmark variant tests. Several measurements of performance evaluation are used, such as precision (p), recall (r) and F1 score as defined below.

$$p = \frac{TP}{TP + FP}$$
$$r = \frac{TP}{TP + FN}$$
$$F1 = \frac{2 * p * r}{p + r}$$

where TP is the number of benchmark variants correctly predicted by a variant caller, and FP is the number of miscalled variants which are not in benchmark variant sets, FN is the number of benchmark variants which cannot be called by a variant caller. $F1$ is the weighted average of p and r, a harmonic measurement of precision and recall. The range of the three measurements is $[0, \quad 1]$: the larger, the better.

Results

Overview of NanoCaller

NanoCaller takes an alignment file of a sequencing data set as input and generates a VCF file for called SNPs and indels. There are several steps in NanoCaller (Figure 1). First, candidate sites of SNPs and indels are generated according to the minimum coverage and minimum frequency of alternative alleles. Then, a set of heterogeneous candidate variants are used to build a pileup for a candidate site with a normalization process (Figure 2). After that, a deep convolutional network (Figure 3) is used to distinguish true variants from false candidate sites with a refined and phased process. The predicted SNPs are then used to phased long reads for indel calling. On each of two sets of phased long reads, multiple sequencing alignment for a local region around a candidate indel site are used to call sequences which are deletions or insertions.

SNP detection performance on 5 human genomes

We evaluate NanoCaller under several strategies for SNP calling: cross-genome testing, cross-chromosome testing, cross-reference genome testing, and cross-platform testing. In the evaluation, NanoCaller is evaluated on the GRCh38 reference genome by default, and evaluation is performed for "gold standard" variants on high-confidence regions unless stated otherwise. RTG tools (the vcfeval submodule) [36] is used to calculate various evaluation measurements, such as precision, recall and F1.

For each variant calling method, we obtain the thresholds of quality scores with the best performance on different genomes, and then report the performance with the averaged threshold of quality scores.

Cross-genome testing strategy

Cross-genome testing means that a model is trained on a genome and then tested on different genomes. This testing strategy demonstrates how a well-trained model works on various genomes which are not used for training, and the performance under this strategy is a good measurement when a variant calling tool is used in real-world scenario. In this study, we present performance of three NanoCaller models trained on whole genome sequencing data with benchmark variant sets on HG001 (Nanopore reads), HG002 (Nanopore reads) and HG003 (PacBio reads), and then tested on other genomes. The SNP performance results on Nanopore reads under this strategy are shown in Figure 4(a), (b), (c), together with the performance of the other existing methods including medaka, Longshot and Clairvoyante. Figure 4 (e), (f) and (g) show SNP performance results on PacBio reads, together with performance of Longshot and Clairvoyante. It is clearly shown from Figure 4(a) and (e) that all methods have excellent precision with greater than 95% for majority of the methods and datasets (excluding HX1 where the precision is not calculated for high-confidence regions), while the recall ranges from 90% to 97%. In Figure 4(c) and (g), NanoCaller demonstrates better or competitive performance on HG002, HG003 and HG004 against all other methods. Since HX1 is not part of GIAB and a "high confidence" region list is not available from GIAB, we created high confidence regions for HX1 by removing low complexity regions [37] from the GRCh38 reference genome.

Cross-chromosome testing strategy

Cross-chromosome testing indicates that the data from a chromosome is excluded from training a model but used for testing the performance of the trained model. In this study, chromosome 1 is used for testing and the other chromosomes from the same genome are used for training. The results under this strategy is shown in Figure 4 (a), (b) and (c) where the performance of "NanoCaller 1" on HG001 and of "NanoCaller 2" on HG002 is based on cross-chromosome testing evaluation. Comparison with cross-genome performance of "NanoCaller 1" and "NanoCaller 2" suggests that cross-genome testing appears to have better performance, potentially indicating less overfitting in NanoCaller.

Cross-reference genome testing strategy

Cross-reference genome testing means that a model is trained on a genome with a specific reference genome but tested on another genome mapped to a different reference genome. In this study, we train NanoCaller on HG001 mapped to the GRCh38 reference genome, but test the NanoCaller model on HG002 mapped to the GRCh37 reference genome. The performance is shown in Figure 4 (d) where the cross-reference genome has slightly lower performance, but the overall performance is still excellent. This indicates that NanoCaller could be used on alignment files generated by mapping to different reference genomes.

Cross-platform testing strategy

In cross-platform testing, NanoCaller is trained on a sequencing platform such as Nanopore, but tested on a different sequencing platform such as PacBio, or vice versa. In Figure 4 (e), (f) and (g), we compared the performances of a NanoCaller model trained on HG001 Nanopore reads and another NanoCaller model trained on HG003 PacBio reads. Both models show similar results and perform better than Clairvoyante on each genome, and competitively against Longshot.

Illustrative examples of SNPs called by NanoCaller

It is worth noting that NanoCaller is able to accurately call multiallelic SNPs and SNPs which are missed by other existing tools. An example of a multiallelic SNP with two different alternative alleles called by NanoCaller is shown in Figure 5. This multiallelic SNP at chr1:58128619 (T>A and T>G) in HG002 genome is in the benchmark variant set. When tested on ONT reads, this SNP was correctly called by NanoCaller (model trained on HG001) but was missed by every other tool. We examined this genomic position further: the reference base is T, and the two alternative alleles are A and G, while the reference base is the last base in a poly-T sequence of length 5 in the reference genome. As shown in Figure 5, there are 55 reads in ONT data and the number of reads supporting each base are $\{T\colon 22, A\colon 17, G\colon 14, deletion\colon 2\}$, with allele frequencies $\{T\colon 40\%, A\colon 31\%,\ G\colon 25\%\}$. All of these reads are phased by Whatshap using SNP calls from NanoCaller, as shown in Figure 5 with both phased reads of Nanopore and PacBio long reads. In one set of phased reads, the numbers of reads supporting each base are $\{G\colon 13, T\colon 12, A\colon 1\}$, whereas in the other set of phased reads, the numbers are $\{A\colon 16, T\colon 10, G\colon 1, deletion\colon 2\}$. NanoCaller was able to make a correct SNP call at this site, despite the high frequency of reference allele. High frequency of reference allele at the candidate site can be attributed to base calling

error due to poly-T sequence, and explains why other variant caller might have missed this SNP. Medaka's call for this variant is a heterozygous SNP with two alternative alleles (T>TG and T>A); Clairvoyante's calls for this variant is a heterozygous SNP with one alternative allele (T>G) and a homozygous deletion at chr1:58128618 (CT>C) since Clairvoyante does not support multiallelic calls. Clair's call for this variant is a heterozygous SNP with one alternative allele (T>G). Longshot's call for this variant is a heterozygous SNP with one alternative allele(T>A).

Figure 6 shows a novel SNP called by NanoCaller, using both Nanopore and PacBio reads, which is not in the benchmark variant set from GIAB, and is outside the GIAB high confidence region. This novel SNP is at HG002 chr6:131901995 with reference allele C and alternative allele G. Medaka, Clair and Clairvoyante were also able to detect this SNP, however Longshot was unable to detect it. In Figure 6 (a), we show phased ONT and PacBio reads aligned around the SNP site in the range chr6:131901647-131902127. We can see 13 other SNPs in phased reads with the novel SNP that are called by NanoCaller. However, GIAB's ground truth calls for HG002 do not have any variant call in the region chr6:131901800-131902100. In Figure 6 (b), we show a zoomed in version of Figure 6 (a) spanning the region chr6:131901976-131902016. We can see three other SNPs: chr6:131901984 (ref: C; alt: T), chr6:131901994 (ref: A; alt: G) and chr6:131901997 (ref: C; alt: T), that are called by NanoCaller and are in the same haplotype as the novel SNP. In Figure 6 (c), we show phased ONT and PacBio reads of HG004, mother of HG002 for the same region as in (b). The four SNPs shown in (c), including the novel SNP site, are recorded as a haplotype in GIAB ground truth calls for HG004. This gives a strong piece of evidence that the novel SNP would be true, since we detect the same set of four SNPs forming a haplotype in HG002 calls by NanoCaller. We note that Longshot was able to call SNPs at chr6:131901984 and chr6:131901994, but not at chr6:131901995 or chr6:131901997.

Indel calling performance

To call indels in a test genome, we use NanoCaller to detect SNPs no matter whether the SNPs are in high-confidence regions, and then the heterogeneous SNP calls are used to phased long reads for indel calling. To avoid overfitting, NanoCaller model trained on HG002 is used to detect SNPs for indel calling on HG001, and NanoCaller model trained on HG001 is used to detect SNPs for indel calling on the rest of the genomes. The indel performance, evaluated by RTG *vcfeval*, is shown in Figure 4 (h), (i) and (j) together with the performance of Medaka and Clairvoyante for benchmarking indels within GIAB high confidence regions. Compared with medaka, NanoCaller demonstrates better performance on HG002,

HG003 and HG004, but slight worse on HG001. On all genomes, NanoCaller achieves much better performance than Clairvoyante. Although NanoCaller shows superior overall performance, the performance on indels is still much lower than for SNPs and there may still be room for future improvements.

Discussion

In this study, we present NanoCaller, a deep learning framework to detect SNPs and small indels from long-read sequencing data. Depending on library preparation and sequencing techniques, long-read data usually have much higher error rates than short-read sequencing data, which substantially challenged precise variant calling and thus stimulated the development of error-tolerant deep learning methods for accurate variant calling. However, the benefits of much longer read length of long-read sequencing are not fully exploited for variant calling in previous studies. The NanoCaller tool that we present here integrates haplotype structure in deep convolutional neural network for the detection of SNPs/indels from long-read sequencing data, and uses multiple sequence alignment to re-align candidate sites for indels, to improve the performance of variant calling. Our results by cross-genome testing, cross-chromosome testing, cross-reference genome testing, and cross-platform testing demonstrate that NanoCaller performs competitively against other long-read variant callers.

One specific advantage of NanoCaller is that we generate pileup of candidate variants from haplotyped set of neighboring heterogeneous variants, each of which is shared by a long read with the candidate site. Given a long read with >20kb, there are on averagely >20 heterogeneous sites, and evidence of SNPs from the same long reads can thus improve SNP calling by deep learning. Evaluated on several human genomes with benchmarking variant sets, NanoCaller demonstrates competitive performance against existing variant calling methods on long reads and with phased SNPs. NanoCaller is also able to make accurate predictions cross sequencing platforms and cross reference genomes. In this study, we have tested NanoCaller models trained on Nanopore data for PacBio long-read data and achieved similar prediction performance. We also tested NanoCaller models trained on GRCh38 for GRCh37 and achieve the same level SNP calling performance. Furthermore, with the advantage of long-read data on repetitive regions, NanoCaller is also able to detect SNPs/indels outside high-confidence regions which cannot be reliably detected by short-read sequencing techniques, and thus provide more candidate

SNPs/indels sites for investigating causal variants on undiagnosed diseases where no disease-causal candidate variants were found by short-read sequencing.

NanoCaller has also flexible design to call multi-allelic variants, which Clairvoyante and Longshot cannot handle. In NanoCaller, the probability of each nucleotide type is assessed separately, and it is allowed that the probability of 2 or 3 or 4 nucleotide type is larger than 0.5 or even close to 1.0 after normalization, and thus suggests strong evidence for a specific position with multiple bases in a test genome. Therefore, NanoCaller can easily generate multi-allelic variant calls, where both alternative alleles differ from the reference allele. Furthermore, NanoCaller can be easily configured to call variants for species with polyploidy or somatic mosaic variants when data are available to train an accurate model. Meanwhile, NanoCaller uses normalized statistics to generate pileup for a candidate site, and normalized statistics is independent on the coverage of a test genome, and thus, NanoCaller is able to handle a test data set with different coverage from the training data set, which might be a challenge for other long-read callers. That is, NanoCaller trained on a whole-genome data has less biases on other data sets with much lower or higher coverage, such as target-sequencing data with thousands folds of coverage. We also note that even with very accurate HiFi reads (<1% error rate) generated by PacBio, in principle, NanoCaller is likely to yield better variant calling performance over short-read based variant callers, because NanoCaller integrates haplotyped long-range information to improve variant calling.

There are several limitations of NanoCaller that we wish to discuss here. First, NanoCaller relies on the accurate alignment and pileup of long-read sequencing data, but incorrect alignments in low-complexity regions might still occur, complicating the variant calling process. Both continuingly improved sequencing techniques and improved alignment tools can benefit NanoCaller with better performance. But if the data is targeted at very complicated regions or aligned with very poor mapping quality, the performance of NanoCaller would be affected. Another limitation of NanoCaller is that the indel detection from mononucleotide repeats might not be accurate, especially on Nanopore long-read data which has difficulty in the basecalling of homopolymers [38, 39]. In Nanopore long-read basecalling process, it is challenging to determine how many repeated nucleotides for a long consecutive array of similar Nanopore signals, potentially resulting in false indel calls at these regions, which can be post-processed from the call set.

In summary, we propose a deep-learning tool using long-range haplotype information and local multiple sequence alignments for accurate SNP/indel calling. Our evaluation on several human genome suggests

that NanoCaller performs competitively against other long-read variant callers, and can generate SNPs/indels calls in complex genomic regions that are removed from variant calling by other software tools. NanoCaller enables the detection of genetic variants from genomic regions that are previously inaccessible to genome sequencing, and may facilitate the use of long-read sequencing in finding disease variants in human genetic studies.

Acknowledgements

The authors would like to thank members of the Wang lab for valuable comments and feedback. We would like to thank GIAB and nanopore-wgs-consortium for providing the sequencing data sets and the gold standard variant call data for use in our evaluation. This study is in part supported by NIH/NIGMS grant GM132713 to KW.

Competing Interests

The authors declare no competing interests.

Author contributions

UA and QL developed the computational method and drafted the manuscript. UA implemented the software tool and evaluated its performance. KW conceived the study, advised on model design and guided implementation/evaluation. All authors read, revised, and approved the manuscript.

References

1. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M *et al*: **The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data**. *Genome Res* 2010, **20**(9):1297-1303.

2. Garrison E, G. M: **Haplotype-based variant detection from short-read sequencing**. *arXiv* 2012, **1207.3907**.
3. Krusche P, Trigg L, Boutros PC, Mason CE, De La Vega FM, Moore BL, Gonzalez-Porta M, Eberle MA, Tezak Z, Lababidi S *et al*: **Best practices for benchmarking germline small-variant calls in human genomes**. *Nature biotechnology* 2019, **37**(5):555-560.
4. Zook JM, McDaniel J, Olson ND, Wagner J, Parikh H, Heaton H, Irvine SA, Trigg L, Truty R, McLean CY *et al*: **An open resource for accurately benchmarking small variant and reference calls**. *Nature biotechnology* 2019, **37**(5):561-566.
5. Cameron DL, Di Stefano L, Papenfuss AT: **Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software**. *Nature communications* 2019, **10**(1):3240.
6. Branton D, Deamer DW, Marziali A, Bayley H, Benner SA, Butler T, Di Ventra M, Garaj S, Hibbs A, Huang X *et al*: **The potential and challenges of nanopore sequencing**. *Nature biotechnology* 2008, **26**(10):1146-1153.
7. Eid J, Fehr A, Gray J, Luong K, Lyle J, Otto G, Peluso P, Rank D, Baybayan P, Bettman B *et al*: **Real-time DNA sequencing from single polymerase molecules**. *Science (New York, NY)* 2009, **323**(5910):133-138.
8. Mantere T, Kersten S, Hoischen A: **Long-Read Sequencing Emerging in Medical Genetics**. *Frontiers in genetics* 2019, **10**:426.
9. Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, Tyson JR, Beggs AD, Dilthey AT, Fiddes IT *et al*: **Nanopore sequencing and assembly of a human genome with ultra-long reads**. *Nat Biotechnol* 2018, **36**(4):338-345.
10. Shi L, Guo Y, Dong C, Huddleston J, Yang H, Han X, Fu A, Li Q, Li N, Gong S *et al*: **Long-read sequencing and de novo assembly of a Chinese genome**. *Nat Commun* 2016, **7**:12065.
11. Pendleton M, Sebra R, Pang AW, Ummat A, Franzen O, Rausch T, Stutz AM, Stedman W, Anantharaman T, Hastie A *et al*: **Assembly and diploid architecture of an individual human genome via single-molecule technologies**. *Nat Methods* 2015, **12**(8):780-786.
12. Seo JS, Rhie A, Kim J, Lee S, Sohn MH, Kim CU, Hastie A, Cao H, Yun JY, Kim J *et al*: **De novo assembly and phasing of a Korean human genome**. *Nature* 2016, **538**(7624):243-247.
13. Cho YS, Kim H, Kim HM, Jho S, Jun J, Lee YJ, Chae KS, Kim CG, Kim S, Eriksson A *et al*: **An ethnically relevant consensus Korean reference genome is a step towards personal reference genomes**. *Nature communications* 2016, **7**:13637.
14. Stephens Z, Wang C, Iyer RK, Kocher JP: **Detection and visualization of complex structural variants from long reads**. *BMC bioinformatics* 2018, **19**(Suppl 20):508.
15. Heller D, Vingron M: **SVIM: structural variant identification using mapped long reads**. *Bioinformatics (Oxford, England)* 2019, **35**(17):2907-2915.
16. Jiang T, Liu B, Jiang Y, Li J, Gao Y, Cui Z, Liu Y, Wang Y: **Long-read-based Human Genomic Structural Variation Detection with cuteSV**. *bioRxiv* 2019:780700.
17. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, Schatz MC: **Accurate detection of complex structural variations using single-molecule sequencing**. *Nat Methods* 2018, **15**(6):461-468.
18. Fang L, Hu J, Wang D, Wang K: **NextSV: a meta-caller for structural variants from low-coverage long-read sequencing data**. *BMC Bioinformatics* 2018, **19**(1):180.
19. Gong L, Wong CH, Cheng WC, Tjong H, Menghi F, Ngan CY, Liu ET, Wei CL: **Picky comprehensively detects high-resolution structural variants in nanopore long reads**. *Nat Methods* 2018, **15**(6):455-460.

20. Ameur A, Kloosterman WP, Hestand MS: **Single-Molecule Sequencing: Towards Clinical Applications**. *Trends in biotechnology* 2019, **37**(1):72-85.
21. Poplin R, Chang PC, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT *et al*: **A universal SNP and small-indel variant caller using deep neural networks**. *Nature biotechnology* 2018, **36**(10):983-987.
22. Luo R, Sedlazeck FJ, Lam TW, Schatz MC: **A multi-task convolutional deep neural network for variant calling in single molecule sequencing**. *Nature communications* 2019, **10**(1):998.
23. Luo R, Wong C-L, Wong Y-S, Tang C-I, Liu C-M, Leung HCM, Lam T-W: **Clair: Exploring the limit of using a deep neural network on pileup data for germline variant calling**. *bioRxiv* 2019:865782.
24. Edge P, Bansal V: **Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing**. *Nature communications* 2019, **10**(1):4660.
25. Wenger AM, Peluso P, Rowell WJ, Chang PC, Hall RJ, Concepcion GT, Ebler J, Fungtammasan A, Kolesnikov A, Olson ND *et al*: **Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome**. *Nature biotechnology* 2019, **37**(10):1155-1162.
26. **medaka: Sequence correction provided by ONT Research** [https://github.com/nanoporetech/medaka]
27. Li H: **Minimap2: pairwise alignment for nucleotide sequences**. *Bioinformatics (Oxford, England)* 2018, **34**(18):3094-3100.
28. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, Weng Z, Liu Y, Mason CE, Alexander N *et al*: **Extensive sequencing of seven human genomes to characterize benchmark reference materials**. *Scientific Data* 2016, **3**(1):160025.
29. **GENOME IN A BOTTLE** [https://jimb.stanford.edu/giab]
30. Liu Q, Fang L, Yu G, Wang D, Xiao CL, Wang K: **Detection of DNA base modifications by deep recurrent neural network on Oxford Nanopore sequencing data**. *Nat Commun* 2019, **10**(1):2449.
31. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R: **The Sequence Alignment/Map format and SAMtools**. *Bioinformatics* 2009, **25**(16):2078-2079.
32. Krizhevsky A, Sutskever I, Hinton GE: **ImageNet classification with deep convolutional neural networks**. *Commun ACM* 2017, **60**(6):84-90.
33. Glorot X, Bengio Y: **Understanding the difficulty of training deep feedforward neural networks**. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.* Edited by Yee Whye T, Mike T, vol. 9. Proceedings of Machine Learning Research: PMLR; 2010: 249--256.
34. Patterson M, Marschall T, Pisanti N, van Iersel L, Stougie L, Klau GW, Schonhuth A: **WhatsHap: Weighted Haplotype Assembly for Future-Generation Sequencing Reads**. *Journal of computational biology : a journal of computational molecular cell biology* 2015, **22**(6):498-509.
35. Edgar RC: **MUSCLE: multiple sequence alignment with high accuracy and high throughput**. *Nucleic acids research* 2004, **32**(5):1792-1797.
36. Cleary JG, Braithwaite R, Gaastra K, Hilbush BS, Inglis S, Irvine SA, Jackson A, Littin R, Rathod M, Ware D *et al*: **Comparing Variant Call Files for Performance Benchmarking of Next-Generation Sequencing Variant Calling Pipelines**. *bioRxiv* 2015:023754.
37. Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, Salit M: **Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls**. *Nat Biotechnol* 2014, **32**(3):246-251.
38. Rang FJ, Kloosterman WP, de Ridder J: **From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy**. *Genome biology* 2018, **19**(1):90.

39. Zascavage RR, Thorson K, Planz JV: **Nanopore sequencing: An enrichment-free alternative to mitochondrial DNA sequencing**. *Electrophoresis* 2019, **40**(2):272-280.

Tables

Table 1. Whole genome statistics of five data sets on human genomes by Nanopore and PacBio sequencing. Each genome is aligned to the GRCh38 reference genome, and only the mapped reads were used to calculate the statistics. Total number of bases is calculated as the sum of length of all mapped reads, and coverage is defined as number of mapped bases divided by the reference genome length.

Platform	Genome	# reads	# bases	N50	Mean read length	Median read length	Coverage
ONT	HG001	15,666,888	132.9 Gb	13,630	8,485	5,387	43X
ONT	HG002	14,063,218	190.2 Gb	50,930	13,527	2,769	62X
ONT	HG003	22,079,048	250.1 Gb	43,745	11,325	1,974	81X
ONT	HG004	29,319,334	279.6 Gb	46,689	9,536	745	91X
ONT	HX1	20,497,769	271.8 Gb	22,273	13,261	10,123	88X
PacBio	HG002	21,511,145	179.0 Gb	11,264	8,323	7,500	58X
PacBio	HG003	10,564,465	85.3 Gb	10,943	8,078	7,251	28X
PacBio	HG004	10,369,228	83.4 Gb	10,869	8,040	7,137	27X

Table 2. Statistics of ground truth variants in chromosomes 1-22 of each genome aligned to the GRCh38 reference genome. For genomes with GIAB ground truth calls, statistics within the high confidence regions are also given. Statistics for HG002 aligned to the reference genome GRCh37 (hg19) are shown in parenthesis. For HX1, high confidence regions are created by removing low complexity regions from the GRCh38 reference genome.

	Whole genome (chr1-22)			High confidence region (chr1-22)				
Genome	**SNPs**	**Multiallelic SNPs**	**Indels**	**SNPs**	**Multiallelic SNPs**	**Indels**	**Total Length**	**% of genome**
HG001	3,004,071	867	516,524	2,961,527	825	483,630	2,329,784,734	81.03
HG002	3,079,462 (3,100,749)	958 (945)	518,055 (509,449)	3,030,495 (3,048,869)	900 (904)	475,332 (464,463)	2,353,170,731 (2,358,060,765)	81.85 (81.85)
HG003	2,927,639	872	510,622	2,883,686	829	471,156	2,261,821,224	78.67
HG004	2,953,590	863	521,646	2,909,326	815	481,114	2,266,589,952	78.84
HX1	3,282,242	1,125	687,501	2,980,737	858	205,591	2,356,619,870	76.31

Figures

Figure 1. A simplified workflow of NanoCaller in generating SNP and indel calls.

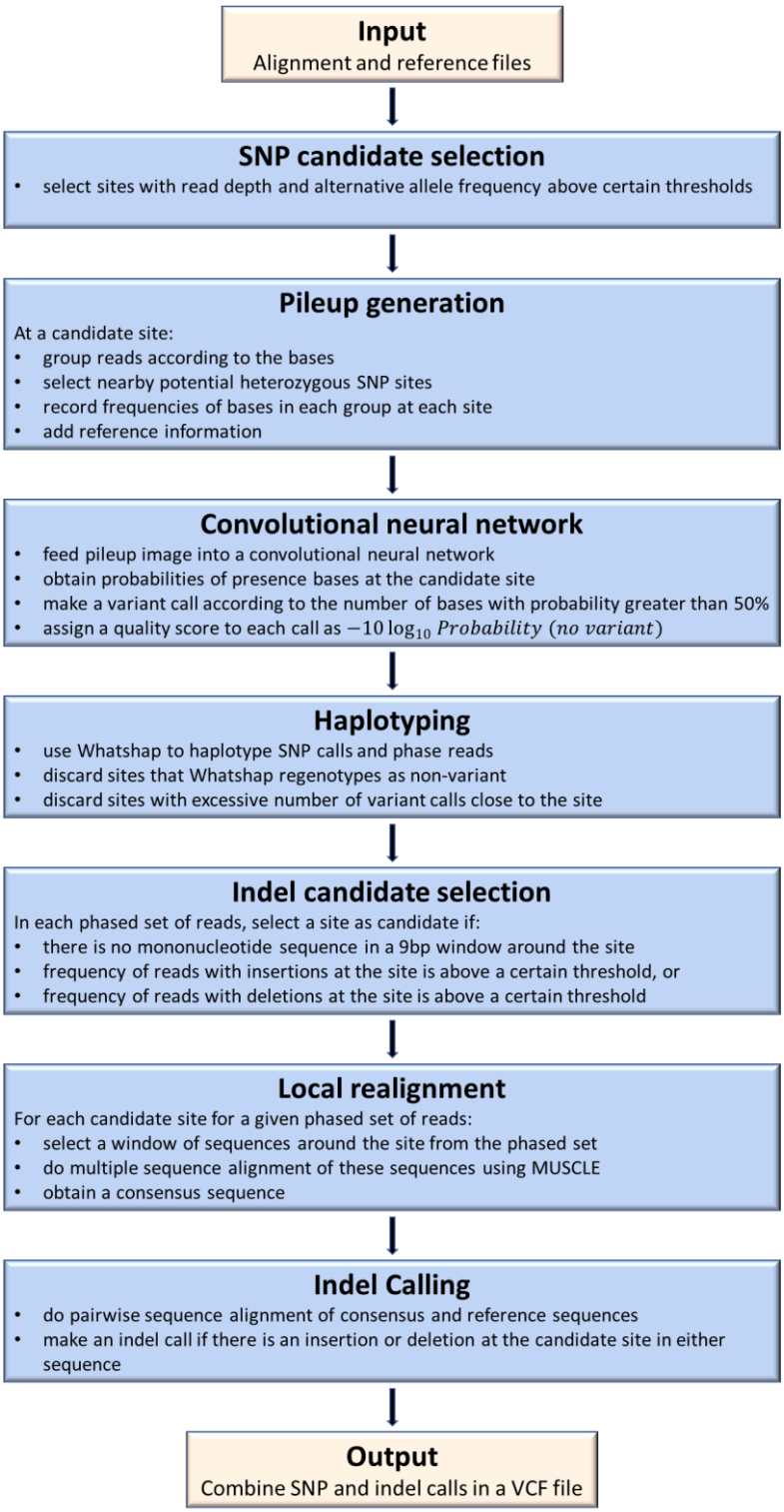

Figure 2 . An example on how to construct image pileup for a candidate site. a) reference sequence and reads pileup at site b and sites in set Z, b) raw counts of bases at sites in Z in each read group, c) frequencies of bases at sites in Z with negative signs for reference bases, d) flattened pileup image with fifth channel and reference sequence row added, e) pileup image used as input for TensorFlow.

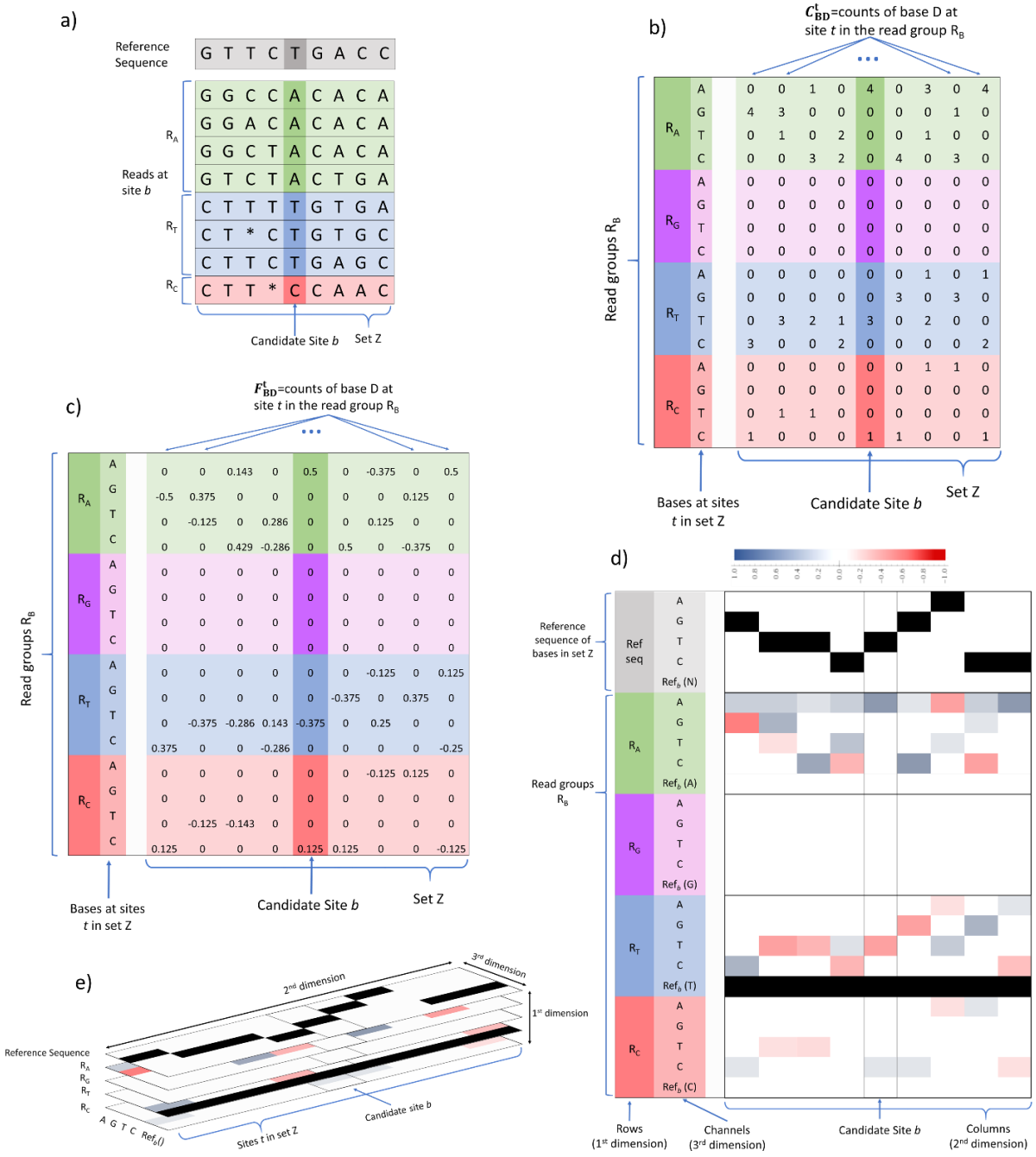

Figure 3. An illustration of the convolutional neural network architecture for NanoCaller. First convolutional layer uses 3 kernels of sizes 1x5, 5x1 and 5x5, whereas the second and third convolutional layers use kernels of size 2x3. Output of third convolutional layer is flattened and is fed to a fully connected layer with 48 nodes. Nodes in this fully connected layer are dropped with 50% probability. Output of this is split into two independent pathways, upper one calculating probabilities of each base and the lower one calculating zygosity probabilities. Zygosity probability is only used in the training process.

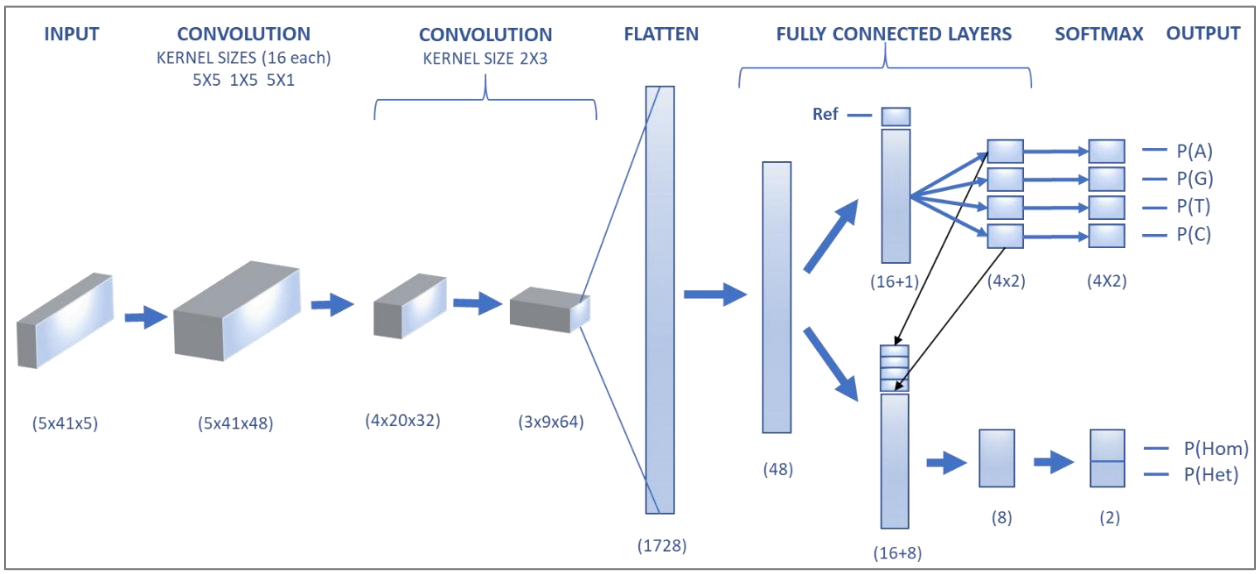

Figure 4. Performance of NanoCaller and other state-of-the-art variant callers on five whole-genome sequencing data sets. SNP performance results on ONT reads: a) SNP precision, b) SNP recall, c) SNP F1 score. d) HG002 (ONT) SNP performance when aligned to GRCh38 vs GRCh37 while NanoCaller is trained on HG001 (ONT) with GRCh38.SNP performance results on PacBio reads: e) SNP precision, f) SNP recall, g) SNP F1 score. Indels performance results on ONT reads: h) Indels precision, i) Indels recall, j) Indels F1 score. NanoCaller 1, NanoCaller 2 and NanoCaller 3 mean NanoCaller models trained on HG001 (ONT), HG002 (ONT), and HG003 (PacBio) respectively. If the genome for training and testing is the same, the chromosome 1 is excluded in training process and used for testing.

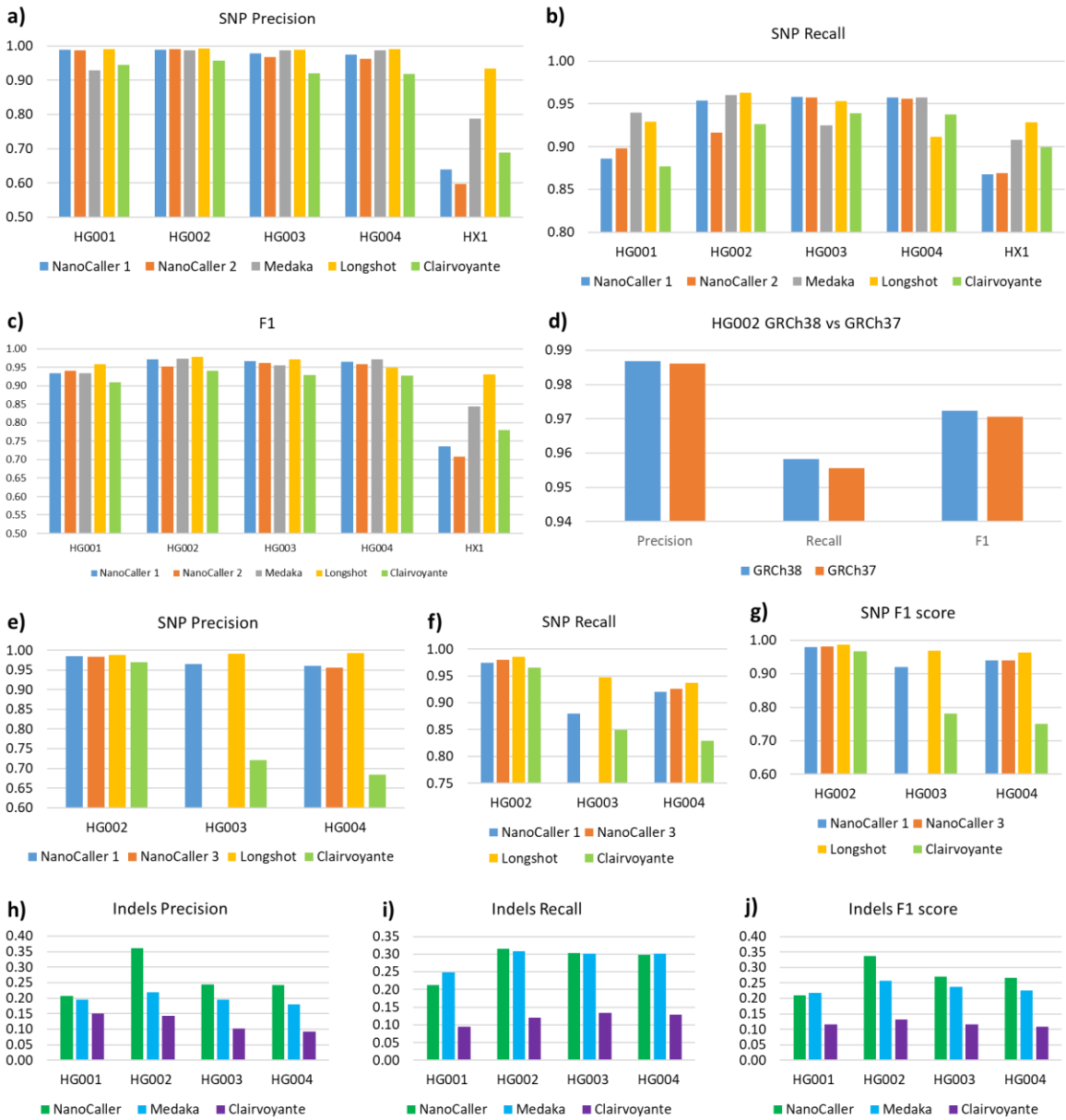

Figure 5. The Integrative Genomics Viewer screen shots for a true multiallelic SNP called by NanoCaller at chr1:58128619 (GRCh38) in HG002 genome. Upper panel shows Nanopore reads and the lower panel show PacBio reads at the SNP site. All the reads are phased by Whatshap using NanoCaller SNP calls. Allele A is shown in green color, and allele G is show in brown.

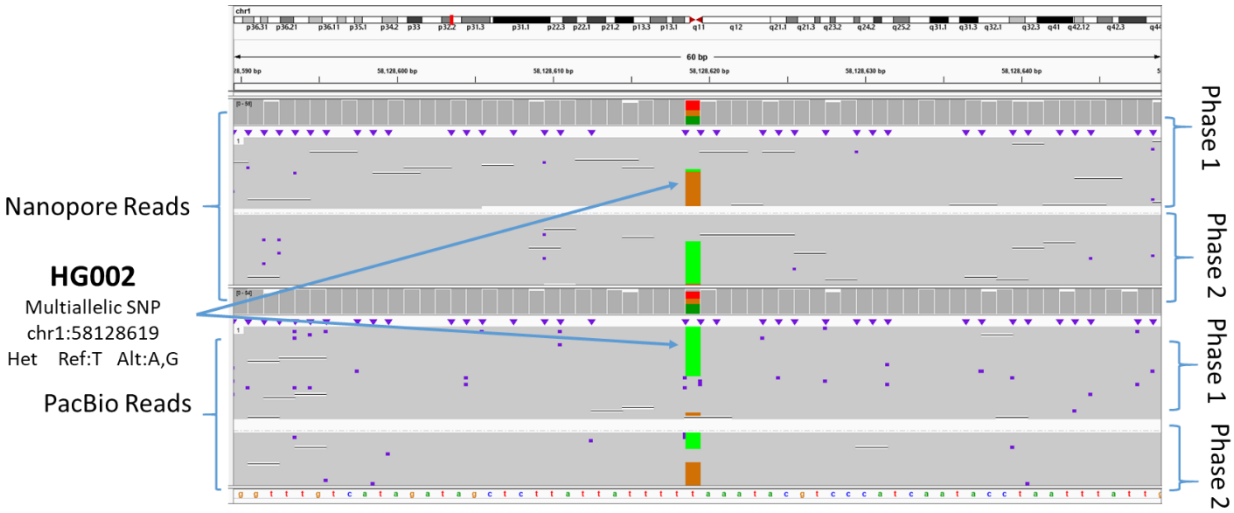

Figure 6. The Integrative Genomics Viewer screen shots for a novel SNP called by NanoCaller at a novel SNP chr6:131901995 (GRCh38) in the HG002 genome. a) shows 479bp around the novel SNP in HG002, with 13 other SNPs called by NanoCaller in the same haplotype in both ONT and PacBio reads. b) shows 41 bp around the novel SNP in HG002. c) shows ONT and PacBio reads for HG004 (mother) for the same region as b) with 4 SNPs from ground truth that form a haplotype. Reads were phased with Whatshap using NanoCaller SNP calls.