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

Umair Ahsan[1,#], Qian Liu[1,#], Li Fang[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 and yet unresolved 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 benchmark SNP/indel calls in the Genome In A Bottle (GIAB) 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 long reads in deep convolutional neural network for the detection of SNPs/indels from long-read sequencing data. NanoCaller uses long-range haplotype structure to generate predictions for each SNP candidate variant site by considering pileup information of other candidate sites sharing reads. Subsequently, it performs read phasing, and carries out local realignment of each set of phased reads and the set of all reads for each indel candidate variant site to generate indel calling, and then creates consensus sequences for indel sequence prediction.

**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. We additionally validate 41 novel variant calls in HG002 (previously not unreported in GIAB gold standard set) by Sanger sequencing, demonstrating the unique advantage of NanoCaller on long-read sequencing in detecting mutations in complex genomic regions that cannot be reliably found previously.

**Conclusions**: In summary, NanoCaller enables the detection of genetic variants from genomic regions that are previously inaccessible to genome sequencing, and may facilitate the combined use of long-read and short-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 endeavor 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 variant detection 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 heterozygous candidate sites with shared long reads, and then fed into a deep convolutional neural network for SNP calling. Predicted heterozygous SNPs are then used to phase alignment reads. For indel calling, multiple sequencing alignment is performed on three groups of long reads aligned against a candidate indel site: entire set of reads and two phased sets of reads. These three locally realigned groups of reads are used to predict indel variant zygosity, and the consensus sequences of long reads from each group of reads are used to infer indel sequences in accordance with zygosity prediction. 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 evaluation demonstrates competitive performance of NanoCaller against existing tools, especially in generating SNP/indel calls in complex genomic regions which cannot be reliably called on short-read data. NanoCaller is publicly available at https://github.com/WGLab/NanoCaller.

# 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 are downloaded from WGS consortium database[9], and aligned to the GRCh38 reference genome using minimap2 [27]; PacBio alignment files for NA12878 are downloaded from the GIAB database [28, 29]. The second genome is NA24385 (HG002) whose alignment files for both ONT and PacBio data are 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 are downloaded from the GIAB database [28, 29] and aligned to the GRCh38 reference genome using minimap2. PacBio alignments files for these two genomes are 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 is aligned to the GRCh38 reference genome using minimap2. Table S1 in *Supplementary Tables* shows the statistics of

mapped reads in the five genomes where the coverage of ONT data ranges from 43 to 91 and the coverage of PacBio data is between 27 and 58.

### Benchmark variant calls

The benchmark set of SNPs and indels (version 3.3.2) for the first four genomes (NA12878, NA24385, NA24149 and NA24143) are 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 1. Benchmark variant calls for HX1 were generated by using GATK on Illumina 300X reads sequenced by us [10] (Table 1).

## NanoCaller framework for variant calling

The procedure of NanoCaller for variant calling is shown in Figure 1 where candidate sites of SNPs and indels are defined according to the input alignment and a reference sequence. Nanocaller has two convolutional neural networks, one for SNP calling and the other for indel prediction, each requiring a different type of input. Input pileup images generated for SNP candidate sites incorporate long-range haplotype information. For indel candidate sites, alignment reads are phased with Whatshap using SNP calls from NanoCaller, and then NanoCaller uses phased alignment reads to generate input pileup images by carrying out local multiple sequence alignment around each site. 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 position.

**SNP candidate sites** are determined using the criteria below. For a genomic site *b* with reference base *R*,

1. Calculate alternative allele frequency $= \frac{\max\{\text{number of reads supporting base B at site } b\}_{\text{B} \in \{\text{A,G,T,C}\} \setminus \text{R}}}{\text{total read depth at site } b}$

2. $b$ is considered a SNP candidate site if the total read depth and the alternative allele frequency are both greater than specified thresholds. We set the alternative allele frequency threshold to be 15%.

**Indel candidate sites** are determined using the criteria below. For a genomic site $b$,

1. Calculate:
    a. For $i \in \{0, 1\}$, $\text{depth}_i = \text{total number of reads in phase } i \text{ at site } b$
    b. Insertion frequency $= \min\{\frac{\text{number of reads in phase } i \text{ supporting insertion at site } b}{\text{depth}_i}\}_{i \in \{0,1\}}$
    c. Deletion frequency $= \min\{\frac{\text{number of reads in phase } i \text{ supporting deletion at site } b}{\text{depth}_i}\}_{i \in \{0,1\}}$
2. $b$ is considered an indel candidate site if:
    a. Both $\text{depth}_0$ and $\text{depth}_1$are greater than a specified depth threshold
    b. Either insertion frequency is greater than a specified insertion frequency threshold or the deletion frequency is greater than a specified deletion frequency threshold.

Thresholds for alternative allele frequency, insertion frequency, deletion frequency and read depths can be specified by the user depending on coverage and base calling error rate of the genome sequencing data.

## Input images of candidate sites for convolutional neural networks

To generate input images of candidate sites for convolutional neural network, we select SNP candidate sites with alternative allele frequency between 30% and 70% and designate them as highly likely heterozygous SNP sites (denoted by $\mathrm{V}$), and then use them as long-range haplotype information to create input images for SNP candidate sites. In detail, the procedure is described below (as shown in Figure 2). For a SNP candidate site $b$:

1. We select sites from the set $\mathrm{V}$ 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 from $\mathrm{V}$ 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, $\mathrm{R}_B = \{\text{reads that support base B at b}\}$, $\mathrm{B} \in \{\mathrm{A, G, T, C}\}$. Reads that do not support any base at $b$ are not used.
4. For each read group in $\mathrm{R}_B$ with supporting base B, we count the number ($\mathrm{C}^{\mathrm{t}}_{\mathrm{BD}}$) of supporting reads for site $t \in$ Z with base $\mathrm{D} \in \{\mathrm{A, G, T, C}\}$ (As shown in Figure 2(b)).
5. Let $\mathrm{F}^{\mathrm{t}}_{\mathrm{BD}} = \mathrm{C}^{\mathrm{t}}_{\mathrm{BD}} * g(D)$, where $\mathrm{g}(D)$ is a function that returns -1 if $D$ is the reference base at site *t* and 1 otherwise. An example is shown in Figure 2(c).
6. 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 nucleotide type B at site *b*, second dimension corresponds to the number of sites *t*, and the third dimension corresponds to nucleotide type 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.
7. 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 Figure 2(d)). In this channel, we have a row of ones for reference base at *b* and rows of zeroes for other bases.
8. We add another row to the image which encodes reference bases of site in Z, and the final image is illustrated in Figure 2(e).

**Input image of indel candidate site** is generated using the procedure below as shown in Figure 3. For an indel candidate site $b$:

1. Denote by $S_{all}$, $S_{phase1}$ and $S_{phase2}$ the set of all reads, reads in a phase and reads in the other phase at site *b*, respectively.
2. Let $seq_{ref}$ be the reference sequence of length 160bp starting at site *b*.
3. For each set $S \in \{S_{all}, S_{phase1}, S_{phase2}\}$, do the following:
    a. For each read $r \in S$, let $seq_r$ be the 160bp long subsequence of the read starting at the site *b*.
    b. Use MUSCLE to carry out multiple sequence alignment of the following set of sequences $\{seq_{ref}\} \cup \{seq_r\}_{r \in S}$ as shown in Figure 3 a).
    c. Let $\{seq'_{ref}\} \cup \{seq'_r\}_{r \in S}$ be the realigned sequences, where $seq'_{ref}$ denotes the realigned reference sequence, and $seq'_r$ denotes realignment of sequence $seq_r$. We truncate all sequences at the length 128.
    d. For $B \in \{A, G, T, C, -\}$ and $1 \leq p \leq 128$, calculate

$$C_{B,p} = \sum_{r \in S} g(seq_r^{'}, B, p)$$

$$M_{B,p} = \frac{C_{B,p}}{\#\ reads\ in\ set\ S}$$

where $g(seq_r^{'}, B, p)$ returns 1 if the base at index $p$ of $seq_r^{'}$ is $B$ and 0 otherwise.

e. Figure 3 c) shows raw counts $C_{B,p}$ for each symbol. Let $M$ be the $5 \times 128$ matrix with entries $M_{B,p}$ as shown in Figure 3 d).

f. Construct a $5 \times 128$ matrix $Q$, with entries $[\mathrm{Q}_{\mathrm{B}}^{\mathrm{p}}]_{\mathrm{B,p}}$, where $\mathrm{Q}_{\mathrm{B}}^{\mathrm{p}} = 1$ if $seq'_{ref}$ has symbol $B$ at index $p$ and 0 otherwise as shown in Figure 3 f). Both matrices $M$ and $Q$ have first dimension corresponding to the symbols $\{A, G, T, C, -\}$, and second dimension corresponding to pileup columns of realigned sequences.

g. Construct a $5 \times 128 \times 2$ matrix $Mat_S$ whose first channel is the matrix $M - Q$ as shown in Figure 3 e) and the second channel is the matrix $Q$.

4. Concatenate the three matrices $Mat_{S_{all}}, Mat_{S_{phase1}}$ and $Mat_{S_{phase2}}$ together to get a $15 \times 128 \times 2$ matrix as input to convolutional neural network.

## Convolutional neural network architectures in NanoCaller

In NanoCaller, we present two convolutional neural networks [32], one for SNP prediction and the other for indel calling, as shown in Figure 4. Both neural networks have three convolutional layers: 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 convolutional layer is flattened and used as input for a fully connected network with dropout (using 0.5 drop date).

For SNP model, the first fully connected layer is followed by two different neural networks of fully connected layers to calculate two types of probabilities. In the first network, we calculate the probability of each nucleotide type B to indicate that B is present at the genomic candidate site; thus for each nucleotide type B, we have a binary label prediction. The second network combines logit output of first network with output of first fully connected hidden layer to estimate probability for zygosity (homozygous or heterozygous) at the candidate site. The second network is used only in the training to propagate errors backwards for incorrect zygosity predictions. During testing, we infer zygosity from the output of first network only.

For indel model, the first fully connected layer is followed by two fully connected hidden layers to produce probability estimates for each of the four zygosity cases at the candidate site: homozygous-reference, homozygous-alternative, heterozygous-reference and heterozygous-alternative (i.e. heterozygous with no reference allele).

These models have 137,678 parameters, a significantly lower number than Clair [23](2,377,818) Clairvoyante[22] (1,631,496). All parameters in NanoCaller are initiated by Xavier's method [33]. We also apply L2-norm regularization, with coefficients 1e-3 and 1e-4 for SNP and indel models respectively, to prevent overfitting of the model.

**Generating variant calls in NanoCaller**

After generating pileup images for SNP calling, NanoCaller by default scales the raw counts of bases in pileup images to adjust for the difference between coverages of the testing genome and the genome used for training of the model selected by user, i.e we replace the counts $C_{B,p}$ shown in Figure 2 (b) by

$$C_{B,p} \times \frac{coverage\ of\ training\ genome}{coverage\ of\ testing\ genome}$$

In order to call SNPs, NanoCaller calculates probabilities of presence of each nucleotide type at the candidate site. If a candidate site has at least two nucleotide types with probabilities exceeding 0.5, it is considered to be heterozygous, otherwise it is regarded as homozygous. For heterozygous sites, two nucleotide types with highest probabilities are chosen with a heterozygous variant call. For homozygous sites, only the nucleotide type with the highest probability is chosen: if that nucleotide type 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 as $-100\log_{10} Probability\ (1 - P(B))$, where $P(B)$ is the probability of the alternative $B$ (in case of multiallelic prediction we choose $B$ to be the alternative allele with smaller probability) 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**.**

For indel calling, NanoCaller calculates the probabilities for four cases of zygosities: homozygous-reference, homozygous-alternative, heterozygous-reference and heterozygous-alternative. No variant call is made if the homozygous-reference label has the highest probability. If homozygous-alternative label has the highest probability, we determine consensus sequence from the multiple sequence alignment of $S_{all}$, and align it against reference sequence at the candidate site using BioPython's

pairwise2 local alignment algorithm with affine gap penalty. Alternative allele is inferred from the indel of the pairwise alignment of the two sequences. In case either of the heterozygous predictions has the highest probability, we use $S_{phase1}$ and $S_{phase2}$ to determine consensus sequences for each phase separately and align them against reference sequence. Indel calls from both phases are combined to make a final indel call at the candidate site.

**Haplotyping and SNP refinement**

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. Afterwards, we use WhatsHap again to phase alignment reads in order to create pileup images for indels. An optional refinement step allows user to filter SNP calls that have excessive number of nearby SNP calls. In this step, we discard a SNP call if the total count of SNP calls in a user specified window around the SNP call exceeds a user specified count threshold.

## 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.

## Sanger validation of 17 selected sites on HG002

To further demonstrate the performance of NanoCaller and other variant callers, we select 17 genomic sites whose SNPs/indels are not in the GIAB ground truth calls (version 3.3.2), and conduct Sanger sequencing for them on HG002. Firstly, we design PCR primers within ~400 bp of a select site of interest and then use a high-fidelity PCR enzyme (PrimeSTAR GXL DNA Polymerase, TaKaRa) to amplify each of the target selected repeat regions. The PCR products are purified using AMPure XP beads and sequenced by Sanger sequencing. We then decipher two sequences from Sanger results for variant analysis. The data and deciphered sequences are in the supplementary files. Please note that more than 17 variant sites are detected in the Sanger results, because PCR regions can contain other variants besides the selected sites.

# Results

## Overview of NanoCaller

NanoCaller takes an alignment 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 are generated according to the minimum coverage and minimum frequency of alternative alleles. Then, a set of heterozygous candidate SNPs are used to build a pileup for a candidate SNP site (Figure 2). After that, a deep convolutional network (Figure 4) is used to distinguish true variants from false candidate sites. The predicted SNPs are phased, as well as the long alignment reads for indel calling. Then, indel candidate sites are selected according to a minimum coverage and insertion/deletion frequency criteria. Pileup images for indels are generated using multiple sequencing alignment on the set of entire reads and on each set of phased reads (Figure 3). Another deep convolutional neural network is designed to determine indel calls, and allele sequence for the indels is predicted by comparing consensus sequences against reference sequence.

## SNP detection performance on 5 human genomes

NanoCaller is evaluated under several strategies for SNP calling: cross-genome testing, cross-chromosome testing, cross-reference genome testing, and cross-platform testing. The evaluation is on the GRCh38 reference genome by default, and performed for benchmark variants on high-confidence

regions unless stated otherwise. RTG tools (the vcfeval submodule) [35] is used to calculate various evaluation measurements, such as precision, recall and F1. For Clair, Clairvoyante and Longshot, we use the recommended quality score cut offs, whereas for NanoCaller and Medaka we obtain quality score cut offs by taking average of cut off thresholds that give best performance on different genomes.

**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 benchmark variant sets of whole genome sequencing data for HG001 (ONT reads), HG002 (ONT reads) and HG003 (PacBio reads), and then test them on other genomes. The performance of SNP calling on Nanopore reads under this strategy are shown in Table 2 and Figure 5 (a), (b), (c), together with the performance of the existing methods including medaka, Longshot and Clairvoyante, while Table 3 and Figure 5 (e), (f) and (g) show the performance of SNP calling on PacBio reads, together with performance of Longshot and Clairvoyante. It is clearly shown from Figure 5 (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 and we created reliable regions on HX1 by removing low complexity regions [36] from the GRCh38 reference genome for performance comparison), while the recall ranges from 90% to 97%. As shown in Figure 5 (c) and (g), NanoCaller demonstrates better or competitive performance on HG002, HG003 and HG004 against all other methods.

**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 are shown in Table 2 and Figure 5 (a), (b) and (c) where the performance of “NanoCaller 1” on HG001 and of “NanoCaller 2” on HG002 (ONT reads) is based on cross-chromosome testing evaluation. Similarly, in Table 3 and Figure 5 (e), (f) and (g), the performance of “NanoCaller 3” on HG003 (PacBio reads) is also 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(ONT reads) mapped to the GRCh37 reference genome. The performance is shown in Table 2 (numbers in parenthesis) and Figure 5 (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 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 Table 3 and Figure 5 (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.

## Indel calling performance

To call indels in a test genome, we use NanoCaller to detect SNPs in both high-confidence and other regions, and then use the heterozygous SNP calls to phase long reads for indel calling. We train NanoCaller indel model on chromosomes 2-22 of HG001 (ONT), and test it on four genomes (HG001, HG002, HG003, HG004) using ONT reads. For HG001, only chromosome 1 is used for testing, whereas chromosomes 1-22 of for other genomes are used for testing. We set insertion and deletion frequency thresholds at 30% and 60% respectively for generating indel pileups. For each genome, we create evaluation regions by removing homopolymer regions of length greater than 5 from GIAB high confidence regions for that genome. The indel performance, evaluated by RTG *vcfeval*, is shown in Table 4 and Figure 5 (h), (i) and (j) together with the performance of Medaka, Clair and Clairvoyante. In this

test, we use the recommended quality cut-off for Clair and Claivoyante, and averaged quality cut-off for NanoCaller and Medaka. The evaluation on four genomes suggests that NanoCaller has much better performance than Clarivoyante and competitive against medaka and Clair.

To understand the variant callers more, we show indel performance of NanoCaller, Medaka and Clair in another two different ways below. On one hand, we evaluate different callers inside non-homopolymer regions with no quality cut-off and only with genomic coordinate match (i.e. any errors in allele sequence or zygosity are ignored). The results are shown in *Table S2 of Supplementary Tables* where NanoCaller performs better than Clair in terms of F1-score and precision, and competitively against Medaka. The comparison of Table 4 and *Table S2 of Supplementary Tables* suggests that NanoCaller need to be improved by better quality scores and allele sequence prediction in future. On the other hand, we provide a detailed breakdown of indel calling performance for NanoCaller, Medaka and Clair according to the characteristic of variants: 1bp deletions or >1 bp deletions, or variants in bi-nucleotide repeats or not. This evaluation is also based on non-homopolymer regions without any quality cut-off and with allele sequence and zygosity matches. The results are shown in *Table S3 of Supplementary Tables*. It is clearly seen from *Table S3 of Supplementary Tables* that (1) NanoCaller has the highest precision by a significant margin and Clair has the best recall performance; (2) all variant callers have better performance for >1bp deletions than that for 1bp deletions; and (3) the indel predictions outside bi-nucleotide repeats by NanoCaller and medaka are usually better than indels in bi-nucleotide repeats. This shows what kinds of predicted indels are more reliable and how to improve variants callers in future.

## Novel Variants called by NanoCaller

We further analyze SNP calls made by NanoCaller on HG002 (ONT reads) that are absent in the GIAB ground truth calls (version 3.3.2) [28], and validate 17 regions of those SNP calls by Sanger sequencing. Sanger sequencing signals along with inferred sequences of both chromosomes for each region are shown in the supplementary folder of “Sanger Sequencing Files” and *Sanger_sequences.xlsx*. By deciphering Sanger sequencing results, we identify 41 novel variants (25 SNPs, 10 insertions and 6 deletions), as shown in Table 5 together with the variant calling results by different methods. As shown in Table 5, Medaka correctly identifies 8 SNPs, 3 insertions and 1 deletion, and Clair identifies 8 SNPs, 2

insertions and 1 deletion, whereas LongShot correctly identifies 8 SNPs; In contrast, NanoCaller is able to correctly identify 18 SNPs and 2 insertions, whereas 10 of these 18 SNPs and 1 of these 2 insertions are not called correctly by other variant callers on HG002 (ONT reads). In particular, there are 2 multiallelic SNPS in Table 5 which can be identified by NanoCaller but cannot be correctly called by all other 3 methods. One of the multiallelic SNPs at chr3: 5336450 (A>T,C) is shown in Figure 6 where both the IGV plots and Sanger results clearly show a multiallelic SNP that was correctly identified by NanoCaller but was missed by other variant callers. In summary, the prediction on these novel variants clearly demonstrate the usefulness of NanoCaller.

In particular, we use Figure 7 to illustrate those variants, which cannot be reliably detected by short-read, are detected by long-read variant callers. In Figure 7, the validated insertion is at chr9:135663795 or chr9:135663804 (there are two correct alignments at the deletion and thus both genomic coordinates are correct.). NanoCaller detects the deletion at chr9:135663805, while Medaka and Clair detect the deletion at chr9:135663799. Although they are several bps away from the expected genomic coordinates (which is normal in long-read prediction), the prediction provides accurate information of the deletion compared with short-read data where little evidence supports the deletion as shown in Figure 7 (a). Sanger sequencing signal data, shown in Figure 7 (b), confirms the presence of a heterozygous indel at the same location which is causing frameshift between the signals from maternal and paternal chromosomes. This is an excellent example that long-read variant callers can detect variants which failed to be reliably called on short-reads.

# 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 has 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 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 evaluations under the 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.

NanoCaller has several advantages to call variants from long-read sequencing data. (1) NanoCaller uses pileup of candidate variants from haplotyped set of neighboring heterozygous 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 heterozygous 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. (2) NanoCaller is 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 calling variants on PacBio long-read data and achieved similar prediction performance. We also test NanoCaller models trained on GRCh38 for GRCh37 and achieve the same level SNP calling performance. (3) With the advantage of long-read data on repetitive regions, NanoCaller is able to detect SNPs/indels outside high-confidence regions which cannot be reliably detected by short-read sequencing techniques, and thus NanoCaller provides more candidate SNPs/indels sites for investigating causal variants on undiagnosed diseases where no disease-causal candidate variants were found by short-read sequencing. (4) NanoCaller has 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, 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 all 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. (5) NanoCaller uses rescaled statistics to generate pileup for a candidate site, and rescaled 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 competitive variant calling performance, because NanoCaller integrates haplotyped long-range information to improve variant calling.

There are several limitations of NanoCaller that we wish to discuss here. One is that NanoCaller relies on the accurate alignment and pileup of long-read sequencing data, and 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 [37, 38]. 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 research* 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**. *Nature 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**. *Nature 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 C-M, Lam T-W: **Exploring the limit of using a deep neural network on pileup data for germline variant calling**. *Nature Machine Intelligence* 2020, **2**(4):220-227.
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 (Oxford, England)* 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. 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.
36. 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.
37. 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.
38. 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. Statistics of benchmark variants in chromosomes 1-22 of each genome aligned to the GRCh38 reference genome. For genomes with GIAB v.3.2.2 benchmark variant calls (HG001, HG002, HG003 and HG004), 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 | | High confidence region | | | | Non homo -polymer region |
|---|---|---|---|---|---|---|---|
| **Genome** | **SNPs** | **Indels** | **SNPs** | **Indels** | **Total Length** | **% of genome** | **Indels** |
| HG001 | 3,004,071 | 516,524 | 2,961,527 | 483,630 | 2,329,784,734 | 81.03 | 239,488 |
| HG002 | 3,079,462 (3,100,749) | 518,055 (509,449) | 3,030,495 (3,048,869) | 475,332 (464,463) | 2,353,170,731 (2,358,060,765) | 81.85 (81.85) | 236,337 |
| HG003 | 2,927,639 | 510,622 | 2,883,686 | 471,156 | 2,261,821,224 | 78.67 | 231,265 |
| HG004 | 2,953,590 | 521,646 | 2,909,326 | 481,114 | 2,266,589,952 | 78.84 | 234,374 |
| HX1 | 3,282,242 | 687,501 | 2,980,737 | 205,591 | 2,356,619,870 | 76.31 | -- |

**Table 2. Performances of SNP predictions by NanoCaller 1, NanoCaller 2 SNP models, and existing variant callers on ONT data (the format in each cell is Precision/Recall/F1 percentages). Performances of Nanocaller 1 on HG002 aligned to GRCh37 and GRCh38 are shown with and without paranthesis respectively.**

| | HG001 | HG002 | HG003 | HG004 | HX1 |
|---|---|---|---|---|---|
| NanoCaller 1 | 98.62/90.52/94.4 | 98.93/95.65/97.26 (98.61/95.56/97.06) | 98.41/95.52/96.95 | 98.43/95.3/96.84 | 70.09/83.56/76.23 |
| NanoCaller 2 | 97.95/91.92/94.84 | 99.16/90.49/94.63 | 98.42/95.29/96.83 | 98.38/95.12/96.72 | 65.7/85.05/74.13 |
| Medaka | 92.94/93.99/93.46 | 98.67/96.02/97.32 | 98.75/92.49/95.51 | 98.77/95.75/97.23 | 78.73/90.77/84.32 |
| Longshot | 98.99/92.9/95.84 | 99.29/96.28/97.76 | 98.92/95.34/97.09 | 99.03/91.18/94.94 | 93.47/92.85/93.15 |
| Clairvoyante | 94.42/87.72/90.94 | 95.68/92.6/94.11 | 91.93/93.88/92.89 | 91.86/93.73/92.78 | 68.78/89.93/77.94 |
| Clair | 97.98/94.99/96.47 | 99.09/97.01/98.04 | 98.91/97.27/98.08 | 98.92/97.12/98.01 | 90.98/93.47/92.21 |

**Table 3. Performances of SNP predictions by NanoCaller 1, NanoCaller 2 SNP models, and existing variant callers on PacBio data (the format in each cell is Precision/Recall/F1 percentages).**

| | HG002 | HG003 | HG004 |
|---|---|---|---|
| NanoCaller 1 | 98.8/96.74/97.76 | 95.91/91.18/93.49 | 95.07/92.01/93.51 |
| NanoCaller 3 | 98.22/97.33/97.77 | 96.27/84.92/90.24 | 88.61/92.46/90.49 |
| Longshot | 98.83/98.61/98.72 | 99.18/94.73/96.9 | 99.2/93.73/96.39 |
| Clairvoyante | 97/96.5/96.75 | 72.15/84.93/78.02 | 68.46/82.95/75.01 |

**Table 4. Performance of indel prediction by NanoCaller and variant callers on ONT data (the format in each cell is Precision/Recall/F1 percentages) with recommended quality cut-off for Clair and Clairvoyante, and averaged quality cut-off for NanoCaller and Medaka.**

| | HG001 | HG002 | HG003 | HG004 |
|---|---|---|---|---|
| NanoCaller | 64.55/48.09/55.12 | 77.85/57.98/66.46 | 72.08/59.13/64.97 | 70.74/57.81/63.62 |
| Medaka | 47.46/56.10/51.42 | 79.52/62.93/70.26 | 80.04/62.24/70.03 | 79.91/63.25/70.61 |
| Clairvoyante | 49.89/24.72/33.06 | 54.95/27.31/36.48 | 43.74/29.67/35.35 | 42.87/28.52/34.25 |
| Clair | 89.29/55.81/68.69 | 86.34/54.56/66.87 | 85.83/57.05/68.54 | 86.39/56.20/68.10 |

**Table 5. Novel variants in HG002 genome discovered by Sanger sequencing together with the prediction information by NanoCaller and other variant callers.**

| Chrom | Position | REF | ALT | Genotype | Nanocaller | Medaka | Clair | Longshot |
|---|---|---|---|---|---|---|---|---|
| chr1 | 78883942 | C | T | 1\|0 | Correct | -- | -- | -- |
| chr1 | 78883952 | ATATATATTTATCCTTTATATATATTCTT | A | 1\|0 | -- | -- | -- | -- |
| chr2 | 227913871 | A | ATATCTATCTATC | 1\|0 | Correct | Correct | Correct | -- |
| chr2 | 227913885 | G | A | 1\|0 | Correct | Correct | Correct | Correct |
| chr2 | 227913889 | G | A | 1\|0 | Correct | Correct | Correct | Correct |
| chr2 | 227913928 | T | TA | 1\|0 | wrong allele | Correct | -- | -- |
| chr2 | 227913931 | A | T | 1\|0 | Correct | -- | -- | -- |
| chr3 | 5336450 | A | T,C | 1\|2 | Correct | wrong allele | wrong allele | wrong allele |
| chr3 | 5336452 | C | CGCGT | 0\|1 | -- | -- | -- | -- |
| chr3 | 5336465 | ACACACACACACG | A | 0\|1 | wrong variant type | wrong allele | -- | -- |
| chr3 | 5336477 | GCA | G | 1\|0 | wrong allele | Correct | wrong allele | -- |
| chr3 | 5336487 | A | G | 0\|1 | wrong variant type | wrong variant type | -- | -- |
| chr6 | 160009985 | C | CTTAA | 0\|1 | wrong allele | -- | wrong allele | -- |
| chr6 | 160009986 | C | A | 0\|1 | wrong zygosity | wrong allele | wrong variant type | wrong allele and zygosity |
| chr6 | 167130970 | G | GGGCCCCCCTCCCTCCGGGACTCCTCCCTCT | 0\|1 | -- | -- | -- | -- |
| chr6 | 167130972 | GA | G | 1\|0 | -- | wrong zygosity | wrong zygosity | -- |
| chr6 | 167130973 | A | G | 0\|1 | -- | -- | -- | -- |
| chr6 | 167130976 | A | C | 1\|1 | Correct | wrong zygosity | Correct | Correct |
| chr6 | 167130986 | T | C | 1\|0 | Correct | wrong allele | -- | wrong allele |
| chr6 | 167130989 | A | G | 1\|1 | Correct | Correct | Correct | Correct |
| chr6 | 167130990 | C | A | 1\|0 | -- | -- | -- | -- |
| chr6 | 167130992 | C | T | 1\|0 | Correct | -- | Correct | Correct |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| chr9 | 134784949 | C | T | 0\|1 | Correct | Correct | Correct | Correct |
| chr9 | 134784951 | G | T | 0\|1 | -- | -- | -- | -- |
| chr9 | 134784955 | G | GGGGGGCA | 0\|1 | -- | -- | -- | -- |
| chr9 | 134784956 | T | G | 0\|1 | Correct | wrong variant type | wrong variant type | -- |
| chr9 | 135663795 | ACAGAGGGGGACCTGGAGGGGCAGAGGAGAGACCTGTGGGG | A | 0\|1 | -- | -- | -- | -- |
| chr9 | 135663892 | A | G | 1\|1 | wrong zygosity | Correct | Correct | Correct |
| chr9 | 135663893 | T | A,G | 1\|2 | Correct | wrong allele | wrong allele | wrong allele |
| chr11 | 113466435 | G | GC | 1\|1 | -- | Correct | Correct | -- |
| chr11 | 113466437 | A | T | 1\|1 | Correct | wrong allele | wrong allele | wrong allele |
| chr12 | 100940063 | A | AT | 0\|1 | Correct | -- | -- | -- |
| chr12 | 100940065 | G | C | 0\|1 | Correct | wrong allele | -- | -- |
| chr14 | 75318035 | C | T | 1\|0 | Correct | Correct | Correct | Correct |
| chr14 | 75318038 | AT | A | 1\|0 | -- | wrong allele | Correct | -- |
| chr14 | 75318052 | T | C | 1\|0 | Correct | -- | -- | -- |
| chr14 | 75318054 | T | G | 1\|0 | -- | -- | -- | -- |
| chr20 | 11064571 | T | TGA | 0\|1 | -- | wrong allele | -- | -- |
| chr20 | 11064574 | A | ATTTTCAAGACTATTGTGACTATGAC | 0\|1 | -- | Correct | -- | -- |
| chr20 | 11064578 | A | T | 0\|1 | Correct | -- | -- | -- |
| chr20 | 11064579 | C | T | 0\|1 | Correct | Correct | wrong variant type | -- |

# Figures

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

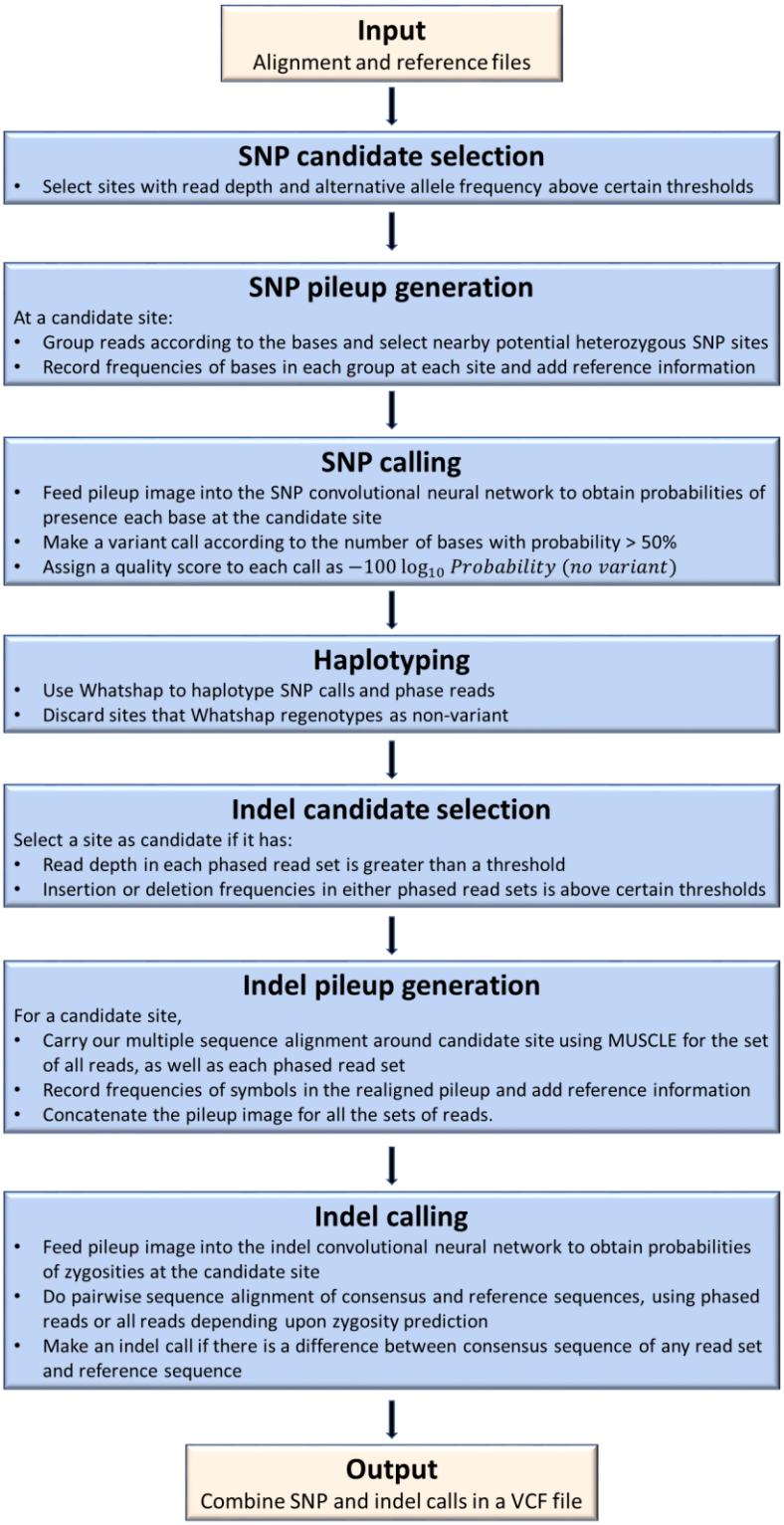


**Figure 2. An example on how to construct image pileup for a SNP candidate site. a) reference sequence and reads pileup at site b and sites in set Z; b) raw counts of bases at sites in Z in each read group split by the nucleotide types at site b; c) frequencies of bases at sites in Z with negative signs for reference bases; d) flattened pileup image with fifth channel after reference sequence row is added; e) pileup image used as input for NanoCaller.**

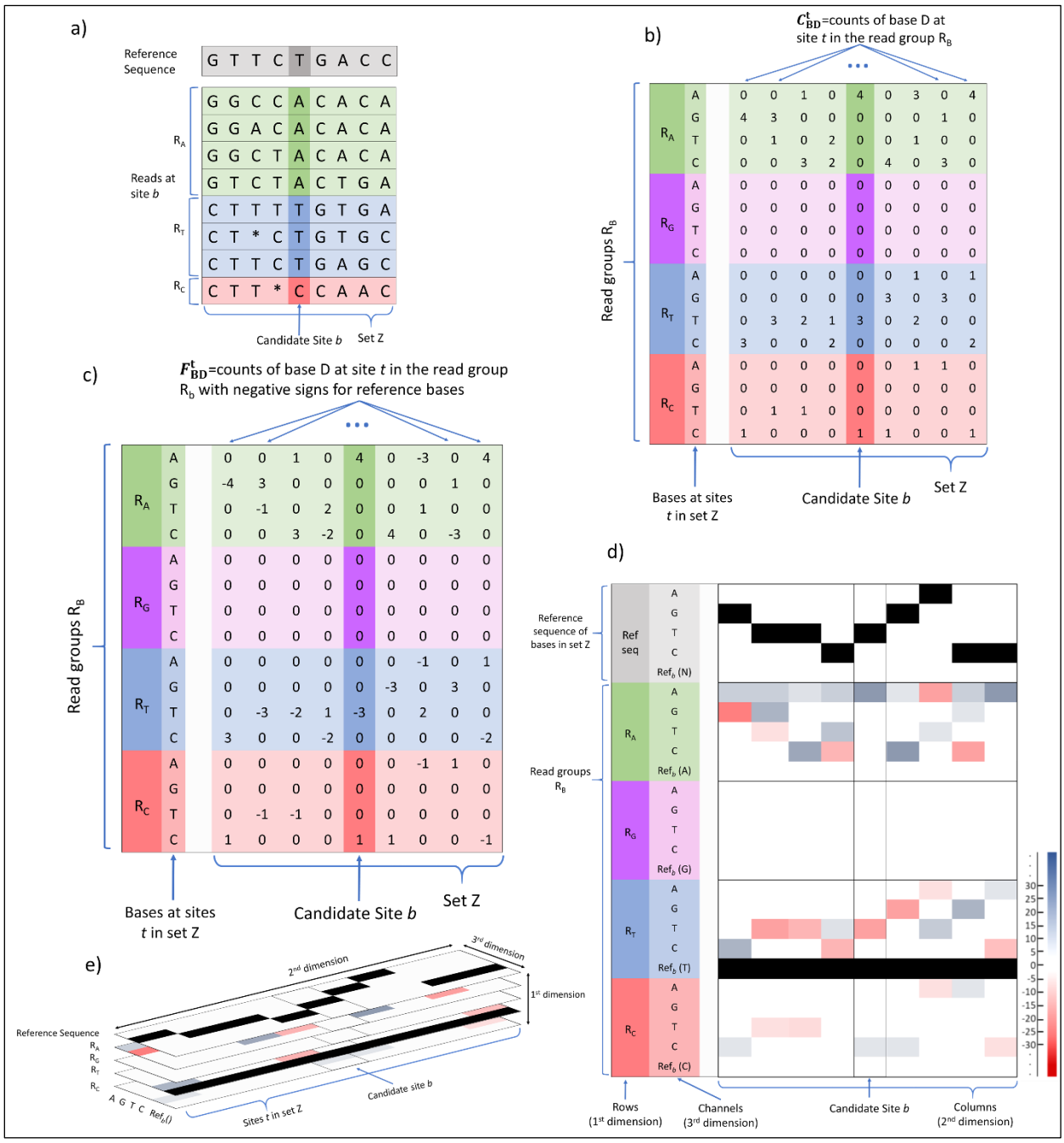
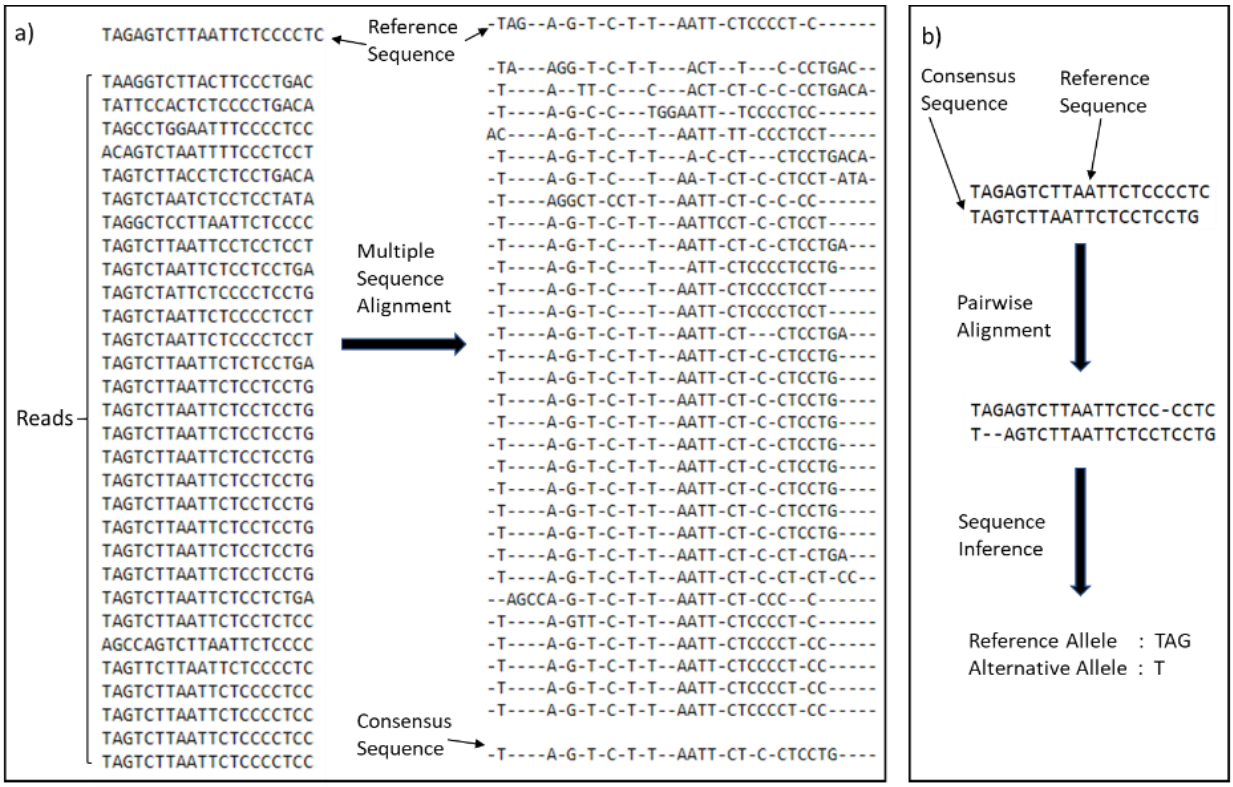


**Figure 3. An example on how to construct image pileup for an indel site.** a) reference sequence and reads pileup at the candidate site before and after multiple sequence alignment, and the consensus sequence after realignment; b) reference sequence and consensus sequence at the candidate site before and after pairwise alignment, and the inferred sequence; c) shows raw count of each symbol at each column of multiple sequence alignment pileup; d) matrix M, showing frequency of each symbol at each column of multiple sequence alignment pileup; e) first channel of input image, matrix M minus Q (one-hot encoding of realigned reference sequence); f) matrix Q, one-hot encoding of realigned reference sequence which forms the second channel of input image.




c)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | 0 | 2 | 0 | 0 | 0 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 26 | 30 | 0 | 0 |
| G | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| T | 0 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 29 | 0 | 0 | 0 | 22 | 0 | 29 | 0 | 0 | 0 | 0 | 26 | 29 |
| C | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 30 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 1 |
| - | 29 | 1 | 28 | 29 | 29 | 29 | 0 | 28 | 1 | 27 | 0 | 30 | 0 | 29 | 8 | 30 | 0 | 29 | 29 | 4 | 0 | 2 | 0 |
| ref | - | T | A | G | - | - | A | - | G | - | T | - | C | - | T | - | T | - | - | A | A | T | T |

d)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.033 | 0 | 0.067 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.867 | 1 | 0 | 0 |
| G | 0 | 0 | 0 | 0.033 | 0 | 0 | 0 | 0.067 | 0.967 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.033 | 0.033 | 0 | 0 | 0 | 0 |
| T | 0 | 0.933 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.067 | 0.967 | 0 | 0 | 0 | 0.733 | 0 | 0.967 | 0 | 0 | 0 | 0 | 0.867 | 0.967 |
| C | 0 | 0.033 | 0 | 0 | 0.033 | 0.033 | 0 | 0 | 0 | 0.033 | 0.033 | 0 | 1 | 0.033 | 0 | 0 | 0.033 | 0 | 0 | 0 | 0 | 0.067 | 0.033 |
| - | 0.967 | 0.033 | 0.933 | 0.967 | 0.967 | 0.967 | 0 | 0.933 | 0.033 | 0.9 | 0 | 1 | 0 | 0.967 | 0.267 | 1 | 0 | 0.967 | 0.967 | 0.133 | 0 | 0.067 | 0 |
| ref | - | T | A | G | - | - | A | - | G | - | T | - | C | - | T | - | T | - | - | A | A | T | T |

e)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.033 | 0 | -0.933 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.133 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | -0.967 | 0 | 0 | 0 | 0.067 | -0.033 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.033 | 0.033 | 0 | 0 | 0 | 0 |
| T | 0 | -0.067 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.067 | -0.033 | 0 | 0 | 0 | -0.267 | 0 | -0.033 | 0 | 0 | 0 | 0 | -0.133 | -0.033 |
| C | 0 | 0.033 | 0 | 0 | 0.033 | 0.033 | 0 | 0 | 0 | 0.033 | 0.033 | 0 | 0 | 0.033 | 0 | 0 | 0.033 | 0 | 0 | 0 | 0 | 0.067 | 0.033 |
| - | -0.033 | 0.033 | 0.933 | 0.967 | -0.033 | -0.033 | 0 | -0.067 | 0.033 | -0.1 | 0 | 0 | 0 | -0.033 | 0.267 | 0 | 0 | -0.033 | -0.033 | 0.133 | 0 | 0.067 | 0 |
| ref | - | T | A | G | - | - | A | - | G | - | T | - | C | - | T | - | T | - | - | A | A | T | T |

f)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| G | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| ref | - | T | A | G | - | - | A | - | G | - | T | - | C | - | T | - | T | - | - | A | A | T | T |

**Figure 4. The deep-learning framework for SNP and indel calling.** **a) An illustration of the convolutional neural network model for SNP calling; b) An illustration of the convolutional neural network model for indel calling. In both models, 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 fed to a fully connected layer followed by a hidden layer with 48 nodes with a 50% dropped rate. In a), output of the hidden layer is split into two independent pathways: one for calculating probabilities of each base and the other for calculating zygosity probabilities. Zygosity probability is only used in the training process. In b), output of first fully connected layer is fed into two fully connected hidden layers to produce probabilities of four possible cases of zygosities.**

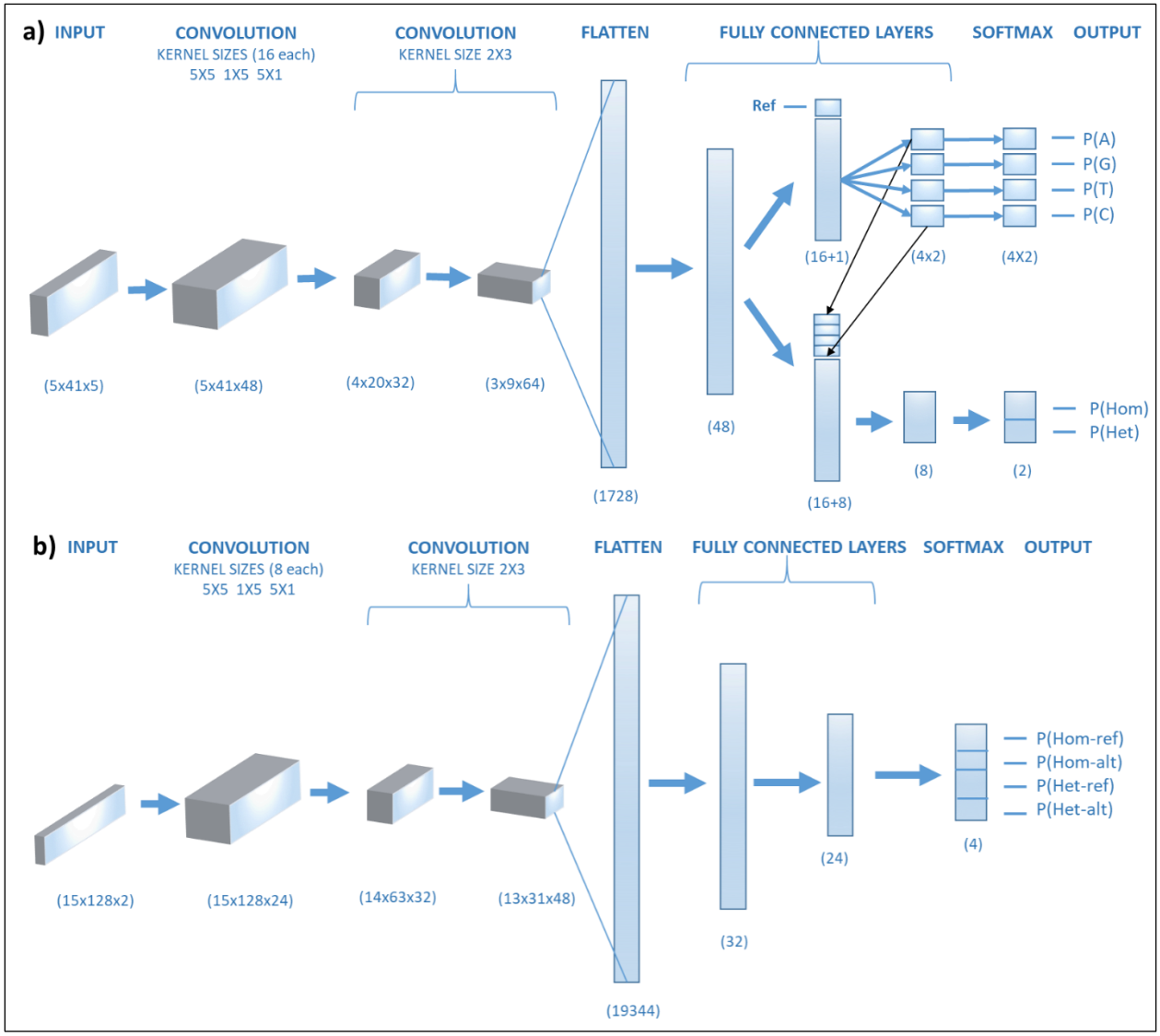


**Figure 5. Performance of NanoCaller and state-of-the-art variant callers on five whole-genome sequencing data sets.** **The performance of SNP predictions on ONT reads: a) precision, b) recall, c) F1 score. d) The performance of SNP predictions on HG002 (ONT) mapped to GRCh38 vs GRCh37 while NanoCaller is trained on HG001 (ONT) with GRCh38. The performance of SNP predictions on PacBio reads: e) precision, f) recall, g) F1 score. The performance of indel predictions on ONT reads: h) precision, i) recall, j) F1 score. NanoCaller 1, NanoCaller 2 and NanoCaller 3 mean NanoCaller models trained on HG001 (ONT), HG002 (ONT), and HG003 (PacBio) respectively. If the same genome is used in training and testing, the chromosome 1 is excluded in training and used for testing. For HX1, the variants, which were called on high-coverage short-read data and filtered out low complexity regions are used as benchmark.**

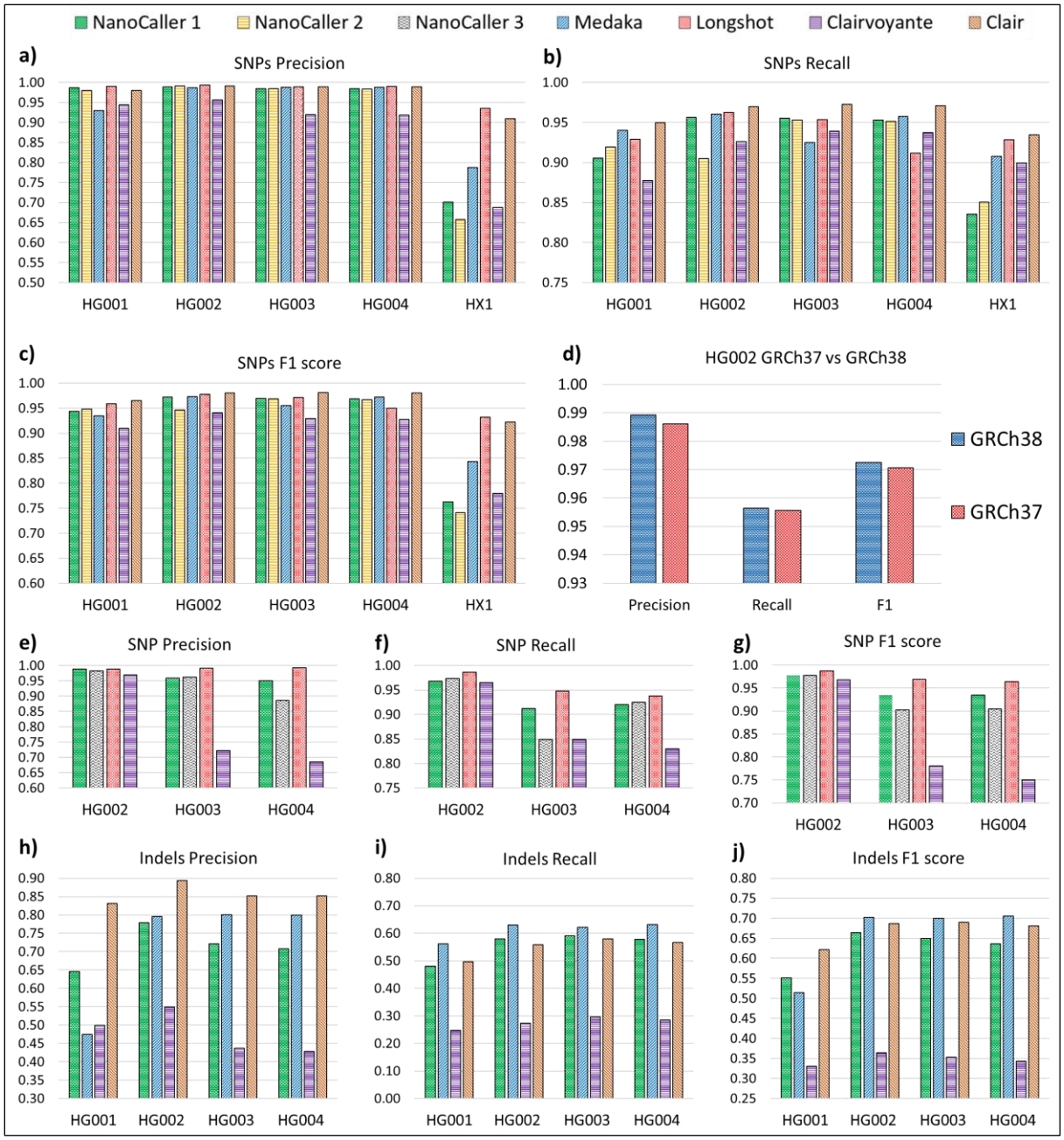


**Figure 6. a) IGV plots of Nanopore, PacBio CCS and Illumina reads of HG002 genome at chr3: 5336450- 5336480. b) Sanger sequencing signal data for the same region. NanoCaller correctly identified the multi-allelic SNP at chr3: 5336450 (A>T,C) shown in black box.**

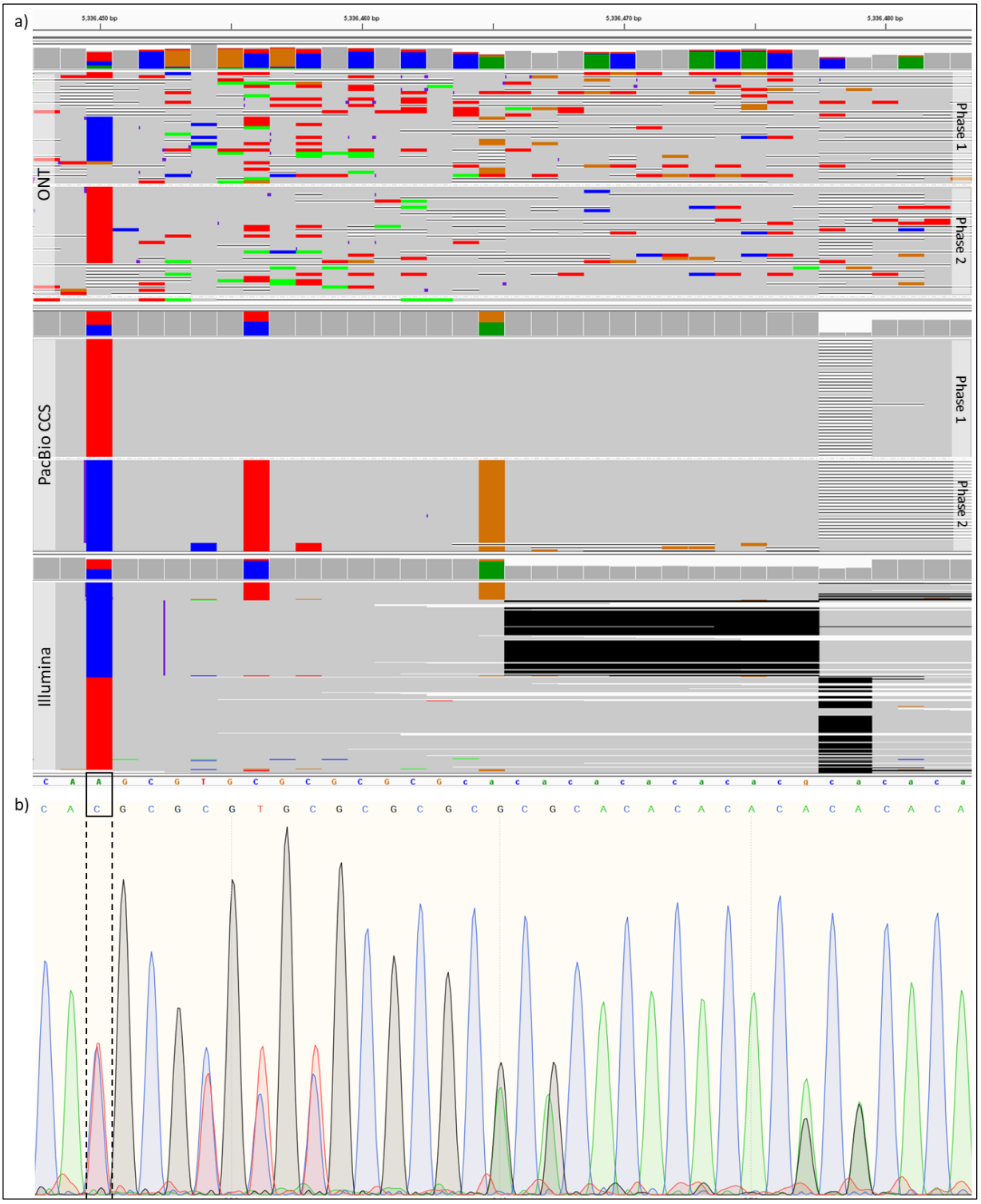
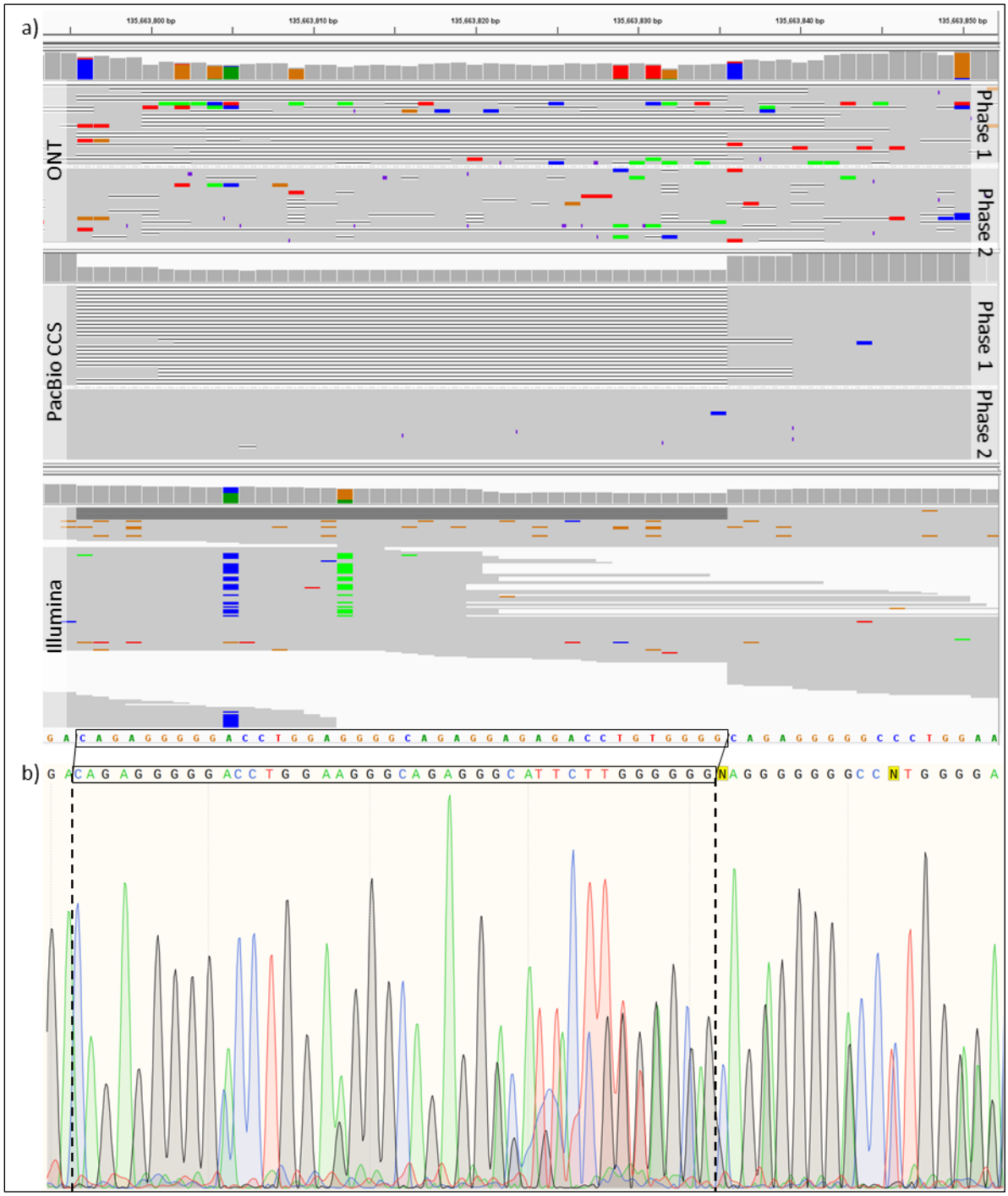



**Figure 7. a) IGV plots of Nanopore, PacBio CCS and Illumina reads of HG002 genome at chr9:135663780-chr9:135663850. The 40bp long deletion shown below in black box was identified using Sanger sequencing at chr9:135663795 or chr9:135663804 (both are correct and the difference is due to two different alignments). b) shows Sanger sequencing signal data around the deletion. In b), multiple bases show strong signals at various locations, indicating presence of a heterozygous indel, which agrees with IGV plots of long-read data.**