NanoCaller for accurate detection of SNPs and indels in difficult-to-map regions from long-read sequencing by haplotype-aware 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

Variants (SNPs/indels) detection from high-throughput sequencing data remains an important yet unresolved problem. Long-read sequencing enables variants detection in difficult-to-map 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, the higher per-base error rate poses unique challenges in variants detection. Existing methods on long-read data typically rely on analyzing pileup information from neighboring bases surrounding a candidate variant, similar to short-read variant callers, yet the benefits of much longer read length are not fully exploited. Here we present a deep neural network called NanoCaller, which detects SNPs by examining pileup information from other candidate SNPs that share the same long reads using long-range haplotype information. With called SNPs by NanoCaller, NanoCaller phases long reads and performs local realignment on two sets of phased reads to call indels by another deep neural network. Extensive evaluation on 5 human genomes (sequenced by Nanopore and PacBio long-read techniques) demonstrates that NanoCaller has greatly improved performance in difficult-to-map regions, compared to other long-read variant callers. We experimentally validated 41 novel variants in difficult-to-map regions in a widely used benchmarking genome, which cannot be reliably detected previously. Finally, we achieved the best performance in Nanopore-based variant calling from MHC regions in the PrecisionFDA Variant Calling Challenge on Difficult-to-Map Regions. In summary, by incorporating haplotype information in deep neural networks, NanoCaller facilitates the discovery of novel variants in complex genomic regions from long-read sequencing data.

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 reliable 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 raw 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 adjacently aligned local genomic positions of 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 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 especially for difficult-to-map regions. 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 long-range 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; Then, pileup of a candidate site is built based on its tens of neighboring heterozygous candidate sites with shared long reads (but hundreds or thousands bp away, which are substantially different from DeepVariant, Clairvoyante [22] and its successor Clair [23]), and fed into a deep convolutional neural network for SNP calling. Predicted heterozygous SNPs are then used to phase alignment reads for indel calling: with long reads aligned against a candidate indel site, two phased sets of reads and entire set of reads are separately used in multiple sequencing alignment, and these three locally realigned groups of reads are used to predict indel variant zygosity with a deep learning framework and to infer indel sequences from the consensus sequences of long reads from each group. We assess NanoCaller on several human genomes, such as HG001 (NA12878), HG002 (NA24385), HG003 (NA24149), HG004 (NA24143) and HX1 with both Nanopore and PacBio long-read data. In particular, we evaluate NanoCaller in difficult-to-map regions to investigate the unique advantages provided by long reads. Our evaluation demonstrates competitive performance of NanoCaller against existing tools, with particularly improved performance in complex genomic regions which cannot be reliably called on short-read data. NanoCaller is publicly available at https://github.com/WGLab/NanoCaller.

Results

Overview of NanoCaller

NanoCaller takes the alignment of a long-read sequencing data set against the reference genome as input, and generates a VCF file for called SNPs and indels. There are several steps in NanoCaller. For SNP calling, candidate SNP sites are initially generated according to the minimum coverage and minimum frequency of alternative alleles (a fraction of them are likely to be false positives given the relaxed thresholds for candidate identification). Then, a set of likely heterozygous candidate SNPs are used to build an input image for each candidate SNP site (Figure 1), followed by a deep convolutional network (Figure 3) to distinguish true variants from false candidate sites. The predicted SNPs are phased, and together with the long-read alignment, are then used in the identification of indels. Unlike other similar software tools, in

NanoCaller, the pileup images for indels are generated using multiple sequencing alignment on the set of entire reads and on each set of phased reads (Figure 2). For indel calling, indel candidate sites are selected according to a minimum coverage and insertion/deletion frequency criteria. After that, 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.

The performance of NanoCaller is evaluated on both Oxford Nanopore reads and PacBio reads, and compared with performances of Medaka, Clair and Longshot with a particular emphasis in difficult-to-map regions. The evaluation is on the GRCh38 reference genome by default 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 each variant caller, we show its performance with the quality threshold determined by *vcfeval* to give the best F1 score. We discuss quality threshold for various NanoCaller models in Supplementary Tables S14 and S15. For SNP calling in NanoCaller, we present performances of three SNP models: NanoCaller1 (trained on HG001 ONT reads), NanoCaller2 (trained on HG002 ONT reads) NanoCaller3 (trained on HG003 PacBio CLR dataset). For indel calling, we show performances of two indel models: NanoCaller1 (trained on HG001 ONT reads) and NanoCaller3 (trained on HG003 PacBio CCS reads).

Evaluation of NanoCaller on Oxford Nanopore Sequencing

Performance on SNP calling

We evaluated NanoCaller on Oxford Nanopore sequencing reads of five genomes: HG001, HG002, HG003, HG004 and HX1 under several testing strategies: cross-genome testing and cross-reference.

Cross-genome testing is critical to demonstrate the performance of a variant caller when used in a real-world scenario: the deep learning model of a variant caller is trained in a genome and tested on other genomes. The performance of NanoCaller, together with three other variant callers, Medaka, Clair and Longshot, are evaluated under this testing strategy, and the results of SNP calling are shown in Table 2 and Figure 4 (a), (b) and (c) on Nanopore reads of five genomes. As shown in Table 2 and Figure 4 (a), (b) and (c), NanoCaller has better precision, recall and precision than Medaka on HG001 and competitive against Clair and LongShot. On

the Ashkenazim trio (HG002, HG003, HG004), NanoCaller has better performance (in term of F1 score) than Clair and LongShot (NanoCaller1: 98.05, 98.87, 98.76%; NanoCaller2: 98.07, 98.87, 98.8% VS Clair: 97.79%, 98.55, 98.6% and Longshot: 98.01%, 97.94%, 97.9%, on HG002, HG003 and HG004 respectively), and competitive against Medaka. Meanwhile, we also evaluate the SNP callings for the four methods on older benchmark and basecalled reads with older Nanopore basecallers for the Ashkenazim trio HG002-4, as shown in Supplementary Tables S5 and S6. We find that all methods have improved performance; this suggests that improved Nanopore basecalling and increasing number of benchmark variants enhance the evaluation of variant calling (as shown in Table 1, the Ashkenazim trio HG002-4 has more variant call set than HG001: 370-400k more SNPs per genome than HG001, and a larger high-confidence region with more complex regions: at least 200mbp larger than HG001 and covering an extra 7% of genome).

We also show that the performance of NanoCaller SNP models is independent upon the reference genome used. Under this cross-reference testing, we evaluate NanoCaller1 model (trained on alignment against GRCh38) for HG002 ONT reads aligned to both GRCh38 and GRCh37 reference genomes, and calculate the performance using v3.3.2 benchmark variants for HG002, the latest version for which we have benchmark variant calls for GRCh37. For reads aligned to GRCh38, we obtain a precision of 97.55%, a recall of 97.36% and an F1 score of 97.46%, whereas for reads aligned to GRCh37, we obtain a precision of 98.40%, a recall of 97.66% and an F1 score of 98.03%. The competitive performance on GRCh38 and GRCh37 indicates that NanoCaller could be used on alignment generated by mapping to different reference genomes.

Performance on SNP calling in difficulty-to-map regions

We further demonstrate that NanoCaller has unique advantage to call SNPs especially in difficult-to-map (DtM) genomic regions. To have an unbiased testing, we use NanoCaller1 SNP model trained on HG001 and test it on the three genomes of the trio together with other variant callers for DtM regions. DtM regions here are defined by GA4GH Benchmarking Team and the Genome in a Bottle Consortium, and downloaded as a BED file from GIAB v2.0 genome stratification. These DtM regions contains all tandem repeats, all homopolymers >6bp, all imperfect homopolymers >10bp, all low mappability regions, all segmental duplications, GC <25% or >65%, Bad Promoters, and other difficult regions such Major Histocompatibility Complex. We intersect the BED file with high-confidence interval for each genome in the trio

and evaluate SNP performance in the intersected DtM regions. As shown Table 2, each genome has at least 600k SNPs in the intersection of DtM regions and high-confidence intervals, which is a significant fraction (18-19%) of all SNPs in the high-confidence regions. The evaluation on these SNPs are shown in Figure 4 (d) and Table 2: NanoCaller (F1 score 95.59%, 96.67% and 96.66% for HG002, HG003 and HG004 respectively) performs better than all other variant callers for each genome. F1 scores of NanoCaller exceed Medaka's, Clair's and Longshot's F1 scores by 0.41%, 0.48% and 1.88% on average. In Supplementary Tables S9-12, we further show a detailed breakdown of performances in the DtM regions and demonstrate that NanoCaller performs better than other variant callers for SNPs in each of the following difficult regions: segmental duplications (Supplementary Tables S10), tandem and homopolymer repeats (Supplementary Tables S11), low mappability regions (Supplementary Tables S9), and Major Histocompatibility Complex (Supplementary Tables S12). The only exception is that only Longshot performs slightly better than NanoCaller in tandem and homopolymer repeat regions for HG002 and HG003.

NanoCaller team participated in PrecisionFDA truth challenge v2 for difficult-to-map regions (held in July 2020, see https://precision.fda.gov/challenges/10), and submitted variant calls for the Ashkenazim trio made by an ensemble of NanoCaller, Medaka and Clair. Our submission won the award for best performance in Major Histocompatibility Complex (MHC) using Nanopore reads. Figure 4 (e) shows the F1 score of SNP performance of the ensemble, NanoCaller, Medaka and Clair. While the ensemble performs better than all other variant callers on each genome, NanoCaller's performance is very close to the ensemble and better than the performances of Medaka and Clair. Therefore, this is an independent assessment of the real-world performance of NanoCaller in detecting variants in complex genomic regions.

Performance on Indel Calling

We train indel model NanoCaller1 on chromosomes 1-22 of HG001 (ONT), and test it on four genomes (HG001, HG002, HG003, HG004) using ONT reads together with Medaka and Clair. For each genome, we create evaluation regions by removing homopolymer >5bp regions from GIAB high confidence regions, and evaluate the indel performance by RTG *vcfeval*. The results are shown in Table 2 and Figure 4 (f), (g) and (h). According to the F1 score in Table 2, NanoCaller generally performs better than Clair (~6% on HG002, ~4% on HG003 and ~3% on HG004) and competitively against Medaka for HG001 to HG004. It is also worth noting that Medaka has better precision, but NanoCaller has much higher recall than Medaka and Clair; for

example, NanoCaller has ~14% and ~6% higher recall than Clair and Medaka respectively on each of the three genomes of the trio. This suggests that NanoCaller complements other existing methods well to identify more indels. Supplementary Figure S1 shows concordance of ground truth variants of the Ashkenazim trio correctly predicted by NanoCaller, Medaka and Clair. Figure S1 (b) shows each tool has a significant number (ranging from 19k-60k) of correctly predicted indel calls, that are not correctly predicted by other variant callers.

Performance on PacBio Sequencing data

Performance on Circular Consensus Sequencing (CCS) reads.

On CCS datasets of four genomes: HG001, HG002, HG003 and HG004, we evaluate NanoCaller SNP models NanoCaller1 and NanoCaller3, and the results are shown in Table 2 and Figure 5 (a), (b) and (c) along with Clair and Longshot. It can be seen from Table 2 and Figure 5 (a), (b) and (c) that (i) both NanoCaller1 and NanoCaller3 have similar performance according to F1 scores, although NanoCaller1 model is trained on Nanopore dataset while NanoCaller3 model is trained on PacBio CLR dataset; thus, NanoCaller is able to perform well cross different sequencing platform; (ii) NanoCaller outperforms LongShot, and (iii) NanoCaller shows competitive performance against Clair (99.78, 99.77, 99.76% on the trio) with difference <=0.1%.

We also evaluate NanoCaller3 indel model on CCS datasets of the Ashkenazim trio: HG002, HG003 and HG004, and the results are shown in Table 2 and Figure 5 (d), (e) and (f) along with the Clair indel performance. The F1-scores on the trio suggest that NanoCaller performs competitively against Clair.

With Continuous Long Read Sequencing (CLR) datasets, we evaluate NanoCaller SNP models NanoCaller1 and NanoCaller3 for the Ashkenazim trio: HG002, HG003 and HG004 (as shown in Table 2 and Figure 5 (g), (h) and (i)). NanoCaller performs worse than Clair and Longshot with CLR reads. This is because PacBio reads are usually shorter than Nanopore reads, and there are less number of heterozygous SNP candidates that share certain long reads with a candidate SNP of interest. This indicates that longer reads would benefit the performance of NanoCaller: longer reads allow more heterozygous SNP candidates to be selected to generate pileup image for deep learning.

Novel Variants called by NanoCaller

We also 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 before v4 benchmark for HG002 was made available (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 3. Based on the 41 novel variants, we conduct the variant calling results by different methods on both older ONT HG002 reads and newly released ONT HG002 reads (As described in the Methods section) to see how more accurate long reads improve variant calling. We find that (1) on the newly released ONT HG002 reads, Medaka correctly identifies 15 SNPs, 6 insertions and 2 deletions, Clair identifies 14 SNPs, 6 insertions and 2 deletions, and LongShot correctly identifies 18 SNPs, while NanoCaller is able to correctly identify 20 SNPs and 6 insertions, as shown in the Supplementary Table S13; and (2) on the older ONT HG002 reads, as shown in Table 3, 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 the older HG002 (ONT reads). This indicates that the improvements in per-base accuracy during base-calling significantly enhances the variant calling performance. Also in Table 3, there are 2 multiallelic SNPs 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, likely due to the unique haplotype-aware feature of NanoCaller. In summary, the prediction on these novel variants clearly demonstrate the usefulness of NanoCaller for SNP calling.

To demonstrate the performance of NanoCaller for indel calling, we use Figure 7 to illustrate those variants that can be detected by long-read variant callers but cannot be detected by short-read data. 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 based variant calling), 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 example how long-read variant callers on long-read data can detect variants that fail to be reliably called on short-read sequencing data.

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 poses a significant challenge to variant calling and thus stimulates 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 generate indel calling. Our evaluations under the cross-genome testing, cross-reference genome testing, and cross-platform testing demonstrate that NanoCaller performs competitively against other long-read variant callers, and outperforms other methods in difficult-to-map genomic regions.

NanoCaller has several advantages to call variants from long-read sequencing data. (1) NanoCaller uses pileup of candidate SNPs from haplotyped set of neighboring heterozygous SNPs (with hundreds or thousands bp away rather than adjacent neighborhood local region of a candidate SNP of interest), 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. (6) With very accurate HiFi reads (<1% error rate) generated by PacBio, NanoCaller is able to yield competitive variant calling performance.

However, 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 for SNP calling and local multiple sequence alignments for accurate 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. 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.

Methods

Datasets

Long-read data

Five long-read data sets for human genomes are used for the evaluation of NanoCaller: HG001, the Ashkenazim trio (consisting of son HG002, father HG003 and mother HG004), and HX1. For HG001, Oxford Nanopore Technology (ONT) rel6 FASTQ files, basecalled with Guppy 2.3.8, are downloaded from WGS consortium database[9], and aligned to the GRCh38 reference genome using minimap2 [27]; PacBio CCS alignment files for HG001 are downloaded from the GIAB database [28, 29]. For HG002, HG003 and HG004 in the Ashkenazim trio, newly released sets of ONT reads (basecalled by Guppy 3.6) and PacBio CCS reads are obtained from NIST through precisionFDA challenge website, and aligned to the GRCh38 reference genome using minimap2 [27]. Meanwhile, alignment files for an older dataset of ONT reads for HG002, basecalled by Guppy 2.3.5, are downloaded from the GIAB database [28, 29] for analysis. PacBio CLR alignment reads for the Ashkenazim trio files are downloaded from the GIAB database [28, 29]. The fifth genome HX1 was sequenced by us using PacBio [10] and Nanopore sequencing [30]. The long-read data is aligned to the GRCh38 reference genome using minimap2 [27]. Supplementary Table S1 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 for HG001 (version 3.3.2), and the Ashkenazim trio (v4.2) are download from the Genome in a Bottle (GIAB) Consortium [28] together with high-confidence regions for each genome. There are 3,004,071; 3,459,843; 3,430,611; and 3,454,689 SNPs for HG001, HG002, HG003 and HG004 respectively, and 516,524; 587,978; 569,180 and 576,301 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

In the framework of NanoCaller for variant calling, candidate sites of SNPs and indels are defined according to an 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 optionally 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 performance of NanoCaller with WhatsHap is slightly higher but largely similar to that without WhatsHap, so we treat it as an optional step.) Afterwards, NanoCaller combines SNP and indel calls to give a final output. The details are described below.

SNP Calling in NanoCaller

There are four steps in NanoCaller to generate SNP calling result for an input genome: candidate site selection, pileup image generation of haplotype SNPs, deep learning prediction, and then optionally SNP refinement.

Candidate site selection: Candidate sites of SNPs 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. In NanoCaller, 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%.

Pileup image generation: After selecting all SNP candidate sites above, we determine a subset of SNP candidate sites as the set of highly likely heterozygous SNP sites (denoted by V). We extract long-range haplotype information from this subset of likely heterozygous SNP sites to create input images for all SNP candidate sites to be used in a convolutional neural network. This subset consists of SNP candidate sites with alternative allele frequencies in a specified range around 50%, and the default range is 40% to 60% for heterozygous site filtering. This range can be specified by the user depending upon the sequencing technology and read lengths. In detail, the procedure of pileup image generation is described below (as shown in Figure 1). For a SNP candidate site b:

1. We select sites from the set V that share at least one read with b and are at most 50,000bp away from b. For SNP calling on PacBio datasets, we set this limit at 20,000bp.
2. In each direction, upstream or downstream, of the site b, we choose 20 sites from V. 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 1 (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 1(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 1(c).
6. We obtain a 4x41x4 matrix M with entries $[\mathrm{F}^{\mathrm{t}}_{\mathrm{BD}}]_{\mathrm{B,t,D}}$ (as shown Figure 1(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 $[Q_B^t]_{B,t}$ where $Q_B^t = 1$ if B is the reference base at site *b* and 0 otherwise (as shown in Figure 1(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 1(e).

Deep learning prediction: In NanoCaller, we present a convolutional neural networks [32] for SNP prediction, as shown in Figure 3. The neural network has three convolutional layers: the first layer uses kernels of three different dimensions and combines the convolved features into 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). 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.

In order to call SNPs for a test genome, 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**.**

SNP refinement: After SNP calling, optionally, NanoCaller phases reads in the input alignment file with WhatsHap [34] using SNP calls made by NanoCaller. By default, 'distrust-genotypes' and 'include-homozygous' settings is used for phasing SNP calls, which allows WhatsHap to switching variants from hetero- to homozygous and vice versa in an optimal phasing solution. This Whatshap setting has minimal impact on NanoCaller's SNP calling performance (as shown in the Supplementary Tables S2, S3 and S4), and it can be turned off by the user. NanoCaller outputs both unphased VCF file generated by NanoCaller and phased VCF file generated by Whatshap.

Indel Calling in NanoCaller

Indel calling in NanoCaller takes a genome with phased reads as input and uses several steps below to generate indel predictions: candidate site selection, pileup image generation, deep learning prediction, and then indel sequence determination. In NanoCaller, long reads are phased with called SNPs as input of WhatsHap [34] (as described above).

Candidate site selection: 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 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.

Pileup image generation: Input image of indel candidate site is generated using the procedure below as shown in Figure 2. 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* (for PacBio datasets, we use reference sequence and alignment sequences of length 260bp).
    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 2 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. Figure 2 c) shows raw counts $C_{B,p}$ for each symbol.

    e. Let M be the 5×128 matrix with entries $M_{B,p}$ as shown in Figure 2 d).
    f. Construct a 5×128 matrix Q, with entries $[\mathrm{Q}^{\mathrm{p}}_{\mathrm{B}}]_{\mathrm{B,p}}$, where $\mathrm{Q}^{\mathrm{p}}_{\mathrm{B}} = 1$ if seq'_{ref} has symbol B at index p and 0 otherwise as shown in Figure 2 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 2 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.

Deep learning prediction: In NanoCaller, we present another convolutional neural networks [32] for indel calling, as shown in Figure 3. This neural network has a similar structure as SNP calling, and the difference is the fully connected network: 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).

Indel sequence determination: After that, 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 phased indel call at the candidate site.

Training and testing

For SNP calling, we have trained three convolutional neural network models on three different genomes. NanoCaller1 and NanoCaller2 are models trained on ONT reads of chr1-22 of HG001 (basecalled by Guppy v2.3.8) and HG002 (basecalled by Guppy v2.3.5.) respectively. NanoCaller3 is a model trained on chr1-22 of HG003 PacBio continuous long reads. For indel calling, we have trained one model, NanoCaller1, on ONT reads of chr1-22 of HG001 (basecalled by Guppy v2.3.8) and another model, NanoCaller3, on PacBio CCS reads of chr1-22 of HG003. All training datasets were aligned to GRCh38.

In NanoCaller, the SNP and indel models have 137,678 parameters in total, 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]. Each model was trained for 100 epochs, using a learning rate of 1e-3 and 1e-4 for SNP and indel models respectively. We also applied L2-norm regularization, with coefficients 1e-3 and 1e-4 for SNP and indel models respectively, to prevent overfitting of the model.

To use NanoCaller on a test genome, it is reasonable that the test genome has different coverage as the genome used for training NanoCaller. To reduce the bias caused by different coverages, 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 1 (b) by

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

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 selected sites on HG002

To further demonstrate the performance of NanoCaller and other variant callers, we select 17 genomic regions 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.

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. We would like to thank PrecisionFDA team for organizing the variant calling Truth Challenge on difficulty-to-map regions and for scoring the submissions. 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. LF conducted wet-lab experiments, including Sanger sequencing of candidate variants. 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.3.3 benchmark variant calls (HG001, HG002, HG003 and HG004), statistics within the high confidence regions are also given. 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	3,459,843	587,978	3,365,334	525,466	2,542,282,520	88.43	276,635
HG003	3,430,611	569,180	3,331,481	504,910	2,530,414,070	88.01	263,242
HG004	3,454,689	576,301	3,355,595	511,524	2,530,026,864	88.00	265,172
HX1	3,282,242	687,501	2,788,450	176,587	2,356,619,870	75.77	--

Table 2. Performances (Precision/Recall/F1 percentages) of SNP and indel predictions by NanoCaller1, NanoCaller2 and NanoCaller3 on ONT and PacBio (CCS and CLR) data and on difficult-to-map (DtM) genomic regions (without WhatsHap), along with the performance of existing variant callers.

Pred		HG001	HG002	HG003	HG004	HX1
For SNPs on ONT data	NanoCaller1	97.26/93.85/95.53	98.49/97.60/98.05	99.07/98.66/98.87	99.05/98.46/98.76	92.66/87.82/90.18
	NanoCaller2	96.96/92.96/94.92	98.37/97.76/98.07	99.15/98.58/98.87	99.08/98.51/98.80	93.56/88.26/90.84
	Medaka	92.70/89.74/91.20	98.76/98.39/98.58	99.15/98.92/99.04	99.18/98.89/99.04	97.60/94.85/96.21
	Clair	98.33/95.09/96.69	98.08/97.49/97.79	98.79/98.31/98.55	98.77/98.42/98.60	96.22/93.84/95.02
	Longshot	98.41/94.41/96.37	99.10/96.93/98.01	99.35/96.56/97.94	99.34/96.49/97.90	97.98/92.87/95.36
For SNPs on ONT data for DtM regions	Total SNPs		628,157	611,270	619,719	
	NanoCaller1		96.69/94.51/95.59	97.46/96.06/96.76	97.35/95.97/96.66	
	Medaka		95.45/94.89/95.17	96.32/96.37/96.35	96.51/96.02/96.27	
	Clair		95.50/94.62/95.06	96.65/95.93/96.29	96.38/96.07/96.23	
	Longshot		96.90/91.98/94.38	97.48/91.83/94.58	97.37/91.63/94.42	
For indels on ONT data	NanoCaller1	70.86/50.26/58.81	75.50/78.67/77.06	74.78/85.10/79.61	72.05/83.86/77.51	
	Medaka	54.46/52.86/53.65	80.53/72.40/76.25	86.15/79.97/82.95	84.42/78.08/81.13	
	Clair	79.41/49.32/60.85	78.52/64.72/70.96	80.75/70.97/75.55	80.32/69.67/74.62	
For SNPs on PacBio CCS data	NanoCaller1	97.44/99.71/98.57	99.56/99.79/99.68	99.58/99.77/99.68	99.58/99.75/99.67	
	NanoCaller3	97.28/99.80/98.53	99.53/99.83/99.68	99.54/99.81/99.68	99.54/99.79/99.67	
	Clair	99.64/99.73/99.69	99.79/99.76/99.78	99.78/99.75/99.77	99.75/99.76/99.76	
	Longshot	99.74/99.03/99.39	99.81/98.25/99.03	99.82/98.26/99.04	99.82/98.22/99.02	
For indels on PacBio CCS data	NanoCaller3		94.88/96.18/95.53	93.81/94.96/94.39	93.47/94.77/94.12	
	Clair		95.13/94.88/95.01	96.63/96.18/96.41	96.56/96.08/96.32	
For SNPs on PacBio CLR data	NanoCaller1		98.57/98.10/98.34	96.31/92.69/94.47	96.03/91.51/93.72	
	NanoCaller3		97.71/97.28/97.50	95.54/89.39/92.37	94.96/88.42/91.58	
	Clair		99.23/98.38/98.81	96.64/93.39/94.99	96.07/92.53/94.27	
	Longshot		99.14/97.68/98.41	98.69/95.76/97.21	98.64/95.11/96.85	

Table 3. 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. 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.

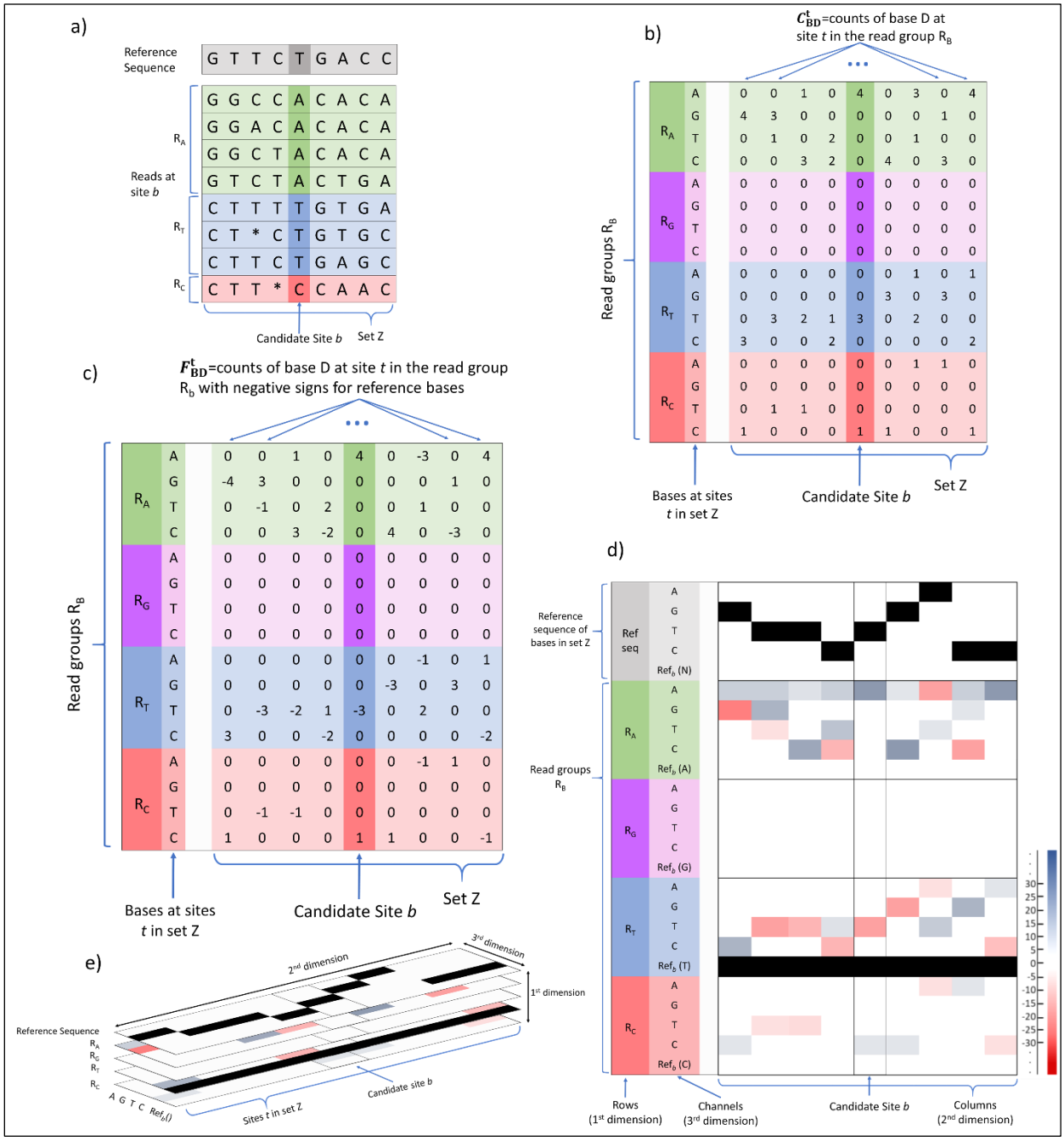
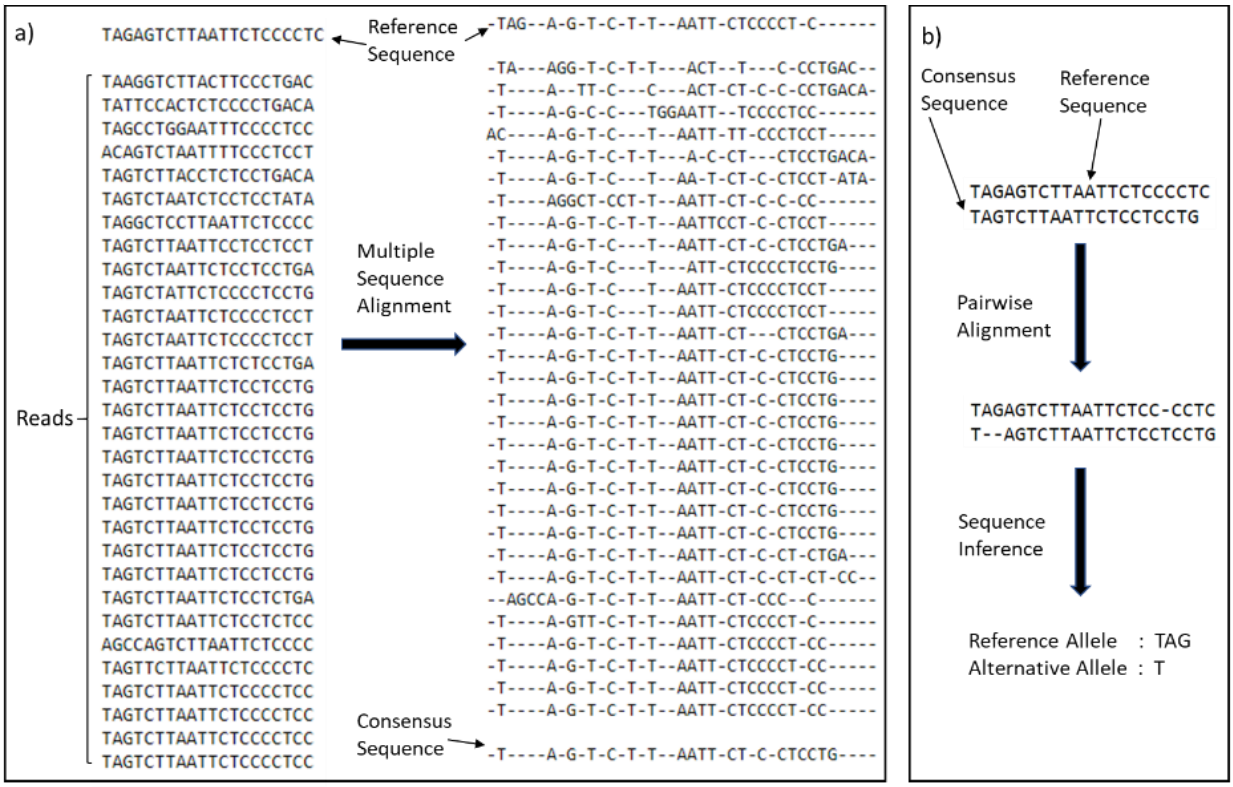


Figure 2. An example on how to construct image pileup for an indel site. **a) reference sequence and reads pileup at the candidate site before and after multiple sequence alignment, and the consensus sequence after realignment; b) reference sequence and consensus sequence at the candidate site before 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 3. 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.**

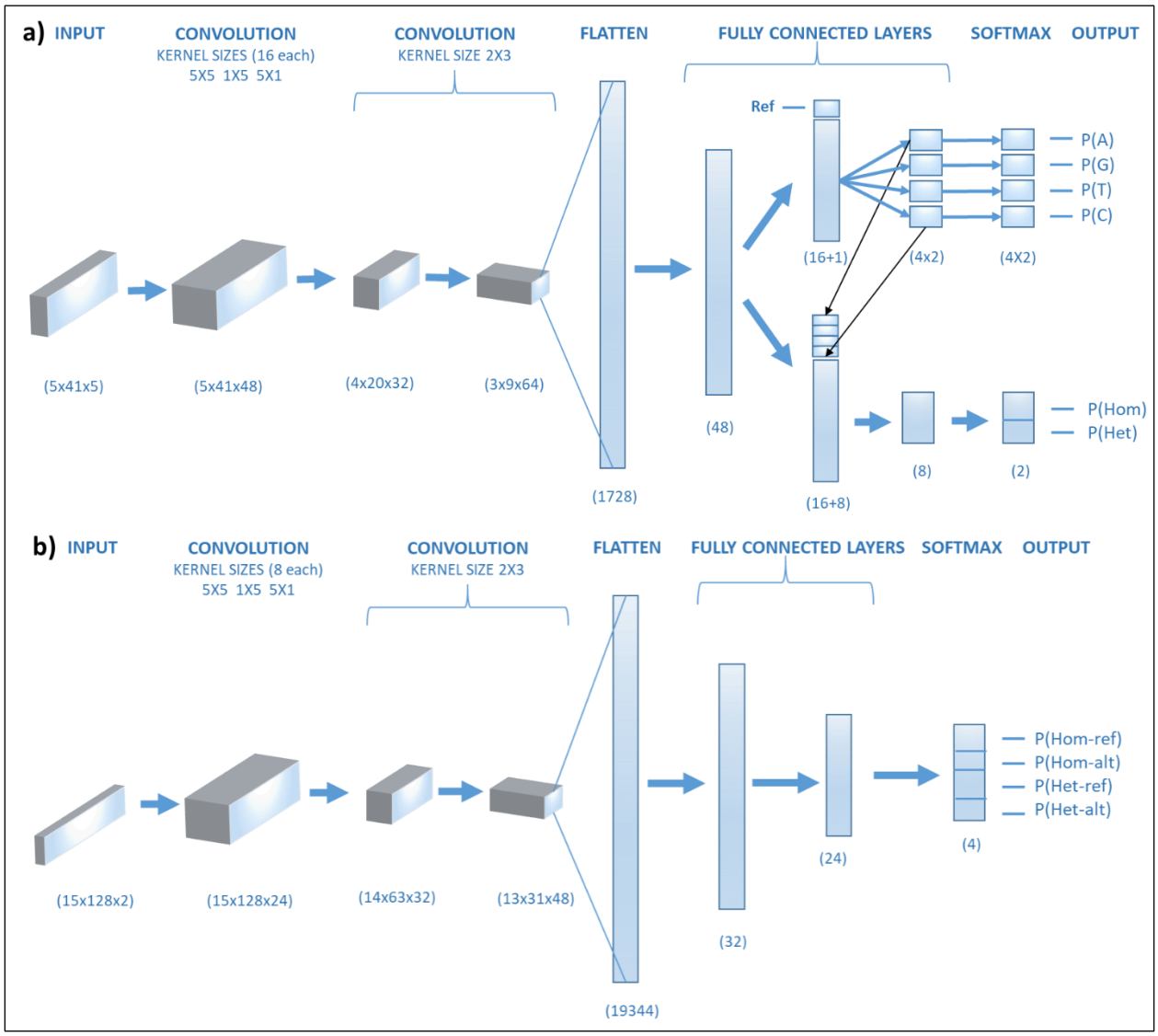


Figure 4. Performance of NanoCaller and state-of-the-art variant callers on five whole-genome Oxford Nanopore sequencing data sets. **The performance of SNP predictions on ONT reads: a) precision, b) recall, c) F1 score. d) The performance of SNP predictions on HG002 (ONT), HG003 (ONT) and HG004 (ONT) in difficult-to-map genomic regions. e) The performance of SNP predictions on HG002 (ONT), HG003 (ONT) and HG004 (ONT) in Major Histocompatibility Complex regions. The performance of indel predictions on ONT reads in non-homopolymer regions: f) precision, g) recall, h) F1 score. For HX1, the variants, which were called on high-coverage short-read data are used as benchmark with complement of difficult-to-map genomic regions used in d) as high-confidence regions.**

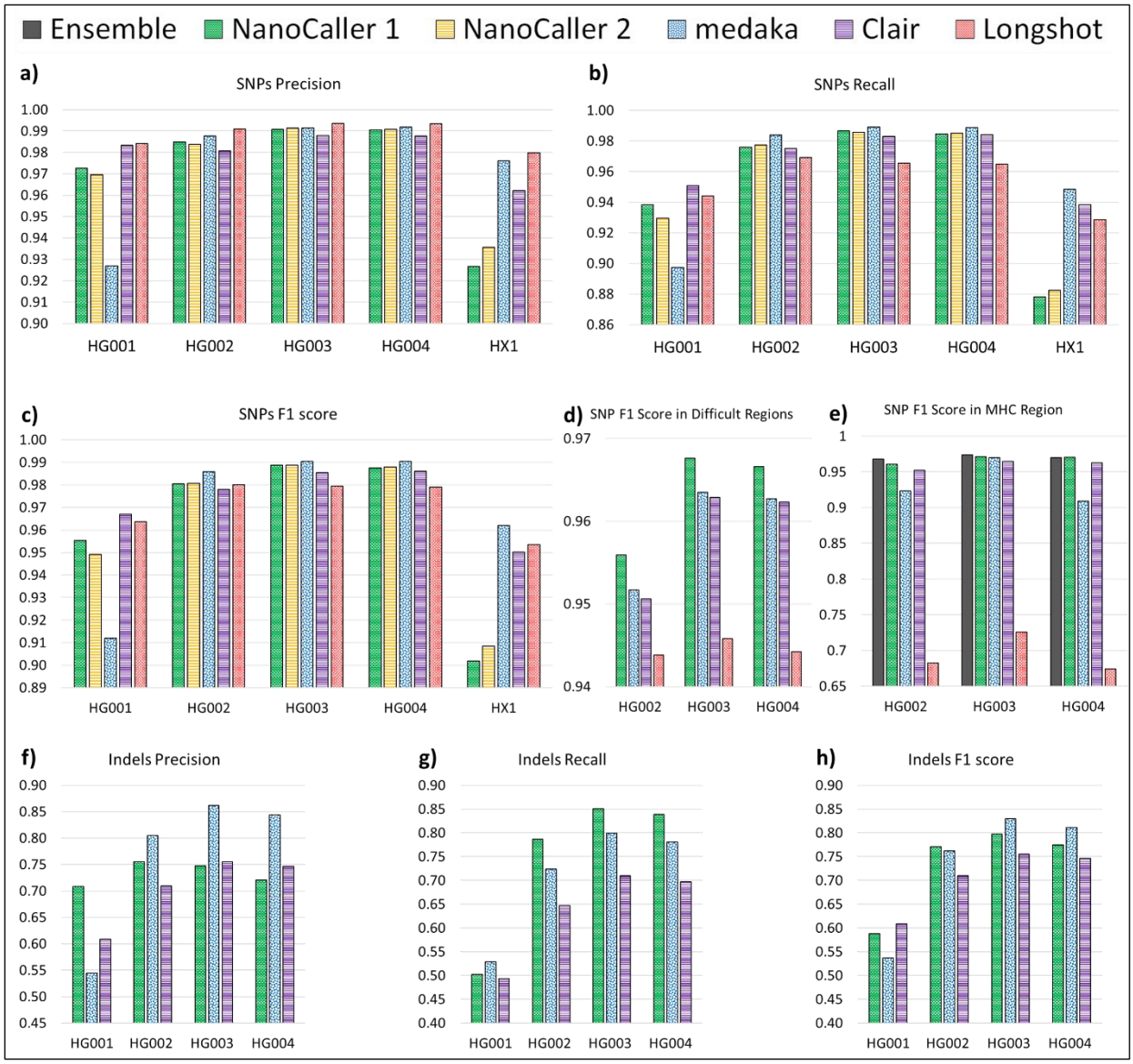


Figure 5. Performance of NanoCaller and state-of-the-art variant callers on PacBio sequencing data sets. **The performance of SNP predictions on PacBio CCS reads: a) precision, b) recall, c) F1 score. The performance of indel predictions on PacBio CCS reads: d) precision, e) recall, f) F1 score. The performance of SNP predictions on PacBio CLR reads: d) precision, e) recall, f) F1 score on PacBio**

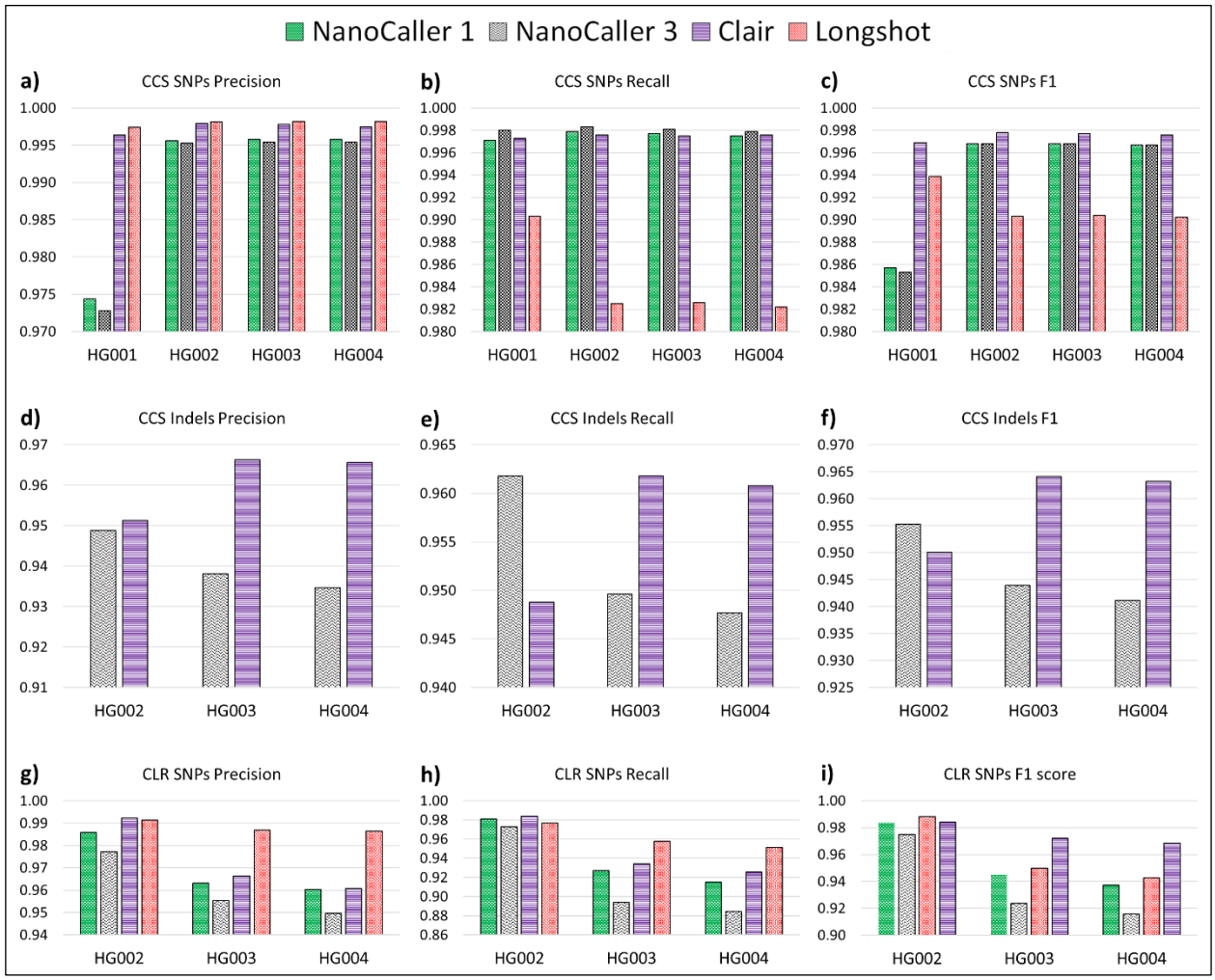


Figure 6. a) IGV plots of Nanopore, PacBio CCS and Illumina reads of HG002 genome at chr3: 5336450- 5336480. b) Sanger sequencing signal data for the same region. NanoCaller on older HG002 data correctly identified the multi-allelic SNP at chr3: 5336450 (A>T,C) shown in black box.

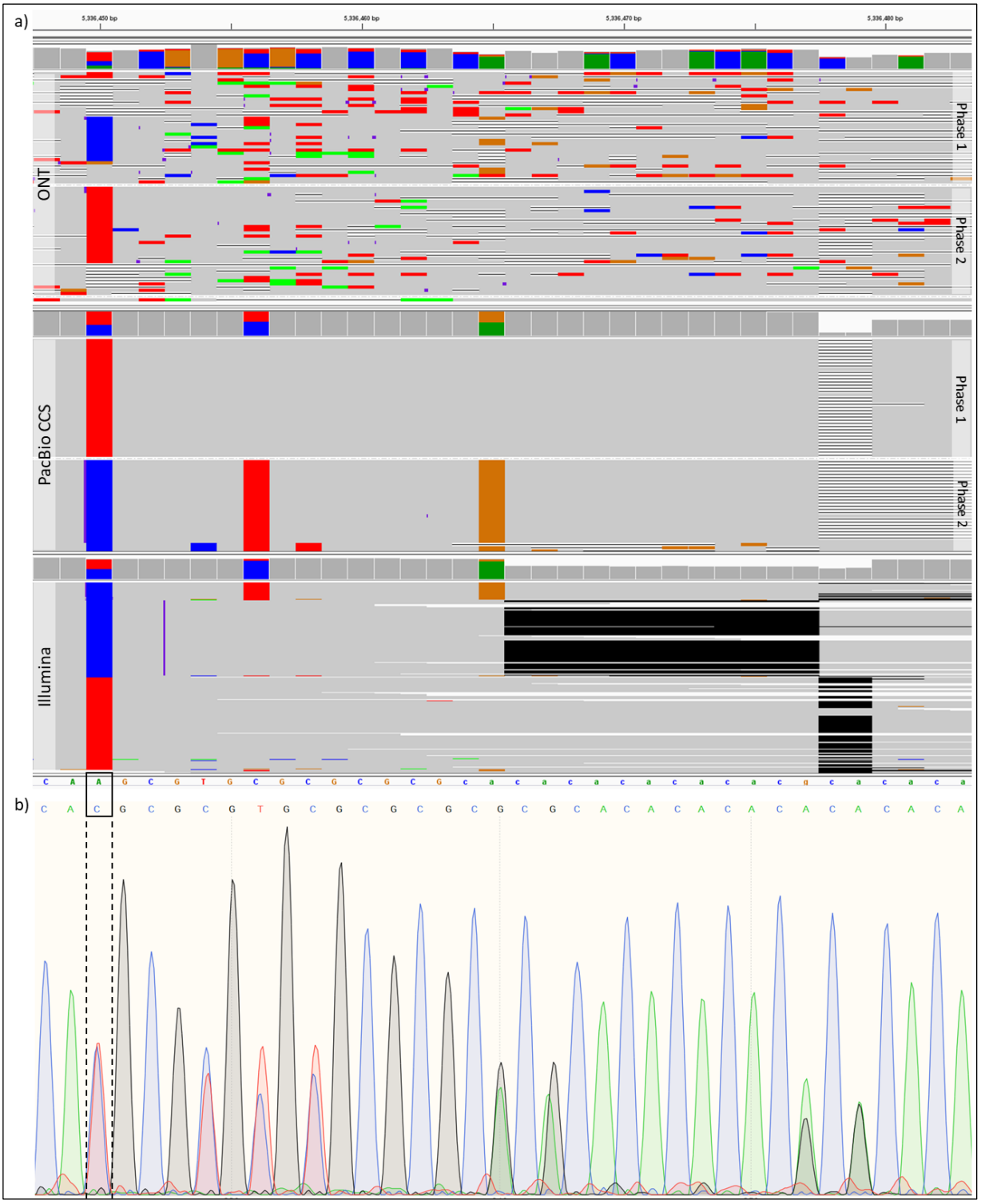
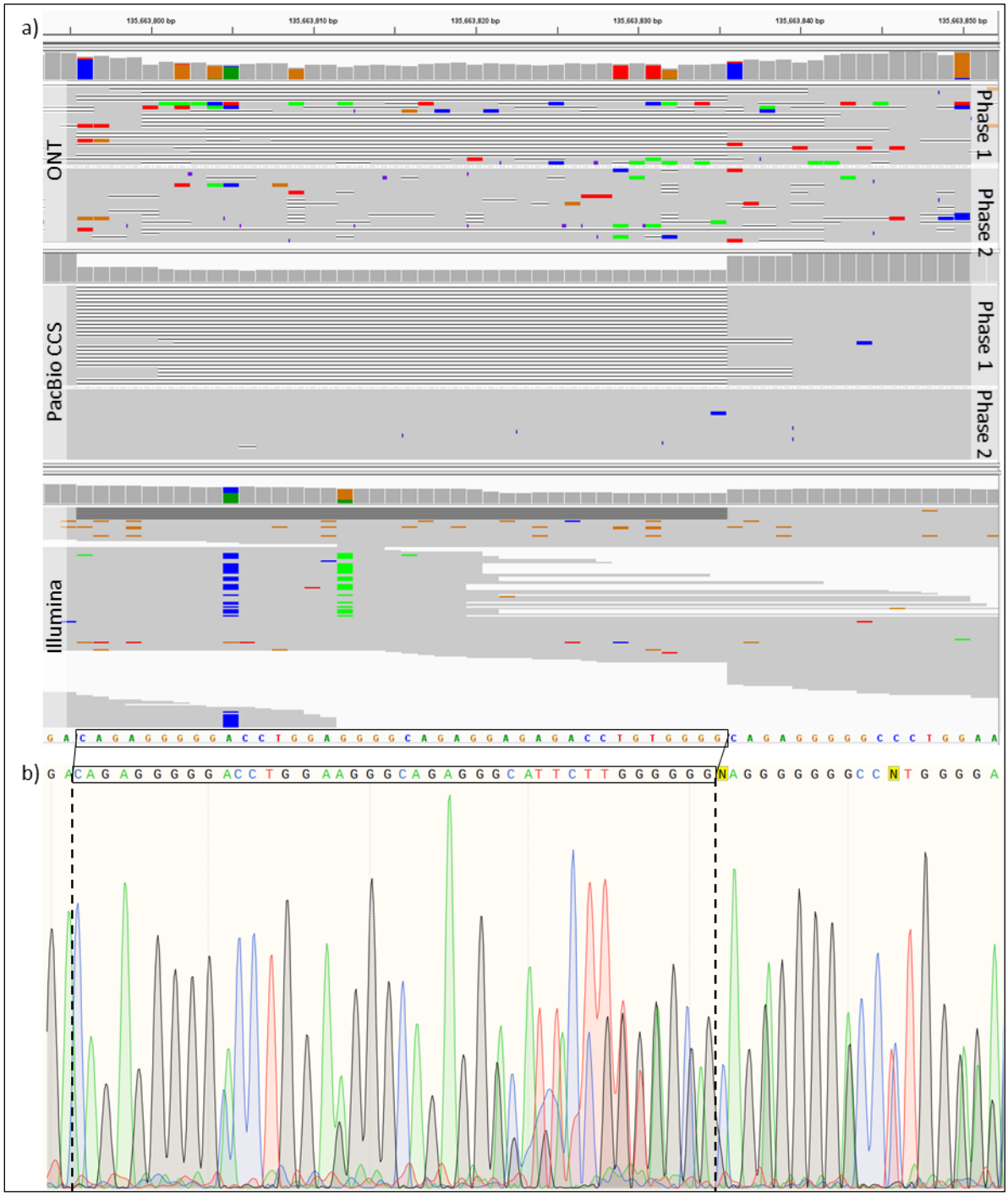



Figure 7. a) IGV plots of Nanopore, PacBio CCS and Illumina reads of HG002 genome at chr9:135663780-chr9:135663850. The 40bp long deletion shown below in black box was identified using Sanger sequencing at chr9:135663795 or chr9:135663804 (both are correct and the difference is due to two different alignments). b) shows Sanger sequencing signal data around the deletion.