Genome Biology

METHOD Open Access

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

Mian Umair Ahsan[1†], Qian Liu[1†], Li Fang[1] and Kai Wang[1,2*]

* Correspondence: wangk@email.chop.edu
†Mian Umair Ahsan and Qian Liu contributed equally to this work.
[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

Abstract

Long-read sequencing enables variant detection in genomic regions that are considered difficult-to-map by short-read sequencing. To fully exploit the benefits of longer reads, here we present a deep learning method NanoCaller, which detects SNPs using long-range haplotype information, then phases long reads with called SNPs and calls indels with local realignment. Evaluation on 8 human genomes demonstrates that NanoCaller generally achieves better performance than competing approaches. We experimentally validate 41 novel variants in a widely used benchmarking genome, which could not be reliably detected previously. In summary, NanoCaller facilitates the discovery of novel variants in complex genomic regions from long-read sequencing.

Keywords: Variant calling, Long-range haplotype, Deep learning, Difficult-to-map regions

Background

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 is thus a fundamentally important problem in using the new generations of high-throughput sequencing data to study genome variations and genome functions. 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 traditional "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 on long-read sequencing data with high error rates. 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 ~ 81% 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 ~ 19% of the human genome is inaccessible to conventional short-read sequencing technologies to find variants reliably (please refer to the "Additional file 1" on how to calculate the percentage of high-confidence genomic regions).

Oxford Nanopore (ONT) [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 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 basecalling errors of 3–15% [20] compared with short-read data (although HiFi/CCS PacBio reads and Nanopore reads generated by the latest flowcells R10.3 have lower error rates, their error rates can still be much higher than that of 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. It is also worth noting that HiFi reads after circular consensus sequencing (CCS) on PacBio long-read sequencing [21] 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 may be more suitable now for specific application scenarios such as capture-based sequencing or amplicon sequencing. As more and more long-read sequencing data becomes 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 [22] 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 [23] and its successor Clair [24] 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 they do not incorporate haplotype structure in variant calling; these methods consider each SNP separately, while a recent testing [21] with DeepVariant has shown that a phased BAM with haplotype-sorted reads can improve variant calling accuracy because grouping long reads from the same haplotype benefits neural network learning from an image of read pileup. However, this testing underutilizes the rich long-range haplotype information available from long reads, even when read phases are explicitly provided in the input BAM file. Moreover, enough variants need to be known beforehand in order to phase a BAM file. Two recent works have endeavored to improve variant calling by using phasing information from long-reads sequencing data. Longshot [25] uses a pair-Hidden Markov Model (pair-HMM) for a small local window around candidate sites to call SNPs on long-read data, and then improves genotyping of called SNPs using HapCUT2 [26] based on the most likely pair of haplotypes given the current variant genotypes. However, Longshot cannot identify indels. The Oxford Nanopore Technologies company also recently released a SNP/indel caller, i.e., Medaka [27], using deep learning on long-read data. Although not published, based on its GitHub repository, Medaka first predicts SNPs from unphased long reads, and then uses WhatsHap to phase reads. Medaka finally makes SNP and indel calling for each group of phased reads. In both methods, mutual information from long-range haplotype SNPs is ignored. 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 into a deep convolutional neural network to improve variant detection on long-read sequencing data. NanoCaller uses haplotype information for SNP calling (without requiring a phased BAM alignment input) and generates input features for a SNP candidate site using only long-range heterozygous SNPs sites that can be up to hundreds or even thousands of bases away from the candidate site; these input features are then fed into a deep convolutional neural network for SNP calling. NanoCaller does not use local neighboring bases that are immediately adjacent to the candidate site on reference genome for feature generation, which is substantially different from DeepVariant, Clairvoyante [23] and its successor Clair [24], as well as Longshot and Medaka where local neighboring bases of SNP sites are used. After that, NanoCaller uses these predicted SNP calls to phase alignment reads with WhatsHap for indel calling. Local multiple sequence alignment of phased reads around indel candidate sites is used to generate consensus sequence and feature inputs for another deep convolutional neural network to predict indel variant zygosity. We assess NanoCaller on 8 human genomes, HG001 (NA12878), HG002 (NA24385), HG003 (NA24149), HG004 (NA24143), HG005 (NA24631), HG006 (NA24694), HG007 (NA24695), and HX1 using 8 Nanopore and 4 PacBio long-read datasets. In particular, we evaluate NanoCaller in difficult-to-map genomic regions for the Ashkenazim trio (HG002, HG003 and HG004) 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 [28].

Results

Overview of NanoCaller

NanoCaller takes alignment of a long-read sequencing data aligned against a reference genome as input and generates a VCF file for predicted SNPs and indels ("Additional file 1: Fig S6"). For SNP calling in NanoCaller, candidate SNP sites are selected according to the specified thresholds for 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). Long-range haplotype features for the candidate sites (Fig. 1) are generated and fed into a deep convolutional network to

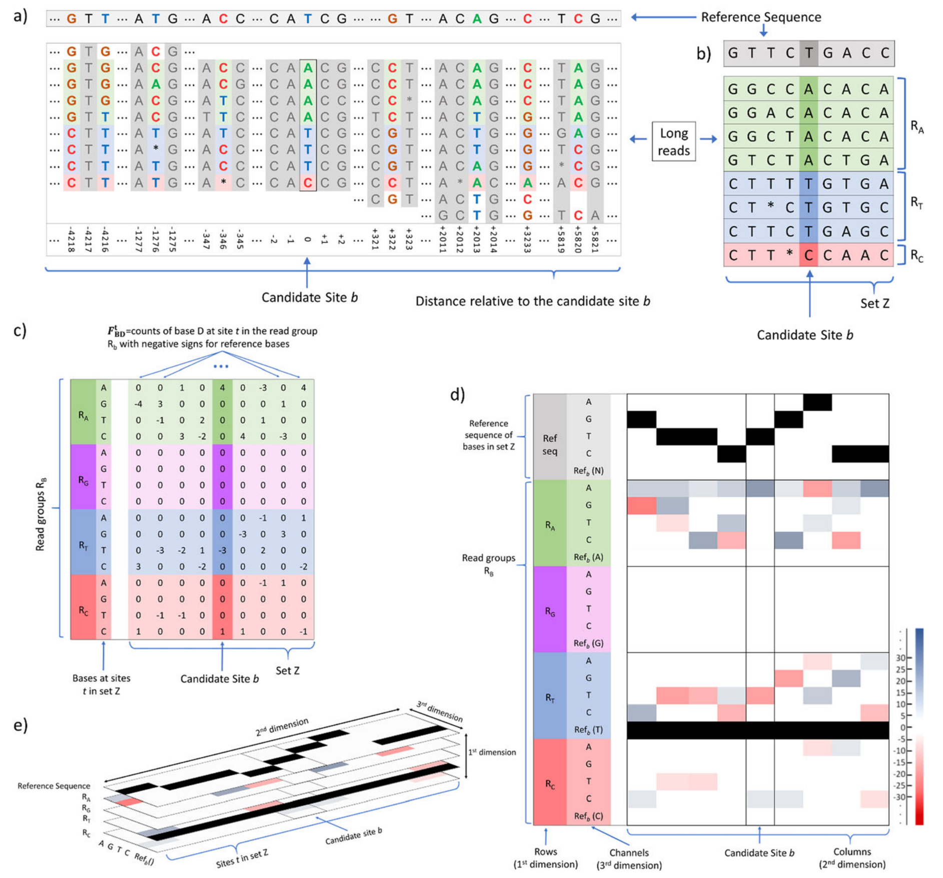
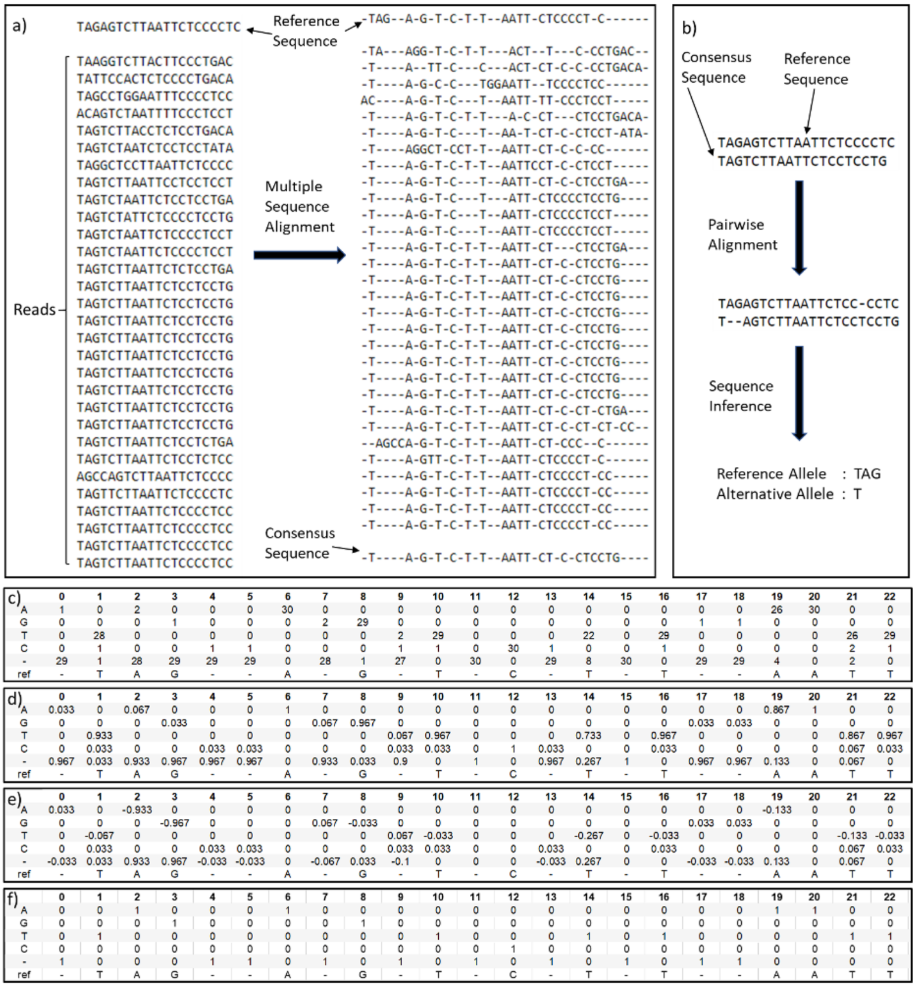


Fig. 1 An example on how to construct input features for a SNP candidate site. **a** Reference sequence and read pileups at candidate site *b* and at other genomic positions that share the same reads. The columns in gray are genomic positions that will not be used in input features for candidate site *b* as they do not satisfy the criteria for being highly likely heterozygous SNP sites. Only the columns with colored bases will be used to generate input features for site *b* and will constitute the set *Z* as described in the SNP pileups generation section of "Methods". These neighboring likely heterozygous sites can be up to thousands of bases away from candidate site *b*. **b** Reference sequence and read pileups for only the candidate site and neighboring highly likely heterozygous SNP sites. **c** Raw counts of bases at sites in the set *Z* for each read group split by the nucleotide types at site *b*. These raw counts are multiplied 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 features for NanoCaller deep convolutional neural network

distinguish true variants from false candidate sites. The predicted SNPs and long-reads are phased and then used in identification of indels. Indel candidate sites are selected according to specified minimum coverage and insertion/deletion frequency thresholds applied to each haploid read set. To reduce the effects of poor alignment on indel calling, NanoCaller uses a sliding window across reference genome to estimate indel frequency. Input features for indel candidate sites are generated using multiple sequencing alignment on the set of diploid reads and on each set of haploid reads (Fig. 2). After that, another deep convolutional neural network is used to determine indel calls and assign variant call quality scores. Allele sequence for the indels is predicted by comparing consensus sequences against reference sequence.

The performance of NanoCaller is evaluated on both Oxford Nanopore and PacBio reads, and compared with performances of Medaka (v0.10.0), Clair (v2.0.1), Longshot



```
TAGAGTCTTAATTCTCCCCTC

TAAGGTCTTACTTCCCTGAC
TATTCCACTCTCCCCTGACA
TAGCCTGGAATTTCCCCTCC
ACAGTCTAATTTTCCCTCCT
TAGTCTTACCTCTCCTGACA
TAGTCTAATCTCCTCCTATA
TAGGCTCCTTAATTCTCCCC
TAGTCTTAATTCCTCCTCCT
TAGTCTAATTCTCCTCCTGA
TAGTCTATTCTCCCCTCCTG
TAGTCTAATTCTCCCCTCCT
TAGTCTAATTCTCCCCTCCT
TAGTCTTAATTCTCTCCTGA
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCCTG
TAGTCTTAATTCTCCTCTGA
TAGTCTTAATTCTCCTCTCC
AGCCAGTCTTAATTCTCCCC
TAGTTCTTAATTCTCCCCTC
TAGTCTTAATTCTCCCCTCC
TAGTCTTAATTCTCCCCTCC
TAGTCTTAATTCTCCCCTCC
TAGTCTTAATTCTCCCCTCC
```

```
-TAG--A-G-T-C-T-T--AATT-CTCCCCT-C------

-TA---AGG-T-C-T-T---ACT--T---C-CCTGAC--
-T----A--TT-C---C---ACT-CT-C-C-CCTGACA-
-T----A-G-C-C---TGGAATT--TCCCCTCC------
AC----A-G-T-C---T--AATT-TT-CCCTCCT-----
-T----A-G-T-C-T-T---A-C-CT---CTCCTGACA-
-T----A-G-T-C---T--AA-T-CT-C-CTCCT-ATA-
-T----AGGCT-CCT-T--AATT-CT-C-C-CC------
-T----A-G-T-C-T-T--AATTCCT-C-CTCCT-----
-T----A-G-T-C---T--AATT-CT-C-CTCCTGA---
-T----A-G-T-C---T---ATT-CTCCCCTCCTG----
-T----A-G-T-C---T--AATT-CTCCCCTCCT-----
-T----A-G-T-C---T--AATT-CTCCCCTCCT-----
-T----A-G-T-C-T-T--AATT-CT---CTCCTGA---
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
-T----A-G-T-C-T-T--AATT-CT-C-CT-CTGA---
-T----A-G-T-C-T-T--AATT-CT-C-CT-CT-CC--
--AGCCA-G-T-C-T-T--AATT-CT-CCC--C------
-T----A-GTT-C-T-T--AATT-CTCCCCT-C------
-T----A-G-T-C-T-T--AATT-CTCCCCT-CC-----
-T----A-G-T-C-T-T--AATT-CTCCCCT-CC-----
-T----A-G-T-C-T-T--AATT-CTCCCCT-CC-----
-T----A-G-T-C-T-T--AATT-CTCCCCT-CC-----

-T----A-G-T-C-T-T--AATT-CT-C-CTCCTG----
```

```
TAGAGTCTTAATTCTCCCCTC
TAGTCTTAATTCTCCTCCTG

TAGAGTCTTAATTCTCC-CCTC
T--AGTCTTAATTCTCCTCCTG
```

Reference Allele : TAG
Alternative Allele : T

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

Fig. 2 An example on how to construct input features for an indel candidate site. **a** Reference sequence and read pileup at the candidate site before and after multiple sequence alignment, and the consensus sequence. **b** Reference sequence and consensus sequence at the candidate site before and after pairwise alignment, and the inferred sequence. **c** Raw counts 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 for NanoCaller deep convolutional neural network

(v0.4.1), DeepVariant (v.1.0.0) and WhatsHap (v1.0) with their default parameters for each type of sequencing technology. By default, evaluation is on benchmark variants in high-confidence intervals of chromosomes 1-22 of the GRCh38 reference genome, unless stated otherwise. RTG tools (the commands for *vcfeval* submodule can be found in the “Additional file 1: Pages 26-27”) [29] is used to calculate various evaluation metrics, such as precision, recall and F1. For whole-genome analysis, we show each variant caller’s performance using its recommended quality threshold if available, e.g., NanoCaller, Clair, Longshot, and DeepVariant; for Medaka, we calculate the quality score thresholds that give highest F1 score for each genome using *vcfeval*, and use their average as the final quality score cut-off to report results. In particular, variant calling performance analysis in difficult-to-map genomic regions requires different quality score cut-offs from whole-genome analysis due to highly specific error profiles of these difficult-to-map regions. Therefore, we use the average best quality score cut-off (in the same manner as the quality score cut-off is determined for Medaka) for each variant caller in each type of difficult genomic region.

In the “Results” section, we present performances of five NanoCaller models: ONT-HG001 (trained on HG001 ONT reads), ONT-HG002 (trained on HG002 ONT reads), CCS-HG001 (trained on HG001 PacBio HiFi/CCS reads), CCS-HG002 (trained on HG002 PacBio HiFi/CCS reads), and CLR-HG002 (trained on HG002 PacBio CLR reads); the first four datasets have both SNP and indel deep learning models, whereas CLR-HG002 consists of only a SNP model. All NanoCaller HG001 models are trained using v3.3.2 of GIAB benchmark variant calls, whereas all NanoCaller HG002 models are trained using v4.2.1 of GIAB benchmark variant calls. Sequencing datasets used for training were aligned to the GRCh38 reference genome. For performance evaluation, the latest available GIAB benchmark variants are used for each genome, i.e., v3.3.2 for HG001 and HG005-7, and v4.2.1 for the Ashkenazim trio HG002-4. For HX1, variant calls produced by GATK on 300× Illumina reads of HX1 are used as benchmark, and high-confidence intervals for HX1 were created by removing difficult-to-map regions from chromosomes 1-22.

Evaluation of NanoCaller on Oxford Nanopore sequencing

Performance on SNP calling

We compared NanoCaller’s SNP calling performance on Oxford Nanopore sequencing reads against several existing tools. For NanoCaller, we used alternative allele frequency threshold of 0.15 for SNP candidates. For testing Clair, we used “*1_124x ONT*” model trained on 124× coverage HG001 ONT reads using v3.3.2 GIAB benchmark variants, whereas for Medaka we used “*r941_min_diploid_snp_model*” model for testing, which is trained on several bacteria and eukaryotic read datasets and variant call sets. Longshot pair-HMM model is not trained on any genome as it estimates parameters during each run. We compared the performances of these methods on eight genomes: HG001-7 and HX1 under two testing strategies: cross-genome testing and cross-reference testing.

Cross-genome testing is critical to demonstrate the performance of a variant caller when used in a real-world scenario: the machine learning model of a variant caller is trained on one set of genomes and tested on other genomes. Under this testing

Table 1 Performances (F1 scores in %) of SNP and indel predictions by NanoCaller, Medaka, Clair, Longshot, and DeepVariants on ONT and PacBio (CCS and CLR) datasets. This evaluation is based on v3.3.2 benchmark variants for HG001 and HG005-7, and v4.2.1 benchmark variants for the Ashkenazim trio (HG002, HG003, and HG004). Bonito and R10.3 refer to different versions of the HG002 ONT datasets

Prediction	Variant caller	HG001	HG002	HG003	HG004	HG005	HG006	HG007	HX1	Bonito	R10.3
SNPs on ONT data in high-confidence intervals	NanoCaller ONT-HG001	98.58	98.35	98.99	98.97	98.10	98.23	97.81	98.45	99.33	98.28
	NanoCaller ONT-HG002	98.63	**98.66**	**99.09**	**99.11**	**98.38**	98.43	98.06	98.60	**99.34**	**98.44**
	Medaka	**99.03**	98.59	99.02	99.04	98.17	98.50	98.24	**98.94**	99.24	96.94
	Clair	98.79	97.77	98.60	98.58	97.73	97.90	97.50	98.53	98.75	90.44
	Longshot	98.78	98.03	97.88	97.90	98.34	**98.53**	**98.51**	98.59	98.59	98.18
Indels on ONT data in high-confidence intervals	NanoCaller ONT-HG001	**57.33**	53.94	**58.52**	**57.71**	56.31	56.14	53.78	73.67	**62.07**	**61.59**
	NanoCaller ONT-HG002	56.69	**54.37**	58.47	57.69	**56.93**	**56.56**	**54.44**	73.90	61.17	60.56
	Medaka	48.67	48.10	53.59	50.19	55.89	52.49	51.83	**81.13**	51.03	53.09
	Clair	49.72	47.64	52.06	51.20	52.58	51.90	50.63	80.59	50.11	44.80
Indels on ONT data in non-homopolymer regions	NanoCaller ONT-HG001	**87.65**	82.28	**87.93**	87.93	81.92	85.70	83.41	**59.47**	**86.12**	**84.43**
	NanoCaller ONT-HG002	87.19	**82.80**	**87.93**	**88.04**	**82.60**	**86.10**	**83.92**	59.17	85.76	83.51
	Medaka	82.07	78.70	85.74	84.23	80.97	84.41	82.91	55.17	78.24	78.75
	Clair	75.25	70.06	75.55	74.85	72.60	75.92	75.04	58.43	70.99	62.93
SNPs on PacBio CCS data in high-confidence intervals	NanoCaller CCS-HG001	99.25	99.80	99.79	99.71						
	NanoCaller CCS-HG002	99.17	99.80	99.79	99.75						
	Clair	99.66	99.84	99.72	99.79						
	Longshot	99.37	99.03	99.05	99.05						
	DeepVariant	**99.82**	**99.93**	**99.91**	**99.84**						
Indels on PacBio CCS data in high-confidence intervals	NanoCaller CCS-HG001	92.67	93.30	93.42	93.10						
	NanoCaller CCS-HG002	93.13	94.10	94.34	93.97						
	Clair	94.87	96.71	97.51	95.57						
	DeepVariant	**98.21**	**99.28**	**99.48**	**98.42**						

Table 1 Performances (F1 scores in %) of SNP and indel predictions by NanoCaller, Medaka, Clair, Longshot, and DeepVariants on ONT and PacBio (CCS and CLR) datasets. This evaluation is based on v3.3.2 benchmark variants for HG001 and HG005-7, and v4.2.1 benchmark variants for the Ashkenazim trio (HG002, HG003, and HG004). Bonito and R10.3 refer to different versions of the HG002 ONT datasets *(Continued)*

Prediction	Variant caller	HG001	HG002	HG003	HG004	HG005	HG006	HG007	HX1	Bonito	R10.3
SNPs on PacBio CLR data in high-confidence intervals	NanoCaller CLR-HG002	94.42	**98.75**	94.41	93.41						
	Clair	95.83	98.38	**94.89**	**94.15**						
	Longshot	**96.81**	98.41	94.35	93.27						

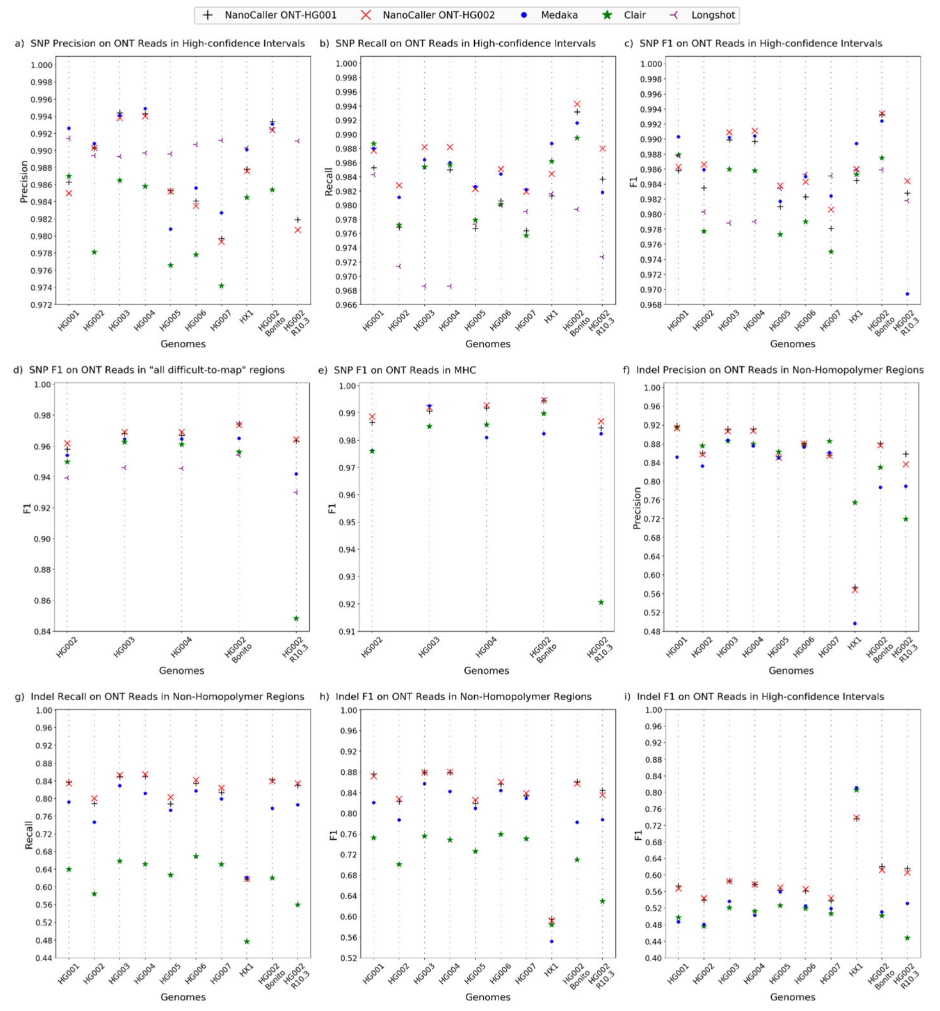


Fig. 3 Performances of NanoCaller and other variant callers on ten ONT datasets. SNP performance on whole-genome high-confidence intervals: **a** precision, **b** recall, **c** F1 score. F1 scores of SNP performances on **d** "all difficult-to-map" regions and **e** MHC. Indel performance non-homopolymer regions: **f** precision, **g** recall, **h** F1 score. **i**: F1 score of indel performance in whole-genome high-confidence intervals

strategy, the performance of SNP calling for NanoCaller, together with three other variant callers, Medaka, Clair, and Longshot, is shown in Table 1 and Fig. 3 on Nanopore reads of eight genomes.

On the Ashkenazim trio (HG002, HG003, HG004), NanoCaller has better performance than Medaka, Clair, and Longshot in terms of F1 score as shown in Table 1 and Fig. 3c: F1 scores of NanoCaller are 98.35, 98.99, 98.97% for ONT-HG001 NanoCaller model and 98.66, 99.09, 99.11% for ONT-HG002 NanoCaller model, compared to Medaka (98.59, 99.02, 99.04%), to Clair (97.77, 98.60, 98.58%), and Longshot (98.03, 97.88, 97.90%), on HG002, HG003, and HG004 respectively. In particular, ONT-HG002 NanoCaller model exceeds Longshot's by 1.2% F1 score on HG003 and HG004. More details of the performances (precision and recall) can be found in Fig. 3a and b, and "Additional file 2: Table S30." Furthermore, we evaluated NanoCaller on two additional HG002 ONT datasets. The first data is produced by R10.3 flowcells and basecalled by Guppy 4.0.11, and the second dataset is produced by R9.4.1 flowcells and basecalled with Bonito 0.30. We found that NanoCaller models performed better than other variant callers in terms of F1 score by significant margins. For example, on HG002 ONT data generated by R10.3 flowcells, the F1 scores are 99.33%, 99.34%, 99.24%, 98.75%, and 98.59% for ONT-HG001, ONT-HG002, Medaka, Clair, and Longshot respectively, whereas on HG002 Bonito dataset, the F1 scores are 98.28%, 98.44%, 96.94%, 90.44%, and 98.18% for ONT-HG001, ONT-HG002, Medaka, Clair, and Longshot respectively. NanoCaller achieves better F1 score than all other methods. More details for these performances can be found in Fig. 3a and b, and "Additional file 2: Table S35."

For HG001 and HG005-7, NanoCaller performs competitively against other methods as shown in Table 1 and Fig. 3: On HG001, F1 scores are 98.58% for ONT-HG001, 98.63% for ONT-HG002, 98.79% for Clair, 98.78% for Longshot, and 99.03% for Medaka. On HG005/HG006/HG007, F1 scores are 98.10/98.23/97.81% for ONT-HG001, 98.38/98.43/98.06% for ONT-HG002, 98.17/98.50/98.24% for Medaka, 97.73/97.90/97.50% for Clair, and 98.34/98.53/98.51% for Longshot, respectively. It is clear that sometimes NanoCaller has best F1 score (for example on HG005), but sometimes other methods show best F1 score (such as Longshot on HG007 and Medaka on HG001). Please note that benchmark variants of HG001and HG005-7 are older (v3.3.2) than the benchmark variants of the Ashkenazim trio HG002-4 (v4.2.1). The Ashkenazim trio HG002-4 has a larger variant callset than HG001 and HG005-7 (370–400 k more SNPs per genome than HG001), and a larger high-confidence region which includes more difficult genomic regions (at least 200 mbp larger than HG001 and covering an extra 7% of the reference genome). This might contribute to the performance variation between different methods.

Next, we show SNP calling performance on HX1 genome sequenced by our lab. On 48× coverage HX1 reads re-basecalled with Guppy 4.5.2 by us, NanoCaller models perform slightly better than Clair and Longshot (F1 scores are 98.45% for ONT-HG001, 98.60% for ONT-HG002, 98.53% for Clair, 98.59% for Longshot, and 98.94% for Medaka). This demonstrates NanoCaller's ability to accurately identify variants in non-GIAB datasets in real-life applications.

Lastly, we show that the performance of NanoCaller SNP models is independent of the reference genome used. Under this cross-reference testing, we evaluated ONT-HG001 model (trained on reads aligned against GRCh38) on HG002 ONT reads aligned to both GRCh38 and GRCh37 reference genomes. For reads aligned to GRCh38, we obtained 99.03%, 97.69%, and 98.35% for precision, recall, and F1 score respectively. Whereas for reads aligned to GRCh37, we obtained 98.99%, 97.70% and 98.34% for precision, recall and F1 score respectively. The similar performance on GRCh38 and GRCh37 indicates that NanoCaller could be used on alignment generated by mapping to different reference genomes.

Performance of SNP calling in difficult-to-map genomic regions

We further demonstrate that NanoCaller has a unique advantage in calling SNPs in difficult-to-map genomic regions. We tested both ONT-HG001 SNP model (trained on HG001 ONT reads with v3.3.2 benchmark variants) and ONT-HG002 SNP model (trained on HG002 ONT reads with benchmark variants v4.2.1) on ONT reads of the three genomes of the Ashkenazim trio together with other variant callers. v4.2.1 of

GIAB benchmark variants for the trio HG002-4 are used for testing because they have a more exhaustive list of true variants and high-confidence intervals in difficult-to-map genomic regions, as shown in “Additional file 2: Table S27.” Difficult-to-map genomic regions here are defined by GA4GH Benchmarking Team and the Genome in a Bottle Consortium, and are downloaded as BED files from GIAB v2.0 genome stratification. These regions contain all tandem repeats, all homopolymers > 6 bp, all imperfect homopolymers > 10 bp, all low mappability regions, all segmental duplications, GC < 25% or > 65%, bad promoters, and other difficult regions such major histocompatibility complex. We intersected the BED files with high-confidence intervals for each genome in the trio and evaluated SNP performance in the intersected regions. As shown in “Additional file 2: Table S27,” each genome has at least 600 k SNPs in the intersection of difficult-to-map 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 is shown in Fig. 3d and Table 2 for NanoCaller and other methods. For HG002/HG003/HG004, F1 scores are 95.80/96.83/96.70% for ONT-HG001, 96.18/96.92/96.92% for ONT-HG002, 95.41/96.46/96.46% for Medaka, 94.98/96.27/96.12% for Clair, and 93.95/94.61/94.55% for Longshot, respectively. NanoCaller performs better than all other variant callers for each genome. In “Additional file 2: Table S33,” we show performances of NanoCaller SNP models trained on v3.3.2 benchmark variants of HG002, and again NanoCaller performs significantly better than other variant callers. In “Additional file 2: Table S33,” we further show a detailed breakdown of performances in the difficult-to-map regions and demonstrate that NanoCaller generally performs better than other variant callers for SNPs in each of the following categories of difficult-to-map regions: segmental duplications, tandem and homopolymer repeats, low mappability regions, and major histocompatibility complex.

To further investigate NanoCaller’s performance, we split difficult-to-map regions into different subgroups according to their length: 0–10 kbp, 10–100 kbp, 100–500

Table 2 Performances (F1 scores in %) of SNP predictions in difficult-to-map regions and Major Histocompatibility Complex (MHC) by NanoCaller, Medaka, Clair, and Longshot on ONT data. These evaluations are performed against v4.2.1 benchmark variants for the Ashkenazim trio (HG002, HG003, and HG004), whereas “HG002 Bonito” and “HG002 R10.3” are different HG002 ONT datasets

Prediction	**Variant caller**	**HG002**	**HG003**	**HG004**	**HG002 Bonito**	**HG002 R10.3**
SNPs on ONT data in difficult-to-map regions	NanoCaller ONT-HG001	95.80	96.83	96.70	**97.44**	96.34
	NanoCaller ONT-HG002	**96.18**	**96.92**	**96.92**	97.38	**96.44**
	Medaka	95.41	96.46	96.46	96.51	94.20
	Clair	94.98	96.27	96.12	95.63	84.83
	Longshot	93.95	94.61	94.55	95.42	93.00
SNPs on ONT data in MHC	NanoCaller ONT-HG001	98.65	99.06	99.18	99.45	98.46
	NanoCaller ONT-HG002	**98.86**	**99.19**	**99.28**	**99.46**	**98.69**
	Medaka	97.62	99.25	98.10	98.24	98.24
	Clair	97.60	98.51	98.57	98.97	92.06
	Longshot	68.52	73.13	69.40	68.48	68.41

kbp, and > 500 kbp, and tested NanoCaller, Medaka, Clair, and Longshot on HG002, HG003, and HG004. We found that when the length of interval increases, the performance advantage of NanoCaller over other methods becomes larger: for example on HG004, NanoCaller's F1 score is 0.02 higher than Longshot for 0–10 kbp subgroup, whereas NanoCaller's F1 score is 0.1793 higher than Longshot for > 500 kbp. This demonstrates NanoCaller can benefit SNP calling with long difficult-to-map regions compared to other methods. The details of these performances can be found in "Additional file 2: Table S41."

Finally, NanoCaller team participated in PrecisionFDA truth challenge v2 for difficult-to-map genomic 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 model (trained on ONT reads of HG001 basecalled with Guppy2.3.8), and Medaka and Clair models described above. The challenge consisted of Guppy3.6 basecalled ONT reads for HG003 and HG004 to predict variant calls, which were then evaluated on GIAB v4.1 benchmark variants of HG003 and HG004 that were made public after the challenge ended. At the conclusion of the challenge, GIAB released v4.2 benchmark variants. Our ensemble submission won the award for best performance in major histocompatibility complex (MHC) using Nanopore reads [30]. "Additional file 1: Fig S4 (e) and (f)" and "Additional file 1: Table S11" show the F1 score of SNPs and overall variant performance of the ensemble, NanoCaller model (trained on ONT reads of HG001 basecalled with Guppy2.3.8), Medaka, and Clair. While the ensemble performs better than all other variant callers, in general, NanoCaller's performance on HG002 and HG004 is very close to the ensemble and is significantly better than the performances of Medaka and Clair (F1 scores NanoCaller: 98.53%, 99.07% vs ensemble: 98.97%, 99.17%; Medaka: 97.15%, 94.29% and Clair: 97.55%, 98.59% for HG002 and HG004 respectively). NanoCaller always outperforms Longshot for SNP calling in MHC regions. Therefore, this is an independent assessment of the real-world performance of NanoCaller in detecting variants in complex genomic regions. For performance of NanoCaller and other variant callers on MHC using latest ONT reads for HG002-4, please refer to Fig. 3e, Table 2, and "Additional file 2: Table S33." Meanwhile, we ran PEPPER-DeepVariant on ONT data basecalled with Guppy v4.2.2 for HG002, HG003, and HG004, and then evaluated the SNP/indel calling against benchmark v4.2.1 (as shown in "Additional file 2: Table S42"), and we found that NanoCaller performs better than PEPPER-DeepVariant on the MHC regions.

Performance on indel calling

We tested NanoCaller indel models and other variant callers on ONT reads of eight genomes: HG001-7 and HX1, similar to SNP evaluation. The settings of NanoCaller are given below: to determine indel candidates, thresholds for haplotype insertion allele frequency and deletion frequency were set to 0.4 and 0.6, respectively, due to the abundance of deletion errors in Nanopore reads. Since Nanopore sequencing is unreliable in homopolymer regions, we break down performances for each genome into three categories: high-confidence intervals, homopolymer regions, and non-homopolymer regions. Homopolymers regions for indel evaluation are defined as perfect homopolymer regions of length greater than or equal to 4 bp as well as imperfect homopolymer regions of length greater than 10 bp. Non-homopolymer regions are created by removing homopolymer regions from high-confidence intervals (more details on homopolymer and non-homopolymer regions are shown in "Additional file 1: Pages 5,6, and 27"). Performances evaluated by RTG *vcfeval* are shown in Table 1 and Fig. 3f, g, and h for NanoCaller together with Medaka and Clair. According to the F1 scores in Table 1, NanoCaller performs better than Clair by 8–10% and Medaka by 2–5% in non-homopolymer regions. It is also worth noting that NanoCaller has a higher recall than Medaka and especially Clair: for example, NanoCaller has ~ 15–20% and ~ 1–4% higher recall than Clair and Medaka respectively on all 7 genomes HG001-7. On high-confidence and homopolymer regions, NanoCaller also achieves higher F1 scores than Medaka and Clair; more details can be found in Table 1 and "Additional file 2: Table S31." This demonstrates the improved performance of NanoCaller for indel calling. In particular, "Additional file 1: Fig S7" shows a true insertion and a true deletion in HG002 ONT reads that are predicted correctly by NanoCaller but missed by Clair and Medaka. Both indels show high discordance in position of indels among the overlapping long reads, which makes it harder to identify these indels. Furthermore, "Additional file 1: Fig S1" further shows concordance of ground truth variants in high-confidence regions (including homopolymer repeat regions) of the Ashkenazim trio correctly predicted by NanoCaller, Medaka and Clair. "Additional file 1: Fig S1 (b)" shows each tool has a significant number (ranging from 19–60 k) of correctly predicted indel calls that are not correctly predicted by other variant callers.

Performance on PacBio sequencing data

We evaluated NanoCaller on PacBio HiFi/CCS and CLR datasets of four genomes: HG001, HG002, HG003, and HG004. For CCS datasets, we evaluated NanoCaller SNP models CCS-HG001 (trained on HG001 CCS reads using benchmark variants v3.3.2) and CCS-HG002 (trained on HG002 CCS reads using benchmark variants v4.2.1). The settings of compared tools are given below: for NanoCaller, the minimum alternative allele frequency threshold for SNP calling was set to 0.15 to identify SNP candidates for both PacBio CCS and CLR reads. For Clair, the PacBio model trained on HG001 and HG005 was used for testing CCS reads, whereas the model trained on seven genomes HG001-HG007 was used for testing CLR reads; both Clair models used v3.3.2 benchmark variants for training. The provided PacBio model in the new DeepVariant release v1.0.0 is used for testing, and this model was trained on CCS reads of HG001-HG006 with v3.3.2 benchmark variants for HG001, HG005-6, and v4.2 for HG002-HG004.

The results for SNP performance on CCS reads are shown in Table 1 and Fig. 4a–c along with Clair, Longshot, and DeepVariant. It can be seen from Table 1 and Fig. 4 that on the Ashkenazim trio, both NanoCaller models (CCS-HG001 and CCS-HG002) perform significantly better than Longshot, and NanoCaller shows competitive performance against Clair (F1 scores are CCS-HG001: 99.80, 99.79, 99.71%; CCS-HG002: 99.80, 99.79, 99.75% vs Clair: 99.84, 99.72, 99.79% vs Longshot: 99.03, 99.05, 99.05% vs DeepVariant: 99.93, 99.93, 99.84% for HG002, HG003, and HG004 respectively). More details performance can be found in "Additional file 2: Table S37."

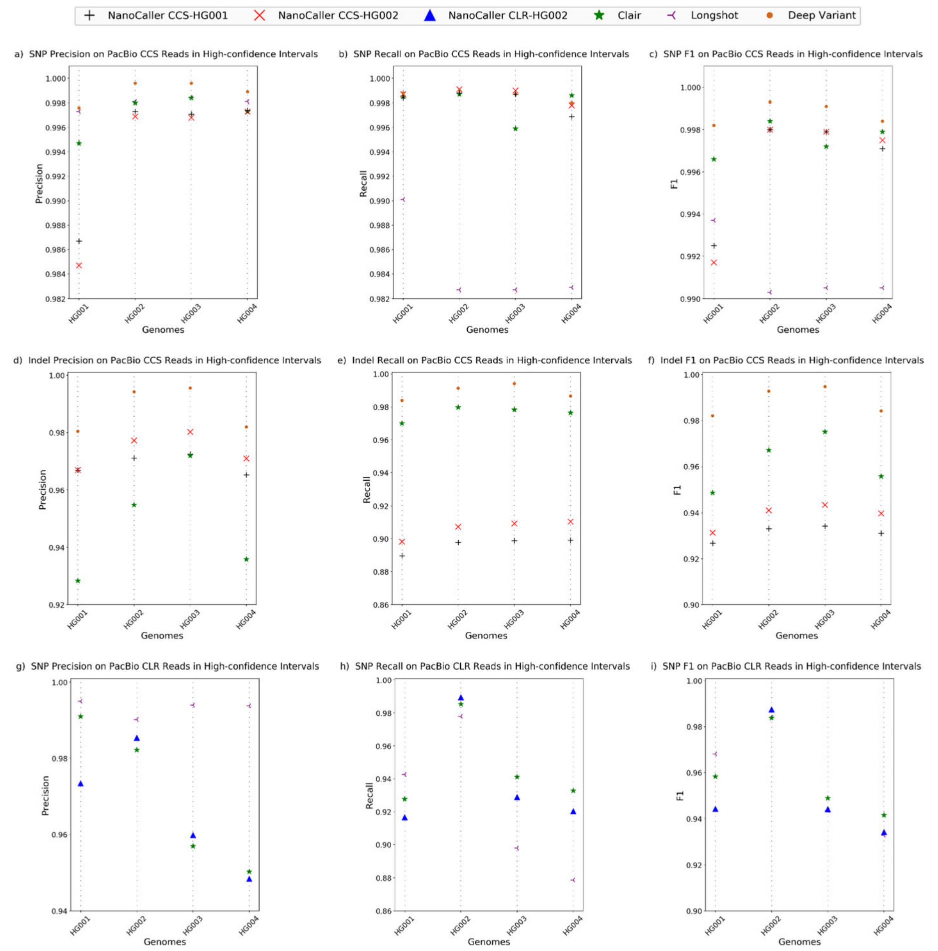


Fig. 4 Performances of NanoCaller and other variant callers on four PacBio CCS and four PacBio CLR datasets. SNP performance on whole-genome high-confidence intervals using CCS reads: **a** precision, **b** recall, **c** F1 score. Indel performance on whole-genome high-confidence intervals using CCS reads: **d** precision, **e** recall, **f** F1 score. SNP performance on whole-genome high-confidence intervals using CLR reads: **g** precision, **h** recall, **i** F1 score

We also evaluated NanoCaller PacBio indel models CCS-HG001 (trained on HG001 CCS reads using benchmark variants v3.3.2) and CCS-HG002 (trained on HG002 CCS reads using benchmark variants v4.2.1) and the results are shown in Table 1 and Fig. 4d–f along with Clair and DeepVariant indel performances. The F1 scores on the trio suggest that NanoCaller performs competitively against Clair. As expected, DeepVariant performs very well on CCS reads because CCS reads have much lower error rates. More details of the performance can be found in "Additional file 2: Table S38."

For PacBio Continuous Long Read Sequencing (CLR) datasets, we evaluated NanoCaller SNP model CLR-HG002 (trained on HG002 PacBio CLR reads using benchmark variants v4.2.1) on the following genomes: HG001 (reads aligned to GRCh37) and the Ashkenazim trio: HG002, HG003, and HG004 (as shown in Table 1 and Fig. 4g–i). Due to drastic differences in coverage of CLR datasets, we used a higher NanoCaller quality score cut-off for HG003 and HG004, compared to HG001 and HG002. NanoCaller performs competitively against Longshot and Clair (F1 scores are CLR-HG002: 98.75%, 94.41%, 93.41% vs Clair: 98.38%, 94.89%, 94.15% and Longshot: 98.41%, 94.35%, 93.27%). More details of the performance can be found in "Additional file 2: Table S39."

Novel variants called by NanoCaller

We also analyzed SNP calls made by NanoCaller on HG002 (ONT reads basecalled by Guppy 2.3.4) that are absent in the GIAB ground truth calls (version 3.3.2) [31] and validated 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 "Additional file 3" zip file). By deciphering Sanger sequencing results, we identified 41 novel variants (25 SNPs, 10 insertions, and 6 deletions), as shown in Table 3. Based on the 41 novel variants, we conducted the variant calling evaluation 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 found that (1) on the newly released ONT HG002 reads, Medaka correctly identified 15 SNPs, 6 insertions and 2 deletions, Clair identified 14 SNPs, 6 insertions and 2 deletions, and Longshot correctly identified 18 SNPs, while NanoCaller was able to correctly identify 20 SNPs, 6 insertions and 2 deletions, as shown in the "Additional file 1: Table S12," whereas one of these 2 deletions was not called correctly by other variant callers; and (2) on the older ONT HG002 reads, as shown in Table 3, Medaka correctly identified 8 SNPs, 3 insertions and 1 deletion, and Clair identified 8 SNPs, 2 insertions and 1 deletion, whereas Longshot correctly identified 8 SNPs. In contrast, NanoCaller was able to correctly identify 18 SNPs and 2 insertions, whereas 10 of these 18 SNPs and 1 of these 2 insertions were not called correctly by other variant callers on the older HG002 (ONT reads). This indicates that the improvements in per-base accuracy during basecalling significantly enhance 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 Fig. 5, 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 demonstrates the usefulness of NanoCaller for SNP calling.

To demonstrate the performance of NanoCaller for indel calling, we used Fig. 6 to illustrate those variants that can be detected by long-read variant callers but cannot be detected by short-read data. In Fig. 6, the validated deletion 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 base pairs 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 Fig. 6a. Sanger sequencing signal data, shown in Fig. 6b, confirms the presence of a heterozygous deletion at the same location which is causing frameshift between the signals from maternal and paternal chromosomes. This is an example to demonstrate how

Table 3 Novel variants in HG002 genome, missing in v3.3.2 benchmark variant, discovered by Sanger sequencing together with the prediction information by NanoCaller and other variant callers using ONT reads basecall with Guppy 2.3.4. NanoCaller model used was trained on ONT HG001 Guppy 2.3.8 basecalled reads

Chrom	Position	REF	ALT	Genotype	Nanocaller	Medaka	Clair	Longshot
chr1	78883942	C	T	1|0	Correct	–	–	–
chr1	78883952	ATATATATTTATCCTTTATATATATATTCTT	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	ACAGAGGGGGACCTGGAGGGGGCAGAGGAGAGACCTG	A	0|1	–	–	–	–

Table 3 Novel variants in HG002 genome, missing in v3.3.2 benchmark variant, discovered by Sanger sequencing together with the prediction information by NanoCaller and other variant callers using ONT reads basecall with Guppy 2.3.4. NanoCaller model used was trained on ONT HG001 Guppy 2.3.8 basecalled reads *(Continued)*

Chrom	Position	REF	ALT	Genotype	Nanocaller	Medaka	Clair	Longshot
		TGGGG						
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	–

long-read variant callers on long-read data can detect variants that fail to be reliably called on short-read sequencing data.

NanoCaller runtime comparison

We assessed NanoCaller’s running time in four modes: “snps_unphased,” “snps,” “indels,” and “both.” In “snps_unphased” mode, NanoCaller uses deep neural network model to predict SNP calls only, whereas in the “snps” mode, NanoCaller SNP calling is followed by an additional step of phasing SNP calls by external haplotyping tools such as WhatsHap. In the “indels” mode, NanoCaller uses phased reads in a BAM input to predict indels only. The entire NanoCaller workflow is the “both” mode, where NanoCaller first runs in “snps” mode to predict phased SNP calls, then uses WhatsHap to phase reads with the SNP calls from the “snps” mode, followed by running “indels” mode on phased reads from the previous step. Table 4 shows the wall-clock runtime of each mode of NanoCaller using 16 CPUs (IntelXeon CPU E5-2683 v4 @ 2.10 GHz) on 49× HG002 ONT (reads basecalled by Guppy 3.6), 35× PacBio CCS (15kb library size), and 58× PacBio CLR reads. NanoCaller takes ~ 18.4 h and ~ 2.8 h to run “both” and “snps_unphased” modes on 49× HG002 ONT reads, compared to ~ 181.6 h for Medaka and ~ 5.6 h for Clair, on the same 16 CPUs. On 35× CCS reads, NanoCaller takes ~

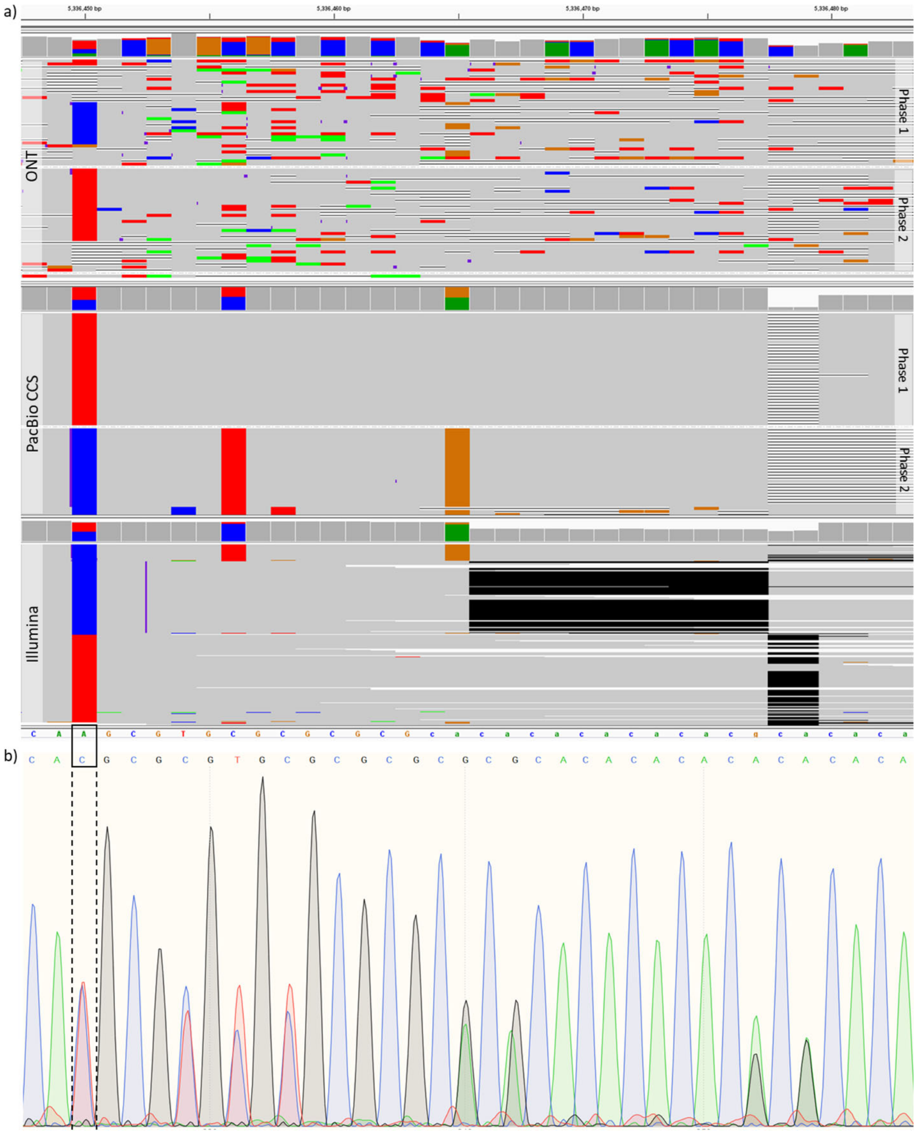


Fig. 5 Evidence for a novel multiallelic SNP. **a** IGV plots of Nanopore, PacBio CCS, and Illumina reads of HG002 genome at chr3:5336450-5336480. **b** Sanger sequencing signal data for the same region. NanoCaller on older HG002 data correctly identified the multiallelic SNP at chr3:5336450 (A>T,C) shown in black box

11.2 h and ~ 2.7 h to run "both" and "snps_unphased" modes, compared to ~ 1.8 h by Clair and ~ 11.8 by DeepVariant, on 16CPUs. NanoCaller usually runs faster than other tools. We summarize the runtime of all variant callers in "Additional file 1: Table S17."

Please note that Medaka's first step also produces unphased SNP calls using a recurrent neural network on mixed haplotypes (Medaka later uses WhatsHap to phase SNP calls and reads for haplotype separated variant calling). Compared with Medaka's first step, NanoCaller's unphased SNP calling not only takes a fraction of the time required for Medaka's first step (~ 2.8 h vs ~ 70.7 h), but also gives much better performance (precision, recall and F1 score on ONT HG002 (reads basecalled by Guppy 3.6) are NanoCaller: 98%, 97.99%, 97.99% vs Medaka 98.01%,

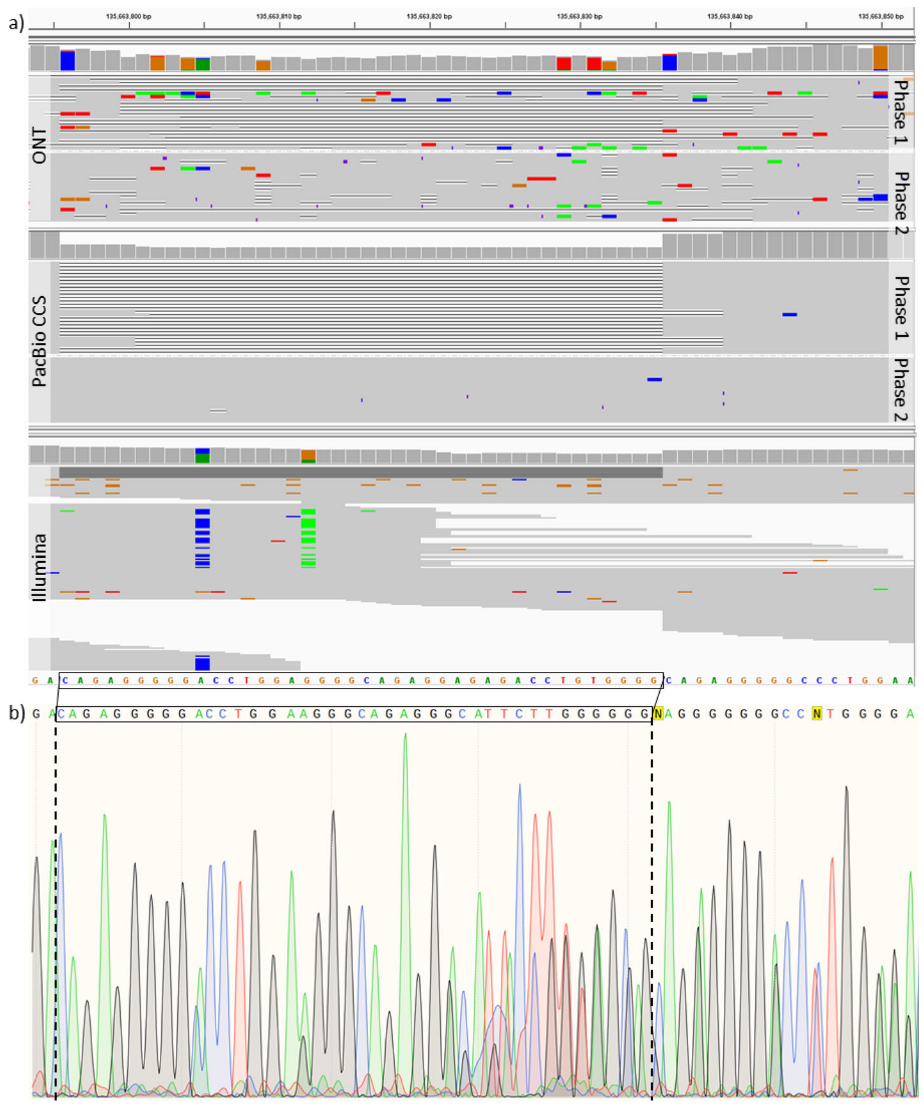


Fig. 6 Evidence for novel deletions. **a** IGV plots of Nanopore, PacBio CCS, and Illumina reads of HG002 genome at chr9:135663780-135663850. The 40-bp-long deletion shown below in black box was identified using Sanger sequencing at chr9:135663795 or chr9:135663804 (both are correct and the difference is due to two different alignments). **b** Sanger sequencing signal data around the deletion

Table 4 Wall-clock runtime in hours of different modes of NanoCaller using 16 CPUs on 49× ONT, 35× CCS, and 58× CLR reads of HG002

NanoCaller mode	snps_unphased	snps	Indels	Both
Description	Only SNP calling by NanoCaller deep convolutional network model	"snps_unphased" mode followed by WhatsHap SNP call phasing	Only indel calling by NanoCaller using phased BAM input	"snps" mode, followed by read phasing by WhatsHap, and "indels" mode
ONT	3.8	5.0	12.6	18.4
CCS	2.7	3.2	7.6	11.2
CLR	3.6	4.8	–	–

92.16%, 94.99%). Similarly, Longshot's first step uses a pair-HMM model to produce SNP calls from mixed haplotypes (Longshot later uses HapCUT2 to update the genotypes of these SNP calls in an iterative manner); on HG002 ONT (reads basecalled by Guppy 3.6), Longshot's first step takes 15.2 h with 93.01%, 95.69%, and 94.33% for precision, recall, and F1 score respectively. Longshot and WhatsHap cannot use multiple CPUs to produce on SNP calls. With single CPU on 49× HG002 ONT reads, Longshot needs ~ 49.7 h and WhatsHap needs ~ 84.3 h for SNP calling.

Effects of various parameters on NanoCaller's performance

In this section, we discuss the effects of tuning various parameters on NanoCaller's performance. The NanoCaller SNP models presented here are as follows: NanoCaller1 (trained on ONT HG001 reads basecalled with Guppy 2.3.8 using benchmark variants v3.3.2), NanoCaller2 (trained on ONT HG002 reads basecalled with Guppy 2.3.4 using benchmark variants v3.3.2), and NanoCaller3 (trained on HG003 PacBio CLR reads using benchmark variants v3.3.2). For testing, we used ONT reads for HG002-4 basecalled with Guppy3.6, CCS reads for HG002-4 (15 kb library size), and HG002-4 CLR reads.

Strategies of choosing heterozygous SNPs for SNP feature generation

In NanoCaller, we generated input features for a SNP candidate site by choosing potentially heterozygous SNP sites that share a read with the candidate site. In the implementation of NanoCaller, at most 20 heterozygous SNP candidates are chosen for downstream and upstream of a candidate site of interest. With an expectation that 1 SNP occurring per 1000 bp, a simple way is to include 20 kb downstream and upstream sequence centered at the candidate site of interest, and then select 20 nearest heterozygous SNP sites. But in some smaller genomic regions, a cluster of heterozygous SNP candidates may be found due to the noise with many false positives, and these false SNPs in a very smaller region would provide a strong co-occurrence evidence for each other but little information for the candidate site of interest. Longshot notices this issue and overcomes high false positive rates due to dense clusters of false positive SNP by simply removing these dense clusters if the number of SNP calls exceeds a certain threshold in a specified range; however, applying hard limits like that can lead to missing out on true SNPs that do occur in dense clusters in certain genomic regions.

We decided to use a different method for selecting nearby potential heterozygous sites by forcing NanoCaller to pick a certain number of sites that were some distance away from the candidate site. More precisely, we force NanoCaller to pick 2, 3, 4, 5, and 6 heterozygous SNP sites between the following distances from the candidate site: 2kbp, 5kbp, 10kbp, 20kbp, and 50kbp. This is illustrated in "Additional file 1: Fig S3 and Table S18." We found that using this method, we achieved better SNP calling performance for ONT reads; "Additional file 1: Table S21" shows that under this strategy, for each genome in the Ashkenazim trio, we achieved higher precision, recall, and F1 score for whole-genome analysis as well as in each difficult-to-map genomic regions. On the other hand, SNP calling performance of PacBio CCS and CLR reads is not affected by using this method of selecting heterozygous SNP sites, as shown in "Additional file 1: Table S22". This might be due to the fact that ONT reads have significantly higher N50 and mean read length compared to PacBio CCS and CLR reads, as shown in “Additional file 1: Table S1”. “Additional file 1: Fig S2” shows the read length distribution of HG004 ONT, CCS, and CLR datasets of 88×, 35×, and 27× coverages respectively. In these datasets, 99.4% of CCS reads and 97% of CLR reads are shorter than 20,000 bp; on the other hand, only 69.3% and 87.4% of the ONT reads are shorter than 20,000 bp and 50,000 bp. This simple comparison clearly demonstrates that NanoCaller is able to utilize the longer reads to improve SNP calling, and comparatively shorter PacBio reads might in part contribute to the less improvement of NanoCaller. Thus, as the read length increases, we expect NanoCaller can have better performance.

Different thresholds for heterozygous SNP sites for SNP feature generation

In order to generate haplotype structure features from long reads for a SNP candidate site, we need to select potentially heterozygous sites. Ideally, heterozygous sites should have approximately 0.5 alternative allele frequency, which is rarely the case due to alignment and sequencing errors. Therefore, a SNP candidate site is determined to potentially heterozygous if its alternative allele frequency is in a small range centered at 0.5: typically, this range is 0.4–0.6 or 0.3–0.7 depending upon the sequencing technology and is called neighbor threshold. “Additional file 1: Table S19” shows how the choice of this threshold affects SNP calling performance for ONT, PacBio CCS, and CLR reads which have different characteristics of error rates and read lengths (which in turn determines the number of candidate sites). We can observe that, generally, using a narrower range around 0.5 allows higher precision, but recall decreases because not enough heterozygous sites are chosen to give informative features. In particular, the performance is very sensitive to increases in the upper limit of threshold and decreases drastically when the threshold is increased. We determined that 0.4–0.6 threshold works best for ONT reads, with 0.3–0.7 and 0.3–0.6 being the best thresholds for CCS and CLR reads. Using a narrower threshold for ONT reads makes sense since longer ONT reads give us plenty of heterozygous sites to choose from, compared to CLR or CCS reads. It should be noted that this threshold is used for testing a sequencing data only, and during training of NanoCaller SNP models, we simply use benchmark heterozygous SNPs as highly likely heterozygous sites for feature generation.

We checked how the threshold for minimum number of neighboring likely heterozygous SNP sites for NanoCaller affects the performance and show the result in “Additional file 1: Table S20”. In NanoCaller, SNP candidates with less than a minimum number of such likely heterozygous SNP sites will be considered as false negatives without prediction. By default, the threshold for minimum number of likely heterozygous SNP sites is 1. In “Additional file 1: Table S20,” different thresholds are checked on Nanopore reads and Pacbio reads. On both data, as this threshold increases, precision increases and recall decreases. The increasing precision suggests that more heterozygous SNP candidates can benefit SNP prediction.

Using WhatsHap “distrust genotype” option for phasing

WhatsHap is able to call SNPs on ONT and PacBio reads, as shown in “Additional file 1: Table S26”. WhatsHap shows similar performance to NanoCaller, albeit much slower, while for ONT reads, WhatsHap shows poor performance with F1 scores around 88–93%. In NanoCaller, WhatsHap is used for phasing SNPs and reads but not for variant calling. Further, WhatsHap allows “distrust genotype” setting for phasing which allows WhatsHap to change genotypes of any SNP, from hetero- to homozygous and vice versa, in an optimal likelihood solution based upon the haplotypes created. In NanoCaller, this setting is disabled by default.

If users of NanoCaller want to use “distrust genotype” setting in WhatsHap, a negligible effect on SNP calling performance was found for ONT reads, while an increase in F1 score of 0.15–0.5% and 0.17–1.4% was found for PacBio CCS and CLR reads, as shown in “Additional file 1: Table S23”. Nevertheless, one should note that using this setting will significantly increase the runtime.

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 solely 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 indel candidate sites 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 long-range 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 > 20 kb, 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 performance evaluation. We also tested NanoCaller models calling variants on PacBio long-read data and achieved similar prediction trained on GRCh38 for GRCh37 and achieved 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 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 thousand folds of coverage. (5) With very accurate HiFi reads (< 1% error rate) generated by PacBio, NanoCaller is able to yield competitive variant calling performance. (6) NanoCaller has flexible design to call multiallelic 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 multiallelic variant calls, where all alternative alleles differ from the reference allele.

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. For instance, most variants missed by NanoCaller in MHC region cannot be observed through IGV either due to alignment errors. 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 [32, 33]. 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. Please also note that although NanoCaller might be able to call somatic multiallelic variants in tumor samples with clonal heterogeneity and variable tumor content, NanoCaller is currently designed to call diploid alleles. However, the frequency of some somatic variants in tumor samples might be too low to be distinguished from noises in long reads. Therefore, the variant calling on tumor samples needs a careful design and parameter tuning if NanoCaller is used. Additionally, better performance could be achieved for a specific training model when more benchmarking tumor data sets are available.

Conclusions

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

Long-read data sets for eight human genomes are used for the evaluation of NanoCaller: HG001, the Ashkenazim trio (consisting of son HG002, father HG003 and mother HG004), the Chinese trio (consisting of son HG005, father HG006 and mother HG007), and HX1. For HG001-7, Oxford Nanopore Technology (ONT) FASTQ files basecalled with Guppy 4.2.2 were downloaded from Human Pangenome Reference Consortium Database. HX1 genome was sequenced by us using PacBio [10] and Nanopore sequencing [34], and Nanopore reads for HX1 were re-basecalled using Guppy 4.5.2 by us. All ONT datasets were aligned to GRCh38 using minimap2 [35]. PacBio CCS alignment files for HG002 and HG003, and FASTQ files for HG001 and HG004 were downloaded from the GIAB database [31, 36]; FASTQ files were aligned to GRCh38 reference genome using minimap2. These CCS datasets were prepared with 15 k and 20 kb library size selection, and therefore have longer reads than CCS reads used for precisionFDA challenge. PacBio CLR alignment reads for HG001-4 were downloaded from the GIAB database [31, 36]. "Additional file 2: Table S29" shows the statistics of mapped reads in the eight genomes where the coverage of ONT data ranges from 34 to 84 and the coverage of PacBio data is between 27 and 58.

Benchmark variant calls

The benchmark set of SNPs and indels for HG001 (v3.3.2), the Ashkenazim trio HG002-4 (v4.2.1 and v3.3.2), and the Chinese trio HG005-7 (v3.3.2) are download from the Genome in a Bottle (GIAB) Consortium together with high-confidence regions for each genome. There are 3,002,314; 3,459,843; 3,430,611; 3,454,689; 2,945,666; 3,030,507; 3,048,404 and 3,489,068 SNPs for HG001, HG002, HG003, HG004, HG005, HG006, and HG007 respectively, and 517,177; 5 87,987; 569,180; 576,301; 432,747; 435,520; 437,866 and 697,736 indels for them, as shown in Table 5. Benchmark variant

Table 5 Statistics of benchmark variants in chromosomes 1–22 of each genome aligned to the GRCh38 reference genome. Four genomes with GIAB benchmark variant calls, with v3.3.2 for HG001 and HG005-7, and v4.2.1 for HG002-4, together with the statistics within the high-confidence regions. For HX1, high-confidence regions are created by removing GIAB "all difficult-to-map" 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	3,002,314	517,177	2,960,486	483,941	2,330,204,759	81.05	181,036
HG002	3,459,843	587,987	3,365,115	525,466	2,542,724,465	88.44	210,352
HG003	3,430,611	569,180	3,327,480	504,497	2,529,085,210	87.97	199,302
HG004	3,454,689	576,301	3,346,597	510,516	2,525,035,837	87.83	200,556
HG005	2,945,666	432,747	2,904,403	403,859	2,290,538,775	79.67	172,678
HG006	3,030,507	435,520	2,982,278	405,828	2,348,035,455	81.67	158,063
HG007	3,048,404	437,866	3,000,039	407,892	2,345,850,549	81.59	157,966
HX1	3,489,068	697,736	2,788,450	176,587	2,182,959,159	75.93	–

calls for HX1 were generated by using GATK on Illumina ~300× reads sequenced by us [10].

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 another for indel prediction, with each requiring a different type of input. Input features or pileup images generated for SNP candidate sites use only long-range haplotype information. Aligned reads are phased with WhatsHap using SNP calls from NanoCaller, and then NanoCaller uses phased reads to generate input features or pileup images for indel candidate sites by carrying out local multiple sequence alignment around each site. 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 phasing of SNP calls.

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" [37] 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, calculate the alternative allele frequency defined as:

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

b is considered to be a SNP candidate site if the total read depth and the alternative allele frequency are both greater than specified thresholds. We set the default 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 Fig. 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 50,000 bp away from b. For SNP calling on PacBio datasets, we set this limit at 20,000 bp.

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 Fig. 1a. More details for how to choose these 40 nearby heterozygous sites from the set V can be found at "Additional file 1: Tables S18-S22 and Fig S2-S3".
3. The set of reads covering *b* is divided into four groups, R_B = {reads that support base B at b}, B ∈ {A, G, T, 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 (C_{BD}^t) of supporting reads for site $t \in$ Z with base D ∈ {A, G, T, C}.
5. Let $F_{BD}^t = C_{BD}^t * g(D)$, where 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 Fig. 1c.
6. We obtain a 4 × 41 × 4 matrix M with entries $[F_{BD}^t]_{B,t,D}$ (as shown Fig. 1d) 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, various base positions as columns, 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 4 × 41 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 Fig. 1d). 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 Fig. 1e.

Deep learning prediction In NanoCaller, we present a convolutional neural network [38] for SNP prediction, as shown in "Additional file 1: Fig S6". 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 2 × 3. 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 the first network with the 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 the 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 variant call 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 allele 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**.**

Phasing of SNP calls After SNP calling, NanoCaller phases predicted SNP calls using WhatsHap [39]. By default, NanoCaller disables "distrust-genotypes" and "include-homozygous" settings of WhatsHap for phasing SNP calls, which would otherwise allow WhatsHap to switch variants from hetero- to homozygous and vice versa in an optimal phasing solution. Enabling these WhatsHap settings has minimal impact on NanoCaller's SNP calling performance (as shown in the "Additional file 1: Tables S2, S3 and S4"), but increases the time required for phasing by 50–80%, depending upon which NanoCaller mode is being run. 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 SNPs calls that are predicted by NanoCaller and phased by WhatsHap [39] (as described above).

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

1. For i $\in$ {0, 1}, calculate:

a. depth_i = total number of reads in phase i at site b
b. Insertion frequency $= \min\left\{\frac{\text{number of reads in phase } i \text{ with insertions in a window of size N at site } b}{\text{depth}_i}\right\}_{i\in\{0,1\}}$
c. Deletion frequency $= \min\left\{\frac{\text{number of reads in phase } i \text{ with deletions in a window of size N at site } b}{\text{depth}_i}\right\}_{i\in\{0,1\}}$

2. b is considered to be 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 the coverage and basecalling error rate of the genome sequencing data. We slide two windows along the reference to calculate what fraction of reads in that window contain an insertion or deletion. The first window uses a larger window size (default is 10) to calculate how many reads contain an insertion or deletion longer than or equal to 3 bp, whereas the second window uses a smaller window size (default is 4) to calculate how many reads contain an insertion or deletion shorter than or equal to 10 bp. The reason for allowing only indels longer than or equal to 3 bp in the large window is to circumvent sequencing errors that give rise to several small 2 bp indels from producing falsely high indel frequency.

Pileup image generation Input image of indel candidate site is generated using the procedure below as shown in Fig. 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 160 bp 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 160-bp-long subsequence of the read starting at the site *b* (for PacBio datasets, we use reference sequence and alignment sequences of length 260 bp).

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 Fig. 2a.

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 from the end.

d) For $B \in \{A, G, T, C, -\}$ and $1 \leq p \leq 128$, calculate

$$C_{B,p} = \sum\nolimits_{r \in S} g(seq'_{r'} B, p)$$
$$M_{B,p} = \frac{C_{B,p}}{\#reads\,in\,set\,S}$$

where $g(seq'_r, B, p)$ returns 1 if the base at index p of seq'_r is B and 0 otherwise. Figure 2c 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 Fig. 2d).

f) Construct a 5×128 matrix Q, with entries $[Q_B^p]_{B,p}$, where $Q_B^p = 1$ if seq'_{ref} has symbol B at index p and 0 otherwise as shown in Fig. 2f). 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 Fig. 2e 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 network [38] for indel calling, as shown in Fig. 2. 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 global alignment algorithm with affine gap penalty. Alternative allele is inferred from the alignmnet gap in 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. Please note that NanoCaller does not filter any indel predictions based on predicted indel length, but 50 bp can be the threshold of predicted indel definition since the majority of predicted indels are < 50 bp.

Training and testing

For SNP calling, we have trained five convolutional neural network models on two different genomes that users can choose from the following: ONT-HG001 (trained on HG001 ONT reads basecalled with Guppy 4.2.2), ONT-HG002 (trained on HG002 ONT reads basecalled with Guppy 4.2.2), CCS-HG001 (trained on HG001 CCS reads with 11-20 kb library size), CCS-HG002 (trained on HG002 CCS reads with 15-20 kb library size), and CLR-HG002 (trained on HG002 PacBio CLR dataset). The first four datasets have both SNP and indel models, whereas CLR-HG002 only has SNP model. All training sequencing datasets were aligned to GRCh38, with only chromosomes 1–22 used for training. We used v3.3.2 GIAB's benchmark variants for training all HG001 models and v4.2.1 benchmark variants for all HG002 models. Please refer to "Additional files 1 & 2" for the performance of more models, such as NanoCaller models trained on ONT reads of HG001 and HG002 basecalled by older Guppy versions, models trained using v3.3.2 benchmark variants for HG002, and models trained on chromosomes 1–21 of Nanopore R10.3 flowcell reads and Bonito basecalled dataset of HG002.

In NanoCaller, the SNP and indel models have 743,790 parameters in total, a significantly lower number than Clair [24] (2,377,818) and Clairvoyante [23] (1,631,496). All parameters in NanoCaller are initiated by Xavier's method [40]. 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−5 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 Fig. 1 (b) by

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

Performance measurement

The performance of SNP/indel calling by a variant caller is evaluated 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. *F*1 is the weighted average of *p* and *r*, a harmonic measurement of precision and recall. The range of the three measurements is [0, 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 “Additional file 3” zip file. Please note that more than 17 variant sites are detected in the Sanger results, because each PCR region can contain 1+ variants.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-021-02472-2.

Additional file 1. : Fig S1-S7 and Tables S1-S26 for various detailed performance evaluation on older Pacbio and Nanopore (basecalled by Guppy 2.3) on GIAB benchmark variants v3.3.2, the definitions of difficult-to-map regions, evaluation with different quality score thresholds, runtime comparisons, discussion of generation of heterozygous sites, long-read statistics, and commands to reproduce performance.

Additional file 2. : Tables S27-S42 for various detailed performance evaluation on more recent Pacbio and Nanopore (basecalled by Guppy v3.3.2) long-read data on GIAB benchmark variants v4.2.1 together with data statistics.

Additional file 3. : The Sanger sequencing results for 17 genomic regions.

Additional file 4. : Review history.

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. Finally, we would like to thank two anonymous reviewers in offering insightful comments and suggestions on improving NanoCaller. This study is in part supported by NIH/NIGMS grant GM132713 to KW.

Review history

The review history is available as additional file 4.

Peer review information

Andrew Cosgrove was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contributions

MUA and QL developed the computational method and drafted the manuscript. MUA implemented the software tool and evaluated its performance. LF conducted wet-lab experiments of 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.

Funding
This study is supported by NIH/NIGMS grant GM132713 and the CHOP Research Institute.

Availability of data and materials

Code availability

NanoCaller is publicly available at https://github.com/WGLab/NanoCaller under MIT license and will be regularly maintained and updated. A detailed description of installing/running NanoCaller and reproducible pipelines/datasets have also been documented in the GitHub repository. Source code used in the manuscript is available via Zenodo with DOI 10.5281/zenodo.5176764 [28].

Genomic datasets

ONT long reads with different versions of basecallers: Human Pangenomics Reference Consortium ONT reads basecalled with Guppy 4.2.2 for HG001 [41], HG002-4 [42], and HG005-7 [41]. precisionFDA truth challenge V2 ONT reads basecalled with Guppy3.6 HG002-4 [30]. HG001 ONT Rel 6 reads [9]. HG002 official ONT release [43], ultra-long GIAB Guppy 2.3.4 ONT reads [44].

PacBio CCS/CLR reads with different library sizes: precisionFDA truth challenge V2 for Pacbio CCS 15 kb library size reads for HG002-4 [30]. HG001 PacBio CCS reads with 11 kb library size [45], HG001-4 PacBio CCS reads with 15–20 kb library size selection [31]. PacBio CLR reads for HG001 [46], HG002 [47], HG003 [48], and HG004 [49].

Benchmark datasets

GIAB benchmark variants v3.3.2 [4] and v4.2.1 [50], and genome stratification BED files [3].

Declarations

Consent to participate

Not applicable.

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 21 November 2020 Accepted: 19 August 2021
Published online: 06 September 2021

References

1. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. 2010;20(9):1297–303. https://doi.org/10.1101/gr.107524.110.
2. Garrison E, Marth G. 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, et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol. 2019;37(5):555–60. https://doi.org/10.1038/s41587-019-0054-x.
4. Zook JM, McDaniel J, Olson ND, Wagner J, Parikh H, Heaton H, et al. An open resource for accurately benchmarking small variant and reference calls. Nat Biotechnol. 2019;37(5):561–6. https://doi.org/10.1038/s41587-019-0074-6.

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. https://doi.org/10.1038/s41467-019-11146-4.
6. Branton D, Deamer DW, Marziali A, Bayley H, Benner SA, Butler T, et al. The potential and challenges of nanopore sequencing. Nat Biotechnol. 2008;26(10):1146–53. https://doi.org/10.1038/nbt.1495.
7. Eid J, Fehr A, Gray J, Luong K, Lyle J, Otto G, et al. Real-time DNA sequencing from single polymerase molecules. Science. 2009;323(5910):133–8. https://doi.org/10.1126/science.1162986.
8. Mantere T, Kersten S, Hoischen A. Long-read sequencing emerging in medical genetics. Front Genet. 2019;10:426. https://doi.org/10.3389/fgene.2019.00426.
9. Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. Nat Biotechnol. 2018;36(4):338–45. https://doi.org/10.1038/nbt.4060.
10. Shi L, Guo Y, Dong C, Huddleston J, Yang H, Han X, et al. Long-read sequencing and de novo assembly of a Chinese genome. Nat Commun. 2016;7(1):12065. https://doi.org/10.1038/ncomms12065.
11. Pendleton M, Sebra R, Pang AW, Ummat A, Franzen O, Rausch T, et al. Assembly and diploid architecture of an individual human genome via single-molecule technologies. Nat Methods. 2015;12(8):780–6. https://doi.org/10.1038/nmeth.3454.
12. Seo JS, Rhie A, Kim J, Lee S, Sohn MH, Kim CU, et al. De novo assembly and phasing of a Korean human genome. Nature. 2016;538(7624):243–7. https://doi.org/10.1038/nature20098.
13. Cho YS, Kim H, Kim HM, Jho S, Jun J, Lee YJ, et al. An ethnically relevant consensus Korean reference genome is a step towards personal reference genomes. Nat Commun. 2016;7(1):13637. https://doi.org/10.1038/ncomms13637.
14. Stephens Z, Wang C, Iyer RK, Kocher JP. Detection and visualization of complex structural variants from long reads. BMC Bioinformatics. 2018;19(S20):508. https://doi.org/10.1186/s12859-018-2539-x.
15. Heller D, Vingron M. SVIM: structural variant identification using mapped long reads. Bioinformatics. 2019;35(17):2907–15. https://doi.org/10.1093/bioinformatics/btz041.
16. Jiang T, Liu Y, Jiang Y, Li J, Gao Y, Cui Z, Liu Y, Liu B, Wang Y. Long-read-based human genomic structural variation detection with cuteSV. Genome Biol. 2020;21:189. https://doi.org/10.1186/s13059-020-02107-y.
17. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, et al. Accurate detection of complex structural variations using single-molecule sequencing. Nat Methods. 2018;15(6):461–8. https://doi.org/10.1038/s41592-018-0001-7.
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. https://doi.org/10.1186/s12859-018-2207-1.
19. Gong L, Wong CH, Cheng WC, Tjong H, Menghi F, Ngan CY, et al. Picky comprehensively detects high-resolution structural variants in nanopore long reads. Nat Methods. 2018;15(6):455–60. https://doi.org/10.1038/s41592-018-0002-6.
20. Ameur A, Kloosterman WP, Hestand MS. Single-molecule sequencing: towards clinical applications. Trends Biotechnol. 2019;37(1):72–85. https://doi.org/10.1016/j.tibtech.2018.07.013.
21. Wenger AM, Peluso P, Rowell WJ, Chang PC, Hall RJ, Concepcion GT, et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. Nat Biotechnol. 2019;37(10):1155–62. https://doi.org/10.1038/s41587-019-0217-9.
22. Poplin R, Chang PC, Alexander D, Schwartz S, Colthurst T, Ku A, et al. A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol. 2018;36(10):983–7. https://doi.org/10.1038/nbt.4235.
23. Luo R, Sedlazeck FJ, Lam TW, Schatz MC. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. Nat Commun. 2019;10(1):998. https://doi.org/10.1038/s41467-019-09025-z.
24. Luo R, Wong C-L, Wong Y-S, Tang C-I, Liu C-M, Leung C-M, et al. Exploring the limit of using a deep neural network on pileup data for germline variant calling. Nature Machine Intelligence. 2020;2(4):220–7. https://doi.org/10.1038/s42256-020-0167-4.
25. Edge P, Bansal V. Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing. Nat Commun. 2019;10(1):4660. https://doi.org/10.1038/s41467-019-12493-y.
26. Edge P, Bafna V, Bansal V. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Genome Res. 2017;27(5):801–12. https://doi.org/10.1101/gr.213462.116.
27. medaka. Sequence correction provided by ONT Research [https://github.com/nanoporetech/medaka]. Accessed 20 Oct 2019.
28. Ahsan MU, Liu Q, Fang L, Wang K. NanoCaller: Zenodo. https://doi.org/10.5281/zenodo.5176764; 2021.
29. 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. https://doi.org/10.1101/023754.
30. Olson ND, Wagner J, McDaniel J, Stephens SH, Westreich ST, Prasanna AG, Johanson E, Boja E, Maier EJ, Serang O, et al. precisionFDA Truth Challenge V2: Calling variants from short- and long-reads in difficult-to-map regions. bioRxiv. 2020; 380741. https://doi.org/10.1101/2020.11.13.380741.
31. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. Scientific Data. 2016;3(1):160025. https://doi.org/10.1038/sdata.2016.25.
32. Rang FJ, Kloosterman WP, de Ridder J. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. Genome Biol. 2018;19(1):90. https://doi.org/10.1186/s13059-018-1462-9.
33. Zascavage RR, Thorson K, Planz JV. Nanopore sequencing: An enrichment-free alternative to mitochondrial DNA sequencing. Electrophoresis. 2019;40(2):272–80. https://doi.org/10.1002/elps.201800083.
34. 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. https://doi.org/10.1038/s41467-019-10168-2.
35. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094–100. https://doi.org/10.1093/bioinformatics/bty191.
36. GENOME IN A BOTTLE [https://jimb.stanford.edu/giab]. Accessed 4 Apr 2021.
37. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25(16):2078–9. https://doi.org/10.1093/bioinformatics/btp352.
38. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84–90. https://doi.org/10.1145/3065386.

39. Patterson M, Marschall T, Pisanti N, van Iersel L, Stougie L, Klau GW, et al. WhatsHap: weighted haplotype assembly for future-generation sequencing reads. J Comput Biol. 2015;22(6):498–509. https://doi.org/10.1089/cmb.2014.0157.
40. 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. 2010;249–56.
41. Human-Pangenome-Reference-Consortium. https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=NHGRI_UCSC_panel; 2020. Accessed 26 Mar 2021.
42. Shafin K, Pesout T, Lorig-Roach R, Haukness M, Olsen HE, Bosworth C, et al. Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes. Nature Biotechnology. 2020;38(9):1044–53. https://doi.org/10.1038/s41587-020-0503-6.
43. Oxford-Nanopore-Technologies. HG002 September and November 2020 release. https://nanoporetech.github.io/ont-open-datasets/gm24385_2020.09/, https://nanoporetech.github.io/ont-open-datasets/gm24385_2020.11/; 2020. Accessed 30 Mar 2021.
44. GIAB. HG002 ultra-long ONT reads. https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/Ultralong_OxfordNanopore/guppy-V2.3.4_2019-06-26/; 2019. Accessed 4 Aug 2019.
45. GIAB: HG001 CCS 11 kb reads. https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/PacBio_SequelII_CCS_11kb/HG001_GRCh38/; 2019. Accessed 2 July 2020.
46. GIAB: HG001 PacBio CLR reads. https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NA12878_PacBio_MtSinai/; 2015. Accessed 3 Oct 2019.
47. GIAB. HG002 PacBio CLR reads. https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/PacBio_minimap2_bam/; 2018. Accessed 3 Oct 2019.
48. GIAB: HG003 PacBio CLR reads. https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG003_NA24149_father/PacBio_MtSinai_NIST/PacBio_minimap2_bam/; 2018. Accessed 5 Dec 2019.
49. GIAB. HG004 PacBio CLR reads. https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG004_NA24143_mother/PacBio_MtSinai_NIST/PacBio_minimap2_bam/; 2018. Accessed 5 Dec 2019.
50. Wagner J, Olson ND, Harris L, Khan Z, Farek J, Mahmoud M, Stankovic A, Kovacevic V, Wenger AM, Rowell WJ, et al. Benchmarking challenging small variants with linked and long reads. bioRxiv. 2020;212712. https://doi.org/10.1101/2020.07.24.212712.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.