Clairvoyante: a multi-task convolutional deep neural network for variant calling in Single Molecule Sequencing

Author

Ruibang Luo[1,2,*], Fritz J. Sedlazeck[3], Tak-Wah Lam[1], Michael C. Schatz[2]

[1] Department of Computer Science, The University of Hong Kong, Hong Kong
[2] Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA
[3] Human Genome Sequencing Center, Baylor College of Medicine, Houston, TX, USA

[*] Correspondence should be addressed to rbluo@cs.hku.hk

Abstract

The accurate identification of DNA sequence variants is an important, but challenging task in genomics. It is particularly difficult for single molecule sequencing, which has a per-nucleotide error rate of ~5%-15%. Meeting this demand, we developed Clairvoyante, a multi-task five-layer convolutional neural network model for predicting variant type (SNP or indel), zygosity, alternative allele and indel length from aligned reads. For the well-characterized NA12878 human sample, Clairvoyante achieved 99.73%, 97.68% and 95.36% precision on known variants, and 98.65%, 92.57%, 87.26% F1-score for whole-genome analysis, using Illumina, PacBio, and Oxford Nanopore data, respectively. Training on a second human sample shows Clairvoyante is sample agnostic and finds variants in less than two hours on a standard server. Furthermore, we identified 3,135 variants that are missed using Illumina but supported independently by both PacBio and Oxford Nanopore reads. Clairvoyante is available open-source (https://github.com/aquaskyline/Clairvoyante), with modules to train, utilize and visualize the model.

Introduction

A fundamental problem in genomics is to find the nucleotide differences in an individual genome relative to a reference sequence, i.e., variant calling. It is essential to accurately and efficiently call variants so that the genomic variants that underlie phenotypic differences and disease can be correctly detected[1]. Previous works have intensively studied the different data characteristics that might contribute to higher variant calling performance including the properties of the sequencing instrument[2], the quality of preceding sequence aligners[3] and the alignability of genome reference[4]. Today, these characteristics are carefully considered by state-of-the-art variant calling pipelines to optimize performance[5,6]. However, most of these analyses were done for short read sequencing, especially the Illumina technology, and require further study for other sequencing platforms.

Single Molecule Sequencing (SMS) technologies are emerging in recent years for a variety of important applications[7]. These technologies generate sequencing reads two orders of magnitude longer than standard short-read Illumina sequencing (10kbp to 100kbp instead of ~100bp), but they also contain 5%-15% sequencing errors rather than ~1% for Illumina. The two major SMS companies, Pacific Biosciences (PacBio) and Oxford Nanopore Technology (ONT) have greatly improved the performance of certain genomic applications, especially genome assembly and structural variant detection[7]. However, single nucleotide and small indel variant calling with SMS remain challenging because the traditional variant caller algorithms fail to handle such a high sequencing error rate, especially one enriched for indel errors.

Artificial Neural Networks (ANNs) are becoming increasingly prominent for a variety of classification and analysis tasks due to their advances in speed and applicability in many fields. One of the most important applications of ANNs has been image classification, with many successes including MNIST[8] or GoogLeNet[9]. The recent DeepVariant[10] package repurposed the Inception convolutional neural network for DNA variant detection by applying it to analyzing images of aligned reads around candidate variants. At each candidate site, the network computes the probabilities of three possible zygosities (homozygous reference, heterozygous reference, and homozygous alternative), allowing accurate determination of the presence or absence of a candidate variant. And then, DeepVariant uses a post-processing step to restore the other variant information including the exact alternative allele and variant type. As the authors pointed out originally in their manuscript, it might be sub-optimal to use an image classifier for variant calling, as valuable information that could contribute to higher accuracy are lost during the image transformation. In the latest version of DeepVariant, the code is built on top of the Tensorflow machine learning framework, allowing users to change the image input into any other formats by rewriting a small part of the code. However, whether it is reasonable or not to use a network (namely inception-v3) specifically designed for image-related tasks to call variants remains unclear.

In this study, we present Clairvoyante, a multi-task convolutional deep neural network specifically designed for variant calling with SMS reads. We explored different ways to enhance Clairvoyante's power to extract valuable genomic features from the frequent background errors present in SMS. Experiments calling variants in multiple human genomes both at known variant sites and genome-wide show that Clairvoyante is on-par with GATK UnifiedGenotyper on Illumina data, and substantially outperforms Nanopolish and DeepVariant on PacBio and ONT data on accuracy and speed.

Methods

In this section, we first introduce the DNA sequencing datasets of three different sequencing technologies: Illumina, PacBio, and ONT. We then formulate variant calling as a supervised machine learning problem. Finally, we present Clairvoyante for this problem and explain the essential deep learning techniques applied in Clairvoyante.

Datasets

While most of the variant calling in previous studies were done using a single computational algorithm on single sequencing technology, the Genome-in-a-Bottle (GIAB) dataset[11] first published in 2014 has been an enabler of our work. The dataset provides high-confidence SNPs and indels for a standard reference sample HG001 (also referred to as NA12878) by

integrating and arbitrating between 14 datasets from five sequencing and genotyping technologies, seven read mappers and three variant callers. For our study, we used as our truth dataset the latest dataset version 3.3.2 for HG001 (**Supplementary Material, Data Source, Truth Variants**) that comprises 3,042,789 SNPs, 241,176 insertions and 247,178 deletions for the GRCh38 reference genome, along with 3,209,315 SNPs, 225,097 insertions and 245,552 deletions for GRCh37. The dataset also provides a list of regions that cover 83.8% and 90.8% of the GRCh38 and the GRCh37 reference genome, where variants were confidently genotyped. The GIAB extensive project[12] published in 2016 further introduced four standard samples, including the Ashkenazim Jewish sample HG002 we have used in this work, containing 3,077,510 SNPs, 249,492 insertions and 256,137 deletions for GRCh37, 3,098,941 SNPs, 239,707 insertions and 257,019 deletions for GRCh37. 83.2% of the whole genome was marked as confident for both the GRCh38 and GRCh37.

Illumina Data

The Illumina data was produced by the National Institute of Standards and Technology (NIST) and Illumina[12]. Both the HG001 and HG002 datasets were generated on an Illumina HiSeq 2500 in Rapid Mode (v1) with 2x148bp paired-end reads. Both have an approximate 300x total coverage and were aligned to GRCh38 decoy version 1 using Novoalign version 3.02.07. In our study, we further down-sampled the two datasets to 50x to match the available data coverage of the other two SMS technologies (**Supplementary Material, Data Source, Illumina Data**).

Pacific Bioscience (PacBio) Data

The PacBio data was produced by NIST and Mt. Sinai School of Medicine[12]. The HG001 dataset has 44x coverage, and the HG002 has 69x. Both datasets comprise 90% P6-C4 and 10% P5-C3 sequencing chemistry and have a sequence N50 length between 10k-11kbp. Reads were extracted from the downloaded alignments and aligned again to GRCh37 decoy version 5 using NGMLR[13] version 0.2.3 (**Supplementary Material, Data Source, PacBio Data**).

Oxford Nanopore (ONT) Data

The Oxford Nanopore data were generated by the Nanopore WGS consortium[14]. Only data for sample HG001 are available to date, thus limiting the "cross sample variant calling evaluation" and "combined sampled training" on ONT data in the Result section. In our study, we used the 'rel3' release sequenced on the Oxford Nanopore MinION using 1D ligation kits (450bp/s) and R9.4 chemistry. The release comprises 39 flowcells and 91.2G bases, about 30x coverage. The reads were downloaded in raw fastq formatted and aligned to GRCh37 decoy version 5 using NGMLR[13] version 0.2.3 (**Supplementary Material, Data Source, Oxford Nanopore Data**).

Variant Calling as Multi-Task Regression and Classification

We model each variant with four categorical variables:

- $A \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ is the alternate base at a SNP, or the reference base otherwise
- $Z \in \{\mathrm{Homozygote}, \mathrm{Heterozygote}\}$ is the zygosity of the variant
- $T \in \{\mathrm{Reference}, \mathrm{SNP}, \mathrm{Insertion}, \mathrm{Deletion}\}$ is the variant type
- $L \in \{0, 1, 2, 3, 4, >4\}$ is the length of an INDEL, where '>4' represents a gap longer than 4bp

For the truth data, each variable can be represented by a vector (i.e. 1-D tensor) using the one-hot or probability encoding, as is typically done in deep learning: $a_b = \Pr\{A = b\}$, $z_i = \delta(i, Z)$, $t_j = \delta(j, T)$ and $l_k = \delta(k, L)$, where $\delta(p, q)$ equals 1 if $p = q$, or 0 otherwise. The four vectors (a, z, t, l) are the outputs of the network. a_b is set to all zero for an insertion or deletion. In the current Clairvoyante implementation, 1) multi-allelic SNPs are excluded from training, and 2) base-quality is not used (see "Discussion" below for a rationale).

With deep learning, we seek a function $F: x \rightarrow (a, z, t, l)$ that minimizes the cost C:

$$C = \frac{1}{N}\sum_{v}\left(\sum_{b=1}^{4}\left(\hat{a}_b^{(v)} - a_b^{(v)}\right)^2 - \sum_{i=1}^{2} z_i^{(v)} \log \hat{z}_i^{(v)} - \sum_{j=1}^{4} t_j^{(v)} \log \hat{t}_j^{(v)} - \sum_{k=1}^{6} l_k^{(v)} \log \hat{l}_k^{(v)}\right)$$

where v iterates through all variants and a variable with a caret indicates it is an estimate from the network. Variable x is the input of the network, and it can be of any shape and contain any information. Clairvoyante uses an x that contains a summarized "piled-up" representation of read alignments. The details will be discussed in the next section named "Clairvoyante".

In our study, good performance implies correct predictions could be made even when the evidence is marginal to distinguish a genuine variant from non-variant (reference) position. To achieve the goal, we paired each truth variant with two non-variants randomly sampled from the genome at all possible non-variant and non-ambiguous sites for model training. With about 3.5M truth variants from the GIAB dataset, about 7M non-variants are added as samples for model training.

We randomly partitioned all samples into 90% for training and 10% for validation. We intentionally did not hold out any sample of the data for testing as other projects commonly do because, in our study, we can use an entirely different dataset for testing samples. For example, we can use the samples of HG002 to test against a model trained on HG001, and vice versa.

Clairvoyante

Clairvoyante is a multi-task five-layer convolution neural network with the last two layers as feedforward layers (**Figure 1**). The multi-task neural network makes four groups of predictions on each input: 1) alternative alleles, 2) zygosity, 3) variant type, and 4) indel length. The predictions in groups 2, 3 and 4 are mutually exclusive while the predictions in group 1 are not. The alternative allele predictions are computed directly from the first fully connected layer (FC4), while the other three group of predictions are computed from the second fully connected layer (FC5). The indel length prediction group has six possible outputs indicating an indel with a length between 0-3bp or ≥4bp of any unbounded length. The prediction limit on indel length is configurable in Clairvoyante and can be raised when more training data on longer indels could be provided. The Clairvoyante network is succinct and fine-tuned for the variant calling purpose. It contains only 1,631,496 parameters, about 13-times fewer than DeepVariant[10] using the Inception-v3 network architecture, which was originally designed for general purpose image recognition. Additional details of Clairvoyante are introduced in the different subsections below.

For each input sample (truth or candidate variants), the overlapping sequencing read alignments are transformed into a multi-dimensional tensor x of shape 33 by 4 by 4. The first dimension '33' corresponds to the position. The second dimension '4' corresponds to the count of A, C, G, or T on the sequencing reads, the way of counting is subject to the third

dimension. The third dimension '4' corresponds to four different ways of counting. In the first dimension, we added 16 flanking base-pairs on both sides of a candidate (in total 33bp), which we have measured to be sufficient to manifest background noise while providing a good computational efficiency. In the second dimension, we separated any counts into four bases. In the third dimension, we used four different ways of counting, generating four tensors of shape 33 by 4. The first tensor encodes the reference sequence and the number of reads supporting the reference alleles. The second, third and fourth tensors use the relative count against the first tensor: the second tensor encodes the inserted sequences, the third tensor encodes the deleted base-pairs, and the fourth tensor encodes alternative alleles. For an exact description of how x is generated, please refer to the pseudo code in "**Supplementary Material, Pseudo code for generating the input**". **Figure 2** illustrates how the tensors can represent a SNP, an insertion, a deletion, and a non-variant (reference), respectively. The non-variant in **Figure 2** also depicts how the matrix will show background noise. A similar but simpler read alignment representation was proposed by Jason Chin[15] in mid-2017, the same time as we started developing Clairvoyante. Different from Chin's representation, ours decouples the substitution and insertion signal into separate arrays and allows us to precisely record the allele of inserted sequence.

Our study used the widely adopted TensorFlow[16] as its primary programming framework. Using the 44x coverage HG001 PacBio dataset as an example, a near optimal model can be trained in three hours using the latest desktop GPU model nVidia GTX 1080 Ti. Using a trained model, about two hours is needed to call variants genome-wide using a 2 x 14-core CPU-only server (without GPU), and it takes only a few minutes to call variants at known variant sites or in an exome (>5,000 candidate sites per second). Several techniques have been applied to minimize computational and memory consumption (see the **Computational Performance** subsection).

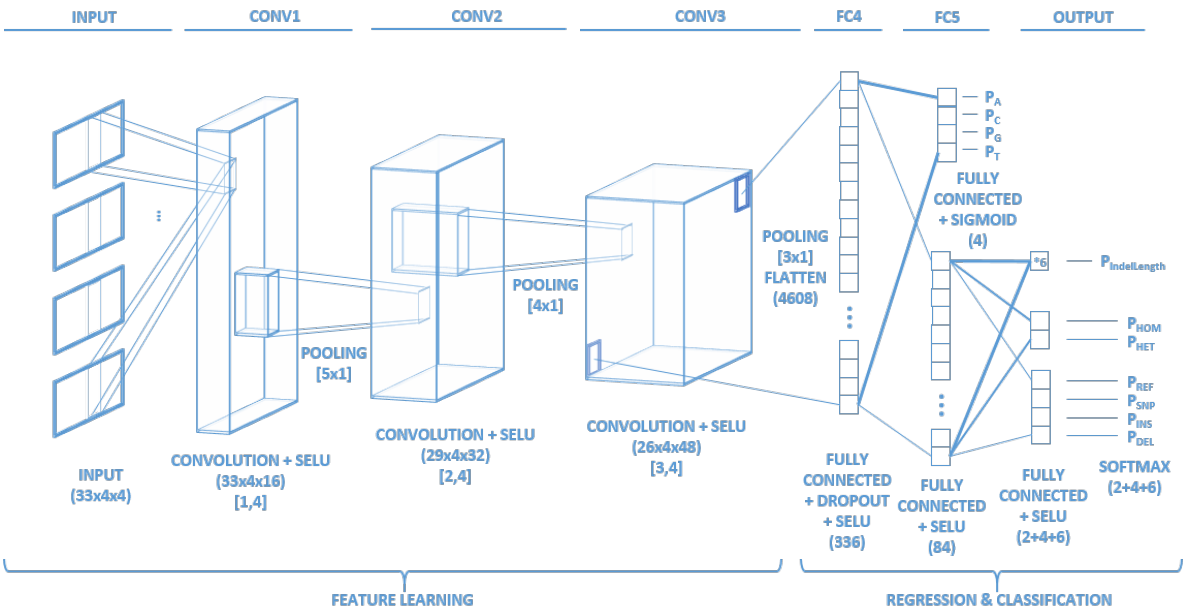

Figure 1. Clairvoyante network architecture and layer details. The descriptions under each layer includes 1) the layer's function; 2) the activation function used; 3) the dimension of the layer in parenthesis (Input layer: Height x Width x Arrays, Convolution layer: Height x Width x Filters, Fully connected layer: Nodes), and; 4) kernel size in brackets (Height x Width).

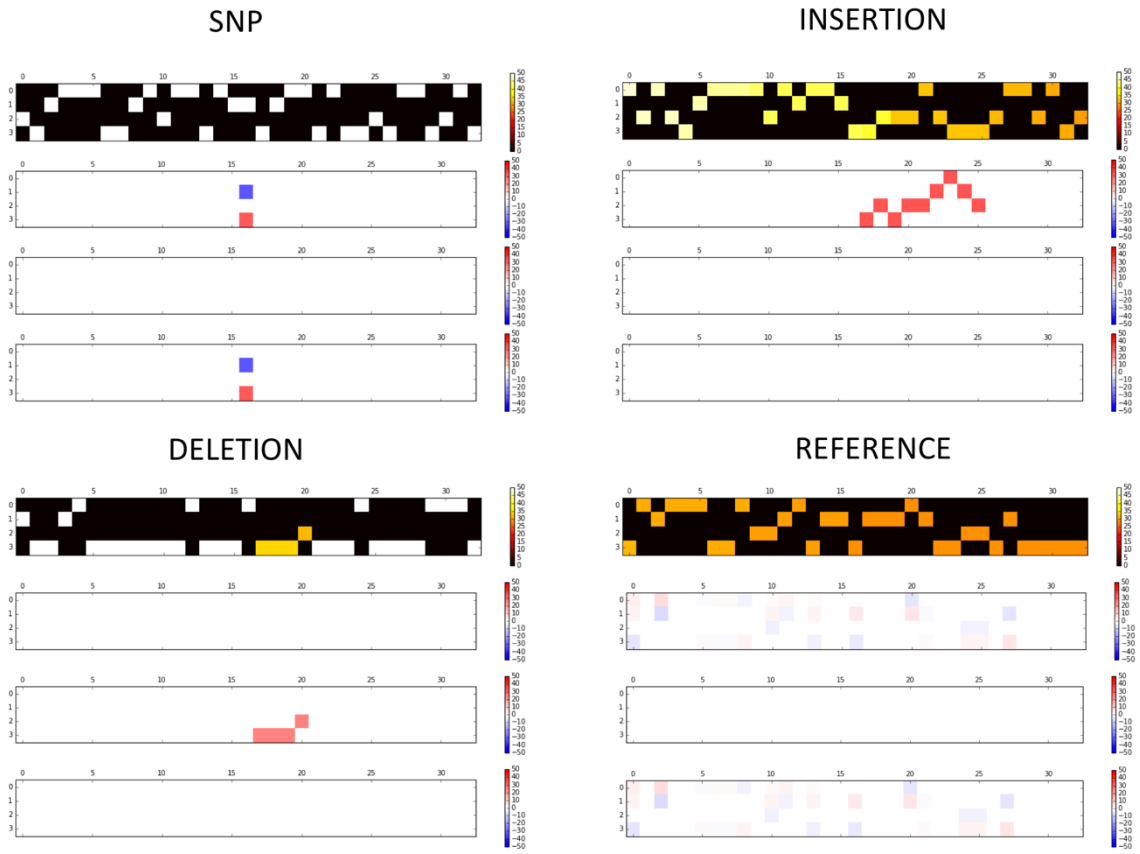

Figure 2. Selected illustrations of how Clairvoyante represents the three common types of small variant, and a non-variant. The figure includes: (**top left**) a C>G SNP, (**top right**) a 9bp insertion, (**bottom left**) a 4bp deletion, and (**bottom right**) a non-variant with reference allele. The color intensity represents the strength of a certain variant signal. The SNP, insertion and deletion examples are ideal with almost zero background noise. The non-variant example illustrates how the background noises look like when not mingled with any variant signal.

Model Initialization

Weight initialization is important to stabilize the variances of activation and back-propagated gradients at the beginning of model training. We used a He initializer[17] to initialize the weights of hidden layers in Clairvoyante, as the He initializer is optimized for training extremely deep models using rectified activation function directly from scratch. For each layer, the weight of each node is sampled from a univariate normal distribution with $\sigma = 1 \div \sqrt[2]{d_i \div 2}$, where d_i denote the number of in-degree of the node.

Activation Function

Batch normalization is a technique to ensure zero mean and unit variance in each hidden layer to avoid exploding or diminishing gradients during training. However, batch normalization has often been identified as a computational bottleneck in neural network training because computing the mean and the standard deviation of a layer is not only a dependent step, but also a reduction step that cannot be efficiently parallelized. To tackle this problem, we will use the new activation function called “Scaled Exponential Linear Units” (SELUs)[18], a variant of the rectified activation function. Different from a standard batch

normalization approach that adds an implicit layer for the named purpose after each hidden layer, SELUs utilizes the Banach fixed-point theorem to ensure convergence to zero mean and unit variance in each hidden layer without batch normalization.

Optimizer and Learning rate

We used an Adam optimizer with default settings[19] to update the weights by adaptive node-specific learning rates, whereas setting a global learning rate only functions as setting an upper limit to the learning rates. This behavior allows Clairvoyante to remain at a higher learning rate for a longer time to speed up the training process.

Although the Adam optimizer performs learning rate decay intrinsically, we found decreasing the global learning rate when the cost of the model in training plateaued can lead to a better model performance in our study. In Clairvoyante, we implemented two types of training modes. The fast training mode is an adaptive decay method that uses an initial learning rate at $1e^{-3}$, decreases the learning rate by a factor of 0.1 when the validation rate goes up and down for five rounds and stops after two times of decay. A second nonstop training mode allows users to decide when to stop and continue using a lower learning rate.

Dropout and L2 Regularization

Although more than three million labeled truth variants are available for training, the scarcity of some labels, especially variants with a long indel length, could fail the model training by overfitting to abundantly labeled data. To alleviate the class imbalance, we apply both dropout[20] and L2 regularization[21] techniques in our study. Dropout is a powerful regularization technique. During training, dropout randomly ignoring nodes in a layer with probability p, then sums up the activations of remaining nodes and finally magnify the sum by $1/p$. Then during testing, the algorithm sums up the activations of all nodes with no dropout. With probability p, the dropout technique is creating up to $1 \div (1-p)^n$ possible subnetworks during the training. Therefore, dropout can be seen as dividing a network into subnetworks with reused nodes during training. However, for a layer with just enough nodes available, applying dropout will require more nodes to be added, thus potentially increasing the time needed to train a model. In balance, we applied dropout only to the first fully connected layer (FC4) with p=0.5, and L2 regularization to all the hidden layers in Clairvoyante. In practice, we set the lambda of L2 regularization the same as the learning rate.

Visualization

We created an interactive python notebook accessible within a web browser or a command line script for visualizing inputs and their corresponding node activations in hidden layers and output layers. **Supplementary Figure 1** shows the input and node activations in all hidden layers and output layers of an A>G SNP variant in sample HG002 test against a model trained with samples from HG001 for a thousand epochs at $1e^{-3}$ learning rate. Each of the nodes can be considered as a feature deduced through a chain of nonlinear transformations of the read alignments input.

Computational Performance

Making Clairvoyante a computationally efficient tool that can run on modern desktop and server computers with commodity configurations is one of our primary targets. Here, we introduce the two critical methods used for decreasing computational time and memory consumption.

Clairvoyante can be roughly divided into two groups of code, one is sample preparation (preprocessing and model training), and the second is sample evaluation (model evaluation and visualization). Model training runs efficiently because it invokes Tensorflow, which is maintained by a large developer community and has been intensively optimized with most of its performance critical code written in C, C++ or CUDA. Using the native python interpreter, sample preprocessing became the bottleneck, and the performance did not improve by using multi-threading due to the existence of Global Interpreter Lock (GIL). We solved the problem by using Pypy[22], a Just-In-Time (JIT) compiler that performs as an alternative to the native python interpreter and requires no change to our code. In our study, Pypy sped up the sample preparation code by 5 to 10 times.

The memory consumption in model training was also a concern. For example, with a naïve encoding, HG001 requires 40GB memory to store the variant and non-variant samples, which could prevent effective GPU utilization. We observed that these samples are immutable and follow the "write once, read many" access pattern. Thus, we applied in-memory compression using the blosc[23] library with the lz4hc compression algorithm, which provides a high compression ratio, 100MB/s compression rate, and an ultra-fast decompression rate at 7GB/s. Our benchmarks show that applying in-memory compression does not impact the speed but decreased the memory consumption by five times.

Results

In this section, we first benchmarked Clairvoyante on Illumina, PacBio, and ONT data at known variant sites. Based on the benchmarking results, we have addressed several important questions regarding the results, the model training, and the input data. Last, we evaluated Clairvoyante's performance to call variants genome-wide.

Training Runtime Performance

We recommend using GPU acceleration for model training and CPU-only for variant calling. **Table 1** shows the performance of different GPU and CPU models in training. Using a high-performance desktop GPU model GTX 1080 Ti, 170 seconds are needed per epoch, which leads to about 5 hours to finish training a model with the fast training mode. However, for variant calling the speed up by GPU is insignificant because CPU workloads such as VCF file formatting and I/O operations dominate. Variant calling at 3.5M known variant sites takes about 20 minutes using 28 CPU cores. Variant calling genome-wide varies between 30 minutes to a few hours subject to which sequencing technology and alternative allele frequency cutoff were used.

Table 1. Time per epoch of different models of GPU and CPU in model training.

Equipment	Seconds per Epoch per 11M samples
GTX 1080 Ti	170
GTX 980	250
GTX Titan	520
Tesla K40 w/ top power setting	580
Tesla K40	620
Tesla K80 (one socket)	700
GTX 680	780
Intel Xeon E5-2680 v4 28-core	2900

Call Variants at Known Sites

Clairvoyante was designed targeting SMS, nevertheless, the method is generally applicable for short read data as well. We benchmarked Clairvoyante on three sequencing technologies: Illumina, PacBio, and ONT using both the fast and the nonstop training mode. In nonstop training mode, we started training the model from 0 to 999-epoch at learning rate $1e^{-3}$, then to 1499-epoch at $1e^{-4}$, and finally to 1999-epoch at $1e^{-5}$. We then benchmarked the model generated by the fast mode, and all three models stopped at different learning rates in the nonstop mode. We also benchmarked variant calling on one sample (e.g., HG001) using a model trained on another sample (e.g., HG002). Further, we ran GATK UnifiedGenotyper[6] and GATK HaplotypeCaller[6] for comparison. Noteworthy, GATK UnifiedGenotyper was superseded by GATK HaplotypeCaller, thus for Illumina data, we should refer to the results of HaplotypeCaller as the true performance of GATK. However, our benchmarks show that UnifiedGenotyper performed better than HaplotypeCaller on the PacBio and ONT data, thus we also benchmarked UnifiedGenotyper for all three technologies for users to make parallel comparisons. We also attempted to benchmark other tools for SMS reads including PacBio GenomicConsensus v5.1[24], and Nanopolish v0.9.0[25], but we only completed the benchmark with Nanopolish. The reason why the other tools failed, and the commands used for generating the results in this section are presented in **Supplementary Material, Call Variants at Known Sites, Commands**.

The benchmarks at known GIAB truth variant sites i) provides a clear view of how sequencing technologies perform differently with Clairvoyante and other tools in the high confident genome regions, which in turn ii) enables the detailed assessment of Clairvoyante including testing for overfitting, higher data quality and network capacity. The benchmarks also iii) support the expected performance of Clairvoyante on a typical precision medicine application that only tens to hundreds of clinically relevant or actionable variants are being genotyped. This is becoming increasingly important in recent days as SMS is becoming more widely used for clinical diagnosis of structural variations, but at the same time, doctors and researchers also want to know if there exist any actionable or incidental small variants without additional short read sequencing[26]. So firstly, we have evaluated Clairvoyante's performance on known GIAB truth variant sites before extending the evaluation genome-wide. The latter is described in the section named "**Genome-wide variant identification**".

We used the submodule *vcfeval* in RTG Tools[27] version 3.7 to benchmark our results and generate three metrics including Precision, Recall, and F1-score. From the number of true positives (*TP*), false positives (*FP*), and false negatives (*FN*), we compute the three metrics as Precision = $TP \div (TP + FP)$, Recall = $TP \div (TP + FN)$, and F1-score = $2TP/(2TP + FN + FP)$. *FP* are defined as variants existing in the GIAB dataset that also identified as a variant by Clairvoyante, but with discrepant variant type, alternative allele or zygosity. *FN* are defined as the variants existing in the GIAB dataset but identified as a non-variant by Clairvoyante. F1-score is the harmonic mean of the precision and recall. RTG *vcfeval* also provides the best variant quality cutoff for each dataset, filtering the variants under which can maximize the F1-score. To the best of our knowledge, RTG *vcfeval* was also used by the GIAB project itself. *vcfeval* cannot deal with Indel variant calls without an exact allele. However, in our study, Clairvoyante was set to provide the exact allele only for Indels ≤4bp. Thus, for Clairvoyante, all Indels >4bp were removed from both the baseline and the variant calls before benchmarking. The commands used for benchmarking are presented in **Supplementary Material, Benchmarking, Commands**.

Table 2 shows the performance of Clairvoyante on Illumina data. The best accuracy is achieved by calling variants in HG001 using the model trained on HG001 at 1499-epoch, with 99.73% precision, 99.62% recall and 99.68% F1-score. A major concern of using deep learning or any statistical learning technique for variant calling is the potential for overfitting to the training samples. Our results show that Clairvoyante is not affected by overfitting, and we validated the versatility of the trained models by calling variants in a genome using a model trained on a second sample. Interestingly, the performance of calling variants in HG002 using a model trained on HG001 (for convenience, hereafter denoted as HG002>HG001) is 0.25% higher (99.52% against 99.27%) than HG002>HG002 and similar to HG001>HG001. As we know the truth variants in HG001 were verified and rectified by more orthogonal genotyping methods than HG002[12], we believe it is the higher quality of truth variants in HG001 than HG002 that gave the model trained on HG001 a higher performance. Clairvoyante achieved 0.14% higher (99.68% against 99.57%) F1-score than GATK UnifiedGenotyper on HG001 but 0.03% lower (99.52% against 99.55%) on HG002. This again corroborated the importance of high-quality truth variants for Clairvoyante to achieve superior performance.

Table 3 shows the performance of Clairvoyante on PacBio data. The best performance is achieved by calling variants in HG001 using the model trained on HG001 at 1999-epoch, with 97.65% precision, 96.53% recall and 97.09% F1-score. As previously reported, DeepVariant[10] has benchmarked the same dataset in their studied and reported 97.25% precision, 88.51% recall and 92.67% F1-score. We noticed our benchmark differs from DeepVariant because we have removed Indels >4bp (e.g. 52,665 sites for GRCh38 and 52,709 for GRCh37 in HG001) from both the baseline and variant calls. If we assume DeepVariant can identify all the 91k Indels >4bp correctly, it's recall will increase to 90.73%, which is still 5.8% lower than Clairvoyante.

Table 4 shows the performance of Clairvoyante on ONT data. As there are no available deep coverage ONT datasets for HG002, we provided two sets of benchmarks including 1) variant calls in all chromosomes of HG001 using models trained on the same chromosomes, and 2) variant calls in the chromosome 1 of HG001 using models trained on all chromosomes of HG001 except for the chromosome 1. The first benchmark (genome-wide training and calling) achieves the best precision of 95.36% at 1499-epoch. The best recall is 88.70%, and the best F1-score is 91.83%, both achieved at 1999-epoch. The second benchmark (variant calls on Chr1 and genome-wide training) is similar to the first benchmark and is slightly better. It shows the best precision is 96.85%, the best recall is 90.69% and the best F1-score is 93.67%, all achieved at 1999-epoch. We also benchmarked Nanopolish[25] using the same dataset, using 28 CPU cores we called variants in chr19 in about eleven hours. Nanopolish achieved 97.09%, 80.56% and 88.06% on precision, recall, and F1-score, respectively (SNP: 98.10%, 88.91% and 93.28%, Indel: 87.49%, 33.52% and 48.47%). In addition, we have applied Nanopolish to the whole genome of HG001. Also using 28 CPU cores, it finished in 40 days and achieved 97.41%, 84.46% and 90.47% on precision, recall, and F1-score, respectively (SNP: 98.28%, 92.60% and 95.36%, Indel: 88.28%, 37.50% and 52.64%).

Table 2. Performance of Clairvoyante on Illumina data at known variant sites. *: fast training mode.

Seq. Tech.			Ending Learnin		Best Varian	Overall	SNP	Indel

	Model Trained on	Trained Epochs	g Rate and Lambda	Call Variants in	t Quality Cutoff	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Illumina	HG001	67*	1.E-05	HG001	54	99.68%	99.50%	99.59%	99.94%	99.92%	99.93%	97.85%	96.81%	97.33%
		999	1.E-03		72	99.71%	99.58%	99.65%	99.94%	99.93%	99.94%	98.07%	97.24%	97.65%
		1499	1.E-04		93	99.73%	99.62%	99.68%	99.94%	99.93%	99.93%	98.24%	97.52%	97.88%
		1999	1.E-05		91	99.73%	99.62%	99.68%	99.94%	99.93%	99.93%	98.23%	97.51%	97.87%
	HG001	67*	1.E-05	HG002	54	99.63%	99.38%	99.50%	99.90%	99.81%	99.85%	97.63%	96.44%	97.03%
		999	1.E-03		82	99.64%	99.41%	99.52%	99.90%	99.82%	99.86%	97.72%	96.60%	97.16%
		1499	1.E-04		118	99.60%	99.38%	99.49%	99.89%	99.81%	99.85%	97.44%	96.44%	96.93%
		1999	1.E-05		129	99.58%	99.37%	99.47%	99.89%	99.81%	99.85%	97.33%	96.33%	96.83%
	HG002	66*	1.E-05	HG001	60	99.26%	98.98%	99.12%	99.50%	99.87%	99.68%	97.48%	93.22%	95.30%
		999	1.E-03		83	99.26%	99.04%	99.15%	99.51%	99.88%	99.70%	97.45%	93.43%	95.40%
		1499	1.E-04		121	99.21%	99.00%	99.11%	99.49%	99.87%	99.68%	97.12%	93.18%	95.11%
		1999	1.E-05		141	99.20%	98.98%	99.09%	99.50%	99.87%	99.68%	97.04%	93.06%	95.01%
	HG002	66*	1.E-05	HG002	51	99.29%	99.07%	99.18%	99.51%	99.85%	99.68%	97.56%	93.59%	95.53%
		999	1.E-03		76	99.32%	99.15%	99.24%	99.53%	99.87%	99.70%	97.76%	94.05%	95.87%
		1499	1.E-04		75	99.33%	99.21%	99.27%	99.52%	99.88%	99.70%	97.82%	94.30%	96.03%
		1999	1.E-05		85	99.33%	99.21%	99.27%	99.52%	99.88%	99.70%	97.83%	94.30%	96.03%
GATK UnifiedGenotyper, HG001					6	99.74%	99.41%	99.57%	99.89%	99.92%	99.90%	98.82%	96.28%	97.54%
GATK HaplotypeCaller, HG001					5	99.90%	99.81%	99.85%	99.99%	99.96%	99.97%	99.34%	98.92%	99.13%
GATK UnifiedGenotyper, HG002					3	99.74%	99.36%	99.55%	99.86%	99.84%	99.85%	98.97%	96.29%	97.61%
GATK HaplotypeCaller, HG002					5	99.91%	99.80%	99.86%	99.97%	99.90%	99.93%	99.51%	99.19%	99.35%

Table 3. Performance of Clairvoyante on PacBio data at known variant sites. *: fast training mode.

Seq. Tech.	Model Trained on	Trained Epochs	Ending Learning Rate and Lambda	Call Variants in	Best Variant Quality Cutoff	Overall			SNP			Indel		
						Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
PacBio	HG001	50*	1.E-05	HG001	45	96.91%	94.35%	95.62%	99.57%	99.47%	99.52%	70.99%	60.14%	65.12%
		999	1.E-03		52	97.46%	95.42%	96.43%	99.69%	99.65%	99.67%	76.69%	67.04%	71.54%
		1499	1.E-04		55	97.68%	96.38%	97.03%	99.74%	99.72%	99.73%	79.84%	73.45%	76.51%
		1999	1.E-05		52	97.65%	96.53%	97.09%	99.73%	99.72%	99.72%	79.91%	74.32%	77.01%
	HG001	50*	1.E-05	HG002	48	96.65%	94.16%	95.39%	99.38%	99.28%	99.33%	70.67%	60.19%	65.01%
		999	1.E-03		58	96.94%	94.43%	95.67%	99.40%	99.30%	99.35%	73.40%	61.82%	67.12%
		1499	1.E-04		63	96.66%	94.35%	95.49%	99.38%	99.28%	99.33%	71.08%	60.31%	65.26%
		1999	1.E-05		60	96.54%	94.37%	95.44%	99.38%	99.29%	99.33%	70.21%	60.11%	64.76%
	HG002	72*	1.E-05	HG001	38	96.97%	93.11%	95.00%	99.33%	99.20%	99.27%	69.94%	52.43%	59.93%
		999	1.E-03		68	97.50%	92.72%	95.05%	99.29%	99.12%	99.21%	74.47%	51.93%	61.19%
		1499	1.E-04		75	96.94%	92.98%	94.92%	99.04%	98.97%	99.00%	72.28%	52.23%	60.64%
		1999	1.E-05		75	96.68%	92.85%	94.73%	98.91%	98.85%	98.88%	70.92%	51.38%	59.59%
	HG002	72*	1.E-05	HG002	34	96.83%	94.46%	95.63%	99.53%	99.47%	99.50%	71.18%	60.74%	65.55%
		999	1.E-03		36	96.83%	95.51%	96.17%	99.64%	99.62%	99.63%	72.85%	66.60%	69.58%
		1499	1.E-04		68	98.13%	95.73%	96.91%	99.74%	99.75%	99.74%	82.76%	70.96%	76.40%
		1999	1.E-05		51	97.65%	96.33%	96.99%	99.67%	99.77%	99.72%	80.21%	72.87%	76.36%
GATK UnifiedGenotyper, HG001					1	68.54%	23.82%	35.36%	68.54%	27.40%	39.14%	-	-	-
GATK HaplotypeCaller, HG001					1	64.66%	1.95%	3.79%	65.30%	2.24%	4.33%	7.02%	0.02%	0.04%
GATK UnifiedGenotyper, HG002					1	69.08%	23.77%	35.37%	69.08%	27.39%	39.23%	-	-	-
GATK HaplotypeCaller, HG002					1	66.82%	1.19%	2.33%	66.82%	1.37%	2.68%	52.00%	0.003%	0.006%

Table 4. Performance of Clairvoyante on ONT data at known variant sites. *: fast training mode.

Seq. Tech.	Model Trained on	Trained Epochs	Ending Learning Rate and Lambda	Call Variants in	Best Variant Quality Cutoff	Overall			SNP			Indel		
						Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Oxford Nanopore	HG001	110*	1.E-05	HG001	33	94.90%	84.35%	89.34%	96.40%	92.73%	94.53%	62.49%	25.53%	36.25%
		999	1.E-03		33	94.07%	85.87%	89.79%	96.02%	93.23%	94.60%	63.49%	34.49%	44.70%
		1499	1.E-04		37	95.36%	88.12%	91.59%	97.31%	95.18%	96.23%	66.91%	40.10%	50.15%
		1999	1.E-05		37	95.20%	88.70%	91.83%	97.28%	95.55%	96.41%	66.67%	42.06%	51.58%
	HG001 (except for chr1)	110*	1.E-05	HG001 (chr1)	29	95.71%	84.83%	89.94%	96.62%	93.02%	94.79%	64.42%	27.45%	38.50%
		999	1.E-03		29	95.88%	87.74%	91.63%	96.11%	93.55%	94.81%	65.27%	37.29%	47.47%
		1499	1.E-04		31	95.85%	89.58%	92.61%	97.71%	95.63%	96.66%	69.09%	43.13%	53.11%
		1999	1.E-05		33	96.85%	90.69%	93.67%	97.47%	95.78%	96.62%	68.63%	43.37%	53.15%
Nanopolish, HG001, chr19 only					-	97.09%	80.56%	88.06%	98.10%	88.91%	93.28%	87.49%	33.52%	48.47%
Nanopolish, HG001					-	97.41%	84.46%	90.47%	98.28%	92.60%	95.36%	88.28%	37.50%	52.64%
GATK UnifiedGenotyper, HG001					0	82.15%	15.43%	25.99%	82.15%	17.75%	29.19%	-	-	-
GATK HaplotypeCaller, HG001					1	75.13%	1.26%	2.48%	75.52%	1.45%	2.84%	16.50%	0.01%	0.03%

Characterization of potential false positives and false negatives

While we have arrived at a highly optimized version of Clairvoyante for the experiments in this paper, it is essential to study the remaining FP and FN variant calls and how they are

distributed to support for future improvements. To achieve this, on Illumina data, we have randomly picked 100 FP and 100 FN from the variants called in HG002 using the model trained on HG001 using the fast training mode (stopped at 67-epoch), generated plots on their input and output and manually inspected each one. A summary of the results is shown in **Figure 3**. The most significant category of FP and FN variants, accounting for 71 FP and 42 FN, are variants with two or more alternative alleles at the same position. Clairvoyante does not currently support this type of variant, and instead, only one allele will be reported (this limitation is further discussed in the **Discussion**). Except for 1 FP and 7 FN that have no read coverage at all (because we have downsampled from 300x to 50x), the other 28 FP and 51 FN are errors that Clairvoyante should avoid. Among them, 13 FP and 2 FN failed because of relatively "difficult reference" (low complexity sequence, tandem repeat or homopolymer run), 3 FP and 18 FN because of "lack of evidence" (depth ≤ 20 or even ≤ 3). The results suggest that to improve Clairvoyante further, we should increase the accuracy of the variants in the "difficult reference" regions and increase the sensitivity of the variants "lack of evidence". Noteworthy, the 1 FP Clairvoyante made with no read coverage at all is specific to the "Call variant at known sites" mode since Clairvoyante will decide on each known site regardless of covered or not. This type of FP could be easily eliminated by filtering the variants with zero depth, but we have retained it in our study to show a complete spectrum of errors Clairvoyante has made. More details for each FP and FN are shown in **Supplementary Tables 1, and 2** and the plots are available online (**Supplementary Material, Call Variants at Known Sites, Resources, FP/FN plots**).

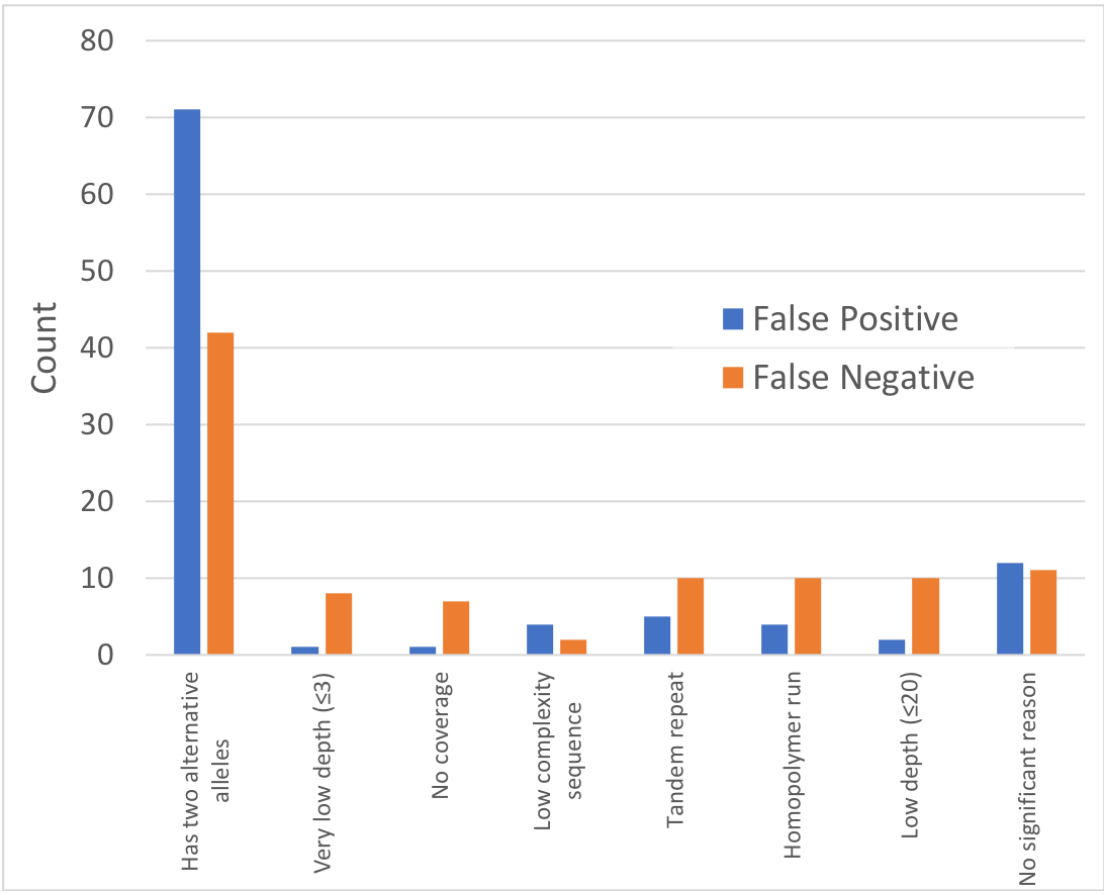

Figure 3. Summary of the reason for failure on 100 randomly picked false positive variants, and 100 randomly picked false negative variants.

Can lower learning rate and longer training provide better performance?

The benchmarking results on the three models stopping at different learning rates allow us to study whether lower learning rate can provide better results and derive how much training is enough. For ONT, both from 999-epoch to 1499-epoch and from 1499-epoch to 1999-epoch, significant improvements were observed. However, in PacBio (**Table 3**), from 1499-epoch to 1999-epoch, the F1-Score increased (97.03% to 97.09%, 96.91% to 96.99%) when both variant calling and model training is using the same sample, but decreased (95.49% to 95.44%, 94.92% to 94.73%) when using different samples. The results suggest that Clairvoyante was overfitting the training data with a too low learning rate. The same behavior is also observed in Illumina data (**Table 2**). Thus, we suggest the Clairvoyante users to 1) stop at a higher learning rate for less noisy data; 2) train multiple samples stopping at different learning rates and select the best through performance evaluation; or 3) use a model trained on truth variants from multiple samples.

Can a model train on truth variants from multiple samples provide better performance?

Intuitively, a model trained on truth variants from two or more samples should perform better than those trained on just a single sample, provided that the truth variants from different samples have similar high quality. The model might even be more versatile if the characteristics of input, such as average depth, differ between samples. To verify our hypothesis, we benchmarked the variants called in HG003 (**Supplementary Material, Data Source, PacBio Data**) on three different models trained on 1) HG001; 2) HG002, and; 3) HG001+HG002. All three models were trained for 1000 epochs at learning rate $1e^{-3}$, then another 500 epochs at learning rate $1e^{-4}$. Noteworthy, the time used for training the HG001+HG002 model doubled, as it doubled the number of true variants and paired non-variants. If our hypothesis is correct, the variant calling performance should increase for HG003 when using the HG001+HG002 model than the HG001 model or the HG002 model. The results are shown in **Table 5**. Using the HG001+HG002 model, the F1-score is 0.55% higher than using HG001 only and 2.88% higher than using the HG002 only. We conclude that using multiple samples for model training can increase the performance of Clairvoyante, although we expect marginal improvement gains when using more than a few samples.

Table 5. Performance of variant calls in HG003 on three different models including HG001 only, HG002 only and HG001+HG002.

Train using Variants in	Best Variant Quality Cutoff	Overall			SNP			Indel		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
HG001	72	95.61%	90.51%	92.99%	95.61%	90.51%	92.99%	95.61%	90.51%	92.99%
HG002	71	93.38%	88.09%	90.66%	93.38%	88.09%	90.66%	93.38%	88.09%	90.66%
HG001 + HG002	56	95.91%	91.29%	93.54%	95.91%	91.29%	93.54%	95.91%	91.29%	93.54%

Can a higher input data quality improve the variant calling performance?

In Table 4, we used the 'rel3' ONT dataset generated by the Nanopore WGS consortium. Very recently, the consortium released an augmented dataset labeled 'rel5' (see **Supplementary Material, Data Source, Oxford Nanopore Data**). The 'rel5' data are a merge of NA12878 DNA sequencing data from 'rel3' (regular sequencing protocols, about 30x) and 'rel4' (ultra-read set, 7.7x extra), recalled with the latest base-caller. Thus, we

expect to see improved performance, given that the input data quality limited the performance of Clairvoyante on ONT. We trained a model on 'rel5' for 999 epochs at learning rate $1e^{-3}$. Compare to 'rel3', the precision improved from 94.07% to 97.21%, the recall improved from 85.87% to 88.80%, and the F1-Score improved from 89.79% to 92.81%. Thus, the results reflect our intuition that Clairvoyante's performance on ONT data is limited by the input data quality and thus will improve over time as the technology, base-calling mature, and more data become available.

Network topology and capacity evaluation

In the previous subsection, we have shown Clairvoyante's capacity to perform better on noisy PacBio and ONT data when trained with more data of higher quality. We next evaluated the performance by considering a "slim version" of Clairvoyante with smaller capacity that could potentially improve computational requirements. With the slim version, we expect to see a greater performance in higher quality Illumina data than noisy data like ONT and PacBio data as the classification problem is easier with less noisy data. The slim version includes 165k parameters, which is about ten times fewer than the original version. Instead of isometrically scaling down the original network, we evaluated several different designs resulting in some network components with significantly reduced runtime than others or even reducing the parameters by ten times while still achieving the best runtime and F1-Score possible.

Our final slim network design removes the pooling between convolutional layers, slightly enlarged the kernel size in convolution and reduced the number of nodes in the two fully-connected layers by ten times. We trained models using the fast training mode on HG001 and benchmarked the Illumina, PacBio and ONT data on both HG001 and HG002. The results are shown in **Table 6**. As expected, the F1-scores degraded least in the Illumina datasets (0.82% and 0.73%) and degraded most in the ONT dataset (2.23%), with PacBio in the middle (1.68% and 1.90%). The slim version is available as a part of the Clairvoyante toolset and can be enabled with option '--slim.'

Table 6. F1-scores of different datasets on different network designs. Both the original models and slim models were trained on HG001 using the fast training mode.

	Illumina		PacBio		ONT
Models:	HG001	HG002	HG001	HG002	HG001
Original	99.59%	99.50%	95.62%	95.39%	89.34%
Slim	98.77%	98.77%	93.94%	93.49%	87.11%
Degraded	0.82%	0.73%	1.68%	1.90%	2.23%

Genome-wide Variant Identification

Beyond benchmarking variants at sites known to be variable in a sample, in this section, we benchmarked Clairvoyante's performance on calling variants genome-wide. Calling variants genome-wide is challenging because it tests not only how good Clairvoyante can derive the correct variant type, zygosity and alternative allele of a variant when evidence is marginal, but also in reverse, how good Clairvoyante can filter/suppress a non-variant even in the presence of sequencing errors or other artificial signals. Instead of naively evaluating all three billion sites of the whole genome with Clairvoyante, we tested the performance at different alternative allele cutoffs for all three sequencing technologies. As expected, a higher allele cutoff speeds up variant calling by producing fewer candidates to be tested by Clairvoyante but worsens recall especially for noisy data like PacBio and ONT. Our experiments provide a

reference point on how to choose a cutoff for each sequencing technology to achieve a good balance between recall and running speed. All models were trained for 1000 epochs with learning rate at $1e^{-3}$. All the experiments were performed on two Intel Xeon E5-2680 v4 using all 28 cores. The commands used for generating the results in this section are presented in **Supplementary Material, Call Variants Genome-wide, Commands**.

The results are shown in **Table 7**. As expected, with higher alternative allele frequency threshold (0.2), the precision was higher while the recall and time consumption was reduced in all experiments. For Illumina data, the best F1-score (with 0.2 allele frequency) for Clairvoyante was 98.65% for HG001 and 98.61% for HG002. The runtime varied between half and an hour (40 minutes for the best F1-score). As expected, GATK HaplotypeCaller topped the performance on Illumina data - achieved F1-score 99.76% for HG001 and 99.70% for HG002; both ran for about 8 hours. GATK UnifiedGenotyper ran as fast as Clairvoyante on Illumina data and achieved F1-score 99.43% for HG001 and 99.08% for HG002. Inspecting the false positive and false negative variant calls for Clairvoyante, we found about 0.19% in FP, and 0.15% in FN was because of scenarios of two alternative alleles. We realized, on Illumina data, Clairvoyante is not performing on-par with the state-of-the-art GATK HaplotypeCaller, which was intensively optimized for Illumina data. However, as Clairvoyante uses an entirely different algorithm than GATK, Clairvoyante's architecture could be used as an orthogonal method, emulating how geneticists manually validate a variant using a genome browser, for filtering or validating GATK's results to increase GATK's accuracy further. We implemented this in a method called Skyhawk. It repurposed Clairvoyante's neural network to work on the GATK's variants, give them another quality score in addition to the existing one by GATK, and give suggestion on disagreed answers. More details are available in Skyhawk's preprint [28]. With the success of developing Skyhawk, we expect to see in the future, more applications would be developed upon Clairvoyante's network architecture.

For the PacBio data, the best F1-scores were also achieved at 0.2 allele frequency cutoff. The best F1-score is 92.57% for HG001 and 93.05% for HG002 running Clairvoyante for ~3.5 hours. In contrast, as reported in their paper[10], DeepVariant has achieved 35.79% F1-score (22.14% precision, 93.36% recall) on HG001 with PacBio data. The runtime for Clairvoyante at 0.25 frequency cutoff is about 2 hours, which is about half the time consumption at 0.2 frequency cutoff, and about 1/5 the time consumption at 0.1 frequency cutoff. For ONT data (rel3), the best F1-score 77.89% was achieved at 0.1 frequency cutoff. However, the F1-score at 0.25 frequency cutoff is just slightly lower (76.95%), but ran about five times faster, from 13 hours to less than three hours. Thus, we suggest using 0.25 as the frequency cutoff. The runtime is on average about 1.5 times longer than PacBio, due to the higher level of noise in data. Using the new rel5 ONT data with better base calling quality, the best F1-score has increased from 87.26% (9.37% higher than rel3). The recall of SNP and the precision of Indel were the most substantially increased.

Table 7. Performance of using Clairvoyante for variant calling genome-wide on Illumina, PacBio and ONT datasets. All models were trained for 1000 epochs with learning rate at $1e^{-3}$.

Seq. Tech	Train using Variants in	Call Variants in	Alt. Allele Freq. Cutoff	Best Variant Quality Cutoff	Time Consumption	Overall			SNP			Indel		
						Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Illumina	HG001	HG001	0.1	189	1:08	98.16%	98.93%	98.55%	98.10%	99.92%	99.00%	98.63%	96.96%	97.79%
			0.2	182	0:43	98.41%	98.88%	98.65%	98.38%	99.90%	99.13%	98.70%	95.10%	96.86%
			0.25	180	0:26	98.71%	97.95%	98.33%	98.72%	99.82%	99.27%	98.62%	87.33%	92.63%

		HG002	0.1	192	1:11	98.13%	98.77%	98.45%	98.08%	99.81%	98.94%	98.50%	96.54%	97.51%
			0.2	183	0:41	98.35%	98.77%	98.56%	98.33%	99.78%	99.05%	98.51%	94.90%	96.67%
			0.25	182	0:30	98.67%	97.88%	98.27%	98.70%	99.69%	99.19%	98.43%	87.34%	92.55%
	HG002	HG001	0.1	198	1:16	98.59%	98.50%	98.54%	98.68%	99.88%	99.27%	97.98%	93.37%	95.62%
			0.2	192	0:47	98.75%	98.39%	98.57%	98.84%	99.86%	99.35%	98.07%	91.69%	94.78%
			0.25	184	0:25	98.94%	97.60%	98.27%	99.07%	99.78%	99.42%	97.91%	85.01%	91.00%
		HG002	0.1	195	1:07	98.53%	98.59%	98.56%	98.59%	99.85%	99.22%	98.12%	93.78%	95.90%
			0.2	188	0:44	98.71%	98.50%	98.61%	98.77%	99.81%	99.29%	98.22%	92.24%	95.14%
			0.25	182	0:25	98.95%	97.73%	98.33%	99.05%	99.71%	99.38%	98.11%	85.73%	91.50%
	GATK UnifiedGenotyper, HG001			51	0:46	99.43%	99.42%	99.43%	99.53%	99.91%	99.72%	98.76%	96.47%	97.60%
	GATK HaplotypeCaller, HG001			5	8:45	99.69%	99.83%	99.76%	99.77%	99.97%	99.87%	99.22%	98.97%	99.09%
	GATK UnifiedGenotyper, HG002			4	0:46	98.76%	99.41%	99.08%	98.73%	99.85%	99.29%	98.91%	96.57%	97.73%
	GATK HaplotypeCaller, HG002			5	8:23	99.59%	99.81%	99.70%	99.62%	99.90%	99.76%	99.39%	99.24%	99.32%
PacBio	HG001	HG001	0.1	157	9:46	96.31%	88.63%	92.31%	96.72%	99.49%	98.09%	79.19%	31.13%	44.69%
			0.2	130	3:53	98.12%	87.62%	92.57%	98.96%	96.60%	97.77%	75.87%	31.50%	44.52%
			0.25	125	2:01	98.62%	83.11%	90.20%	99.39%	91.38%	95.22%	78.55%	27.10%	40.30%
		HG002	0.1	153	9:24	97.00%	89.08%	92.87%	97.90%	99.15%	98.52%	71.57%	34.26%	46.34%
			0.2	132	3:34	97.93%	88.30%	92.86%	99.03%	97.05%	98.03%	73.37%	34.06%	46.53%
			0.25	116	1:46	98.06%	84.69%	90.89%	99.18%	92.53%	95.74%	75.56%	31.24%	44.21%
	HG002	HG001	0.1	163	14:55	95.58%	86.69%	90.92%	96.64%	98.96%	97.79%	59.19%	24.52%	34.67%
			0.2	147	3:29	97.49%	85.64%	91.18%	98.94%	96.13%	97.51%	58.24%	23.65%	33.64%
			0.25	139	1:39	98.16%	81.47%	89.04%	99.27%	90.90%	94.90%	66.31%	21.11%	32.02%
		HG002	0.1	150	15:31	97.10%	89.31%	93.04%	98.33%	99.14%	98.73%	69.19%	39.77%	50.51%
			0.2	134	3:34	98.09%	88.51%	93.05%	99.35%	97.30%	98.32%	72.20%	36.59%	48.57%
			0.25	118	1:46	98.20%	84.76%	90.98%	99.47%	92.77%	96.00%	72.94%	31.44%	43.94%
Oxford Nanopore (rel3)	HG001	HG001	0.1	140	13:01	86.24%	71.01%	77.89%	86.79%	91.85%	89.25%	55.36%	10.69%	17.93%
			0.2	139	4:47	87.24%	70.21%	77.80%	87.72%	87.97%	87.85%	59.05%	9.90%	16.96%
			0.25	136	2:40	87.76%	68.51%	76.95%	88.25%	82.86%	85.47%	59.41%	9.26%	16.03%
			0.35	130	1:30	90.96%	57.43%	70.41%	91.34%	65.82%	76.51%	67.35%	6.62%	12.06%
Oxford Nanopore (rel5)	HG001	HG001	0.2	162	5:53	88.76%	85.81%	87.26%	88.95%	93.76%	91.29%	72.10%	8.32%	14.92%
			0.25	159	3:12	89.14%	82.20%	85.53%	89.34%	90.09%	89.71%	72.45%	8.02%	14.45%
			0.35	148	1:51	91.22%	67.88%	77.83%	91.48%	75.18%	82.53%	71.25%	6.54%	11.98%

For readers to compare the whole-genome benchmarks to those at the GIAB known sites more efficiently, we summarized the best precision, recall, and F1-score of both types of benchmarks in **Supplementary Table 3**.

Benchmarks of other state-of-the-art variant callers

DeepVariant is the first deep neural network based variant caller[10]. After the first preprint of Clairvoyante was available, Google released a new version of DeepVariant (v0.6.1). On Illumina data, the new version was reported to be outperforming the previous versions. We benchmarked the new version to see how it performs on Illumina data and especially on SMS data. We used DeepVariant version 0.6.1 for benchmarking following guide "Improve DeepVariant for BGISEQ germline variant calling" written by Pi-Chuan Chang available at link https://goo.gl/tg4FWG with specific guidelines for how to run DeepVariant, including 1) model training using transfer-learning and multiple depths, and 2) variant calling.

On Illumina data, DeepVariant performed extraordinarily (**Table 8**) and matched with the figures previously reported. Following the guide, we applied transfer-learning using both the truth variants and reference calls in chromosome 1 upon the trained model named "DeepVariant-inception_v3-0.6.0+cl-191676894.data-wgs_standard/model.ckpt" that was delivered together with the software binaries. Using a nVidia GTX1080 Ti GPU, we kept running the model training process for 24 hours and picked the model with the best F1-score (using chromosome 22 for validation purpose), which was achieved at about 65 minutes after the training had started. The variant calling step comprises three steps: 1) create calling candidates, 2) variant calling, and 3) post-processing. Using 24 CPU cores, step one ran for 392 minutes and generated 42GB of data. The second step utilized GPU and took 166 minutes. Step 3 ran for only 25 minutes and occupied significantly more memory (15GB) than the previous two steps. For the HG001 sample, the precision rate is 0.9995, and the

recall rate is 0.9991, both extraordinary and exceeding all other available variant callers including Clairvoyante on Illumina datasets.

DeepVariant requires base-quality, thus failed on the PacBio dataset, in which base-quality is not provided. On ONT data (rel5), DeepVariant performed much better than the traditional variant callers that were not designed for long reads, but it performed worse than Clairvoyante (**Table 8**). We also found that DeepVariant's computational resource consumption on long reads is prohibitively high and we were only able to call variants in few chromosomes. The details are as follows. Using transfer-learning, we trained two models for ONT data on chromosome 1 and 21 respectively, and we called variants in chromosome 1 and 22 against the different models. In total we have benchmarked three settings, 1) call variants in chromosome 1 against the chromosome 21 model, 2) call variants in chromosome 22 against the chromosome 21 model, and 3) call variants in chromosome 22 against the chromosome 1 model. Training the models required about 1.5 days until the validation showed a decreasing F1-score with further training. Using 24 CPU cores, the first step of variant calling generated 337GB candidate variants data in 1,683 minutes for chromosome 1 and generated 53G data in 319 minutes for chromosome 21. The second step of variant calling took 1,171 and 213 minutes to finish for chromosome 1 and 22, respectively. The last step took 160 minutes and was very memory intensive, requiring 74GB of RAM for chromosome 1. In terms of F1-score, DeepVariant has achieved 83.05% in chromosome 1, and 77.89% in chromosome 22, against the model trained on chromosome 21. We verified that more samples for model training do not lead to better variant calling performance - using the model trained on chromosome 1, the F1-score dropped slightly to 77.09% for variants in chromosome 22. Using the computational resource consumption on chromosome 1, we estimate the current version of DeepVariant would require 4TB storage and about one month for whole genome variant calling of a genome sequenced with long reads.

We further benchmarked three additional variant callers[29], including Vardict[30] (v20180724), LoFreq[31] (v2.1.3.1), and FreeBayes[32] (v1.1.0-60-gc15b070) (**Table 8**). The performance of Vardict on Illumina data matches the previous study[29]. Vardict requires base quality, thus failed on the PacBio dataset, in which base quality is not provided. Vardict identified only 62,590 variants in the ONT dataset, among them only 231 variants are true positives. The results match with Vardict's paper that was tested on the Illumina data but not yet ready for Single Molecule Sequencing long reads. The performance of LoFreq on Illumina data matches the previous study[29] calling SNP only. To enable Indel calling in LoFreq, BAQ (Base Alignment Quality)[33] needs to be calculated in advance. However, the BAQ calculation works only for Illumina reads, thus for LoFreq, we only benchmarked its performance in SNP calling. Meanwhile, LoFreq does not provide zygosity in the result, prohibited us from using "RTG vcfeval" for performance evaluation. Thus, we considered a true positive in LoFreq as having a matched truth record in 1) chromosome, 2) position and 3) alternative allele. LoFreq requires base quality, thus failed on the PacBio dataset, in which base quality is not provided. The results suggest that LoFreq is capable of SNP detection in Single Molecule Sequencing long reads. Unfortunately, we were unable to finish running Freebayes on both the PacBio dataset and the ONT dataset after they failed to complete on either dataset after running for one month. According to the percentage of genome covered with variant calls, we estimate several months, 65 and 104 machine days on a latest 24-core machine, are required for a single PacBio and ONT dataset, respectively.

GIAB datasets were constructed from a consensus of multiple short-variant callers, thus tend to bias toward easy regions that are accessible by these algorithms[34]. So, we next

benchmarked the Syndip dataset, which is a recent benchmark dataset from the *de novo* PacBio assemblies of two homozygous human cell lines. As reported, the dataset provides a relatively more accurate and less biased estimate of small-variant-calling error rates in a realistic context [34]. The results are in **Table 8** and show that, when using Syndip variants for training, the performance of calling variants in both HG001 and HG002 at known variants remains as good as previously reported. However, using the same model (Syndip), the performance dropped both at the Syndip known sites (excluding variants >4bp, from 99.51% (HG001) to 98.52%) and for the whole genome (excluding variants >4bp, from 94.88% (HG001) to 94.02%). The results support that Syndip contains variants that are harder to identify. To improve Clairvoyante's performance in the hard regions, we suggest users to also include Syndip for creating models.

Table 8. Additional benchmark performance of using Clairvoyante and other state-of-the-art variant callers.

Seq. Tech.	Tool	Model Trained on	Call Variants in	Excluding Indel >4bp	Best Variant Quality Cutoff	Time Consumption	Overall			SNP			Indel		
							Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Illumina	Vardict	N.A.	HG001		N.A.	2:51	90.56%	98.13%	94.19%	94.09%	99.46%	96.70%	70.64%	88.85%	78.71%
	LoFreq	N.A.	HG001		N.A.	4:03	SNP only			75.88%	94.24%	84.07%	-	-	-
	FreeBayes	N.A.	HG001		43	4:54	98.71%	97.95%	98.33%	98.72%	99.82%	99.27%	98.62%	87.33%	92.63%
	DeepVariant	TL + Chr1	HG001		3	9:43	99.94%	99.91%	99.93%	99.99%	99.96%	99.98%	99.64%	99.59%	99.62%
	FreeBayes	N.A.	Syndip		19	5:42	96.00%	94.21%	95.10%	98.91%	97.17%	98.03%	78.20%	74.54%	76.32%
	Clairvoyante	Syndip	Syndip		70	1:23	91.95%	93.05%	92.49%	93.49%	96.32%	94.88%	80.00%	71.13%	75.30%
	Clairvoyante	Syndip	Syndip	Y	70	1:23	92.75%	95.32%	94.02%	93.60%	96.22%	94.89%	84.76%	86.94%	85.83%
	Clairvoyante	Syndip	Syndip (Known Sites)	Y	20	0:09	98.83%	98.22%	98.52%	99.23%	98.69%	98.96%	94.97%	93.82%	94.39%
	Clairvoyante	Syndip	HG001		76	1:05	92.17%	96.05%	94.07%	92.63%	97.71%	95.10%	89.08%	85.91%	87.46%
	Clairvoyante	Syndip	HG001	Y	76	1:05	92.77%	97.09%	94.88%	92.64%	97.72%	95.11%	93.71%	92.67%	93.18%
	Clairvoyante	Syndip	HG001 (Known Sites)	Y	6	0:08	99.53%	99.50%	99.51%	99.81%	99.83%	99.82%	97.58%	97.16%	97.37%
	Clairvoyante	Syndip	HG002		74	1:11	92.19%	96.07%	94.09%	92.61%	97.55%	95.02%	89.25%	86.64%	87.93%
	Clairvoyante	Syndip	HG002	Y	74	1:11	92.73%	97.06%	94.85%	92.62%	97.56%	95.02%	93.63%	93.43%	93.53%
	Clairvoyante	Syndip	HG002 (Known Sites)	Y	9	0:08	99.51%	99.43%	99.47%	99.77%	99.74%	99.75%	97.64%	97.13%	97.38%
Oxford Nanopore	Vardict	N.A.	HG001		N.A.	17:12	0.42%	0.01%	0.01%	2.56%	0.01%	0.01%	0.04%	0.00%	0.01%
	LoFreq	N.A.	HG001		N.A.	6:58	SNP only			82.69%	54.75%	65.88%	-	-	-
	DeepVariant	TL + Chr1	HG001 (Chr22)		3	9:19	90.97%	66.77%	77.02%	91.61%	76.70%	83.50%	65.54%	8.12%	14.45%
	DeepVariant	TL + Chr21	HG001 (Chr1)		3	50:14	93.47%	74.62%	82.99%	94.36%	84.39%	89.10%	65.68%	12.06%	20.38%
	DeepVariant	TL + Chr21	HG001 (Chr22)		3	9:11	92.19%	67.34%	77.83%	92.87%	77.06%	84.23%	69.87%	10.33%	17.99%

Potential novel variants unraveled by PacBio and ONT

The truth SNPs and Indels provided by GIAB were intensively called and meticulously curated, and the accuracy and sensitivity of the GIAB datasets are unmatched. However, since the GIAB variants were generated without incorporating any SMS technology[12], it is possible that we can consummate GIAB by identifying variants not yet in GIAB, but specifically detected both by using the PacBio and the ONT data. For the HG001 sample (variants called in HG001 using a model trained on HG001), we extracted the so-called "false positive" variants (identified genome-wide with a 0.2 alternative allele frequency cutoff) called in both the PacBio and ONT dataset. Then we calculated the geometric mean of the variant qualities of the two datasets, and we filtered the variants with a mean quality lower than 135 (calculated as the geometric mean of the two best variant quality cutoffs, 130 and 139). The resulting catalog of 3,135 variants retained are listed in **Supplementary Table 4**. 2,732 are SNPs, 298 are deletions, and 105 are insertions. Among the SNPs, 1,602 are

transitions, and 1,130 are transversions. The Ti/Tv ratio is ~1.42, which is substantially higher than random (0.5), suggesting a true biological origin. We manually inspected the top ten variants in quality using IGV[35] to determine their authenticity (**Figure 4a** and **Supplementary Figure 2a-2i**). Among the ten variants, we have one convincing example at 2:163,811,179 (GRCh37) that GIAB has previously missed (**Supp. Fig. 2h**). Another seven examples have weaker supports that need to be further validated using other orthogonal methods. Possible artifacts including 1) 7:89,312,043 (**Supp. Fig. 2g**) has multiple SNPs in its vicinity, which is a typical sign of false alignment, 2) 1:566,371 (**Supp. Fig. 2a**), 20:3,200,689 (**Figure 4a**) are located in the middle of homopolymer repeats, which could be caused by misalignment, 3) X:143,214,235 (**Supp. Fig. 2b**) shows significant strand bias in Illumina data, and 4) X:140,640,513 (**Supp. Fig. 2d**), X:143,218,136 (**Supp. Fig. 2e**), and 9:113,964,088 (**Supp. Fig. 2f**) are potential heterozygous variants but with allele frequency notably deviated from 0.5. Two examples are because of the difference in representation - 13:104,270,904 (**Supp. Fig. 2c**) and 10:65,260,789 (**2i**) have other GIAB truth variants in their 5bp flanking regions. Manually inspecting all the 3,135 variants is beyond the scope of this paper. However, our analysis suggests SMS technologies, including both PacBio and ONT, can indeed generate some variants that are not identifiable by short read sequencing. We advocate for additional efforts to look into these SMS specific candidate variants systematically. The targets include not only shortlisting truth variants not yet in GIAB, but also new alignment and variant calling methods and algorithms to avoid detecting spurious variants in SMS data. Our analysis also serves as another piece of evidence that the GIAB datasets are of superior quality and are the enabler of machine learning based downstream applications such as Clairvoyante.

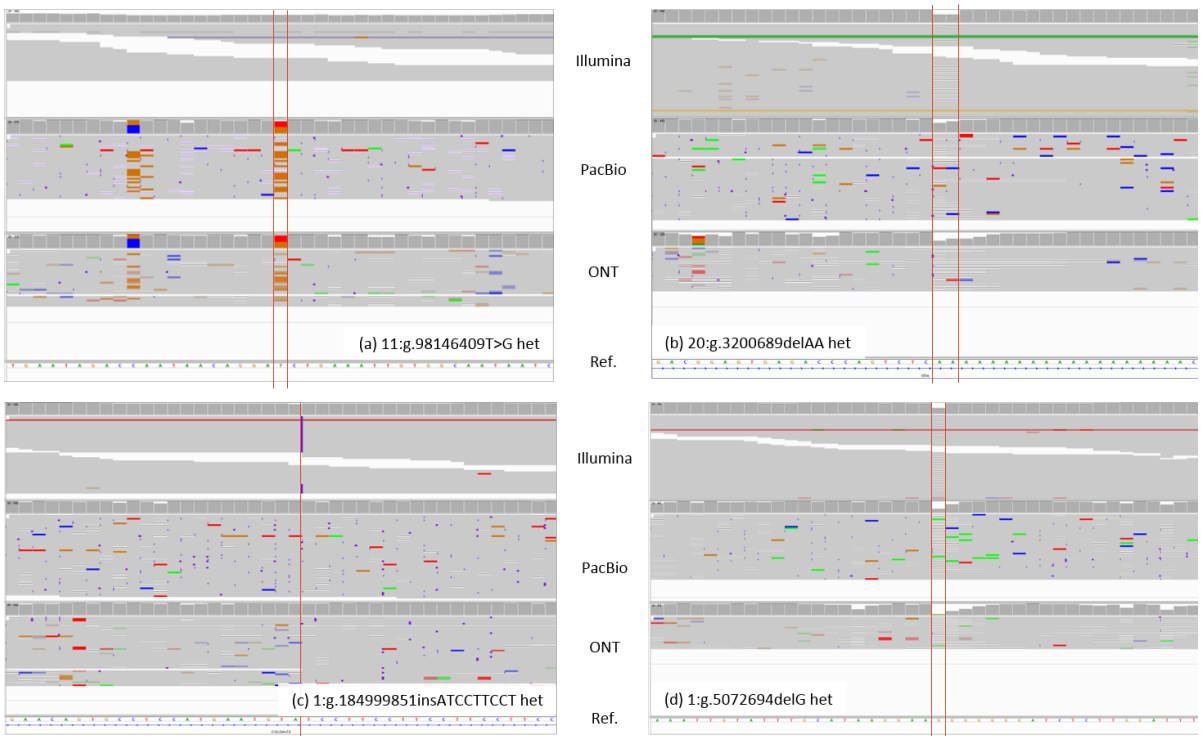

Figure 4. The IGV screen capture of (**a**) a heterozygote SNP from T to G at chromosome 11, position 98,146,409 called only in the PacBio and ONT data, (**b**) a heterozygote deletion AA at chromosome 20, position 3,200,689 not called in all three technologies, (**c**) a heterozygote insertion ATCCTTCCT at chromosome 1, position 184,999,851 called only in the Illumina data, and; (**d**) a heterozygote deletion G at chromosome 1, position 5,072,694 called in all

three technologies. The tracks from top to down show the alignments of the Illumina, PacBio, and ONT reads from HG001 aligned to the human reference GRCh37.

We also analyzed why the PacBio and ONT technologies cannot detect some variants. **Figure 5** shows the number of known variants undetected by different combinations of sequencing technologies. We inspected the genome sequence immediately after the variants and found among the 12,331 variants undetected by all three sequencing technologies, 3,289 (26.67%) are located in homopolymer runs, and 3,632 (29.45%) are located in short tandem repeats. Among the 178,331 variants that cannot be detected by PacBio and ONT, 102,840 (57.67%) are located in homopolymer runs, and 33,058 (18.54%) are located in short tandem repeats. For Illustration, **Figure 4b** to **d** depicted b) a known variant in homopolymer runs undetected by all three sequencing technologies, c) a known variant in short tandem repeats that cannot be detected PacBio and ONT, and d) a known variant flanked by random sequenced detected by all three sequencing technologies. It is a known problem that single molecule sequencing technologies have significantly increased error rates at homopolymer runs and short tandem repeats[36]. Future improvements to the base-calling algorithm and sequencing chemistries will lead to raw reads with higher accuracy at these troublesome genome regions and hence, further decrease the number of known variants undetected by Clairvoyante.

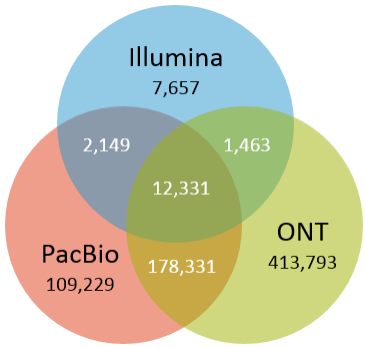

Figure 5. A Venn diagram that shows the number of undetected known variants by different sequencing technologies or combinations.

Discussion

In this paper, we presented Clairvoyante, a multi-task convolutional deep neural network for variant calling using single molecule sequencing. Its performance is on-par with GATK UnifiedGenotyper on Illumina data and outperforms Nanopolish and DeepVariant on PacBio and ONT data. We analyzed the false positive and false negative variant calls in depth and found complex variants with multiple alternative alleles to be the dominant source of error in Clairvoyante. We further evaluated several different aspects of Clairvoyante to assess the quality of the design and how we can further improve its performance by training longer with lower learning rate, combining multiple samples for training, or improving the input data quality. Our experiments on using Clairvoyante to call variants genome-wide suggested a range to search for the best alternative allele cutoff to balance the run time and recall for each sequencing technology. To the best of our knowledge, Clairvoyante is the first method for SMS to finish a whole genome variant calling within two hours on a single CPU-only server, while providing better precision and recall than other state-of-the-art variant callers such as Nanopolish. A deeper look into the so-called “false positive” variant calls has identified 3,135 variants in HG001 that are not yet in GIAB but detected by both PacBio and ONT

independently. Inspecting ten of these variants manually, we identified one strongly supported variant that should be included by GIAB, seven variants with weak or uncertain supports that call for additional validation in a future study, and two variants actually exist in GIAB but with different representation.

Clairvoyante relies on high-quality training samples to provide accurate and unbiased variant calling. This hinders Clairvoyante from being applied to completely novel sequencing technologies and chemistries, for which high-quality sequencing dataset on standard samples such as GIAB has yet been produced. Nevertheless, with the increasing agreement for NA12878 as a gold-standard reference, this requirement seems to be quite manageable. Although Clairvoyante performed well on detecting SNPs, it still has a large room to be improved in detecting Indels, especially for ONT data, in which the Indel F1-score remains around 50%. To make the Indel results also practically usable, our target is to improve Clairvoyante further to reach an Indel F1-score over 80%. The current design of Clairvoyante ignore variants with two or more alternative alleles. Although the number of variants with two or more alternative alleles is small, a few thousands of the 3.5M total sites, the design will be improved in the future to tackle this small but important group of variants. Due to the rareness of long indel variants for model training, Clairvoyante was set to provide the exact alternative allele only for indel variants ≤4bp. The limitation can be lifted with more high-quality training samples available. The current Clairvoyante implementation also does not consider the base quality of the sequencing reads as Clairvoyante was targeting SMS, which do not have meaningful base quality values to improve the quality of variant calling. Nevertheless, Clairvoyante can be extended to consider base quality by imposing it as a weight on depth or add it as an additional tensor to the input. We do not suggest removing any alignment by their mapping quality because low-quality mappings will be learned by the Clairvoyante model to be unreliable. This provides valuable information about the trustworthiness of certain genomic regions. In future work, we plan to extend Clairvoyante to support somatic variant calling and trio-sample based variant calling. Based on GIAB's high confidence region lists for variant calling, we also plan on making PacBio-specific, and ONT-specific high confidence region lists by further investigating the false positive and false negative variant calls made by Clairvoyante on the two technologies.

Acknowledgments

We thank Heng Li and the two anonymous reviewers for their constructive reviews and suggestions. We thank Guangyu Yang for adding code to Clairvoyante to enable visualization using TensorBoard. We thank Chi-Man Liu and Yifan Zhang for benchmarking Nanopolish. R.L. was supported by the General Research Fund No. 27204518, HKSAR. R. L. and T. L. were partially supported by Innovative and Technology Fund ITS/331/17FP from the Innovation and Technology Commission, HKSAR. This work was also supported, in part, by awards from the National Science Foundation (DBI-1350041) and the National Institutes of Health (R01-HG006677 and UM1-HG008898).

Author Contributions

R.L. and M.S. conceived the study. All authors analyzed the data and wrote the manuscript.

References

1 Goodwin, S., McPherson, J. D. & McCombie, W. R. Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet* **17**, 333-351, doi:10.1038/nrg.2016.49 (2016).

2 Nakamura, K. *et al.* Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Res* **39**, e90, doi:10.1093/nar/gkr344 (2011).

3 Hatem, A., Bozdag, D., Toland, A. E. & Catalyurek, U. V. Benchmarking short sequence mapping tools. *BMC Bioinformatics* **14**, 184, doi:10.1186/1471-2105-14-184 (2013).

4 Li, H. Toward better understanding of artifacts in variant calling from high-coverage samples. *Bioinformatics* **30**, 2843-2851, doi:10.1093/bioinformatics/btu356 (2014).

5 Luo, R., Schatz, M. C. & Salzberg, S. L. 16GT: a fast and sensitive variant caller using a 16-genotype probabilistic model. *GigaScience* (2017).

6 Van der Auwera, G. A. *et al.* From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. *Curr Protoc Bioinformatics* **43**, 11 10 11-33, doi:10.1002/0471250953.bi1110s43 (2013).

7 Sedlazeck, F. J., Lee, H., Darby, C. A. & Schatz, M. C. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nat Rev Genet*, doi:10.1038/s41576-018-0003-4 (2018).

8 LeCun, Y. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/* (1999).

9 Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2818-2826.

10 Poplin, R. *et al.* A universal SNP and small-indel variant caller using deep neural networks. *Nature Biotechnology* **2018/09/24/online**, doi:10.1038/nbt.4235 (2018).

11 Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat Biotechnol* **32**, 246-251, doi:10.1038/nbt.2835 (2014).

12 Zook, J. M. *et al.* Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci Data* **3**, 160025, doi:10.1038/sdata.2016.25 (2016).

13 Sedlazeck, F. *et al.* Accurate detection of complex structural variations using single-molecule sequencing. *Nat Methods* **15**, 461-468 (2018).

14 Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat Biotechnol* **36**, 338-345, doi:10.1038/nbt.4060 (2018).

15 Chin, J. *Simple Convolutional Neural Network for Genomic Variant Calling with TensorFlow*, <https://towardsdatascience.com/simple-convolution-neural-network-for-genomic-variant-calling-with-tensorflow-c085dbc2026f> (2017).

16 Abadi, M. *et al.* Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

17 He, K., Zhang, X., Ren, S. & Sun, J. in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)* 1026-1034 (IEEE Computer Society, 2015).

18 Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S. Self-Normalizing Neural Networks. *arXiv preprint arXiv:1706.02515* (2017).

19 Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

20 Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

21 Cortes, C., Mohri, M. & Rostamizadeh, A. in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence.* 109-116 (AUAI Press).

22 Rigo, A. *et al. Pypy*, <https://pypy.org/> (2018).

23 Alted, F. *Blosc: A blocking, shuffling and lossless compression library*, <http://blosc.org/> (2018).

24 Biosciences, P. *Genomic Consensus*, <https://github.com/PacificBiosciences/GenomicConsensus> (2018).

25 Loman, N. J., Quick, J. & Simpson, J. T. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature methods* **12**, 733 (2015).

26 Leija-Salazar, M. *et al.* Detection of GBA missense mutations and other variants using the Oxford Nanopore MinION. *bioRxiv*, 288068 (2018).

27 Cleary, J. G. *et al.* Joint variant and de novo mutation identification on pedigrees from high-throughput sequencing data. *J Comput Biol* **21**, 405-419, doi:10.1089/cmb.2014.0029 (2014).

28 Luo, R., Lam, T.-W. & Schatz, M. Skyhawk: An Artificial Neural Network-based discriminator for reviewing clinically significant genomic variants. *bioRxiv*, 311985 (2018).

29 Sandmann, S. *et al.* Evaluating variant calling tools for non-matched next-generation sequencing data. *Scientific reports* **7**, 43169 (2017).

30 Lai, Z. *et al.* VarDict: a novel and versatile variant caller for next-generation sequencing in cancer research. *Nucleic acids research* **44**, e108-e108 (2016).

31 Wilm, A. *et al.* LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic acids research* **40**, 11189-11201 (2012).

32 Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907* (2012).

33 Li, H. Improving SNP discovery by base alignment quality. *Bioinformatics* **27**, 1157-1158, doi:10.1093/bioinformatics/btr076 (2011).

34 Li, H. *et al.* A synthetic-diploid benchmark for accurate variant-calling evaluation. *Nature methods* **15**, 595 (2018).

35 Robinson, J. T., Thorvaldsdottir, H., Wenger, A. M., Zehir, A. & Mesirov, J. P. Variant Review with the Integrative Genomics Viewer. *Cancer Res* **77**, e31-e34, doi:10.1158/0008-5472.CAN-17-0337 (2017).

36 Lu, H., Giordano, F. & Ning, Z. Oxford Nanopore MinION sequencing and genome assembly. *Genomics, proteomics & bioinformatics* **14**, 265-279 (2016).