# Clair: Exploring the limit of using a deep neural network on pileup data for germline variant calling

Ruibang Luo*, Chak-Lim Wong, Yat-Sing Wong, Chi-Ian Tang, Chi-Man Liu, Chi-Ming Leung, Tak-Wah Lam*

Department of Computer Science, The University of Hong Kong, Hong Kong, China

* Correspondence and requests for materials should be addressed to R. L. (email: rbluo@cs.hku.hk) and T. L. (twlam@cs.hku.hk)

## Abstract

Single-molecule sequencing technologies have emerged in recent years and revolutionized structural variant calling, complex genome assembly, and epigenetic mark detection. However, the lack of a highly accurate small variant caller has limited the new technologies from being more widely used. In this study, we present Clair, the successor to Clairvoyante, a program for fast and accurate germline small variant calling, using single molecule sequencing data. For ONT data, Clair achieves the best precision, recall and speed as compared to several competing programs, including Clairvoyante, Longshot and Medaka. Through studying the missed variants and benchmarking intentionally overfitted models, we found that Clair may be approaching the limit of possible accuracy for germline small variant calling using pileup data and deep neural networks. Clair requires only a conventional CPU for variant calling and is an open source project available at https://github.com/HKU-BAL/Clair.

## Introduction

Fast and accurate variant calling is essential for both research and clinical applications of human genome sequencing[1,2]. Algorithms, best practices and benchmarking guidelines have been established for how to use Illumina sequencing to call germline small variants, including single-nucleotide polymorphisms (SNPs) and insertions/deletions (indels)[3-6]. In recent years, single-molecule sequencing (SMS) technologies have emerged for a variety of important applications[7]. These technologies, which are also known as the third-generation sequencing technologies, generate sequencing reads two to three orders of magnitude longer than Illumina reads (10–100kbp versus 100–250bp). The long read length has made the new SMS technologies, including Pacific Biosciences (PacBio) and Oxford Nanopore Technology (ONT), unprecedentedly powerful for resolving complex genome assembly problems and for detecting large structural variants[8]. However, currently available SMS technologies also have a significantly higher base error rate of 3–15%[9], making the variant calling methods previously designed for Illumina sequencing inapplicable to SMS technologies. The lack of accurate tools for efficient variant calling has limited SMS technologies from being applied to the many problems that require SNPs and small indels.

In our previous work, we developed Clairvoyante[10], a germline small variant caller for single molecule sequencing data. Clairvoyante does not require sequence assembly and calls variants directly from read alignments. Clairvoyante adopts a deep convolutional neural network, so that by using the truth variants called and orthogonally verified in seven human individuals by the Genome In A Bottle (GIAB) consortium[11-13], Clairvoyante can be trained for variant calling on any new type of sequencing data without the need to look into its

error profile and build a hand-crafted model. Clairvoyante takes pileup data as input and runs quickly. However, Clairvoyante's design is unable to call multiallelic variants or indels longer than four bases. These defects remain to be solved. Meanwhile, the limit of using pileup data and deep neural networks for variant calling remains to be explored.

In this study, we present Clair, a fast and accurate system for germline small variant calling using single molecule sequencing data. With an entirely different network architecture and learning tasks (i.e. output components), Clair resolves the multiallelic and long indel variant calling problems that have prevented Clairvoyante from calling all types of small variants. We describe in detail the methods we tried that either worked or did not work for improving Clair's performance. For ONT datasets[14], our experiments on whole-genome variant calling in GIAB samples show that Clair outperforms Clairvoyante and other variant callers, including Longshot[15] and Medaka[16], in terms of precision, recall and speed. For high accuracy reads, including both PacBio CCS (Circular Consensus Sequencing)[17] and Illumina datasets[13], DeepVariant[18] had modestly improved F1-scores over Clair by .11% to .13%, although Clair was seven times faster. Looking into the false positive (FP) and false negative (FN) variants of the three sequencing technologies showed that except for variants with insufficient coverage by chance, most of the others could be resolved using complete read alignments instead of pileup data or else could not be resolved at all, even with a manual inspection.

# Results

## Overview of Clair

Clair is a four-task, five-layer recurrent neural network with two bi-directional LSTM layers followed by three feedforward layers (**Figure 1**). Clair takes a BAM file as input to find candidate variants with any minor allele frequencies larger than a threshold (typically between 0.1 and 0.2), and then computes a pileup of the candidates and converts the summaries into a tensor. In a tensor, the allelic counts of bases and gaps on both strands of a candidate variant and its 16 flanking bases are encoded into 1,056 integer values. More details and pseudo code are available in the Methods section. As discussed in the Clairvoyante paper, one major unsolved problem was how to support the calling of multi-allelic variants (i.e., variants with two alternative alleles). In Clair, the problem is solved by using four new (deep learning) tasks that are entirely different from Clairvoyante. These are: 1) a 21-genotype probabilistic model with 21 probability outputs; 2) the use of three probabilities for the input, including a homozygous reference (0/0 genotype), a heterozygous variant (0/1) or a homozygous variant (1/1); 3) the length of the first indel allele, with 33 probabilities representing a length of '<-15bp', '-15bp', '-14bp', …, '-1bp', '0bp', '1bp', …, '15bp', '>15bp'; and 4) the length of the second indel allele. The 21-genotype probabilistic model can represent all possible genotypes of a diploid sample at the genome position. The length of indels longer than 15bp cannot be directly inferred from the third and fourth tasks, so Clair includes an additional step that re-scans the alignments. More details on each of these steps can be found in the Methods section. The four tasks make their own decisions and are designed to cross-validate each other. For example, task two is a coarse-grained version of task one and can veto the decision made by task one. Tasks

three and four should indicate 0bp indel length if an SNP variant is decided by task one. More details on how the four tasks make a joint decision are available in the Methods section. We used the 'focal loss' deep-learning technique to solve the problem of unbalanced variant types in training data. We used the 'cyclical learning rate' deep learning technique to achieve the maximum possible variant calling performance and speed up the training process to be able to handle larger training datasets. To improve Clair's performance at lower sequencing coverages, we augmented the training data with 10 subsampled coverages of each dataset. The parameters of these three new techniques are in the Methods section.

Clair has 2,377,818 parameters, which is 45.7% more than Clairvoyante (1,631,496 parameters) but only one tenth as many as DeepVariant (23,885,392 parameters). In terms of variant calling speed, Clair takes about 30 minutes, 1.5 hour, and 5 hours for a 50-fold coverage WGS sample using Illumina, PacBio CCS and ONT data, respectively, using 24 CPU cores. In our experiments, Clair was 10–20% slower than Clairvoyante, but significantly faster than DeepVariant, Longshot and Medaka.

The Methods section includes a description of procedures to augment the training data or improve Clair's network architecture that we tested but that did not improve precision and recall of variant calling. Developers working on further improving Clair's performance can save time by avoiding the same methods, or the same settings in a method.

## Performance on ONT

ONT datasets are currently available for two GIAB samples, HG001 and HG002. The HG001 rel6 dataset generated by the Nanopore WGS Consortium[14] contains approximately 44.3-fold coverage of human genome (the dataset is also referred to as 1:44x, where '1' means the sample suffix and '44x' means the coverage). The rel6 dataset was base-called with Guppy 2.3.8, using the HAC (High-ACcuracy) model. In addition to the rel6 dataset, we obtained a separate 124.1-fold coverage dataset for HG001 (1:124x) directly from Oxford Nanopore (Philipp Rescheneder, personal communication). That dataset was base-called with Guppy 2.2.3 using the Flip-Flop model. In some experiments, we combined 1:44x and 1:124x to form a new dataset 1:168x to maximize the coverage. For HG002, we used a dataset with ~64-fold coverage (2:64x) from the GIAB consortium, which was base-called with Guppy 2.3.5 using the Flip-Flop model. The links to the datasets are available in the Supplementary Notes. The details about "the GIAB truth variant datasets", "removing GA4GH (The Global Alliance for Genomics and Health) low-complexity regions[6] from benchmarking", and "the benchmarking methods and metrics" are available in "Methods – Benchmarking".

**Figure 2** shows the precision and recall of Clair and other variant callers on SNPs and indels in multiple experiments with ONT data. Supplementary Table 1 contains more details, including precision, recall and F1-score in five categories, including overall, SNP, indel, insertion, and deletion. Our results show that Clair not only outperformed other variant callers, including Clairvoyante, Longshot, and Medaka, but also ran much faster. Using 1:168x|2:64x (i.e., test variant calling using HG002 with 64-fold coverage against a model trained using HG001 with 168-fold coverage) as Clair's primary result, Clair achieved 98.36%

138 precision, 96.46% recall, and 97.40% F1-score overall performance. In terms of SNPs, the
139 three metrics were 99.29%, 97.78% and 98.53%, respectively. For indels, they were
140 somewhat lower at 81.15%, 73.88%, and 77.34%. Clair significantly outperformed its
141 predecessor Clairvoyante on both SNP and indel calling (overall F1-score 97.40% versus
142 93.45%). Clair had a slightly higher F1-score on SNPs than Longshot (98.53% versus 98.41%),
143 but Longshot detects only SNPs, and Clair ran five times faster than Longshot (320 versus
144 1,797 minutes). Clair had a better performance than Medaka (overall F1-score 97.40%
145 versus 94.81%) and ran 30 times faster (320 versus 10,817 minutes). It is worth mentioning
146 that we didn't benchmark Nanopolish[19], which is also capable of variant calling on ONT data,
147 because it also requires raw signals as input, which are not publicly available for HG002.
148
149 We ran further experiments to answer five additional questions about Clair, as follows.
150
151 **Is the Clair model reference-genome specific?** In our experiments, performance did not
152 depend on whether we used GRCh37 or GRCh38. The performance of 1:168x|2:64x and
153 1:168x|2:64x(b37) was similar; the latter experiment tested HG002 GRCh37 read alignments
154 on a model trained using HG001 GRCh38 read alignments. Actually, 1:168x|2:64x(b37)
155 performed slightly better than 1:168x|2:64x, with a 0.18% better F1-score on SNPs, and
156 1.4% on indels.
157
158 **Does higher coverage in the test sample helps improve variant calling performance?** Yes,
159 but improvement seems to asymptote at ~60-fold coverage. In a comparison of
160 1:168x|2:64x to 1:168x|2:32x, the overall F1-score increased from 94.10% to 97.40%
161 (+3.30%), the SNP from 95.51% to 98.53% (+3.02%), and the indel from 68.87% to 77.34%

(+8.47%). Further increasing the coverage in the test sample will note significantly increase the variant calling performance as we discuss below.

**Does higher coverage for model training help improve variant calling performance?** Yes, but it depends on the coverage of the test sample. In a comparison of 1:124x|2:64x to 1:44x|2:64x, the overall F1-score increased from 96.84% to 97.51% (+0.67%), the SNP from 98.01% to 98.54% (+0.53%), and the indel from 75.78% to 78.44% (+2.66%). In a comparison of 1:168x|2:64x to 1:124x|2:64x, the performance was similar, or even slightly dropped from 97.51% to 97.40% overall. One possible reason is that the lower coverage test sample cannot benefit from the much higher coverage used for model training. We propose how to deal with excessively high coverage in test samples (i.e., coverage exceeding that used in model training) in the Discussion section below.

**Does multiple subsampled coverage for model training improved variant calling performance?** Yes. in a comparison of 1:44x|2:64x to '1:44x (single cov.)|2:64x', the latter used only the full coverage 44-fold in model training; the overall F1-score increased from 95.47% to 96.84% (+1.37%), the SNP from 96.94% to 98.01% (+1.07%), and the indel from 75.78% to 78.44% (+2.86%). The results show that even without sufficient coverage for model training, using multiple subsampled coverage still improved the variant calling performance significantly.

**What is the upper bound on performance?**

To determine Clair's performance cap using the current ONT data, we intentionally overfitted Clair by adding the samples we are going to test to the model training. Even

186 though Clair is designed with multiple generalization techniques, including ‘dropout’ and ‘L2
187 regularization’, exposing the test samples to model training is a biased evaluation, and if a
188 true variant is not called even after this biased training, this suggests the input signal is
189 simply too weak. The two tests we did were 1:168x+2:64x|2:64x and 1:168x+2:64x|1:168x.
190 Although the test sample coverage in the first test was much lower than that in the second
191 (64-fold against 168-fold), their performance was similar, with the overall F1-score at
192 97.77% and 97.82%, SNP at 98.75% and 98.77%, and indel at 79.92% and 81.37%. The
193 biased test 1:168x+2:64x|2:64x did not significantly outperform 1:168x|2:64x; the overall
194 F1-score increased from 97.40% to 97.77% (+0.33%), SNP from 98.53% to 98.75% (+0.22%),
195 and indel from 77.34% to 79.92% (+2.58%). Even with this biased experiment, we observed
196 that the performance of using Clair on the current ONT data was capped at about 97.8% F1-
197 score overall, 98.8% on SNPs, and 80% on indels. We consider how the new ONT chemistry
198 that provides a lower base error rate can raise the upper bound of Clair’s variant calling
199 performance in the Discussion section below.
200
201 We analyzed and categorized the FP and FN results of Clair on ONT data. We randomly
202 extracted 100 FPs and 100 FNs from the 1:168x|2:64x experiment. **Figure 3** shows a
203 summary and examples of different categories, and Supplementary Table 2 shows a detailed
204 analysis of each FP and FN. Within the 100 FPs, the three largest categories are "Incorrect
205 allele with AF≥0.2" (41/100), "Homopolymer" (25/100), and "Tandem repeat" (11/100).
206 "Incorrect allele with AF≥0.2" means that at the FP variant, an incorrect allele dominates
207 other alleles in the read alignments (including the correct one), and the incorrect allele has a
208 frequency ≥20%. "Homopolymer", "Tandem repeat", and "Low complexity region" mean
209 that the FP variant is in a repetitive region, which remains difficult for ONT base-calling. It is

210 worth mentioning that these repetitive regions are ≤10bp because we removed all GA4GH
211 low-complexity regions longer than 10bp from benchmarking. It may not be possible to
212 perfectly resolve these three categories for FP variants using pileup data for variant calling,
213 although complete read alignments might help to provide better precision. Three out of 100
214 FPs had "Incorrect insertion bases", while two out of 100 were categorized as "Overlapping
215 insertions", which means that the alleles of two consecutive insertions overlapped each
216 other in an input tensor; thus, the correct allele cannot be resolved for both insertions.
217 These two categories of errors can be resolved using the '--pysam_for_all_indel' option in
218 Clair, but this slows down Clair for ONT data by a factor of up to ten times. Other errors,
219 including "Incorrect indel length" and "Incorrect zygosity", are errors made by Clair's neural
220 network. In the 100 FNs, the three major categories are "Correct allele with AF<0.25"
221 (54/100), "Homopolymer" (18/100), and "Tandem repeat" (7/100). "Correct allele with
222 AF<0.25" means that at the location of the missed (FN) variant, the signal of the correct
223 allele is rather weak, with allele frequency lower than 25%. One FN categorized as "More
224 than two possible alternative alleles" is an error due to an alignment error in segmental
225 duplications, in which more than two alternative alleles seem correct.
226

227 ## Performance on PacBio CCS

228 In early 2019[17], PacBio developed a protocol based on single-molecule, circular consensus
229 sequencing (CCS) to generate highly accurate (99.8%) long reads averaging as much as
230 13.5kb. PacBio published CCS datasets for HG001 (in this section also referred to as 1:30x; 1
231 as the sample suffix and 30x means 30-fold coverage), HG002 (2:33x) and HG005 (5:33x). All
232 three samples are involved in model training. To demonstrate a possible overfitting

233 phenomenon on deep learning based variant callers, both HG002 and HG005 are used in
234 benchmarking.
235
236 Supplementary Table 3 shows the results of Clair and three other variant callers:
237 Clairvoyante, Longshot, and DeepVariant. Testing on HG002, DeepVariant performed the
238 best, with an overall F1-score of 99.96%, SNP of 99.97%, and indel of 99.92%. The primary
239 result of Clair 1:30x+5:33x|2:33x had an overall F1-score of 99.83%, which was 0.13% lower
240 than DeepVariant, but outperformed both Clairvoyante and Longshot. On SNP,
241 1:30x+5:33x|2:33x had an F1-score of 99.88%, which was 0.09% lower than DeepVariant,
242 0.43% higher than Longshot, and 0.17% higher than Clairvoyante. On indel,
243 1:30x+5:33x|2:33x had an F1-score at 99.07%, which was 0.85% lower than DeepVariant,
244 but 19.17% higher than Clairvoyante, showing that the new methods applied to Clair have
245 effective solved the indel-calling problem in Clairvoyante. In terms of speed, Clair (147
246 minutes) is slightly faster than Longshot (206 minutes), and about seven times faster than
247 DeepVariant (1,072 minutes). We also tested HG005. Interestingly, while Clair, Clairvoyante,
248 and Longshot all performed better on HG005 than HG002, DeepVariant performed worse.
249 Comparing 1:30x|2:33x to 1:30x|5:33x, Clair's overall F1-score increased from 99.77% to
250 99.80%. Clairvoyante's overall F1-score increased from 98.61% to 98.70%. Longshot's SNP
251 F1-score increased from 99.45% to 99.46%. The performance of the three callers verifies the
252 quality of the HG005 dataset. However, DeepVariant's F1-score dropped from 99.96% to
253 99.92%, the SNP F1-score decreased from 99.97% to 99.93%, and the indel F1-score
254 dropped most significantly, from 99.92% to 99.78%. The most probable reason is that,
255 DeepVariant's current PacBio CCS model was trained completely using HG002[20]. We suggest
256 using DeepVariant's result on HG005 as its real performance on PacBio CCS data. The biased

257 test 1:30x+2:33x+5:33x|2:33x found the performance cap of Clair at 99.88% on SNP, which
258 was the same as 1:30x+5:33x|2:33x, and 99.28% on indel, which was 0.21% higher than
259 1:30x+5:33x|2:33x. While in 1:30x+5:33x|2:33x, the highest coverage used for model
260 training was only 33x, we expect to fill the performance gap on indel calling by using higher
261 coverage for model training. The performance gap between Clair and DeepVariant on
262 HG005 (99.28% against 99.78%, -0.5%) is the result of Clair using pileup data, while
263 DeepVariant uses complete read alignments that contain information at a per-read level.
264 This is also a reason DeepVariant runs slower than Clair. We discuss the possibility of
265 improving Clair to use complete read alignments without slowing down performance in the
266 Discussion section below.
267

268 ## Performance on Illumina

269 Approximately 300x coverage in 148-bp Illumina paired-end read data is available for five
270 GIAB samples, including HG001, HG002, HG003, HG004 and HG005[11]. We used HG001,
271 HG003, HG004, HG005 for model training, and HG002 for benchmarking. To resemble the
272 typical coverage in whole genome sequencing, we used full coverage of HG001 (306-fold)
273 and HG005 (352-fold), but down-sampled HG002, HG003 and HG004 to 52-, 57-, and 66-
274 fold.
275

276 Supplementary Table 4 shows the results of Clair and DeepVariant. DeepVariant performed
277 better, with an overall F1-score of 99.94%. The primary result of Clair
278 1:306x+3:57x+4:66x+5:352x|2:52x was an overall F1-score of 99.83%, which was 0.11%
279 lower than DeepVariant's. For SNPs, the F1-score of Clair was 0.09% lower than that of
280 DeepVariant (99.85% versus 99.94%). For Indel, the F1-score of Clair was 0.42% lower than

DeepVariant's (99.48% versus 99.90%). In terms of speed, Clair was about seven times faster than DeepVariant (77 versus 537 minutes). The biased test 1:306x+2:52x+3:57x+4:66x+5:352x|2:52x found the performance cap of Clair to be 99.87% for SNPs, which was 0.02% higher than the primary result, but 0.07% lower than that of DeepVariant, and 99.57% for indels, which was 0.09% higher than the primary result, but 0.33% lower than that of DeepVariant. Similar to the ONT and PacBio CCS experiments, we expect to fill in the performance gap through partially making use of complete read alignments, as discussed in the Discussion section.

## Discussion

In this paper we present Clair, a germline small variant caller for single molecule sequencing data. The name Clair means 'clear' in French, echoing its predecessor, named Clairvoyante, meaning 'clear seeing'. Clair adds new methods to solve problems that Clairvoyante had trouble with, including multiallelic variant calling and long indel calling. In our experiments on ONT data, Clair outperformed all existing tools in terms of precision, recall and speed. On PacBio CCS and Illumina data, Clair performed slightly worse than DeepVariant, but ran about an order of magnitude faster. Looking closer at the FP and FN variants shows that Clair is approaching the limit on how accurately it can call variants using pileup data. Some of the erroneous variant calls can be corrected using complete read alignments instead of pileup data. However, dealing with complete read alignments requires a more powerful neural network design with much greater computational demands. In the future, we will explore using an ensemble method to handle the majority of the variants using Clair, while for the extremely tricky ones we will use a new, more sophisticated method.

304 The quality and sufficiency of training data is key to the performance of Clair, as well as
305 other deep learning based variant callers, such as DeepVariant. To train a model for
306 production purposes, we used five samples (HG001 to 5) for Illumina data, but only two
307 samples (HG001 and HG002) for ONT, due to the limited availability of public high-coverage
308 whole genome sequencing datasets for the GIAB samples. ONT sequencing of the other
309 GIAB samples is ongoing, and more data will be available in the near future. With additional
310 datasets, we expect to see even higher performance in Clair on ONT data.
311
312 On ONT data, although Clair performed the best, its indel calling precision and recall were
313 only about 80%, even excluding GA4GH low-complexity regions, which leaves substantial
314 room for improvement. While the precision can be further improved by considering
315 complete read alignments, the recall is bounded by input and can be improved only with a
316 lower read-level base-calling error rate. Future improvements in ONT technology offer the
317 possibility of reducing the error rate to 2-3%, which in turn should improve Clair's ability to
318 detect indels in these data.
319
320 The GIAB datasets we used for model training have moderate whole-genome sequencing
321 coverage. Although we can use samples with very high coverage (over 300-fold, which is
322 sometimes seen in amplicon sequenced data) with Clair for variant calling, such samples
323 might show degraded performance because very high coverage variants were not
324 adequately observed in model training. To solve this problem, we propose two methods.
325 One method is to do transfer learning using a trained model on additional datasets with
326 very high coverage. Clair supports transfer learning and can be applied to additional
327 datasets instantly. Another method is an ensemble method, which generates multiple

copies of randomly subsampled read alignments at a candidate variant for Clair to call variant. A majority vote or a decision tree can be used to make the final decision, using the results of each copy.

A limitation of Clair is that it cannot be applied to polyploid species, which are inconsistent with its neural network design. For the same reasons, Clair is not applicable to somatic variant calling, where a single sample might hold multiple distinct populations of cells. Our next steps include extending Clair to support polyploid species and somatic variant calling.

# Method

## Clair's input/output

### Input

For a truth variant for training or a candidate variant for calling, the read alignments that overlap or are adjacent to the variant are summarized (i.e. pile-up data) into a three-dimensional tensor of shape 33 by 8 by 4, comprising 1056 integer numbers. The three dimensions correspond to the position, the count of four possible bases from two different strands, and four different ways of counting. In the first dimension, 33 positions include the starting position of a variant at the center and 16 flanking bases on both sides. The second dimension corresponds to the count of 'A+', 'A-', 'C+', 'C-', 'G+', 'G-', 'T+' or 'T-', with the symbols +/- denoting the count from the forward/reverse strand. The third dimension replicates the first two dimensions with four different ways of counting to highlight 1) the allelic count of the reference allele, 2) insertions, 3) deletions and 4) single nucleotide alternative alleles. "Supplementary Note – Pseudocode for generating the input tensor"

351 shows the pseudo code of the exact algorithm of how the input tensor is generated.
352 Supplementary Figure 1 demonstrates how the tensors are look like for ONT data at a
353 random ‘non-variant’, a ‘SNP’, an ‘Insertion’, and a ‘Deletion’.
354

355 ## Output

356 The output of Clair has four tasks (a.k.a. four output components, in total 90 probabilities),
357 including 1) the 21-genotype probabilistic model (21 probabilities); 2) zygosity (3
358 probabilities); 3) the length of the first indel allele (33 probabilities); and 4) the length of the
359 second indel allele (33 probabilities). One of the breakthroughs in Clair is the invention of
360 the 21-genotype probabilistic model. It comprises all of the possible genotypes of a diploid
361 sample at a genome position, including 'AA', 'AC', 'AG', 'AT', 'CC', 'CG', 'CT', 'GG', 'GT', 'TT',
362 'AI', 'CI', 'GI', 'TI', 'AD', 'CD', 'GD', 'TD', 'II', 'DD', and 'ID', where 'A', 'C', 'G', 'T', 'I' (insertion)
363 and 'D' (deletion) denote the six possible alleles. The new model covers variants with two
364 alternative alleles, which could not be called in Clairvoyante. The zygosity task outputs the
365 probability of the input being 1) a homozygous reference (0/0); 2) heterozygous with 1 or 2
366 alternative alleles (0/1 or 1/2); or 3) a homozygous variant (1/1). The zygosity task is
367 partially redundant to the 21-genotype task, but it makes decisions independently, and it
368 crosschecks the decision made by the 21-genotype task. Tasks three and four have the same
369 design. They output the length of up to two indel alleles. Each task outputs 33 probabilities,
370 including the likelihood of 1) more than 15bp deleted (<-15bp); 2) any number between -
371 15bp and 15bp, including 0bp, and; 3) more than 15bp inserted (>15bp). In training, the
372 indel allele with a smaller number is set as the first indel allele. For example, for a
373 heterozygous 1bp deletion, the first indel allele is set as -1bp, the second as 0bp (-1bp/0bp).
374 For a heterozygous 1bp insertion, 0bp/1bp is set. This design makes the non-0bp training

375 variants for both tasks balanced. For a heterozygous indel with two alternative alleles, say,
376 one -2bp and one 5bp, -2bp/5bp are set. For a homozygous indel, two indel alleles are set to
377 the same value. For indels longer than 15bp, the exact length is determined using an
378 additional step (Supplementary Note – New methods used in Clair – Dealing with indels
379 longer than 15bp). The output of the two indel allele tasks are also used for crosschecking
380 with the 21-genotype task, with 0bp supporting an SNP allele, and non-0bp supporting an
381 indel allele. More details about how the four tasks crosscheck each other to come up with a
382 result coherently are in "Method – New methods used in Clair – Determining the most
383 probable variant type using the four tasks of Clair".
384

385 ## New methods used in Clair

386 Clair has been fully revamped while a few basic deep-learning techniques in Clairvoyante
387 have been retained, including 1) model initialization; 2) activation function; 3) optimizer; 4)
388 dropout; 5) regularization; and 6) combining multiple samples for model training. Below we
389 discussed the new methods we have applied in Clair.
390

391 ### Dealing with indels longer than 15bp

392 For each candidate variant, Clair directly outputs the length of up to two alternative indel
393 alleles. However, if an insertion goes beyond 15bp, or a deletion goes below -15bp, Clair
394 runs an additional step to decide its exact length and allele. In the additional step, Clair
395 gathers all possible insertion/deletion alleles longer than 15bp at a genome position
396 through pysam (a wrapper around htslib and the samtools[21] package). Depending on the
397 genotype concluded by Clair, we choose 1) the insertion/deletion with the highest allelic
398 count for 'AI', 'CI', 'GI', 'TI', 'AD', 'CD', 'GD' and 'TD'; 2) the insertions with the highest and/or

399 the second-highest allelic count for 'II'; 3) the deletions with highest and/or the second-
400 highest allelic count for 'DD', or; 4) both the insertion and deletion with the highest allelic
401 count for 'ID'. The additional step is slow, but it is required only for indels longer than 15bp.
402 We investigated HG001 and found 570,367 indels in its truth variant set; only 10,672
403 (1.87%) were >15bp. In our experiments, we found the slowdown was acceptable. Users can
404 set an option in Clair to enable this additional step for all indels, but our experiments found
405 that while the improvement in precision is small, it slows down Clair by about two times
406 with Illumina and PacBio CCS data, and by more than 10 times on ONT data.
407

408 ### Determining the most probable variant type using the four Clair tasks

409 Clair outputs data on four tasks. With an independent penultimate layer (Figure 1, FC5
410 layer) immediately before each task, the output of each task is considered independent. We
411 made two observations from our experiments: 1) for true positive variants, a random task
412 or two will make a mistake occasionally, but usually, the best and the second-best
413 probabilities are near and can be disambiguated if considered with other tasks; 2) for false
414 positive variants, the tasks do not usually agree well with each other, leading to two or
415 more possible decisions with similar probabilities. Thus, in Clair, we implemented a method
416 as a submodule for making a decision using the output of all four tasks. Variants are divided
417 into 10 categories: 1) a homozygous reference allele; 2) a homozygous 1 SNP allele; 3) a
418 heterozygous 1 SNP allele, or heterozygous 2 SNP alleles; 4) a homozygous 1 insertion allele;
419 5) a heterozygous 1 insertion allele, or heterozygous 1 SNP and 1 insertion alleles; 6)
420 heterozygous 2 insertion alleles; 7) a homozygous 1 deletion allele; 8) a heterozygous 1
421 deletion allele, or heterozygous 1 SNP and 1 deletion alleles; 9) heterozygous 2 deletion
422 alleles; and 10) a heterozygous 1 insertion and 1 deletion alleles. The likelihood value of the

10 categories is calculated for each candidate variant, and the category with the largest likelihood value is chosen (Pseudocode in "Supplementary Note – Pseudo code for determining the most probable variant type"). The variant quality is calculated as the square of the Phred score of the distance between the largest and the second-largest likelihood values.

### Cyclical learning rate

The "initial learning rate" and "how the learning rate decays" are two critical hyperparameters in training a deep neural network model. A model might be stuck at a local optimum (i.e. unable to achieve the best precision and recall) if the initial learning rate is too large, or the decay is too fast. But a large initial learning rate, and a slow decay rate make the training process either unstable or take too long to finish. So in common practice, a tediously long grid search that is very costly is needed to find the best hyperparameters. Furthermore, through a grid search, we found that different sequencing technologies differ in their best hyperparameters. This problem makes model training too complicated and largely impedes Clair from being applied to new datasets and sequencing technologies. To solve the problem, we implemented Cyclical Learning Rate (CLR)[22] in Clair. CLR is a new deep learning technique that eliminates the need to find the best values of the two hyperparameters. CLR gives a way to schedule the learning rate in an efficient way during training, by cyclically varying between a lower and higher threshold. Following the CLR paper, we determined the higher threshold to be 0.03 and the lower threshold to be 0.0001. The two thresholds worked well on the training variants of all three sequencing technologies (Illumina, PacBio CCS and ONT). In terms of which CLR scheduler to use, we chose the triangular schedule with exponential decay. In our experiments, on PacBio CCS

447 and Illumina datasets, CLR decreased model training time by about 1–3 times, while often
448 outperforming the three-step decay method introduced in Clairvoyante for both precision
449 and recall. However, on ONT datasets, CLR has a lower, but almost negligible, performance
450 than the three-step decay. We provide both CLR and three-step decay options in Clair. To
451 train a model for production, we suggest users try both options and choose the best
452 through benchmarking. In our results, we used CLR for PacBio CCS and Illumina datasets,
453 and the three-step decay method for ONT datasets.
454

455 ## Focal loss

456 Our training data uses the truth variants from the GIAB consortium and is unbalanced in
457 terms of variant type. For example, the number of heterozygous variants is nearly twice that
458 of the homozygous variants. SNPs are about five times more numerous than indels. Worst
459 of all, only ~1.1% (39,898 of 3,619,471 in HG001) of variants have two or more alternative
460 alleles. And among them, only 884 (~0.024%) are multiallelic SNPs. This problem leads to
461 degenerate models, as the numerous easy variants contribute no useful learning signals and
462 overwhelm training. In our practice, if we leave the problem unaddressed, we observe a
463 significant drop in recall for the underrepresented variant types. For multiallelic SNPs, the
464 recall dropped to zero. To solve this problem, we used the "Focal loss" technique[23], which
465 applies a modulating term to the cross-entropy loss in Clair's output to focus training on
466 underrepresented hard variants and down-weight the numerous easy variants. Focal loss
467 calculates the loss as $(1-p_t)^{\gamma} \times \alpha_t \times -\log(p_t)$, where $p_t = p$, $\alpha_t = \alpha$, if the prediction
468 matches the truth, or $p_t = (1-p)$, $\alpha_t = (1-\alpha)$ otherwise. In addition to the traditional
469 cross entropy loss, focal loss uses two more parameters: $\gamma$ (the focusing parameter) to
470 differentiate easy/hard training examples, and $\alpha$ (the balancing parameter) to balance the

importance of positive/negative training examples. We determined $\gamma = 2$ and $\alpha = 0.25$ work best for the GIAB truth variants with a 1:2 ratio of truth variant and non-variant. The use of focal loss significantly increases the performance of underrepresented variant types. It also allows us to be more lenient on variant type balance when augmenting the training data.

### Training data augmentation using subsampled coverage

Lower coverage usually leads to lower precision and recall in variant calling. To train Clair to achieve better performance on variants with lower coverages, we subsampled each dataset into four or nine additional datasets with lower coverages. The subsampling factors *f* are determined as $(\sqrt[h]{4 \div c})^n$, where *c* is full coverage of each sample, 4 is the minimal coverage, $h$ is either 4 or 9, and $n$ is from 1 to $h$. Using HG002 as an example, its full coverage is 63.68-fold, and the nine subsampled coverages are 46.82-, 34.43-, 25.31-, 18.61-, 13.69-, 10.06-, 7.40-, 5.44- and 4.00-fold. If variant samples were lower than 4x after subsampling, we removed them from training. We used the command "samtools view -s *f*" to generate a subsampled BAM. A different seed counting from zero for random number generation was set for each coverage. The use of subsampled coverages improved the recall on indel significantly.

### Methods tested but showed no improvement to accuracy

In this section we discuss methods we tested that had no effect on Clair's performance. For researchers working on further improving the performance of Clair, these methods could be avoided or revised.

495 ### Extend input tensor from 33bp to 49bp and 65bp

496 Intuitively, a larger input tensor with more flanking bases provides additional information
497 on the surrounding read alignments, which might lead to better precision and recall. Our
498 experiments show that extending the input tensor from 33bp (16bp flanking bases) to 49bp
499 (24bp flanking bases) and 65bp (32bp flanking bases) slows down Clair by 5.4% and 12.6%,
500 respectively. But the improvement was negligible in terms of precision or recall with both
501 SNP and indel.

502

503 ### Using non-variants adjacent to true variants as negative samples for model training

504 Clair, by default, uses a ratio of 1:2 on true variants and non-variants for model training, and
505 the non-variants are randomly selected from the genome, except for the positions with a
506 true variant or insufficient coverage. We experimented using non-variants adjacent to true
507 variants (we tried ±2bp, ±8bp and ±16bp) as negative samples for model training and
508 adjusted the ratio to 1:1:1 on true variants adjacent non-variants and random non-variants.
509 We used adjacent non-variants for training because their input is true variant alike, but a
510 few bases shifted. The hypothesis was that using them as adversarial training samples
511 against the true variants might improve Clair's performance at high density variants and
512 alignment errors. However, our experiments show that the method decreased recall slightly
513 on both SNP and indel.

514

515 ### Incorporating less confident GIAB variants for model training

516 The GIAB HG001 truth variant dataset includes 3,619,471 truth variants passing all criteria
517 (with the 'PASS' tag), and 2,264,796 variants failing one or more criteria. The criteria details
518 were explained by Zook et al. in 2019[13]. Among the failed variants, 310,113 had the

519 ‘allfilteredbutagree’ tag, which means at the same position, the variants called in all the
520 supporting datasets agreed with each other, even though none of them were in the callable
521 regions, in which a range of coverage and minimum alignment quality are met. These
522 variants are considered less confident than those passing all criteria, but might still
523 contribute to training a better model because while a deep neural network can tolerate
524 moderate errors in training data, if any new patterns are provided in additional data, it will
525 be learned by the model and, in turn, improve the performance. We experimented adding
526 the variants with the ‘allfilteredbutagree’ tag to training. However, our results show that the
527 recall went down significantly on SNP, and the precision went down significantly on indel.
528

529 ### Discarding homopolymer variants in model training

530 Variant calling in homopolymer sequences is usually more challenging, and the problem is
531 even worse in SMS technologies since the length of homopolymers is usually
532 underestimated. At longer homopolymers, the signals are usually too discordant, so it is
533 common for humans to make mistakes with them. From the feature engineering point of
534 view, variants in homopolymer sequences are confusing and less informative, and might
535 lead to a degenerate model. We tested model training without variants at homopolymer
536 sequences longer than 5bp. Our results show that both precision and recall degrade
537 significantly if homopolymer variants are not used in model training.
538

539 ## Benchmarking

540 ### The GIAB truth variant datasets

541 We used the GIAB version 3.3.2 datasets as our truth variants. Depending on the availability
542 of deep sequencing data, our ONT experiments used samples HG001 or HG001+HG002 for

543 model training, our PacBio CCS experiments used HG001 or HG001+HG005, and our Illumina
544 experiments used HG001 or HG001+HG003+HG004+HG005. For benchmarking, ONT, PacBio
545 CCS and Illumina experiments have used HG002, HG005, and HG002, respectively. The links
546 to the truth variants and high-confidence regions are available in “Methods – Data sources –
547 Truth variants”. Depending on the reference genome used in the already available read
548 alignments, we used GRCh38 for our ONT and Illumina experiments, and GRCh37 for our
549 PacBio CCS experiments. The links to the reference genomes we used are available in
550 “Methods – Data sources – Reference genomes”
551

552 ### Removing GA4GH low-complexity regions from benchmarking

553 Krusche et al.[6] from the GA4GH benchmarking team and the GIAB consortium published the
554 low-complexity regions, including homopolymers, STRs, VNTRs, and other repetitive
555 sequences for stratifying variants in their paper titled "Best practices for benchmarking
556 germline small-variant calls in human genomes". In the low-complexity regions larger than
557 10bp, ONT's performance degraded significantly (precision -11.41%, recall -55.33%), while
558 that of PacBio CCS and Illumina dropped only 0.99–1.67% in precision and recall
559 (Supplementary Table 5). Thus, when computing variant calling using ONT, we suggest
560 removing the variants called in the low-complexity regions. In our benchmarks for all
561 datasets, in addition to using the high-confidence regions of each sample provided by GIAB,
562 we removed the low-complexity regions. The procedures are available in "Supplementary
563 Note – Commands – Remove GA4GH low complexity regions from GIAB's high-confidence
564 regions". There was retention of 92.61–93.47% high-confidence regions in GRCh38, and
565 94.40–95.05% in GRCh37 of the five samples HG001 to 5 after removing the low-complexity
566 regions (Supplementary Table 8).

567

568 ## Benchmarking methods and metrics

569 Clair trains a model either for 30 epochs, using the Cyclical Learning Rate (used for PacBio
570 CCS and Illumina datasets), or by decaying the learning rate three times (by one tenth each
571 time) until the validation losses converge (used for ONT datasets). While the performance of
572 last few epochs are generally similar, the best-performing one will be chosen for
573 benchmarking. We did not run replications of model training because choosing from the
574 best epoch actually resembles the process of having multiple replications. In ONT and
575 Illumina experiments, the GRCh38 reference genome was used, while in PacBio CCS
576 experiments, GRCh37 was used. For each variant calling experiment, we used the
577 submodule vcfeval in RTG Tools[24] version 3.9 to generate three metrics, 'Precision', 'Recall',
578 and 'F1-score', for five categories of variants: 'Overall', 'SNP', 'Indel', 'Insertion', and
579 'Deletion'. All time consumptions were gauged on two 12-core Intel Xeon Silver 4116 (in
580 total 24 cores), with 12 concurrent Clair processes, each with 4 Tensorflow threads. As Clair
581 has some serial steps that use only one thread, we observed our setting sufficient to
582 maximize the utilization of all 24 cores. For other variant callers, including DeepVariant,
583 Longshot and Medaka, options were to set to use all 24 cores for the best speed.

584

585 ## Computational performance

586 Clair requires Python3, Pypy3 and Tensorflow. Variant calling using Clair requires only a
587 CPU. For a typical 30-fold human WGS sample, Clair takes about an hour for Illumina data
588 and PacBio CCS data, and five hours on ONT data, using two 12-core Intel Xeon Silver 4116
589 processors. Memory consumption depends on both input data and concurrency. ONT data
590 has a higher memory footprint than Illumina and PacBio CSS, while Clair is capped at 7GB

per process (helper scripts at 4.5GB and Tensorflow at 2.5GB). Model training requires a high-end GPU; we used the Nvidia Titan RTX 24GB in our experiment. Using Clair's default parameters, generating 1 million training samples takes about 38 seconds. For example, the Illumina model with four samples (HG001, 3, 4, 5) and 30 coverages in total (10 for 1 and 5, 5 for 2 and 3) has 284,367,735 training samples and takes about 11,000 seconds per epoch. In comparison, the Nvidia RTX 2080 Ti 11GB is about 15% slower, and the Nvidia GTX 1080 Ti 11GB is about 35% slower.

### Code availability

Clair is open source, available at https://github.com/HKU-BAL/Clair.

### Data availability

The authors declare that all data supporting the findings of this study are available at the links in the paper and its supplementary information files.

## Acknowledgements

We thank Steven Salzberg, Mike Schatz, and Fritz Sedlazeck for their valuable comments. R. L. was supported by the ECS (Grant No. 27204518) of the HKSAR government, and the URC fund at HKU. T. L., C. W., Y. W., C. T., C. Li. and C. Le. were supported by the ITF (Grant No. ITF/331/17FP) from the Innovation and Technology Commission, HKSAR government.

## Author contributions

R. L. and T. L. conceived the study. R. L, C. W., Y. W., C. T., C. Li. and C. Le. analyzed the data and wrote the paper.

## Competing interests

The authors declare no competing interests

616 ## References


617 1 Goodwin, S., McPherson, J. D. & McCombie, W. R. Coming of age: ten years of next-
618 generation sequencing technologies. *Nat Rev Genet* **17**, 333-351,
619 doi:10.1038/nrg.2016.49 (2016).
620 2 Ashley, E. A. Towards precision medicine. *Nat Rev Genet* **17**, 507-522,
621 doi:10.1038/nrg.2016.86 (2016).
622 3 Li, H. Toward better understanding of artifacts in variant calling from high-coverage
623 samples. *Bioinformatics* **30**, 2843-2851, doi:10.1093/bioinformatics/btu356 (2014).
624 4 Luo, R., Schatz, M. C. & Salzberg, S. L. 16GT: a fast and sensitive variant caller using a
625 16-genotype probabilistic model. *GigaScience* (2017).
626 5 Van der Auwera, G. A. *et al.* From FastQ data to high confidence variant calls: the
627 Genome Analysis Toolkit best practices pipeline. *Curr Protoc Bioinformatics* **43**, 11 10
628 11-33, doi:10.1002/0471250953.bi1110s43 (2013).
629 6 Krusche, P. *et al.* Best practices for benchmarking germline small-variant calls in
630 human genomes. *Nat Biotechnol* **37**, 555-560, doi:10.1038/s41587-019-0054-x
631 (2019).
632 7 The long view on sequencing. *Nat Biotechnol* **36**, 287, doi:10.1038/nbt.4125 (2018).
633 8 Sedlazeck, F. J., Lee, H., Darby, C. A. & Schatz, M. C. Piercing the dark matter:
634 bioinformatics of long-range sequencing and mapping. *Nat Rev Genet*,
635 doi:10.1038/s41576-018-0003-4 (2018).
636 9 Ameur, A., Kloosterman, W. P. & Hestand, M. S. Single-Molecule Sequencing:
637 Towards Clinical Applications. *Trends Biotechnol* **37**, 72-85,
638 doi:10.1016/j.tibtech.2018.07.013 (2019).
639 10 Luo, R., Sedlazeck, F. J., Lam, T. W. & Schatz, M. C. A multi-task convolutional deep
640 neural network for variant calling in single molecule sequencing. *Nat Commun* **10**,
641 998, doi:10.1038/s41467-019-09025-z (2019).
642 11 Zook, J. M. *et al.* Extensive sequencing of seven human genomes to characterize
643 benchmark reference materials. *Sci Data* **3**, 160025, doi:10.1038/sdata.2016.25
644 (2016).
645 12 Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of
646 benchmark SNP and indel genotype calls. *Nat Biotechnol* **32**, 246-251,
647 doi:10.1038/nbt.2835 (2014).
648 13 Zook, J. M. *et al.* An open resource for accurately benchmarking small variant and
649 reference calls. *Nature Biotechnology* **37**, 561-566, doi:10.1038/s41587-019-0074-6
650 (2019).
651 14 Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-
652 long reads. *Nat Biotechnol* **36**, 338-345, doi:10.1038/nbt.4060 (2018).
653 15 Edge, P. & Bansal, V. Longshot enables accurate variant calling in diploid genomes
654 from single-molecule long read sequencing. *Nat Commun* **10**, 4660,
655 doi:10.1038/s41467-019-12493-y (2019).
656 16 *medaka: Sequence correction provided by ONT Research*.
657 https://github.com/nanoporetech/medaka, accessed Nov 17 2019.
658 17 Wenger, A. M. *et al.* Accurate circular consensus long-read sequencing improves
659 variant detection and assembly of a human genome. *Nat Biotechnol* **37**, 1155-1162,
660 doi:10.1038/s41587-019-0217-9 (2019).

18 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).
19 Simpson, J. T. *et al.* Detecting DNA cytosine methylation using nanopore sequencing. *Nature methods* **14**, 407 (2017).
20 Poplin, R. *et al. DeepVariant training data*. https://github.com/google/deepvariant/blob/r0.9/docs/deepvariant-details-training-data.md, accessed Nov 22 2019.
21 Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078-2079, doi:10.1093/bioinformatics/btp352 (2009).
22 Smith, L. N. in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV).* 464-472 (IEEE).
23 Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. in *Proceedings of the IEEE international conference on computer vision.* 2980-2988.
24 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).

# Figures

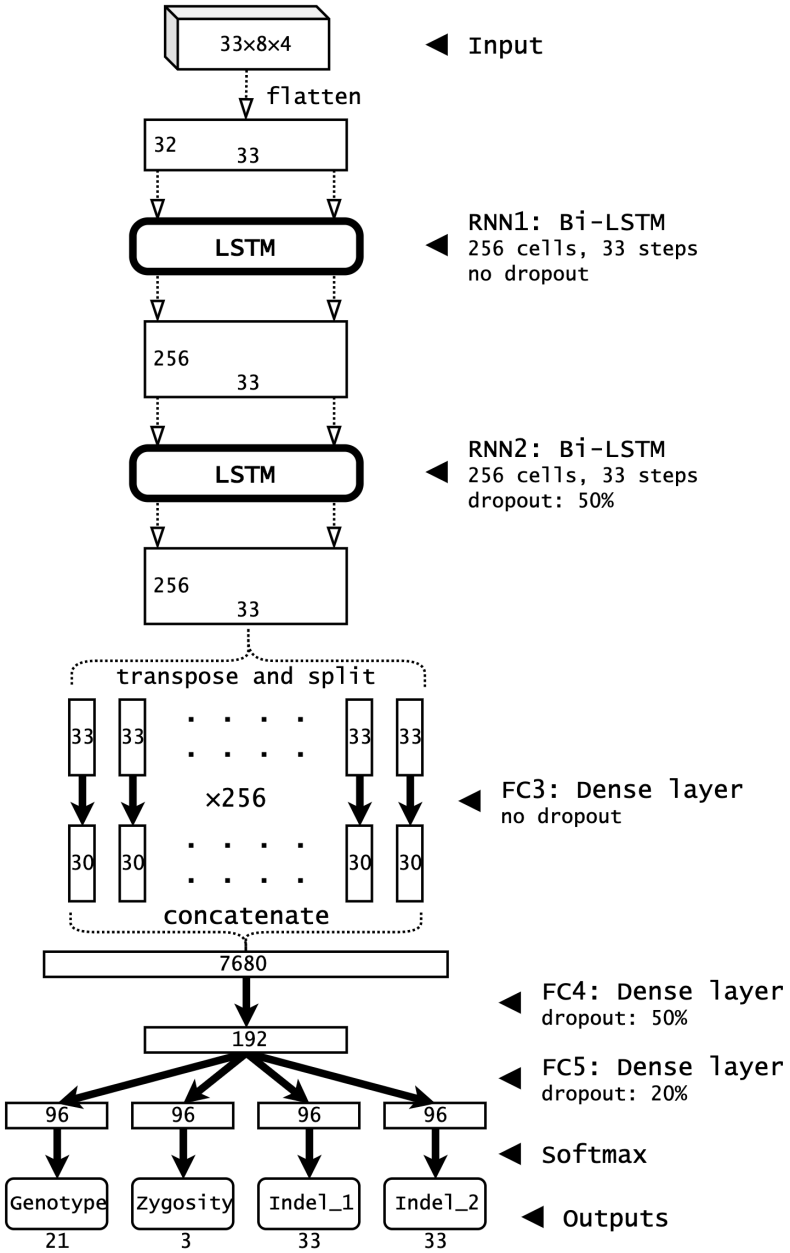


Figure 1. Clair network architecture and layer details. RNN: Recurrent Neural Network. FC: Fully Connected layer. Bi-LSTM: Bi-directional Long Short-Term Memory layer.

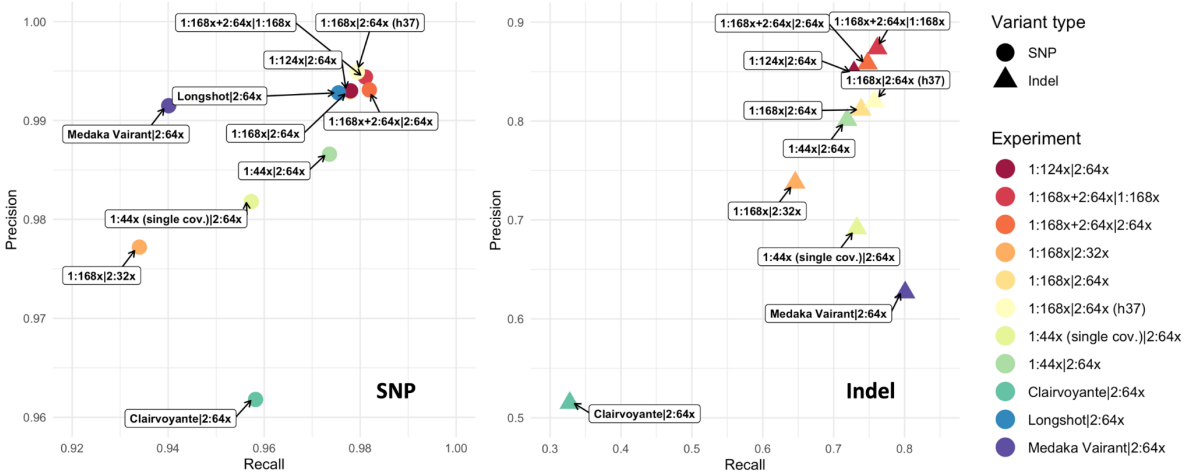


Figure 2. ONT benchmarking results. For Clair, the datasets used for model training and testing are separated with a vertical bar '|', and are written as '*a*:*b*x', where *a* denotes the suffix of the GIAB sample ID (e.g., 1 means HG001), and *b* denotes the coverage of the dataset. Longshot calls only SNP variants, so it is not shown in the indel results.

Figure 3. The category distribution of FPs and FNs made by Clair in the 1:168x|2:64x experiment on ONT data, and six genome browser screen captures showing examples of different categories. In the screen captures, bases A, C, G, and T are green, blue, yellow, and red, respectively. Gaps (i.e., deletions) are dark gray. Insertions are purple dots between two bases and are wider when the insertion is longer.