

1 Symphonizing pileup and full-alignment 2 for deep learning-based long-read variant 3 calling

4 Zhenxian Zheng, Shumin Li, Junhao Su, Amy Wing-Sze Leung, Tak-Wah Lam, Ruibang Luo*

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

6
7
8
9 * Email: rbluo@cs.hku.hk

10 11 Abstract

12 Deep learning-based variant callers are becoming the standard and have achieved superior
13 SNP calling performance using long reads. In this paper, we present Clair3, which makes the
14 best of two major method categories: pile-up calling handles most variant candidates with
15 speed, and full-alignment tackles complicated candidates to maximize precision and recall.
16 Clair3 ran faster than any of the other state-of-the-art variant callers and performed the
17 best, especially at lower coverage.

18 19 Maintext

20 The first preprint of DeepVariant¹ was released in late 2016, marking the beginning of the
21 use of deep learning-based methods (DL methods) instead of traditional statistical methods
22 for variant calling. Over the years, several DL methods have been developed. We are now
23 witnessing a complete take-over, led by DeepVariant for short-read variant calling. Long-
24 read variant calling, using Oxford Nanopore (ONT) data, on the other hand, has been
25 dominated by DL-methods since the beginning, primarily owing to the difficulty caused by
26 its higher base error rate in general. Although the DL methods for short-read and long-read
27 have a lot in common, the problem of long-read variant calling is considered more difficult.
28 This led to two major designs – using pileup or full-alignment as the input of the decision-
29 making neural network – which are significantly different in both performance and speed.
30 Long-read variant callers, including Clairvoyante², Clair³, and Nanocaller⁴, are pileup-based,

31 in which the read alignments are summarized into features and counts before being
32 inputted into a variant calling network. PEPPER-Margin-DeepVariant⁵ (PEPPER) is full
33 alignment-based. The input to the DeepVariant variant calling network is kept with spatial
34 information in the read alignments and is tens of times larger than the pileup inputs in
35 terms of size. Medaka⁶ is consensus-based; it uses pileup input to generate a diploid
36 consensus in the first iteration and two haploid consensuses in the second. The differences
37 between the reference and consensuses are identified and combined into variants. These
38 are all state-of-the-art algorithms; the pileup-based algorithms are usually superior in terms
39 of time efficiency and the full-alignment algorithms provide the best precision and recall.
40 However, while the two designs are not mutually exclusive, there have not been any studies
41 combining pileup calling and full-alignment calling.

42

43 To fill the gap, we developed Clair3, the successor to Clair, which makes the best of both
44 designs. It runs as fast as the pileup-based callers and performs as well as the full alignment-
45 based callers. **Supplementary Figure 1** shows the workflow for Clair3. The philosophy
46 behind Clair3 is to trust the full-alignment model unless the pileup model can make a quick
47 but reliable decision. First, the pileup calling network goes through all the variant candidates
48 that passed a coverage threshold and an alternative allele frequency threshold. Next, the
49 high-quality pileup calls are used to phase the alignments and as part of the final output.
50 Then, the alignments phased by WhatsHap⁷ are used to generate full-alignment input that is
51 ~23 times larger in size than the pileup input for each low-quality pileup call for full-
52 alignment calling. Finally, the full-alignment calls are integrated with the high-quality pileup
53 calls as the final output. More details and parameters about the Clair3 workflow,
54 input/output, and network architecture are provided in **Methods**.

55

56 We benchmarked Clair3 against PEPPER (the current top-performing long-read variant
57 caller), Medaka (ONT's in-house developed variant caller), Longshot⁸ (non-deep learning-
58 based; works only with SNP), and Clair (the Clair3 predecessor) on two GIAB^{9, 10} samples:
59 HG003 and HG004. HG003 was tested on models (including a pileup and a full-alignment
60 model) trained on HG001, 2, 4 and 5. HG004 was tested on models trained on HG001, 2, 3
61 and 5. The model availability and training details are in **Methods**. We chose to use ONT data
62 base-called using Guppy 4 (version 4.2.2) for two reasons: 1) compared to the Guppy 5,

63 which was released in mid-2021, Guppy 4's read accuracy is $\sim 1.8\%$ lower¹¹, which is more
64 challenging to variant calling, so it can better test the speed and performance of different
65 variant calling methods, and 2) as at the completion date of this paper, Guppy 4 base-called
66 reads were still the latest version available for download by the Human Pangenome
67 Reference Consortium¹². A summary of the datasets used for training and testing is shown
68 in **Supplementary Table 1**. The correct PEPPER and Medaka model for Guppy 4 data was
69 chosen for benchmarking. The links to the dataset, and the versions, commands and
70 parameters used for each tool are available in the **Supplementary Notes**.

71
72 The benchmarking results at coverage from 10x to 90x are shown in **Figure 1a**,
73 **Supplementary Table 2**, and **Supplementary Table 3**. The observations of different tools on
74 HG003 and HG004 are almost identical, ruling out the possibility of any tools' overfitting to a
75 particular sample. In terms of the SNP F1-score, Clair3 outperformed the other tools at
76 lower coverage (10x to 30x) and performed similar to PEPPER above 30x. Above 50x, the
77 SNP F1-score improvement became more subtle. However, the Indel F1-score kept
78 increasing with coverage, although it slowed down above 50x. Looking at the precision and
79 recall at 50x (**Figure 1b**), in terms of SNP, Clair3 achieved 99.67% and 99.60%, which is
80 similar to PEPPER's 99.61% and 99.63%. In terms of Indel, Clair3 achieved 90.86% and
81 64.73%, higher than PEPPER's 87.62% and 57.42%. In terms of speed (**Figure 1c**), Clair3 and
82 Clair ran the fastest (~ 8 hours), and PEPPER was second-fastest (~ 30 hours). We then
83 compared Clair3 to PEPPER using the CMRG v1 small variant benchmarking dataset¹³, which
84 covers repetitive and highly polymorphic medically relevant genes, so it is more challenging
85 than using GIAB. However, CMRG v1 is based on HG002. To ensure no testing variant was
86 involved in training, instead of training a new model with HG002 left out, we selectively
87 benchmarked the 5,837 (out of 21,232) small variants that are in CMRG v1, but not GIAB
88 HG002. The results are shown in **Supplementary Figure 2** and **Supplementary Table 4**.
89 Similar to the trends observed for HG003 and HG004, Clair3 outperformed PEPPER at 10x to
90 30x on SNP, and had a similar performance above 30x. We compared Clair3 to PEPPER by
91 different genomic contexts according to the GIAB genome stratifications¹⁴ v2.0 on HG003 at
92 50x. The results are shown in **Supplementary Figure 3** and **Supplementary Table 5**. In SNPs,
93 Clair3 outperformed PEPPER on precision in low complexity and functional regions, but not
94 in low mappability and segmental duplication regions. Clair3 and PEPPER had the same

95 recall in different regions. In Indels, Clair3 outperformed PEPPER in both precision and recall
96 in all regions.

97

98 The success of the Clair3 method lies in the effective distinction between true and false calls
99 during pileup calling, so that only necessary candidates are sent to the much more
100 computationally intensive full-alignment calling. **Figure 2a** shows that an effective
101 distinction was achieved using variant quality. Using HG003 at 50x as an example, most false
102 variant calls and false reference calls had a quality between 0 to 10, while the true calls
103 were between 15 to 30. In reality, while the correctness of a pileup call is not known in
104 advance, we empirically decided to send the bottom 30% of the pileup variant calls and the
105 bottom 10% of the pileup reference calls to full-alignment calling as the default settings of
106 Clair3 (See **Methods**). In the previous example, quality cut-off 16 was chosen for the variant
107 calls, which resulted in 98% of the false variant calls and only 9% of the true variant calls
108 being sent to full-alignment calling. Similarly, cut-off 19 was chosen for the reference calls,
109 so that 98% of the false reference calls and only 11% of the true reference calls were sent to
110 full-alignment calling. **Figure 2b** shows that ~62% of the pileup failed variant calls and ~31%
111 of the pileup failed reference calls were correctly called in full-alignment calling. We tested
112 sending different percentages of pileup variant calls to full-alignment calling, from 0%
113 (pileup calling only) to 100% (full-alignment calling only). The results are shown in **Figure 2c**
114 and **Supplementary Table 6**. Clair3's default, which had a similar performance to full-
115 alignment calling but ran ~4 times faster, showed that integrating pileup and full-alignment
116 calling is a better strategy than relying on only one of them.

117

118 The benchmarks focused on the more challenging ONT data, but the Clair3 method is not
119 restricted to a certain sequencing technology. It should work particularly well in terms of
120 both runtime and performance on noisy data. Clair3 was released six months ago and is
121 currently in its ninth revision, having integrated plenty of feedback from the community and
122 ONT. We observed in PEPPER's most recent update (r0.7 on Dec 22nd, 2021) that a module
123 in the front of the pipeline that was used solely for variant candidate selection was
124 repurposed to output summary-based variant calls to relieve the heavy full-alignment
125 calling workload. We expect integrating pileup and full-alignment calling to be a common
126 practice in deep learning-based variant calling in the future.

127 Method

128 The Clair3 workflow

129 As **Supplementary Figure 1** shows, pileup candidates that are above a coverage threshold
130 and an allele frequency threshold are extracted, and then called using the pileup network.
131 The pileup calls are grouped into variant calls (genotype 0/1, 1/1, and 1/2) and reference
132 calls (0/0). Both groups are ranked according to variant quality (QUAL). High-quality
133 heterozygous SNP calls (top 70% in 0/1 calls) are used in WhatsHap phasing to produce
134 phased alignment for input to the full-alignment network. Low-quality pileup calls
135 (defaulted to the lowest 30% of variants and 10% of reference calls) are then called again
136 using the full-alignment network. Finally, the full-alignment calls and high-quality pileup
137 calls are outputted. Clair3 supports both VCF and GVCF output formats.

138

139 Input/Output

140 Clair3 uses a pileup input design simplified from that of its predecessors, and a full-
141 alignment input to cover as many details in the read alignments as possible. **Supplementary**
142 **Figure 4** visualizes the pileup and full-alignment inputs of a random SNP, insertion, deletion,
143 or non-variant. **The pileup input** is 594 integers – 33 genome positions wide with 18
144 features at each position – A+, C+, G+, T+, I_S+, I¹_S+, D_S+, D¹_S+, D_R+, A-, C-, G-, T-, I_S-, I¹_S-, D_S-,
145 D¹_S-, and D_R-. A, C, G, T, I, D, +, - means the count of read support of the four nucleotides:
146 insertion, deletion, positive strand, and negative strand. Superscript “¹” means only the
147 indel with the highest read support is counted (i.e., all indels are counted if without “¹”).
148 Subscript “_S”/“_R” means the starting/non-starting position of an indel. For example, a 3bp
149 deletion with the most reads support will have the first deleted base counted in either D¹_S+
150 or D¹_S-, and the second and third deleted bases counted in either D_R+ or D_R-. The design was
151 determined experimentally, but the rationale is that for 1bp indels that are easy to call, look
152 into the differences between the “_S” counts, but reduce the quality if the “_R” counts and
153 discrepancy between positions increase. **The pileup output** is the same as that for Clair, but
154 short of the two indel length tasks. The indel allele (or two indel alleles) with the highest
155 reads support is used as the output according to the decision made in the 21-genotype task.
156 **The full-alignment input** is 23,496 integers – 8 channels of 33 genome positions and 89
157 maximum representable reads. The description of the eight channels is in the

158 Supplementary Note. **The full-alignment output** is the same as that of Clair. The two indel
159 length tasks can represent the exact indel length from -15 to 15bp, or below -15bp/ above
160 15bp. An indel call with an exact length will output the most reads-supported allele at that
161 length. Otherwise, the most reads-supported allele below -15bp/ above 15bp is outputted.
162 In training, indel length task 1 is given the smaller number, and in all our variant calling
163 experiments, no length predictions in task 1 larger than in task 2 were observed. **The**
164 **maximum supported coverage** of pileup/full-alignment input was 144/89. Random
165 subsampling was done on excessive coverage. If the coverage in a full-alignment input was
166 below 89, the reads were centered.

167

168 [Network architecture](#)

169 The pileup and full-alignment networks are shown in **Supplementary Figure 5**. **The pileup**
170 **network** uses two bidirectional long short-term memory (Bi-LSTM) layers with 128 and 160
171 LSTM units. Stacked LSTM layers enable the network to learn the characteristics of raw
172 sequential signal from different aspects at each position, but without increasing memory
173 capacity, which enables the network to converge faster. Compared to Clair, the transpose-
174 split layer is removed for a 40% speedup with a small performance loss that is taken care of
175 in full-alignment calling. **The full-alignment network** is residual neural network (ResNet)
176 alike and uses three standard residual blocks. A convolutional layer is added on top of each
177 residual block to expand channels but reduce dimensionality across channels. A spatial
178 pyramid pooling¹⁵ (SPP) layer is used to tackle the problem of varying coverage in full-
179 alignment input. SPP is a pooling layer that removes a network's fixed-size constraint, thus
180 avoiding the need for input cropping or warping at the beginning. The SPP layer generates
181 various receptive fields using three pooling scales (1x1, 2x2, and 3x3) in each channel. It
182 then pools the receptive fields of all channels and generates a fixed-length output for the
183 next layer. In both networks, the dropout rates of 0.2 for the flatten layer, 0.5 for the
184 penultimate dense layer, and 0.2 for the task-specific final dense layers, are empirically
185 determined.

186

187 Model availability and training

188 Pretrained models are provided in Clair3's installation. Models for specific chemistries and
189 basecallers that are tested and supported by the ONT developers are available through
190 Rerio (<https://github.com/nanoporetech/rerio>). The detailed steps, options and caveats for
191 training a pileup model and a full-alignment model are available in Clair3's GitHub repo and
192 are continually updated. The pretrained models, while targeted for use in production, were
193 trained using multiple GIAB samples with known variants and 10 coverages for each sample,
194 as described in Clair, but they always hold out chromosome 20 in Clair3. We used the
195 following new training technics in Clair3. **(1) Representation Unification:** a variant can be
196 represented in multiple forms¹⁴. Traditional variant calling methods rely on postprocessing
197 (e.g., hap.py, RTG Tools) to equate multiple forms. However, to generate correct training
198 samples, Clair3 must unify a variant's representations between the alignments and the truth
199 variants. **Supplementary Figure 6** shows four cases in which the alignments and the truth
200 variants have different representations that would confuse the training if not unified. Clair3
201 chooses to align the truth variants' representation to the alignments. The five detailed steps
202 are shown in **Supplementary Figure 7**. First, the truth variants and alignments are phased (if
203 not yet done) using WhatsHap. Second, among the candidates with alternative allele
204 frequency ≥ 0.15 , confident and *in situ* matches between the alignments and truth variants
205 are identified and excluded from computationally intensive step 3. Third, the best match
206 between the possible haplotypes of the truth variants and candidates is sought. Each of the
207 truth variants can be either positive (using its reported genotype) or negative (using 0|0),
208 and their Cartesian product forms possible haplotypes of the truth variants. Similarly, each
209 candidate can be either 0|0, 0|1 (or 1|0 according to the phased alignments), or 1|1, and
210 their Cartesian product forms the possible haplotypes of the candidates. A pairwise
211 comparison is then done to find equivalent haplotypes between the two Cartesian products,
212 and among all equivalents, the candidate haplotype with the most reads support is selected.
213 The variants in the haplotype are used as the new truth variants. This step is
214 computationally intensive, so in practice, we applied the step to partitions with at most 15
215 candidates and required less than 100bp between the candidates. Fourth, low alternative
216 allele frequency (≥ 0.08 but < 0.15) candidates with *in situ* matches between the alignments
217 and the truth variants were chosen. Fifth, the truth variants or unified variants generated in
218 steps 2, 3 and 4 were merged. In our benchmarks, representation unification alone in

219 general increased the SNP recall by $\sim 0.2\%$ and Indel recall by $\sim 2\%$. **(2) Ratio of variants to**
220 **non-variants samples for training:** In Clair, the ratio was fixed at 1:2. In Clair3, we tested
221 ratios up to 1:10 for both pileup and full-alignment model training, and we observed a
222 monotonic but decelerated performance increase with more non-variants added to the
223 training. Since focal loss is used to alleviate the effect of training class imbalance, another
224 possible explanation is that the 21-genotype output task that Clair3 relies primarily on is
225 insensitive to the ratio because it judges only the genotype of a candidate instead of
226 whether a candidate is a variant or not. We chose 1:5 and 1:1 as the default ratio for pileup
227 and full-alignment model training, respectively, to strike a balance between model
228 performance and training speed. **(3) Use of phased alignments:** Deep-learning and full-
229 alignment based variant callers DeepVariant and PEPPER concluded that using phased
230 alignments is essential to their high performance. In Clair3, high-quality heterozygous pileup
231 calls are used to phase the input alignments using the ‘phase’ and ‘haplotag’ modules in
232 WhatsHap. The phased alignments are used as input for full-alignment calling. When
233 training a full-alignment model, two training samples for each variant, one using phased
234 alignments and the other unphased, are used to ensure the model works when alignments
235 cannot be properly phased. In our benchmarks, the use of phased alignments alone, in
236 general, increased the SNP F1-score by $\sim 0.1\%$, and the Indel F1-score by $\sim 6\%$. **(4) New**
237 **optimization methods:** Clair3 removed both the cyclical learning rate and learning rate
238 decay strategies used in Clair, and now uses the Ranger optimizer (RectifiedAdam¹⁶ plus
239 Lookahead¹⁷) for automated warm-up, faster convergence, minimal computational
240 overhead, etc. In our benchmarks, compared to Clair, the new optimizer alone, in general,
241 increased the overall F1-score of pileup calling by $\sim 0.2\%$.

242

243 [Benchmarking methods and computational concerns](#)

244 We used five GIAB samples, HG001 to 5, for either model training or testing. When using
245 either HG003 or HG004 for testing, the other four samples were used for training. We
246 selected 10% of the training samples for validation and chose the best-performing epoch in
247 the first 30 epochs in the validation data for benchmarking. We used hap.py¹⁴ to compare
248 the called variants against the true variants, and used Clair3’s ‘GetOverallMetrics’ module to
249 generate three metrics, ‘precision’, ‘recall’, and ‘F1-score’, for five categories: ‘overall’,

250 ‘SNP’, ‘Indel’, ‘Insertion’, and ‘Deletion’. We used qfy.py with V2.0 GIAB genome
251 stratifications to evaluate Clair3’s performance in challenging and targeted regions of the
252 genome. Runtimes were gauged on a server with two 2.1GHz Intel Xeon Silver 4116s, with
253 24 cores, and 256GB memory at 2666MHz. With the same setting, Clair3 finished in ~8
254 hours using ~50x of ONT Guppy 4 data and in ~4.5 hours with the same amount of Guppy 5
255 data. The memory consumption of each Clair3 calling process was capped at 1GB.

256

257 [Brief summary of methods tested showing no or negligible improvement](#)

258 **(1) Use of more residual blocks in the full-alignment network:** We added a fourth residual
259 block with 512 channels. The number of parameters increased from 2,989,210 to 9,812,634.
260 The runtime doubled, but the performance change was negligible, even though the terminal
261 training loss fell. **(2) Local realignment:** This technique is essential for high indel calling
262 performance in state-of-the-art, short-read, small variant callers. But it worked differently
263 on long-read. We tried local realignment using a 2000bp window in regions with a high
264 density of candidates using a local realignment algorithm similar to that of DeepVariant. We
265 observed that while it increased the recall a bit, local realignment tripled the runtime and
266 introduced ~10% of new non-variant candidates, which in turn, lowered the precision a bit.
267 In Clair3, we implemented local realignment, but disabled it on long-read as the default. **(3)**
268 **Including variants outside high-confidence regions in training:** To increase variant training
269 samples, we explored including variants outside the high-confidence regions in training, but
270 observed negative performance improvement in Clair. In Clair3, the GIAB truth datasets we
271 used were upgraded from version 3.3.2 to 4.2.1, but we had the same observation that
272 including variants outside the high-confidence regions in training jeopardized model
273 performance. **(4) Selecting candidates for full-alignment calling based on reference**
274 **sequence complexity:** Variant calling is more difficult in the “low complexity” and “difficult
275 to map” regions. In addition to selecting candidates by pileup calling quality ranking for full-
276 alignment calling, we added those candidates at positions with relatively low sequence
277 entropy (the lowest 30% of the whole genome). About three times more candidates were
278 selected for full-alignment calling, but the performance increase was negligible.

279

280 Code availability

281 Clair3 is open-source software (BSD 3-Clause license), hosted by GitHub at
282 <https://github.com/HKU-BAL/Clair3>, and available through Docker, Bioconda, and
283 Singularity.

284

285 Data availability

286 The 1) links to the reference genomes, truth variants, benchmarking materials, and ONT
287 data, and 2) the commands and parameters used in this study, are available in the
288 Supplementary Notes. All analysis output, including the VCFs and running logs, are available
289 at http://www.bio8.cs.hku.hk/clair3/analysis_result.

290

291 Acknowledgements

292 R. L. was supported by Hong Kong Research Grants Council grants GRF (17113721), ECS
293 (27204518), and TRS (T21-705/20-N and T12-703/19-R), and the URC fund at HKU.

294

295 Author contributions

296 R. L. conceived the study. Z. Z. and R. L. designed the algorithms, implemented Clair3, and
297 wrote the paper. All authors evaluated the results and revised the manuscript.

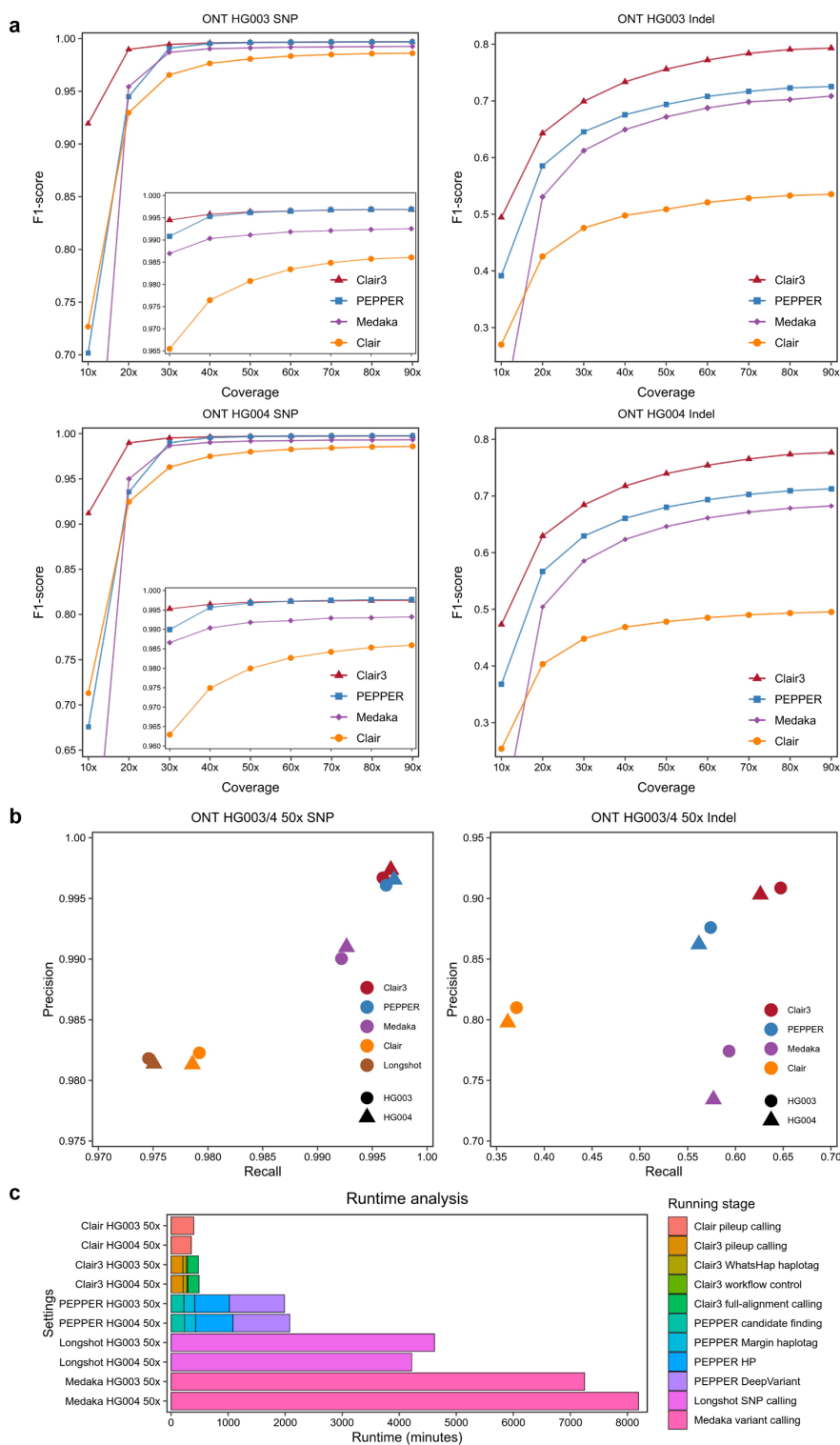
298

299 Competing interests

300 R. L. receives research funding from ONT. The remaining authors declare no competing
301 interests.

302

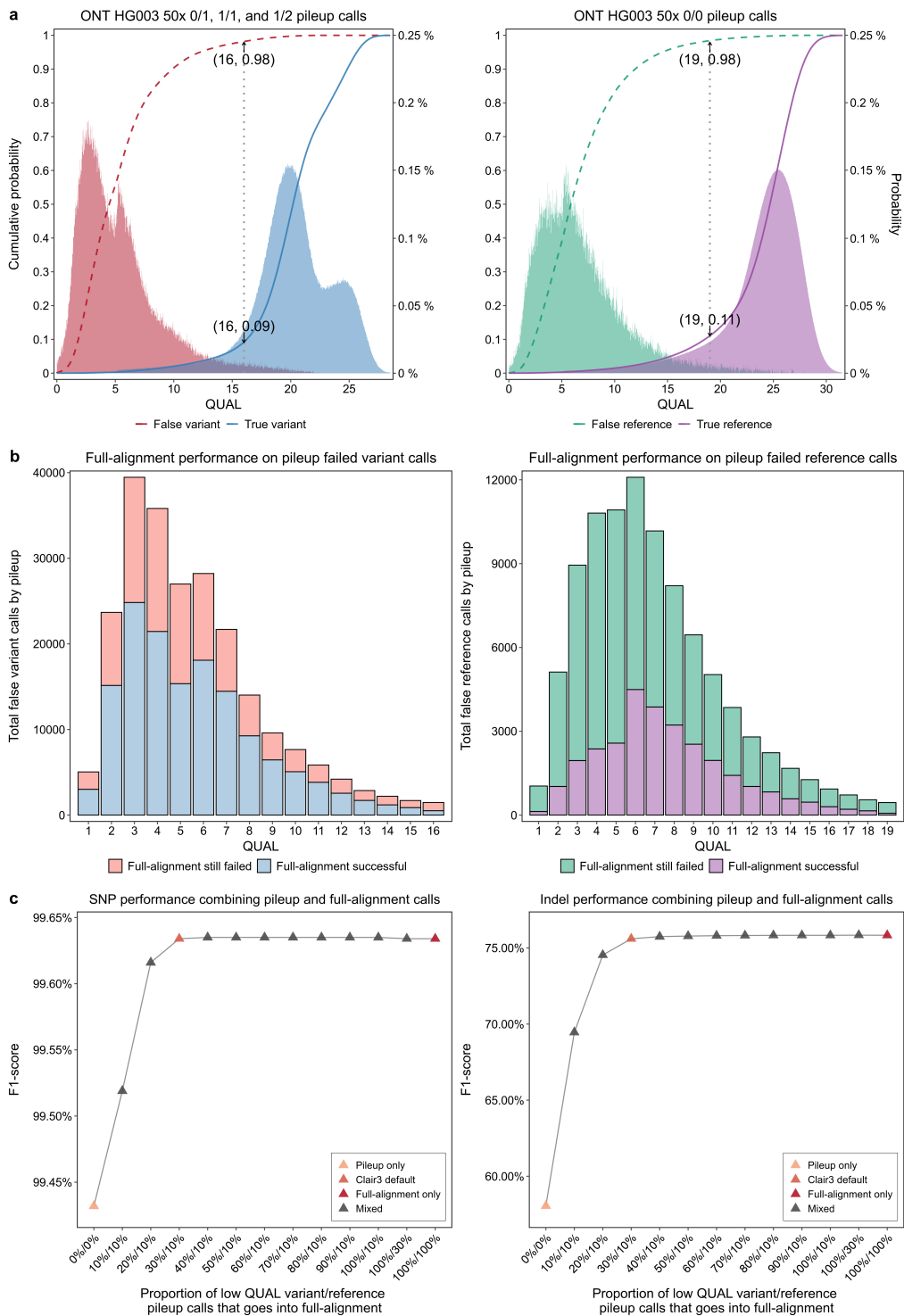
303 Figures



304

305 **Figure 1. Benchmarking results on HG003 and HG004.** (a) The SNP/Indel F1-score of
 306 different tools at multiple coverage from 10x to 90x. (b) The precision against the recall of
 307 different tools at 50x coverage. (c) The runtime breakdowns of different tools at 50x
 308 coverage.

309



310

311 **Figure 2. Pileup and full-alignment calling working details and synergy on HG003 at 50x**
 312 **coverage.** (a) The variant quality distribution of the true and false variant/reference pileup
 313 calls. (b) The performance of full-alignment on pileup failed variants of different variant
 314 quality. (c) The F1-score when different proportions of low-quality variant/reference calls
 315 enter full-alignment calling.

317 References

- 318 1. Poplin, R. et al. A universal SNP and small-indel variant caller using deep neural
319 networks. *Nature biotechnology* **36**, 983-987 (2018).
- 320 2. Luo, R., Sedlazeck, F.J., Lam, T.-W. & Schatz, M.C. A multi-task convolutional deep
321 neural network for variant calling in single molecule sequencing. *Nature*
322 *communications* **10**, 1-11 (2019).
- 323 3. Luo, R. et al. Exploring the limit of using a deep neural network on pileup data for
324 germline variant calling. *Nature Machine Intelligence* **2**, 220-227 (2020).
- 325 4. Ahsan, M.U., Liu, Q., Fang, L. & Wang, K. NanoCaller for accurate detection of SNPs
326 and indels in difficult-to-map regions from long-read sequencing by haplotype-aware
327 deep neural networks. *Genome Biology* **22**, 261 (2021).
- 328 5. Shafin, K. et al. Haplotype-aware variant calling with PEPPER-Margin-DeepVariant
329 enables high accuracy in nanopore long-reads. *Nature methods* **18**, 1322-1332
330 (2021).
- 331 6. Medaka, <https://github.com/nanoporetech/medaka>.
- 332 7. Patterson, M. et al. WhatsHap: weighted haplotype assembly for future-generation
333 sequencing reads. *Journal of Computational Biology* **22**, 498-509 (2015).
- 334 8. Edge, P. & Bansal, V. Longshot enables accurate variant calling in diploid genomes
335 from single-molecule long read sequencing. *Nature communications* **10**, 1-10 (2019).
- 336 9. Olson, N.D. et al. precisionFDA Truth Challenge V2: Calling variants from short-and
337 long-reads in difficult-to-map regions. *Biorxiv*, 2020.2011.2013.380741 (2021).
- 338 10. Wagner, J. et al. Benchmarking challenging small variants with linked and long reads.
339 *BioRxiv* (2020).
- 340 11. Nanopore EPI2ME Labs, https://labs.epi2me.io/gm24385_2021.05/.
- 341 12. Shafin, K. et al. Nanopore sequencing and the Shasta toolkit enable efficient de novo
342 assembly of eleven human genomes. *Nature biotechnology* **38**, 1044-1053 (2020).
- 343 13. Wagner, J. et al. Towards a Comprehensive Variation Benchmark for Challenging
344 Medically-Relevant Autosomal Genes. *bioRxiv* (2021).
- 345 14. Krusche, P. et al. Best practices for benchmarking germline small-variant calls in
346 human genomes. *Nature biotechnology* **37**, 555-560 (2019).
- 347 15. He, K., Zhang, X., Ren, S. & Sun, J. Spatial pyramid pooling in deep convolutional
348 networks for visual recognition. *IEEE transactions on pattern analysis and machine*
349 *intelligence* **37**, 1904-1916 (2015).
- 350 16. Liu, L. et al. On the variance of the adaptive learning rate and beyond. *arXiv preprint*
351 *arXiv:1908.03265* (2019).
- 352 17. Zhang, M.R., Lucas, J., Hinton, G. & Ba, J. Lookahead optimizer: k steps forward, 1
353 step back. *arXiv preprint arXiv:1907.08610* (2019).

354