ARTICLES
https://doi.org/10.1038/s41587-022-01435-7
nature biotechnology

DeepConsensus improves the accuracy of sequences with a gap-aware sequence transformer

Gunjan Baid[1,3], Daniel E. Cook[1,3], Kishwar Shafin[1], Taedong Yun[1], Felipe Llinares-López[1], Quentin Berthet[1], Anastasiya Belyaeva[1], Armin Töpfer[2], Aaron M. Wenger[2], William J. Rowell[2], Howard Yang[1], Alexey Kolesnikov[1], Waleed Ammar[1], Jean-Philippe Vert[1], Ashish Vaswani[1], Cory Y. McLean[1], Maria Nattestad[1,3], Pi-Chuan Chang[1,3] and Andrew Carroll[1,3] ✉

Circular consensus sequencing with Pacific Biosciences (PacBio) technology generates long (10–25 kilobases), accurate 'HiFi' reads by combining serial observations of a DNA molecule into a consensus sequence. The standard approach to consensus generation, pbccs, uses a hidden Markov model. We introduce DeepConsensus, which uses an alignment-based loss to train a gap-aware transformer–encoder for sequence correction. Compared to pbccs, DeepConsensus reduces read errors by 42%. This increases the yield of PacBio HiFi reads at Q20 by 9%, at Q30 by 27% and at Q40 by 90%. With two SMRT Cells of HG003, reads from DeepConsensus improve hifiasm assembly contiguity (NG50 4.9 megabases (Mb) to 17.2 Mb), increase gene completeness (94% to 97%), reduce the false gene duplication rate (1.1% to 0.5%), improve assembly base accuracy (Q43 to Q45) and reduce variant-calling errors by 24%. DeepConsensus models could be trained to the general problem of analyzing the alignment of other types of sequences, such as unique molecular identifiers or genome assemblies.

Modern genome sequencing samples the genome in small, error-prone fragments called reads. At the read level, the higher error of single-molecule observations is mitigated by consensus observations. In Illumina data, the consensus is spatial through clusters of amplified molecules[1]. Pacific Biosciences (PacBio) uses repeated sequencing of a circular molecule to build consensus across time[2]. The accuracy of these approaches, and the manner in which they fail, ultimately limits the read lengths of these methods and the analyzable regions of the genome[3,4].

Recent improvements in PacBio throughput have enabled highly accurate (99.8%) long reads (>10 kilobases (kb)), called HiFi reads[5], to set new standards in variant-calling accuracy[6] and the first telomere-to-telomere human assembly[7]. The remaining sequencing errors are strongly concentrated in homopolymers[3,8], and the need to manage these errors constrains the minimum number of passes required for acceptable accuracy and therefore the yield and quality of PacBio sequencing.

The existing algorithm for consensus generation from HiFi sequencing data uses a hidden Markov model to create a draft consensus sequence, which is iteratively polished[9]. The underlying process of removing errors using an alignment of reads is also used in genome assembly[10] and in assembly polishing methods such as Racon[11], Pilon[12] and PEPPER-Margin-DeepVariant[13]. All of these methods correct from a given alignment to a reference or contig. These methods use statistical heuristics for the correction model itself except for PEPPER-Margin-DeepVariant.

To improve consensus generation of HiFi sequencing data, we introduce a deep learning-based approach using a transformer[14] architecture. Transformers have gained rapid adoption in natural language processing[15] and computer vision[16]. In biology, transformers have been applied to multiple sequence alignment (MSA) of protein sequences[17] and dramatically improved AlphaFold2's protein structure prediction[18]. We present DeepConsensus, an encoder-only transformer model that uses an MSA of the PacBio subread bases and a draft consensus from the current production method (pbccs). DeepConsensus incorporates auxiliary base-calling features to predict the full sequence in a window (by default 100 base pairs (bp)). Because insertion and deletion (INDEL) errors are the dominant class of error in these data, we train the model with an alignment-based loss function inspired by differentiable dynamic programming[19]. This gap-aware transformer–encoder (GATE) approach more accurately represents misalignment errors in the training process.

DeepConsensus reduces errors in PacBio HiFi reads by 41.9% compared to pbccs in human sequence data. We stratify performance across mismatches, homopolymer INDELs and non-homopolymer INDELs, and DeepConsensus improves accuracy in each category. DeepConsensus increases the yield of reads at 99% accuracy by 8.7%, at 99.9% accuracy by 26.7% and at 99.99% accuracy by 90.9%. We demonstrate that using reads from DeepConsensus improves the contiguity, completeness and correctness of genome assembly compared to assemblies generated using pbccs reads. Similarly, we demonstrate improved accuracy of variant calling when using DeepConsensus reads. Finally, we demonstrate that improvements in accuracy allow for longer PacBio read lengths while retaining acceptable read accuracy, enabling improvements in contiguity of genome assembly and increasing the experimental design options for PacBio sequencing.

Results

Overview of DeepConsensus. An overview of the DeepConsensus algorithm is shown in Fig. 1. PacBio circular consensus sequencing (CCS) produces a set of subreads that is processed by pbccs to produce a consensus (CCS) read. Subreads are aligned to the CCS read. The alignment is then divided into 100-bp partitions based on the MSA length. Each partition is then transformed into a tensor to be used as input to the DeepConsensus model for training or inference.

[1]Google LLC, Mountain View, CA, USA. [2]Pacific Biosciences, Menlo Park, CA, USA. [3]These authors contributed equally: Gunjan Baid, Daniel E. Cook, Maria Nattestad, Pi-Chuan Chang, Andrew Carroll. ✉e-mail: awcarroll@google.com

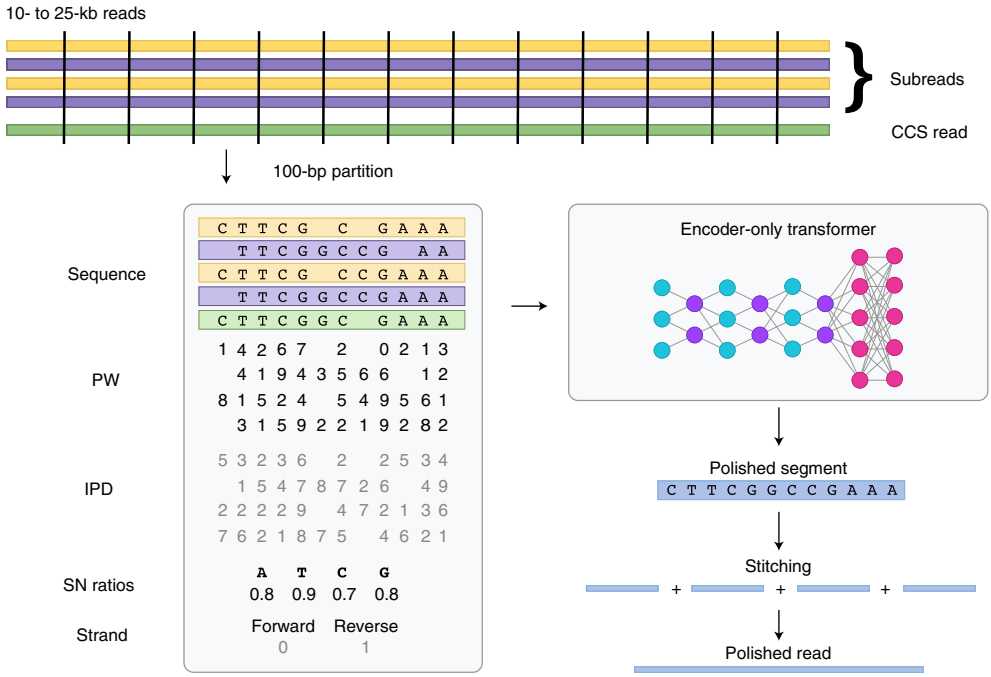


Fig. 1 | Overview of DeepConsensus. Illustration of the DeepConsensus workflow. Subreads are aligned to a CCS read divided into 100-bp partitions. Each partition is converted to a tensor object containing the PW, IPD, SN ratios and strand information. These tensors can then be used during training or inference using an encoder-only transformer. The trained model produces a polished segment that is stitched together to produce a polished read.

The tensor contains additional information beyond the sequence extracted from each subread. This includes the pulse width (PW) and interpulse duration (IPD). These are raw values provided by the basecaller that are used to call bases. Additionally, DeepConsensus incorporates the signal-to-noise (SN) ratio for each nucleotide and strand information. For training, we use a custom loss function that considers the alignment between the label and predicted sequence. For inference, the outputs for each 100-bp partition in the full sequence are stitched together to produce the polished read.

To assess the contributions of the various input features and the alignment loss strategy, we conducted an additional set of training experiments with some or all of the features. Training with all input features and alignment loss showed 37.57% error reduction over pbccs on our test dataset containing HG002 chromosome 19 (chr19) and chr20. Training with the same feature set and cross-entropy loss resulted in a 9.31% error reduction. Training with only sequence information and no pulse metadata showed a 21.0% error reduction. We see additional error reduction from each input feature (Supplementary Table 1).

DeepConsensus increases HiFi accuracy and yield. We first evaluated the performance of DeepConsensus (v0.1) by aligning polished, 11-kb chr20 reads from HG002 against a high-quality diploid assembly[20]. HiFi reads output from pbccs were processed similarly, and both sets were filtered at their predicted Q20 ($Q_{predicted} > 20$). We then used a custom script to calculate an empirical Phred-scaled read accuracy score for each read ($Q_{concordance}$; Methods). When examining the intersection of reads to assess relative improvement, we observed that accuracy improvements are distributed across the full range of pbccs $Q_{concordance}$ scores (Fig. 2a). We observed an average $Q_{concordance}$ of 28.94 for DeepConsensus and 26.6 for pbccs, which corresponds to an average read quality improvement of 2.34 $Q_{concordance}$ points. We also examined read accuracy by the number of subreads used to generate each HiFi read and observed $Q_{concordance}$ improvements for all subread bins (Fig. 2b).

Sequencing errors can be classified by type (mismatch and INDEL) and according to their sequence context (homopolymer and non-homopolymer). Homopolymer INDELs have previously been characterized as the largest contributors to PacBio HiFi error rates[5]. We used bamConcordance[5] to examine the improvements for each error class. Notably, DeepConsensus reduces errors across all error classes, including substantial reductions in homopolymer INDELs and a 70.50% reduction in non-homopolymer insertions (Table 1).

We next asked how improvements in read accuracy contribute to increases in sequencing yield. DeepConsensus and pbccs are both configured to output reads with a predicted $Q > 20$. We compared the total yield and yields at Q thresholds of 20, 30, 40 and perfect match and observed that DeepConsensus increases sequencing yield across all quality bins (Table 2).

In addition to producing a polished sequence, our model also outputs predicted base qualities. The average base quality $Q_{predicted}$ should match the $Q_{concordance}$. We filtered reads with identity = 1 and found that the mean$(Q_{predicted} - Q_{concordance}) = 2.77$. A comparison of pbccs and DeepConsensus is available in Supplementary Fig. 1.

Recently, Lal et al. applied a recurrent neural network to polish PacBio HiFi reads[21] and used a similar approach to PEPPER and other variant-calling techniques. We downloaded their polished reads and compared the data against the same reads polished by DeepConsensus. DeepConsensus achieves a $Q_{concordance}$ of 29.93 versus 29.18 reported in Lal et al.[21] (Supplementary Table 2) on this dataset, corresponding to a reduction in average base pair errors per 10 kb from 16.53 to 10.09 with DeepConsensus versus 12.08-bp errors per 10 kb for their approach (Supplementary Table 3).

Using DeepConsensus reads improves de novo assembly. To evaluate the improvements achieved in de novo assembly with the increased yield (Supplementary Figs. 2–5 and Supplementary Table 4) and higher-quality reads from DeepConsensus, we generated phased assemblies of four human genome samples using the hifiasm[22] assembler. We generated assemblies with reads from two

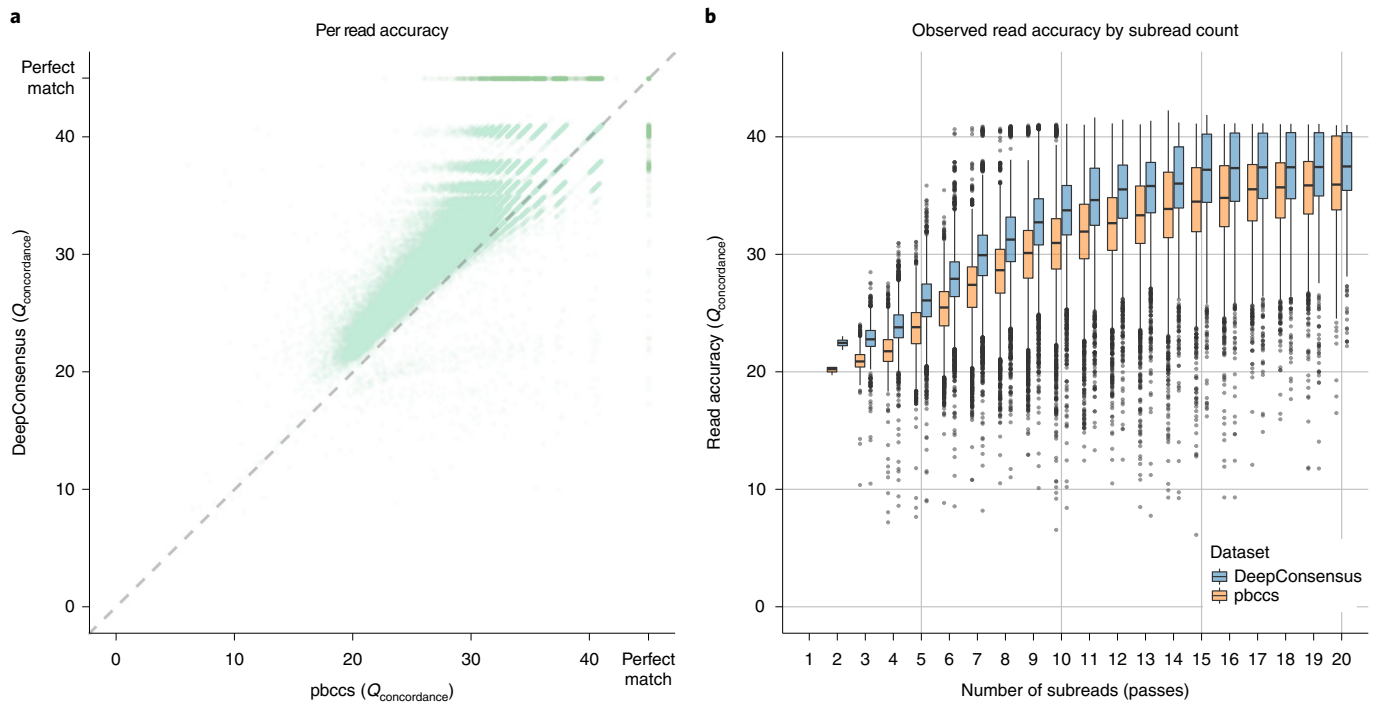


Fig. 2 | DeepConsensus improves the accuracy of CCS reads. a, Comparison of observed read accuracy ($Q_{concrdance}$) for the intersection of pbccs and DeepConsensus (HG002 chr20 11-kb) reads. Each light green dot corresponds to a single read. Dark green dots on the margins represent reads that were perfect matches to the reference. **b**, Observed read accuracy across the number of subreads (passes) for the intersection of available (HG002 chr20 11-kb) reads for DeepConsensus and pbccs ($n = 89{,}927$). Perfect match reads are excluded. The center line of each box plot corresponds to the median. Box bounds correspond to the 25th and 75th percentiles, respectively. Whiskers extend to 1.5× the interquartile range above and below each box. Points correspond to outlier observations beyond the interquartile range.

single-molecule real-time (SMRT) Cells (HG003, HG004, HG006 and HG007) and three SMRT Cells (HG003, HG004 and HG006). To assess the contiguity, we derived the contig N50 (the shortest contig in the assembly for which that contig and all longer contigs compose 50% of the assembly length), NG50 (the shortest contig in the assembly for which that contig and all longer contigs compose 50% of the true genome length) and genome coverage against GRCh38 using QUAST[23]. In Fig. 3, we show the improvements in assembly quality and contiguity as a result of increased yield and quality of reads from DeepConsensus. With reads from two SMRT Cells, we see that the NG50 of the assemblies with DeepConsensus reads (17.23 megabases (Mb), 12.37 Mb, 31.54 Mb and 8.48 Mb) are, on average, threefold higher than assembly NG50 with pbccs reads (4.91 Mb, 3.72 Mb, 18.55 Mb and 1.94Mb; Fig. 3a and Supplementary Table 5).

We evaluated the correctness of the assembly using YAK[22], which overlaps the assembly with *k*-mers observed in short-read sequencing. The YAK-estimated quality of the assemblies with DeepConsensus reads achieves Q44, on average, compared to Q42 with assemblies using pbccs reads (Fig. 3b and Supplementary Table 6). We also used dipcall[24] to derive the small variants from the assembly and compared the small variants against Genome-In-a-Bottle (GIAB) truth sets[25] of the associated sample. We observed that the assemblies derived from DeepConsensus reads have, on average, 43% fewer total errors (false positives and false negatives) than the assemblies derived from pbccs reads (Supplementary Tables 7 and 8).

To evaluate the gene completeness of the assemblies, we used asmgene[26] with the Ensembl *Homo sapiens* cDNA sequences as input and GRCh38 as the reference sequence. We observed that the assemblies generated with pbccs have a twofold higher false duplication rate (average of 540 false duplications) than the assemblies generated with DeepConsensus (average of 231 false duplications; Supplementary Tables 9 and 10).

Table 1 | Corrections by error type

Metric	pbccs errors (per 10 kb)	DeepConsensus errors (per 10 kb)	Percent decrease
Mismatch	1.62	0.88	45.70%
Homopolymer deletion	8.37	5.99	28.40%
Homopolymer insertion	9.14	4.46	51.20%
Non-homopolymer deletion	0.93	0.86	7.50%
Non-homopolymer insertion	1.83	0.54	70.50%
All errors	21.89	12.73	41.80%

The average numbers of errors per 10 kb for each error class are listed for pbccs and DeepConsensus. The percent decrease reflects the reduction in errors in DeepConsensus compared to pbccs.

Similarly, in assemblies generated with three SMRT Cells, we observed that the contig NG50 values of the assemblies with DeepConsensus reads (55 Mb, 41 Mb and 51 Mb) are, on average, 1.3-fold higher than the contig NG50 values of the assemblies with pbccs reads (33 Mb, 36 Mb and 41 Mb; Fig. 3c and Supplementary Table 5). The average assembly quality was Q49.4 with DeepConsensus reads compared to Q48.1 for assemblies with pbccs reads (Fig. 3d and Supplementary Table 6). The assembly-based small-variant evaluation showed that assemblies from DeepConsensus reads have 33% fewer total errors than assemblies with pbccs reads (Supplementary Tables 7 and 8).

The gene completeness analysis showed that the assemblies generated with pbccs (average of 162 false duplications) have a higher number of false duplications than the assemblies generated with DeepConsensus (average of 134 false duplications; Supplementary Tables 9 and 10).

To investigate the ability of DeepConsensus to generalize to non-human species, we generated de novo assemblies of *Zea mays* (Maize B73) using hifiasm with reads from two SMRT Cells to evaluate the performance of DeepConsensus on a non-human sample. Similar to the human samples, we observed that the NG50 of the Maize assembly improves from 45.5 Mb to 57.6 Mb by using reads from DeepConsensus (Supplementary Fig. 6 and Supplementary Table 11). The gene completeness analysis with asmgene suggested that the assembly with DeepConsensus has the same gene completeness (99.71%) as the assembly with pbccs reads (Supplementary Table 12). We further applied DeepConsensus to two additional non-human species and observed improvements to assembly contiguity for *Mus musculus* (Supplementary Fig. 7) and *Rana muscosa* (Supplementary Fig. 8). The well-annotated *M. musculus* reference genome allowed us to quantify that use of DeepConsensus reads in assembly improved NG50 from 22.4 Mb to 41.3 Mb (Supplementary Table 11), improved gene completeness from 99.44% to 99.53% (Supplementary Table 12) and improved complete multicopy genes from 84.33% to 85.03% (Supplementary Table 13).

To quantify assembly improvements that result from increased sequence depth or through increased read accuracy, we conducted additional assemblies from the same set of reads from pbccs and DeepConsensus. In one set, we used any read that reached the Q20 filter in DeepConsensus, while in the other set, we used any read that reached the Q20 filter in pbccs. These results showed that either increased sequence depth or increased read accuracy improved assembly properties independent of the other and that the combination of increased depth and accuracy resulted in the best results (Supplementary Fig. 9 and Supplementary Tables 14–19).

In summary, we observe consistent improvements in contiguity, correctness and completeness in assemblies generated with reads from DeepConsensus using either two or three SMRT Cells across human and multiple non-human species.

Using DeepConsensus reads improves variant-calling accuracy. To assess small-variant-calling improvements with DeepConsensus reads, we mapped pbccs and DeepConsensus reads to the GRCh38 reference with pbmm2 (ref. [26]) and called variants with DeepVariant[27] for four human genome samples. We used the DeepVariant v1.2 PacBio HiFi model for variant calling with pbccs reads, and we trained a custom DeepVariant model to call variants with DeepConsensus reads from chr1 to chr19, with HG002 GIAB v4.2.1 as the small-variant benchmark set.

For the variant-calling analysis, we used HG003, HG004, HG006 and HG007 samples. For all samples, we used the GIAB v4.2.1 benchmark set to evaluate the variants. We used hap.py[28] to assess the variants against the GIAB benchmark set. For each sample, we report the number of false-positive and false-negative variants in single nucleotide polymorphism (SNP) and INDEL categories.

In Fig. 4, we show the variant-calling performance of DeepVariant with DeepConsensus and pbccs reads for two and three SMRT Cells. Variant calling with DeepConsensus reads from two SMRT Cells had, on average, 25% fewer errors for HG003 and HG004 and 30% fewer errors for HG006 and HG007 samples than variants with pbccs reads (Fig. 4a,c,e and Supplementary Fig. 10). Similarly, variants derived from DeepConsensus reads from three SMRT Cells had, on average, 8% fewer total errors for HG003 and HG004 and 28% fewer errors for HG006 than variants with pbccs reads. Furthermore, we observed that SNP errors, on average, decreased 35% for two and 8% for three SMRT Cells of HG003 and HG004 samples (Fig. 4b,d,f). Similarly, INDEL errors, on average, decreased 15% for two and 6% for three SMRT Cells in variants with DeepConsensus reads for HG003 and HG004 samples (Supplementary Table 20). In summary, DeepConsensus improves variant-calling performance across samples in both SNP and INDEL categories with reads from two and three SMRT Cells.

Table 2 | Yield improvement

Dataset	Total reads	$Q > 20$	$Q > 30$	$Q > 40$	Perfect match
pbccs	90,432	88,561	47,767	10,207	4,395
DeepConsensus	103,093	96,260	60,490	19,485	9,291
Percent increase	14.00%	8.69%	26.64%	90.90%	111.40%

Polished HG002 chr20 11-kb reads from pbccs and DeepConsensus were quantified according to the total number of reads, reads at given thresholds ($Q_{concrdance} > 20$, 30 and 40) and reads that perfectly match the diploid assembly. Total reads represent the set of initial reads output by pbccs and DeepConsensus using $Q_{predicted} > 20$. The percentage increase in terms of yield achieved by DeepConsensus is listed for each category.

To assess the performance of DeepConsensus in various genomic contexts, we used the GIAB stratification files on assembly-based variant calling of HG006 (a sample unrelated to the training sample of HG002). This showed relatively consistent error reductions across dinucleotide repeats, homopolymers and high- and low-GC content regions (Supplementary Table 21). Assessments in centromeric and telomeric regions are difficult due to a lack of reliable assembly and GIAB truth sets. We quantified read yield at predicted Q20 or above and found a 30.3% increase in DeepConsensus yield relative to pbccs in the genomic contexts considered.

Use of longer reads improves yield, assembly and variant-calling accuracy. With higher consensus accuracy for HiFi reads, the number of passes can be reduced while maintaining accuracy (Fig. 2b), potentially allowing for sequencing of longer insert sizes while preserving the quality of downstream analyses. To test this, we sequenced a HG002 sample with 15-kb and 24-kb insert sizes, each with two SMRT Cells on the Sequel II System using Chemistry 2.2. We generated DeepConsensus reads for the 15-kb and 24-kb insert size (Supplementary Fig. 11a and Supplementary Table 22). Details on the library preparation protocol for 24-kb reads are provided in the Methods.

We show the improvements in genome assembly and variant calling we achieved with 24-kb reads compared to 15-kb reads of the HG002 sample (Supplementary Fig. 11). The hifiasm assembly with 24-kb reads achieved a higher contig NG50 of 34.05 Mb than the hifiasm assembly with 15-kb reads, with an NG50 of 24.81 Mb, although the assembly quality is higher with 15-kb (Q51.7) reads than with 24-kb reads (Q50.8; Supplementary Tables 23 and 24). The assembly-based variant-calling evaluation showed that the assembly with 24-kb reads has higher INDEL accuracy and comparable SNP accuracy than the assembly with 15-kb reads (Supplementary Tables 25 and 26). Notably, the multicopy gene completeness in the assembly with 24-kb reads was 80.91% compared to 76.93% in the assembly with 15-kb reads, while the single-copy gene completeness remained comparable (97.2% with 24-kb reads and 97.3% with 15-kb reads; Supplementary Tables 27 and 28). In variant calling with DeepVariant, the 24-kb DeepConsensus reads had fewer total errors than the 15-kb reads in HG002 chr20 (Extended Data Fig.1c and Supplementary Table 29).

In summary, the increased accuracy of DeepConsensus expands the window of experimental choices. This allows researchers to consider using longer reads for applications that disproportionately benefit, such as the assembly of genomes with high duplication rates, difficult to assemble regions (such as the major histocompatibility complex), phasing across a long gene or amplicon or variant detection in hard-to-map regions.

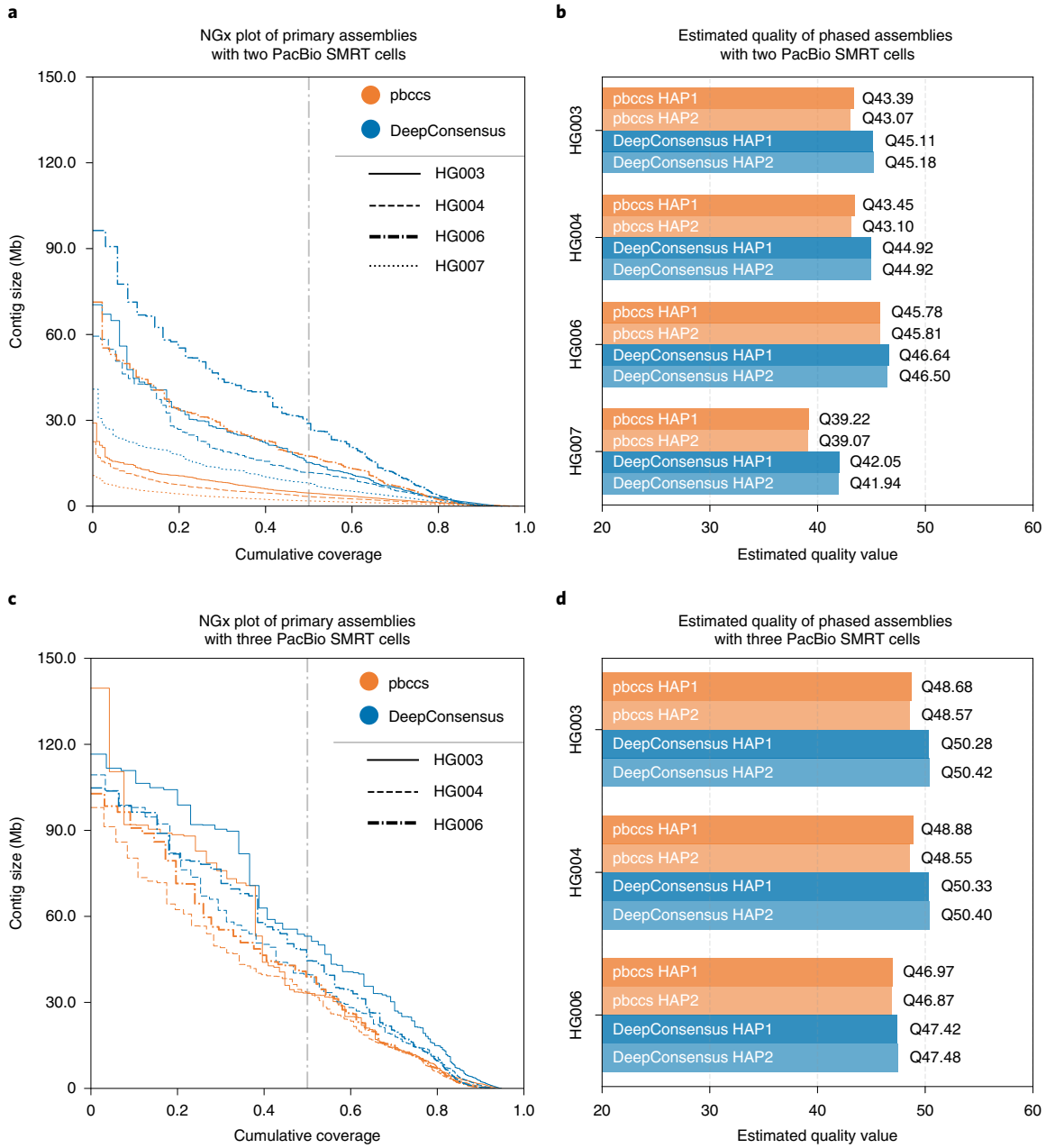


Fig. 3 | DeepConsensus improves the contiguity and quality of the genome assemblies generated with hifiasm. a, Contiguity of the hifiasm assemblies with reads from pbccs and DeepConsensus with two PacBio SMRT Cells. **b**, Reference-free estimated quality (using YAK) of the hifiasm phased assemblies with reads from pbccs and DeepConsensus with two PacBio SMRT Cells. **c**, Contiguity of the hifiasm assemblies with three PacBio SMRT Cells. **d**, Estimated quality of the hifiasm phased assemblies with three PacBio SMRT Cells.

Assessments of runtime. Subsequent to the development and benchmarking of DeepConsensus v0.1 presented in previous sections, we greatly improved speed in a new release, DeepConsensus v0.2, which was used for the *M. musculus* and *R. muscosa* assemblies.

DeepConsensus v0.2 processes the 11-kb HG002 data used for Table 1 at a rate of 0.883 zero-mode waveguide (ZMW) s^{-1} on a 16-thread central processing unit (CPU)-only machine (Google Cloud Platform (GCP) instances n2-standard-16) and at a rate of 2.93 ZMW s^{-1} on a 16-thread machine with an attached NVIDIA P100 graphics processing unit (GPU). This corresponds to ~16,000 CPU hours to process this SMRT Cell with CPU-only machines and ~5,000 CPU hours with an attached GPU.

Discussion

The correction of errors in sequencing data is fundamental to both the generation of initial data from a sequencer and to downstream analyses that assemble, map and analyze genomes[29–31]. We introduce a transformer-based consensus generation method that reduces errors in PacBio HiFi reads by 42% and increases yield of 99.9% accurate reads by 27%. We show that with existing downstream methods, the improved reads result in better assembly contiguity, completeness and accuracy and more accurate variant calling.

The problem of correcting errors from an MSA of repeated sequencing is a single example of a broader category of problems that analyze the alignment of similar sequences. The most similar adjacent applications are error correction of unique molecular

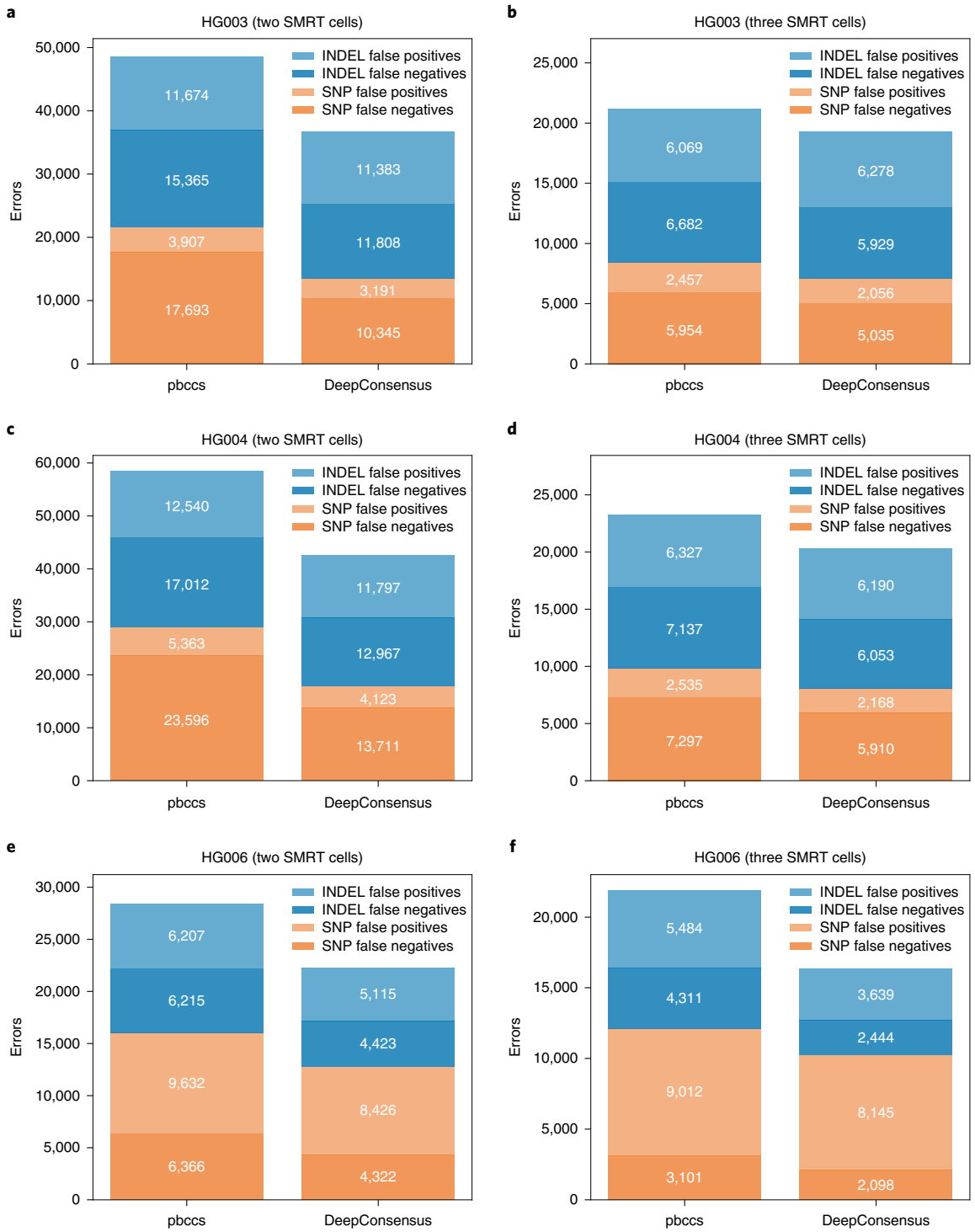


Fig. 4 | DeepConsensus improves variant-calling performance of DeepVariant. a–f, HG003 variant-calling performance of DeepVariant with pbccs and DeepConsensus reads from two and three SMRT Cells for HG003 (**a**,**b**), HG004 (**c**,**d**) and HG006 (**e**,**f**).

identifiers[32] and error correction of Oxford Nanopore Duplex reads. Genome assembly polishing, which uses alignments of sequences from many molecules, is a similar application[11,13,33]. DeepConsensus models could be trained for these applications with minimal changes to its architecture. The gap-aware loss function used in the GATE approach could have utility to broader MSA-related problems. For example, related work by Rao[17] demonstrated improved prediction performance across multiple tasks, including contact maps and secondary structure, and Avsec et al.[34] used a long-range Enformer to predict gene expression. These applications could potentially benefit from the incorporation of alignment-based loss used in DeepConsensus, or the DeepConsensus framework could be applied to similar problem areas.

DeepConsensus presents opportunities to alter experimental design to better use its improvements to accuracy. We demonstrate that DeepConsensus allows for longer read lengths while maintaining a high standard of read accuracy and yield. Certain applications, such as assembling difficult genome regions, may disproportionately

benefit from the use of longer reads. Additionally, because DeepConsensus learns its error model directly from training data, it allows a tighter coupling between library preparation, instrument iteration and informatics. DeepConsensus could be trained on data from a modified procedure or additional data stream to more accurately estimate the potential advantage of the new method, decreasing the chance that the modification's advantages might not be apparent due to optimization of the informatics to the older approach.

The improvements we demonstrate to assembly and variant calling use unmodified downstream tools (hifiasm) or tools with unmodified heuristics that use an adapted model (DeepVariant). Further iterating on the heuristics in these methods may allow them to take additional advantage of the DeepConsensus error profile or better use its higher yield of longer reads.

Future improvements to DeepConsensus include training with an expanded dataset that includes additional samples and chemistries, because our current training datasets only include Sequel II data from a few SMRT Cells. Supplementing training data with diverse species is an area of active development. There are substantial opportunities for improvements by refining the attention strategy (for example, AlphaFold2 uses a modified axial attention[35]) or by leveraging efficiency improvements to the transformer self-attention layer to consider wider sequence contexts[36–38]. We experimented with self-supervised pretraining for learning contextualized embeddings as an additional input for DeepConsensus, which is described in more detail in Methods. While these embeddings did not improve DeepConsensus performance, this may be due to the limited quantity of data used for pretraining. Using larger unlabeled databases for this pretraining is an area for future exploration. Investigating the trade-offs between model size and accuracy could also enable faster versions that preserve high accuracy. These and other improvements will enable DeepConsensus to help scientists realize the potential yield and quality of their sequencing instruments and projects.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-022-01435-7.

Received: 28 October 2021; Accepted: 15 July 2022;
Published online: 1 September 2022

References

1. Bentley, D. R. et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **456**, 53–59 (2008).
2. Travers, K. J., Chin, C.-S., Rank, D. R., Eid, J. S. & Turner, S. W. A flexible and efficient template format for circular consensus sequencing and SNP detection. *Nucleic Acids Res.* **38**, e159 (2010).
3. Mc Cartney, A. M. et al. Chasing perfection: validation and polishing strategies for telomere-to-telomere genome assemblies. *Nat. Methods* **19**, 687–695 (2022).
4. Altemose, N. et al. Complete genomic and epigenetic maps of human centromeres. *Science.* **376**, eabl4178 (2022).
5. Wenger, A. M. et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat. Biotechnol.* **37**, 1155–1162 (2019).
6. Olson, N. D. et al. precisionFDA Truth Challenge V2: calling variants from short- and long-reads in difficult-to-map regions. *Cell Genom.* **2**, 100129 (2022).
7. Nurk, S. et al. The complete sequence of a human genome. *Science* **376**, 44 (2022).
8. Logsdon, G. A., Vollger, M. R. & Eichler, E. E. Long-read human genome sequencing and its applications. *Nat. Rev. Genet.* **21**, 597–614 (2020).
9. Chin, C.-S. et al. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat. Methods* **10**, 563–569 (2013).
10. Chin, C.-S. et al. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat. Methods* **13**, 1050–1054 (2016).
11. Vaser, R., Sović, I., Nagarajan, N. & Šikić, M. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res.* **27**, 737–746 (2017).
12. Walker, B. J. et al. Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS ONE* **9**, e112963 (2014).
13. Shafin, K. et al. Haplotype-aware variant calling enables high accuracy in nanopore long-reads using deep neural networks. *Nat. Methods* **18**, 1322–1332 (2021).
14. Vaswani, A. et al. Attention is all you need. Preprint at https://doi.org/10.48550/arXiv.1706.03762 (2017).
15. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. Preprint at https://doi.org/10.48550/arXiv.1810.04805 (2018).
16. Dosovitskiy, A. et al. An image is worth 16×16 words: transformers for image recognition at scale. Preprint at https://doi.org/10.48550/arXiv.2010.11929 (2020).
17. Rao, R. et al. MSA transformer. Preprint at *bioRxiv* https://doi.org/10.1101/2021.02.12.430858 (2021).
18. The AlphaFold team. AlphaFold: a solution to a 50-year-old grand challenge in biology. *DeepMind* https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology
19. Mensch, A. & Blondel, M. Differentiable dynamic programming for structured prediction and attention. *Proc. 35th International Conference on Machine Learning* **80**, 3462–3471 (2018).
20. Nurk, S. et al. HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *Genome Res.* **30**, 1291–1305 (2020).
21. Lal, A. et al. Improving long-read consensus sequencing accuracy with deep learning. Preprint at *bioRxiv* https://doi.org/10.1101/2021.06.28.450238 (2021).
22. Cheng, H., Concepcion, G. T., Feng, X., Zhang, H. & Li, H. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nat. Methods* **18**, 170–175 (2021).
23. Gurevich, A., Saveliev, V., Vyahhi, N. & Tesler, G. QUAST: quality assessment tool for genome assemblies. *Bioinformatics* **29**, 1072–1075 (2013).
24. Li, H. et al. A synthetic-diploid benchmark for accurate variant-calling evaluation. *Nat. Methods* **15**, 595–597 (2018).
25. Wagner, J. et al. Benchmarking challenging small variants with linked and long reads. *Cell Genom.* **2**, 100128 (2020).
26. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
27. Poplin, R. et al. A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.* **36**, 983–987 (2018).
28. Krusche, P. et al. Best practices for benchmarking germline small-variant calls in human genomes. *Nat. Biotechnol.* **37**, 555–560 (2019).
29. Hu, J., Fan, J., Sun, Z. & Liu, S. NextPolish: a fast and efficient genome polishing tool for long-read assembly. *Bioinformatics* **36**, 2253–2255 (2020).
30. Warren, R. L. et al. ntEdit: scalable genome sequence polishing. *Bioinformatics* **35**, 4430–4432 (2019).
31. Morisse, P., Marchet, C., Limasset, A., Lecroq, T. & Lefebvre, A. Scalable long read self-correction and assembly polishing with multiple sequence alignment. *Sci. Rep.* **11**, 761 (2021).
32. Islam, S. et al. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nat. Methods* **11**, 163–166 (2014).
33. Shafin, K. et al. Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes. *Nat. Biotechnol.* **38**, 1044–1053 (2020).
34. Avsec, Ž. et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
35. Huang, Z. et al. CCNet: criss-cross attention for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, 603–612 (2020).
36. Choromanski, K. et al. Rethinking attention with performers. Preprint at https://doi.org/10.48550/arXiv.2009.14794 (2020).
37. Wang, S., Li, B. Z., Khabsa, M., Fang, H. & Ma, H. Linformer: self-attention with linear complexity. Preprint at https://doi.org/10.48550/arXiv.2006.04768 (2020).
38. Katharopoulos, A., Vyas, A., Pappas, N. & Fleuret, F. Transformers are RNNs: fast autoregressive transformers with linear attention. Preprint at https://doi.org/10.48550/arXiv.2006.16236 (2020).

Methods

Generation of 24-kb PacBio reads. DNA was extracted from HG002/NA24385 cell pellets (Coriell Institute) with the MasterPure Complete DNA and RNA Purification kit (Lucigen, MC85200) and sheared on a Megaruptor3 (Diagenode) at a speed of 30. SMRTbell libraries were constructed with a SMRTbell Express Template Prep kit 2.0 (PacBio, 100-038-900). Size selection was performed with BluePippin (Sage Science) with an 18-kb high-pass filter. Sequencing was performed on the Sequel II System using Chemistry 2.2 and 30-h movies.

Dataset preparation. For all SMRT Cells, we ran pbccs on the subreads to generate CCS reads. pbccs generates a prediction for the overall read quality for each CCS read, and reads below Q20 are filtered out of the final HiFi read set. For dataset generation, we did not apply any filtering based on read quality for the CCS reads, and reads of qualities were included for training and inference. To generate labels for each set of subreads, the CCS sequence predicted by pbccs was mapped to the HG002 diploid assembly. The coordinates of the primary alignment were used to extract the label sequence from the HG002 diploid assembly.

Subreads and labels were aligned to the corresponding CCS read. The CIGAR string from this alignment was used to match bases across the subreads and assign a label for each position. Subreads were broken up into 100-bp windows, and the corresponding label for each window was extracted from the full label sequence. In some cases, the label was longer than the subreads due to bases for which there was no support in the subreads.

Each subread base has associated PW and IPD values, and each set of subreads has four SN values, one for each of the four canonical bases. PW and IP values were capped at 9, and SN values were rounded to the closest integer and capped at 15.

Model and training. The transformer has emerged as the primary architecture for language understanding and generation tasks[14,15] and uses self-attention to efficiently capture long- and short-range interactions between words crucial for understanding text. In recent work, this capability has been successfully used to improve modeling of protein sequences[39].

We trained a six-layer encoder-only transformer model with a hidden dimension of 560 and two attention heads in each self-attention layer. The inner dimension of the feedforward network in each encoder layer is 2,048. The model considers 100 bases at a time from the full subreads, and the input at each position contains subread sequences and auxiliary features. The maximum number of subreads considered is 20. Auxiliary features include the PW and IPD measured by the basecaller, the SN ratio for the sequencing reaction, the strand of each subread and the sequence of the CCS read as predicted by pbccs. Each feature type is embedded using a separate set of learned embeddings, which are trained jointly with the model. An embedding size of two is used for the subread strand, and all other embeddings are of size eight. We used positional encodings that were a mix of sampled sine and cosines, as defined in the transformer[14]. For training, the Adam optimizer[40] was used with a learning rate of 1×10^{-4}, and input, attention and ReLU dropout values were set to 0.1. Our implementation builds off the one provided in the Tensorflow Model Garden.

For some examples, there exists a base in the label for which there is no evidence in any of the subreads. The predicted sequence for such examples would be longer than the input sequence length. The transformer–encoder block outputs an encoding for each input token. In natural language applications, variable-length prediction is implemented using a decoder block, which is not constrained in the number of outputs. For consensus generation, we did not use a decoder block due to computational constraints. To allow for variable-length prediction using only the encoder, we added a fixed number of padding tokens to the input sequence for each window. This allows the model to predict sequences longer than the subread sequences by replacing some of the padding tokens with additional bases.

The outputs from the encoder block are independently decoded using a shared feedforward layer with softmax activation. At each position, we predicted a distribution over the vocabulary, which consists of the four canonical bases (A, C, T and G) and an additional token to represent alignment gaps or padding, which we denote as '$'.

For training, we used chr1–chr18 from 11-kb PacBio Sequel II sequencing of HG002 (ref. [5]), an extensively characterized genome curated by GIAB[41]. Truth labels for HG002 were derived from a HG002 diploid assembly (Data availability). Only training examples where the truth label could be uniquely mapped to a CCS read with pbmm2 were used, and non-unique mappings between truth and CCS were discarded. Chr21 and chr22 were used for tuning model parameters, and chr19 and chr20 were held out entirely during training and used for final assessment. For additional full holdouts, we used PacBio Sequel II sequencing of HG003, HG004, HG006 and HG007.

Models were trained for 50 epochs on 128 core v3 tensor processing units (TPUs) with a batch size of 256 for each core. Five models were trained with the production settings, and we chose the checkpoint with lowest loss on the tuning data. A custom gap-aware alignment loss was used, which is described in more detail in the following section. We call the combination of the gap-aware loss with transformer–encoder architecture GATE.

Loss function. Given an input MSA consisting of subreads and a consensus read and auxiliary features, the output of the transformer is a sequence $y = y_1 y_2 ... y_N$ of probability distributions over the five-letter alphabet $N = \{A, T, C, G, \$\}$, where \$ refers to an empty character to model possible insertion errors in the HiFi consensus or padding. In other words, each y_i is a probability distribution of non-negative entries that satisfies $y_i(A) + y_i(T) + y_i(C) + y_i(G) + y_i(\$) = 1$. At inference time, the predicted nucleotide sequence $z = z_1 z_2 ... z_N$ is simply obtained by keeping the character with largest probability at each position, that is, $z_i = \text{argmax}_{a \in N} y_i(a)$ and removing the \$ character from the resulting sequence. At train time, when parameters of the transformer-based model are updated, we need to define a loss function $\text{loss}(y, t)$ differentiable with respect to the transformer output y given the correct nucleotide sequence $t = t_1 t_2 ... t_M$ (notice that the lengths N of the transformer output and M of the correct nucleotide sequence may differ due to possible insertion or deletion in the consensus read). If we know that a given position $1 \le i \le N$ of the transformer output should predict the nucleotide at position $1 \le j \le M$ of the true sequence, then it is natural to use the cross-entropy loss $\text{loss}_{\text{CE}}(y_i, t_j) = -\log y_i(t_j)$ to assess how good the prediction is. However, we need to choose which position of y predicts which position of t. For that purpose, we formally define an alignment of length k as an increasing subset of k positions

$$\pi = \{1 \le \pi(y, 1) < \pi(y, 2) < ... < \pi(y, k) \le N, 1 \le \pi(t, 1) < \pi(t, 2) < ... < \pi(t, k) \le M\}$$

in both y and t, such that position $\pi(y, v)$ in y predicts position $\pi(t, v)$ in t, for $v = 1, ..., k$. Given such an alignment, positions $\pi(\bar{y})$ of y not in the alignment correspond to insertion errors, and ideally the prediction in those positions should be \$ so that they are removed from the prediction at test time. For those positions, we therefore use the cross-entropy loss $\text{loss}_{\text{CE}}(y_i, \$)$. However, positions $\pi(\bar{t})$ of t not in the alignment correspond to deletion errors, that is, nucleotides in the correct sequence that are missed in the MSA. For those errors, we consider a fixed error $\gamma > 0$, which is a parameter to be tuned. In total, given an alignment π, the total loss is defined as the sum of cross-entropy losses over aligned positions and insertion/mutation losses

$$\text{loss}_\pi(y, t) = \sum_{v=1}^{k} \text{loss}_{\text{CE}}(y_{\pi(y,v)}, t_{\pi(t,v)}) + \sum_{v \in \pi(\bar{y})} \text{loss}_{\text{CE}}(y_v, \$) + \sum_{v \in \pi(\bar{t})} \gamma.$$

This loss depends on the arbitrary alignment π, which ideally should be chosen as a function of y and t so that the total loss is small. We therefore finally define the alignment loss as a (smooth) minimum over π, $\text{loss}_\epsilon(y, t) = -\epsilon \log(\sum_\pi e^{-\text{loss}_\pi(y,t)/\epsilon})$, where $\epsilon \ge 0$ is a parameter to control how suboptimal alignments contribute to the loss. At the limit $\epsilon = 0$, we simply keep the best alignment $\text{loss}_0(y, t) = \min_\pi \text{loss}_\pi(y, t)$, and taking $\epsilon > 0$ allows us to create a smoother loss function to better align y and t. This loss is a particular case of the losses studied previously[19], and we follow this approach to derive an efficient implementation to compute the loss and its gradient in y using differentiable dynamic programming, with a specific wavefront formulation to accelerate the computation on GPUs and TPUs.

Experiments with self-supervised pretraining. Inspired by the success of self-supervision in natural language processing (NLP), we used two pretraining tasks, predicting masked bases in the input sequence and predicting whether two fragments are contiguous, to learn contextualized representations of bases in a given subread. We used the same model architecture and training data described earlier, masked 15% of bases in the input sequence, varied the number of transformer layers, the hidden dimension, the number of multiattention heads and the sequence lengths and omitted the next-fragment pretraining objective. Although the pretrained model was effective in predicting masked bases in the held-out data, the contextualized embeddings learned in pretraining did not improve the accuracy of consensus generation, which is likely due to the limited amount of unlabeled subreads used for pretraining. Pretraining was removed from subsequent experiments to reduce the memory footprint of the final model and save computational resources.

Output FASTQ generation. DeepConsensus predictions for each 100-bp window were joined together, and \$ tokens were removed to produce the final sequence that was output to FASTQ. Predicted base quality scores were generated from the output distribution at each position. The raw quality score for each base, q_i, was computed as follows, where y_i is the output distribution at position i: $q_i = -10\log_{10}(1 - \max(y_i))$. Each raw quality score was rounded to the closest integer and capped at a maximum value of 60 to produce the final base quality score, Q_i. Final base qualities were used to compute an overall read quality, Q_{pred}, in the following calculation, which sums over all positions in the predicted sequence: $Q_{\text{pred}} = -10\log_{10} \sum_i 10^{(-Q_i/10)}$. Reads with an overall predicted quality above 20 were written to the final output FASTQ along with the corresponding quality string.

Analysis methods. *Assessing read accuracy.* HG002 11-kb predictions were mapped to a high-quality HG002 diploid assembly[20]. For each primary alignment, the calculate_identity_metrics.py script was used to compute identity, which is defined as

$$\text{identity} = \text{matches}/(\text{matches} + \text{mismatches} + \text{deletions} + \text{insertions}).$$

The read identity values were used to compute the concordance read qualities, $Q_{\text{concordance}}$, which are computed as Phred-scaled scores of the identity: $Q_{\text{concordance}} = -10\log_{10}(1 - \text{identity})$. Reads with identity scores of 1 were separately categorized as having a 'perfect match.' Subread counts were determined using the np tag (number of full-length passes). The np tag was extracted from the consensus reads BAM output by pbccs.

We also used the bamConcordance tool, which reports the concordance between a read and a reference sequence along with error counts for each read. Error counts are broken down into five categories: mismatches, homopolymer insertions and deletions and non-homopolymer insertions and deletions. We used the bamConcordance output to assess the quality of reads and calculate the percentage error reduction across different categories.

Generating phased diploid assemblies with hifiasm. We used hifiasm version 0.15.3-r339 to generate phased assemblies and the default hifiasm parameters, which have duplication purging on for the phased assemblies. We converted the primary assembly graph to get the primary assembly sequence and each of the haplotype graphs to generate the assembly sequences for each haplotype. Detailed execution parameters and commands are provided in the Supplementary Notes.

Reference-free assembly quality estimation with YAK. We used YAK version 0.1-r62-dirty to derive estimated quality of the assemblies. For each sample, we generated a *k*-mer set with $k = 31$ from Illumina short reads of the same sample. We then ran YAK to determine the quality of each haplotype sequence that we generated during the hifiasm assembly generation process. YAK reports a *Q* value for assemblies, which is a Phred-scale contig base error rate derived by comparing 31-mers in contigs and 31-mers in short reads of the same sample. We report the balanced_qv value reported by YAK as the estimated quality value of the assembly. The parameters and commands used are provided in the Supplementary Notes.

Assembly-based small-variant-calling assessment using dipcall. We used dipcall version 0.3 to derive small variants from the phased assemblies. Dipcall aligns the contigs to a reference sequence and derives a set of variants from the contig to the reference alignment. We then compared the derived small variants against the GIAB truth set of the associated sample. For all male samples, we used the –x hs38.PAR.bed parameter as suggested in the documentation of dipcall.

To assess the small variants derived from all samples, we used GRCh38 as a reference and GIAB v4.2.1 as the truth set for small variants. All truth sets are the latest available truth sets from GIAB for the associated sample. We used hap.py to assess the quality of the variant calls. Commands and parameters used to run dipcall are provided in the Supplementary Notes.

Gene completeness assessment with asmgene. We used asmgene version v2.21 to determine the gene completeness of the assemblies. First, we aligned the Ensembl cDNA sequences release 102 to the GRCh38 reference genome using minimap2 (v2.21) and found 35,374 single-copy genes and 1,253 multicopy genes in the reference. For each sample, we then aligned the sample cDNA sequences to each of the haplotype sequences of the assemblies and derived how many single-copy genes remained single copy (full_sgl reported by asmgene) and how many were duplicated (full_dup reported by asmgene). Similarly, we reported how many multicopy genes remained multicopy in the assembly (dup_cnt reported by asmgene). We derived the following metrics to assess the gene completeness of the assemblies:

$$\text{gene completeness}\ (\%) = \frac{\text{full\_sgl}_{\text{assembly}}}{\text{full\_sgl}_{\text{GRCh38}}},$$

$$\text{duplication}\ (\%) = \frac{\text{full\_dup}_{\text{assembly}}}{\text{full\_sgl}_{\text{GRCh38}}},$$

$$\text{complete multicopy}\ (\%) = \frac{\text{dup\_cnt}_{\text{assembly}}}{\text{dup\_cnt}_{\text{GRCh38}}}\ \text{and}$$

$$\text{missing multicopy}\ (\%) = \frac{\text{dup\_cnt}_{\text{GRCh38}} - \text{dup\_cnt}_{\text{assembly}}}{\text{dup\_cnt}_{\text{GRCh38}}}.$$

Detailed commands and parameters of asmgene are provided in the Supplementary Notes.

Assembly statistics with QUAST. We used QUAST version v5.0.2 to derive assembly N50, NG50, total assembly size and genome completeness of the assembly. QUAST is a reference-based assembly evaluation method that uses a reference sequence of the same sample or to determine the quality of the assembly. For our analysis, we used GRCh38 as the reference for each assembly.

We used N50, which is the sequence length of the shortest contig at 50% of the total assembly length, to determine contiguity of the assembly. NG50 is the sequence length of the shortest contig at 50% of the estimated genome length. For our human genome assemblies, we used GRCh38 as the reference sequence, so we used 3,272,116,950 bp (3.2 gigabases) as the estimated genome length to derive NG50. We only report N50, NG50, total assembly length and genome completeness against GRCh38 from the QUAST report. Detailed parameters and commands are provided in the Supplementary Notes.

Variant calling. DeepVariant performs variant calling in three stages: make_examples, call_variants and postprocess_variants. The make_examples stage identifies candidate variants and generates input matrices containing pileup information. call_variants runs the input matrices through a neural network model, and postprocess_variants converts the neural network outputs to a variant call and outputs a Variant Call Format (VCF) file.

We used the latest DeepVariant model for PacBio HiFi data, v1.2, to call variants in pbccs predictions. Polished DeepConsensus reads or pbccs HiFi reads were aligned to GRCh38. This model was fine tuned from the Illumina WGS v1.2 model using PacBio HiFi sequencing reads generated using pbccs. Because DeepConsensus reads display different error characteristics than pbccs reads, we fine tuned a DeepVariant model for DeepConsensus. This model was also initialized from the v1.2 Illumina whole-genome sequencing (WGS) model, and the training data consisted of 11-kb and 24-kb Sequel II reads for HG002. We mixed both phased and unphased reads for the training similar to what is done for the v1.2 PacBio model. Chr1–chr19 were used for training, chr21 and chr22 were used for tuning, and chr20 was held out entirely.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Sequencing data, predictions and analysis files are available at https://console.cloud.google.com/storage/browser/brain-genomics-public/research/deepconsensus/publication.

Code availability

Code and pretrained models are available at https://github.com/google/deepconsensus. Sequencing data are available from the following sources:

- Sequel II data from Novogene[42] at https://console.cloud.google.com/storage/browser/brain-genomics-public/research/sequencing
- 15-kb HG002 and 24-kb HG002 reads from PacBio at https://console.cloud.google.com/storage/browser/brain-genomics-public/research/deepconsensus/publication/sequencing
- Sequel II data from PacBio at https://downloads.pacbcloud.com/public/dataset/HG002_SV_and_SNV_CCS/
- HG002 diploid assembly at https://obj.umiacs.umd.edu/marbl_publications/hicanu/hg002_hifi_hicanu_combined.fasta.gz

References

39. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2021).
40. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://doi.org/10.48550/arXiv.1412.6980 (2014).
41. Zook, J. M. et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci. Data* **3**, 160025 (2016).
42. Baid, G. et al. An extensive sequence dataset of gold-standard samples for benchmarking and development. Preprint at *bioRxiv* https://doi.org/10.1101/2020.12.11.422022 (2020).

Acknowledgements

We thank F. Liu of the Google TensorFlow Model Garden team for improving our use of open-source implementation of the transformer architecture.

Author contributions

G.B., P.-C.C. and A.C. conceived the study. G.B. and D.E.C. wrote DeepConsensus and trained models. G.B., D.E.C., K.S., T.Y., M.N. and A.B. performed experiments with DeepConsensus reads and made figures and documentation. F.L.-L., Q.B. and J.-P.V. conceived and implemented the alignment loss strategy, which D.E.C. integrated into DeepConsensus. A.M.W., W.J.R. and A.T. provided insight into PacBio data, identified areas for improvement, suggested informative features and provided code for preprocessing and evaluation. W.A. experimented with embedding strategies. A.K. and A.T. contributed to efficient processing of PacBio reads. H.Y. coordinated data acquisition and research agreements. J.-P.V., A.V., C.Y.M., M.N., P.-C.C. and A.C. provided guidance on experimental design, architecture and code review. G.B., D.E.C., K.S., T.Y., F.L.-L., Q.B., A.M.W., W.J.R., M.N., J.-P.V., A.V., C.Y.M., P.-C.C. and A.C. wrote the paper.

Competing interests

G.B., D.E.C., K.S., T.Y., F.L.-L., Q.B., A.B., M.N., H.Y., A.K., W.A., J.-P.V., A.V., C.Y.M., P.-C.C. and A.C. are employees of Google LLC and own Alphabet stock

as part of the standard compensation package. A.M.W., A.T. and W.J.R. are full-time employees and shareholders of Pacific Biosciences. This study was funded by Google LLC.

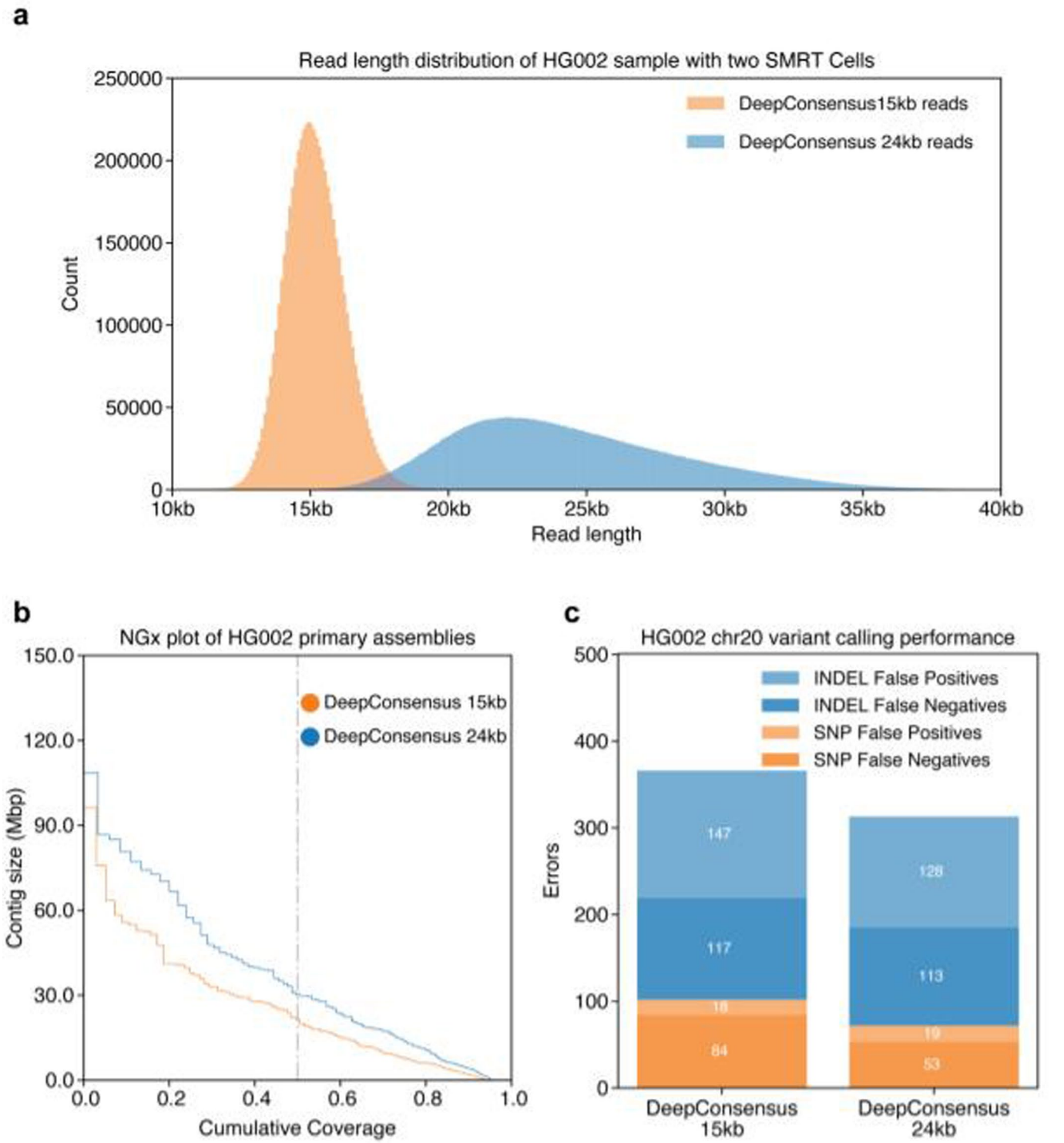
Additional information

Extended data is available for this paper at https://doi.org/10.1038/s41587-022-01435-7.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-022-01435-7.

Correspondence and requests for materials should be addressed to Andrew Carroll.

Peer review information *Nature Biotechnology* thanks Justin Zook, Andrey Bzikadze and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.



Extended Data Fig. 1 | DeepConsensus with longer reads improves genome assembly contiguity. (**a**) HG002 read length distribution for 15kb and 24kb DeepConsensus reads from two SMRT Cells. (**b**) Contiguity of the HG002 hifiasm assembly with 15kb and 24kb DeepConsensus reads from two SMRT Cells. (**c**) HG002 variant calling performance for 15kb and 24kb reads from DeepConsensus for two SMRT Cells.

nature research

Corresponding author(s): Andrew Carroll

Last updated by author(s): July 20, 2022

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Data generated for this study was produced by PacBio instrument sequencing an analysis with pbccs v4.2.0 (https://github.com/PacificBiosciences/ccs)
Data analysis	Full commands and versions for all programs run are found in the Software Commands section of supplementary material

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Sequencing data, predictions, and analysis files are available at:
https://console.cloud.google.com/storage/browser/brain-genomics-public/research/deepconsensus/publication

Sequencing data is available from the following sources:
Sequel II data from Novogene : https://console.cloud.google.com/storage/browser/brain-genomics-public/research/sequencing
15kb HG002 and 24kb HG002 reads from PacBio: https://console.cloud.google.com/storage/browser/brain-genomics-public/research/deepconsensus/publication/sequencing

Accession identifiers for non-human PacBio SMRT sequencing:
Rana muscosa: SRR11606868, Mus musculus: SRR11606870, Zea mays: SRR11606869

Sequel II data from PacBio: https://downloads.pacbcloud.com/public/dataset/HG002_SV_and_SNV_CCS/
HG002 diploid assembly:
https://obj.umiacs.umd.edu/marbl_publications/hicanu/hg002_hifi_hicanu_combined.fasta.gz

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	All available sequence datasets were used for HG002-HG007. (an initial HG002 PacBio sequencing run from earlier publications), 3 flowcells of new, long insert HG002 was provided by PacBio. We contracted with Novogene for 3 flowcells each of HG003, HG004, HG006, and HG007 in an earlier study (described in: https://www.biorxiv.org/content/10.1101/2020.12.11.422022v1).
Data exclusions	A single flowcell of HG007 was excluded from analysis due to a file corruption issue in the file received from the sequencing vendor. The file corruption issue prevented all downstream analysis from this single flowcell.
Replication	Results were evaluated across the full genome for every available human sample not used in model training (HG003, HG004, HG006, HG007) with concordant findings for genome assembly and variant calling. Results were evaluated across three non human species for which PacBio sequencing data was publicly available at the subread level (mouse, frog, and maize) with concordant findings for genome assembly.
Randomization	Randomization was not relevant for this study. The machine learning training followed standard practices for train-tune-and test data sets. Training is only conducted with Sequel II, Chemistry V1 of HG002. All other samples evaluated (HG003, HG004, HG006, and HG007) were never trained on.
Blinding	Investigators were not blind to groups, as all data was pooled together and publicly available. The machine learning training followed standard practices for separating train, tune, and test data sets.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☐	☒ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Eukaryotic cell lines

Policy information about cell lines

Cell line source(s)	HG002-HG007 cell lines from the Coriell Institute
Authentication	The full genome of the cell lines were sequenced and aligned back to a truth set for
Mycoplasma contamination	The cell lines were not tested for Mycoplasma contamination. However, only the germline DNA content of the cell lines are required, not any transcriptional or other cell phenotype.
Commonly misidentified lines (See ICLAC register)	No commonly misidentified lines were used.