Knowledge distillation for fast and accurate DNA sequence correction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Accurate genome sequencing can improve our understanding of biology and the genetic basis of disease. The standard approach for generating DNA sequences from PacBio instruments relies on HMM-based models. Here, we introduce Distilled DeepConsensus - a distilled transformer–encoder model for sequence correction, which improves upon the HMM-based methods with runtime constraints in mind. **Distilled DeepConsensus is 1.3x faster and 1.5x smaller** than its larger counterpart while **improving the yield of high quality reads (Q30) over the HMM-based method by 1.69x (vs. 1.73x for larger model)**. With improved accuracy of genomic sequences, Distilled DeepConsensus improves downstream applications of genomic sequence analysis such as **reducing variant calling errors by 39% (34% for larger model)** and **improving genome assembly quality by 3.8% (4.2% for larger model)**. We show that the representations learned by Distilled DeepConsensus are similar between faster and slower models.

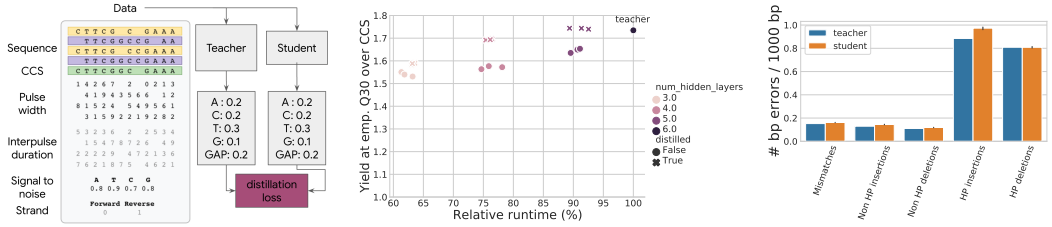
1 Introduction

Genome sequencing is widely used to advance our understanding of biological processes, the genetic basis of diseases, and is increasingly used for personalized medicine directly in the clinic (1). Thus, obtaining accurate genome sequences is of paramount importance. Over the years, sequencing platforms have been developed by Illumina, Pacific Biosciences (PacBio), and Oxford Nanopore, that have had either short and accurate reads or long and noisy reads. Recently, PacBio developed a novel long read HiFi sequencing technology (2) that has led to new state-of-the-art performance in variant calling (3) and the first telomere-to-telomere human assembly (4). While highly accurate, some errors still exist and here we focus on whether we can use machine learning, and in particular transformers, to further improve the accuracy of the genomic sequences with on-device runtime constraints in mind.

PacBio HiFi sequencing works by circularizing DNA, allowing for multiple sequencing passes over the same molecule. This produces a set of noisy sequences/subreads of the same region. The subreads are then combined into a consensus sequence/read using a hidden Markov model (HMM) (2). In order to further improve upon the accuracy of the HMM-based method for correct DNA sequence generation, DeepConsensus - an encoder-only transformer model has been developed (5). However, the accuracy and yield improvements of DeepConsensus come with latency and model-size costs. For example, unlike HMMs, transformer models are too large to be run on sequencing platforms with resource constraints. As sequencing throughput increases, the problem compounds.

In this work, we use Knowledge Distillation (KD) (6) to improve the runtime of the transformer-based DeepConsensus model, using techniques similar to (7; 8; 9). Our model is much smaller and faster than the original model, with slight performance degradation. Importantly, these runtime boosts are critical for enabling real time on-sequencing-device inference. In other domains of biology, as the amount of data increases, such as sequencing millions of single cells, and model size increases (10),

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) (b) (c)

Figure 1: (a) Overview of the Distilled DeepConsensus model. To train Distilled DeepConsensus, in addition to the alignment-based loss function we compute the distillation loss between teacher and student probabilities per position. (b) Performance versus runtime curve for Distilled DeepConsensus student models and undistilled models. The performance is calculated relative to pbccs (HMM-based model). (c) Error profiles of the teacher and distilled student models.

the runtime considerations may become increasingly important. This work explores initial steps in this direction and our contributions are as follows:

1. We show that knowledge distillation can improve model accuracy, resulting in a faster model without a significant degradation in performance as compared to a large model.
2. We demonstrate relative performance parity between larger and smaller models on downstream tasks like variant calling and *de novo* assembly generation.
3. We show that the representations learned by larger and smaller models are similar.

2 Model

DeepConsensus (5) takes as input a tensor of length 100bp containing information about the subreads such as the noisy subread sequences, strand direction, pulse width, interpulse duration as well as general information such as signal-to-noise ratio of each nucleotide and the consensus sequence given by pbccs HMM-based algorithm (Fig. 1a). The goal is to predict the correct consensus sequence from the noisy subread sequences and the other auxiliary information. Since this problem is similar to sequence-to-sequence tasks, Transformers (11) are leveraged for this problem.

Both the teacher and student models are encoder-only transformers. The teacher has six encoder stacks. We vary the number of encoder stacks from 3 to 5 to obtain smaller (student) Distilled DeepConsensus models of different sizes. Each encoder stack is composed of a self-attention layer with hidden dimension of 280 and 2 attention heads and a feedforward network with a filter size of 2048. The output of the entire encoder block, which represents the contextual embedding of each base, is decoded with the feedforward layer followed by a softmax function, that for each base outputs the probability of A, C, T, G or a gap. We use this simpler decoder to save computational resources.

For our labels, we leverage a high quality assembly of the human genome (T2T-CHM13), recently published by the Telomere-to-Telomere consortium (4), as part of the effort to obtain the complete sequence of the human genome and use data (2 SMRT Cells) from a CHM13 sample with 24-kb insert sizes from (5).

We use an alignment-based loss function, which is based on differentiable dynamic programming (12), $L(y, p)$ to train the teacher model as in (5), where $y = (y_1, ..., y_m)$ is the true sequence and $p = (p_1, ..., p_n)$ are the predicted probabilities with p_i representing the probability over A, C, T, G or a gap classes.

In order to train smaller student models from the teacher, in addition to the alignment-based loss function, we use a typical knowledge distillation loss (6), to update the model weights of the student:

$$L_{total} = L(y, p^{student}) + \alpha L_{KD}(p^{student}, p^{teacher}), \tag{1}$$

where $p^{student}$ and $p^{teacher}$ are the probabilities output by the student and the teacher models, respectively, and α is determined using a hyper-parameter search. We explore both the Kullback–Leibler

(KL) divergence and the mean squared error (computed per base and averaged across positions) for L_{KD}. In addition, we initialize the student layers from the teacher layers, since this was found to be beneficial in (7).

3 Results

3.1 Performance vs. runtime trade-off

We measure the model performance by comparing the yield at empirical Q30 (number of bases in reads with Q30 or higher quality score, i.e. identity of 0.999 or higher) produced by DeepConsensus models versus pbccs. This metric (yield at empirical Q30 over CCS) indicates how many more high quality bases we obtain with a particular model as compared to pbccs. The models are trained on the data collected from CHM13 cell line (chromosomes 1-18, 2 SMRT Cells), hyper-parameters are optimized using CHM13 cell line (chromosomes 21-22, 2 SMRT Cells), and the models are tested on a completely held-out genome HG002 (chromosome 20, 1 SMRT Cell). Here, we use the KL loss for matching teacher and student probabilities and initialize the student model from top layers of the teacher. Model runtime is measured using number of seconds on a Google Cloud VM with Intel 32 CPU cores (n1-standard-32).

In order to assess whether distillation is useful, we also train student models of the same size but without the distillation loss, only with $L(y, p_{student})$. We show that distillation consistently results in improved yield at empirical Q30 over CCS across different student architectures (3, 4, 5 layers) and replicates (Fig. 1b).

Some student models, for example the 5 layer student model, **achieve a similar yield at empirical Q30 over CCS as the teacher while taking 90% of teacher's runtime**. For the remaining analysis, we focus on the student model with 4 layers, 1.3x faster and 1.5x smaller (in terms of # parameters) than the teacher, and slight performance degradation (1.69 vs 1.73) compared to the teacher.

3.2 Teacher versus student error profiles

We analyze whether the teacher and the student make similar kinds of errors. We quantify the number of base pairs with mismatches, insertions and deletions per read and additionally subdivide errors by their sequence context. It has been shown that PacBio HiFi sequencing has higher error rates within homopolymer (HP) regions (2), which are regions with consecutive repeating bases (e. g. AAAA), and thus we split indel errors by HP and non-HP context. Fig. 1c shows that the error types made by the teacher and the student are similar with HP indels and in particular HP insertions for the student being the major error type.

3.3 Performance on downstream tasks: variant calling and assembly

Variant calling We perform variant calling analysis using a custom DeepVariant (13) model (warm-started from DeepVariant 1.4) trained on sequences from 2 SMRT Cells generated by DeepConsensus teacher and student models respectively and evaluated on held-out chromosome 20. For sequences output by pbccs, we use DeepVariant 1.4 (trained on pbccs output). Fig. 2 shows that the student model has 39% error reduction in variant calling in comparison to pbccs with the student being slightly better than the teacher on this task (34% reduction for larger model).

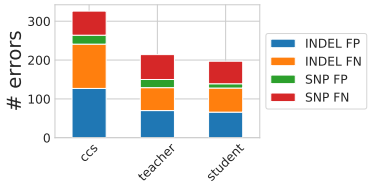


Figure 2: Distilled DeepConsensus improves variant calling: HG002 (chromosome 20) variant-calling performance of DeepVariant with pbccs, DeepConsensus and Distilled DeepConsensus reads from 2 SMRT Cells.

Model	Mean QV (↑)	Complete (%, ↑)	Duplicated (%, ↓)
pbccs	49.8625	96.893	0.572
Teacher	**51.999**	**97.094**	**0.502**
Student	51.782	97.063	0.516

Table 1: Mean quality, gene completeness, and false duplication rates of the genome assemblies generated with pbccs, DeepConsensus and Distilled DeepConsensus reads from 2 SMRT Cells.

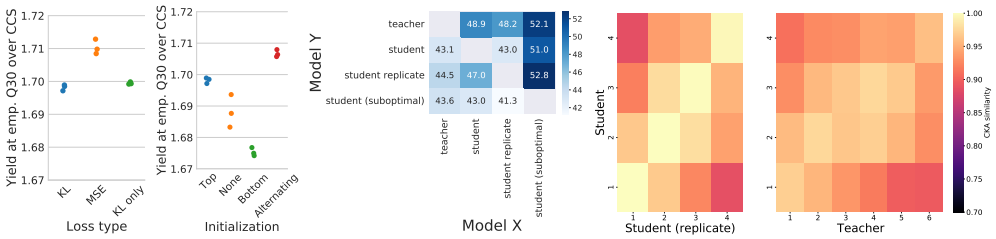
Assembly analysis Given increased yield and higher quality reads generated by the DeepConsensus student model and the larger teacher model, we compare the student and the teacher models in terms of *de novo* assembly generation. Phased assemblies were generated with hifiasm assembler using reads (from 2 SMRT Cells) output by pbccs, and DeepConsensus student and teacher models, similar to (5). In Table 1, we report the estimated qualities, using YAK (14), of the assemblies and show that the mean quality value (across haplotypes) is similar between the teacher and the student models with both being better than pbccs (3.8% and 4.2% for student and teacher, respectively). Additionally, both the teacher and the student models increase assembly gene completeness and decrease false gene duplication rate.

4 Analysis

We performed a study of distillation training options to evaluate which factors contributed the most to good performance in the student models. We experimented with three distillation losses: alignment to ground truth and Kullback Leibler (KL) divergence with the teacher probabilities, alignment to ground truth and MSE with the teacher probabilities, and just KL with the teacher probabilities. Additionally, we experimented with four types of initialization schemes. The relative benefits of each can be seen in Fig. 3a and b. We found that MSE outperformed KL divergence, and that initializing the student network to "alternating" layers (1, 2, 4, 6 of the teacher) performed best among layer initialization schemes.

Fig. 3c shows a per-example agreement analysis (15) to better understand whether models with better overall performances are just correctly predicting more examples, or if they're correctly predicting different examples. The results show that the better models are more accurate only slightly more than the worse models when there is a disagreement. This means that the set of examples correctly predicted by the better models is a larger and different set than the set of examples correctly predicted by the worse models, but neither is as a superset of the other.

To test whether teachers and students learn the same representations, we apply linear Centered Kernel Alignment (CKA) between pairs of layers across networks, following the methodology of (16). CKA computes a [0, 1]-valued similarity between two Gram matrices computed from two representations over the same sample of input examples (17). We see from the student vs. student plot (Fig. 3d, left) that the internal representations of the student model are very stable across different replicates. In the teacher vs. student plot (Fig. 3d, right), we find that layers in the student network correspond well with particular layers in the teacher. Interestingly, we find that despite initializing with top layers from the teacher ($1 \rightarrow 1, 2 \rightarrow 2, 4 \rightarrow 3, 6 \rightarrow 4$), the student layers after training have a different correspondence ($1 \rightarrow 1, 2$ or $4 \rightarrow 2, 4 \rightarrow 3, 6 \rightarrow 4$). This representational difference might also explain why the multiheaded attention of the student is different than the teacher: for example for layer 3, the student attention head looks an average of 1.8 base pairs different than the teacher for layer 4.



(a) (b) (c) (d)

Figure 3: a) Effect of loss choice on distillation. b) Effect of initialization on distillation. c) Per-example agreement between the teacher, students with the same hyperparameters but different replicates, and students with different less optimal hyperparameters (we take the lowest performing student from Fig. 3b. We measure the probability that model Y is strictly better (>, equality is omitted and thus probabilities do not add up to 100%) than model X (in terms of alignment identity) given that model X and Y disagree on the prediction (edit distance > 0). d) CKA analysis between student and a replicate (left), showing representations are stable and CKA between student and teacher (right), showing representations are similar.

References

[1] J. E. Gorzynski, S. D. Goenka, K. Shafin, T. D. Jensen, D. G. Fisk, M. E. Grove, E. Spiteri, T. Pesout, J. Monlong, G. Baid *et al.*, "Ultrarapid nanopore genome sequencing in a critical care setting," *New England Journal of Medicine*, vol. 386, no. 7, pp. 700–702, 2022.

[2] A. M. Wenger, P. Peluso, W. J. Rowell, P.-C. Chang, R. J. Hall, G. T. Concepcion, J. Ebler, A. Fungtammasan, A. Kolesnikov, N. D. Olson *et al.*, "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome," *Nature biotechnology*, vol. 37, no. 10, pp. 1155–1162, 2019.

[3] N. D. Olson, J. Wagner, J. McDaniel, S. H. Stephens, S. T. Westreich, A. G. Prasanna, E. Johanson, E. Boja, E. J. Maier, O. Serang *et al.*, "Precisionfda truth challenge v2: Calling variants from short and long reads in difficult-to-map regions," *Cell Genomics*, vol. 2, no. 5, p. 100129, 2022.

[4] S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman *et al.*, "The complete sequence of a human genome," *Science*, vol. 376, no. 6588, pp. 44–53, 2022.

[5] G. Baid, D. E. Cook, K. Shafin, T. Yun, F. Llinares-López, Q. Berthet, A. Belyaeva, A. Töpfer, A. M. Wenger, W. J. Rowell *et al.*, "Deepconsensus improves the accuracy of sequences with a gap-aware sequence transformer," *Nature Biotechnology*, pp. 1–7, 2022.

[6] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[8] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[9] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: a compact task-agnostic bert for resource-limited devices," *arXiv preprint arXiv:2004.02984*, 2020.

[10] Ž. Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley, "Effective gene expression prediction from sequence by integrating long-range interactions," *Nature methods*, vol. 18, no. 10, pp. 1196–1203, 2021.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] A. Mensch and M. Blondel, "Differentiable dynamic programming for structured prediction and attention," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3462–3471.

[13] R. Poplin, P.-C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, D. Newburger, J. Dijamco, N. Nguyen, P. T. Afshar *et al.*, "A universal snp and small-indel variant caller using deep neural networks," *Nature biotechnology*, vol. 36, no. 10, pp. 983–987, 2018.

[14] H. Cheng, G. T. Concepcion, X. Feng, H. Zhang, and H. Li, "Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm," *Nature methods*, vol. 18, no. 2, pp. 170–175, 2021.

[15] J. Shor, A. Jansen, W. Han, D. S. Park, and Y. Zhang, "Universal paralinguistic speech representations using self-supervised conformers," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3169–3173, 2022.

[16] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[17] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3519–3529.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The data is publicly available and the code can be found at [REDACTED FOR ANONYMITY] github repo.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] The license can be found at [REDACTED FOR ANONYMITY] github repo.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]