Simulation of ancient DNA sequences using transformer-based techniques.

[Poster/Poster demo/Talk submission]

Théo Boury [1,2], Jazeps Medina-Tretmanis [3], Maria Avila-Arcos [4], Emilia Huerta-Sanchez [3], Burak Yelmen [2,5], Flora Jay [2]

[1]Computer Science Department, Ecole Normale Supérieure de Lyon, France; [2]U Paris-Saclay, CNRS, INRIA, Laboratoire Interdisciplinaire des Sciences du Numérique, France; [3] Center for Computational Molecular Biology, Brown U, USA; [4] International Laboratory for Human Genome Research, U Nacional Autónoma de México, México; [5]Institute of Genomics, U of Tartu, Estonia.

Abstract. Ancient DNA (aDNA) sequences, that correspond to DNA extracted post-mortem from fossils or ancient skeletons, have become critical to understand the evolutionary histories of species and are necessary complements to modern DNA sequences. However, DNA drastically degrades through time resulting in aDNA of poorer quality than modern DNA. Thus, the robustness of modern DNA techniques should be tested before being applied to aDNA. Alternatively, novel methods have been designed specifically for aDNA and require to be tested in various degraded conditions. Such challenges makes the simulation of aDNA sequences of high interest. Few simulators exist to degrade sequences but, the most famous one, named Gargammel [6], is time-consuming and is hard to parameterize.
We developed a method able to simulate aDNA sequences corresponding to a given string of an undamaged DNA sequence. It is, to our knowledge, the first method that generates aDNA sequences based on language models. The method is based on an encoder transformer trained on couples of undamaged and aDNA. The relevance of the simulated data is checked through base pairs comparisons with the sequences obtained with Gargammel. In the future, such a method could become a good alternative to Gargammel with faster simulation time.

Keywords: ancient DNA, language model, sequence generation, pre-trained encoder transformer, K-Top sampling

1 Motivations

DNA studies have led to the possibility to infer the past evolutionary histories of species and populations. These include the size of the population, migration rates, selection pressures, and, more generally, any property that impacts the distribution through time of the ancestors of a sample of individuals. Such studies are well complemented with the use of DNA obtained from fossils, called aDNA, that carry direct information on past periods.
Nonetheless, due to the degradations, mostly deamination and contamination, aDNA sequences are heavily fragmented with a significant number of missing data and several wrongly called genotypes. Combined with the scarcity of aDNA data, it led to a real need for simulated aDNA sequences that, in addition, allow evaluation in a controlled environment.
A sequence-to-sequence aDNA simulator, able to damage sequences, already exists called Gargammel [6], based on physics and statistics. Gargammel is integrated into a more complete pipeline, already used to test the robustness of neural networks [3], that includes sequencing post-processing steps (trimming, mapping, calling). Nonetheless, its parameterization and computational time are prohibitive when large datasets are necessary. We propose an alternative: a sequence-to-sequence simulator based on transformers. Our learning steps are done from undamaged DNA sequences and their aDNA "translations". Our aDNA "translations" are obtained using Gargammel itself and the full post-processing pipeline, which circumvent the scarcity of aDNA data and allow us to focus on the approximation of Gargammel. As "translations" are only used for training purposes, a user does not require Gargammel when using our tool on their input sequences.
In addition, note that generation with transformers has already been achieved, but almost exclusively with a focus on natural languages. In parallel, a couple of encoder-only transformers were pre-trained on the DNA language mainly for classification purposes. Our current work aims to fill this gap as, for now, only a one-month-old preprint tackles DNA generation. [8]

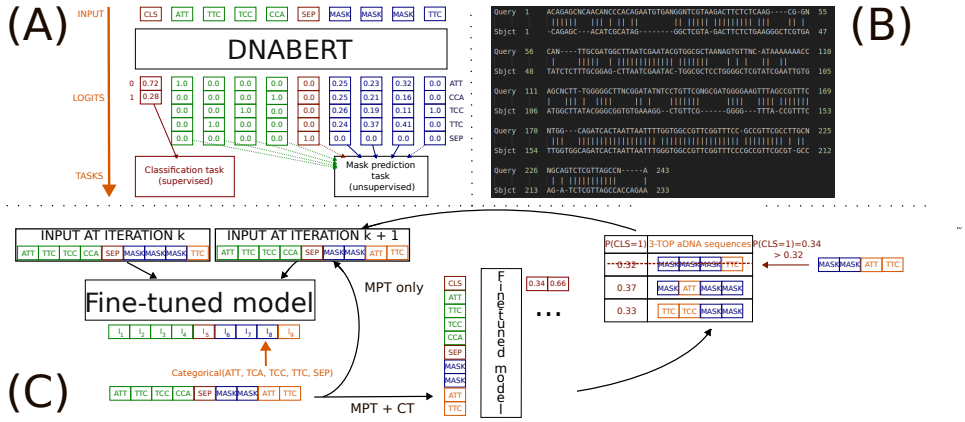

Fig. 1. (A) Adapted DNABERT pre-trained model. We add special tokens in input couples: CLS token, a placeholder for classification logit, and SEP token between K_U and K_A. (B) Alignement between Gargammel sequences (top) and generated sequences (bottom). We observe high identity in nucleotides. In general, missing data are in close proportions in both sequences, but we observe unwanted gaps in the top sequence, preventing a proper study of these missing data. It should be addressed through Needleman and Wunsch algorithm modifications. (C) The two generation algorithms from iteration k to iteration $k+1$. In both cases, a random mask is filled (left). With generation using the MPT+CT model, we use a K-Top table (right)

2 Method

Our method relies on BERT [4], a large language model including attention-based transformers. .We used specifically DNABERT [5], an encoder BERT pre-trained on modern DNA sequences from the reference human genome, originally designed for classification purposes. Our data are couples of undamaged and aDNA sequences (s_U, s_A). We used msprime [1] to simulate a simple demographic scenario of constant population size for a 25Mbp haploid chromosome and sampled one 100-generation-old individual (s_U) and one reference individual from the present time. Because msprime only produces undamaged data, s_A, the ancient (damaged) version of s_U, is generated using the full pipeline that wraps Gargammel. Both sequences are 25 million long including nucleotides (A, C, G, T). Only s_U contains missing data. We tokenized (s_U, s_A) couples into overlapping 6-mers sequences (K_U, K_A) with additional special tokens (CLS, SEP, MASK) as depicted in part (A) of Fig. 1. Because DNABERT only supports input of size below 512, we slice s_U and s_A in around 100000 chunks each, before building couples. Missing data symbols in s_A are removed, considered as gaps, leading to K_U and K_A of different sizes, as allowed by the architecture.

We fine-tuned the model using up to two tasks:

- A Mask Prediction task (MPT): to understand the 'language" of our DNA simulations and, in particular, of aDNA. The MPT is done by replacing contiguous positions of size 6 in the whole sequence (or exclusively in K_A) by MASK tokens up to 15 percent.
- A (binary) classification task (CT): to be able to predict if a couple (s_U, s_A) is such that s_A is the 'translation" (i.e. ancient damaged version) of s_U.

For both fine-tuning tasks, the loss is based on cross-entropy, and when combining MPT and CT, we add up their losses. We propose two alternative approaches for generation: one using the pre-trained DNABERT model, fine-tuned on MPT only (MPT model), and one using the same model but fine-tuned on both CT and MPT (MPT+CT model). Our first generation algorithm uses the MPT model in a fashion similar to bert-gen [7]. Given s_U and given a maximal sequence size n, we build a tokenized input of the form:
$I = \mathsf{CLS}.K_U.\mathsf{SEP}.\mathsf{MASK}^{n-|K_U|-2}$. The masked part on the right (repeated MASK tokens) represents the unknown tokenized ancient sequence K_A that we want to generate. Masks are filled iteratively based on the predicted word probabilities (logits) at masked positions, as illustrated in part (C) of Figure 1.

Our second generation algorithm uses the MPT+CT model, allowing awareness of wrong translations that are still plausible DNA sequences: we store a $K - \mathsf{Top}$ of the most relevant sequences, which we update iteratively by keeping only the generated sequences with the highest classification scores for class 1 (correct translation). Those sequences serve as starting points for the next generation step as represented in part (C) of Fig 1.

3 Results

With $n = |(s_A, s_U)|$, and K, the size of the K-Top table, we have a time complexity $O(n^3)$ for the first generation algorithm and $O(n^3K)$ for the second. In practice, computations can be done in batches, allowing to generate multiple sequences s_A simultaneously or to process at once the K-Top table. In comparison, complexity is in $O(n \times c \times f)$ for Gargammel with c, the desired coverage (no more than 30X) and with f, the number of fragments sampled by Gargammel to ensure a base pairs compositions. f can lead to a large overhead in practice. It makes our complexity of interest, especially with no stochasticity involved.
We have studied precisely 30 aDNA chunks generated with our MPT model. Using Needleman and Wunsch algorithm in its Blast implementation [2], we align them with their Gargammel counterparts. An alignment example is available in part (B) of Figure 1. In these alignments, we observed that the nucleotides, excluding the gaps and missing ones, are identical for around 74 percent. It emphasizes our capacity to mimic Gargammel properly. Additionally, the generated sequences are not direct copies of Gargammel, but the quality and properties of such alternatives should still be assessed.
With the MPT + CT model, the generation algorithm gave no better results than the one with only the MPT model for now, even with different labeling of data: classification is too easy when s_A parts of "negative" examples come from different genomic regions of the same individual and too hard when they come from the same regions but from different individuals. Future research should explore alternative classification tasks of in-between difficulties.

4 Discussion/Conclusion

We have introduced a new method to simulate aDNA sequences. The method is based on a fine-tuning of an encoder transformer trained on couples of undamaged aDNA and their ancient damaged counterpart followed by a generation algorithm based on the mask prediction capability of this encoder. Complexity, in its simpler version, is in $O(n^3)$ with n the length of entry couples, and the use of batches can speed up computations in practice. It is, to the best of our knowledge, the first method using DNA language models for aDNA generation. We have generated sequences that look plausible through comparisons with Gargammel. Nonetheless, future work should focus on ways to assess the quality of the generated aDNA. For now, note that Gargammel is more versatile as it can generate any level of damage/coverage while we currently need to fine-tune our model for different levels. In the future, it would be of interest to enable conditional generation (conditioned on continuous level of coverage and damages) so that only one training would be necessary. One limitation is that entries should contain below 512 nucleotides: longer sequences are divided into smaller chunks, which in return might prevent the model from capturing long-range correlations. Possible solutions could include increasing the size of the network and integrating linear (instead of quadratic) attention mechanisms. Fortunately, such modifications go in the sense of the recent progress in transformers, even for DNA, and could improve the quality of our results and lower complexity without requiring any change in our method.

References

1. F. Baumdicker et al. Efficient ancestry and mutation simulation with msprime 1.0. Genetics.
2. C. Camacho et al. BLAST+: architecture and applications. BMC Bioinformatics, 2009.
3. T. Cury et al. Inferring effective population sizes of bacterial populations while accounting for unknown recombination and selection: a deep learning approach. Workshop Machine Learning for Microbial Genetics at ECML/PKDD, 2022.
4. J. Devlin et al. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
5. Y. Ji et al. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. Bioinformatics, 2021.
6. G. Renaud et al. gargammel: a sequence simulator for ancient DNA. Bioinformatics, 2016.
7. A. Wang et al. Bert has a mouth, and it must speak: Bert as a markov random field language model, 2019.
8. D. Zhang et al. Dnagpt: A generalized pretrained tool for multiple dna sequence analysis tasks. bioRxiv, 2023.