

EMBED-SEARCH-ALIGN: DNA SEQUENCE ALIGNMENT USING TRANSFORMER MODELS

Pavan Holur^{1,*}, K. C. Enevoldsen^{2,3,*}, Lajoyce Mboning⁴, Thalia Georgiou⁵,
Louis-S. Bouchard⁴, Matteo Pellegrini⁶ & Vwani Roychowdhury^{1,*}

¹Department of Electrical and Computer Engineering, UCLA, ²Center for Humanities Computing, Aarhus University, ³Center for Quantitative Genetics and Genomics, Aarhus University, ⁴Department of Chemistry and Biochemistry, UCLA, ⁵Department of Biochemistry, Biophysics, and Structural Biology (MBIDP), UCLA, ⁶Molecular, Cell and Developmental Biology, UCLA
{pholur, lajoycemboning, thaliageorgiou, matteop, vwani}@ucla.edu, kenneth.enevoldsen@cas.au.dk, louis.bouchard@gmail.com

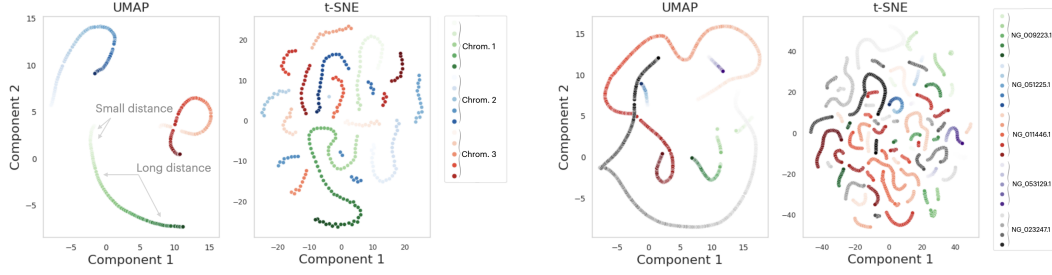
ABSTRACT

DNA sequence alignment involves assigning short DNA reads to the most probable locations on an extensive reference genome. This process is crucial for various genomic analyses, including variant calling, transcriptomics, and epigenomics. Conventional methods, refined over decades, tackle this challenge in two steps: genome indexing followed by efficient search to locate likely positions for given reads. Building on the success of Large Language Models (LLM) in encoding text into embeddings, where the distance metric captures semantic similarity, recent efforts have explored whether the same Transformer architecture can produce numerical representations for DNA sequences. Such models have shown early promise in tasks involving classification of short DNA sequences, such as the detection of coding- vs non-coding regions, as well as the identification of enhancer and promoter sequences. Performance at sequence classification tasks does not, however, translate to *sequence alignment*, where it is necessary to conduct a genome-wide search to successfully align every read. We address this open problem by framing it as an “Embed-Search-Align” task. In this framework, a novel encoder model *DNA-ESA* generates representations of reads and fragments of the reference, which are projected into a shared vector space where the read-fragment distance is used as surrogate for alignment. In particular, DNA-ESA introduces: (1) Contrastive loss for self-supervised training of DNA sequence representations, facilitating rich sequence-level embeddings, and (2) a DNA vector store to enable search across fragments on a global scale. DNA-ESA is > 97% accurate when aligning 250-length reads onto a human reference genome of 3 gigabases (single-haploid), far exceeds the performance of 6 recent DNA-Transformer model baselines and shows task transfer across chromosomes and species.

1 BACKGROUND

DNA sequencers have enabled scientists and medical professionals to determine comprehensive maps of organisms’ genomes and measure variations in genetic data to develop personalized medicine, such as targeted therapies (Li & Durbin, 2009b; Alkan et al., 2011; Metzker, 2010; Cibulskis et al., 2013). The most widely used sequencers, however, are limited to only generating reads that are short segments of DNA, typically represented as a sequence of bases (ATGC). The quality/precision and lengths of such reads depend on the technology being used, and while long-read technologies exist, short-read sequencers remain the most cost-effective per base (Liu et al., 2012). However, genome sequences are often billions of nucleotides long, and therefore the analysis of short reads typically starts by mapping individual reads to the most likely location in a reference genome (Batzoglou, 2005).

*Equal contribution.



(a) A 20,000-long nucleotide sequence from Chr. 1 is picked randomly and broken up into 100 *consecutive* reference fragments each of length 1000 and stride 200. Each fragment is encoded using DNA-ESA into 384-dimensional embeddings. The same is done for two 20K-long sequences picked from Chr. 2 and Chr. 3. The 300 resulting embeddings are visualized using 2D UMAP (McInnes et al., 2018) and t-SNE (Van der Maaten & Hinton, 2008). The embeddings are color-coded with Green (Chr. 1), Blue (Chr. 2) and Red (Chr. 3) respectively. The position of the fragment along each nucleotide sequence is coded by its color intensity.

(b) Five gene sequences from different chromosomes (listed above in the inset) are divided into *consecutive* fragments of length 1000 and stride 200. These are similarly encoded as in part (a). Note that for UMAP projections (both (a) and (b)), the consecutive fragments belonging to the same nucleotide sequence constitute an order-preserving 1D manifold. The tSNE projections, on the other hand, map sequences into multiple order-preserving 1D manifolds. Density-based clustering, such as HDBSCAN (McInnes et al., 2017), would map these 1D manifolds into distinct clusters thus preserving locality in the reference genome.

Figure 1: Visualizing DNA-ESA’s ability to preserve locality of sequences from the reference genome in the embedding space: In DNA-ESA, the reference genome \mathcal{R} is broken into fragments \mathcal{F}_i and their embeddings $h(\mathcal{F}_i)$ are computed. In order to solve the sequence alignment problem, one would expect structures to emerge in the representation space such as: (1) If \mathcal{F}_i and \mathcal{F}_j are overlapping, then their respective embeddings are nearby; and (2) a set of consecutive fragments comprising a long sequence should be mapped to a corresponding low-dimensional manifold in the embedding space. Subfigures (a) and (b) generated using DNA-ESA, visually captures evidence of such emergent geometry in the embedding space. The presence of such structures for all sequences sampled from any part of the reference genome can only be confirmed by extensive sequence alignment results, as presented in Sec. 5.

This paper addresses this foundational task of *Sequence Alignment*. Alignment of reads is critical for most tasks, such as detecting variations in DNA sequences across individuals, which then enables correlating these variations with traits in genome-wide association studies. Furthermore, sequence alignment is critical for the analysis of RNA sequence data, and is the first step in the analysis of epigenomic data that maps DNA methylation, transcription factor binding, or histone modifications.

The simplest sequence alignment task applies to single-end¹ reads. Given a reference sequence $\mathcal{R} := \{b_1, b_2, \dots, b_N\}$ – for the single-haploid human genome (Nurk et al., 2022), $N \approx 3$ gigabases (gb) – the primary objective is to identify the most probable start- and end-positions within this reference for a short DNA read,

$$r := \{\tilde{b}_n, \tilde{b}_{n+1}, \dots, \tilde{b}_{n+Q}\}, \quad Q \ll N, \quad 1 \leq n \leq N - Q \quad (1)$$

which may contain mutations due to base insertions, deletions, and substitutions. Computational simulators have been developed to generate synthetic reads that have properties of real reads. These simulators mimic the read quality and characteristics produced by actual sequencing machines, thus providing a scalable means for validating new alignment approaches (Huang et al., 2012).

Conventional sequence alignment methods make use of algorithmic solutions. The most naïve implementation employs the Smith-Waterman (SW) distance (Smith & Waterman, 1981) wherein the similarity between a read r and the reference \mathcal{R} is scored by assigning values for matches, mismatches, insertions, and deletions. Such a simple approach has computational limitations: it takes $O(MN)$ iterations to align each read. For large genomes like the human genome, this amounts to trillions of computations, making it impractical for high-throughput data. Several key optimizations have emerged over the years: (a) *Construction of search-optimal read and reference representations*

¹A DNA fragment is ligated to an adapter and then sequenced from one end only.

to enable more efficient querying of the genomic data; and (b) *Fast and efficient access to the entire reference genome* to further minimize computational overhead. These developments form the foundation of modern sequence aligners. Specific improvements in state-of-the-art aligners, such as BWA-MEM, include various techniques to further speed up the alignment process. For example, (a) *Sharding* partitions the reference \mathcal{R} into smaller fragments, each of which can be searched individually, reducing the computational load. (b) *Progressive search* leverages phylogenetic tree structures and distance heuristics to scale the complexity logarithmically, making it more feasible for large data sets. (c) *Compression methods*, like the Burrows-Wheeler transform and suffix trees (Li & Durbin, 2009a), significantly truncate the effective search length of the reference sequence. Lastly, (d) *Multi-core/thread implementations and Database instantiations* not only speed up computations through hardware parallelism but also improve data recall rates via advanced caching and indexing methodologies (Li, 2018; Vasimuddin et al., 2019; Langmead et al., 2018).

Here we explore whether a different paradigm can be used to align a read to a genome. We begin with the observation that the preprocessing done to index the genome into a hierarchical structure in Sequence Alignment has similarities to traditional Natural Language Processing (NLP) approaches: Sentences are diagrammed into grammatical components using rule-based techniques, and the resulting dependency parse trees are utilized to establish semantic relationships among words and phrases. The introduction of Deep Learning (DL) models, specifically Transformers, however, drastically altered the NLP landscape by demonstrating the capability to identify syntactic and semantic structures without the need for explicit rule-based frameworks or manual annotations. Motivated by such advancements, researchers today are applying Transformers to bioinformatics, aiming to mitigate the need for cumbersome, genome-specific manual adjustments. In this work, we contribute to this emergent field by introducing a novel method that leverages Transformers for the task of DNA sequence alignment.

1.1 TRANSFORMER MODELS: WRITTEN LANGUAGE TO DNA SEQUENCE ALIGNMENT

Large Language Models (LLMs) have recently demonstrated strong performance in capturing the statistical properties of word sequences and encoding these patterns in their internal weights (Devlin et al., 2019; Reimers & Gurevych, 2019; Touvron et al., 2023). These models have established a new standard in a range of Natural Language Processing (NLP) tasks, including Sentiment Analysis, Entity Recognition, and Question-Answering (Qiu et al., 2020). This success of the underlying Transformer architecture (Vaswani et al., 2017) has led to its application across different data modalities, such as vision with Vision Transformer (Dosovitskiy et al., 2020), and auditory and neurological signals with AudioTransformer and BrainBERT (Verma & Berger, 2021; Wang et al., 2023). Such adaptability of the Transformer architecture to various types of sequence data highlights its potential applicability to bioinformatics tasks, including DNA sequence alignment.

DNA sequences share remarkable similarities with written language, offering a compelling avenue for the application of Transformer models. Like written language, these are sequences generated by a small alphabet of nucleotides $\{A, T, G, C\}$. Indeed, classical DNA modeling efforts have already accommodated mature encoding and hashing techniques initially developed for written language – such as Suffix trees/arrays and Huffman coding (Huffman, 1952; Manber & Myers, 1993) – to successfully parse and compress DNA sequences. Furthermore, just as written language contains repeated subsequences (words, phrases) to represent real-world objects, DNA sequences similarly possess repeating “words” and groupings of such words into a “sentence” representing, for example, genes. Similarly, groups of genes work together to define organism-level functionalities involving many proteins, similar to sentences describing a complex situation.

Within the last few years, several Transformer-based models have been developed for DNA sequence analysis. Notably, DNABERT-2 (Ji et al., 2021; Zhou et al., 2023), Nucleotide Transformer (Dalla-Torre et al., 2023), GenSLM (Zvyagin et al., 2022), and GENA-LM (Fishman et al., 2023) have been designed to discern relationships between short genetic fragments and their functions. Specifically, Nucleotide Transformer representations have shown utility in classifying key genomic features such as enhancer regions and promoter sequences. Similarly, GENA-LM has proven effective in identifying enhancers and Poly-adenylation sites in *Drosophila*. In parallel, DNABERT-2 representations have also been found to cluster in the representation space according to genetic function. Given these advances, a natural question arises: Can these Transformer architectures be readily applied to the task of Sequence Alignment? We delineate the associated challenges as follows:

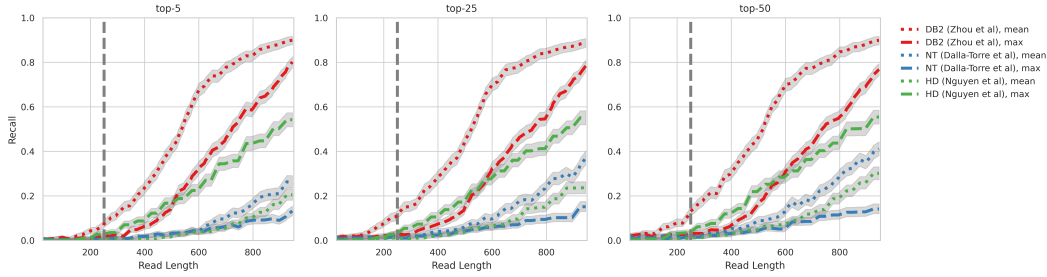


Figure 2: **Alignment recall performance of existing SOTA Transformer-DNA baselines as a function of read length:** We adapted the existing Transformer-DNA models for sequence alignment using standard pooling techniques employed in NLP: mean-/max- pooling the token representations to create a representative embedding for the entire input sequence. Alignment is performed using the same Embed-Search-Align procedure as used in DNA-ESA. Each trendline corresponds to a baseline SOTA Transformer-DNA sequence model and depicts the recall (top- K) over 40K reads sampled uniformly across different read lengths and across the full human genome (3gb - single-haploid). Error bars (Clopper-Pearson Interval (Clopper & Pearson, 1934) @ 95% confidence interval) are depicted in grey. The vertical line ($x = 250$) is a typical read length for which conventional alignment systems function. Across the board, existing baseline models perform poorly. Implementation details are provided in Sec. 4

[L1] Two-Stage Training: DNA-based Transformer models typically undergo pretraining via a *Next Token/Masked Token Prediction* framework, a method originally developed for natural language tasks. To form sequence-level representations, these models often employ pooling techniques that aggregate token-level features into a single feature vector. However, these pooling methods are known to generate suboptimal aggregate features (Reimers & Gurevych, 2019).

[L2] Computation Cost: The computational requirements for Transformer models grow quadratically with the length of the input sequence. This is particularly challenging for sequence alignment tasks that necessitate scanning entire genomic reference sequences.

Figure 2 shows the sequence alignment performance (recall) of several Transformer-DNA models. The testing protocols are elaborated in Sec. 4. Notably, these models exhibit subpar recall performance when aligning typical read lengths of 250.

2 OUR CONTRIBUTIONS

In this paper, we argue that both limitations **L1**, **L2** of Transformer-DNA models can be mitigated by formulating sequence alignment as a vector search and retrieval task. Our approach is twofold: (A) We introduce a sequence encoder, termed *DNA-ESA*, trained through self-supervision. This encoder is designed to map DNA reads to relevant fragments in a reference sequence within a shared embedding space. (B) We leverage a specialized data structure, termed a *DNA vector store*, as a memory bank. This provides efficient access to the entire reference sequence for each read alignment.

In the context of NLP, similar strategies have been explored: (A) Sequence-to-embedding training using contrastive loss mechanisms has shown improved performance at abstractive semantic tasks such as prose summarization and paragraph classification (Gao et al., 2021; Chen et al., 2020a). These models often outperform traditional two-stage models that initially train token-level embeddings and later apply pooling techniques. (B) Specialized data structures known as *vector stores* or *vector databases*, exemplified by FAISS (Johnson et al., 2019) and *Pinecone*, employ advanced indexing and retrieval algorithms to facilitate scalable numerical representation searches. Codebase is *linked*.

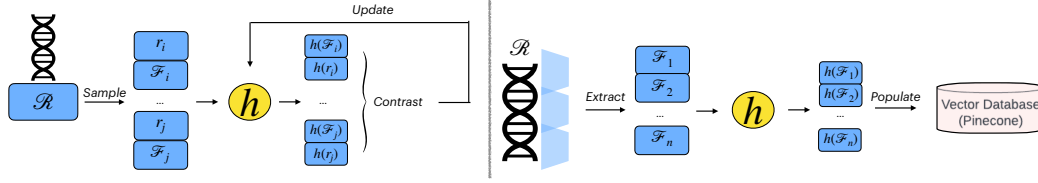


Figure 3: **System overview [A] - Training the encoder and populating the vector store:** Fragments \mathcal{F}_i are sampled from the reference genome and within each, a pure read r_i is randomly sampled (positive pair). Both the fragments and reads are transformed into numerical representations by the shared encoder h . The encoder is trained in a contrastive fashion (using equation 5) to minimize the distance d in the representation space between the read and its corresponding reference fragment and maximize the distance between the read and another random fragment (negative pair). Once the encoder is trained, the reference genome is broken into overlapping fragments that span the entire genome, which are then encoded, and uploaded into a vector store.

3 METHODS

3.1 SEQUENCE ALIGNMENT AS A SEARCH-AND-RETRIEVAL TASK

We formulate the problem of Sequence Alignment as minimizing a sequence alignment function, SA, applied to a read r and a reference sequence \mathcal{R} as

$$v^* = \min_q \text{SA}(r, \mathcal{R}) \quad (2)$$

where $q \in \mathbb{N}_0$ is a candidate reference starting position and v^* is the optimal alignment score. Lower scores indicate better alignments. This optimization exhibits the following property:

[P1] Sharding for sequence alignment: for a read segment r of length Q and reference \mathcal{R} of length N , the complexity of $\text{SA}(r, \mathcal{R})$ scales as $\mathcal{O}(Q)$ when $N \rightarrow Q$.

Using **P1**, we can simplify the optimization problem by breaking it into sub-tasks with significantly shorter reference sequences. Specifically:

$$v^* \approx \min_{\mathcal{F}_i \in \{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_K\}} \text{SA}(r, \mathcal{F}_i). \quad (3)$$

Here, each \mathcal{F}_j is a fragment of \mathcal{R} (i.e., $\mathcal{F}_j \in \mathcal{R}$), and K is the number of these sub-tasks². This approximation is effective under the conditions:

- (1) Fragment \mathcal{F}_j lengths are on the order of the read length (r), not the longer reference (\mathcal{R});
- (2) There are enough fragments \mathcal{F}_i to cover \mathcal{R} , i.e. $\cup \mathcal{F}_j = \mathcal{R}$;
- (3) K is significantly smaller than $\frac{N}{Q}$. If $\frac{N}{Q}$ then this amounts to scanning the whole reference.

Conditions (1) and (2) imply that fragments should be short and numerous enough to cover the reference genome. Condition (3) restricts the number of retrieved reference fragments per read — that we deem to be most likely to contain r — to a small value K . Analogous methods have shown efficacy in text-based Search-and-Retrieval tasks (Peng et al., 2023; Dai et al., 2022) on Open-Domain Question-Answering, Ranking among other tasks. Subsequent sections describe a parallel framework for retrieving reference fragments given a read. The pipeline is shown in Figs. 3 and 4.

3.2 DESIGNING EFFECTIVE SEQUENCE REPRESENTATIONS

An optimal sequence encoder model h is such that the corresponding embeddings of any read r and reference fragment $\mathcal{F} - h(r), h(\mathcal{F})$ respectively — obey the following constraints over a pre-determined distance metric d :

$$d\{h(r_j), h(\mathcal{F}_i)\} > d\{h(r_j), h(\mathcal{F}_j)\}, \quad i \neq j \quad (4)$$

²We denote the relative distance between an alignment and the optimal score at read length Q as $d_{SW} = \frac{|mQ - v^*|}{|mQ|}$ where $m (= -2)$ is the match score while computing the SW distance.

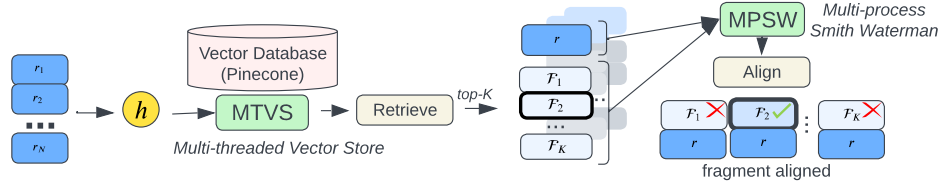


Figure 4: **System overview [B] - Inference on a new read:** A read – as defined in equation 1 and generated by ART (Huang et al., 2012) – is transformed to its numerical representation using encoder h and compared to reference fragment representations in the populated vector database. After retrieving the nearest-K fragments in the embedding space per read, the final optimal alignment is computed using equation 6. Retrieval (multi-threaded) and fine-alignment (multi-process) across fragments are optimized for throughput.

Here i and j serve to distinguish whether a read is aligned to a particular reference fragment, a *positive sample* $\{r_j, \mathcal{F}_j\}$, or there is a mismatch (*negative sample*): $\{r_j, \mathcal{F}_i\}$. Observe that these inequalities constitute the only requirements for the encoder. As long as the *neighborhood* of r_j in the representation space *contains* the representation for \mathcal{F}_j , it will be recovered in the nearest neighbors (top-K set) and alignment will succeed. This motivates using self-supervision (Hadsell et al., 2006; Chen et al., 2020a; Gao et al., 2021) where we are only concerned about the relative distances between positive and negative (read, reference fragment) pairs.

3.2.1 SELF-SUPERVISION AND CONTRASTIVE LOSS

A popular choice for sequence learning using self-supervision involves a contrastive loss setup described by Chen et al. (2020a) and Gao et al. (2021): i.e. for a read r aligned to reference fragment \mathcal{F}_j , the loss l_r simultaneously minimizes the distance of $h(r)$ to $h(\mathcal{F}_j)$ and maximizes the distance to a batch of random fragments of size $B - 1$:

$$l_r = -\log \frac{e^{-d(h(r), h(\mathcal{F}_j))/\tau}}{e^{-d(h(r), h(\mathcal{F}_j))/\tau} + \sum_{i=1}^{B-1} e^{-d(h(r), h(\mathcal{F}_i))/\tau}}. \quad (5)$$

Here τ is a tuneable temperature parameter. To stabilize the training procedure and reach a non-trivial solution, the encoder applies different dropout masks to the reads and fragments similar to the method described in Chen et al. (2020b). Similar setups have been shown to work in written language applications, most notably in Sentence Transformers (Reimers & Gurevych, 2019; Gao et al., 2021; Muennighoff et al., 2023), which continue to be a strong benchmark for several downstream tasks requiring pre-trained sequence embeddings.

3.3 ENCODER IMPLEMENTATION

DNA-ESA uses a Transformer-encoder stack similar to BERT (Devlin et al., 2019; Vaswani et al., 2017), comprising 12 heads and 6-layers of encoder blocks. The size of vocabulary is 10,000. Batch size B is set to 16 with gradient accumulation across 16 steps, the learning rate is annealed using one-cycle cosine annealing (Smith & Topin, 2019), dropout is set to 0.1, and τ is set to 0.05. Reference fragment $|\mathcal{F}_i| \sim \mathcal{U}[800, 2000]$ and read $|r_i| \sim \mathcal{U}[150, 500]$. The distance metric used is *Cosine Similarity*. Models converge after $\sim 20K$ steps of training (see Appendix Sec. A).

3.4 SEARCH AND RETRIEVAL

An outline of the search and retrieval process is presented in Fig. 4. Every read is encoded using the trained model and matched to reference fragments in the vector database. The top-K retrieved fragments per read are then aligned using a SW alignment library to find the optimal alignment. The following sections describe the indexing and retrieval part in more detail.

3.4.1 INDEXING

For a given reference genome \mathcal{R} , we construct a minimal set of reference fragments $\mathcal{F} := \mathcal{F}_1, \mathcal{F}_2, \dots$ to span \mathcal{R} . Note that the fragments overlap at least a read length; i.e. $|\mathcal{F}_i \cap \mathcal{F}_{i+1}| \geq Q$

to guarantee that every read is fully contained within some fragment in the set. In our experiments with external read generators (Huang et al., 2012), $Q_{max} = 250$, $|\mathcal{F}_i| = 1250$. Each reference fragment is encoded using the trained *DNA-ESA* model, and the resulting sequence embeddings ($\in \mathbb{R}^{384}$) – 3M vectors for a reference of 3B nucleotides – are inserted into a Pinecone database. Once populated with all the fragments, we are ready to perform the alignment.

3.4.2 RETRIEVAL

Given a read r , we project its corresponding *DNA-ESA* representation into the vector store and retrieve the approximate nearest- K set of reference fragment vectors and the corresponding fragment metadata $\{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_K\}$.

Diversity priors: While the top- K retrieved fragments can be drawn from across the entire vector store (genome), contemporary recommendation systems that use the top- K retrieval setup *rank and re-rank* top search results (*Slate Optimization* – see Zhu et al. (2007)) to ensure rich and diverse recommendations. Similarly, we apply a uniform prior wherein every retrieval step selects the top- K *per* Chromosome.

Then, a standard SW distance library (Cock et al., 2009) is used to solve equation 3, which can be executed concurrently across the K -reference fragments. This latter part is trivial as described in Sec. 3.1. Let the optimal fragment be \mathcal{F}^* . The metadata for each vector includes (a) the raw \mathcal{F}^* sequence; (b) the start position of \mathcal{F}^* within the reference \mathcal{R} , $q_{\mathcal{F}^*}$. Upon retrieval of a fragment and fine-alignment to find the fragment-level start index, $q_{\mathcal{F}^*}^*$, the global reference start position is obtained as:

$$q^* = q_{\mathcal{F}^*}^* + q_{\mathcal{F}^*}. \quad (6)$$

4 TRANSFORMER-DNA BASELINES: PURE READ RECALL WITHOUT DIVERSITY PRIORS

We detail the testbench setup for evaluating the existing Transformer-DNA baselines whose recall curves are illustrated in Fig. 2. Three recent architectures that encode sequences at the nucleotide level are selected: [NT] *NucleotideTransformer* ($\in \mathbb{R}^{1280}$) (Dalla-Torre et al., 2023); [DB2] *DNABERT-2* ($\in \mathbb{R}^{768}$) (Ji et al., 2021); [HD] *HyenaDNA* ($\in \mathbb{R}^{256}$) (Nguyen et al., 2023). For each of the NT, DB2, HD architecture, a mean- and-max pooling (2) of the token representations yields the sequence representation ($2 \times 3 = 6$ baselines total). Kindly note the following implementation details: (a) An independent vector store is used for each of the six baselines; i.e. reference fragments from the entire genome of 3gb length are encoded using each of the 6 encoders (see Sec. 3.4); (b) 40K pure reads – reads without mutations and variations – of length $Q \sim \mathcal{U}[25, 1000]$ (x) are sampled from across the reference and the average recall (y) for top-5, top-25, and top-50 retrieved fragments – without diversity priors – is reported with respect to the nearest neighbors K (see Fig. 2). While the baselines perform poorly across the board, mean-pooling performs better than max-pooling (except for HD). And DB2 (mean-pooled) and HD (max-pooled) perform better than the rest. We will use these two baseline models in Table 1 to compare to DNA-ESA.

5 RESULTS AND DISCUSSION

DNA-ESA convergence plots are presented in Figs. 5a and 5b (see Appendix) and demonstrate that training using the loss function proposed in equation 5 is stable. Model checkpoints (anonymized) are available at OSF. In Fig. 1, representations of short 1,000-length sequences sampled from sequential (in-order) and gene-specific locations in the reference are visualized in a reduced 2D-UMAP (McInnes et al., 2018) and t-SNE (Van der Maaten & Hinton, 2008) space. The representation space demonstrates *exciting* and desired properties suitable for successfully performing alignment: (a) Sequences sampled in order form a trajectory in the representation space: The loss function described in equation 5 encourages a pair of sequences *close* to one another to have a short distance between them in the representation space, and pairs further apart to have a larger distance. These constraints correspond to a trajectory; (b) Representations of sequences drawn from specific gene locations – despite not being close to one another – show gene-centric clustering: This demonstrates that the DNA-ESA representation space partially acquires *higher-order* function-level

separation as a byproduct of imposing *local* alignment constraints. We present quantitative results next.

5.1 SEQUENCE ALIGNMENT OF ART-SIMULATED READS

The results from Sec. 4 demonstrate that even for pure reads, baseline models do not generate adequate representations to perform sequence alignment. In this section, DNA-ESA and the two best baselines – *DB2*, *mean* and *HD*, *max* – are evaluated using reads generated from an external read simulator (ART) – see Huang et al. (2012). ART has served as a reliable benchmark for evaluating other contemporary alignment tools and provides many controls to model the mutations and variations common in reads generated by Illumina machines.

Simulator configurations: The different simulation configuration options and settings are listed: (A) *Phred quality score* $Q_{PH} \in \{[30, 60], [60, 90]\}$: the likelihood of errors in base-calls of a generated read (computed using large lookup tables); (B) *Insertion rate* $I \in \{0, 10^{-2}\}$: the likelihood of adding a base to a random location in a read; (C) *Deletion rate* $D \in \{0, 10^{-2}\}$: the likelihood of deleting a base in the read; (Others): *Simulator system*: MSv3 [MiSeq] (longest supported read generator system); *Read length*: 250.

Recall configurations: Once the top-K fragments have been retrieved, the first step is to solve equation 3, and for this, we need to compute the SW distance. For all presented results, the settings are: `match_score = -2`, `mismatch_penalty = +1`, `open_gap_penalty = +0.5`, `continue_gap_penalty = +0.1`. After alignment, we get q^* – see equation 6 – as the estimated location of a read in the genome. Let \hat{q}^* be its true location. If $q^* = \hat{q}^*$, it is a perfect match and the recall is successful. In cases where there is a mutation in the first or last position in a read, the fine-alignment will return $q_{\mathcal{F}^*}$ offset by at most 2 locations, resulting in $\hat{q}^* = q^* \pm 2$. Hence, the condition for an exact location match: $|q^* - \hat{q}^*| \leq 2$.

Distance bound, $d_{SW} \in \{\text{None}, 1\%\}$: It is well known that short fragments frequently repeat in the genome (Li & Freudenbergh, 2014) and q^* can correspond to the position of the read in a different location than from where it was sampled. In this case, $q^* \neq \hat{q}^*$, but the SW distance is the minimum possible ($d_{SW} = 0$ – see Footnote 2). Moreover, when reads have mutations, the reference sequence corresponding to the read is no longer a perfect match; i.e. $d_{SW} > 0$. The best an alignment algorithm could do is to find an exact match for the read leading to an optimal alignment score for that read length ($-2Q$). We consider an alignment (with $Q = 250$) to be successful if $d_{SW} < 0.01$; i.e. a mismatch of at most 2 bases.

		DNA-ESA (ours)			<i>DB2</i> , <i>mean</i>	<i>HD</i> , <i>max</i>
I	D	Top-K @ 50 / Chr.	$d_{SW} = 1\%$	Best	Top-K @ 50 / Chr. + $d_{SW} = 1\%$	
$Q_{PH} \in [60, 90]$						
0	0	95.2 \pm 0.63	+ 1.6	96.8 \pm 0.52	37.4 \pm 1.35	14.6 \pm 1.01
0	0.01	95.8 \pm 0.59	+ 1.3	97.1 \pm 0.50	37.4 \pm 1.35	14.8 \pm 1.01
0.01	0	95.1 \pm 0.63	+ 2.1	97.2 \pm 0.49	35.9 \pm 1.35	14.6 \pm 1.01
0.01	0.01	95.8 \pm 0.59	+ 1.2	97.0 \pm 0.51	34.9 \pm 1.34	14.5 \pm 1.01
$Q_{PH} \in [30, 60]$						
0	0	95.2 \pm 0.63	+ 1.4	96.6 \pm 0.54	35.4 \pm 1.34	13.7 \pm 0.98
0	0.01	95.1 \pm 0.64	+ 2.0	97.1 \pm 0.49	37.3 \pm 1.35	14.2 \pm 0.99
0.01	0	94.2 \pm 0.68	+ 2.1	96.3 \pm 0.56	36.0 \pm 1.35	13.8 \pm 0.98
0.01	0.01	94.9 \pm 0.64	+ 1.5	96.4 \pm 0.55	36.8 \pm 1.35	13.5 \pm 0.98

Table 1: **Performance of DNA-ESA with respect to baselines:** (with diversity priors – see Def. 3.4.2) The performance of DNA-ESA reaches that of conventional algorithmic approaches such as StrobeAlign, Minimap, and BWA-Mem2 (Sahlin, 2022; Vasimuddin et al., 2019; Li, 2018) and far exceeds the performance of Transformer-DNA baselines, *DB2*, *mean* and *HD*, *max* (top performing baselines from Fig. 2). A separate vector store is populated with fragments of the entire genome for our model and each of the baselines while keeping the search strategy identical. For definitions of I , D , Q_{PH} , K , d_{SW} please see *Recall/Simulator Configurations*.

		<i>Homo Sapiens</i>			<i>Pan Troglodyte</i>		<i>Rattus Norvegicus</i>	
Train	Reads	2 (seen)	3	Y	2A	2B	1	2
Chr. 2	Pure	97.5±0.46	98.2±0.41	99.3±0.27	94.5±0.67	94.3±0.68	93.3±0.73	93.2±0.74
	ART	97.9±0.43	97.6±0.44	98.0±0.43	97.1±0.51	97.2±0.50	95.4±0.62	96.4±0.55

Table 2: **Evidence of task transfer - DNA-ESA trained only on Human Chr. 2 but alignment performed on chromosomes from both humans (2,3,Y) and other species:** DNA-ESA is trained using fragments and reads drawn only from Chr. 2 (longest). The model is then used to encode fragments and reads that were *never used during training*. In particular, independent vector stores are populated with fragments drawn from a set of human chromosomes (3,Y) and other species (*chimpanzee* - 2A,2B, *rat* - 1,2). Recall is computed using pure and ART-generated reads. The performance is similar to those reported in Table 1 suggesting that DNA-ESA is capable of learning the compositional structure of DNA sequences rather than memorizing the sequences used in training.

Performance: Table 1 contains the spotlight result of sweeps along these several parameters in addition to a direct comparison to DB2, mean and HD, max baselines, the best-performing baselines on pure reads in Sec. 4. We observe the following: (A) DNA-ESA demonstrates strong recall of $\sim 97\%$ across a variety of read generation and recall configurations described in Sec. 5.1; (B) Reads with less noise – high Q_{PH} , low I, D – are more often correctly aligned; (C) The d_{SW} bound adds $\sim 1.5 - 2\%$ in recall (to cover 5% of misalignments) – a $\sim 20\%$ boost. This suggests that in the cases of misses, while finding the exact index match, the retrieved reference fragments are still high-quality retrievals that differ by at most 2 bases; (D) While the precise method of (a) generating reads (ART settings, gapped, long vs. short, paired-end vs. single-end, etc.), (b) accounting for the variation in the reference (with or without mitochondrial DNA, single- or double haploid, etc.), (c) detecting a successful alignment (duplicates, distance criteria) vary considerably, DNA-ESA approaches the performance reported by mature algorithmic methods of StrobeAlign, BWA-Mem2, and Minimap (Sahlin, 2022; Vasimuddin et al., 2019; Li, 2018).

5.2 TASK TRANSFER FROM CHROMOSOME 2

Are these results indicative of DNA-ESA’s adaptability to new genomic sequences, rather than a strict adherence to its training data? This would suggest the model’s learning to solve the sequence alignment task rather than memorizing the genome.

Experiment setup: DNA-ESA is trained on Chromosome 2 – the longest chromosome – and recall is computed on unseen chromosomes from the human genome (3, Y) (*inter-chromosome*) and select chromosomes from chimpanzee (2A,2B) and rat (1,2) DNA (*inter-species*). Reads are either pure or ART-generated – as in Sec. 5.1, with the following simulator configurations: $I = 10^{-4}$, $D = 10^{-4}$, $Q_{PH} = [60, 90]$, $d_{SW} = 1\%$, $Q = 250$. Top- K is set to 50, reads per setting = 5,000. Independent vector stores are constructed for each chromosome; representations for reference fragments (staging) and reads (testing) are generated by the Chr. 2-trained model. The results are reported in Table 2.

Performance: The convergence plot is presented in Appendix A. Recall across unseen *human* chromosomes (3, Y) match those reported in Table 1 *despite the model being trained on only a single chromosome (Chr. 2)*: this suggests that DNA-ESA indeed learns to generate representations that solve the task of sequence alignment and is able to generalize this ability across unseen chromosomes with different overall nucleotide distributions. Additionally observe that the recall across species follows a decreasing trend: Human (2, 3, Y) > Chimpanzee (2A,2B) > Rat (1,2). This slight edge to chimpanzees may be attributed to hereditary similarities between the Chr. 2 found in chimpanzees and humans (Ijdo et al., 1991).

5.3 ABLATION STUDIES

Several additional experiments on DNA-ESA are presented in the Appendix. (A) In Sec. C.1, we present performance under different $d_{SW} \in \{5\%, 10\%\}$ bounds and top- $K \in \{10, 20\}$ / chromosome settings. The performance at different K values indicates the extent of clustering in the representation space. High performance under low K (smaller neighborhood radius) would imply a better representation space. Varying d_{SW} quantifies the accuracy of retrieved fragments given a

read, despite not finding an exact index match. (B) In Sec. C.2, we forego the diversity priors in the search (top- K per chromosome) and instead scan across the entire genome (top- K). The performance is predictably worse; (C) In Sec. C.3, we evaluate the performance of DNA-ESA on multiple read lengths beyond $Q = 250$. Performance on longer reads is higher (including for read lengths outside the distribution used during training).

6 FUTURE WORK

While the DNA-ESA method demonstrates competitive recall performance when compared to traditional algorithmic approaches, it exhibits limitations in the context of computational speed, currently operating at a throughput of 200 reads per minute. This rate of processing is sub-optimal for large-scale genomic studies that often involve analyzing millions of reads. The alignment task is highly parallelizable, however, which is captured by property [P1]. As part of our ongoing and future work, we aim to address the time efficiency bottleneck. For a more formal complexity analysis, the reader is referred to Appendix Sec. B. Optimization strategies under consideration include model compilation techniques to accelerate the inference stage, as well as enhanced parallelization schemes, specifically in vector store search operations and the alignment of fragments to reads (as illustrated in Fig. 4). Another area for improvement is the performance of DNA-ESA with respect to shorter read lengths. This particular challenge stems from the inherent difficulty in matching short sequences with high confidence. To this end, we continue to explore alternative training regimens, incorporating additional read/reference features, and applying data augmentation strategies, among other techniques.

7 CONCLUDING REMARKS

DNA sequence alignment has witnessed the development of increasingly refined methods over the last few decades. These traditional methods have integrated DNA-specific optimizations for both indexing and retrieval tasks. In contrast to these established techniques, we introduced a fundamentally data-driven model within the “Embed-Search-Align” (ESA) framework. Employing a Transformer-based DNA-ESA encoder, our approach performs sequence alignment through self-supervised learning. This model interacts with modern vector databases for efficient search, as outlined in Sec. 3.4.1. Our experimental validation, detailed in Figs. 1 and 2 and Tables 1, 3 and 4, reveals that DNA-ESA achieves high accuracy levels across diverse simulation and recall scenarios. The model also exhibits an ability for task transfer, including alignment across different chromosomes and even species, as indicated in Table 2.

While the performance of DNA-ESA only reaches those reported by existing methods, our framework can be viewed as an alternate paradigm for sequence alignment with potential for solving tasks that are currently considered difficult. For example, we redefine sequence alignment as a problem solvable through parameters within the DNA-ESA encoder, independent of the length of the reference genome used for alignment: the same DNA-ESA encoder weights trained with Chr. 2 can be used to align reads from the *Rattus Norvegicus* genome of a different length. It provides an alternate means of representing subsequences of DNA of various lengths in a fixed-dimensional vector space: both reads (of 250–length) and fragments (of 1250–length) can be projected into the same space and compared using a distance metric (cosine similarity). Thus, our method can be considered as a flat search in contrast to conventional alignment methods which perform an explicit hierarchical search. We envision that a combination of different frameworks of sequence similarity would address seemingly ambitious tasks such as the alignment of reads onto the Pan Genome with high accuracy (Liao et al., 2023).

REFERENCES

- Can Alkan, Bradley P Coe, and Evan E Eichler. Genome structural variation discovery and genotyping. *Nature reviews genetics*, 12(5):363–376, 2011.
- Serafim Batzoglou. The many faces of sequence alignment. *Briefings in bioinformatics*, 6(1):6–22, 2005.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020b.
- Kristian Cibulskis, Michael S Lawrence, Scott L Carter, Andrey Sivachenko, David Jaffe, Carrie Sougnez, Stacey Gabriel, Matthew Meyerson, Eric S Lander, and Gad Getz. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature biotechnology*, 31(3):213–219, 2013.
- C. J. Clopper and E. S. Pearson. The Use of Confidence or Fiducial Limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 12 1934. ISSN 0006-3444. doi: 10.1093/biomet/26.4.404. URL <https://doi.org/10.1093/biomet/26.4.404>.
- Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 03 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp163. URL <https://doi.org/10.1093/bioinformatics/btp163>.
- Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*, 2022.
- Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza Revilla, Nicolas Lopez Carranza, Adam Henryk Grywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pp. 2023–01, 2023.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- Veniamin Fishman, Yuri Kuratov, Maxim Petrov, Aleksei Shmelev, Denis Shepelin, Nikolay Chekanov, Olga Kardymon, and Mikhail Burtsev. GENA-LM: A Family of Open-Source Foundational Models for Long DNA Sequences. *bioRxiv*, pp. 2023–06, 2023. Publisher: Cold Spring Harbor Laboratory.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL <https://aclanthology.org/2021.emnlp-main.552>.

- Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.
- Weichun Huang, Leping Li, Jason R. Myers, and Gabor T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012. Publisher: Oxford University Press.
- David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952. Publisher: IEEE.
- JW Ijdo, Antonio Baldini, DC Ward, ST Reeders, and RA Wells. Origin of human chromosome 2: an ancestral telomere-telomere fusion. *Proceedings of the National Academy of Sciences*, 88(20):9051–9055, 1991.
- Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V. Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021. Publisher: Oxford University Press.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. Publisher: IEEE.
- Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.
- Ben Langmead, Christopher Wilks, Valentin Antonescu, and Rone Charles. Scaling read aligners to hundreds of threads on general-purpose processors. *Bioinformatics*, 35(3):421–432, 07 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty648. URL <https://doi.org/10.1093/bioinformatics/bty648>.
- Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 05 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty191. URL <https://doi.org/10.1093/bioinformatics/bty191>.
- Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009a.
- Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009b.
- Wentian Li and Jan Freudenberg. Mappability and read length. *Frontiers in Genetics*, 5, 2014. ISSN 1664-8021. doi: 10.3389/fgene.2014.00381. URL <https://www.frontiersin.org/articles/10.3389/fgene.2014.00381>.
- Wen-Wei Liao, Mobin Asri, Jana Ebler, Daniel Doerr, Marina Haukness, Glenn Hickey, Shuangjia Lu, Julian K Lucas, Jean Monlong, Haley J Abel, et al. A draft human pangenome reference. *Nature*, 617(7960):312–324, 2023.
- Lin Liu, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and Maggie Law. Comparison of next-generation sequencing systems. *Journal of Biomedicine and Biotechnology*, 2012, 2012.
- Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 22(5):935–948, 1993. Publisher: SIAM.
- Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.
- Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- Michael L Metzker. Sequencing technologies—the next generation. *Nature reviews genetics*, 11(1):31–46, 2010.

- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL <https://aclanthology.org/2023.eacl-main.148>.
- Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, and Yoshua Bengio. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *arXiv preprint arXiv:2306.15794*, 2023.
- Sergey Nurk, Sergey Koren, Arang Rhie, Mikko Rautiainen, Andrey V. Bzikadze, Alla Mikheenko, Mitchell R. Vollger, Nicolas Altemose, Lev Uralsky, Ariel Gershman, Sergey Aganezov, Savannah J. Hoyt, Mark Diekhans, Glennis A. Logsdon, Michael Alonge, Stylianos E. Antonarakis, Matthew Borchers, Gerard G. Bouffard, Shelise Y. Brooks, Gina V. Caldas, Nae-Chyun Chen, Haoyu Cheng, Chen-Shan Chin, William Chow, Leonardo G. de Lima, Philip C. Dishuck, Richard Durbin, Tatiana Dvorkina, Ian T. Fiddes, Giulio Formenti, Robert S. Fulton, Arkarachai Fungtammasan, Erik Garrison, Patrick G. S. Grady, Tina A. Graves-Lindsay, Ira M. Hall, Nancy F. Hansen, Gabrielle A. Hartley, Marina Haukness, Kerstin Howe, Michael W. Hunkapiller, Chirag Jain, Miten Jain, Erich D. Jarvis, Peter Kerpedjiev, Melanie Kirsche, Mikhail Kolmogorov, Jonas Korlach, Milinn Kremitzki, Heng Li, Valerie V. Maduro, Tobias Marschall, Ann M. McCartney, Jennifer McDaniel, Danny E. Miller, James C. Mullikin, Eugene W. Myers, Nathan D. Olson, Benedict Paten, Paul Peluso, Pavel A. Pevzner, David Porubsky, Tamara Potapova, Evgeny I. Rogaev, Jeffrey A. Rosenfeld, Steven L. Salzberg, Valerie A. Schneider, Fritz J. Sedlazeck, Kishwar Shafin, Colin J. Shew, Alaina Shumate, Ying Sims, Arian F. A. Smit, Daniela C. Soto, Ivan Sović, Jessica M. Storer, Aaron Streets, Beth A. Sullivan, Françoise Thibaud-Nissen, James Torrance, Justin Wagner, Brian P. Walenz, Aaron Wenger, Jonathan M. D. Wood, Chunlin Xiao, Stephanie M. Yan, Alice C. Young, Samantha Zarate, Urvashi Surti, Rajiv C. McCoy, Megan Y. Dennis, Ivan A. Alexandrov, Jennifer L. Gerton, Rachel J. O’Neill, Winston Timp, Justin M. Zook, Michael C. Schatz, Evan E. Eichler, Karen H. Miga, and Adam M. Phillippy. The complete sequence of a human genome. *Science*, 376(6588):44–53, 2022. doi: 10.1126/science.abj6987. URL <https://www.science.org/doi/abs/10.1126/science.abj6987>.
- Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. Check your facts and try again: Improving large language models with external knowledge and automated feedback, 2023.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10): 1872–1897, 2020.
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL <https://aclanthology.org/D19-1410>.
- Kristoffer Sahlin. Strobealign: flexible seed size enables ultra-fast and accurate read alignment. *Genome Biology*, 23(1):260, 2022.
- Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pp. 369–386. SPIE, 2019.
- Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981. Publisher: Elsevier Science.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Md. Vasimuddin, Sanchit Misra, Heng Li, and Srinivas Aluru. Efficient architecture-aware acceleration of bwa-mem for multicore systems. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 314–324, 2019. doi: 10.1109/IPDPS.2019.00041.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Prateek Verma and Jonathan Berger. Audio transformers: Transformer architectures for large scale audio understanding. adieu convolutions. *arXiv preprint arXiv:2105.00335*, 2021.

Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. BrainBERT: Self-supervised representation learning for intracranial recordings. *arXiv preprint arXiv:2302.14367*, 2023.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome, 2023.

Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 97–104, 2007.

Maxim Zvyagin, Alexander Brace, Kyle Hippe, Yuntian Deng, Bin Zhang, Cindy Orozco Bohorquez, Austin Clyde, Bharat Kale, Danilo Perez-Rivera, and Heng Ma. GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. *bioRxiv*, pp. 2022–10, 2022. Publisher: Cold Spring Harbor Laboratory.

APPENDIX

A CONVERGENCE

Fig. 5 plot the convergence of the DNA-ESA encoder model discussed in the main text. We see convergence after $\sim 20k$ steps.

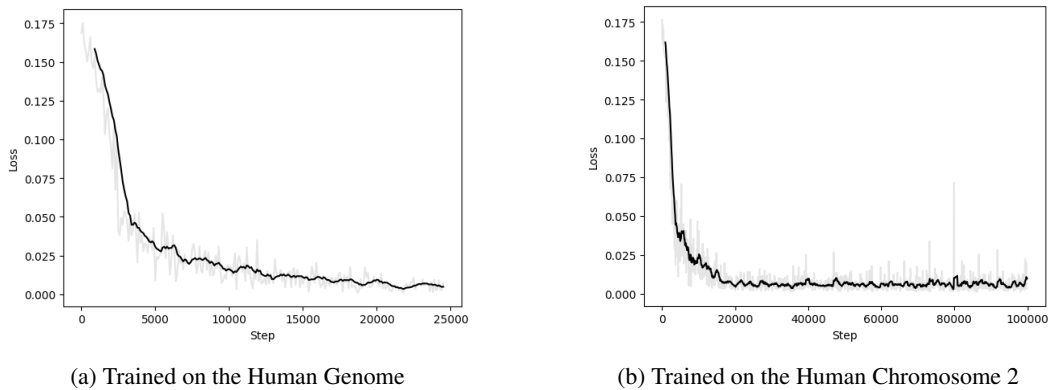


Figure 5: DNA-ESA convergence plots

B COMPLEXITY OF COMPUTING ALIGNMENT

B.1 COST OF CONSTRUCTING A NEW REPRESENTATION

Computing the embedding \mathcal{E} of a sequence of length F using DNA-ESA encoding – a typical Transformer-based attention architecture – has the following computation complexity:

$$\mathcal{O}(LH * (F^2 * d + d^2 * F)) \Rightarrow \mathcal{O}(F^2 * d + d^2 * F) \quad (7)$$

Where d is the embedding dimension of the model, L is the number of layers in the Transformer and H is the number of heads per layer. As d is a controllable parameter for the model, we can further simplify:

$$\mathcal{O}(F^2 * d + d^2 * F) \Rightarrow \mathcal{O}(F^2) \quad (8)$$

The F^2 complexity follows the basic implementation of attention in transformers, but recent efforts (Beltagy et al., 2020; Kitaev et al., 2020) have developed shortcuts to reduce the cost. These have already been applied to DNA sequence modeling (Nguyen et al., 2023).

B.2 VECTOR STORE UPSTREAM

Vector store \mathcal{D} is populated once (in bulk) with encoded fragment-length sequences drawn from the entire genome; *constant time complexity* C to upload $< 10M$ vectors.

B.3 RETRIEVAL COST

Given a new embedding, $\epsilon(G, K)$ is the cost of retrieving top- K nearest neighbors across the fragment embeddings, where G is the length of the reference genome. In modern vector databases, ϵ scales logarithmically with G .

B.3.1 FINE-GRAINED ALIGNMENT

Existing libraries/algorithms (e.g. the Smith–Waterman algorithm) can identify the alignment between a fragment sequence (of length F) and read (of length Q) in $\mathcal{O}(FQ)$.

B.3.2 TOTAL COMPLEXITY

Total complexity involves (a) constructing the representation of a read; (b) querying the vector store; (c) running fine-grained alignment with respect to the K returned reference fragment sequences:

$$\mathcal{O}(F^2 + FQK + \epsilon(G, K)) \Rightarrow \mathcal{O}(F(F + QK) + \epsilon(G, K)) \Rightarrow \mathcal{O}(FQK + \epsilon(G, K))$$

C ABLATION STUDIES

C.1 SWEEPING TOP-K AND SW DISTANCE BOUND

In Table 3, we report the performance of DNA-ESA with larger Smith-Waterman distance bounds $d_{SW} \in \{10\%, 20\%\}$ and a smaller number of recalled fragments $K \in \{10, 20\}$. Distance bound $d_{SW} < 10\%$ is equivalent to an acceptable mismatch of at most ~ 8 bases between the read (length $Q = 250$) and recalled reference fragments. We observe that with small K , the recall is $> 91\%$ (exact index match). As the distance bound increases, performance predictably improves; however, the performance gain from $d_{SW} = 25$ to $d_{SW} = 50$ is small. This implies that many retrieved fragments are high-quality retrievals; i.e. of high-likelihood to align with the read, and *do not benefit* from a more generous distance bound to improve recall performance significantly. Furthermore, decreasing the top- K per chromosome from $20 \rightarrow 10$ does not substantially worsen performance ($\sim 2\%$) indicating that the optimal retrievals are usually the closest in the embedding space.

I	D	$Q_{PH} \in [30, 60]$			$Q_{PH} \in [60, 90]$		
		Exact	$d_{SW} < 5\%$	$d_{SW} < 10\%$	Exact	$d_{SW} < 5\%$	$d_{SW} < 10\%$
Top-10 / chromosome							
0.0	0.0	91.5±0.81	96.6±0.54	96.6±0.54	91.9±0.79	96.4±0.55	97.3±0.49
	0.01	92.0±0.79	96.5±0.55	96.8±0.52	92.0±0.79	96.2±0.57	97.6±0.47
0.01	0.0	91.9±0.79	96.4±0.55	96.5±0.55	92.0±0.79	96.6±0.54	97.4±0.48
	0.01	91.8±0.80	96.3±0.56	97.2±0.50	91.5±0.81	96.3±0.56	97.3±0.49
Top-20 / chromosome							
0.0	0.0	93.4±0.73	97.5±0.47	98.0±0.43	93.7±0.71	97.7±0.45	97.8±0.44
	0.01	93.2±0.74	97.2±0.49	98.2±0.41	93.6±0.72	97.5±0.47	97.8±0.45
0.01	0.0	93.0±0.74	97.9±0.44	97.9±0.43	92.6±0.76	97.4±0.48	97.8±0.44
	0.01	93.0±0.74	97.4±0.48	97.8±0.45	93.4±0.72	97.8±0.45	98.1±0.42

Table 3: **Sequence alignment recall of DNA-ESA sweeping top-K and d_{SW} :** The various parameters are described in Sec. 5.1. DNA-ESA presents a recall of $> 97\%$ across several read configurations rivaling contemporary algorithmic models. Performance predictably degrades as more noise is introduced into a read. Performance improves with larger search radius (top- K), higher quality reads (Q_{PH}) and large distance bound d_{SW} .

C.2 WITHOUT DIVERSITY PRIORS IN TOP-K

In Table 4, we report the performance without diversity priors used in the retrieval step: i.e. the nearest- K neighbors in the embedding space are selected from the *entire* set of fragments spanning the genome rather than uniformly sampling from each chromosome. The performance predicably falls in comparison to those reported in Table 1 since fewer fragments scattered unevenly across the different chromosomes are being retrieved per read.

I	D	$Q_{PH} \in [30, 60]$			$Q_{PH} \in [60, 90]$		
		Exact	$d_{SW} < 5\%$	$d_{SW} < 10\%$	Exact	$d_{SW} < 5\%$	$d_{SW} < 10\%$
Top-100							
0.0	0.0	82.6±1.08	92.0±0.79	93.2±0.74	88.5±0.92	91.9±0.79	93.0±0.74
	0.01	82.3±1.09	91.8±0.80	92.7±0.76	89.3±0.89	93.2±0.74	93.9±0.70
0.01	0.0	83.3±1.06	92.3±0.77	93.3±0.73	88.7±0.91	92.4±0.77	93.3±0.73
	0.01	82.8±1.08	91.6±0.81	92.5±0.76	88.7±0.91	92.4±0.77	93.2±0.73
Top-50							
0.0	0.0	81.1±1.11	90.6±0.84	92.0±0.79	86.7±0.97	90.7±0.84	91.9±0.79
	0.01	80.5±1.13	90.4±0.85	91.6±0.80	87.4±0.95	91.7±0.80	92.7±0.76
0.01	0.0	81.3±1.11	90.8±0.83	92.3±0.78	86.8±0.97	91.0±0.83	92.2±0.78
	0.01	81.2±1.11	90.4±0.85	91.7±0.80	87.1±0.96	91.2±0.82	92.4±0.77

Table 4: **Sequence alignment recall of DNA-ESA - without diversity priors:** The various parameters are described in Sec. 5.1. DNA-ESA presents a recall of $\approx 90\%$. Similar to the result presented in the main text, performance improves with larger search radius (top- K), higher quality reads (Q_{PH}) and large distance bound (d_{SW}).

C.3 ACROSS READ LENGTHS

In Table 5, the performance of DNA-ESA is reported across read lengths. We observe *Zero-shot performance at longer read lengths*: The model performs better at longer read lengths (even exceeding the read length bound established during training $\mathcal{U}[150, 500]$); while evaluating for longer reads, we make sure to guarantee that the reads exist as subsequences of fragments. Improving the performance at shorter read lengths is the subject of future work.

Top-K @ 50					
I	D	$Q = 150$	$Q = 200$	$Q = 250$	$500 < Q < 1000$
$Q_{PH} \in [60, 90]$					Pure Reads
0	0	84.6 ± 1.03	94.9 ± 0.65	96.8 ± 0.52	>97
0	0.01	85.9 ± 0.99	95.2 ± 0.63	97.1 ± 0.50	
0.01	0	85.8 ± 0.99	94.5 ± 0.66	97.2 ± 0.49	
0.01	0.01	86.2 ± 0.98	95.3 ± 0.63	97.0 ± 0.51	
$Q_{PH} \in [30, 60]$					
0	0	84.6 ± 1.03	94.6 ± 0.66	96.6 ± 0.54	>96
0	0.01	85.2 ± 1.01	95.0 ± 0.64	97.1 ± 0.49	
0.01	0	85.7 ± 1.00	94.5 ± 0.66	96.3 ± 0.56	
0.01	0.01	85.8 ± 0.99	94.4 ± 0.67	96.4 ± 0.55	

Table 5: **DNA-ESA recall performance across read lengths:** Performance of DNA-ESA is higher for longer reads including those lengths on which the model was not trained ($Q \in \mathcal{U}[150, 500]$). Shorter reads are more challenging for the model potentially due to replicates found across the reference.