MSA-LM: Integrating DNA-level Inductive Biases into DNA Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent advances in DNA language modeling have been limited by computational constraints and the ability to capture long-range dependencies within genomic data effectively. While effective, traditional transformer-based models suffer from quadratic complexity and limited context windows, making them unsuitable for large-scale DNA modeling. In contrast, subquadratic models, while efficient, often lack bidirectionality and struggle with training scalability. We introduce MSA-LM, an inductive-bias-aware subquadratic DNA Multiple Sequence Alignment (MSA) model that addresses these limitations. MSA-LM integrates a bidirectional Mamba model for sequence mixing, providing transformer-like expressibility without the associated quadratic complexity. By utilizing a sparse attention mechanism, MSA-LM selectively processes the main DNA sequence while incorporating evolutionary information from MSA data, significantly reducing computational overhead. Our results demonstrate that MSA-LM achieves state-of-the-art performance on long-context variant effect prediction tasks and Genomic Benchmarks, particularly excelling in regulatory sequence analysis. The proposed model not only surpasses existing transformer-based and subquadratic approaches in efficiency but also maintains high accuracy across diverse genomic tasks, marking a significant improvement in DNA language modeling capabilities.

1 Introduction

Advances in model sizes and architectures have brought about a revolution in sequence modeling capabilities. The introduction of recurrence [26], attention [1], and memory [24] have led to many performance improvements. The transformer model [44], commonly used in large language models (LLMs) [6], applies self-attention and implicit memory [14] to sequence modeling.

Transformers have shown impressive generalization capabilities in natural language processing, prompting researchers to extend the models' abilities to sequences beyond language. Transformers have been applied to protein sequences [30] and genomics data [39]. Recently, they have been used in DNA modeling [10]. However, The human genome consists of 3 billion base pairs, with gene sizes ranging from 10 thousand to 2 million base pairs [32]. These large DNA sequences are expensive to analyze using a transformer due to the quadratic nature of self-attention [27] and the model's instability across extended context windows [31].

Subquadratic models ([38], [19], [16]) have been explored as alternate method to transformers for DNA modeling. They have shown high performance on Genomic Benchmarks tasks [41] and have context lengths ranging up to 128k base pairs [36].

Recent DNA language modeling methods have added information augmentations [2] or improved tokenizers/information aggregation to the original DNA sequence [40]. One of the most common

DNA augmentations is multiple sequence alignment (MSA) data. This information provides key evolutionary relationship information relative to each base pair. Transformer-based methods for DNA MSA processing have shown state-of-the-art performance in tasks with a basis in evolutionary mutations (variant effect prediction) [3]. However, these models leverage quadratic sequence-wise attention and axial attention-based methods, which do not scale well to long sequences. Because of this, transformer-based DNA MSA models have only been trained at short context lengths[1].

Subquadratic MSA models have been proposed as alternatives to transformer-based approaches [43]. However, these models lack bidirectionality, hindering modeling accuracy greatly. In addition, subquadratic MSA models are difficult to train at scale due to running subquadratic sequence mixers on all auxiliary sequences in addition to the main sequence in an MSA. Batch size scaling is difficult in these settings, leading to inefficient training and inference.

To correct the shortcomings of subquadratic DNA MSA models, we propose MSA-LM, an inductive-bias-aware subquadratic DNA MSA model. This model leverages a bidirectional Mamba model as a sequence mixer [25], which provides similar expressibility to full self-attention in transformers without quadratic complexity. In addition, MSA-LM only runs the Mamba operation on the main sequence, using sparse attention computations to integrate MSA data into one main sequence representation [8]. Through this, we leverage MSA data as auxiliary information relative to the main sequence and fix problems in the expressibility of previous subquadratic MSA models. Evaluations of MSA-LM show state-of-the-art (SOTA) performance in 3 Genomic Benchmarks tasks[2] (see Table 6.2) and shows similar performance to SOTA models in long-context variant effect prediction (see Table 6.1).

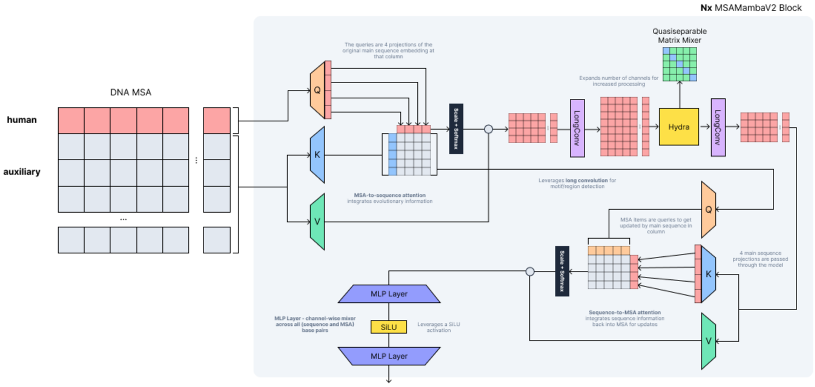
Nx MSAMambaV2 Block
DNA MSA
human
auxiliary
...
Q
K
V
The queries are 4 projections of the original main sequence embedding at that column
Scale + Softmax
MSA-to-sequence attention
integrates evolutionary information
LongConv
Expands number of channels for increased processing
Quasiseparable Matrix Mixer
Hydra
LongConv
Leverages long convolution for motif/region detection
MSA items are queries to get updated by main sequence in column
4 main sequence projections are passed through the model
Sequence-to-MSA attention
integrates sequence information back into MSA for updates
Scale + Softmax
MLP Layer
SiLU
MLP Layer
MLP Layer - channel-wise mixer across all (sequence and MSA) base pairs
Leverages a SiLU activation

Figure 1: A diagram of the MSA-LM architecture. The architecture consists of multiple MSA-LM blocks, each of which contains a bidirectional mamba (quasiseparable matrix mixer) wrapped by long convolutions. It also includes an MSA to sequence mixer and a sequence to MSA mixer to integrate evolutionary information across the MSA.

2 Background

Deoxyribonucleic acid is a polymer made up of 4 base nucleotides (adenine, cytosine, guanine, and thymine). The polymer forms a double helix structure from two complementary strands. DNA contains regions known as genes, which can code for different proteins to cause cellular change. Genes also consist of control sequences. These include enhancers, which can increase the DNA transcription of a specific gene into a protein; promoters, which allow the initiation of transcription; and silencers, which prevent transcription from occurring. [5]

DNA sequences contain introns and exons. Exons contain DNA information used to form the final protein, while introns are non-coding regions that can be spliced out in different combinations to

[1] GPN-MSA, a prominent DNA MSA transformer model, trains on sequence lengths of 128, which cannot capture global relationships inherent in genomic data

[2] excluding dummy and demo datasets

create varying gene outputs. Genes can vary in length from thousands to millions of base pairs, increasing the need for models with a large and effective context window.

DNA MSAs DNA Multiple Sequence Alignments (MSAs) are combinations of DNA sequences across different species. These sequences are aligned such that base pairs that evolve similarly are in the same column across genomes. Aligned columns in the MSA provide crucial evolutionary information between species. A DNA sequence for a species can be considered as a function of a different species' genome. This function consists of multiple mutations, such as insertions, deletions, and replacements. By aligning these sequences using MSA creation algorithms, models can implicitly extract conservation, coevolution, and homology information. DNA MSAs are also used to find motifs (short, repetitive sequences across genes). Implicit detection of these motifs in AI models can provide enhanced information for genome analysis. [42]

2.1 Transformer Models

Initial work in DNA language models involved leveraging the transformer architecture [44]. The transformer consists of multiple blocks [9], each containing a self-attention and MLP block. The self-attention block (see Eq. 1) functions as a fully connected sequence mixer, comparing all tokens to each other without any causal or window-based restrictions[3]. The comparison operation is computed using a dot product between two input space projections (Q, K). This dot product is passed through a row softmax and scaling operation before being multiplied by a value (V) projection. This acts as a weighted importance operation to emphasize important relationships while diminishing unimportant ones.

$$O = \mathbf{softmax}(\frac{QK^\top}{d_{attn}})V \tag{1}$$

The MLP block acts as a channel-wise mixer, increasing the size of the model dimension from d_{model} to d_{ff} and decreasing back down to the model dimension. This upscaling and downscaling projection allows for an integration with implicit memory that the transformer gains within its expanded MLP weights while training. Between both operations, a residual connection [23] and normalization [45] operation are included to prevent vanishing/exploding gradient problems during the backpropagation process. [15]

Transformer models that have been applied to DNA-MSA modeling show high accuracy in evolution-based modeling tasks. However, they have small context windows. This prevents transformers from attending to long-context relationships between regions, motifs, and other areas across genes.

2.2 Subquadratic Models

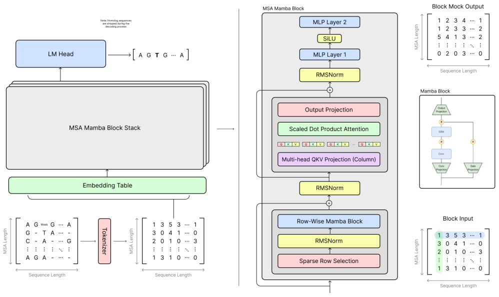

Figure 2: A diagram of the MSAMamba architecture, which leverages a selective-scan operation in the sequence dimension and a global-positioned attention process in the vertical dimension. [43]

Subquadratic models have initially been proposed as methods to decrease the expensive quadratic complexity of transformers in language modeling. However, they have also been applied to DNA modeling [19]. Some subquadratic models leverage long convolutions, which can be optimized to be

[3] Excluding masked tokens in the masked language modeling setting

computed in linear time [38]. These long convolutions can extract motif and region information, but they lack expressibility with few channels. In addition, long convolutions cannot attend to global relationships between regions due to the restrictivity of the kernel size and lack of state tracking across long contexts [33].

State-space model methods [21] have been proposed to fix the shortcomings of long convolution-based models. The original SSM formulation consists of four matrices that act as gates across a continuous data stream.

$$h_{t+1} = Ah_t + Bx_{t+1} \tag{2}$$

$$y_{t+1} = Ch_{t+1} + Dx_{t+1} \tag{3}$$

In the discrete-time formulation, these matrices are discretized[4] [37] with a Δ value representing a step size across a continuous sequence.

$$\bar{A} = \exp(\Delta A) \tag{4}$$

$$\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I)\Delta B \tag{5}$$

The original SSM formulation is linear time-invariant, allowing it to be computed as an efficient 1-dimensional convolution over a sequence. However, the Mamba SSM variant [19] makes the B, C, and D matrices input-dependent, allowing more adaptability using gating (The A matrix is determined using the HiPPO matrix formulation for long context data storage [20]). Although this model is no longer time-invariant, it does not use activation functions, allowing the model to be computed in an `O(N)` associative scan [4] using a parallelized, hardware-aware kernel [11].

The original Mamba formulation was tested on Genomic Benchmarks tasks [18] and had shown state-of-the-art performance on long-context tasks. However, it shows lower performance in shorter contexts, while transformers excel.

MSAMamba has been proposed as an alternative subquadratic DNA MSA model that leverages Mamba as the main sequence mixer [43]. While it shows improved performance compared to transformer-based models in long-context variant effect prediction tasks and Genomic Benchmarks tasks, it lacks training efficiency. MSAMamba runs a selective scan operation on all rows of the MSA, which can prevent batch size scaling during training[5].

3 Methods

We propose MSA-LM, a DNA MSA language model that improves the efficiency of previous methods by running a bidirectional selective scan operation on only one main sequence. MSA information is integrated into the main sequence using sparse attention across MSA data. In this section, we provide an overview of the components and structure of MSAMamba.

3.1 MSA Attention

The MSA-LM block architecture consists of two MSA-length attention processes that integrate MSA-level (column) and sequence-level (row) information. The first process is MSA-to-sequence attention, which alters the full column-wise self-attention process to attend only to the first sequence's base pairs as a query. This integrates MSA information into the main sequence while preventing inter-MSA attention[6]. In addition, this computation decreases the computational complexity of the MSA attention process from quadratic to linear[7], preserving the subquadratic nature of the model in both the sequence and MSA dimensions.

[4] Recent work has shown that using the fixed HiPPO matrix and discretization cannot perform well in state-tracking tasks [33]. We acknowledge this approach, but we use the original Mamba implementation due to its memory-efficient selective scan kernel

[5] MSAMamba was trained on a physical batch size of 2 1024 base-pair sequences on a NVIDIA P100 GPU

[6] mixing of inter-MSA information is unnecessary, as only evolutionary information relative to the main sequence (human genome) is required

[7] with reference to MSA Length

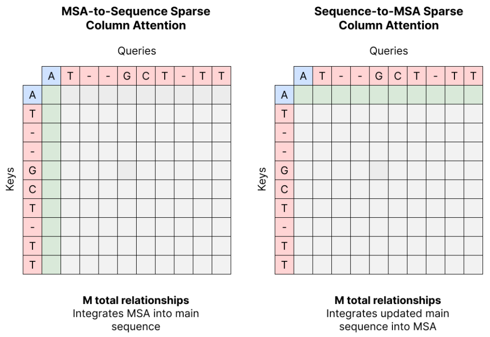
MSA-to-Sequence Sparse Column Attention
Queries
A T - - G C T - T T
Keys
A T - - G C T - T T
M total relationships
Integrates MSA into main sequence
Sequence-to-MSA Sparse Column Attention
Queries
A T - - G C T - T T
Keys
A T - - G C T - T T
M total relationships
Integrates updated main sequence into MSA

Figure 3: A diagram of the attention processes at the start and end of each MSA block. The MSA-to-sequence attention block acts as an integration of evolutionary information into the sequence, while the sequence-to-MSA attention process integrates sequence information into the MSA.

The second attention process is a sequence-to-MSA block, which integrates information present in the main sequence with MSA information across an MSA column. Similar to the first attention block, Inter-MSA relations are ignored for computational complexity benefits. With this formulation, other parts of the algorithm only have to process the top sequence, since MSA information is implicitly integrated and updated through these sparse attention processes.[8]

Each attention process consists of multiple sequence heads. This allows for multiple channels of information to be integrated into the main sequence position in a column. This is integrated into the channel formulation of the following convolution blocks.

3.2 Hydra + Convolution Block

The MSA-to-sequence block integrates MSA information into the target sequence. The auxiliary MSA tensor is saved for the later sequence-to-MSA block but is not used in the Hydra and convolution block computation[9]. There are $n_{channels}$ number of channels in the output, based on the number of query heads leveraged in the sequence-to-MSA attention block. These channels are then expanded to $expand \cdot n_{channels}$ by a long convolution block [17], which functions as a motif/region extractor [35]. All channels of the main sequence are then passed to a bidirectional Mamba formulation [25]. The output of the Mamba algorithm is passed through another long convolution, which decreases the number of channels back to $n_{channels}$. The output is passed to the sequence-to-MSA block to integrate sequence information back into the MSA augmented information tensor.

Both long convolution blocks are implemented with a fast Fourier transform algorithm[10] [7]. The first convolution operation expands the number of sequence channels. This expansion is done to increase the number of computation heads in the bidirectional Mamba model to learn a robust representation of the data. The second convolution decreases the number of channels.[11]

The output of the first convolution is passed as a multi-headed tensor to the bidirectional Mamba model. We leverage the Hydra model, which uses a quasiseparable matrix mixer [25], to implement the bidirectional mamba model[12]. Previous formulations use two Mamba models and add corresponding outputs. However, the quasiseparable matrix formulation allows for higher training and inference efficiency using two semiseparable matrix formulations [12].

[8] Both MSA attention processes leverage absolute position embeddings, which allows the model to identify each MSA species individually

[9] This is done to decrease the computational requirements of the main sequence mixer by integrating all information into one sequence

[10] FFT-based convolutions have shown higher performance at large kernel sizes

[11] For all models, we leverage an expansion factor of 2 and 4 main channels of computation. Scaling these factors can improve the representation capability of the model to handle longer contexts and more nuanced relationships.

[12] This model was chosen over other bidirectional Mamba formulations [41] due to increased computational efficiency

Algorithm 1 MSA-LM Masked Language Modeling

```
Input: MSA $x$ : (B, M, L, D), $M_{row}$ : (B, M), $y_t$ : (B, L, D), lr, $\theta$ (Model Params)
Output: $y$ : (B, L, D)
$h_0$ = mask($x$, p=0.15)
for $i = 1$ to $n_{layers}$ do
    $h_{sparse}$ = $h_i[M_{row}]$
    $O_{mamba}$ = scatter(Mamba($x_{sparse}$), $M_{row}$) + $h_i$
    $O_{att}$ = SelfAttention($O_{mamba}$) + $O_{mamba}$
    $h_{i+1}$ = MLP($O_{att}$)
end for
loss = CrossEntropy($h_{n_{layers}-1}[h_0 = MASK]$, $y_t$)
$\theta \leftarrow$ AdamW(lr)
```

4 Training

This section overviews the datasets and methods used to pre-train MSAMamba.

4.1 Pre-Training: MultiZ100Way

During model pre-training, we leverage the MultiZ100Way dataset, which consists of an MSA of the length of the human genome without any gap sequences[13] in the human sequence. It also consists of 99 auxiliary aligned sequences (with gap sequences) from related species. This data has been curated from the public UCSC Genome Browser [34]. We use a modified version of this dataset, which excludes ten auxiliary sequences of organisms that are very similar to those of humans [3]. This modification was done to decrease training time and memory requirements while losing minimal auxiliary information.[14]

This dataset was used to train MSA-LM and all MSA-based baseline models[15]. The same random seeds were also used for data shuffling and batch loading during pre-training for all models.

4.2 Data Preprocessing

The initial training data was collected from the MultiZ100Way dataset by sampling random locations across the genome and selecting DNA sequences based on the required context length for training (We use a context length of 1024 across all training steps).[16]

Data in the MultiZ100Way dataset was parsed using a tokenizer with a vocabulary size of 6. This consists of 4 nucleotides, one token for gap sequences, and one mask token. There was no need for `<PAD>` tokens due to all excerpts from the dataset being the same length.

This data was preprocessed based on the masked language modeling algorithm. This involves masking 15% of the sequence, where 80% of masked tokens are replaced with the `<MASK>` token, 10% is replaced with a random token, and the final 10% is not replaced [13].

Note: *Only the top sequence in the MSA (the human sequence) is masked due to the focus on the human genome, with other genomes being additional information*

4.3 Model Sizes

We trained 4 different MSA-LM models (see Table 1). Two of these models have a model dimension of 64, while others have a model dimension of 128. In all cases, we leverage an expansion factor of 2 for the SSM process. In addition, all models contain 3 MSA-LM layers except for one model with a model dimension of 128. Sequence length was gradually increased across model sizes.

[13] Gap sequences occur in MSAs when alignment moves around nucleotides to fit the proper evolutionary configuration, leaving placeholders for locations affected by shift/insertion/deletion mutations

[14] The MultiZ90Way is publicly accessible through HuggingFace datasets [29]

[15] Non-MSA models used as baselines were trained on the regular human genome without MSA augmentation

[16] We were unable to train on the entire genome due to lack of computational power

All models were trained on the same amount of data. However, only the final model (d_{model} = 128 and sequence length = 1024) is leveraged for its evaluations due to it having the highest performance based on training and validation loss results.

Table 1: Table of model configurations that underwent the training, fine-tuning, and evaluation processes with comparison to baseline models with similar parameters

d_{model}	d_{ssm}	n_{layers}	SEQ. LEN
64	128	3	128
64	128	3	512
128	256	3	1024
128	256	4	1024

4.4 Hyperparameter Selection

MSA-LM was trained using a masked language modeling formulation[17]. This method involves using Cross Entropy Loss on logit outputs to determine the accuracy of mask predictions (see Algorithm 1). The Adam optimizer was used for all training runs.

Before a full training run, we swept across multiple learning rates for an initial epoch of training[18]. The following learning rates were evaluated based on first-epoch performance: `3e-5`, `9e-5`, `3e-4`, `1e-3`, `8e-3`[19]. The learning rate of `3e-4` was found to perform the best during pre-training. A warmup scheduler is used to gradually increase the learning rate from 0 to `3e-4` across 25% of all gradient steps in the training run.

We train on sequences that are 1024 base pairs in length and use a physical batch size of 4 sequences. Due to computational constraints, we accumulate gradients across every 12 batches to increase the precision of gradient steps. With this formulation, the model is trained on 49152 base pairs per gradient step.

All training runs use a gradient clip value of `5.0` and a weight decay of `1e-3`. In addition, the Adam optimizer uses $(0.9, 0.95)$ as beta values.

5 Fine-Tuning and Evaluation

We provide an overview of the datasets and methods used for fine-tuning the MSA-LM model. In addition, we use similar formulations of the datasets for baseline models[20]. (Dataset processing information in A)

5.1 Fine-Tuning Method and Parameters

All fine-tuning tasks leveraged a full-parameter fine-tuning methodology. In addition, we padded all sequences during the fine-tuning process to a length of 1024. The only exception to this padding length is during the OMIM and ClinVar tasks, where we fine-tune two models on a sequence length of 1024 and two other models on a sequence length of 512.

All fine-tuning jobs leveraged the Adam optimizer and similar hyperparameters. We leveraged a learning rate of `3e-4`, a weight decay value of `1e-3`, and betas of $(0.9, 0.95)$.

Each fine-tuning process consisted of 3 epochs, each with 15000 steps. Fine-tuning was done using a batch size of 4 and gradient accumulation across every 8 iterations. This amounts to 32768 base pairs being attended to per gradient step.

[17] Masked language modeling was chosen over causal language modeling to learn full representations of DNA without restrictions from causal masks or specific decoding methods

[18] This epoch used the same shuffling seed to ensure equal performance

[19] a learning rate of `8e-3` was leveraged in Mamba and long-convolution-based models

[20] Datasets are modified to use MSA or single-sequence versions based on the capability of the specified baseline model

Task Name	GPN-MSA	MSAMamba	MSA-LM
ClinVar (512)	**0.967**	0.965	0.965
OMIM (512)	0.130	**0.131**	0.129
ClinVar (1024)	0.962	**0.978**	0.976
OMIM (1024)	0.118	0.139	**0.143**

Table 2: Evaluation of MSA-LM, GPN-MSA, MSAMamba, HyenaDNA, and DNABERT on variant effect prediction tasks using the AUROC metric for ClinVar and AUPRC for OMIM

Task Name	DNABERT	HyenaDNA	GPN-MSA	MSAMamba	MSA-LM
Mouse Enhancers	66.9	85.1	76.4	82.7	**86.8**
Coding vs Intergenomic	92.5	91.3	90.3	90.0	**92.7**
Human vs Worm	93.0	96.6	**98.9**	98.5	98.6
Human Enhancers Cohn	74.0	**74.2**	73.1	72.7	72.8
Human Enhancers Ensembl	85.7	89.2	89.3	88.8	**89.7**
Human Regulatory	88.1	93.8	93.5	94.4	**95.1**
Human Nontata Promoters	85.6	96.6	90.9	94.2	**97.0**
Human OCR Ensembl	75.1	80.9	76.8	**82.5**	81.9

Table 3: Evaluation of MSA-LM, GPN-MSA, MSAMamba, HyenaDNA, and DNABERT on GenomicBenchmarks tasks using top-1 accuracy (%) metric

6 Results

We show evaluation results for fine-tuned versions of MSA-LM on Genomic Benchmarks tasks and Long-Context ClinVar and OMIM Tasks[21]. In addition, we evaluate inference and training step times for MSA-LM and relevant baseline models.

6.1 Variant Effect Prediction

We evaluate MSA-LM on both the ClinVar and OMIM variant effect prediction tasks. Each variant effect prediction task involved two fine-tuning jobs: one with a context length of 512, and another with a context length of 1024. Results show that MSA-LM performs similarly to MSAMamba at context lengths of 1024 and slightly below average with reference to GPN-MSA regarding smaller context lengths.

This most likely occurs due to the MSA-LM's bias towards longer sequences during training. In contrast, GPN-MSA's full self-attention formulation is more robust at shorter context lengths. However, MSA-LM is advantageous in longer context lengths due to its training data being mostly from this distribution. The model shows similar performance to MSAMamba, with only minor differences in metrics. Overall, MSA-LM can generalize to long sequences for downstream tasks with a higher computational efficiency compared to previous methods.

6.2 Genomic Benchmarks

In addition to variant effect prediction tasks, we evaluate MSA-LM and baseline models on Genomic Benchmarks tasks. We fine-tune the model on sequences of length 1024, and we also evaluate the following baseline models:

- DNABERT (110 million parameters) - a BERT transformer architecture trained to represent DNA sequences
- HyenaDNA - long convolution-based architecture for DNA processing. The HyenaDNA-tiny version was used with a model dimension of 128 and a sequence length of 16k
- MSA-based models: GPN-MSA - a transformer model that processes DNA MSAs. MSAMamba - subquadratic MSA model leveraging Mamba selective scan

[21] maximum sequence length is capped at 1024 base pairs due to computational constraints

MSA-LM shows state-of-the-art performance in 3 Genomic Benchmarks tasks. While lacking in "OCR Ensembl" and "Enhancers Cohn" tasks, MSA-LM shows the highest performance when fine-tuning on regulatory sequences (e.g. promoters, enhancers). This shows that MSA-LM's training dataset may have been biased towards these regions during training. It is also possible that convolution operators inserted in the architecture can efficiently extract regulatory sequence information and influence across long-context inputs.

6.3 Training Complexity Analysis

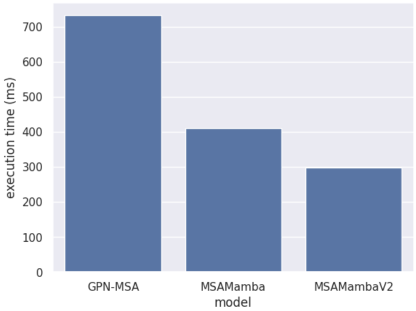

Figure 4: A comparison of time benchmarks for 3 DNA MSA sequence processing models. Each model is evaluated on one NVIDIA T4 GPU to determine the time taken to process the forward and backward pass of a batch of 4 1024-base-pair sequences

In addition to experimental evaluations, we provide a wall-clock complexity comparison of MSA-LM.

Wall clock time-based computational complexity evaluations of MSA-LM, along with 2 baseline models (GPN-MSA, MSAMamba) are computed. The time taken to evaluate the forward and backward pass of a batch of 4 1024-length sequences is computed[22]. All experiments use a model dimension of 128 and default derivations of other model dimensions[23]. We find that MSA-LM has the fastest training step performance. This is due to the relative efficiency of the sequence-level mixer operation in comparison to MSAMamba and GPN-MSA.

7 Discussion

MSA-LM is a promising architecture for DNA MSA analysis. Previous methods for DNA MSA analysis have lacked robust training on long context lengths due to computational complexity constraints. In addition, many previous models were not equipped to extract inductive biases inherent in DNA effectively. MSA-LM modifies the previous MSAMamba architecture to fix these problems. MSA-LM has higher training efficiency compared to previous methods due to sequence-level processing only happening on the main sequence instead of all sequences. This allows the model to be subquadratic in both the sequence and MSA dimension, and remove the restriction of low batch sizes due to expensive sequence-level computations. MSA-LM shows state-of-the-art/similar to state-of-the-art (SOTA) performance in long-context variant effect prediction tasks. The model also shows SOTA performance on Genomic Benchmarks tasks, showing particularly high performance in regulatory sequence analysis.

MSA-LM can be applied to mutation detection and effect prediction, as well as general causal analysis of DNA sequences for editing sequence generation or plasmid generation.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[22] experiment is repeated 20 times per model and averaged for accurate results

[23] e.g. d_{attn} for GPN-MSA will be $d_{model}/2$, Mamba uses a 2x expand on the original d_{model} value. Others can be found in relevant model repositories

[2] Bachir Balech, Saverio Vicario, Giacinto Donvito, Alfonso Monaco, Pasquale Notarangelo, and Graziano Pesole. Msa-pad: Dna multiple sequence alignment framework based on pfam accessed domain information. *Bioinformatics*, 31(15):2571–2573, March 2015.

[3] Gonzalo Benegas, Carlos Albors, Alan J. Aw, Chengzhong Ye, and Yun S. Song. Gpn-msa: an alignment-based dna language model for genome-wide variant effect prediction. October 2023.

[4] Guy E. Blelloch. Prefix sums and their applications. Technical Report CMU-CS-90-190, School of Computer Science, Carnegie Mellon University, November 1990.

[5] T.A. Brown. *Introduction to Genetics: A Molecular Approach*. CRC Press, 2012.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[7] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4479–4488. Curran Associates, Inc., 2020.

[8] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.

[9] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[10] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, 2023.

[11] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022.

[12] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality, 2024.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[14] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: pure attention loses rank doubly exponentially with depth. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2793–2803. PMLR, 18–24 Jul 2021.

[15] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth, 2023.

[16] Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models, 2023.

[17] Daniel Y. Fu, Elliot L. Epstein, Eric Nguyen, Armin W. Thomas, Michael Zhang, Tri Dao, Atri Rudra, and Christopher Ré. Simple hardware-efficient long convolutions for sequence modeling, 2023.

[18] Katarína Grešová, Vlastimil Martinek, David Čechák, Petr Šimeček, and Panagiotis Alexiou. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 24(1):25, 2023.

[19] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[20] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Re. Hippo: Recurrent memory with optimal polynomial projections, 2020.

[21] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022.

[22] A. Hamosh. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33(Database issue):D514–D517, December 2004.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[25] Sukjun Hwang, Aakash Lahoti, Tri Dao, and Albert Gu. Hydra: Bidirectional state space models through generalized matrix mixers, 2024.

[26] M I Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. 5 1986.

[27] Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention, 2022.

[28] Melissa J. Landrum, Jennifer M. Lee, George R. Riley, Wonhee Jang, Wendy S. Rubinstein, Deanna M. Church, and Donna R. Maglott. Clinvar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Research*, 42(D1):D980–D985, November 2013.

[29] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. Datasets: A community library for natural language processing, 2021.

[30] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. July 2022.

[31] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[32] Inês Lopes, Gulam Altab, Priyanka Raina, and João Pedro de Magalhães. Gene size matters: An analysis of gene length in the human genome. *Frontiers in Genetics*, 12, February 2021.

[33] William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models, 2024.

[34] Luis R Nassar, Galt P Barber, Anna Benet-Pagès, Jonathan Casper, Hiram Clawson, Mark Diekhans, Clay Fischer, Jairo Navarro Gonzalez, Angie S Hinrichs, Brian T Lee, Christopher M Lee, Pranav Muthuraman, Beagan Nguy, Tiana Pereira, Parisa Nejad, Gerardo Perez, Brian J Raney, Daniel Schmelter, Matthew L Speir, Brittney D Wick, Ann S Zweig, David Haussler, Robert M Kuhn, Maximilian Haeussler, and W James Kent. The ucsc genome browser database: 2023 update. *Nucleic Acids Research*, 51(D1):D1188–D1195, November 2022.

[35] Eric Nguyen, Michael Poli, Matthew G Durrant, Armin W Thomas, Brian Kang, Jeremy Sullivan, Madelena Y Ng, Ashley Lewis, Aman Patel, Aaron Lou, Stefano Ermon, Stephen A Baccus, Tina Hernandez-Boussard, Christopher Re, Patrick D Hsu, and Brian L Hie. Sequence modeling and design from molecular to genome scale with evo. February 2024.

[36] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, Stefano Ermon, Stephen A. Baccus, and Chris Ré. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. 2023.

[37] Georgia Pechlivanidou and Nicholas Karampetakis. Zero-order hold discretization of general state space systems with input delay. *IMA Journal of Mathematical Control and Information*, 39(2):708–730, April 2022.

[38] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models, 2023.

[39] Yanay Rosen, Yusuf Roohani, Ayush Agarwal, Leon Samotorčan, Stephen R. Quake, and Jure Leskovec. Universal cell embeddings: A foundation model for cell biology. November 2023.

[40] Melissa Sanabria, Jonas Hirsch, Pierre M. Joubert, and Anna R. Poetsch. Dna language model grover learns sequence context in the human genome. *Nature Machine Intelligence*, July 2024.

[41] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling, 2024.

[42] Mohammad Yaseen Sofi, Afshana Shafi, and Khalid Z. Masoodi. Chapter 6 - multiple sequence alignment. In Mohammad Yaseen Sofi, Afshana Shafi, and Khalid Z. Masoodi, editors, *Bioinformatics for Everyone*, pages 47–53. Academic Press, 2022.

[43] Vishrut Thoutam and Dina Ellsworth. MSAMamba: Adapting subquadratic models to long-context DNA MSA analysis. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, 2024.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[45] Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019.

A Fine-Tuning Datasets

A.0.1 Variant Effect Prediction Tasks

We use the OMIM and ClinVar Datasets during the evaluation process. The OMIM dataset relates gene sequences to different genetic disorders and their forms [22], while ClinVar relates aggregated gene variance information to overall human health [28]. Fine-tuning on this dataset evaluates a DNA MSA model's ability to perceive overall and individual gene relationships to determine its properties. The addition of MSA information provides key evolutionary information that is useful for these tasks [3].

These two datasets were used at two sequence lengths: 512, and 1024. MSA-LM is trained on sequence lengths of 1024, while previous models were trained with sequence lengths varying from 128 base pairs to 16 kilo-base pairs depending on model capabilities. We compare evaluations from the fine-tuning processes across these two context windows as a median context window for all models to generalize to.

The original OMIM and ClinVar datasets consisted of 128-length sequences. We modified these original sequences to include the area around the original sequence to add up to larger context lengths.

This tests models' abilities to detect and analyze specific mutations and segments within longer sequences.

All sequences were retrieved from the MultiZ90Way database given each sequence's chromosome index, start indices, and end indices. These sequences were not masked but passed as a tuple with a binary label as the fine-tuning target.

A.0.2 Genomic Benchmark Tasks

MSA-LM and other relevant models were also evaluated on the GenomicBenchmarks dataset [18]. This dataset consists of 8 different tasks relating to sequence-level classification[24]. The original GenomicBenchmarks datasets are single-sequence, containing only the human genome. However, we use start indices, stop indices, and chromosome metadata from the datasets along with the MultiZ90Way database to generate MSA versions of these evaluation datasets.

These datasets were not modified for different sequence lengths and were only trained on their original sequence lengths.

Note: *Ethical considerations were carefully addressed during the data curation/processing step. All genome data used in this study were obtained and modified from publicly available datasets (e.g., MultiZ100Way, OMIM, ClinVar)*

[24] We exclude the first three tasks seen in Table 6.2 from discussion, due to their relatively small size and designation as "demo" or "dummy" datasets