No Clear Winner at Small Scale: Comparing Modern Sequence Architectures and Training Strategies for Genomic Language Models

Vera Milovanović [1] **Antonio Orvieto** [2 3 4]

Abstract

Pretrained large language models based on a variety of sequence modeling architectures (e.g. Transformers, Mamba, Hyena) are increasingly being applied beyond natural language processing (NLP). In genomics, they have shown potential to reveal intricate structures and dependencies within DNA sequences, particularly within non-coding regions. To guide a principled development of training methods and architectures in the genomics domain, in this work we examine the most common classes of sequence modeling architectures found in language models and further explore transfer learning paradigms such as pretraining on large-scale external datasets as well as self pretraining (on the same data, using a reconstruction loss). In contrast to recent works, focusing specifically on finetuning large transformers, our results suggest that most recent recurrent models (Mamba) and implicit convolution based models (Hyena), that are increasingly used for genomic language models, might not offer an advantage over Attention-based Transformer models, especially after pretraining on the human reference genome. To enable thorough and controlled comparisons, we adopt a fixed training pipeline and limit our experiments to relatively small-scale model – an approach that still aligns well with the performance trends observed in recent studies.

1. Introduction

The Transformer architecture (Vaswani et al., 2017) and the self-supervised language modeling pretraining tasks have demonstrated an impressive capability to model sequences, particularly text (Radford et al., 2018), (Devlin, 2018), unlocking the rise of large language models (LLMs). This LLM pipeline has since been extended beyond natural language to other domains such as computer vision (Dosovitskiy et al., 2021) and biological sequences, including proteins (Lin et al., 2023; Brandes et al., 2022), RNA (Shulgina et al., 2024), and DNA (Ji et al., 2021; Dalla-Torre et al., 2024; Nguyen et al., 2023; Schiff et al., 2020; Nguyen et al., 2024a).

A central component of the architecture is the softmax Attention layer (Bahdanau et al., 2014), which enables global information exchange across the sequence. However, this layer has a critical drawback when modeling long sequences due to its quadratic time and space complexity relative to the sequence length. This limitation has motivated the development of subquadratic sequence mixing layers. Notable examples include state space models (SSMs) such as S4 (Gu et al., 2022), S6 (Gu & Dao, 2024), and the related Hyena operator (Poli et al., 2023), which scale linearly with the sequence length. These models have inspired research into subquadratic LLMs for genome analysis, given the long-range dependencies spanning over 100k+ nucleotides (Nguyen et al., 2023; Avsec et al., 2021).

gLMs So far, genomic language models (gLMs) have been trained and benchmarked at different scales: from 400K to 70B (Nguyen et al., 2023), (Schiff et al., 2020), (Ku et al., 2025); using different sequence modeling architectures: Convolutional Neural Networks (Bo et al., 2025), Transformers (Ji et al., 2021), (Sanabria et al., 2024), (Dalla-Torre et al., 2024), SSM (Schiff et al., 2020), Hyena (Nguyen et al., 2023) and their hybrids (Nguyen et al., 2024a), (Ma et al., 2025); pretraining objectives: causal language modeling (CLM) (Poli et al., 2023), (Ku et al., 2025), and masked language modeling (MLM) (Dalla-Torre et al., 2024), (Schiff et al., 2020), (Sanabria et al., 2024); tokenizers: k-mer (Dalla-Torre et al., 2024), (Sanabria et al., 2024), character-level (Nguyen et al., 2023), (Schiff et al., 2020) and pretraining corpora: human reference genome (Genome Reference Consortium, 2013) (Nguyen et al., 2023), (Schiff et al., 2020), multi-species genomes (Dalla-Torre et al., 2024), (Nguyen et al., 2024a). *Due to the many potential axes of variation in the pipeline design, it is hard to make conclusive statements about model architectures*, especially

[1]University of Tübingen, Germany [2]ELLIS Institute Tübingen, Germany [3]Max Planck Institute for Intelligent Systems, Germany [4]Tübingen AI Center, Germany. Correspondence to: Vera Milovanović <milovanovic.veraa@gmail.com>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

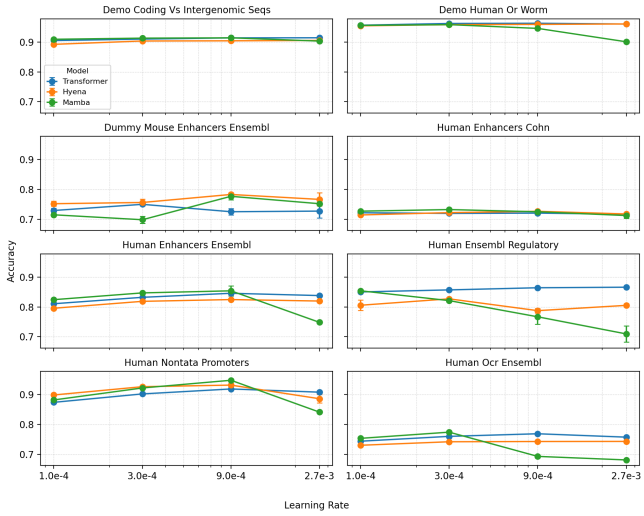

Figure 1. Learning rate sensitivity of finetuned models with **context length of 4096 tokens**. Models are pretrained on human reference genome (HG38) and finetuned on each of the tasks separately. Mean test accuracy and its variation across two random seeds for the train-validation datasets split on the held-out test set is reported. Variation of the test accuracy is given as the difference between maximum and minimum deviation from the mean. The x axis is in the log scale.

when these are benchmarked together with tokenization strategies and pretraining objectives (e.g. causal Mamba vs. masked self-Attention). In modern literature, such practices have also led to the surprising conclusion that small Hyena and SSM-based gLMs outperform orders of magnitude larger Transformer models (Nguyen et al., 2024b), (Schiff et al., 2020). We suspect that there might be some confounding factors (datasets, tokenizer, training time), so in this work we focus on comparing sequence model classes for genome modeling while keeping the rest of the pipeline fixed. We also focus on small models (1.6 M parameters), to test if at that scale SSM and Hyena based gLMs are indeed better. This setting is supported by findings that many tasks in genomics smaller models perform on pair, if not better than orders of magnitude larger ones (Xu et al., 2024). Moreover, focusing on small scale models is important in the context of developing them within the academic research labs for the specific genomics tasks.

Pretraining What current gLMs have in common is the goal of learning generalizable features from long-range genomic data for transfer learning to downstream tasks. To achieve that, the effort in the community has been dominantly going along the lines of scaling models in context length, data (by using whole-genome sequences) and compute (Dalla-Torre et al., 2024), (Nguyen et al., 2024a), (Ku et al., 2025), (Ma et al., 2025) as in the NLP field (Kaplan et al., 2020). One study (Tang & Koo, 2024), focusing on learning cis-regulatory patterns in the non-coding genome, concludes that gLMs pretrained on whole genomes do not offer significant advantages over traditional machine learning models using one-hot encoded sequences. Hence, the usage of a large pretraining corpus and the resources needed for learning embeddings are unjustified for the task at hand. It also finds that using embeddings from a model trained in a supervised manner on data more relevant to the downstream task is most predictive of downstream performance. These findings highlight that pretraining on next or masked token prediction task is not universally effective for constructing good features and that the choice of pretraining task should be guided by domain-specific requirements. However, this study only probes learned embeddings of gLMs and does not evaluate downstream performance after finetuning. In contrast, we investigate whether data-driven initialization of gLMs aids in constructing effective features for finetuning, especially in the context of the small-scale models.

Self pretraining Similarly to (Tang & Koo, 2024), several studies from the field of NLP (Krishna et al., 2022) and computer vision (El-Nouby et al., 2021) have shown that self-supervised pretraining with denoising objectives on large-scale, task-agnostic data does not significantly outperform self-supervised pretraining on task-specific data followed by supervised finetuning. This approach is referred to as self pretraining (**SPT**). Related to this, it has been demonstrated that domain-adaptive pretraining — continued pretraining of a pretrained language model on unlabeled task-specific data or a small domain-relevant corpus can provide significant benefits (Gururangan et al., 2020). Furthermore, (Amos et al., 2023) demonstrates across multiple long-range tasks and data scales that models like S4 (Gu et al., 2022), its diagonalized variant (Gupta et al., 2022), and Transformers can achieve substantial performance gains (up to 30 accuracy points on the LRA benchmark (Tay et al., 2020)) when trained with SPT, making them competitive with more specialized SSM architectures. Motivated by these findings, we ask whether SPT provides similar benefits in genomics and whether its utility depends on the underlying sequence architecture.

To summarize, our main contributions are as follows:

1. "Head-to-head" comparison of three sequence modeling architectures: Transformers, Mamba and Hyena at small scale on a task of regulatory annotation tasks,

2. We examine whether there is an interplay between architecture and a (pre)training method. To do so, we evaluate:

   (a) supervised training from randomly initialized weights (from scratch),

Table 1. Average classification test accuracies across Genomic Benchmark (Grešová et al., 2023) tasks consisting of real data – more relevant for the analysis, focused on identifying regulatory elements and open chromatin regions. The tasks include: Human Enhacers Cohn, Human Enhancers Ensembl, Human Regulatory, Human Nontata Promoters, Human Open Chromatin Accessibility (OCR) Ensembl. See Tables 2, 3, 4 for results on each task individually. We highlight in **bold** the best (pre)training strategy for each of the model class.

PRETRAINING	TRANSFORMER	HYENA	MAMBA
NO (FROM SCRATCH)	81.88	**84.20**	**84.08**
HG38	82.43	81.07	83.25
TASK-SPECIFIC	81.83	84.10	83.61
DOMAIN-SPECIFIC	**82.52**	83.63	82.22

(b) pretraining on whole-genome sequence - human reference genome,

(c) self pretraining (SPT) on task-specific data.

We conduct our analysis in a principled manner by comparing models trained at the same scale, using the same tokenizer, data, and training procedure. We also perform extensive hyperparameter tuning to ensure models operate as close to optimal as possible.

Although limited by the variety of datasets and benchmarks considered, we believe our results provide an interesting step towards understanding the impact of modern architectural choices and (pre)training methods in real-world applications oriented toward solving specific tasks, with potentially very small amount of data (by language models standards).

In the described setting, we find that all three considered sequence modeling architectures, on average, perform on-par over different tasks from the Genomic Benchmark and (pre)training methods. Notably, when looking at individual task performances, supervised training from random initialization often outperforms transfer-learning paradigms. However, this is less the case with Transformer architecture, which appears to benefit a bit more from self pretraining. We also showcase the overall ineffectiveness of transfer learning by demonstrating that being better at predicting the next token in the genome does not translate to better performance on different task, such as identifying regulatory elements and open chromatin regions in the genome.

2. Experiments

2.1. Data

For pretraining experiments we use human reference genome (Genome Reference Consortium, 2013), which is commonly used in studies in the realm of gLMs. It consists of around 3.5 billion nucleotide base pairs (tokens) in the training split.

For evaluation, we focus on Genomic Benchmark (Grešová et al., 2023), as it is present in almost all gLM studies. It includes eight datasets centered around regulatory elements — such as promoters, enhancers, and open chromatin regions—from three model organisms: human, mouse, and roundworm. We use character-level tokenizer to encode a sequence vocabulary of 4 nucleotides A, T, C, G.

2.2. Methods

We keep the number of parameters fixed at 1.6 million, as well as number of layers at 2. Unless stated otherwise, we do a hyperparemeter search over four learning rate values (1e-4, 3e-4, 9e-4, 2.7e-3), and two batch sizes (128, 256). All supervised training runs for 10 epochs, on two different train-validation dataset splits. The final evaluation is done on the held-out test set. We report the mean $\pm$ the difference between the maximum and minimum deviation from the mean. When pretraining, we use causal language modeling objective with 10000 optimizer steps. Details about specific values of hyperparameters can be found in Appendix A.1. To prevent overfitting, we monitor the validation loss. We test the model that achieved the lowest validation loss during pretraining. We then further finetune the best-performing model on the downstream task.

Training from scratch. To establish a baseline for evaluating the benefits of pretraining strategies, we train all models from random initialization, using the same recipe as (Grešová et al., 2023).

Pretraining and finetuning. We experiment with three context lengths during pretraining (1024, 2048 and 4096) to assess whether a larger genomic context improves downstream performance.

Self pretraining. Given task-specific data (X_{train}, y_{train}), the first stage of SPT trains the model on X_{train} alone using an autoregressive next-token prediction objective, minimizing a cross-entropy loss. We construct X_{train} in two ways: (1) by combining training sequences from all downstream tasks in the Genomic Benchmark, and (2) using training sequences from each individual task. Models are evaluated on all downstream tasks in setup (1), and only on the test set of the pretrained task in setup (2). By comparing the performance across the two setups we test whether pretraining on multiple related tasks of regulatory elements brings any advantage compared to using only individual task data.

3. Results

In our setting, upon fixing the whole model backbone to be the same and tuning Attention (Transformer), Hyena and Mamba models, we demonstrate that all models (when max pooling on the pretraining strategy) perform quite similarly on average across the most relevant tasks (human regula-

Table 2. **Classification accuracies for the Transformer model with and without pretraining across various genomic datasets**. Context length of 4096 is used for pretraining on the human reference genome (HG38).Mean test accuracy and its variation across two random seeds for the train-validation datasets split on the held-out test set is reported. Variation of the test accuracy is given as the difference between maximum and minimum deviation from the mean. The best mean performance for each task is emphasized in **bold**, the second best is underlined.

MODEL	TRANSFORMER			
PRETRAINED	NO	HG38	TASK-SPECIFIC	DOMAIN-SPECIFIC
MOUSE ENHANCERS	**79.75 ± 0.00**	75.00 ± 0.00	76.24 ± 0.00	75.85 ± 0.01
CODING VS INTERGENOMIC	92.13 ± 0.00	91.45 ± 0.00	**93.08 ± 0.00**	91.07 ± 0.00
HUMAN VS WORM	95.82 ± 0.12	96.31 ± 0.00	**96.33 ± 0.00**	96.31 ± 0.00
HUMAN ENHANCERS COHN	70.82 ± 0.00	72.18 ± 0.00	71.76 ± 0.49	**72.70 ± 0.00**
HUMAN ENHANCERS ENSEMBL	**87.73 ± 0.00**	84.58 ± 0.00	85.23 ± 0.00	84.76 ± 0.00
HUMAN REGULATORY	**87.43 ± 0.00**	86.61 ± 0.00	82.51 ± 0.00	86.45 ± 0.00
HUMAN NONTATA PROMOTERS	92.04 ± 0.00	91.90 ± 0.00	**92.49 ± 0.00**	91.65 ± 0.00
HUMAN OCR ENSEMBL	71.36 ± 0.00	76.87 ± 0.00	**77.16 ± 0.00**	77.06 ± 0.00

Table 3. **Classification accuracies for the Hyena model with and without pretraining across various genomic datasets**. Context length of 4096 is used for pretraining on the human reference genome (HG38). Mean test accuracy and its variation across two random seeds for the train-validation datasets split on the held-out test set is reported. Variation of the test accuracy is given as the difference between maximum and minimum deviation from the mean. The best mean performance for each task is emphasized in **bold**, the second best is underlined.

MODEL	HYENA			
PRETRAINED	NO	HG38	TASK-SPECIFIC	DOMAIN-SPECIFIC
MOUSE ENHANCERS	**79.75 ± 1.24**	78.31 ± 0.00	79.53 ± 0.01	78.45 ± 0.00
CODING VS INTERGENOMIC	90.68 ± 0.23	90.56 ± 0.00	**90.72 ± 0.00**	90.55 ± 0.00
HUMAN VS WORM	**96.19 ± 0.12**	96.08 ± 0.00	96.07 ± 0.00	95.99 ± 0.00
HUMAN ENHANCERS COHN	72.58 ± 0.08	**72.66 ± 0.00**	72.74 ± 0.39	72.49 ± 0.38
HUMAN ENHANCERS ENSEMBL	**89.33 ± 0.00**	82.47 ± 1.00	87.30 ± 0.00	86.60 ± 0.00
HUMAN REGULATORY	**87.30 ± 0.00**	82.71 ± 0.00	86.57 ±0.00	86.07 ± 0.00
HUMAN NONTATA PROMOTERS	94.95 ± 0.00	93.19 ± 0.00	**95.71 ± 0.00**	95.08 ± 0.00
HUMAN OCR ENSEMBL	76.83 ± 0.00	74.31 ± 0.00	**78.19 ± 0.00**	77.91 ± 0.00

tory elements and open chromatin regions) and (pre)training methods; see Table 1. In particular, Hyena and Mamba models achieve the best performance on average when trained in a supervised fashion from random initialization, and reach around 1.5 accuracy points more compared to the pretrained Transformer model on domain-specific data — which we found to be the best overall option. However, a deeper examination beyond average performance showcases some peculiar differences.

Mamba is most sensitive to hyperparameters. Firstly, we find that Transformer and Hyena models have more consistent performance across wider range of learning rates compared to Mamba, a property which makes them favorable in the case of a very limited compute budget for tuning: their performance does not degrade significantly if the (near-)optimal learning rate is not found. To illustrate this, in Figures 3 - 1, we show the sensitivity of downstream task performance with respect to the learning rate. For instance, in Figure 3, for the Human Enhancers Ensembl task, the accuracy gap between the best and the worst performing learning rate for the Mamba model is up to 30 accuracy points, making the model potentially as good as random. We find similar effects across models when they are trained from scratch or self-pretrained. A similar finding was reported by (Okpekpe & Orvieto, 2025) on in-context recall tasks.

From-scratch performance is often best. Secondly, from the standpoint of assessing *architecture-agnostic* effectiveness of (self-)pretraining methods, we find that on most of the tasks, vanilla supervised training (i.e., from scratch) actually performs best; see Tables 2 - 4. This suggests that processing orders of magnitude larger genomic datasets might not necessarily help build feature representations or processing mechanisms useful for the subsequent adaptation to the downstream task. However, we note that a confounding reason might be the fixed model size used in our experiments (1.6 million parameters). Furthermore, in the case of pretraining on the genome-wide pretrain corpus (HG38) with a causal language modeling objective, the model may learn to predict genome regions irrelevant to the downstream task at hand. One might expect that pretraining on the task-specific data could help build better representations. However, our

Table 4. **Classification accuracies for the Mamba model with and without pretraining across various genomic datasets**. Context length of 4096 is used for pretraining on the human reference genome (HG38). Mean test accuracy and its variation across two random seeds for the train-validation datasets split on the held-out test set is reported. Variation of the test accuracy is given as the difference between maximum and minimum deviation from the mean. The best mean performance for each task is emphasized in **bold**, the second best is underlined.

MODEL	MAMBA			
PRETRAINED	NO	HG38	TASK-SPECIFIC	DOMAIN-SPECIFIC
MOUSE ENHANCERS	**79.89 ± 2.48**	77.69 ± 0.00	78.31 ± 0.00	75.21 ± 0.00
CODING VS INTERGENOMIC	90.64 ± 0.15	**91.43 ± 0.00**	91.23 ± 0.00	91.14 ± 0.00
HUMAN VS WORM	96.42 ± 0.11	95.85 ± 0.00	96.40 ± 0.00	**96.51 ± 0.00**
HUMAN ENHANCERS COHN	**74.05 ± 0.06**	73.24 ± 0.00	72.65 ± 0.00	73.32 ± 0.00
HUMAN ENHANCERS ENSEMBL	85.27 ± 0.00	85.40 ± 0.00	**88.78 ± 0.00**	84.27 ± 0.00
HUMAN REGULATORY	**87.71 ± 0.00**	85.41 ± 0.00	85.84 ± 0.00	83.54 ± 0.00
HUMAN NONTATA PROMOTERS	**94.99 ± 1.18**	94.81 ± 0.00	92.46 ± 0.00	91.79 ± 0.00
HUMAN OCR ENSEMBL	**78.36 ± 0.00**	77.41 ± 0.00	78.34 ± 0.00	78.19 ± 0.00

experiments show that, on average across the benchmark, this is not the case. However, specific tasks and model classes - like Hyena, see Table 1 might benefit). The reason might be a small pretraining corpus, consisting of just the downstream task data.

(Self-)Pretraining is slightly more effective for Attention. Further, regarding our question about the potential *interplay* between the (pre)training method and considered sequence architectures, based on Table 1, we find that on average, there is a slight benefit of pretraining Transformer architecture, compared to Hyena and Mamba. This is in line with the previously discussed findings (Amos et al., 2023) in Section 1. However, we emphasize that, when it comes to establishing relationship between each model and training with the supervised objective from random initialization, on average, all models perform rather similarly.

Pretraining on longer sequences does not help. Another interesting question that arises is whether the improved pretraining performance translates to better downstream task accuracy. However, since our benchmark focuses on short-range tasks, a longer context does not necessarily yield improved results. This is supported by Figure 2, which shows that although perplexity on the test set decreases as the model processes more tokens, Table 5 indicates that average accuracy over all tasks does not improve with increasing context length during pretraining. Interestingly, for the Hyena architecture, the trend is actually reversed: on average, across all tasks, the downstream performance slightly decreases as the context length (and thus the number of seen tokens) increases. It is important to note that model size remains fixed across these experiments. Additionally, we remark that perplexity values remain relatively high, even though the vocabulary size is small.

4. Discussion and Conclusion

In this work, we use the Genomic Benchmark (Grešová et al., 2023) to compare Transformer (Attention-based), Mamba (recurrence-based) and Hyena (gated long-convolution-based) at small scale. We give evidence against the common belief that Transformer models (GPT-like) are inferior compared to Mamba and Hyena architectures at this scale. We also show that pretraining does not offer substantial advantages: we analyzed downstream task performance when models are pretrained on the whole human genome, downstream task training data, or domain-specific data. In our setting, we find no strong evidence that any of these strategies is effective across architectures. Yet, we report that, on average, Attention-based model does slightly benefit from (self-)pretraining. Finally, show that for a fixed model size, seeing more tokens during the pretraining phase — hence decreasing perplexity — does not translate to better downstream task performance.
This work and its findings advocate for more principled model comparisons and exploring the training strategies for small genomic language models, beyond what we have seen in the NLP and other fields, but more tailored to biological sequences.

Limitations. Our work could be further improved by including more benchmarks with a variety of genomics-relevant tasks, in particular those that consist of long-range tasks. It remains to be seen whether our findings hold in larger-scale models and when training with longer context lengths.

Impact Statement

The aim of this work is to advance the field of machine learning. Although our work has a potential societal benefits, it could also be a subject to misuse, as all other works

focusing on language model architectures.

References

Amos, I., Berant, J., and Gupta, A. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. *arXiv preprint arXiv:2310.02980*, 2023.

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D. R. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Bo, Y., Mao, W., Shao, Y., Bai, W., Ye, P., Ma, X., Zhao, J., Chen, H., and Shen, C. Revisiting convolution architecture in the realm of dna foundation models. *arXiv preprint arXiv:2502.18538*, 2025.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Lopez Carranza, N., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., de Almeida, B. P., Sirelkhatim, H., et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 21(11):1–11, 2024.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 2022.

Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., and Grave, E. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021.

Genome Reference Consortium. Genome reference consortium human build 38 (grch38). https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.26/, 2013. National Center for Biotechnology Information.

Grešová, K., Martinek, V., Čechák, D., Šimeček, P., and Alexiou, P. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 24(1):25, 2023.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Gupta, A., Gu, A., and Berant, J. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, 2022.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Krishna, K., Garg, S., Bigham, J. P., and Lipton, Z. C. Downstream datasets make surprisingly good pretraining corpora. *arXiv preprint arXiv:2209.14389*, 2022.

Ku, J., Nguyen, E., Romero, D. W., Brixi, G., Yang, B., Vorontsov, A., Taghibakhshi, A., Lu, A. X., Burke, D. P., Brockman, G., et al. Systems and algorithms for convolutional multi-hybrid language models at scale. *arXiv preprint arXiv:2503.01868*, 2025.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Ma, M., Liu, G., Cao, C., Deng, P., Dao, T., Gu, A., Jin, P., Yang, Z., Xia, Y., Luo, R., et al. Hybridna: A hybrid transformer-mamba2 long-range dna language model. *arXiv preprint arXiv:2502.10807*, 2025.

Nguyen, E., Poli, M., Faizi, M., Thomas, A., Wornow, M., Birch-Sykes, C., Massaroli, S., Patel, A., Rabideau, C., Bengio, Y., et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36: 43177–43201, 2023.

Nguyen, E., Poli, M., Durrant, M. G., Kang, B., Katrekar, D., Li, D. B., Bartie, L. J., Thomas, A. W., King, S. H., Brixi, G., et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723): eado9336, 2024a.

Nguyen, E., Poli, M., Faizi, M., Thomas, A., Wornow, M., Birch-Sykes, C., Massaroli, S., Patel, A., Rabideau, C., Bengio, Y., et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. In *Advances in Neural Information Processing Systems*, 2024b.

Okpekpe, D. and Orvieto, A. Revisiting associative recall in modern recurrent models. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025. URL https://openreview.net/forum?id=CcqAd5RPk5.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Sanabria, M., Hirsch, J., Joubert, P. M., and Poetsch, A. R. Dna language model grover learns sequence context in the human genome. *Nature Machine Intelligence*, 6(8): 911–923, 2024.

Schiff, Y., Kao, C.-H., Gokaslan, A., Dao, T., Gu, A., and Kuleshov, V. Caduceus: Bi-directional equivariant long-range dna sequence modeling, 2024. *URL https://arxiv.org/abs/2403.03234*, 2020.

Shulgina, Y., Trinidad, M. I., Langeberg, C. J., Nisonoff, H., Chithrananda, S., Skopintsev, P., Nissley, A. J., Patel, J., Boger, R. S., Shi, H., et al. Rna language models predict mutations that improve rna function. *Nature Communications*, 15(1):1–17, 2024.

Tang, Z. and Koo, P. K. Evaluating the representational power of pre-trained dna language models for regulatory genomics, march 2024, 2024.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Xu, Z., Gupta, R., Cheng, W., Shen, A., Shen, J., Talwalkar, A., and Khodak, M. Specialized foundation models struggle to beat supervised baselines. *arXiv preprint arXiv:2411.02796*, 2024.

A. Appendix

A.1. Experimental details

In this section, we give more details about the experimental setup. Our implementation is inspired by HyenaDNA (Nguyen et al., 2023) and (Schiff et al., 2020). Additionally, the code repository is cloned from Caduceus and further developed to our needs.

A.2. Data

A.2.1. Downstream tasks

Data for downstream tasks are taken from Genomic Benchmark (Grešová et al., 2023). It consists of eight datasets that focus on regulatory elements (promoters, enhancers, open chromatin region) from three organisms - human, mouse and roundworm. The tasks are binary classification, except for human ensembl regulatory tasks, which is has three classes to predict from. Sequences present in the datasets vary in length and number of sequences. It comes with train and test splits. For more details, please refer to the original paper.

A.2.2. Pretraining

For pretraining we either use human reference genome HG38 (Genome Reference Consortium, 2013), take training data of a specific downstream task, or construct the dataset by combining all training data of downstream tasks. HG38 dataset is obtained from the Enformer study (Avsec et al., 2021) and contains 34,021 training, 2,213 validation, and 1,937 test sequences from the human genome. Chromosome 14 is used exclusively for test set, chromosome 14 for test and validation sets. Sequences from other chromosomes are used for all three splits.

A.3. Model details

We use 3 models for our experiment: Decoder-only Transformer, Hyena and Mamba. In the end, all of the three considered architectures are of size 1.6 M, which was chosen in the hyperparameter search of parameters that affect model size (see details below). All models use two layers. For Hyena and Transformer model that means layers with blocks of a sequence mixer layer (Flash Attention (Dao et al., 2022) or Hyena operator (Poli et al., 2023)) and MLP. For Mamba model, a layer consists of two branches with MLP layers, where in one branch it is followed by convolution, sequence mixing layer (S6) and another MLP layer. In our setting, we use 8-headed Transformer model, with rotary positional embeddings and we also implement a QK-norm.
We also try two model widths (dimension of the input embeddings): 128 and 256. For the pretraining stage, we evaluate models based on the validation perplexity and chose the best hyperparameter setting. When using human reference genome, we sweep over three sequence lengths: 1024, 2048 and 4096. In the end, all models achieved the best validation perplexity being trained with 4096 context length and model width of 256. For self pretraining on all tasks, we sweep over context lengths of 256, 512 and 1024, as these are the usual lengths present in the task considered. When training on supervised objective, as well as self pretraining on the individual downstream task, maximum input length is set to to maximum length of the task it is trained on.

A.4. Training and evaluation

Supervised training from a random initialization is performed for a maximum of 200 epochs. Finetuning on all initializations is performed for 10 epochs. We apply early stopping based on validation accuracy to prevent overfitting. Pretraining (on all pretraining corpora) is done using causal language modeling task over 10000 steps. In the case of self pretraining, we perform early stopping on the validation perplexity and use the best model for downstream task finetuning.
For all training tasks we sweep learning rate from 1e-4 to 6e-3 with a power step of 3, which amounts to 4 different values and try batch sizes of 128 and 256.
We use AdamW optimizer with weight decay 0.1 and for learning rate schedule cosine annealing with linear warmup of duration 10% and minimum value of 1e-6. We use Nvidia A100 GPU for all experiments.
All supervised trainings are evaluated using 5-fold cross-validation (CV) with different train-validation random seeds 0-4. We report the mean $\pm$ the difference between the maximum and minimum deviation from the mean. Pretraining stage is evaluated on validation perplexity. Random seed globally to 2222.

A.5. Evaluation of performance transferability between pretraining and finetuning as a function of number of seen tokens

Table 5. Classification accuracies of finetuned Transformer, Hyena, and Mamba models pretrained on **different pretraining context lengths** from human reference genome HG38. All models are pretrained trained for the batch size of 256 and for 10000 steps. The number of seen tokens during pretraining depends only on the context length. All models are finetuned for 10 epochs. We report mean accuracy over two random seeds for the train-validation dataset split. Held-out test set remains the same. Performance does not strongly depend on context length during pretraining. In **bold** we emphasize the performance that surpasses others within the same model class by at least 1 %.

Dataset / Context lenght	Transformer			Hyena			Mamba		
	1024	**2048**	**4096**	**1024**	**2048**	**4096**	**1024**	**2048**	**4096**
Mouse Enhancers	**77.89**	75.21	75.00	**77.48**	76.65	78.31	74.17	76.86	77.69
Coding vs Intergenomic	91.25	90.95	91.45	90.64	90.69	90.56	91.05	90.67	91.43
Human vs Worm	96.24	96.16	96.31	96.05	96.11	96.08	96.51	96.32	95.85
Human Enhancers Cohn	72.28	72.84	72.18	72.17	72.38	72.66	72.34	72.47	73.24
Human Enhancers Ensembl	83.80	84.11	84.58	**86.42**	84.78	82.47	83.09	83.35	**85.40**
Human Regulatory	87.11	87.27	86.61	**87.07**	85.99	82.71	86.14	83.62	85.41
Human Nontata Promoters	87.95	91.55	91.90	93.50	**94.51**	93.19	94.63	94.63	94.81
Human OCR Ensembl	75.97	76.31	76.87	**76.95**	75.75	74.31	78.66	78.82	77.41
Avg. Accuracy	84.06	84.30	84.36	85.04	84.61	83.79	84.57	84.59	85.15

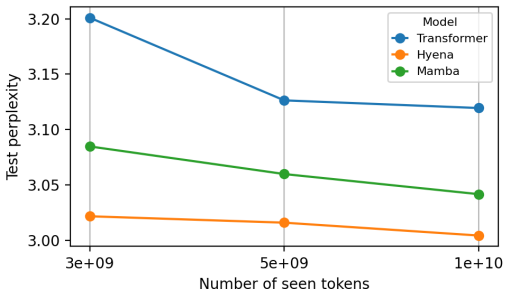

Figure 2. Test perplexity after pretraining model on different number of tokens for different gLM architectures. Batch size is chosen to be 256 (which also corresponds to the best - in terms of validation loss) and number of steps 10000. Context length is set to 1024, 2048 and 4096, which corresponds to different number of seen tokens.

A.6. Learning rate sensitivity for models with different context lenghts

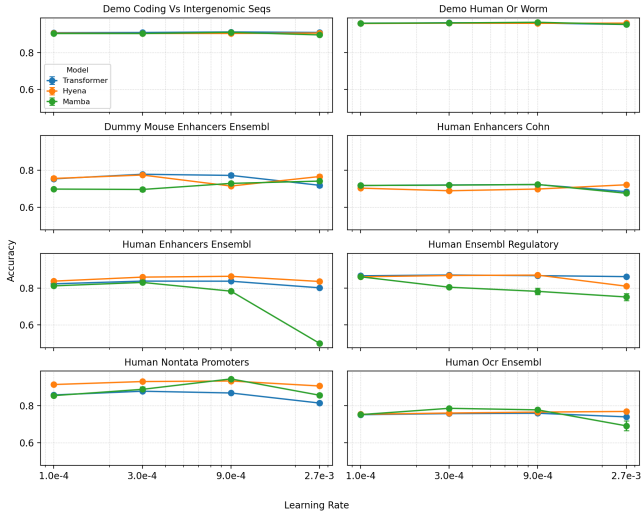

Figure 3. Learning rate sensitivity of finetuned models with **context length of 1024 tokens**. Models are pretrained on human reference genome (HG38) and finetuned on each of the tasks separately. Mean test accuracy and its variation across two random seeds for the train-validation datasets split on the held-out test set is reported. Variation of the test accuracy is given as the difference between maximum and minimum deviation from the mean.

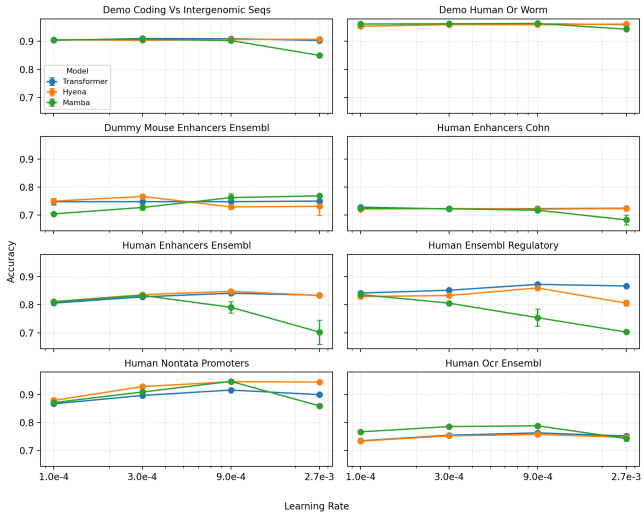

Figure 4. Learning rate sensitivity of finetuned models with **context length of 2048 tokens**. Models are pretrained on human reference genome (HG38) and finetuned on each of the tasks separately. Mean test accuracy and its variation across two random seeds for the train-validation dataset split on the held-out test set is reported. Variation of the test accuracy is given as the difference between maximum and minimum deviation from the mean. The x axis is in the log scale. The x axis is in the log scale.