The Genomics Long-Range Benchmark: Advancing DNA Language Models

Evan Trop[*]
InstaDeep
e.trop@instadeep.com

Yair Schiff [*]
Cornell University

Edgar Marroquin[*]
Cornell University

Chia-Hsiang Kao
Cornell University

Aaron Gokaslan
Cornell University

McKinley Polen
InstaDeep, MIT

Mingyi Shao
Cornell University

Bernardo P. de Almeida
InstaDeep

Thomas Pierrot
InstaDeep

Yang Li
University of Chicago

Volodymyr Kuleshov
Cornell University

Abstract

The advent of language models (LMs) in genomics necessitates benchmarks that can assess models' capabilities and limitations. In contrast to protein models, DNA LMs can be used to study non-coding regions of the genome and must account for unique challenges, especially interactions across long sequence lengths. However, existing benchmarks for DNA LMs are defined over short sequence datasets and can involve tasks that are often not considered to be biologically meaningful. Here, we present the Genomics Long-Range Benchmark (LRB), which focuses on biologically meaningful tasks and supports long-range contexts. We complement our benchmark with fine-tuning recipes that meaningfully improve performance and affect model evaluation. We evaluate DNA LMs across nine compiled tasks and observe that DNA LMs achieve competitive performance relative to supervised baselines on several tasks (e.g., genome annotation), but there remains a significant gap in domains, such as variant effect and gene expression prediction. Additionally, we introduce a visualization tool to examine model performance split by various genomic properties. Lastly, we present methods for context-length extrapolation of transformer-based models that enable studying the effect of context length on DNA LM performance. The Genomics LRB is publicly available on Hugging Face.

1 Introduction

Pre-training models on a large corpus of unlabeled data and subsequently fine-tuning to solve downstream tasks has demonstrated widespread success across domains, such as natural language processing [2, 77] and computer vision [57, 65]. More recently this paradigm has shown promise in biological applications, enabled by the wealth of unlabeled data coming from next-generation sequencing technologies. A prominent example are protein language models (LMs), which have been used to predict the effects of coding mutations on protein function [47], generate viable protein sequences conditioned on functional properties [49], and accurately predict protein structure from

[*] Equal contribution.

Submitted to the 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks. Do not distribute.

amino acid sequences [48]. The development of these models has been made possible by benchmarks, such as CASP [42], TAPE [66], PEER [87], and ProteinGym [56].

Genomics represents a potential new frontier for LMs in biology. The common pre-training tasks in language modeling (i.e., filling in missing tokens based on input context) inherently train LMs to model evolutionary forces, such as conservation and co-evolution, and the statistical patterns that these models learn can map to genomic motif identification, which is useful in accurate gene annotation. Indeed, significant progress has been made, with various LMs tailored to DNA sequences [8, 9, 18, 34, 54, 55, 67, 92]. However, modeling genomic data presents unique challenges compared to proteomics. When modeling DNA, we have to account for non-coding regions and contend with interactions that can be orders of magnitude larger [26]. To guide the principled development of new DNA LMs, there is a need for robust benchmarks that accurately reflect these nuances. While several benchmarks have been proposed, these existing works contain important limitations. The vast majority of tasks proposed across existing benchmarks only consider short input contexts (less than 2k base pairs) [18, 27, 50, 92], disregarding long-range interactions that are highly impactful in genomics. Additionally, tasks in some benchmarks may be overly simplistic, failing to reflect real-world use cases, e.g., some benchmarks have used synthetic data to construct negative sets [18].

To bridge these gaps, we propose the Genomics Long-Range Benchmark (LRB), a compilation of biologically meaningful tasks in human genomics. Our benchmark deliberately incorporates tasks hypothesized to span both short and long genomic contexts. Allowing users to select arbitrary sequence length inputs for any given dataset enables us for the first time to understand empirically the importance of long-range inputs for our proposed tasks. We also include available genomic annotations and provide a visualization tool that allows users to analyze results in more detail. We demonstrate the benefit of full model fine-tuning compared to previous approaches that keep backbone DNA LM weights frozen during downstream training. Finally, we introduce methods for extending the context size of existing DNA LMs, which allows us quantify the benefits of long-range context on DNA LM performance. To summarize, we make the following contributions:

1. Release the Genomics Long-Range Benchmark, composed of biologically meaningful tasks that cover both short- and long-range genomic scales. We provide evaluation results for a selection of prominent DNA LMs in both zero-shot and fine-tuning settings along with comparisons against reference baselines. We find that on genomic annotation tasks DNA LMs perform competitively with existing supervised models, but on the long-range prediction tasks of gene expression and zero-shot mutation effect prediction there persists a large gap.

2. Develop and analyze improved fine-tuning methods that better reflect real-world usage in downstream tasks, finding that full model weight fine-tuning significantly improves performance.

3. Introduce an analysis and visualization tool to examine models' performance across different genomic properties. This tool enables deeper analyses that reveal more nuanced evidence that DNA LMs lag behind a well-regarded and long-range supervised baseline, Enformer [6], in modeling long-range interactions. The visualization tool is available here.

4. Conduct context-length extension for the Nucleotide Transformer LM to probe the impact of increasing context length on performance on our benchmark.

2 Background

2.1 Language Modeling for DNA

Supervised machine learning methods have been successfully applied to genomics [3, 6, 19, 89, 91]. However, these models depend on large amounts of labeled data and tend to be task-specific. LMs have recently gained traction in the genomics domain: the abundance of unlabeled sequences supports robust model pre-training and the widely-used pre-training objectives of next token prediction (NTP) or masked language modeling (MLM) directly lend themselves to models identifying genomic motifs and evolutionary patterns, e.g., conservation. Some notable recent works include DNABERT [34, 92], GPN [8, 9], Nucleotide Transformer (NT) [18], GENA-LM [24], HyenaDNA [54], Evo [55] and Caduceus [67]. A more thorough review of recent DNA LMs is deferred to Appendix A.2.

2.2 DNA LM Evaluation

The goal of DNA LMs is to learn meaningful representations that can be used to improve performance on downstream tasks. Existing DNA benchmarks, which include the Nucleotide Transformer tasks (NT [18]), Genomic Benchmark (GB [27]), Genome Understanding Evaluation (GUE [92]), and Benchmark for DNA LMs (BEND [50]), have been crucial for establishing baseline model capabilities. (see Appendix A.3 for a more complete description of existing works). However, these benchmarks contain several important shortcomings: they do not focus on long-range sequences, they can contain synthetic examples, and their evaluations do not take full advantage of pre-trained models.

3 The Genomics Long-Range Benchmark

Below we describe the nine tasks that we compiled from various human genome data sources that comprise our proposed Genomics Long-Range Benchmark (LRB). Our suite consists of tasks that are hypothesized to require only short-range contexts as well as those thought to need longer sequences for accurate prediction. By enabling users to download data at arbitrary length scales (the first benchmark to support this feature), these hypotheses can be rigorously tested. Our tasks span various applications that are of interest to practitioners, namely variant effect prediction, gene expression prediction, regulatory element detection, and chromatin factor identification; see Table 2. Below, for each task, we provide details on the biological relevance that motivated its inclusion, a formal task definition, and rationale for hypothesized long-range dependencies (where applicable). We defer additional details, e.g., data source and processing, train / test splits, and metric definition, to Appendix B.

Table 1: Comparison to existing benchmarks.

	Long range	**Human centric**	**Biologically meaningful**
NT [18]	✖	✖	✖
GB [27]	✖	✖	✖
GUE [92]	✖	✖	✔
BEND [50]	✖	✔	✔
Genomics LRB	✔	✔	✔

Table 2: Overview of the tasks contained in the Genomics Long-Range Benchmark.

	Type	**# Outputs**	**# Train / Test**	**% Pos. Label**
Variant Effect Prediction				
Causal eQTL	SNP Classification	1	89k / 9k	50.0
Pathogenic OMIM	SNP Classification	1	- / 2.3M	0.02
Pathogenic ClinVar	SNP Classification	1	39k / 1k	56.1
Gene Expression Prediction				
Bulk RNA-seq	Seq-wise Regression	218	23k / 1k	-
CAGE profile	Binned Regression	50 / bin	34k / 2k	-
Regulatory Element Detection				
Promoter	Seq-wise Classification	1	953k / 96k	4.7
Enhancer	Seq-wise Classification	1	1.9M / 192k	52.5
Chromatin Feature Identification				
Histone Mark Prediction	Seq-wise Classification	20	2.2M / 227k	7.0
Chromatin Accessibility	Seq-wise Classification	20	2.2M / 227k	4.4

3.1 Variant Effect Prediction

3.1.1 Causal eQTL

Biological Relevance Predicting the effects of genetic variants, particularly expression quantitative trait loci (eQTLs), is essential for understanding the molecular basis of several diseases. eQTLs are genomic loci that are associated with variations in mRNA expression levels among individuals. By linking genetic variants to causal changes in mRNA expression, researchers can uncover how certain variants contribute to disease development [17].

Task Definition The task is formulated as a binary classification problem to distinguish eQTLs from GTEx [17] from a set of matched negatives identified in Avsec et al. [6]. Inputs are sequences centered around candidate single nucleotide polymorphisms (SNPs) each assigned a causal probability by fine-mapping using the “Sum of Single Effects" (SuSiE) model [83]. Following Avsec et al. [6], variants with causal probability greater than 0.9 are labeled as positive and variants with causal probability less than 0.01 are labeled as negative.

Long-Range The regulation of gene expression is modulated by distal, cis-regulatory elements, called enhancers, that can be more than several hundred thousand base pairs (bps) away from a target gene [26]. Variants that impact gene expression are often located at such distal elements, and thus, to predict such variants, models should have long context windows [6].

3.1.2 Pathogenic OMIM

Biological relevance Predicting the effects of regulatory variants on pathogenicity is crucial for understanding disease mechanisms [51]. Elements that regulate gene expression are often located in non-coding regions, and variants in these areas can disrupt normal cellular function, leading to disease. Accurate predictions can identify biomarkers and therapeutic targets, enhancing personalized medicine and genetic risk assessment.

Task Definition The task is formulated as a binary classification problem where inputs are DNA sequences centered around a SNP and outputs are binary labels. The dataset was constructed following Benegas et al. [8], where the negative class corresponds to a common (mean allele frequency $> 5\%$) SNP in gnomAD [14] and the positive class corresponds to a pathogenic SNP, defined as a SNP in a regulatory region having an implication in a Mendelian disorder in the Online Mendelian Inheritance in Man database [72].

Long-Range Regulatory elements like enhancers and silencers can exist far from the genes they regulate [26]. Variants in these regulatory elements can lead to aberrant gene expression patterns and ultimately disease, but identifying such regulatory variants is difficult since regulatory elements can modulate the expression of proximal or distal genes. Models that can capture interactions between possibly distal regulatory elements and their target genes while still being able to capture the proximal interactions are essential to identifying non-coding pathogenic variants.

3.1.3 Pathogenic ClinVar

Biological Relevance A coding variant refers to a genetic alteration that occurs within the protein-coding regions of the genome, also known as exons. Such alterations can impact protein structure, function, stability, and interactions with other molecules, ultimately influencing cellular processes and potentially contributing to the development of genetic diseases [46]. Predicting variant pathogenicity is crucial for guiding research into disease mechanisms and personalized treatment strategies, enhancing our ability to understand and manage genetic disorders effectively.

Task Definition This task is formulated as a binary classification problem where inputs are sequences centered around SNPs. The dataset was constructed following Benegas et al. [8], where the negative class corresponds to a common (minor allele frequency $> 5\%$) SNP in gnomAD [14] and the positive class to pathogenic SNPs identified in ClinVar [44].

3.2 Gene Expression Prediction

3.2.1 Bulk RNA-seq

Biological Relevance Gene expression involves the process by which information encoded in a gene directs the synthesis of a functional gene product, typically a protein, through transcription and translation. Transcriptional regulation determines the amount of mRNA produced, which is then translated into proteins. Developing a model that can predict RNA expression levels solely from sequence data is crucial for advancing our understanding of gene regulation, elucidating disease mechanisms, and identifying functional sequence variants.

Task Definition This task is described as a multi-variable, sequence-wise regression task. Data was constructed following Zhou et al. [90] such that inputs are DNA sequences centered around the transcription start site (TSS) of each gene where the TSS was identified using a combination of annotations from GENCODE [30] and CAGE data from FANTOM5 [25]. Outputs are RPKM normalized RNA expression counts for each gene obtained from Consortium [17] that were $\log(1+x)$ normalized and standardized. For each gene, there are 218 different counts corresponding to the RNA expression level in different tissue types.

Long-Range RNA gene expression is regulated by non-coding elements, such as enhancers and silencers, which can be located hundreds of kilo-bps away from the gene [26], indicating the possible presence of long-range interactions in transcription regulation.

3.2.2 Cap Analysis Gene Expression (CAGE) Profile

Biological Relevance CAGE provides accurate high-throughput measurements of RNA expression by mapping TSSs at a nucleotide-level resolution [75]. This is vital for detailed mapping of TSSs, understanding gene regulation mechanisms, and obtaining quantitative expression data to study gene activity comprehensively.

Task Definition This task is described as a multi-variable, binned nucleotide-wise regression task. The data was constructed following the approach outlined in Basenji [37]. Inputs are DNA sequences and the outputs are $\log(1 + x)$ normalized CAGE expression counts from FANTOM5 [25] given for each 128 bp bin of the input sequence. For each bin in a sequence, there are 50 different values corresponding to expression amounts across 50 human cell / tissue types.

Long-Range The production of RNA via transcription as measured by CAGE is regulated by non-coding elements that can be located hundreds of kilo-bps away from the gene, indicating the presence of long-range interactions in transcription regulation [26].

3.3 Cis-Regulatory Element Detection

Biological Relevance Cis-regulatory elements, such as promoters and enhancers, control the spatial and temporal expression of genes [4]. These elements are essential for understanding gene regulation mechanisms and how genetic variations can lead to differences in gene expression.

Task Definition This task is described as a binary classification problem. Data from Search Candidate Regulatory Elements by ENCODE (SCREEN [79]) was processed according to our approach outlined in Appendix B.3. Inputs are sequences sampled from across the entire human genome and outputs are binary values, where a positive label is assigned to a sequence if the center 200 bps of the input sequence overlap by at least 50% with an annotated enhancer or promoter. This task is composed of two sub-tasks: (1) predicting the presence of promoters and (2) predicting the presence of enhancers.

3.4 Chromatin Feature Identification

Biological Relevance Predicting chromatin features, such as histone marks and DNA accessibility, is crucial for understanding gene regulation, as these features indicate chromatin state and are essential for transcription activation [91].

Task Definition This task is a multi-label binary classification problem constructed following Zhou & Troyanskaya [89], where sequences were sampled from the human genome as inputs and outputs correspond to binary labels for different chromatin profiles. The task contains two sub-tasks: one for predicting histone marks and another for predicting chromatin accessibility. For histone marks, each of the 20 binary values represents a different histone mark in a specific cell type. For DNA accessibility, each of the 20 binary values corresponds to a different tissue/cell type. A value is labeled as positive if the center 200 bps of the input sequence overlaps by at least 50% with a peak region measured by ChIP-seq (histone marks) or DNase-seq (DNA accessibility) obtained from ENCODE and the Roadmap Epigenomics consortium [10, 79].

3.5 Improved Evaluation with Full Fine-tuning

To evaluate DNA LMs we perform fine-tuning, i.e., we train a model in a supervised manner on a downstream task. Our fine-tuning strategy involves extracting embeddings from each model which are then input to a task-specific prediction head (see Appendix D for details). In previous benchmarks, authors fine-tuned models by freezing the embeddings [50]. We perform a systematic study of fine-tuning strategies and discover that this strategy significantly hurts DNA LM performance. We therefore provide a recipe for full-parameter fine-tuning and show that it significantly improves performance across many tasks, enabling us to evaluate models more fairly than in previous works and setting new best-practices for DNA LMs (independent of our benchmark).

3.6 Additional Novel Features of the LRB

In addition to our careful curation of tasks and improved fine-tuning methodology, we highlight two more novel aspects of the LRB.

Visualization Tool We provide benchmark users with a visualization tool in the form of an interactive `jupyter` [41] notebook. To create this tool we collected additional genomic annotation datasets from SCREEN, GENCODE, RepeatMasker [30, 73, 79] and aligned them to our benchmark task datasets; see Appendix B.5 for details and screenshots. Our tool enables a deeper level of analysis compared to what other benchmarks afford. For example, users can view models' performance in aggregate, by specific annotations, and also by distance to TSSs.

Arbitrary Sequence Length Our benchmark allows users to download arbitrary sequence lengths for any given tasks. This enables the probing of the effect of sequence length and lets users evaluate their LMs on the same context size on which they performed pre-training, mitigating any confounding from sequence length generalization effects.

3.7 Selected Baselines

To contextualize the performance of DNA LMs, we curate a set of task-specific expert methods that are comprised of well-regarded supervised models.

Combined annotation dependent depletion (CADD) [69] is a SVM developed for detecting deleterious DNA variants trained on predicted neutral variants and simulated deleterious variants. We use this method as an expert baseline for our zero-shot variant effect prediction tasks.

Enformer [6] is composed of both convolutional and transformer layers and trained in a supervised multi-task manner on various biological tasks using a context length of up to 196k bps. We use Enformer as the expert method for fine-tuning versions of variant effect prediction, gene expression prediction, and regulatory element detection tasks.

DeepSEA [91] is a convolutional network trained to predict chromatin profile data, such as transcription factor binding, histone marks, and DNA accessibility. As our chromatin feature tasks are derived from DeepSEA, we use it as the expert method for these tasks.

4 Context Length Extension

Motivated by the long-range sequences present in the LRB, we explore methods for extending the context size of existing models. To that end, we focus on the Nucleotide Transformer model (NTv2 [18]), which originally has a context size of 12k bp and uses rotary positional embeddings (RoPE [74]). However, processing longer sequences with LMs like NTv2, which use the transformer architecture [82], faces two main challenges. First, transformers rely on the attention mechanism, which scales quadratically in sequence length. Second, LMs struggle with generalizing to sequence lengths beyond those seen during pre-training, known as length extrapolation [5, 23, 36, 62].

Methodology To address the compute constraints, we use a memory-efficient attention implementation, computing attention scores sequentially and in chunks of $\sqrt{L}$, reducing memory usage from $\mathcal{O}(L^2)$ to $\mathcal{O}(\sqrt{L})$, where L denotes sequence length [64]. To solve the length generalization issue, we apply the 'NTK-aware' method presented in Peng et al. [60]. This method re-scales the frequencies in RoPE embeddings to handle longer sequences by converting length extrapolation into *interpolation*. For more details on these approaches, see Appendix C.

5 Results

5.1 Experimental Setup

We evaluate several prominent DNA LMs on our benchmark: the Nucleotide Transformer v2 (NTv2) series [18], DNABERT-1 [34] and 2 [92], and the HyenaDNA series [54], representing a range of pre-training datasets and objectives, architectures, and model sizes. For fine-tuning, we use an MLP as the prediction head and train both the DNA LM and MLP weights (see Appendix D for full details).

For classification tasks with highly imbalanced labels (see Table 2), we use area under precision-recall curve (AUPRC) as opposed to receiver operator curve (AUROC) as the metric.

Fine-tuning Models are trained using either mean-squared error loss for regression tasks or cross-entropy loss for classification tasks. For each task, we perform five-fold cross-validation (CV) using different random seeds, where we create different train / validation splits, select the best-performing model using early stopping on validation loss, and evaluate it on the held-out test set. We report the mean ± standard deviation performance across folds as final metrics.

Zero-Shot Prediction We also evaluate the zero-shot performance on our three variant effect prediction tasks to account for the fact that, in practice, determining pathogenicity or causality of variants is difficult, which often results in smaller datasets not suitable for fine-tuning. Given the extreme class imbalance in the Pathogenic OMIM dataset, we only perform zero-shot evaluation for this task and do not report fine-tuning results.

5.2 Main DNA LM Results

In Tables 3 and 4, we present the top performing DNA LMs (full results in Appendix E).

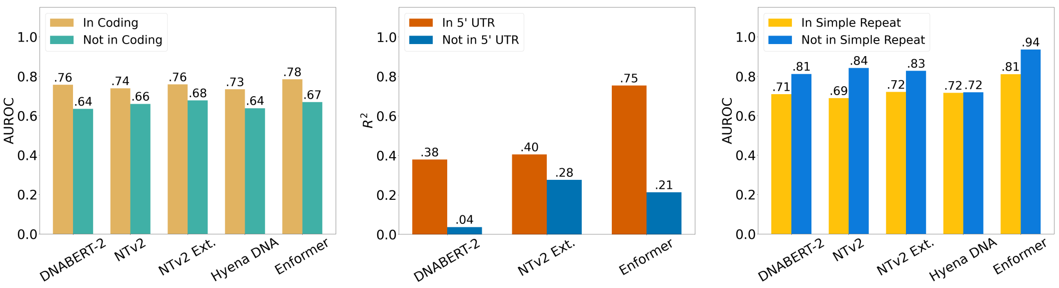
Table 3: Benchmarking performance of DNA LMs and baselines on variant effect prediction tasks. Models were evaluated using both fine-tuning and zero-shot. Best LM values are **bolded** and in **green** if LM beats baseline. *Extended NTv2 was fine-tuned with 60k bp sequences due to compute constraints.

Model Name	**Context** *(bps)*	**Causal eQTL** *(AUROC)*		**Pathogenic ClinVar** *(AUROC)*		**Pathogenic OMIM** *(AUPRC)*
		Fine-tune	*Zero-shot*	*Fine-tune*	*Zero-shot*	*Zero-shot*
DNABERT-2	10k	0.72 ± 0.008	0.50	0.74 ± 0.013	0.50	0.002
NTv2 500M	12k	0.72 ± 0.003	**0.51**	**0.78** ± 0.009	**0.68**	0.003
Extended	96k*	0.74 ± 0.004	**0.51**	0.75 ± 0.018	0.53	0.002
HyenaDNA	32k	0.72 ± 0.002	**0.52**	0.66 ± 0.012	0.50	0.002
Baseline		0.76 ± 0.002 (Enformer)	0.56 (CADD)	0.65 ± 0.031 (Enformer)	0.97 (CADD)	0.253 (CADD)

Variant Effect Prediction For zero-shot evaluation, we observe that DNA LMs are outperformed by the CADD baseline on all variant effect prediction tasks. Additionally, for zero-shot Causal eQTL, we find that all models struggle, with near-random performance. Predicting pathogenicity, is the

Table 4: Benchmarking performance of DNA LMs and baselines on gene expression, regulatory element, and chromatin features tasks. Models were evaluated in only a fine-tuned setting for this set of tasks. Best LM values are **bolded** and in **green** if LM beats baseline.

	Context *(bps)*	**Bulk RNA** *(R^2)*	**CAGE** *(R^2)*	**Promoter** *(AUPRC)*	**Enhancer** *(AUROC)*	**Histone Marks** *(AUPRC)*	**DNA Accessibility** *(AUPRC)*
		Fine-tune	*Fine-tune*	*Fine-tune*	*Fine-tune*	*Fine-tune*	*Fine-tune*
DNABERT-2	10k	0.51 ± 0.050	-	0.71 ± 0.112	0.81 ± 0.022	0.24 ± 0.091	0.15 ± 0.064
NTv2 500M	12k	**0.57** ± 0.016	**0.39** ± 0.011	**0.79** ± 0.006	**0.82** ± 0.002	0.38 ± 0.003	**0.3** ± 0.007
Extended	96k	0.56 ± 0.037	0.36 ± 0.011	0.78 ± 0.003	**0.82** ± 0.005	**0.38** ± 0.004	**0.3** ± 0.006
HyenaDNA	32k	0.47 ± 0.010	0.22 ± 0.007	0.72 ± 0.007	**0.82** ± 0.002	0.22 ± 0.003	0.084 ± 0.001
Baseline		0.80 ± 0.010 (Enformer)	0.49 ± 0.000 (Enformer)	0.86 ± 0.006 (Enformer)	0.92 ± 0.002 (Enformer)	0.35 (DeepSea)	0.44 (DeepSea)



(a) Fine-tuned Causal eQTL prediction; by protein coding annotation.

(b) Bulk RNA prediction; by 5′ UTR annotation

(c) Enhancer Annotation; by simple repeat annotation.

Figure 1: Results split by genomic annotations.

clearest example where DNA LMs fall short of CADD, which has nearly 2x better performance in ClinVar and about 100x in OMIM. When fine-tuning, we find that DNA LM performance on both variant tasks greatly improves, matching or surpassing the strong Enformer baseline.

Gene Expression Prediction While NTv2 is the best performing DNA LM for Bulk RNA and CAGE tasks, the baseline Enformer outperforms LMs by a wide margin.

Regulatory Element Detection DNA LMs are able to accurately predict the presence of regulatory elements, especially considering the class-imbalance present in promoter detection, with NTv2 performing best among DNA LMs. However, there remains a gap to the supervised Enformer model.

Chromatin Feature Identification For both histone mark and DNA accessibility, NTv2 is the best performing DNA LM, even exceeding the supervised baseline on the former task, and demonstrating significantly better performance than the other DNA LMs.

5.3 Analyzing Results by Genomic Annotations

We developed an analysis and visualization tool to examine models performance across different genomic properties and annotations. Using our tool we are able to perform deeper analyses and extract insights about the performance of each model, which are inaccessible to users of existing benchmarks. We detail some examples in Figure 1.

Causal eQTL Prediction (Fine-tune) By stratifying SNPs into protein-coding and non-coding regions in Figure 1a, we find a potential failure mode for both DNA LMs and supervised models. Non-coding variants presumably entail regulatory and possibly longer-range interactions, and all models perform worse in these regions.

Bulk RNA Expression Prediction In Figure 1b, we see that the performance of DNA LMs and Enformer drops precipitously when focusing on non-5′ regions that likely entail longer-range interactions. However, we also observe that the context-extended NTv2 outperforms Enformer on this region, implying that the majority of the performance gap between DNA LMs and the Enformer baseline lies in modeling variants in the $5'$ regions.

Enhancer Detection In Figure 1c, we observe that most models, including Enformer, suffer a performance hit when identifying enhancers within simple repeat regions, likely due to the difficulty of detecting enhancers within repetitive regions of the genome.

5.4 Length Extension

To create the context extended model, we conduct additional training (∼5B tokens) on the pre-training dataset using the methodology described in Section 4 (and in Appendix D.5). For certain long-range tasks, the additional context extension pre-training improves performance. For example, for Causal eQTL prediction (with fine-tuning) in Figure 2 we see that the context extended NTv2 has the best DNA LM performance and that this trend is more pronounced when stratifying by SNP distance to TSS.

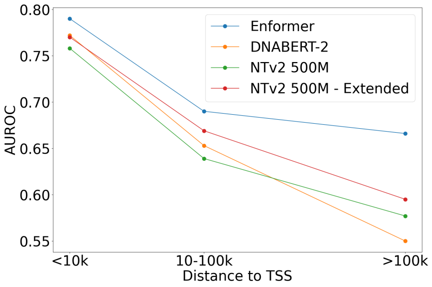


Figure 2: Fine-tuned Causal eQTL variant task; by distance to nearest TSS.

5.5 Effect of Fine-tuning Methodology

In Table 5, we demonstrate the importance of our proposed fine-tuning. For two of the DNA LMs (see additional results in Appendix E.2), we show how full fine-tuning, as opposed to freezing LM weights and only training a prediction head, a common practice in existing benchmarks such as BEND [50], drastically improves model performance almost uniformly across tasks. We also believe our methodology is more in line with how practitioners would use DNA LMs in real-world settings.

Table 5: Difference in performance of DNA LM fine-tuning strategies. Percent increase in performance of full fine-tuning vs. freezing LM weights and only training prediction heads.

	Causal eQTL *(AUCROC)*	**Pathogenic ClinVar** *(AUROC)*	**Bulk RNA** (R^2)	**CAGE** (R^2)	**Promoter** *(AUPRC)*	**Enhancer** *(AUROC)*	**Histone Marks** *(AUCPRC)*	**DNA Accessibility** *(AUPRC)*
NTv2 500M	+0.49	+4.27	+18.29	+42.14	-1.45	+0.90	+22.46	+47.96
HyenaDNA	+0.35	+11.58	+107.48	+102.91	+5.09	+5.39	+14.43	-22.67

6 Discussion and Conclusion

In this work, we introduced the Genomics LRB. Our benchmark is the first to truly evaluate long-range capabilities. We provided initial results for several prominent DNA LMs, with more in-depth analysis than previous benchmarks explored. Our results demonstrate the importance of fully fine-tuning models. Additionally, we identify several domains where a large performance gap needs to be bridged before DNA LMs can be reliably used and some failure modes of DNA LMs. Namely, zero-shot DNA LM variant effect prediction is not yet mature enough to replace widely-used tools, such as CADD. Similarly, for gene expression prediction, DNA LMs lag far behind supervised methods. In contrast, for annotation tasks, DNA LMs already demonstrate competitive performance relative to proven methods. These results demonstrate that future DNA LM efforts should focus on the more difficult tasks that entail long-range interactions, and we hope that our benchmark spurs such development.

Future Work One potential limitation of our work is the lack of hyperparameter search for fine-tuning; a more extensive search would better differentiate models. Another limitation is the lack of experimentally verified enhancer-gene pairings, which would allow for a more complete examination of the long-range capabilities of models. In future iterations of our benchmark, we also plan to add more tissue-specific analyses, bp-level annotation tasks, and tasks covering multiple species.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[4] Robin Andersson and Albin Sandelin. Determinants of enhancer and promoter activities of regulatory elements. *Nature Reviews Genetics*, 21:71–87, 2020.

[5] Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35: 38546–38556, 2022.

[6] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[7] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, 53(3):354–366, 2021.

[8] 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. *bioRxiv*, 2023.

[9] Gonzalo Benegas, Sanjit Singh Batra, and Yun S Song. Dna language models are powerful predictors of genome-wide variant effects. *Proceedings of the National Academy of Sciences*, 120(44):e2311219120, 2023.

[10] Bradley E Bernstein, John A Stamatoyannopoulos, Joseph F Costello, Bing Ren, Aleksandar Milosavljevic, Alexander Meissner, Manolis Kellis, Marco A Marra, Arthur L Beaudet, Joseph R Ecker, et al. The nih roadmap epigenomics mapping consortium. *Nature biotechnology*, 28(10): 1045–1048, 2010.

[11] bloc97. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation, 2023. URL https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have.

[12] James Bradbury, Roy Frostig, Peter Hawkins, Matthew Johnson James, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

[13] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

[14] Siwei Chen, Laurent C. Francioli, Julia K. Goodrich, Ryan L. Collins, Masahiro Kanai, Qingbo Wang, Jessica Alföldi, Nicholas A. Watts, Christopher Vittal, Laura D. Gauthier, Timothy Poterba, Michael W. Wilson, Yekaterina Tarasova, William Phu, Mary T. Yohannes, Zan Koenig, Yossi Farjoun, Eric Banks, Stacey Donnelly, Stacey Gabriel, Namrata Gupta, Steven Ferriera, Charlotte Tolonen, Sam Novod, Louis Bergelson, David Roazen, Valentin Ruano-Rubio, Miguel Covarrubias, Christopher Llanwarne, Nikelle Petrillo, Gordon Wade, Thibault Jeandet, Ruchi Munshi, Kathleen Tibbetts, gnomAD Project Consortium, Anne O'Donnell-Luria, Matthew Solomonson, Cotton Seed, Alicia R. Martin, Michael E. Talkowski, Heidi L. Rehm, Mark J. Daly, Grace Tiao, Benjamin M. Neale, Daniel G. MacArthur, and Konrad J. Karczewski. A genome-wide mutational constraint map quantified from variation in 76,156 human genomes. *bioRxiv*, 2022. doi: 10.1101/2022.03.20.485034. URL https://www.biorxiv.org/content/early/2022/03/21/2022.03.20.485034.

[15] Deanna M Church, Valerie A Schneider, Tina Graves, Katherine Auger, Fiona Cunningham, Nathan Bouk, Hsiu-Chuan Chen, Richa Agarwala, William M McLaren, Graham RS Ritchie, et al. Modernizing reference genome assemblies. *PLoS biology*, 9(7):e1001091, 2011.

[16] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.

[17] The GTEx Consortium. The gtex consortium atlas of genetic regulatory effects across human tissues. *Science*, 369(6509):1318–1330, 2020.

[18] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pp. 2023–01, 2023.

[19] Bernardo P de Almeida, Franziska Reiter, Michaela Pagani, and Alexander Stark. Deepstarr predicts enhancer activity from dna sequence and enables the de novo design of synthetic enhancers. *Nature genetics*, 54(5):613–624, 2022.

[20] Bernardo P de Almeida, Hugo Dalla-Torre, Guillaume Richard, Christopher Blum, Lorenz Hexemer, Maxence Gélard, Javier Mendoza-Revilla, Priyanka Pandey, Stefan Laurent, Marie Lopez, et al. Segmentnt: annotating the genome at single-nucleotide resolution with dna foundation models. *bioRxiv*, pp. 2024–03, 2024.

[21] DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL http://github.com/google-deepmind.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[23] Yann Dubois, Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. Location attention for extrapolation to longer sequences. *arXiv preprint arXiv:1911.03872*, 2019.

[24] 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. *bioRxiv*, 2023.

[25] Alistair RR Forrest, Hideya Kawaji, Michael Rehli, JK Baillie, MJL De Hoon, T Lassmann, M Itoh, KM Summers, H Suzuki, CO Daub, et al. a, carninci p, hayashizaki y. a promoter-level mammalian expression atlas. *Nature*, 507:462–70, 2014.

[26] Eileen E. M. Furlong and Michael Levine. Developmental enhancers and chromosome topology. *Science*, 361(6409):1341–1345, 2018. doi: 10.1126/science.aau0320. URL https://www.science.org/doi/abs/10.1126/science.aau0320.

[27] 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.

[28] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[29] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

[30] Jennifer Harrow, Adam Frankish, Jose M Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, et al. Gencode: the reference human genome annotation for the encode project. *Genome research*, 22(9):1760–1774, 2012.

[31] Tom Hennigan, Trevor Cai, Tamara Norman, Lena Martens, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020. URL http://github.com/deepmind/dm-haiku.

[32] Angela S Hinrichs, Donna Karolchik, Robert Baertsch, Galt P Barber, Gill Bejerano, Hiram Clawson, Mark Diekhans, Terrence S Furey, Rachel A Harte, Fan Hsu, et al. The ucsc genome browser database: update 2006. *Nucleic acids research*, 34(suppl_1):D590–D598, 2006.

[33] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9 (3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[34] 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.

[35] kaiokendev. Things i'm learning while training superhot, 2023. URL https://kaiokendev.github.io/til#extending-context-to-8k.

[36] Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *arXiv preprint arXiv:2305.19466*, 2023.

[37] David R Kelley. Cross-species regulatory sequence activity prediction. *PLoS computational biology*, 16(7):e1008050, 2020.

[38] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[39] Daehwan Kim, Joseph M Paggi, Chanhee Park, Christopher Bennett, and Steven L Salzberg. Graph-based genome alignment and genotyping with hisat2 and hisat-genotype. *Nature biotechnology*, 37(8):907–915, 2019.

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[41] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt (eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87 – 90. IOS Press, 2016.

[42] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (casp)—round xiv. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1607–1617, 2021.

[43] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[44] Melissa J Landrum, Shanmuga Chitipiralla, Garth R Brown, Chao Chen, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, Kuljeet Kaur, Chunlei Liu, Vitaly Lyoshin, Zenith Maddipatla, Rama Maiti, Joseph Mitchell, Nuala O'Leary, George R Riley, Wenyao Shi, George Zhou, Valerie Schneider, Donna Maglott, J Bradley Holmes, , and Brandi L Kattman. Clinvar: improvements to accessing data. *Nucleic Acids Res*, 8:D835–D844, 2020.

[45] Tong Ihn Lee and Richard A. Young. Transcriptional regulation and its misregulation in disease. *Cell*, 152(6):P1237–1251, 2013.

[46] Monkol Lek, Konrad J. Karczewski, Eric V. Minikel, Kaitlin E. Samocha, Eric Banks, Timothy Fennell, Anne H. O'Donnell-Luria, James S. Ware, Andrew J. Hill, Beryl B. Cummings, Taru Tukiainen, Daniel P. Birnbaum, Jack A. Kosmicki, Laramie E. Duncan, Karol Estrada, Fengmei Zhao, James Zou, Emma Pierce-Hoffman, Joanne Berghout, David N. Cooper, Nicole Deflaux, Mark DePristo, Ron Do, Jason Flannick, and Exome Aggregation Consortium. Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, 536:285–291, 2016.

[47] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022: 500902, 2022.

[48] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[49] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8): 1099–1106, 2023.

[50] Frederikke Isa Marin, Felix Teufel, Marc Horlacher, Dennis Madsen, Dennis Pultz, Ole Winther, and Wouter Boomsma. Bend: Benchmarking dna language models on biologically meaningful tasks. In *The Twelfth International Conference on Learning Representations*, 2023.

[51] Shruti Marwaha, Joshua W. Knowles, and Euan A. Ashley. A guide for the diagnosis of rare and undiagnosed disease: beyond the exome. *Genome Medicine*, 14:23, 2022.

[52] Wouter Meuleman, Alexander Muratov, Eric Rynes, Jessica Halow, Kristen Lee, Daniel Bates, Morgan Diegel, Douglas Dunn, Fidencio Neri, Athanasios Teodosiadis, et al. Index and biological spectrum of human dnase i hypersensitive sites. *Nature*, 584(7820):244–251, 2020.

[53] Jill E Moore, Michael J Purcaro, Henry E Pratt, Charles B Epstein, Noam Shoresh, Jessika Adrian, Trupti Kawli, Carrie A Davis, Alexander Dobin, et al. Expanded encyclopaedias of dna elements in the human and mouse genomes. *Nature*, 583(7818):699–710, 2020.

[54] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *arXiv preprint arXiv:2306.15794*, 2023.

[55] 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 Ré, Patrick D. Hsu, and Brian L. Hie. Sequence modeling and design from molecular to genome scale with evo. *bioRxiv*, 2024. doi: 10.1101/2024.02.27.582234. URL https://www.biorxiv.org/content/early/2024/02/27/2024.02.27.582234.

[56] Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: large-scale benchmarks for protein fitness prediction and design. *Advances in Neural Information Processing Systems*, 36, 2024.

[57] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[60] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

[61] 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. *arXiv preprint arXiv:2302.10866*, 2023.

[62] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[63] Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

[64] Markus N Rabe and Charles Staats. Self-attention does not need $O(n^2)$ memory. *arXiv preprint arXiv:2112.05682*, 2021.

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

[66] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

[67] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling. *arXiv preprint arXiv:2403.03234*, 2024.

[68] Valerie A Schneider, Tina Graves-Lindsay, Kerstin Howe, Nathan Bouk, Hsiu-Chuan Chen, Paul A Kitts, Terence D Murphy, Kim D Pruitt, Françoise Thibaud-Nissen, Derek Albracht, et al. Evaluation of grch38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome research*, 27(5):849–864, 2017.

[69] Max Schubach, Thorben Maass, Lusiné Nazaretyan, Sebastian Röner, and Martin Kircher. Cadd v1. 7: using protein language models, regulatory cnns and other nucleotide-level scores to improve genome-wide variant predictions. *Nucleic Acids Research*, 52(D1):D1143–D1154, 2024.

[70] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[71] Matthew D Shirley, Zhaorong Ma, Brent S Pedersen, and Sarah J Wheelan. Efficient" pythonic" access to fasta files using pyfaidx. Technical report, PeerJ PrePrints, 2015.

[72] Damian Smedley, Max Schubach, Julius OB Jacobsen, Sebastian Köhler, Tomasz Zemojtel, Malte Spielmann, Marten Jäger, Harry Hochheiser, Nicole L Washington, Julie A McMurry, et al. A whole-genome analysis framework for effective identification of pathogenic regulatory variants in mendelian disease. *The American Journal of Human Genetics*, 99(3):595–606, 2016.

[73] AFA Smit, R Hubley, and P Green. Repeatmasker open-4.0. 2013–2015, 2015.

[74] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

[75] Hazuki Takahashi, Timo Lassmann, Mitsuyoshi Murata, and Piero Carninci. 5XXXXXX end–centered expression profiling using cap-analysis gene expression and next-generation sequencing. *Nature Protocols*, 7:542–561, 2012.

[76] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[77] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[78] The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL https://github.com/pandas-dev/pandas.

[79] The ENCODE Project Consortium. Expanded encyclopaedias of dna elements in the human and mouse genomes. *Nature*, 583(7818):699–710, 2020.

[80] 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.

[81] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

[83] Gao Wang, Abhishek Sarkar, Peter Carbonetto, and Matthew Stephens. A simple new approach to variable selection in regression, with application to genetic fine mapping. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(5):1273–1300, 2020.

[84] Qingbo S. Wang, David R. Kelley, Jacob Ulirsch, Masahiro Kanai, Shuvom Sadhuka, Ran Cui, Carlos Albors, Nathan Cheng, Yukinori Okada, The Biobank Japan Project, Francois Aguet, Kristin G. Ardlie, Daniel G. MacArthur, and Hilary K. Finucane. Leveraging supervised learning for functionally informed fine-mapping of cis-eqtls identifies an additional 20,913 putative causal eqtls. *Nature Communications*, 12:3394, 2021.

[85] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

[86] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[87] Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Ma Chang, Runcheng Liu, and Jian Tang. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35:35156–35173, 2022.

[88] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[89] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[90] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8):1171–1179, 2018.

[91] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8):1171–1179, 2018.

[92] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.

A Extended Background

A.1 Terminology

The genome is a sequence of four nucleotides (*Adenine*, *Cytosine*, *Thymine*, and *Guanine*) organized into a double-stranded helical structure called *deoxyribonucleic acid* (DNA). This structure encodes the information required for the development, maintenance, and function of cells. Genetic information flows from DNA to *messenger ribonucleic acid* (mRNA) by a process called *transcription*, and mRNA is used as a blueprint to create *proteins* via a process called *translation*. Proteins are responsible for initiating and sustaining the cellular processes, while DNA encodes the information necessary for their production.

The genome is organized into functional elements, including *coding* and *non-coding* regions. Coding regions comprise genes responsible for protein synthesis, while non-coding regions can play vital regulatory roles. *Promoters*, a type of regulatory region, are situated close to genes and serve as sites for transcription initiation. *Enhancers*, another regulatory element located farther from genes, modulate gene expression by recruiting transcription factors, a type of protein that regulates transcription. Notably, a single gene can be regulated by multiple promoters and enhancers simultaneously.

DNA does not exist solely as a linear molecule but is instead tightly packaged around *histone* proteins, forming a sphere of wound DNA called *nucleosomes*. These nucleosomes further assemble into *chromatin*, which constitutes the 23 pairs of *chromosomes* in humans. Chromatin can exist in an open (*euchromatin*) or closed (*heterochromatin*) state, influencing the ability of the underlying DNA to be transcribed. Chemical modifications to histones play significant roles in chromatin remodeling acting as signals that recruit proteins to either condense the chromatin structure (making it less accessible) or relax it (making it more accessible), thereby influencing gene activity.

Mutations in the genome, including *single nucleotide polymorphisms* (SNPs), insertions, and deletions, can alter DNA sequences, potentially disrupting functional genomic elements or affecting the structure and function of proteins. Understanding the impact of these sequence variations on disease remains a central challenge in biology. Such mutations can lead to genetic disorders or contribute to the development of complex diseases.

A.2 Recent DNA LMs

DNABERT Arguably the first DNA LM, DNABERT proposed in Ji et al. [34] applies the BERT architecture from Devlin et al. [22], with a few modifications, to genomic sequences. The authors train on the human genome and use k-mer tokens generated with sliding windows. Input sequences were 512 tokens, and the model was trained using the MLM objective, but with the restriction that masking was performed for contiguous tokens within a sequence. The downstream tasks focused on genome annotation, with promoter, transcription factor binding sites, and splice site classification. Of note, although DNABERT was pre-trained on human genome, it was fine-tuned on mouse downstream tasks as well, yielding competitive performance relative to supervised learning baselines.

Nucleotide Transformer Following the success of model scaling in other domains, Dalla-Torre et al. [18] explore scaling DNA foundation models in introducing the Nucleotide Transformer. They explore various model sizes – ranging from 500 million parameters to 2.5 billion, in their first generation release, and 50 million to 500 million parameters in their subsequent version 2 – and various pre-training data setups, including human reference genome, 3,000 diverse human genomes, and 850 multi-species reference genomes. They utilize non-overlapping 6-mer tokenization and a BERT-style architecture trained with an MLM objective. Other notable differences between the first and second version is that in version 2 input context size was scaled from 1,000 tokens to 2,000 and positional embeddings used in version 1 were learned whereas version 2 used rotary embeddings [74], which have been shown to better extend to longer contexts. This work also introduced the Nucleotide Transformer suite of tasks, described in more detail below.

DNABERT-2 Building on the initial success of DNABERT, Zhou et al. [92] present a model trained on multi-species genomes: 135 species, across 7 categories. They also change tokenization to byte-pair-encoding [43, 70], with a vocabulary size of 4,096, arguing that overlapping k-mer tokenization makes the MLM task 'too easy' by leaking information across tokens and that non-overlapping k-mer tokenization suffers from the drawback that minor changes to the input sequence, e.g., removing the first character, lead to drastically different tokenization outputs. They use input sequence lengths of 128 tokens. Additionally, Zhou et al. [92] replace the learned positional embeddings from DNABERT with ALiBi [63]. DNABERT-2 was evaluated on a suite of downstream tasks introduced in Zhou et al. [92] known as the Genome Understanding Evaluation (GUE).

HyenaDNA In contrast to the other language models reviewed above, the HyenaDNA model from Nguyen et al. [54] is a next token prediction, uni-directional model. Using character-level tokenization and the Hyena layers [61] as a backbone, Nguyen et al. [54] also propose a training recipe for scaling input context sizes up to 1 million bps. To evaluate their model they use a combination of downstream tasks, including the suite of tasks from Nucleotide Transformer [18], a set of mouse and human genome annotation tasks presented in Grešová et al. [27], the chromatin profiling tasks from DeepSea [89], and a species classification task, where the model takes in sequences of various species and needs to output the correct species label.

Other DNA LMs While the models above represent those that we initially validate on our benchmark, the field of DNA LMs is growing at a rapid pace and consists of several notable works that we briefly describe below.

While not developed specifically as a DNA LM, the **BigBird** architecture proposed in Zaheer et al. [88] was applied to genomic sequences to demonstrate its usefulness in long context tasks. Using sparse attention to reduce computational complexity of transformer [82] blocks from quadratic to linear, BigBird is able to effectively scale up to longer contexts. In Fishman et al. [24], the authors present a family of foundation models, **GENA-LM**, aimed specifically at modeling longer DNA sequences. Pre-training with an MLM objective on human and multi-species genomes, they use BPE with a vocabulary size of 32,000. The backbone architectures are either BERT [22] or BigBird [88], allowing them to extend input lengths up to 36k bps.

Focusing on plant genomes, Benegas et al. [9] pre-train a MLM model on unaligned reference genomes of the *Arabidopsis thaliana* species and seven related species within the Brassicales order. Using character-level tokenization they use input lengths of 512 bps with dilated convolutions to create their **GPN** model. With 25 layers, despite the relatively short training sequences, GPN can theoretically extend to sequence inputs of millions of bps.

In the recent **Mamba** work [28], the authors pre-train various sized models that use the Mamba backbone on the human reference genome. Similar to HyenaDNA , the pre-training objective is next token prediction, tokenization is by nucleotide base, and input sequences are scaled up to 1 million bps. Building off this work, Schiff et al. [67] introduced **Caduceus**, a bi-directional Mamba-based model that contains reverse complement equivariance inductive biases, demonstrating state-of-the art performance on several tasks, including several Nucleotide Transformer tasks [18] and the Genomic Benchmark [27].

A.3 Existing DNA Language Model Benchmarks

Existing benchmarks vary in several aspects, including the species considered, the specific tasks of interest, the framing of these tasks, and the evaluation methodologies employed. These proposed benchmarks include the Nucleotide Transformer Benchmark [18], Genomic Benchmarks [27], Genome Understanding Evaluation (GUE, [92]), and Benchmarking DNA Language Models on Biologically Meaningful Tasks (BEND; Marin et al. [50]).

Existing DNA benchmarks are primarily composed of classification tasks for sequence-wise predictions, ranging from cis-regulatory elements and splice sites to chromatin features and variant effects.

These benchmarks not only compile and build datasets but also carry out evaluations of DNA LMs using both fine-tuning methods, where pre-trained models are trained in a supervised manner on the downstream tasks, and zero-shot prediction, where models are evaluated in their pre-trained state without additional fine-tuning.

Nucleotide Transformer Benchmark Dalla-Torre et al. [18] compile a set of 18 distinct genomic datasets framed as sequence-wise classification tasks. These tasks included 10 datasets related to epigenetic mark prediction in yeast genomes, three tasks predicting the presence of promoters in mouse and human genomes, two tasks predicting enhancer presence and activity levels in the human genome, and three tasks predicting splice sites in multiple diverse species. Sequence lengths in this benchmark ranged from 200 to 600 bps. Additionally, the authors evaluated a set of DNA LMs and a supervised genomic model, Enformer Avsec et al. [6], by fine-tuning these models on their benchmark using a robust 10-fold cross-validation protocol. Parameter-efficient fine-tuning methods with a classification head were used for Enformer, DNABERT, and NT models, while full fine-tuning with a classification head was applied to the HyenaDNA models. Limitations of this benchmark include the focus on short-range contexts, the inclusion of synthetic sequences as negative examples, and limited supervised baselines.

Genomic Benchmarks Genomic Benchmarks [27] is a collection of datasets for genomic sequence classification, composed of existing datasets and novel ones scraped from publicly available databases. The benchmark includes nine tasks focusing on regulatory element prediction, such as promoters, enhancers, and open chromatin regions. These tasks cover human, mouse, roundworm, and fly genomes, with average sequence lengths ranging from 200 to 2,370 bps. The authors also provide code to train simple convolutional network that can be used as a baseline. Similar to the Nucleotide Transformer benchmark, this benchmark focuses on short-range tasks, does not present a robust set of baselines, and contains potentially less impactful tasks, e.g., distinguishing between human and worm genomic sequences.

Genomic Understanding Evaluation (GUE) The authors of the DNABERT-2 [92] introduced the Genomic Understanding Evaluation (GUE) benchmark, which is divided into two groups by sequence length: GUE and GUE+. This benchmark comprises seven classification tasks, such as cis-regulatory element prediction and species classification, built from 28 datasets from multiple species. The inclusion of multiple species allows for the assessment of DNA LMs' generalizability. The tasks are curated to be appropriately challenging, including measures such as class balancing, adversarial sample inclusion, and reduction of training sample volume. GUE features sequence lengths ranging from 70 to 1k bps, while GUE+ includes sequence lengths from 5k to 10k bps. GUE evaluated DNABERT1 and 2, NT, and HyenaDNA models on their benchmark. HyenaDNA models are fully fine-tuned while DNABERT and NT models are fine-tuned using parameter efficient methods. The GUE benchmark results are limited since they do not cover a robust set of baselines but rather only present the simple supervised convolutional network from the Genomic Benchamark [27]. Additionally, only binary or multi-class sequence-wise classification tasks are considered and tasks of biological importance, such as variant effect prediction and gene expression are not included.

Benchmarking DNA LLMs on Biologically Meaningful Tasks (BEND) BEND [50] is a recently proposed benchmark focused on compiling tasks that capture the complexity and intricacies of real-world genomic analysis. The authors collected seven different datasets, all from the human genome, covering gene finding, enhancer annotation, chromatin accessibility, histone modification, CpG methylation, and two types of variant effect prediction. Unlike previous benchmarks that focused solely on sequence-wise classification tasks, BEND also includes the task “Gene finding", which tests nucleotide-resolution modeling. In five out of seven tasks the input length is 512 bps, as these tasks are considered short-range. “Gene finding" task use sequences up to 14k bps. Their “Enhancer annotation" task uses 100k bp sequences, but it only contains 285 input sequences. Notably, for tasks in BEND that overlap with our benchmark (such as variant effect prediction), BEND uses a fixed context length of 512 bp, thus not evaluating the importance of extended context and variant-gene distal interactions on this type of task. Therefore, this benchmark is mostly limited to short-range tasks and does not include gene expression, an important and challenging task in genomics. This benchmark however makes progress in including a broader set of supervised methods as baselines. Unlike our work, models are only evaluated using partial fine-tuning, where backbone DNA LM weights are frozen for downstream task training.

B Additional Details about Genomic Long Range Benchmark

We note that our datasets do not contain any personally identifiable information or offensive content.

Table 6 provides details describing the evaluation method used, dataset sizes, metric, and data sources. Additional details on task specific data curation and processing are described in the following subsections.

Table 6: Additional information for Genomic LRB tasks, including number of samples in train and test splits, metric, and data source.

Task	Eval	Test split	Metric	Data Source
Variant Effect Prediction				
Causal eQTL	Fine-tune & Zero-shot	Chromosome 9, 10	AUROC	GTEx (via [7])
Pathogenic OMIM	Zero-shot	-	AUPRC	OMIM, gnomAD (via [8])
Pathogenic ClinVAR	Fine-tune & Zero-shot	Chromosome 8	AUROC	ClinVar, gnomAD (via [8])
Gene Expression Prediction				
Bulk RNA Expression	Fine-tune	Chromosome 8	R^2	GTEx, FANTOM5 (via [90])
CAGE	Fine-tune	Random	R^2	FANTOM5 (via [37])
Regulatory Element Detection				
Promoter	Fine-tune	Chromosome 8, 9	AUPRC	SCREEN
Enhancer	Fine-tune	Chromosome 8,9	AUROC	SCREEN
Chromatin Feature Identification				
Histone Marks	Fine-tune	Chromosome 8, 9	AUPRC	ENCODE, Roadmap Epigenomics (via [89])
DNA Accessibility	Fine-tune	Chromosome 8, 9	AUPRC	ENCODE, Roadmap Epigenomics (via [89])

B.1 Variant Effect Prediction

B.1.1 Causal eQTL

Data Processing Processed data in the form of `vcf` files for positive and negative variants across 49 different tissue types were obtained from Avsec et al. [6]. Fine-mapped GTEx [17] eQTLs originate from Wang et al. [84], while the negative matched set of variants comes from Avsec et al. [6]. The statistical fine-mapping tool SuSiE [83] was used to label variants. Variants from the fine-mapped eQTL set were selected and given positive labels if their posterior inclusion probability was > 0.9, as assigned by SuSiE. Variants from the matched negative set were given negative labels if their posterior inclusion probability was < 0.01. DNA sequences were obtained from the human reference genome assembly GRCh38 [68].

B.1.2 Pathogenic OMIM

Data Processing Processed data was obtained from Benegas et al. [8] in the form of `parquet` files with columns for SNP location, reference and alternative alleles, and pathogenicity label. Positive labeled data originates from a curated set of pathogenic variants located in the Online Mendelian Inheritance in Man (OMIM) [72] catalog. The negative set is comprised of variants that are defined as common from gnomAD [14]. gnomAD version `3.1.2` was downloaded and filtered to variants with allele number of at least 25,000. Common variants were defined as those with minor allele frequency (MAF) $> 5\%$. The input sequences were constructed by selecting the appropriate genomic region from the human reference genome assembly GRCh38 [68] and applying the changes specified by the given variants.

B.1.3 Pathogenic ClinVar

Data Processing Processed data was obtained from Benegas et al. [9] in the form of `parquet` files with columns for SNP location, reference and alternative alleles, and pathogenicity label. Positive labels correspond to pathogenic variants originating from ClinVar [44] whose review status was described as having at least a single submitted record with a classification but without assertion criteria. The negative set are variants that are defined as common from gnomAD [14]. gnomAD version `3.1.2` was downloaded and filtered to variants with allele number of at least 25,000. Common variants were defined as those with MAF $> 5\%$. Sequences were obtained from the human reference genome assembly GRCh38 [68].

Short-Range The ClinVar dataset is mostly variants in coding regions, and since most human protein sequences have less than 1,000 amino acids predicting the impact of coding variants should require orders of magnitude smaller context windows than non-coding variants. Therefore, we consider this task as potentially short-range.

B.2 Gene Expression Prediction

B.2.1 Bulk RNA-seq

Data Processing Processed data in the form `csv` files that contained gene TSS locations, strand, and RNA expression RPKM counts across 218 tissue types was obtained from ExPecto [90]. Expression data originates from GTEx [17], while representative TSS locations were determined in ExPecto. The authors of ExPecto determined representative TSS for Pol II transcribed genes based on quantification of CAGE reads from the FANTOM5 project [25]. The specific procedure they used is as follows, a CAGE peak was associated to a GENCODE [30] gene if it was withing 1000 bps from a GENCODE `v24` annotated TSS. The most abundant CAGE peak for each gene was then selected as the representative TSS. When no CAGE peak could be assigned to a gene, the annotated gene start position was used as the representative TSS. We $\log(1 + x)$ normalized then standardized the RNA-seq counts before training models. Sequences centered around the TSS were obtained from the human reference genome assembly GRCh37 [15].

B.2.2 Cap Analysis Gene Expression (CAGE) Profile

Data Processing Processed data was obtained from Basenji2 [37], where input sequence locations were collected as `bed` files and CAGE counts as `TensorFlow` [1] records. Original data comes from the FANTOM5 project [25]. Data was processed to produce CAGE labels for non-overlapping 128 bp bins within a sequence of 114,688 bps. For each bin, there are 638 different predictions corresponding to the CAGE count in various cell, tissue, or treatment types (e.g., fibroblast, heart, or monocytes treated with Salmonella). This resulted in an output array of 896 bins × 638 tracks for a single sample. DNA sequences were obtained from the human reference genome assembly GRCh38 [68].

The compute requirements to store and process this data make it more difficult and less accessible to users. To achieve a balance of user-friendliness while also maintaining a representative view of the data, we sub-sampled the number of tracks to 50 by using the following guidelines:

1. Only select one cell line.
2. Only keep mock treated and remove other treatments.
3. Only select one donor.

The 50 specific tracks which were selected can be found in Table 7 below. This maintains the number of sequences in the entire dataset but reduces the number of labels for each sequence from 638 to 50 thus reducing storage requirements from $\sim$84GB to $\sim$7GB.

Table 7: The 50 CAGE tracks sub-sampled for the Genomic LRB from the original 638 tracks.

Track Index	Description
0	CAGE:adipose tissue, adult, pool1
1	CAGE:bladder, adult, pool1
2	CAGE:brain, adult, pool1
3	CAGE:cervix, adult, pool1
4	CAGE:colon, adult, pool1
5	CAGE:esophagus, adult, pool1
6	CAGE:heart, adult, pool1
7	CAGE:kidney, adult, pool1
8	CAGE:liver, adult, pool1
9	CAGE:lung, adult, pool1
10	CAGE:ovary, adult, pool1
11	CAGE:placenta, adult, pool1
12	CAGE:prostate, adult, pool1
13	CAGE:skeletal muscle, adult, pool1
14	CAGE:small intestine, adult, pool1
15	CAGE:spleen, adult, pool1
16	CAGE:testis, adult, pool1
17	CAGE:thymus, adult, pool1
18	CAGE:thyroid, adult, pool1
19	CAGE:trachea, adult, pool1
20	CAGE:retina, adult, pool1
21	CAGE:temporal lobe, adult, pool1
22	CAGE:postcentral gyrus, adult, pool1
23	CAGE:pons, adult, pool1
24	CAGE:parietal lobe, adult, pool1
25	CAGE:paracentral gyrus, adult, pool1
26	CAGE:occipital pole, adult, pool1
27	CAGE:nucleus accumbens, adult, pool1
28	CAGE:medulla oblongata, adult, pool1
29	CAGE:insula, adult, pool1
30	CAGE:frontal lobe, adult, pool1
31	CAGE:dura mater, adult,
32	CAGE:corpus callosum, adult, pool1
33	CAGE:adenocarcinoma cell line:IM95m
34	CAGE:breast carcinoma cell line:MCF7
35	CAGE:diffuse large B-cell lymphoma cell line:CTB-1
36	CAGE:glioma cell line:GI-1
37	CAGE:liposarcoma cell line:SW 872
38	CAGE:Sebocyte,
39	CAGE:CD4+ T Cells,
40	CAGE:Natural Killer Cells,
41	CAGE:Neutrophils,
42	CAGE:Pericytes,
43	CAGE:Alveolar Epithelial Cells,
44	CAGE:Renal Mesangial Cells,
45	CAGE:Nucleus Pulposus Cell,
46	CAGE:Keratocytes,
47	CAGE:Mesenchymal Stem Cells - adipose,
48	CAGE:Mammary Epithelial Cell,
49	CAGE:Osteoblast,

B.3 Cis-Regulatory Element Detection

Data Processing Original data was sourced from Search Candidate cis-Regulatory Elements `v3` (SCREEN) registry by ENCODE [53]. The data is processed as follows, we break the human reference genome into 200 bp non-overlapping chunks. If the 200 bp chunk overlaps by at least 50% or more with a contiguous region from the set of annotated cis-regulatory elements (promoters or enhancers), we label them as positive, else the chunk is labeled as negative. The resulting dataset was composed of ∼15M negative samples and ∼50k positive promoter samples and ∼1M positive enhancer samples We randomly sub-sampled the negative set to 1M samples, and kept all positive samples, to make this dataset more manageable in size. DNA sequences were obtained from the human reference genome assembly GRCh38 [68].

Short-Range Since this task involves predicting the presence of a regulatory element within a specific sequence, only local context is believed to be important. The activity of promoters and enhancers in different cell types is dictated by the presence of binding sites for specific proteins [4] and thus likely do not require long-distance interactions, as demonstrated by the high predictive value of models using less than 1k bp input sequences [7, 38].

B.4 Chromatin Feature Identification

Data Processing Processed data was obtained from DeepSea [89] in the form of 1k bp sequences and labels as `txt` files. Original chromatin profiling data comes from ENCODE and Roadmap Epigenomics [53, 10]. The authors of DeepSea processed the data by chunking the human genome into 200 bp bins where for each bin labels were determined for hundreds of different chromatin features. Only bins with at least one transcription factor binding event were considered for the dataset. If the bin overlapped with a peak region of the specific chromatin profile by more than half of the sequence, a positive label was assigned. DNA sequences were obtained from the human reference genome assembly GRCh37 [15]. To make the dataset more accessible, we randomly sub-sampled the chromatin profiles from 125 to 20 tracks for the histones dataset and from 104 to 20 tracks for the DNase dataset. The sub-sampled tracks for both datasets can be found in Table 8 and Table 9.

Short-Range Chromatin features are not expected to be strongly influenced by long-range interactions. Most of the information affecting these chromatin features occurs locally and depends on the binding of different proteins [45]. This is also corroborated by the high predictive value of models using less than 1k bps input sequences [38, 89].

Table 8: 20 Histone tracks sub sampled for the Genomic LRB from the original 104 tracks with histone mark and cell type information.

Track Index	**Histone Mark**	**Cell Type**
0	H2BK12ac	H1-hESC
1	H3K4me1	NHEK
2	H3K4me2	NH-A
3	H3K9me1	K562
4	H4K20me1	NHEK
5	H2BK5ac	H1-hESC
6	H3K4me3	NH-A
7	H4K8ac	H1-hESC
8	H3K4me2	Monocytes-CD14+RO01746
9	H3K27me3	Osteoblasts
10	H3K36me3	Monocytes-CD14+RO01746
11	H3K23me2	H1-hESC
12	H3K27ac	NHLF
13	H3K36me3	NHEK
14	H2BK20ac	H1-hESC
15	H3K9ac	NHLF
16	H3K36me3	Osteoblasts
17	H2BK120ac	H1-hESC
18	H3K79me2	K562
19	H3K4me1	K562

Table 9: 20 DNase tracks sub sampled for the Genomic LRB from the original 125 tracks with cell type and treatment information.

Track Index	**Treatment**	**Cell Type**
0	None	SAEC
1	None	HRPEpiC
2	None	SK-N-MC
3	None	RWPE1
4	None	Th2
5	None	Adult_CD4_Th0
6	None	HMEC
7	None	NHEK
8	UT189	Urothelia
9	None	pHTE
10	None	Urothelia
11	None	WERI-Rb-1
12	None	Huh-7
13	None	A549
14	None	Th1
15	None	HA-h
16	None	RPTEC
17	None	HMVEC-dBl-Ad
18	None	HGF
19	None	HMF

B.5 Visualization Tool

The annotations that we join to our task datasets come from the human reference genome assembly GRCh38 [68]. To obtain these annotation we follow the methodology reported in SegmentNT [20] for data curation. Annotations include genomic elements, such as enhancers, exon, intron, 5' UTR, etc. The location of all gene elements and polyA signals were obtained from GENCODE (`v44`) [30] gene annotation. Promoter, enhancer, and CTCF-bound sites were retrieved from ENCODE's SCREEN database [79]. Promoters and enhancers were split into tissue-invariant and tissue-specific annotations, following the tissue-invariant annotations from Meuleman et al. [52]. Briefly, if a promoter or enhancer overlapped at all with a region annotated as tissue-invariant, that promoter or enhancer was annotated as tissue-invariant. All other promoters and enhancers were tagged as tissue specific. Scripts from HISAT2 [39] were used to extract respective intron and splice site annotations. Annotations of repeat regions were collected from RepeatMasker [73].

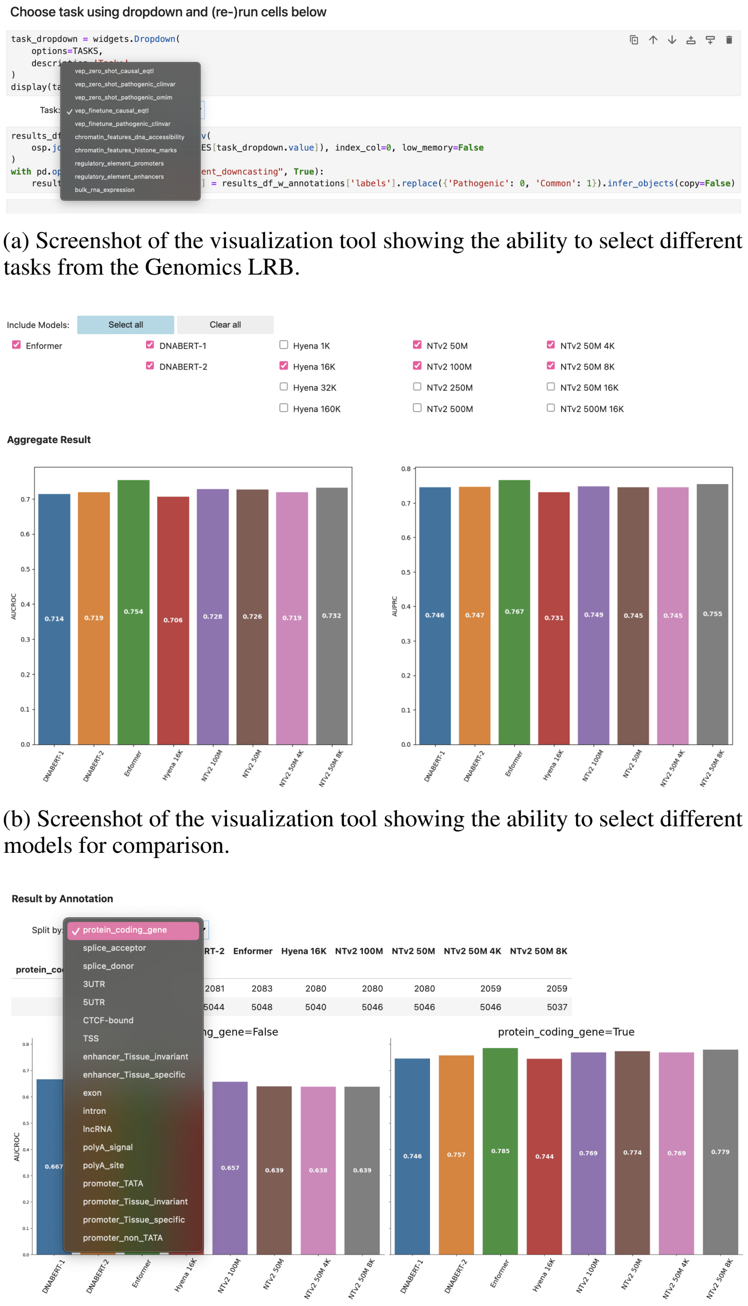
Annotations were merged into the dataset by aligning chromosome and regions (start / stop position) of annotations with the genomic locations associated with the compiled tasks in the Genomics LRB. That is, if the sequence positions in our dataset overlapped with regions in the annotation files, the sequence was tagged with the corresponding annotation. For example, for variant effect prediction tasks, the SNP location was used for the merge; for regulatory element detection tasks, the start and stop positions were used. Specifically, a sample in our dataset was associated with an annotation if the sample position was both greater than the starting position of the annotation and less than the ending position of the annotation.

The UCSC liftover browser tool [32] was used to convert GRCh38 annotations to the GRCh37 reference assembly locations to be associated with datasets relying on GRCh37 locations.

With annotations merged into the datasets in our Genomics LRB, we develop a visualization tool that enables users to 'slice' results. Our tool is an interactive `jupyter` Kluyver et al. [41] notebook that enables toggling different models and has visualizations for aggregate results, results by distance to nearest TSS / enhancer, and results by annotation. In Figure 3, we provide selected screenshots from our visualisation tool demonstrating how a user can view results for each task, select different models, and split by various annotations.

B.6 Arbitrary Sequence Length

To enable users to download arbitrarily long sequence lengths, samples for each task are stored either as single positions in the genome (e.g., the SNP location for variant effect prediction or the TSS for bulk RNA expression) or as start and stop locations for tasks like regulatory element and chromatin feature prediction. In addition we store the human reference genome assemblies GRCH38 [68] and GRCH37 [15]. The `PyFaidx` Python package [71] is used to create an indexed `FASTA` file object from the reference genomes for fast random access to any subsequence. With the user's requested sequence length, we symmetrically extend sequence locations from our datasets and use these extended indices to extract the underlying DNA sequence from the indexed reference genomes. If the extended sequence indexes beyond a chromosome boundary, the sample is not returned.



Choose task using dropdown and (re-)run cells below

```
task_dropdown = widgets.Dropdown(
    options=TASKS,
    descri
)
display(ta
```

Task: ✓ vep_finetune_causal_eqtl

```
results_df                                          v(
    osp.jc                                          ES[task_dropdown.value]), index_col=0, low_memory=False
)
with pd.op                                          ent_downcasting", True):
    result                                          ] = results_df_w_annotations['labels'].replace({'Pathogenic': 0, 'Common': 1}).infer_objects(copy=False)
```

vep_zero_shot_causal_eqtl
vep_zero_shot_pathogenic_clinvar
vep_zero_shot_pathogenic_omim
✓ vep_finetune_causal_eqtl
vep_finetune_pathogenic_clinvar
chromatin_features_dna_accessibility
chromatin_features_histone_marks
regulatory_element_promoters
regulatory_element_enhancers
bulk_rna_expression

(a) Screenshot of the visualization tool showing the ability to select different tasks from the Genomics LRB.

Include Models: Select all | Clear all

☑ Enformer
☑ DNABERT-1
☑ DNABERT-2
☐ Hyena 1K
☑ Hyena 16K
☐ Hyena 32K
☐ Hyena 160K
☑ NTv2 50M
☑ NTv2 100M
☐ NTv2 250M
☐ NTv2 500M
☑ NTv2 50M 4K
☑ NTv2 50M 8K
☐ NTv2 50M 16K
☐ NTv2 500M 16K

Aggregate Result

(b) Screenshot of the visualization tool showing the ability to select different models for comparison.

Result by Annotation

Split by: ✓ protein_coding_gene

protein_co	RT-2	Enformer	Hyena 16K	NTv2 100M	NTv2 50M	NTv2 50M 4K	NTv2 50M 8K
	2081	2083	2080	2080	2080	2059	2059
	5044	5048	5040	5046	5046	5046	5037

(c) Screenshot of the visualization tool showing the ability to select different annotations by which to split results.

Figure 3: Sample screenshots from our interactive visualization tool.

C Context Length Extension

Rotary Embeddings In attention-based modules, such as those used in transformer models [82], for a sequence of length L, the model takes embeddings in $\{\mathbf{x}\}_{j=1}^{L}, \mathbf{x}_j \in \mathbb{R}^d$, where d is the dimension of the embeddings, and computes query, key, and value vectors at every m^{th} and n^{th} position in the sequence:

$$\begin{aligned}\mathbf{q}_m &= f_q(\mathbf{x}_m, m)\\ \mathbf{k}_n &= f_k(\mathbf{x}_n, n)\\ \mathbf{v}_n &= f_v(\mathbf{x}_n, n).\end{aligned}$$

f_q, f_k, f_v are query, key, and value transformations, respectively. For rotary embeddings (RoPE [74]), we can think of $\mathbb{R}^d$ as equivalent to the complex field $\mathbb{C}^{d/2}$ and define f_q and f_k as:

$$\begin{aligned}f_q(\mathbf{x}_m, m) &= \mathrm{e}^{im\Theta}\mathbf{W}_q\mathbf{x}_m\\ f_k(\mathbf{x}_n, n) &= \mathrm{e}^{in\Theta}\mathbf{W}_k\mathbf{x}_n,\end{aligned}$$

where $\mathbf{W}_q$ and $\mathbf{W}_k$ are linear transformations and $\Theta = \mathrm{diag}(\theta_1, \ldots, \theta_{d/2})$ is a diagonal matrix, with $\theta_j = b^{-2j/d}$ and $b = 10000$.

RoPE Position Interpolation In the concurrent works of Chen et al. [13] and kaiokendev [35], the method of position interpolation was introduced, whereby longer sequences of length $L' > L$ are accommodated by simply rescaling the position input to f_q and f_k, e.g., $f_q(\mathbf{x}_m, m\frac{L}{L'})$.

NTK-aware RoPE Interpolation An alternative interpolation scheme, attributed to bloc97 [11], is motivated by the hypothesis that position interpolation may lead to the loss of high frequency information. The approach that purportedly resolves this issue is related to the theory of Neural Tangent Kernels (NTK) by means of an analogy between RoPE and Fourier Features [76], and is thus named "NTK-aware" interpolation. This scheme is characterized by a rescaling applied not to the position but rather to the basis of rotation, as follows:

$$\begin{aligned}\theta_j &= b'^{-2j/d}\\ b' &= b \cdot \left(\frac{L}{L'}\right)^{\frac{d}{d-2}}\end{aligned}$$

In the experiments on context extension presented in the main text, we adopt this interpolation scheme.

We note that the authors in Peng et al. [60] further tweak and build on NTK-aware interpolation to create their proposed interpolation scheme, which they title YaRN. However, the full YaRN approach, as presented in Peng et al. [60] requires several manually tuned hyperparameters, which were carefully selected for the decoder-only generative Llama-2 7 billion parameter model [80, 81]. We therefore adopted the simpler NTK-aware approach in our experiments.

D Additional Experimental Details

D.1 Evaluated DNA Language Models

In Table 10, we list the DNA LMs included in the initial evaluation of our benchmark.

Table 10: Overview of DNA LMs evaluated in this study.

	Pre-training	**Data**	**Parameters**	**Architecture**	**Context** (bps)	**Tokenization**
NTv2	MLM	Multi-Species	50M, 100M, 250M, 500M	Transformer	12k	6-mer
DNABERT-1	MLM	Human Reference	88.6M	Transformer	512 bps	6-mer
DNABERT-2	MLM	Multi-Species	116.6M	Transformer	700 (train), up to 10k (eval)	Byte Pair Encoding
HyenaDNA	NTP	Human Reference	1.6M, .6M, 3.9M,12.9M	SSM	1k, 16k, 32k, 160k	Single Base Pair

D.2 Zero-Shot Evaluation

For masked DNA LMs, zero-shot scores are computed by masking the variant position in the sequence, performing inference on the masked sequence, and obtaining the probability distribution at the variant position. A score is then calculated using the probabilities of the reference allele token and the alternative allele token. For auto-regressive DNA LMs, no masking is required due to their unidirectional nature. Instead, a forward pass is done with the reference sequence, and the probability distribution is extracted from the token immediately preceding the variant position. Scores are computed as the log probability ratio for the reference (ref) and alternative (alt) allele tokens:

$$\text{variant effect score} = \log\left(\frac{P_{\text{ref}}}{P_{\text{alt}}}\right)$$

Details about additional processing required for zero-shot prediction are given below.

D.2.1 Causal eQTL

The original dataset used for this tasks contains tissue information for each sequence. Given that zero-shot evaluate cannot account for tissue, we process variants appearing across multiple tissue types as follows: first, we find variants appearing in multiple tissues and determining a consensus label for a given variant across tissues using a 70% majority class agreement threshold. Variants appearing across multiple tissues whose majority class agreement was below this threshold were dropped. When computing metrics we only count variants appearing across tissues once.

D.2.2 Pathogenic-OMIM

Due to computational considerations and given that this data set totals $\sim$2.3M examples, we only considered a subset of the common variants for carrying out zero-shot prediction. Specifically, we sub-sampled 200k common variants and kept all 406 original pathogenic variants.

D.3 Fine-tuning Evaluation

To fine-tune models on our benchmark tasks, we first extracted model embeddings, in the case of DNA LMs this involves extracting the output of the last layer before the LM head, and in the case of Enformer, this involves extracting the model embeddings before the final supervised prediction head. Model embeddings were then processed in a task specific manner and subsequently fed into a task specific MLP, both of which are outlined below. We note that for Enformer, since it is a model that was originally trained in a multi-task supervised fashion and not intended to be fine-tuned, embeddings were frozen and only the prediction head was trained.

D.3.1 Causal eQTL

Embedding Extraction We extract model embeddings for both the reference and alternative sequences and average embeddings across a window of size 1536 bps symmetrically around the SNP position. The mean embeddings for the reference and alternative are concatenated. Tissue information is converted to one-hot and additionally concatenated to the reference-alternative embedding vector.

MLP Head MLP hidden dimensions are sized in an adaptive way such the hidden state size is equal to two times the base model's embedding dimension. The MLP is composed of one linear layer with size $2 \times \text{embedding dimension}$, a softplus activation, another linear layer with size $2 \times \text{embedding dimension}$, a softplus activation, and a final linear layer for binary prediction.

Hyperparameters The parameters used to fine-tune models on this task include batch size = 64, learning rate = $1\mathrm{e}^{-5}$, ADAM [40] optimizer with β_1= 0.9, β_2 = 0.999, and $\epsilon = 1\mathrm{e}^{-8}$, trained for 1 epoch on the task's training dataset. Validation is carried out every 70 parameter update steps.

D.3.2 Pathogenic ClinVar

Embedding Extraction We extract model embeddings for both the reference and alternative sequences and take a window mean of size 1536 bps symmetrically around the SNP position. The mean embeddings for the reference and alternative are concatenated together.

MLP Head MLP hidden dimensions are sized in an adaptive way such the hidden state size is equal to two times the base model's embedding dimension. The MLP is composed of one linear layer with size $2 \times \text{embedding dimension}$, a softplus activation, and a final linear layer for binary prediction.

Hyperparameters The parameters used to fine-tune models on this task include batch size = 64, learning rate = $1e^{-5}$, ADAM optimizer with β_1 = 0.9, β_2 = 0.999, and ϵ = $1e^{-8}$, trained for 3 epochs on the task's training dataset. Validation is carried out every 40 parameter update steps.

D.3.3 Bulk RNA Expression

Embedding Extraction We extract model embeddings for the input sequence and take perform mean pooling on a window centered on the TSS with 383 bps before the TSS and 256 bp after.

MLP Head MLP hidden dimensions are sized in an adaptive way such the hidden state size is equal to two times the base model's embedding dimension. The MLP is composed of one linear layer with size $2 \times \text{embedding dimension}$, a softplus activation, and a final linear layer for predicting 218 regression values.

Hyperparameters The parameters used to fine-tune models on this task include batch size = 64, learning rate = $3e^{-5}$, ADAM optimizer with β_1= 0.9, β_2 = 0.999, and ϵ = $1e^{-8}$ trained for 3 epochs on the task's training dataset. Validation is carried out every 50 parameter update steps.

D.3.4 CAGE Prediction

Embedding Extraction Base model embeddings were extracted and fed into the task MLP predictor.

MLP Head MLP hidden dimensions are sized in an adaptive way such the hidden state size is equal to two times the base model's embedding dimension. The MLP is composed of one linear layer with size $2 \times \text{embedding dimension}$, a softplus activation, and a final linear layer for predicting 218 regression values.

Hyperparameters The parameters used to fine-tune models on this task include batch size = 64, learning rate = $3e^{-5}$, adam optimizer with β_1= 0.9, β_2 = 0.999, and ϵ = $1e^{-8}$ trained for 1 epoch of the training dataset. Validation is carried out every 50 parameter update steps.

D.3.5 Regulatory Elements

Due to computational considerations, we only fine-tuned models on a randomly sampled 100k subset of the full $\sim$1-2M samples in the training set . Models were evaluated on the full test dataset. The subset data is provided in our HuggingFace repository.

Embedding Extraction Given that the task is defined on predicting the presence of a regulatory element in the center 200 bp of the sequence, we extract a central window of 200 bps from the sequence of embeddings and perform mean pooling. This mean embedding is then passed as input to the MLP predictor head.

MLP Head MLP hidden dimensions are sized in an adaptive way such the hidden state size is equal to two times the base model's embedding dimension. The MLP is composed of one linear layer with size $2 \times \text{embedding dimension}$, a softplus activation, and a final linear layer for predicting binary values.

Hyperparameters The parameters used to fine-tune models on this task include batch size = 64, learning rate = $3e^{-5}$, ADAM optimizer with β_1= 0.9, β_2 = 0.999, and ϵ = $1e^{-8}$ trained for 1 epoch of the sampled training dataset for each task. Validation is carried out every 30 parameter update steps.

D.3.6 Chromatin Features

Due to computational considerations, we only fine-tuned models on a randomly sampled 100k subset from the full $\sim$2M sample training set. Models were evaluated on the full test dataset. The subset data is provided in our HuggingFace repository.

Embedding Extraction Given that the task is defined on predicting the presence of a chromatin feature in the center 200 bp of the sequence, we extract a central window of 200 bps from the sequence of embeddings and perform mean pooling. This mean embedding is then passed as input to the MLP predictor head.

MLP Head MLP hidden dimensions are sized in an adaptive way such the hidden state size is equal to two times the base model's embedding dimension. The MLP is composed of one linear layer with size $2 \times \mathrm{embedding\ dimension}$, a softplus activation, and a final linear layer for predicting the 20 binary labels.

Hyperparameters The parameters used to fine-tune models on this task include batch size = 64, learning rate = $3\mathrm{e}^{-5}$, adam optimizer with β_1= 0.9, β_2 = 0.999, and ϵ = $1\mathrm{e}^{-8}$ trained for 1 epoch of the training dataset. Validation is carried out every 30 parameter update steps.

D.4 Fine-tuning Ablation Details

For the fine-tuning ablation study, we compared training only the task MLP with DNA LM embeddings frozen against training all DNA LM weights in conjunction with the task MLP. All training setup details regarding embedding extraction and hyperparameters were kept constant except for learning rate which was adjusted to account for training larger networks when full fine-tuning. The following learning rates for each task were used in the MLP only training:

- Variant effect prediction tasks: $1\mathrm{e}^{-4}$
- Bulk RNA: $2.5\mathrm{e}^{-4}$
- CAGE: $2\mathrm{e}^{-4}$
- Regulatory elements: $2.5\mathrm{e}^{-4}$
- Chromatin features: $2.5\mathrm{e}^{-4}$.

D.5 Context Extension Implementation Details

To conduct context length extension of NTv2, we first used the 50M model due to computation considerations. We started with the pre-trained NTv2 50M checkpoint from Dalla-Torre et al. [18], pre-trained on 12k bp sequences, and extended the context length by factors of two to 24k, 48k, and 96k bps using a second stage of masked language modeling on a multi-species dataset from Dalla-Torre et al. [18]. After proving out this methodology for the 50M model, we conducted context length extension for the 500M model at 96k bps.

Hyperparameters For the 50M NTv2 model we use the following hyperparameters: batch size = 1M tokens, full precision training, masking ratio = 0.15, masking probability = 0.8, random token probability = 0.1. The ADAM optimizer with weight decay regularization was used with weight decay = 0.01, β_1 = 0.9, β_2 = 0.999, ϵ = $1\mathrm{e}^{-8}$, a modified square decay learning rate schedule, with initial learning rate of $6\mathrm{e}^{-5}$ and end learning rate of $8\mathrm{e}^{-4}$ with 1000 warm up steps. Training was conducted over $\sim$5 billion tokens totalling $\sim$5k parameter update steps.

All hyperparameters were kept constant for the NTv2 500M model, however due to limited memory resources, mixed precision training was used.

E Additional Results

E.1 Full DNA LM Series Evaluations

In Tables 11 and 12 we display results for the full set of models evaluated on our benchmark.

DNABERT-2 was not fine-tuned on the CAGE task due to the incompatibility between the byte pair tokenization this model employs and binned labels used in this task.

Table 11: Benchmarking performance of DNA LMs and baselines on variant effect prediction tasks. Models were evaluated in both fine-tuning and zero-shot settings. *Extended NTv2 500 M was fine-tuned with 60k bp sequences due to compute constraints..

Model Name	**Context** *(bp)*	**Causal eQTL** *(AUROC)*		**Pathogenic ClinVar** *(AUROC)*		**Pathogenic OMIM** *(AUPRC)*
		Fine-tune	*Zero-shot*	*Fine-tune*	*Zero-shot*	*Zero-shot*
DNABERT 1	512	0.72 ± 0.003	0.51	0.67 ± 0.037	0.50	0.002
DNABERT 2	10k	0.72 ± 0.008	0.50	0.74 ± 0.013	0.50	0.002
NTv2 50M	12k	0.72 ± 0.005	0.51	0.75 ± 0.008	0.53	0.002
NTv2 100M	12k	0.73 ± 0.003	0.51	0.76 ± 0.009	0.56	0.002
NTv2 250M	12k	0.72 ± 0.003	0.51	0.78 ± 0.013	0.58	0.002
NTv2 500M	12k	0.72 ± 0.003	0.51	0.78 ± 0.009	0.68	0.003
HyenaDNA 1K	1k	0.71 ± 0.005	0.51	0.63 ± 0.027	0.49	0.002
HyenaDNA 16K	16k	0.71 ± 0.005	0.51	0.66 ± 0.016	0.49	0.002
HyenaDNA 32K	32k	0.72 ± 0.002	0.51	0.66 ± 0.012	0.50	0.002
HyenaDNA 160K	160k	0.71 ± 0.010	0.51	0.56 ± 0.073	0.49	0.002
Extended NTv2 50M 24K	24k	0.72 ± 0.004	0.51	0.75 ± 0.009	0.53	0.002
Extended NTv2 50M 48K	48k	0.73 ± 0.008	0.51	0.65 ± 0.059	0.52	0.002
Extended NTv2 50M 96K	96k	0.73 ± 0.006	0.51	0.74 ± 0.019	0.51	0.002
Extended NTv2 500M 96K*	96k	0.74 ± 0.004	0.51	0.75 ± 0.018	0.53	0.002
Baseline		0.76 ± 0.002 (Enformer)	0.56 (CADD)	0.65 ± 0.031 (Enformer)	0.97 (CADD)	0.205 (CADD)

Table 12: Benchmarking performance of DNA LMs and baselines on gene expression prediction, regulatory element, and chromatin features prediction tasks. Models were evaluated in only a fine-tuned setting for this set of tasks. DNABERT-2 was not fine-tuned on the CAGE task due to the incompatibility of the byte pair tokenization with binned labels.

	Context *(bp)*	**Bulk RNA** *(R^2)*	**CAGE** *(R^2)*	**Promoter** *(AUPRC)*	**Enhancer** *(AUROC)*	**Histone Marks** *(AUPRC)*	**DNA Accessibility** *(AUPRC)*
		Fine-tune	*Fine-tune*	*Fine-tune*	*Fine-tune*	*Fine-tune*	*Fine-tune*
DNABERT-1	512	0.47 ± 0.007	0.14 ± 0.025	0.72 ± 0.009	0.80 ± 0.005	0.23 ± 0.003	0.18 ± 0.006
DNABERT-2	10k	0.51 ± 0.050	-	0.71 ± 0.112	0.81 ± 0.022	0.24 ± 0.091	0.15 ± 0.064
NTv2 50M	12k	0.52 ± 0.074	0.35 ± 0.030	0.75 ± 0.008	0.78 ± 0.041	0.34 ± 0.007	0.18 ± 0.005
NTv2 100M	12k	0.52 ± 0.081	0.3 ± 0.030	0.78 ± 0.008	0.82 ± 0.010	0.34 ± 0.007	0.22 ± 0.012
NTv2 250M	12k	0.57 ± 0.024	0.37 ± 0.008	0.8 ± 0.008	0.84 ± 0.002	0.37 ± 0.013	0.28 ± 0.006
NTv2 500M	12k	0.57 ± 0.016	0.39 ± 0.011	0.79 ± 0.006	0.82 ± 0.002	0.38 ± 0.003	0.3 ± 0.007
HyenaDNA 1K	1k	0.44 ± 0.014	0.11 ± 0.015	0.7 ± 0.006	0.80 ± 0.002	0.21 ± 0.001	0.13 ± 0.003
HyenaDNA 16K	16k	0.46 ± 0.008	0.17 ± 0.014	0.7 ± 0.007	0.80 ± 0.006	0.22 ± 0.002	0.091 ± 0.003
HyenaDNA 32K	32k	0.47 ± 0.010	0.22 ± 0.007	0.72 ± 0.007	0.82 ± 0.002	0.22 ± 0.003	0.084 ± 0.001
HyenaDNA 160K	160k	0.46 ± 0.006	0.19 ± 0.032	0.67 ± 0.009	0.74 ± 0.009	0.25 ± 0.004	0.11 ± 0.002
Extended NTv2 50M 24K	24k	0.53 ± 0.063	0.37 ± 0.010	0.75 ± 0.007	0.83 ± 0.002	0.35 ± 0.007	0.19 ± 0.006
ExtendedNTv2 50M 48K	48k	0.54 ± 0.038	0.36 ± 0.012	0.76 ± 0.008	0.82 ± 0.002	0.35 ± 0.007	0.19 ± 0.006
Extended NTv2 50M 96K	96k	0.54 ± 0.034	0.3 ± 0.019	0.76 ± 0.015	0.83 ± 0.001	0.35 ± 0.005	0.19 ± 0.007
Extended NTv2 500M 96K	96k	0.56 ± 0.037	0.36 ± 0.011	0.78 ± 0.003	0.82 ± 0.005	0.38 ± 0.004	0.3 ± 0.006
Baseline		0.80 ± 0.010 (Enformer)	0.49 ± 0.000 (Enformer)	0.86 ± 0.006 (Enformer)	0.92 ± 0.002 (Enformer)	0.35 (DeepSea)	0.44 (DeepSea)

E.2 Additional Fine-tuning Ablation

In Table 13, we display results for the the full NTv2 series and additional HyenaDNA models. We find that the same pattern discussed in Section 5.5 holds for this larger set of models as well. Namely, full fine-tuning almost uniformly improves model performance relative to partial fine-tuning, by margins that can range up to > 100%. Tasks on which DNA LMs already perform competitively, e.g., regulatory element annotation, seem to benefit less from full-fine tuning, but even here we do see gains.

E.3 Additional Results by Genomic Annotations

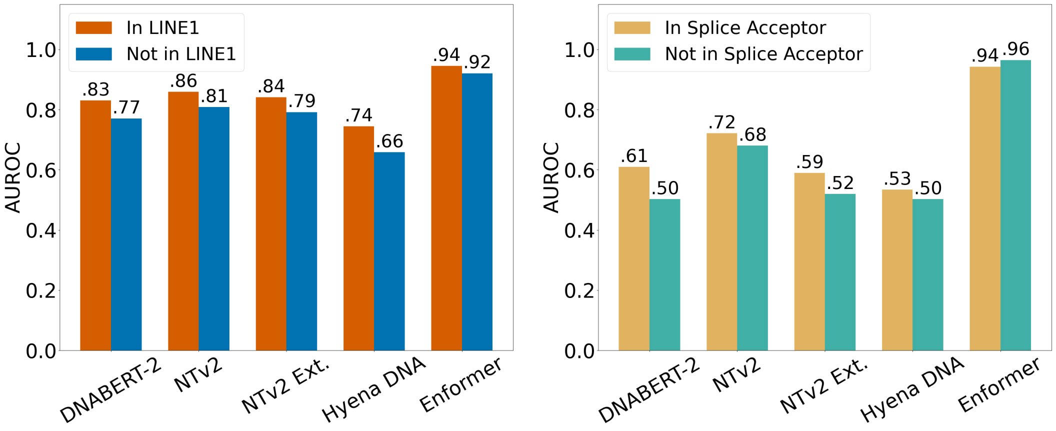
In Figure 4, we display additional results from splitting the tasks by genomic annotations.

Table 13: Ablation study examining the difference in performance of DNA LM fine-tuning strategies. Results shown correspond to the percent increase in performance of full fine-tuning with respect to freezing LM weights and only training the MLP head.

	Causal eQTL *(AUCROC)*	**Pathogenic ClinVar** *(AUROC)*	**Bulk RNA** (R^2)	**CAGE** (R^2)	**Promoter** *(AUPRC)*	**Enhancer** *(AUROC)*	**Histone Marks** *(AUCPRC)*	**DNA Accessibility** *(AUPRC)*
NTv2 50M	+1.13	+9.30	+30.23	+71.60	+1.93	-2.05	+32.03	+33.43
NTv2 100M	+0.98	+6.24	+13.70	+27.72	+2.16	+2.83	+32.70	+40.54
NTv2 250M	+0.36	+3.57	+21.70	+40.41	+2.07	+3.71	+31.01	+54.44
NTv2 500M	+0.49	+4.27	+18.29	+42.14	-1.45	+0.90	+22.46	+47.96
HyenaDNA 1K	+0.95	+15.39	+16.50	+45.22	+7.13	+4.68	+23.61	+22.65
HyenaDNA 16K	+0.21	+22.81	+75.53	+133.52	+16.08	+6.16	+42.83	-9.62
HyenaDNA 32K	+0.35	+11.58	+107.48	+102.91	+5.09	+5.39	+14.43	-22.67

Enhancer Detection We find that DNA LMs have increased performance at identifying enhancers in some repetitive elements, such as LINE1 transposons, as shown in Figure 4a. LINE1 elements are commonly interspersed along the human genome, and individual LINE1 elements may have uncertain regulatory effects, but DNA LMs appear to be able to call enhancers in LINE1 elements better than in non-LINE1 regions. However, their performance still lags that of the Enformer baseline.

Zero-shot Pathogenic-ClinVar In Figure 4b, we observe that most models exhibit increased performance within splice site acceptor regions, with the exception of Enformer, although Enformer demonstrates high performance in both splits.



(a) Enhancer detection; split by enhancers located within a LINE1 (transposon) annotation.

(b) Zero-shot Pathogenic ClinVar prediction; by splice site acceptor annotation.

Figure 4: Additional results split by genomic annotations.

F Hosting, Licensing, and Maintenance Plan

The Genomics Long Range Benchmark is hosted as a dataset repository on Hugging Face[86] at this url. Users may view and download data by accessing the repository through their browser or can use Hugging Face's API to access and download the dataset programmatically, with a single line of code. In addition to ease-of-use, Hugging Face repositories provide a reliable, stable way to store data.

Users may raise issues and submit pull requests via the Genomic LRB's Hugging Face webpage which will be addressed by the authors.

The Genomics LRB is licensed under the Creative Commons Attribution Non Commercial Share Alike 4.0 license

G Potential Societal Impacts

As our work introduces a benchmark, we do not believe it poses any inherent negative societal impacts. In fact, our work will hopefully create a positive impact by accelerating the development of useful DNA LMs that can bring about a deeper understanding of biology.

H Assets

In Table 14, we list the open source libraries and repositories used in this work, with their coresponding licenses.

Table 14: Open source libraries (and corresponding licenses) used in this work.

Library	License
Biopython [16]	Biopython license
Haiku [31]	Apache 2.0
HuggingFace [86]	Apache 2.0
Jax [12]	Apache 2.0
Jupyter [41]	BSD 3-Clause
NumPy [29]	NumPy license
Matplotlib [33]	Matplotib license
Pandas [78]	BSD 3-Clause “New" or “Revised"
Optax [21]	Apache 2.0
PyFaidx [71]	BSD-3-Clause
PyTorch [58]	BSD-3 Clause
Scikit-Learn [59]	BSD 3-Clause
Seaborn [85]	BSD 3-Clause “New" or “Revised"
TensorFlow [1]	Apache 2.0

I Computational Resources

All research in this study was conducted using Cloud TPU’s provided by Google’s TPU Research Cloud program. Specifically, a TPU-v4-64 slice was used for all context length extension pre-training. Single TPU-v4 machines were used in parallel to conduct all benchmarking and evaluations including fine-tuning, zero-shot, and inference experiments.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Contributions of the benchmark are outlined in Section 3 and results are detailed in Section 5.
   (b) Did you describe the limitations of your work? [Yes] Limitations discussed in Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix G.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The link to download the benchmark dataset is included in the abstract. The link to our visualization tool is included in the contributions list at the end of Section 1.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Full experimental details are stated in Section 5 and Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Error bars were attained from standard deviation across five runs of cross validation with different random seeds used to split training / validation sets.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix I for details on compute used in this work.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Table 14. Additionally we cite all original data sources in Section 3 and Appendix B.
   (b) Did you mention the license of the assets? [Yes] See Table 14.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Link to benchmark datasets is included in the abstract.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix B.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]