Accurate and General DNA Representations Emerge from Genome Foundation Models at Scale

Caleb N. Ellington[1], **Ning Sun**[1,2*], **Nicholas Ho**[1,3*], **Tianhua Tao**[1,4*], **Sazan Mahbub**[1,3*], **Dian Li**[1], **Yonghao Zhuang**[1,3*], **Hongyi Wang**[1], **Le Song**[1,2†], **Eric P. Xing**[1,2,3†]

[1]GenBio AI

[2]Mohamed bin Zayed University of Artificial Intelligence
[3]Carnegie Mellon University
[4]University of Washington

Abstract

Language models applied to protein sequences have become a panacea, enabling therapeutics development, materials engineering, and core biology research. Despite the successes of protein language models, genome language models remain nascent. Recent studies suggest the bottleneck is data volume or modeling context size, since long-range interactions are widely acknowledged but sparsely annotated. However, it may be the case that even short DNA sequences are modeled poorly by existing approaches, and current models are unable to represent the wide array of functions encoded by DNA. To study this, we develop AIDO.DNA, a pretrained module for DNA representation in an AI-driven Digital Organism [1]. AIDO.DNA is a seven billion parameter encoder-only transformer trained on 10.6 billion nucleotides from a dataset of 796 species. By scaling model size while maintaining a short context length of 4k nucleotides, AIDO.DNA shows substantial improvements across a breadth of supervised, generative, and zero-shot tasks relevant to functional genomics, synthetic biology, and drug development. Notably, AIDO.DNA outperforms prior encoder-only architectures *without* new data, suggesting that new scaling laws are needed to achieve compute-optimal DNA language models. Models and code are available through ModelGenerator in `https://github.com/genbio-ai/AIDO` and on Hugging Face.

1 Introduction

Genomes are the product of billions of years of evolution and selection under the forces of fitness, competition, niches, and stochasticity. They are a nearly universal requirement for life, encoding all cellular systems and their components – RNA, proteins, their assembly, and their regulation – through a simple 4-chemical vocabulary allowing self-replication, recombination, and inheritance. The conservation of genomic elements across evolutionary timelines is highly correlated to their functional roles and interactions within the living systems they encode. Aligning sequences to determine their elementwise conservation has helped to discover protein structures, binding sites, regulatory elements, 3D genome structure, pathologies, and genealogies. However, exact function is difficult to determine from conservation alone. Instead, conservation analysis often plays the role of data preprocessing for feature selection, which has traditionally limited downstream genomic analyses to use features which are generally well-sampled, well-characterized, and conserved across

[*] Work done during internship at GenBio AI.

[†] Corresponding authors: le.song@genbio.ai, eric.xing@genbio.ai

AI for New Drug Modalities at NeurIPS 2024.

Table 1: Catalog of deep learning methods for sequence representation and prediction.

Name	Year	Model Primitive	Tokenization	Train Context (kbp)	Parameters
DeepSEA[2]	2015	Classification	Nucleotide	1	40M
Basenji[3]	2020	Regression	128bp Bins	131	$\sim$100M
DNABERT[4]	2021	Joint	k-mer	3	89M
Enformer[5]	2021	Regression	128bp Bins	200	252M
genSLM[6]	2022	Joint	Codon	30	25B
DNABERT-2[7]	2023	Joint	k-mer BPE	10	117M
HyenaDNA[8]	2023	Causal SSM	Nucleotide	32	1.6M
Mamba[9]	2023	Causal SSM	Nucleotide	1000	40M
NT v1[10]	2023	Joint	k-mer	8	2.5B
NT v2[11]	2023	Joint	k-mer	12	500M
GPN-MSA[12]	2023	Joint	Nuc + MSA	0.1	86M
Caduceus[13]	2024	2-way SSM	Nucleotide	131	7.7M
Evo[14]	2024	Causal SSM	Nucleotide	131	7B
Species-LM[15]	2024	Joint	k-mer	3	89M
gLM[16]	2024	Joint	Nucleotide	92	1B
LucaOne[17]	2024	Joint	Nucleotide	4	1.8B
DNA Foundation	ours	Joint	Nucleotide	4	7B

species. Accurate and general genome analysis requires new computational and statistical tools for pre-processing genomic data outside of this regime.

Recently, large unsupervised pretrained foundational models for predicting protein sequence conservation have achieved widespread success on difficult and diverse tasks that decades of prior work has sought to address, including protein folding and *de novo* design [18, 19], exonic variant pathogenicity prediction [20, 21], and representation of evolutionary dynamics beyond multiple sequence alignment [22]. These conservation models accurately predict protein structure, catalytic activity, and a protein's role and fitness within a broader biological system , the same way a traditional multiple-sequence alignment would be used, while also generalizing to human-only and novel sequences [20], and generating new sequences for desired functions [23]. Despite these successes, the architectures for protein foundation models have not shown the same success when applied more broadly to genomes [24, 25, 26].

Recent works suggest the gap in performance may be related to the length of the total genomic context accommodated by the model, and thus propose new architectures to learn long-range interactions and their role in genetic regulation [8, 13, 10]. However, increasing context length to millions of nucleotides has shown diminishing returns [9]. While the field appears at an impasse, reviewing the growth of deep learning models in genomics reveals successful variants and their proliferation in research (Table 1). While no consensus has been reached on the optimal context size for representation learning, parameter sizes continue to grow and vanilla BERT-style encoder-only transformer architectures have been widely successful.

However, DNA encoders are still dwarfed by their protein cousins [23]. This is alarming, since a general genome language model should implicitly contain a protein language model while also representing all other molecular functions encoded by DNA. Given this broad modeling scope, we believe that these models require vastly more representation capacity than has previously been attempted. To learn accurate and general representations of genomic functions, we develop AIDO.DNA, one of the largest unsupervised pretrained DNA encoders to date at seven billion parameter scale. We train this model at single nucleotide resolution on 796 species' genomes with 10.6 billion nucleotides. To address concerns about context length, we explicitly set the context to a relatively short 4,000 nucleotides. By achieving a new state-of-the-art (SOTA) on canonical tasks in transfer learning, zero-shot prediction, and sequence generation, AIDO.DNA represents an important step toward the development of an AI-driven Digital Organism [1], and directly enables new approaches to core biology research, genomics, synthetic biology, metabolic engineering, and therapeutics design.

2 Methods

We develop AIDO.DNA at 300M and 7B parameter scales, using large-scale distributed pretraining [27] and parameter efficient transfer learning frameworks [28] to rapidly scale pretraining and

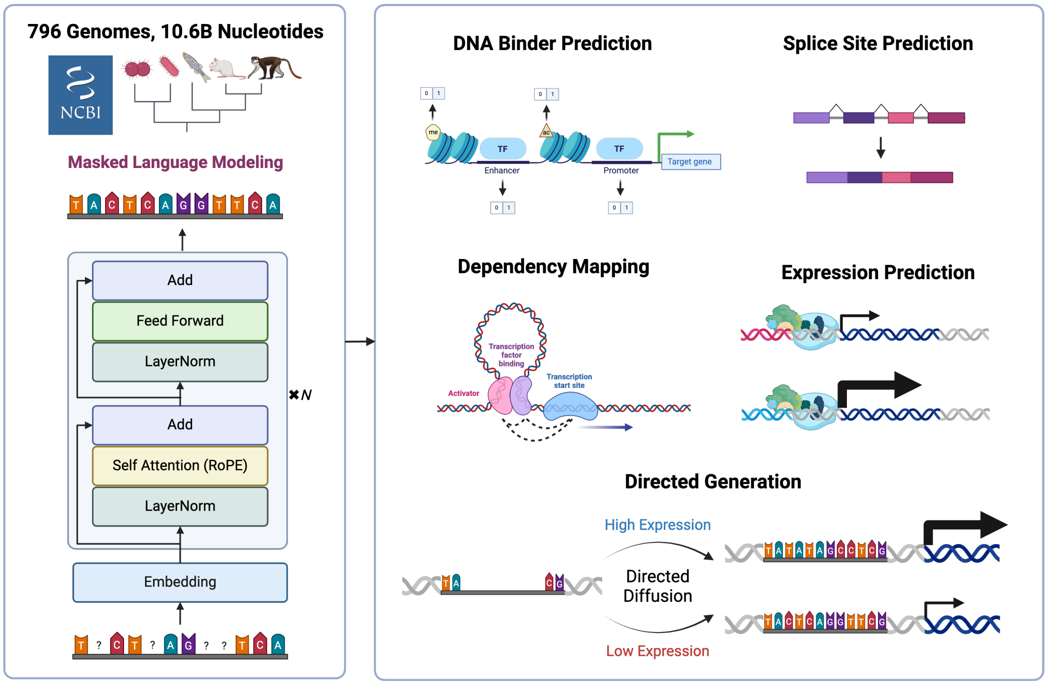


Figure 1: AIDO.DNA pretraining (left) and evaluation (right).

finetuning. Our model is based on the bidirectional transformer encoder (BERT) architecture [29] with single-nucleotide tokenization, and is optimized using a masked language modeling (MLM) training objective.

2.1 Pretraining

Data To test whether representation capacity has limited the development of DNA language models in previous studies, we utilize the data set and splits from the Nucleotide Transformer [10]. Starting from a total of 812 genomes with 712 for training, 50 for validation, and 50 for testing, we removed 17 entries which had been deleted from NCBI since the original dataset's publication on Hugging Face. One of these was the important model organism *Rattus norvegicus*, which we replaced with the current reference genome. This resulted in 696 genomes for training, 50 for validation, and 50 for testing. With a total of 10.6 billion training tokens, we pretrained AIDO.DNA at 300M and 7B parameter scales.

Tokenization To appropriately adapt langauge models to biology, it is important to be clear that DNA is not a sentence. DNA is an aperiodic crystal composed of four chemical building blocks – the nucleotides adenine (A), thymine (T), cytosine (C), and guanine (G) – which are paired and polymerized on a phosphate backbone. DNA has no periods, no sentences, no paragraphs, and no global grammar. With its very small vocabulary, DNA encodes an enormous diversity of biomolecules and functions. Compared with natural language, it has a very low per-character information density on average, and grammars are highly local, relativistic, and context-specific. Prior works which compress nucleotides into "words" [10, 7] make it difficult to learn relative and local grammars (e.g. codons) and dilute the representation of high impact context-specific effects such as single nucleotide pathogenic variants [30].

To minimize bias and learn high-resolution single-nucleotide dependencies, we opted to align closely with the real data and use character-level tokenization with a 5-letter vocabulary: A, T, C, G, N, where N is commonly used in gene sequencing to denote uncertain elements. Sequences were also prefixed with a `[CLS]` token and suffixed with a `[EOS]` token as hooks for downstream tasks. We chose a context length of 4,000 nucleotides as the longest context which would fit within our largest

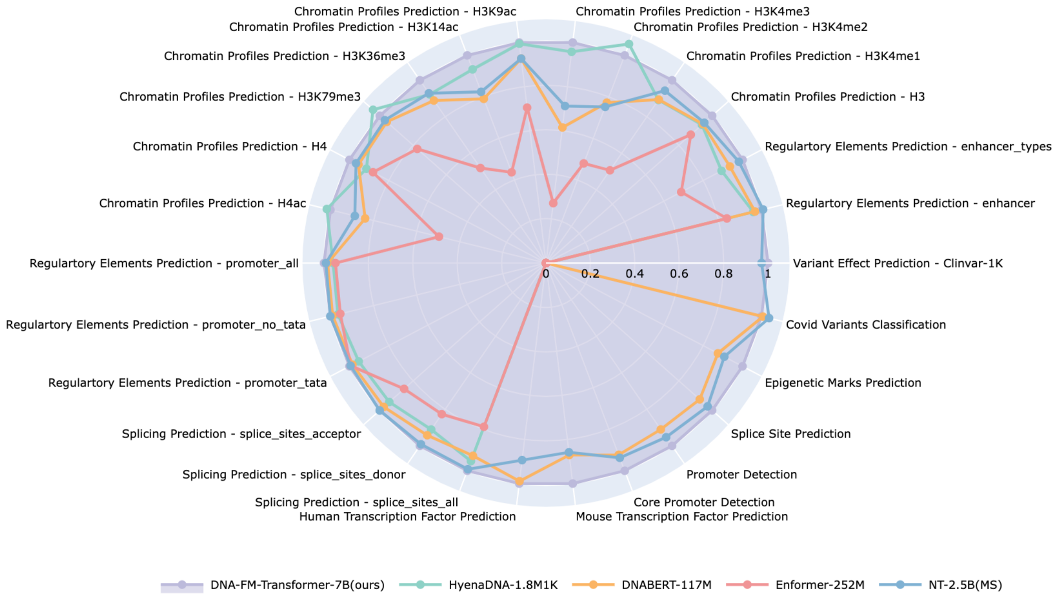


Figure 2: Benchmarking summary for AIDO.DNA, normalized to AIDO.DNA performance

7B model size during pretraining, and chunked our dataset of 796 genomes into non-overlapping segments.

Architecture To learn semantically meaningful representations, we employed an BERT-style encoder-only dense transformer architecture [29]. We make minor updates to this architecture to align with current best practices, including using SwiGLU [31] and LayerNorms [32]. Additionally, we use Rotary Positional Embeddings (RoPE) [33], given that DNA syntax does not function based on absolute nucleotide positions but nucleotides interact in highly local and context-specific ways. Our 300M model used 24 transformer blocks with embedding size 1,024, where each block contains 16 self-attention heads and a feed forward size 2,688. Our 7B model used 32 transformer blocks with embedding size 4,352, where each block contains 32 self-attention heads and feed forward size of 11,584 (Figure 1).

Training The weights of our seven billion parameter model occupy over 200GB of memory in 32 bit precision. To train a model of this size, we use model parallelism to split training across 256 H100 GPUs using the Megatron LM framework [27]. We also employed bfloat16 mixed precision training and FlashAttention-2 [34] to allow for training with large context length at scale. With these optimizations, our maximum possible global batch size was 1024 with a per-device micro batch size of 2. With this configuration, the model took 8 days to train to 100,000 iters. The 300 million parameter model trained for 4 days on 32 A100 GPUs. Full training configurations and practical considerations for both models are available in the Appendix D.1.

Optimization We trained AIDO.DNA using a classic masked language modeling (MLM) objective, choosing 15% of tokens in each sequence at random to alter. Of the chosen tokens, 80% are masked, 10% are corrupted (uniformly re-sampled), and 10% are untouched. We use a cross-entropy loss on these tokens, comparing with the original unaltered sequence. We used the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We also used a cosine learning rate scheduler with a 2% linear warmup and a minimum learning rate of 10^{-5}.

2.2 Evaluation

We evaluate the benefits of pretraining our 300M and 7B models by conducting a comprehensive series of experiments related to functional genomics, genome mining, metabolic engineering, synthetic biology, and therapeutics design, covering supervised, unsupervised, and generative objectives.

Unless otherwise stated, hyperparameters were determined by optimizing model performance on a 10% validation split of the training data, and models were tested using the checkpoint with the lowest validation loss.

Sequence Property Classification We apply the pretrained AIDO.DNA models to sequence classification and property prediction tasks, using standard classification benchmarks from prominent works on DNA encoders covering a breadth of genomic functions related to transcriptional regulation and transcript processing [10, 7]. We use a binary cross-entropy objective, full parameter finetuning with `[CLS]` pooling and a linear prediction head. For transcription factor classification tasks, we continued our unsupervised masked language modeling pretraining objective for several epochs on test set input sequences before applying the supervised finetuning objective, given recent work on the importance of well-aligned inductive biases in dense transformers [35].

Zero-shot Variant Effect Prediction To evaluate the utility of AIDO.DNA's pretrained embeddings in a zero-shot setting, we use a curated set of ClinVar pathogenic and GnomAD benign variant data to predict variant effects [10]. We inferred embeddings for each sequence variant by performing inference on a 1kbp window centered on the variant, then averaging embeddings across the sequence length. Assuming benign variants will produce similar embeddings to the reference while pathogenic variants will produce different embeddings, we quantify the AUROC of our model by taking the max-normalized L2 distances between reference and variant sequence embeddings as the probability of variant pathogenicity.

Dependency Mapping for Genome Mining We apply AIDO.DNA to discover unnannotated but functionally important regions of the human genome by implementing an *in silico* mutagenesis-based nucleotide dependency mapping strategy [30]. This method produces a 2D grid of dependency values between each nucleotide, indicating their co-conservation and likely functional dependency. The grid of predicted dependencies is composed of dependency pixels $e_{i,j}$ for each nucleotide pair i and j, where

$$e_{i,j} = \max_{k,q \in \{A,T,C,G\}} \left| \log_2 \left(\frac{\hat{\text{odds}}(n_j = k \mid n_1, ..., n_i = q, ..., n_L)}{\hat{\text{odds}}(n_j = k \mid n_1, ..., n_L)} \right) \right|,$$

such that k and q are the key and query nucleotide types, n is a length L DNA sequence, and $\hat{\text{odds}}$ are the odds inferred using the pretrained AIDO.DNA. Crucially, this method leverages the pretrained model without finetuning, allowing us to evaluate the richness of representations inferred from unsupervised learning alone.

Predicting Gene Expression from Genomic Contexts We propose a new variant of the canonical task of predicting gene expression from genomic contexts based on a dataset of 10 million synthetic promoter sequences with paired expression measurements [36]. This dataset enjoys desirable properties for downstream analysis and evaluation of models, well-characterized transcription factor-promoter interactions and random sequence construction, which affords useful analytical properties. We finetune AIDO.DNA using the same setup as classification with a mean squared-error objective to predict gene expression from sequence.

Directed Promoter Generation Applying AIDO.DNA to generate functional genetic code, we invert our gene expression prediction task to generate promoter sequences toward a desired expression level. MLM-trained models are not suitable for complex sequence generation, so we finetune AIDO.DNA with a simple masked diffusion language modeling objective [37]. To add the expression level condition, we append our pretrained model with 2 transformer blocks, whose input is the pretrained embeddings added to a linear expression level encoding. To allow our model to ignore conditioning information if it is uninformative of the data distribution and leverage the pretrained decoder head, we also add a skip connection from the pretrained embeddings to the output of the conditioned transformer blocks.

Table 2: Genome Understanding Evaluation Benchmarks [7]. Metrics are Matthew's Correlation Coefficient (MCC) unless otherwise stated. DB is DNABERT. NT is Nucleotide Transformer. †Uses self pre-training before supervised finetuning [35]

	DB	**DB-2 old**	**DB-2**	**NT**	**DNA FM**	**DNA FM**
Task	89M	117M	117M	2.5B	300M	7B
human-TF-0	66.84	71.99	69.12	66.64	68.192	**72.216**†
human-TF-1	70.14	76.06	71.87	70.28	70.656	**77.929**†
human-TF-2	61.03	66.52	62.96	58.72	70.987	**72.273**†
human-TF-3	51.89	**58.54**	55.35	51.65	47.214	55.129†
human-TF-4	70.97	**77.43**	74.94	69.43	75.003	76.879†
mouse-TF-0	44.42	56.76	64.23	63.31	56.71	**67.292**†
mouse-TF-1	78.94	84.77	86.28	83.76	82.447	**86.295**†
mouse-TF-2	71.44	79.32	81.28	71.52	84.729	**91.110**†
mouse-TF-3	44.89	66.47	73.49	69.44	77.246	**86.769**†
mouse-TF-4	42.48	52.66	50.8	47.07	47.848	**55.375**†
core_promoter_all	68.9	69.37	67.5	70.33	**72.977**	71.80
core_promoter_no_tata	70.47	68.04	69.53	71.58	**72.054**	71.623
core_promoter_tata	76.06	74.17	76.18	72.97	82.221	**85.045**
Promoter all	90.48	86.77	88.31	91.01	93.229	**93.532**
promoter no TATA	93.05	94.27	94.34	94	94.165	**94.955**
Promoter TATA	61.56	71.59	68.79	79.43	86.057	**89.104**
Splice Reconstruction	84.07	84.99	85.93	89.35	90.727	**91.16**
H3	73.1	78.27	80.17	78.77	/	**82.485**
H3K14ac	40.06	52.57	57.42	56.2	/	**64.927**
H3K36me3	47.25	56.88	61.9	61.99	/	**67.441**
H3K4me1	41.44	50.52	53	55.3	/	**57.826**
H3K4me2	32.27	31.13	39.89	36.49	/	**47.031**
H3K4me3	27.81	36.27	41.2	40.34	/	**44.788**
H3K79me3	61.17	67.39	65.46	64.7	/	**68.278**
H3K9ac	51.22	55.63	57.07	56.01	/	**64.17**
H4	79.26	80.71	81.86	81.67	/	**81.947**
H4ac	37.43	50.43	50.35	49.13	/	**61.2**
COVID (F1)	55.5	71.02	68.49	**73.04**	69.714	70.575
Average TF	64.174	70.108	66.848	63.344	66.41	**70.885**
Average Mouse TF	56.434	67.996	71.216	67.02	69.796	**77.368**
Average Core Promoter	71.81	70.527	71.07	71.627	75.751	**76.399**
Average Promoter_300	81.697	84.21	83.813	88.147	91.15	**92.53**
Average Histone	49.101	55.98	58.832	58.06	/	**64.009**
Average Overall	60.69	66.649	67.749	66.707	/	**73.334**

3 Results

We developed AIDO.DNA, the largest and most performant encoder-only foundation model to date for representation, transfer learning, and generation of DNA sequences. While long context models have dominated recent literature, AIDO.DNA shows that substantial gains can be made on most tasks by scaling model depth on a short context length of 4,000 tokens at single-nucleotide resolution. On suites of transfer learning benchmarks from recent works, AIDO.DNA achieves a new state-of-the-art on sequence property prediction and zero-shot variant effect prediction (Fig. 2). Furthermore, pretraining at scale reveals functional dependencies without the need for curated finetuning data, helping to define and annotate new regulatory elements (Fig. 3). Finally, we show that AIDO.DNA enables directed design of promoter sequences toward a desired expression level (Table 6).

3.1 Scaling DNA Encoders to Seven Billion Parameters

AIDO.DNA is the largest encoder-only DNA foundation model to date, with 7 billion parameters trained on 10.6 billion nucleotides from a multi-species dataset of 796 species. This multispecies dataset is nearly identical to the one used by Nucleotide Transformer [10], after removing genomes which have since been deleted from NCBI. In comparison with other encoder-only models Nucleotide

Table 3: Nucleotide Transformer Benchmarks (MCC) [10].

	HyenaDNA	**DB**	**DB-2**	**NT**	**NT v2**	**DNA FM**	**DNA FM**
Task	1.8M	89M	117M	2.5B	500M	300M	7B
Enhancer	52.059	49.559	52.5	54.559	55.9	49.47	**59.670**
E types	40.294	36.618	42.206	44.265	43.8	45.067	**45.113**
H3	78.088	76.324	78.529	79.265	78.6	79.929	**83.207**
H3K4me1	51.176	39.706	51.324	54.118	55	56.352	**57.39**
H3K4me2	**45.588**	28.382	33.382	32.5	32	36.295	43.229
H3K4me3	55	25.882	35.294	40.882	40.6	42.841	**57.42**
H3K9ac	58.529	50.588	54.559	54.559	56.7	57.808	**58.845**
H3K14ac	60.882	40.441	51.618	53.824	54.9	56.727	**65.296**
H3K36me3	61.324	47.353	59.118	61.765	62.4	64.739	**66.499**
H3K79me3	**66.765**	57.794	61.471	62.206	63	61.911	64.156
H4	76.324	78.382	79.706	80.735	79.9	82.883	**83.549**
H4ac	**56.324**	35.882	46.471	49.118	49.6	53.103	55.387
Promoter all	91.912	92.206	94.265	95	**97.6**	94.355	95.080
Prom no TATA	91.912	92.353	94.265	95.294	**97.6**	94.39	95.244
Prom TATA	87.941	91.029	90.882	91.765	**96.5**	90.801	92.308
Splice Acceptor	91.618	x	94.853	97.206	**98.1**	98.038	97.144
Splice Donor	89.412	x	92.5	97.353	**98.7**	97.142	98.216
Splice All	93.382	96.176	90.882	97.206	**98.4**	97.657	97.949
Average MCC	69.363	58.667	66.879	68.979	69.961	69.973	**72.83**

Transformer [10] and DNABERT [4] models, AIDO.DNA also makes only modest architecture changes. However, we find that the main limitation of such models is their size. By scaling DNA encoders to seven billion parameters, we see substantial gains across standard benchmarks, and notice a variety of improvements to qualitative tasks relevant to genome mining, metabolic engineering, and synthetic biology (Tables 2, 3, 4, 5, 6).

3.2 Scaling Improves Transfer Learning

AIDO.DNA enables high-accuracy recognition of functional genomic elements, revealing a new state-of-the-art on standard benchmarks covering human, mouse, and yeast genomics (Tables 2, 3). These benchmarks, proposed in [10, 11, 4, 7], primarily focus on transcriptional regulation and transcript processing in eukaryotes, which have complicated and sparsely-characterized regulatory grammars. In transfer learning, these accuracy improvements are driven by pretrained representations of DNA function, enabling data-efficient genome mining and experiment prioritization.

3.3 Fine-grained Tokenization Enables Accurate Variant Effect Prediction

Table 4: Zero-shot variant effect prediction using pretrained model embeddings.

	NT [10]	**AIDO.DNA**	**AIDO.DNA**
	2.5B	300M	7B
AUROC	.720	**.807**	.785

AIDO.DNA's single nucleotide tokenization improves recognition of variants with outsize effects on human health, establishing a new SOTA for single-sequence pretrained models (Table 4). While the 300M and 7B versions both exceed the previous SOTA by a wide margin, the smaller 300M model is most performant on this task. We reason that this is due to the smaller embedding size in the 300M model, while the 7B model may have more channels unrelated to variant pathogenicity, creating a lower signal-to-noise ratio on this task.

3.4 AIDO.DNA Learns Co-evolving Genome Elements Without Supervision

AIDO.DNA learns functional dependencies between DNA sequences without labeled data (Figure 3). Self-supervised pretraining is a powerful tool to infer conditional dependencies, which can be

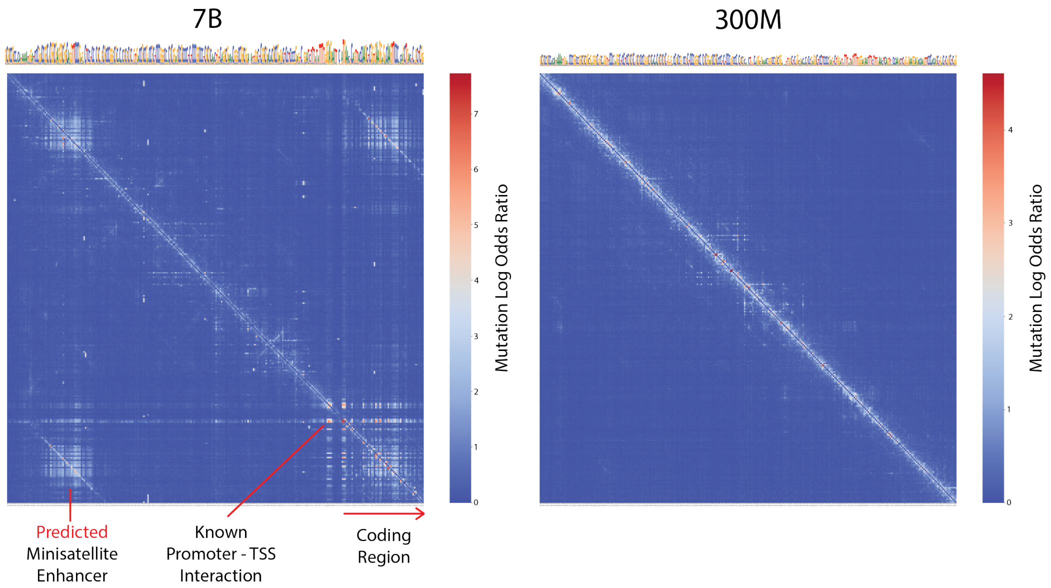


Figure 3: Unsupervised discovery of regulatory elements using *in silico* mutagenesis, applied to the 3' UTR and coding region of KIF26B (hg38.p14, chr1:4752-5051).

probed and cataloged through *in silico* mutagenesis. While *in silico* mutagenesis studies normally require $O(L^2)$ inferences using a supervised model such as Enformer [5] to compute all 2nd order mutation effects, self-supervised models infer the probability of all key mutations under a given query mutation at once, allowing us to compute this dependency mapping with only $O(L)$ inferences.

In our experiments, we find a rich landscape of both well-characterized and uncharacterized DNA-DNA interactions (Figure 3). In this example, we apply dependency mapping to the 3' UTR and coding region of KIF26B (hg38.p14, chr1:4752-5051). While originally known to contain a putative promoter, we also discover an unlabeled minisatellite with an imperfect repeat, mirroring part of the coding region. Such minisatellites are implicated as expression regulators, but exact mechanisms are unknown. Notably, these dependencies are inferred from pretraining alone and revealed only at the largest modeling scales. AIDO.DNA provides a strong foundation for genome mining experiments through predictive co-conservation and co-evolution, extending beyond MSA-based models and allowing *in silico* mutagenesis studies on any DNA sequence.

3.5 Pretraining Enables Sample-efficient Prediction of Expression from Genomic Contexts

Table 5: Predicting the activity of promoter sequences with decreasing training data volume, leveraging AIDO.DNA for transfer learning with data scarcity.

Train Samples	10M	1M	100k	10k	1k	100	10
Data %	100%	10%	1%	0.1%	0.01%	0.001%	0.0001%
MSE	.365	.390	.598	.604	.675	.869	.988
Pearson's ρ	.802	.789	.645	.641	.584	.408	.237

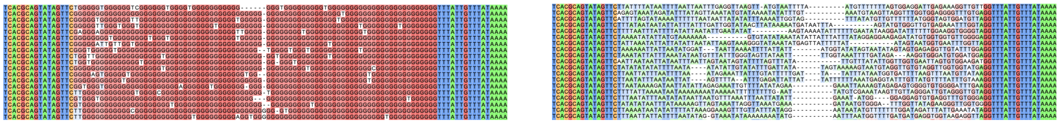
AIDO.DNA accurately and efficiently predicts gene expression from synthetic promoter sequences (Table 5). During transfer learning, AIDO.DNA exploits high-level functional features learned during pretraining, such as promoters, enhancers, transcription factor binding sites (Figure 3), to accurately predict gene expression from genomic contexts with limited labeled data. With just 0.01% of the data (1000 samples), AIDO.DNA retains 73% of the performance on the full 10M dataset. Only 0.001% of the data (100 samples), is required to retain 50% of the top performance.

Table 6: Accuracy of generated promoter sequences.

	AIDO.DNA
Unconditional Diffusion	0.460
Directed Diffusion	0.528

3.6 Directed Generation of Regulatory DNA Sequences

AIDO.DNA learns eukaryotic regulatory sequence grammar, enabling tunable generation of promoters toward a desired level of gene expression (Table 6). We invert the task of expression prediction from sequence, using the same dataset of 10 million random TATA promoter sequences with paired expression level measurements to adapt our model for conditional sequence generation [38]. The dataset contains only random synthetic sequences, allowing us to analytically characterize the performance of an optimal unconditional generation method. With a constant 16bp upstream scaffold and 19bp downstream scaffold on either side of the 80bp random sequence and few bp indels at the insertion sites, a perfect unconditional generator will achieve accuracy of 0.45-0.48. Values above this range require learning promoter grammars which correspond to specific expression levels. Our method, which adapts the pretrained AIDO.DNA with a lightweight 4.5M parameters conditional diffusion head and a simple masked diffusion language modeling objective[37], exceeds this crucial threshold, achieving 0.528 accuracy on per-base recovery when generating sequences *de novo*, using only the desired expression level condition (Table 6).



(a) 20 promoters generated for high expression

(b) 20 promoters generated for low expression

Figure 4: Directed generation of promoter sequences toward a desired expression level. Sequences are aligned and colored by nucleotide with Clustal Omega nucleotide to highlight similarities.

Alignment and visual inspection of generated sequences under the highest and lowest expression conditions reveals biologically meaningful motifs (Fig. 4). The promoters generated for the highest expression use almost exclusively G to connect the promoter scaffolds with several regions showing no deviation. Those generated for the lowest expression show notable G enrichment at the 5' end and a noisy alternating TATA motif throughout, possibly hindering expression by re-binding the polymerase after it is recruited to the TATA binding site in the 5' scaffold.

4 Discussion

The large feature space of genomes coupled with the relative scarcity of labeled data presents a difficult challenge for statistical modeling approaches, usually dubbed the *big p small n* regime. In this setting, the low information density of the feature space coupled with extreme sample scarcity requires high-bias-low-variance modeling approaches such as additivity or sparsity, or otherwise requires the use of prior knowledge or hand-picked feature sets, none of which are ideal for data-driven research. However, this issue is directly related to a pervasive one-model-one-task approach to research, where users apply curated task-specific data toward a single modeling objective, usually within the scope of a single organism or a single genome. This approach has the dual consequences of limiting the amount of data available for modeling, while also making learned features and effects unlikely to generalize at test-time to new contexts. Especially as heterogeneous and observational data have become more abundant, the one-model-one-task approach seems poorly suited for modern biology research.

Foundation models hint at a new generation of models beyond *big p small n* constraints. Rather than requiring models to learn higher order features and their effects under sample constraints, pretrained foundation models frontload semantic feature learning, and unsupervised models do this without paired or labeled data. To enabling accurate biological models with *big p small n* data, we develop

AIDO.DNA, the largest DNA encoder to date with seven billion parameters, which learns informative and transferable representations of DNA by pretraining at scale. We evaluate AIDO.DNA using a wide range of biologically and medically relevant tasks including genome mining and annotation (Tables 2, 3, Fig. 2), synthetic biology (Table 5), therapeutics design (Fig. 4), and disease diagnosis (Table 4). Indeed, we find that unsupervised pretraining learns surprisingly rich representations of DNA which has been mostly unexplored in previous works and only emerge with billions of parameters (Fig. 3).

Despite gaps in performance and utilization relative to protein language models, genome language models are essential for the advancement of biological models beyond the one-model-one-task setting. Genomes are a universal context for biological models, and context-adaptive models are already well-poised to leverage genome language models within both statistical modeling and deep learning workflows [39, 40, 41, 42, 43, 44, 45, 46]. AIDO.DNA marks a step toward relaxing restrictive model designs that are currently required for *big p small n* data, promising to enable accurate and personalized machine learning in biology and medicine, as well as more general and complex modeling approaches in the development of an AI-driven Digital Organism [1].

References

[1] Le Song, Eran Segal, and Eric Xing. Toward AI-Driven Digital Organism: Multiscale Foundation Models for Predicting, Simulating, and Programming Biology at All Levels . *Technical Report*, 2024.

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

[3] David R. Kelley. Cross-species regulatory sequence activity prediction. *PLoS Computational Biology*, 16(7):e1008050, July 2020.

[4] 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, August 2021.

[5] 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, October 2021. Number: 10 Publisher: Nature Publishing Group.

[6] Maxim Zvyagin, Alexander Brace, Kyle Hippe, Yuntian Deng, Bin Zhang, Cindy Orozco Bohorquez, Austin Clyde, Bharat Kale, Danilo Perez-Rivera, Heng Ma, Carla M. Mann, Michael Irvin, J. Gregory Pauloski, Logan Ward, Valerie Hayot-Sasson, Murali Emani, Sam Foreman, Zhen Xie, Diangen Lin, Maulik Shukla, Weili Nie, Josh Romero, Christian Dallago, Arash Vahdat, Chaowei Xiao, Thomas Gibbs, Ian Foster, James J. Davis, Michael E. Papka, Thomas Brettin, Rick Stevens, Anima Anandkumar, Venkatram Vishwanath, and Arvind Ramanathan. GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. *bioRxiv*, page 2022.10.10.511571, November 2022.

[7] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. DNABERT-2: Efficient Foundation Model and Benchmark For Multi-Species Genome, March 2024. arXiv:2306.15006 [cs, q-bio].

[8] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, Stefano Ermon, Stephen A. Baccus, and Chris Ré. HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution, November 2023. arXiv:2306.15794 [cs, q-bio].

[9] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, December 2023. arXiv:2312.00752 [cs].

[10] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim,

Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics, January 2023. Pages: 2023.01.11.523679 Section: New Results.

[11] 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, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics, September 2023. Pages: 2023.01.11.523679 Section: New Results.

[12] 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*, page 2023.10.10.561776, April 2024.

[13] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-Directional Equivariant Long-Range DNA Sequence Modeling, March 2024. arXiv:2403.03234 [cs, q-bio].

[14] 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, March 2024. Pages: 2024.02.27.582234 Section: New Results.

[15] Alexander Karollus, Johannes Hingerl, Dennis Gankin, Martin Grosshauser, Kristian Klemon, and Julien Gagneur. Species-aware DNA language models capture regulatory elements and their evolution. *Genome Biology*, 25(1):83, April 2024.

[16] Yunha Hwang, Andre L. Cornman, Elizabeth H. Kellogg, Sergey Ovchinnikov, and Peter R. Girguis. Genomic language model predicts protein co-regulation and function. *Nature Communications*, 15(1):2880, April 2024. Publisher: Nature Publishing Group.

[17] Yong He, Pan Fang, Yongtao Shan, Yuanfei Pan, Yanhong Wei, Yichang Chen, Yihao Chen, Yi Liu, Zhenyu Zeng, Zhan Zhou, Feng Zhu, Edward C. Holmes, Jieping Ye, Jun Li, Yuelong Shu, Mang Shi, and Zhaorong Li. LucaOne: Generalized Biological Foundation Model with Unified Nucleic Acid and Protein Language, May 2024. Pages: 2024.05.10.592927 Section: New Results.

[18] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. Number: 7873 Publisher: Nature Publishing Group.

[19] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Charlotte Rochereau, George M Church, Peter K Sorger, and Mohammed AlQuraishi. Single-sequence protein structure prediction using language models from deep learning. Publication Title: bioRxiv, August 2021.

[20] Nadav Brandes, Grant Goldman, Charlotte H. Wang, Chun Jimmie Ye, and Vasilis Ntranos. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, pages 1–11, August 2023. Publisher: Nature Publishing Group.

[21] Jun Cheng, Guido Novati, Joshua Pan, Clare Bycroft, Akvilė Žemgulytė, Taylor Applebaum, Alexander Pritzel, Lai Hong Wong, Michal Zielinski, Tobias Sargeant, Rosalia G. Schneider, Andrew W. Senior, John Jumper, Demis Hassabis, Pushmeet Kohli, and Žiga Avsec. Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science*, 381(6664):eadg7492, September 2023. Publisher: American Association for the Advancement of Science.

[22] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and Alexander Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction, July 2022. Pages: 2022.07.20.500902 Section: New Results.

[23] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model, July 2024. Pages: 2024.07.01.600583 Section: New Results.

[24] Ziqi Tang and Peter K. Koo. Evaluating the representational power of pre-trained DNA language models for regulatory genomics, March 2024. Pages: 2024.02.29.582810 Section: New Results.

[25] Shushan Toneyan, Ziqi Tang, and Peter K. Koo. Evaluating deep learning for predicting epigenomic profiles, May 2022. Pages: 2022.04.29.490059 Section: New Results.

[26] Ziqi Tang, Shushan Toneyan, and Peter K. Koo. Current approaches to genomic deep learning struggle to fully capture human genetic variation. *Nature Genetics*, 55(12):2021–2022, December 2023. Publisher: Nature Publishing Group.

[27] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism, March 2020. arXiv:1909.08053 [cs].

[28] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. arXiv:2106.09685 [cs].

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. arXiv:1810.04805 [cs].

[30] Pedro Tomaz Da Silva, Alexander Karollus, Johannes Hingerl, Gihanna Galindez, Nils Wagner, Xavier Hernandez-Alias, Danny Incarnato, and Julien Gagneur. Nucleotide dependency analysis of DNA language models reveals genomic functional elements, July 2024.

[31] Noam Shazeer. GLU Variants Improve Transformer, February 2020. arXiv:2002.05202 [cs, stat].

[32] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On Layer Normalization in the Transformer Architecture, June 2020. arXiv:2002.04745 [cs, stat].

[33] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, November 2023. arXiv:2104.09864 [cs].

[34] Tri Dao. Faster Attention with Better Parallelism and Work Partitioning.

[35] Ido Amos, Jonathan Berant, and Ankit Gupta. Never Train from Scratch: Fair Comparison of Long-Sequence Models Requires Data-Driven Priors, April 2024. arXiv:2310.02980 [cs].

[36] Carl G. de Boer, Eeshit Dhaval Vaishnav, Ronen Sadeh, Esteban Luis Abeyta, Nir Friedman, and Aviv Regev. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nature Biotechnology*, 38(1):56–65, January 2020. Publisher: Nature Publishing Group.

[37] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and Effective Masked Diffusion Language Models, June 2024. arXiv:2406.07524 [cs].

[38] Richard Leslie, Christopher J. O'Donnell, and Andrew D. Johnson. GRASP: analysis of genotype-phenotype results from 1390 genome-wide association studies and corresponding open access database. *Bioinformatics (Oxford, England)*, 30(12):i185–194, June 2014.

[39] Caleb N. Ellington, Benjamin J. Lengerich, Thomas BK Watkins, Jiekun Yang, Hanxi Xiao, Manolis Kellis, and Eric P. Xing. Contextualized Networks Reveal Heterogeneous Transcriptomic Regulation in Tumors at Sample-Specific Resolution, December 2023. Pages: 2023.12.01.569658 Section: New Results.

[40] Benjamin Lengerich, Caleb N. Ellington, Andrea Rubbi, Manolis Kellis, and Eric P. Xing. Contextualized Machine Learning, October 2023. arXiv:2310.11340 [cs, stat].

[41] Caleb N. Ellington, Benjamin J. Lengerich, Wesley Lo, Aaron Alvarez, Andrea Rubbi, Manolis Kellis, and Eric P. Xing. Contextualized: Heterogeneous Modeling Toolbox. *Journal of Open Source Software*, 9(97):6469, May 2024.

[42] Maruan Al-Shedivat, Avinava Dubey, and Eric P. Xing. Contextual Explanation Networks, September 2020. arXiv:1705.10301 [cs, stat].

[43] Benjamin Lengerich, Bryon Aragam, and Eric P. Xing. Learning Sample-Specific Models with Low-Rank Personalized Regression, October 2019. arXiv:1910.06939 [cs, stat].

[44] Ben Lengerich, Caleb Ellington, Bryon Aragam, Eric P. Xing, and Manolis Kellis. NOTMAD: Estimating Bayesian Networks with Sample-Specific Structures and Parameters, November 2021. arXiv:2111.01104 [cs, stat].

[45] Benjamin J Lengerich, Mark E Nunnally, Yin Aphinyanaphongs, Caleb Ellington, and Rich Caruana. Automated Interpretable Discovery of Heterogeneous Treatment Effectiveness: A COVID-19 Case Study. *J. Biomed. Inform.*, page 104086, April 2022.

[46] Benjamin J. Lengerich, Maruan Al-Shedivat, Amir Alavi, Jennifer Williams, Sami Labbaki, and Eric P. Xing. Discriminative Subtyping of Lung Cancers from Histopathology Images via Contextual Deep Learning, November 2022. ISSN: 2014-0053 Pages: 2020.06.25.20140053.

[47] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only, June 2023. arXiv:2306.01116 [cs].

A Data and Code Availability

B Data and Code Availability

We developed the ModelGenerator package to reproduce, apply, and extend the results in this manuscript `https://github.com/genbio-ai/ModelGenerator`.

Pre-trained models and data splits are also available on Huggingface at `https://huggingface.co/genbio-ai`.

C Pre-training Data

Our pre-training dataset is based on the first version of the multi-species dataset from [10].

Our only modifications were to remove genomes which have since been deleted from NCBI, and to replace the deleted *Rattus norvegicus* genome with its current reference genome. We used the included dataloader in this repository, modified to chunk and save non-overlapping 4kbp sequences.

Notably, we found that in the splits from [10] were highly correlated with taxonomy, with the vast majority of the training data being comprised of bacteria, fungi, and invertebrates. In preliminary experiments on reshuffled splits where training included mammals, vertebrates, and popular model organisms, we saw a dramatic decrease in performance as more eukaryotes were added to the training data. While outside the scope of this study, we hypothesize that low-quality, noisy data, such as those common in intergenic eukaryotic sequences, are particularly detrimental to unsupervised model performance. We also note the current shift in natural language processing to small, high-quality datasets [47]. Although prokaryotes make up the majority of the training set and have drastically different genomes, these genomes have relatively little noise or junk. We reason it may be helpful for DNA language models to learn the basic dependency structures of conservation and co-conservation on these highly constrained genomes, improving performance on eukaryotic data despite the dramatic distribution shift.

D Pre-training Configs

D.1 AIDO.DNA 7B

```
{
    "os": "Linux-5.15.0-1013-gcp-tcpx-x86_64-with-glibc2.35",
    "python": "3.10.12",
    "docker": null,
    "cuda": null,
    "args": [
        "--num-layers",
        "32",
        "--hidden-size",
        "4352",
        "--num-attention-heads",
        "32",
        "--seq-length",
        "4000",
        "--max-position-embeddings",
        "4000",
        "--micro-batch-size",
        "2",
        "--global-batch-size",
        "1024",
        "--lr",
        "1e-4",
        "--train-iters",
        "100000",
        "--lr-decay-iters",
        "100000",
        "--lr-decay-style",
        "cosine",
        "--min-lr",
        "1e-5",
        "--weight-decay",
        "2e-2",
        "--lr-warmup-iters",
        "2000",
        "--clip-grad",
        "1.0",
        "--split",
        "9998,1,1",
        "--vocab-file",
        "dna_vocab.txt",
        "--log-interval",
        "1",
        "--seed",
        "42",
        "--save-interval",
        "500",
        "--eval-interval",
        "500",
        "--eval-iters",
        "10",
        "--distributed-backend",
        "nccl",
        "--dataloader-type",
        "cyclic",
        "--num-workers",
        "64",
        "--make-vocab-size-divisible-by",
        "16",
        "--bert-no-binary-head",
        "--mask-prob=0.15",
        "--swiglu",
        "--distributed-timeout-minutes",
        "600",
        "--attention-dropout=0.1",
        "--hidden-dropout=0.1",
        "--tensorboard-queue-size",
        "80",
        "--log-timers-to-tensorboard",
        "--log-memory-to-tensorboard",
        "--log-params-norm",
        "--adam-beta1",
        "0.9",
        "--adam-beta2",
        "0.95",
        "--no-position-embedding",
        "--use-rotary-position-embeddings",
        "--normalization=LayerNorm",
        "--use-flash-attn",
        "--overlap-grad-reduce",
        "--transformer-impl",
        "local",
        "--bf16",
        "--distributed-timeout-minutes",
        "600",
        "--pipeline-model-parallel-size",
        "8",
    ],
    // Note below is the resources of one node.
    "cpu_count": 104,
    "cpu_count_logical": 208,
    "disk": {
        "/": {
            "total": 5949.60994720459,
            "used": 40.23439025878906
        }
    },
    "gpu": "NVIDIA H100 80GB HBM3",
    "gpu_count": 8,
    "gpu_devices": [
        {
            "name": "NVIDIA H100 80GB HBM3",
            "memory_total": 85520809984
        },
        ...
    ],
    "memory": {
        "total": 1842.574909210205
    }
}
```

D.2 AIDO.DNA 300M

```
{
    "os": "Linux-5.15.0-60-generic-x86_64-with-glibc2.35",
    "python": "3.10.12",
    "docker": null,
    "cuda": null,
    "args": [
        "--num-layers",
        "24",
        "--hidden-size",
        "1024",
        "--num-attention-heads",
        "16",
        "--seq-length",
        "4000",
        "--max-position-embeddings",
        "4000",
        "--micro-batch-size",
        "4",
        "--global-batch-size",
        "256",
        "--lr",
        "3e-4",
        "--train-iters",
        "100000",
        "--lr-decay-iters",
        "100000",
        "--lr-decay-style",
        "linear",
        "--min-lr",
        "1.5e-5",
        "--weight-decay",
        "2e-2",
        "--lr-warmup-iters",
        "2000",
        "--clip-grad",
        "1.0",
        "--split",
        "9998,1,1",
        "--vocab-file",
        "dna_vocab.txt",
        "--log-interval",
        "1",
        "--seed",
        "42",
        "--save-interval",
        "500",
        "--eval-interval",
        "500",
        "--eval-iters",
        "10",
        "--distributed-backend",
        "nccl",
        "--dataloader-type",
        "cyclic",
        "--num-workers",
        "64",
        "--make-vocab-size-divisible-by",
        "16",
        "--bert-no-binary-head",
        "--mask-prob=0.15",
        "--swiglu",
        "--distributed-timeout-minutes",
        "600",
        "--attention-dropout=0.1",
        "--hidden-dropout=0.1",
        "--tensorboard-queue-size",
        "80",
        "--log-timers-to-tensorboard",
        "--log-memory-to-tensorboard",
        "--log-params-norm",
        "--adam-beta1",
        "0.9",
        "--adam-beta2",
        "0.95",
        "--no-position-embedding",
        "--use-rotary-position-embeddings",
        "--normalization=LayerNorm",
        "--use-flash-attn",
        "--overlap-grad-reduce",
        "--transformer-impl",
        "local",
        "--bf16",
        "--distributed-timeout-minutes",
        "600",
    ],
    // Note below is the resources of one node.
    "cpu_count": 64,
    "cpu_count_logical": 64,
    "disk": {
        "/": {
            "total": 0.015625,
            "used": 1.1444091796875e-05
        }
    },
    "gpu": "NVIDIA A100-SXM4-80GB",
    "gpu_count": 4,
    "gpu_devices": [
        {
            "name": "NVIDIA A100-SXM4-80GB",
            "memory_total": 85899345920
        },
        ...
    ],
    "memory": {
        "total": 503.22465896606445
    }
}
```