Genome Biology

RESEARCH

Open Access

Evaluating the representational power of pre-trained DNA language models for regulatory genomics

Ziqi Tang[1], Nirali Somia[1], Yiyang Yu[2] and Peter K. Koo[1*]

*Correspondence:
koo@cshl.edu

[1] Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA
[2] The Fu Foundation School of Engineering and Applied Science, Columbia University, New York, NY, USA

Abstract

Background: The emergence of genomic language models (gLMs) offers an unsupervised approach to learning a wide diversity of cis-regulatory patterns in the non-coding genome without requiring labels of functional activity generated by wet-lab experiments. Previous evaluations have shown that pre-trained gLMs can be leveraged to improve predictive performance across a broad range of regulatory genomics tasks, albeit using relatively simple benchmark datasets and baseline models. Since the gLMs in these studies were tested upon fine-tuning their weights for each downstream task, determining whether gLM representations embody a foundational understanding of cis-regulatory biology remains an open question.

Results: Here, we evaluate the representational power of pre-trained gLMs to predict and interpret cell-type-specific functional genomics data that span DNA and RNA regulation for six major functional genomics prediction tasks. Our findings suggest that probing the representations of current pre-trained gLMs do not offer substantial advantages over conventional machine learning approaches that use one-hot encoded sequences. Nevertheless, highly tuned supervised models trained from scratch using one-hot encoded sequences can achieve performance competitive with or better than pre-trained models across the datasets explored in this study.

Discussion: This work highlights a major gap with current gLMs, raising potential issues in conventional pre-training strategies for the non-coding genome.

Keywords: Deep learning, DNA language model, Regulatory genomics

Background

Large language models (LLMs) have demonstrated remarkable capabilities in natural language processing [1–4] and protein sequence analysis [5–8]. These LLMs, often termed "foundation models," are trained through self-supervised learning to encode input data as contextual embeddings (also known as representations). The strength of pre-trained LLMs lies in the versatility of their embeddings, which can be leveraged for a

broad spectrum of downstream predictive tasks. For instance, representations from pre-trained protein language models have been used to predict protein structures [9–11], predict non-synonymous variant effects [12, 13], design novel protein sequences [14–16], and study protein evolution [17, 18].

LLMs pre-trained on genomic DNA sequences offer a promising paradigm to accelerate our understanding of functional elements in the non-coding genome [19]. Genomic language models (gLMs) could, in principle, aid in deciphering the complex coordination of transcription factors (TFs) that control the activity of cis-regulatory elements (CREs). They may also enable more accurate predictions of the functional consequences of non-coding mutations, which can help to prioritize disease-associated variants. Furthermore, gLMs that are capable of learning cis-regulatory rules could play a key role in the design of novel regulatory sequences with desirable functional properties. In addition, such models may support functional comparisons of non-coding sequences across species, a task that remains challenging due to substantial evolutionary drift in non-coding regions.

Recently, there has been a surge of pre-trained gLMs [20–49]. gLMs take as input DNA sequences that have undergone tokenization, an encoding scheme applied to either a single nucleotide or *k*-mer of nucleotides. Through self-supervised pre-training, the gLM learns a vector representation for each token in the DNA sequence via masked language modeling (MLM) [1] or causal language modeling (CLM) [50]. In masked language modeling (MLM), a subset of input tokens undergo masking: most are replaced by a special `[MASK]` token, some by random tokens, and others left unchanged. The model learns to predict the original `[MASK]` tokens leveraging context from other unmasked positions. Various masking strategies explore different granularities (words, phrases, entities) and approaches (permutation sampling, importance-based selection, random replacements) to enhance the self-supervised pre-training task's effectiveness [51–55]. On the other hand, CLM is an autoregressive pre-training task with the goal of predicting the next token in a sequence given the previous tokens. Both language modeling objectives result in learning self-supervised representations of input sequences that capture information about individual tokens and the complex interrelationships with other tokens.

The burden of learning biologically meaningful features is paid upfront during the pre-training. Afterward, the gLM's representations can be leveraged for a broad spectrum of downstream prediction tasks as inputs to simpler models, bypassing the need to learn essential features for each task from scratch. In contrast, the conventional one-hot representation of DNA sequences treats each element independently, assigning an identical representation for the same nucleotide character irrespective of their position in the sequence or what context is nearby. Consequently, the responsibility of learning important patterns and their dependencies falls solely on the machine learning model being employed.

Current gLMs are composed of different choices for the tokenization, base architecture, language modeling objective, and pre-training data (Additional file 1: Table S1). *Tokenization* of DNA sequences is employed for either single nucleotide [20–22] or *k*-mer of fixed size [23–25] or a *k*-mer of variable sizes via byte-pair tokenization [26, 27, 33, 56], which aims to aggregate DNA in a manner that reduces the *k*-mer bias in the

genome, a problem known as rare token imbalance. The *base architecture* is typically a stack of transformer layers [57], with a vanilla multi-head self-attention [23–25, 27–31] or an efficient variant (e.g., flash attention [26, 58]) or an exotic attention variant (e.g., sparse attention [32, 33]). Alternatively, the base architecture has also been constructed with a stack of residual-connected dilated convolutional layers [20] or selective state-space models, such as a Hyena [21, 22, 59] or Mamba [43, 46]. The *pre-training data* can vary significantly, encompassing the whole genome of a single species [20, 24, 32] or the whole genomes across multiple species [23, 25, 26, 28, 33] or focused only within specific regions of the genomes, such as the untranslated regions (UTRs) [29], pre-mRNA [30], non-coding RNA [60], promoters [22], coding regions [35, 36, 45], non-coding RNA [39, 60], or conserved sites [34].

Notably, Nucleotide Transformer [23] is a collection of BERT-style models [1] that consider non-overlapping *k*-mer tokenization and is pre-trained via MLM on either a single human genome, a collection of 3202 human genomes from the 1000 Genomes Project [61] alone or in combination with 850 genomes across diverse species. DNABERT2 [26] is also a BERT-style architecture but uses flash attention, considers byte-pair tokenization, and is trained via MLM on the genomes of 850 species. Genomic Pre-trained Network (GPN) is a convolution-based model with a stack of residual-connected dilated convolutions, uses single-nucleotide tokenization, and is trained via MLM on *Arabidopsis thaliana* genome and seven related species within the Brassicales order [20]. Similarly, HyenaDNA [21] is a selective state-space model using Hyena layers and single-nucleotide tokenization and is trained via CLM on the human reference genome.

The utility of gLMs pre-trained on whole genomes for studying the non-coding genome has been limited. Previous benchmarks have largely considered gLMs that have been fine-tuned on each downstream task [23, 24, 26, 30, 39]. gLM fine-tuning involves adjusting the weights of all layers or through parameter efficient fine-tuning methods, such as LoRA (Low-Rank Adaptation) [26, 62, 63], (hard or soft) prompt tuning [21, 64], and $(IA)^3$ [23, 65]. In each benchmark, a fine-tuned gLM has demonstrated improved predictions on a host of downstream prediction tasks, often based on the classification of functional elements, such as histone marks or promoter annotations. However, the chosen benchmarks do not reflect the complexity of *cis*-regulatory mechanisms observed in gene regulation, and the baseline models used in the comparisons often do not represent the state-of-the-art. Hence, the capabilities of gLMs in understanding the regulatory genome have yet to be demonstrated in a fair assessment.

However, fine-tuning makes it challenging to assess the contribution of the prior knowledge gained via pre-training on each downstream task. Moreover, benchmarks that do not fine-tune gLMs are limited in their downstream tasks [66–68], relying on either binary classification of functional activity, which does not reflect the complexity of *cis*-regulatory biology [69, 70] or lack a more comprehensive set of benchmarking tasks. Thus, the extent to which existing gLMs pre-trained on whole genomes can genuinely serve as foundation models that can transfer their knowledge to predict and interpret functional genomics data without necessitating additional fine-tuning of the gLM weights.

Here we perform a focused evaluation to assess the informativeness of learned representations of various gLMs pre-trained on whole genomes (without fine-tuning any

existing layers) for six major functional genomics prediction tasks, which encompass different levels of cell type-specific *cis*-regulation complexity at DNA and RNA levels (see Fig. 1). In particular, we compared the predictive power via probing representations from pre-trained gLMs—namely Nucleotide Transformer, DNABERT2, HyenaDNA, and a custom GPN pre-trained on the human reference genome—versus one-hot encoded DNA and representations acquired from a supervised "foundation model" pre-trained on a large corpus of functional genomics data.

Our results suggest that current gLMs pre-trained on whole genomes do not provide noticeable advantages over conventional approaches for analyzing human functional genomics data, namely deep neural networks that consider one-hot sequences. By contrast, supervised foundation models pre-trained on functional genomics data appear to encapsulate more relevant information and their representations transfer better to other functional genomics data, albeit when the source pre-training tasks and the target tasks are closely aligned. Nevertheless, highly tuned supervised models trained from scratch using one-hot encoded sequences can achieve performance competitive with or better than pre-trained models across the datasets explored in this study. Our results suggest that current gLMs struggle to understand cell-type specific functional elements during pre-training and, therefore, fall short of recognition as a foundation model for the regulatory regions of the non-coding human genome.

Results and discussion

Task 1: predicting cell-type specific regulatory activity from lentiMPRA data

Understanding the mechanisms that drive CRE activity is a major goal in functional genomics; it is challenging due to complex rules of cell-type-specific TF binding [71, 72]. In the first task, we compared the performance of various machine learning models that consider different input representations of DNA sequences at predicting experimentally measured enhancer activity via lentiMPRA (lentiviral Massively Parallel Reporter Assay) [73]. Specifically, this task involves taking a 230 nucleotide (nt) DNA sequence as input, represented either as a gLM embedding or one-hot sequence, and predicting a scalar value that represents the CRE's functional activity measured in a given cellular context via lentiMPRA (see Methods section). This task enables a direct comparison in performance across matched downstream models for each sequence encoding scheme. By considering two cell types, namely HepG2 and K562, we can assess whether pre-trained

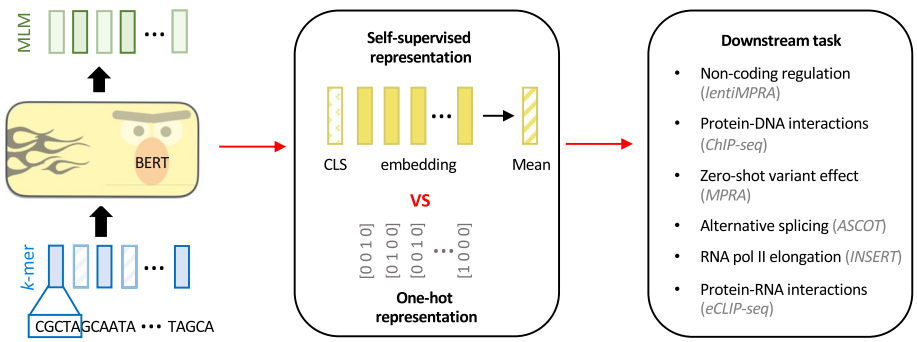

Fig. 1 Experimental overview. Comparison of gLM embeddings versus one-hot representations for various functional genomics prediction tasks

gLM representations capture cell-type-specific CRE activity. While the original lentiMPRA study included the WCT11 cell type, we excluded it from our analysis due to the lack of correspondence with the cell types used in Task 3's zero-shot single-nucleotide variant effect generalization.

For each gLM, we probed the embeddings from the penultimate layer using a linear model or multi-layer perceptron (MLP) based on the classification token (CLS) or the mean embedding, which is standard practice for harnessing sequence summarization of LLM embeddings. We also employed a baseline convolutional neural network (CNN) that analyzed the full embeddings of the penultimate layer as well as one-hot sequences for comparison (see Methods section). We also considered embeddings from the penultimate layer of Sei [74], a supervised foundation model pre-trained on 21,907 chromatin profiling datasets across over 1300 cell lines and tissues. As a supervised baseline, we included MPRAnn, which was presented in the original study [73]. To assess the performance against a more sophisticated supervised model, we trained a ResidualBind-like model (ResNet) using one-hot sequences [75]. These choices provide a fair benchmark to assess whether embeddings from foundation models, acquired via unsupervised gLMs or supervised CNNs, are more informative for downstream models than naive one-hot sequences.

We found that a CNN trained on the whole sequence embedding led to improved performance over the linear or MLP models that analyzed CLS or mean embeddings (Fig. 2a). This suggests that summarized gLM representations lack sufficient information to predict cell-type-specific regulatory activity. In contrast, CNNs can build upon

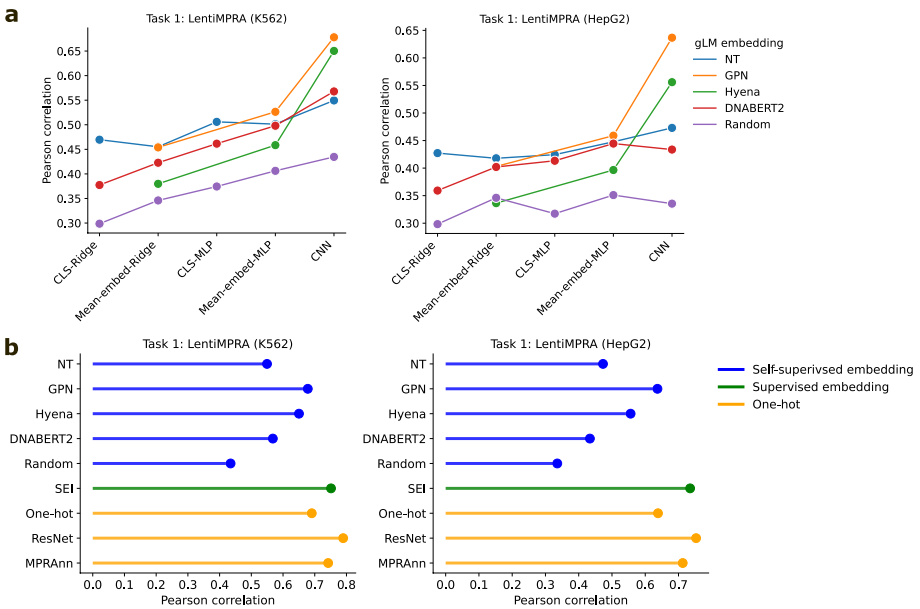

Fig. 2 Performance comparison on cell-type-specific regulatory activity prediction tasks from lentiMPRA data. **a** Comparison of predictive performance across various downstream machine learning models, including ridge regression and MLP using either the gLM's CLS token or mean embedding, and a CNN trained using the full embedding of the penultimate layer of gLMs. **b** Predictive performance using a baseline CNN trained using different gLM embedding inputs, one-hot sequences, or supervised embeddings from Sei. MPRAnn and ResNet represent the performance of more sophisticated models that are trained using one-hot sequences

the full embeddings to better discriminate cell-type specific features. Moreover, the performance gap between MLPs and linear models suggests that the mapping between the pre-trained representations and the functional readouts of lentiMPRA data is highly non-linear. While a small-scale hyperparameter grid search showed comparable performance across different model capacity sizes (Additional file 1: Fig. S1), a more comprehensive architecture and hyperparameter search could potentially identify model settings that lead to further performance gains. However, for the scope of this study, we focused on simpler models—as is standard practice—in order to primarily assess the out-of-the-box utility of the learned gLM representations.

In a broader comparison, we also observed that CNNs trained using sequence embeddings from gLMs generally under-performed standard one-hot sequences, except our custom-trained GPN (Fig. 2b). Notably, the performance of all gLM-based representations was significantly lower than the supervised representations given by Sei and a LASSO regression baseline using features based on Enformer's [76] predictions, similar to Ref. [73] (Additional file 1: Table S2). Due to differences in the data splits for Sei and Enformer, it is unclear to what extent data leakage might lead to performance inflation. Nevertheless, the ResNet model, which was trained from scratch using one-hot sequences from the LentiMPRA dataset, achieved the best performance (Fig. 2b). Fine-tuning the weights of the gLMs to directly predict the lentiMPRA data considerably improved their predictive performance, achieving comparable performance as ResNet (Additional file 1: Table S2). Together, these results suggest that pre-trained gLM embeddings may not provide beneficial context for CREs that cannot already be learned from one-hot sequences for the lentiMPRA dataset.

To control for the possibility that gLM embeddings from the penultimate layer may not be optimal, we performed the same analysis using embeddings from other layers of Nucleotide Transformer. While some layers yielded modest improvements, particularly layer 10, the overall trends held and thus did not change the conclusions (Additional file 1: Fig. S2).

Task 2: predicting TF binding sites from ChIP-seq data

Since TF binding is a cell-type-specific phenomenon, but standard language modeling objectives are not cell-type aware, we surmised that the low performance of gLMs on the lentiMPRA prediction task may be due to losing information about key motifs during the pre-training. To test this hypothesis, we evaluated whether the gLM embeddings can predict cell-type-specific TF binding sites measured via ChIP-seq (Chromatin Immunoprecipitation sequencing [77]). Briefly, this task is framed as a binary classification where a model takes a 200 nt DNA sequence, represented either as a gLM embedding or a one-hot sequence, as input and predicts whether the sequence corresponds to a ChIP-seq peak. We consider ten ChIP-seq datasets spanning different TFs in GM12878 cells; a separate single-task model was trained for each TF (see Methods section).

Evidently, CNNs trained using one-hot sequences modestly outperformed the whole embeddings from DNABERT2, HyenaDNA, and Nucleotide Transformer. On the other hand, the custom GPN occasionally led to improved performance (Fig. 3). Since the TF binding tasks were included in the original pre-training of Sei, data leakage might lead to Sei's inflated performance. Nevertheless, the modest performance

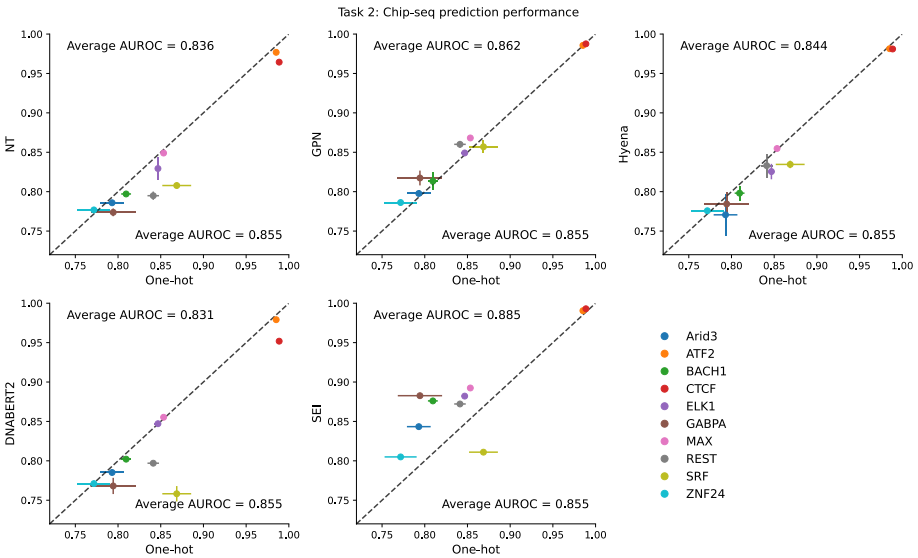

Fig. 3 Performance comparison on TF binding prediction tasks from ChIP-seq data. Comparisons of CNNs trained using different gLM embeddings versus CNNs trained using one-hot sequences for 10 TF ChIP-seq datasets. Performance is measured by the average area-under the receiver-operating characteristic curve (AUROC) and error bars represent the standard deviation of the mean across 5 different random initializations. Average AUROC represents the average performance across all ChIP-seq datasets

differences across sequence encoding schemes, including the similar or better performance of one-hot encoding compared to gLM embeddings, suggest that the gLM embeddings are likely not actively encoding explicit information about TF motifs. Rather, the embeddings appear to retain the essential sequence information necessary for downstream models like CNNs to learn TF binding patterns, akin to how CNNs can learn from one-hot encoded sequences that do not contain any inherent TF-related information.

As a control experiment, we trained MLP or linear models using the CLS token of Nucleotide Transformer. CLS tokens should fully encode a summary of the sequence semantics, which in the gLM case means it should contain information about TF motifs. We observed that CNNs trained on the whole embedding yielded substantially higher performance than an MLP trained using the CLS token (Additional file 1: Fig. S3a). However, the MLP still demonstrated proficiency in predicting TF binding overall. To further validate our findings and rule out the possibility of dataset biases creating a trivial prediction task, we also trained an MLP model on bag-of-dinucleotide frequencies. Indeed, the MLP based on dinucleotide frequencies yielded comparable performance to the gLM-derived CLS token (Additional file 1: Fig. S3a), except for CTCF, a broadly acting protein involved in chromatin organization across cell types. These results suggest that gLMs may not encode strong TF-related information in their embeddings. While not definitive, the CLS token appears to be only marginally more informative than low-level dinucleotide statistics. Importantly, CNNs are capable of learning motif features directly from sequence without relying on pretrained gLM representations, consistently achieving higher performance. Similarly, CNNs can extract relevant motif patterns when trained on the full gLM embeddings—something they could not achieve when restricted to the CLS token alone.

Task 3: zero-shot variant effect prediction with MPRA data

A major use case of highly accurate sequence-function models is their ability to predict the functional consequences of non-coding mutations [76]. In previous studies, Nucleotide Transformer and GPN have demonstrated an ability to predict single-nucleotide variant effects, albeit as part of a binary classification task [20, 23]. However, it is not intuitive how gLMs pre-trained on whole genomes could yield good zero-shot predictions of cell-type-specific variant effects in the non-coding region of human genomes since they are trained without any cell-type information. Thus, we assessed the ability of gLMs, specifically Nucleotide Transformer, GPN, and HyenaDNA, to quantitatively predict single-nucleotide variant effects within CREs using saturation mutagenesis data measured via MPRAs (Massively Parallel Reporter Assay) [78]. This task involves calculating the zero-shot variant effect predictions of gLMs either by the cosine similarity of embedding vectors for the input sequence with mutant or wild-type allele (e.g., Nucleotide Transformer and Hyena) or the log2-ratio of predicted variant and wild-type nucleotide via single-nucleotide masking (e.g., GPN). These variant effect scores are compared with experimentally measured variant effects according to the Pearson correlation coefficient (see Methods section). This analysis includes MPRA measurements for three CREs in HepG2 cells and one CRE in K562 cells as part of the CAGI5 challenge [78, 79].

We found that all tested gLMs (without fine-tuning) exhibited poor variant effect predictions in this quantitative zero-shot single-nucleotide generalization task (Table 1). These results extended to all Nucleotide Transformer models [23], including a 2.5 billion parameter BERT-based gLM trained on 3202 diverse human genomes and 850 genomes from various species. On the other hand, CNNs trained using lentiMPRA data based on gLM embeddings yielded substantially better performance relative to their pre-trained counterparts (Table 1). Moreover, gLMs that were fine-tuned on the lentiMPRA data

Table 1 Zero-shot variant effect generalization on CAGI5 dataset

Training task	Model	HepG2	K562
Self-supervised pre-training	NT (2B51000G)	0.125	0.007
	NT (2B5Species)	0.112	0.135
	NT (500MHuman)	0.020	0.088
	NT (500M1000G)	0.041	0.068
	GPN (human)	0.002	0.037
	HyenaDNA	0.064	0.021
LentiMPRA-embedding	CNN-GPN	0.332	0.437
	CNN-NT	0.185	0.198
	CNN-SEI	0.579	0.701
LentiMPRA-one-hot	CNN	0.324	0.365
	Residualbind	0.485	0.601
	MPRAnn	0.381	0.437
Supervised one-hot	SEI	0.545	0.641
	Enformer (DNase)	0.510	0.685

The values represent the Pearson correlation between the variant effect predictions and experimental saturation mutagenesis values of a given CRE measured via MPRAs. Values are reported for a single CRE experiment for K562 and the average of three CRE experiments for HepG2

also yielded improved performance (Additional file 1: Table S3). In contrast, sophisticated supervised models trained using one-hot sequences, such as Enformer [76], which is a state-of-the-art model trained with supervised learning on a wide variety of functional genomics data using one-hot sequences, and Sei yielded better performance than all CNNs trained using gLM representations. However, a CNN trained using Sei embeddings on the lentiMPRA dataset yielded the best overall performance. Together, these results highlight a major gap in the zero-shot variant effect performance of gLMs with the state-of-the-art.

Task 4: predicting alternative splicing from RNA-seq data

Previous studies demonstrated that Nucleotide Transformer and GPN have learned properties related to gene definition and splice sites [20, 23]. Thus, we surmised that gLMs pre-trained on whole genomes might be more beneficial for RNA regulation tasks. To investigate this, we tested the informativeness of gLM embeddings to predict mRNA alternative splicing quantified using RNA-seq (RNA-sequencing) from the ASCOT dataset [80]. Specifically, the prediction task takes as input two sequences—a sequence with 300 nt upstream of the splice acceptor and 100 nt downstream of the acceptor and a sequence with 100 nt upstream of the splice donor and 300 nt downstream of the donor—with the goal of predicting the percentage-spliced-in (PSI) across 56 tissues as a multi-task regression; a task introduced by MTSplice [81]. Similar to the DNA analysis, a baseline CNN was trained to take as input the full embeddings from gLMs or the embeddings of a pre-trained supervised model (see Methods section).

Our results mirrored those seen for regulatory DNA, with embedding-based models largely under-performing compared to one-hot-based models (Fig. 4a). In contrast, Sei's embeddings led to substantially lower performance than most gLM embeddings for this task. This is likely due to Sei's pre-training focus on DNA-based functional genomics

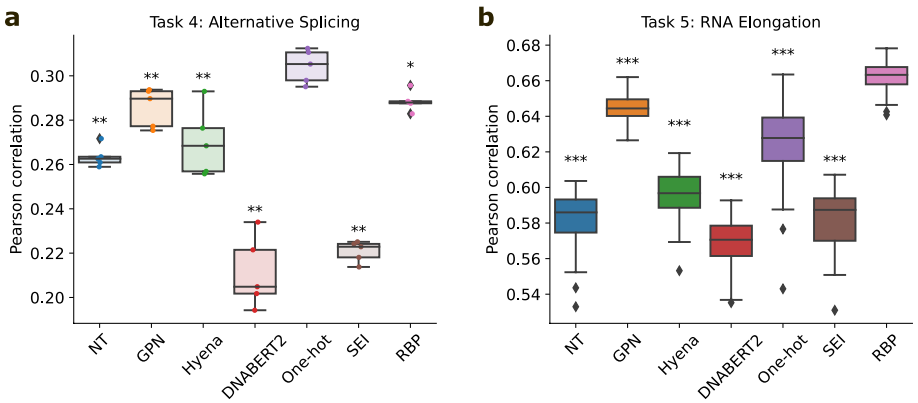

Fig. 4 Performance comparison on RNA regulation tasks. **a** Box-plots of the average Pearson correlation across tissues on test data for various models trained with different encoding schemes on an alternative splicing prediction task using MTSplice data. **b** Box-plot of the Pearson correlation for various models trained with different encoding schemes on a RNA poll II elongation potential prediction task using INSERT-seq data. Box-plots show the first and third quartiles, central line is the median, and the whiskers show the range of data. Box-plots represent 5 different random initializations for **a** and 50 different random initializations for **b**. Statistical significance represents the Mann-Whitney *U* test with a *p* value < 0.05 (*), < 0.01 (**), and < 0.001 (***)

data, which leads to learning a set of DNA regulatory features that do not transfer well to RNA regulation. To test whether a more relevant set of features acquired through supervised learning could transfer better for RNA regulation, we trained a multi-task ResidualBind-like model to classify RNA-protein binding (RBP) sites from a large trove of eCLIP-seq data (see Methods section). The task is to take 1000 nt sequences as input and predict binding for 120 RBPs in K562 cells as a multi-task classification. Indeed, the embeddings from this RBP-trained supervised model led to substantially better performance than the gLM embeddings, except GPN, which yielded comparable results (Fig. 4a).

Task 5: predicting RNA pol II elongation potential from INSERT-seq data

Next, we performed a similar analysis for a prediction task that takes 173 nt RNA sequences as input and predicts RNA pol II elongation potential measured via INSERT-seq (INtegrated Sequences on Expression of RNA and Translation using high-throughput sequencing) [82]. The INSERT-seq dataset is modest in size, containing only 10,774 sequences. This small data regime may not provide sufficient examples to learn all relevant patterns using one-hot sequences. Training a large deep learning model on this dataset can easily lead to over-fitting. Thus, this task can help evaluate a scenario (i.e., the low data regime) where a baseline CNN that uses gLM embeddings might have an advantage over one-hot sequences.

Similar to the other tasks, we found that the baseline CNNs trained using gLM embeddings yielded lower performance than one-hot RNA sequences, except for the custom GPN, which performed slightly better (Fig. 4b). Again, the CNN performance based on Sei's supervised embeddings was worse, and the best-performing model was achieved using embeddings from the supervised multi-task model pre-trained to classify RBPs. These results highlight that generic pre-training strategies are not always beneficial; when carefully selecting pre-training tasks, one should consider which relevant features are needed to ensure more positive outcomes on downstream applications.

While the custom GPN was the only embedding that demonstrated improved performance over one-hot sequences, we hypothesized that further down-sampling of the training data could lead to situations where gLM embeddings become more beneficial than one-hot sequences. We systematically down-sampled both the alternative splicing and INSERT-seq datasets and retrained the same baseline CNNs using different input encoding schemes. Interestingly, the GPN embeddings consistently outperformed other embeddings (Additional file 1: Fig. S4). The improved performance by GPN suggests that gLMs may specialize more effectively in specific genomic regions. Specifically in this dataset, capturing 5′ splice sites is a critical feature [82]. Thus, understanding what features gLMs learn well can help to identify suitable downstream tasks for which they can thrive.

Task 6: predicting RNA-binding protein binding with eCLIP-seq data

RBPs are essential for various RNA processing stages, so next, we examined the ability of gLMs to predict RBP binding sites using eCLIP-seq (enhanced chromatin immunoprecipitation sequencing) datasets [83]. Briefly, the task involves taking 200 nt DNA sequences as input and predicting binary labels of whether the sequence corresponds to

an eCLIP-seq peak or not (see Methods section). Ten eCLIP-seq datasets spanning different RBPs were used in the evaluation. We trained a baseline CNN model using different sequence encoding schemes similar to previous tasks.

We found that CNNs trained using the full gLM embeddings performed slightly worse on average compared to the one-hot sequences (Fig. 5a), in agreement with the ChIP-seq results of Task 2. The narrow performance difference between models using gLM embeddings and one-hot sequences also indicates that RBP motif information is not lost in the gLM embeddings. In a similar control, we found that an MLP based on Nucleotide Transformer's CLS token led to slightly better performance than an MLP based on dinucleotide frequencies (Additional file 1: Fig. S3b). This supports that gLM embeddings encode information that is slightly more informative than low-level sequence statistics in regulatory regions of RNA. Again, we found that Sei embeddings lead to a substantial decline in performance, further highlighting the importance of selecting appropriate pre-training tasks.

Uncovering cell-type-specific motifs learned by gLMs is challenging

As a follow-up, we performed an attribution analysis to identify motifs captured by gLMs. Attribution maps were generated for a given sequence by systematically masking one input token (i.e., a single nucleotide position for GPN and a non-overlapping *k*-mer for Nucleotide Transformer) at a time and calculating the entropy over the predicted distribution of the masked token; ΔEntropy, which is the difference between the maximum entropy value across the whole sequence and the entropy values at each position, was used to identify positions that yielded informative nucleotides (see Methods section). For comparison, we generated gradient-corrected Saliency Maps [84] for a CNN

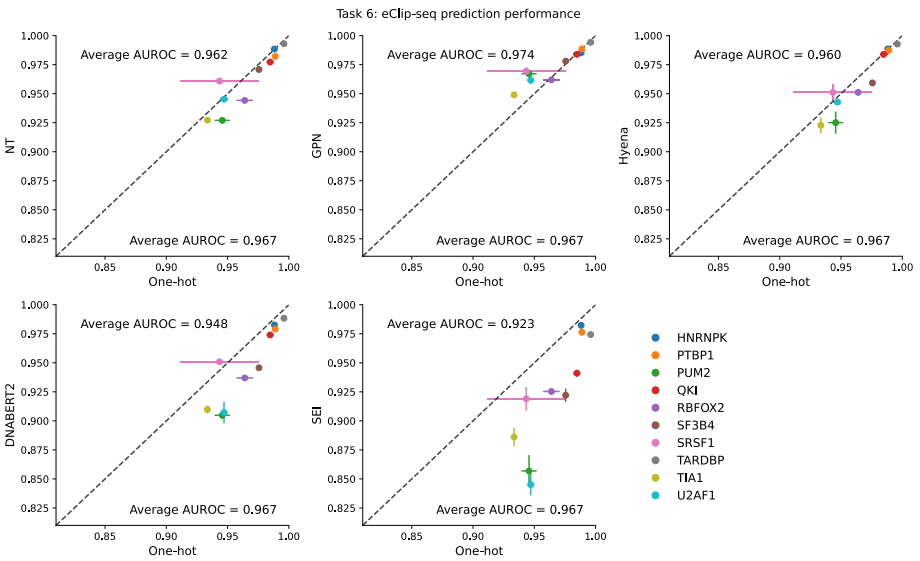

Fig. 5 Performance comparison on RBP binding prediction tasks from eCLIP-seq data. Comparisons of CNNs trained using different gLM embeddings versus CNNs trained using one-hot sequences for 10 RBP eCLIP-seq datasets. Performance is measured by the average area-under the receiver-operating characteristic curve (AUROC) and error bars represent the standard deviation of the mean across 5 different random initializations. Average AUROC represents the average performance across all eCLIP-seq datasets

trained using one-hot sequences. The analysis focused on lentiMPRA and CTCF ChIP-seq data to cover tasks from different systems with varying levels of complexity.

As expected, the attribution maps for pre-trained gLMs alone (i.e., not considering the downstream task) were difficult to interpret for both lentiMPRA (Fig. 6a) and ChIP-seq data (Additional file 1: Fig. S5a). The attribution maps did not reflect any known motifs, nor did they match any of the patterns captured in the CNN's Saliency Maps. This disparity can arise if the probed locus is used across different cell types for multiple purposes. If cell-type-specific *cis*-regulatory patterns are projected onto a single DNA sequence, the overlapping set of motifs can lead to complex attribution maps that may not resemble distinct cell-type-specific motifs. Alternatively, the complex patterns that seem to span the length of the sequence could also reflect low-level sequence statistics that are memorized. Without ground truth, interpreting attribution maps remains challenging.

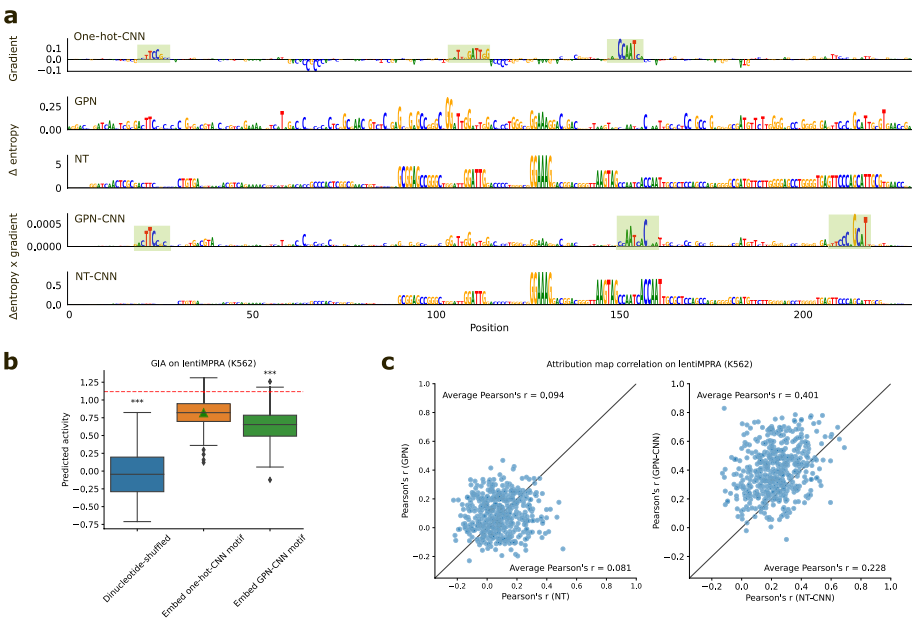

Fig. 6 Attribution analysis comparison for sequences from the lentiMPRA dataset. **a** Representative example of attribution maps for a regulatory sequence. Attribution maps include (top to bottom): the gradient-times-input of a one-hot-trained CNN; the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN; the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained Nucleotide Transformer; the gradient of a CNN-trained using GPN embeddings multiplied by the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained GPN; and the gradient of a CNN-trained using Nucleotide Transformer embeddings multiplied by the delta entropy of predicted nucleotides via single-nucleotide masking from a pre-trained Nucleotide Transformer. **b** Box-plot of the one-hot-trained CNN's predicted activity for 300 dinucleotide-shuffled sequences from **a**, dinuc-shuffled sequences with the annotated patterns from the Saliency Map of the one-hot-trained CNN, and dinuc-shuffled sequences with the annotated patterns from the CNN trained using GPN embeddings (GPN-CNN). Green triangle represents the global importance analysis value. Red dashed line represents the prediction of the wild type sequence according to the one-hot-trained CNN. Box-plots show the first and third quartiles, central line is the median, and the whiskers show the range of data. **c** Scatter plot comparison of the attribution map correlations for different pre-trained gLMs (left) and CNNs trained using gLM embeddings (right). Attribution map correlations reflect the Pearson correlation coefficient between the attribution map generated by the gLM-based attribution method with the Saliency Map generated by a one-hot-trained CNN. Each dot represents a different sequence in the lentiMPRA dataset ($N = 500$)

Next, we evaluated attribution maps generated by the downstream CNN that used gLM embeddings as input. Specifically, we scaled the gLM's entropy-based attribution map with the maximum gradients at each position based on the downstream CNN (see Methods section). Through a qualitative comparison, we noticed that the attribution maps generated by GPN appear to be visually aligned with Saliency Maps generated by the one-hot-trained CNN compared to Nucleotide Transformer (Fig. 6a), even after accounting for the block-like structure which arises due to the *k*-mer tokenization. This trend was also observed for other loci (Additional file 1: Fig. S6).

To validate the importance of the putative binding sites identified via Saliency Maps for the baseline one-hot-trained CNN, we employed global importance analysis (GIA) [75]. Specifically, we embedded the three annotated patterns into different dinucleotide-shuffled sequences, which serve as background sequences with low CRE activities, and measured the effect of introducing these patterns on model predictions. GIA revealed that the motif patterns identified by Saliency Maps from the one-hot-trained CNN drove predictions closer to wild-type activity levels than those identified by the GPN-CNN (Fig. 6b). This suggests that the one-hot-trained CNN learned motifs with higher efficacy in terms of sufficiency—i.e., the ability of these patterns to recapitulate wild-type activity when placed in otherwise inactive sequence contexts.

We then quantified the correlation between the attribution maps generated by the one-hot-trained CNN and the gLM-based attribution maps. We found that attribution maps generated by pre-trained gLM are not well-aligned with each other, nor are the attribution maps generated by the one-hot-trained CNN (Fig. 6c, Additional file 1: Fig. S5b). By contrast, attribution maps generated by CNNs trained with gLM embeddings led to improved alignment between their attribution maps and with one-hot-trained CNNs. These results suggest that the gLMs learn a different distribution of features during pre-training, but a downstream model can still use them to build cell-type-specific motifs (that are better aligned with motifs learned by one-hot-trained CNNs).

Together, the attribution maps given by pre-trained gLMs seem to visually capture a more diffuse set of patterns, which speculatively reflect low-level statistics of genomic sequences. Downstream models, like CNNs, use these seemingly uninformative gLM embeddings (especially from GPN) to build cell-type-specific regulatory features relevant for downstream prediction tasks.

Conclusion

Genomic language models are growing rapidly in numbers and architectural diversity, yet the scope and rigor of their evaluation have not kept pace. Many current gLMs trained on whole genomes are primarily evaluated on genic regions or tasks involving broad genomic annotations that are not cell type–specific. However, a central challenge in regulatory genomics is that the genome encodes cell type–specific usage of cis-regulatory elements, and it remains unclear whether current gLMs capture information relevant to this context.

To assess the transferability of knowledge acquired during pre-training, we evaluated the predictive utility of representations from four gLMs—each pre-trained on whole genomes without fine-tuning—across six functional genomics tasks that emphasize biologically meaningful, cell type–specific regulatory activity. Our benchmarks included

appropriate supervised baselines for comparison. Within cis-regulatory elements, we found that gLM representations provided little to no advantage over standard one-hot encoded sequences when used by downstream models to predict regulatory activity.

As part of a control experiment, we compared different ways of utilizing gLM embeddings. While the CLS token is commonly used as a compact summary of sequence-level information, CNNs trained directly on the full gLM embeddings consistently outperformed models using the CLS token across all tasks. The CLS token appeared to encode information only marginally more informative than bag-of-dinucleotide statistics, suggesting that it captures low-level sequence properties rather than biologically meaningful regulatory features. We also note that our evaluation used random train/test splits, some of which were predetermined by the original datasets. Although this practice is standard and enables comparability across models, it may favor CNNs operating directly on raw sequences by enabling them to more readily learn from homologous patterns shared across splits [85]. This could partially explain the performance gap between models trained using the full sequence embeddings versus the CLS token. However, we do not believe this significantly alters the main conclusions and include it here for completeness.

We chose not to fine-tune gLM weights on each downstream task, in contrast to prior benchmarks where fine-tuning was standard practice [23, 24, 26, 30, 39]. While fine-tuning reliably improves gLM performance, the goal of our study was to probe what biological knowledge is encoded in gLMs during pre-training. Our findings indicate that cell type–specific cis-regulatory features are largely learned during fine-tuning, and that the value of pre-training may lie mostly in initializing models with low-level sequence statistics. Further studies are needed to understand how biological knowledge is refined from pre-training to fine-tuning.

On a relative basis, GPN (a convolution-based gLM) yielded slightly more informative representations in the non-coding genome compared to more parameter-heavy BERT-style models. This suggests that incorporating stronger architectural inductive biases can modestly improve performance. Interestingly, HyenaDNA (a lightweight state space model) performed comparably to GPN and consistently outperformed larger foundation models across our benchmarks. Notably, both GPN and HyenaDNA operate at the nucleotide level. While their improved performance may appear to stem from tokenization strategies, models such as Evo2 [86], which builds on HyenaDNA but uses byte-pair encoding, suggest that tokenization alone does not fully explain the performance differences. Instead, these results emphasize the importance of architectural inductive biases for efficient learning in the non-coding genome. Convolutions in GPN likely aid in local pattern detection, which may be particularly useful for non-coding sequences, where regulatory signals are sparse and often embedded within variable sequence contexts. Notably, GPN and HyenaDNA are also significantly smaller in parameter size than other gLMs, suggesting that simply scaling model size may only incrementally improve performance in regulatory genomics.

Previous studies have shown that pre-training on focused genomic regions (e.g., coding regions, UTRs, or promoters) or simpler organisms with compact genomes can yield more promising results [28, 34, 87]. For example, codon-level gLMs trained on coding regions offer advantages over amino acid–level protein LMs by preserving synonymous

signals such as codon usage bias [35, 36, 45]. However, our evaluation shows that extending the pre-training task across the whole genome struggles to capture meaningful representations in the non-coding genome, a greatly understudied region for gLMs.

The performance gap may be due to differences in the structure of the coding regions versus the non-coding regions. To elaborate, protein sequences have a clear start and end with low-level grammars (i.e., secondary structures) and high-level grammars (i.e., protein domains) shared throughout most globular proteins, with structures conserved across species. On the other hand, the non-coding genome contains a variety of short sequence motifs that vary broadly in binding affinities and are sparsely located in seemingly random DNA, with usage and rules that vary across loci and cell types. Few non-coding elements exhibit deep conservation that is typical in proteins. The differing selection pressures in the non-coding regions lead to loss of synteny, which makes it difficult to study sequence and functional conservation. Thus, treating each nucleotide position equally, whether informative or uninformative, makes this a challenging language modeling task. In the non-coding genome, this is tantamount to expecting the LLM to predict predominantly random nucleotides, which, by definition, can only be achieved via memorization. Hence, this may explain why gLMs have also found greater utility in learning *cis*-regulatory features in simpler organisms with compact genomes, such as bacteria [40, 87, 88], arabidopsis [20], or yeast [28], which have substantially reduced junk DNA [89–91].

Recent supervised foundation models like Enformer [92] and Borzoi [93] may better reflect the potential of foundation models in genomics. However, their input requirements, often hundreds of kilobases, limit their use on short-sequence tasks like those in this study. These models require either heavy zero-padding (which may introduce a covariate shift) or context marginalization strategies [69, 75], which are computationally intensive. Furthermore, non-uniform data splits in many of these studies raise the possibility of data leakage and inflated performance estimates. In future evaluations, we plan to include more foundation models, including Enformer, Borzoi, and new gLMs that emerge, focusing on a broader set of chromatin-based functional genomics prediction tasks.

One core appeal of gLMs is that they do not require labeled data during pre-training, enabling them to capture general patterns that may be missed by supervised models trained on limited annotations. However, our findings suggest that current gLMs do not yet learn a foundational set of cis-regulatory features that have a competitive advantage over standard bespoke modeling in cell-type specific gene regulation tasks. In contrast, supervised deep learning models trained on large-scale functional genomics data are capable of learning discriminative cis-regulatory features. Yet these models are biased toward the specific experiments they are trained on and often fail to generalize across cell types.

However, our results suggest that gLMs have yet to learn a foundational set of *cis*-regulatory features in the non-coding genome of humans that can be harnessed via probing in prediction tasks across cell types. By contrast, supervised deep learning models trained on large troves of functional genomics data in a multitask setting can learn discriminative features related to *cis*-regulatory mechanisms in the non-coding

genome [70, 92–97]. Yet these models are biased toward the specific experiments they are trained on and often fail to generalize across cell types.

Although gLMs must currently be fine-tuned to match the performance of supervised models trained on one-hot inputs, the flexibility of fine-tuning remains a practical advantage. gLMs can be quickly adapted to diverse downstream tasks without extensive architectural tuning or hyperparameter optimization, making them appealing for broad applications once meaningful representations are learned during pre-training.

Determining what gLMs learn during pre-training remains a fundamental challenge. Predictive modeling only provides indirect evidence. Interpretability methods offer a more direct view into the representations learned by gLMs, but our attribution analysis was inconclusive. Attribution maps from pre-trained gLMs did not align well with known regulatory features, suggesting that the learned representations may reflect diffuse, low-level sequence statistics rather than biologically meaningful motifs. Further development can build upon the initial progress [98–100] towards more meaningful domain-inspired model interpretation tools to bridge this gap. Future work should develop more principled, domain-inspired interpretability tools and evalautaion benchmarks to clarify what features gLMs encode. Benchmark efforts such as DART-eval are beginning to move in this direction [101], but more comprehensive task suites will continue to be needed. Moreover, as longer sequence contexts are considered, evaluating whether gLMs learn meaningful long-range interactions, such as chromatin structure or enhancer-promoter interactions, should be included in future benchmarks.

Looking forward, it remains unclear whether LLMs will drive a similar paradigm shift in genomics as seen in NLP and protein science. Current efforts to scale gLMs via increased model size and longer sequence contexts have yielded only modest improvements and may become increasingly inefficient given the limited diversity and informativeness of available genomic data [21, 32]. As with neural scaling laws [102], whether continued scaling with MLM or CLM objectives will eventually enable emergent biological reasoning, such as capturing cell-type-specific cis-regulation, remains an open question. The human genome alone may not provide enough variation to enable pre-training to learn these complex regulatory grammars, especially given the vast spatiotemporal diversity of regulatory activity encoded in each individual. It is also unclear how much evolution-sampled genomes will bridge this gap as the conservation in human regulatory elements is not as strong as coding regions.

While gLMs are still in early stages, one conclusion is already clear: directly porting natural language processing-based pre-training objectives to genomics is unlikely to yield models with deep biological understanding of the non-coding genome. Incorporating functional genomics data during pre-training may be necessary for learning cell-type-specific regulatory logic. Even in protein modeling, where sequences are more structured with strong evolutionary constraints, both at the sequence and covariation levels, LLMs trained solely on amino acid sequences fall short in generalization to functional tasks [103]. For genomics, region-specific pre-training objectives may be needed to accommodate the high entropy and sparsity of functional signals in the non-coding genome. Given the high cost of training gLMs and the modest

downstream benefits observed thus far, we argue that future progress will require domain-informed innovations in pre-training strategies—beyond generic language modeling—in order for gLMs to truly earn their status as "foundation" models for the non-coding genome.

Methods

Pre-trained language models

Nucleotide transformer

Nucleotide Transformer consists of multiple BERT-based language models with 2 different model sizes (i.e., 500 million and 2.5 billion parameters) and trained on various sets of genome sequences: human reference genome, 1000 genomes project, and 850 genomes from several species. Details of the tokenizer, model structure, and training procedure can be found in the original paper [23]. We acquired weights for each Nucleotide Transformer model from the official GitHub repository. In this analysis, we mostly used representations from NT2.5B-1000G, except for the zero-shot variant effect generalization analysis, which considered all Nucleotide Transformer models. Since Nucleotide Transformer models allow flexible input sizes, no padding was necessary for any evaluation tasks.

Custom GPN

The GPN model is a convolutional neural network that was originally trained on *Arabidopsis* genome sequences via masked language modeling with an input size of 512 nucleotides [20]. It consists of 25 convolutional blocks, where each convolutional block includes a dilated convolutional layer followed by a feed-forward layer, connected by intermediate residual connections and layer normalization. The dilation rate for each convolutional layer cycles with increasing exponentially by factors of 2, from 1 to 32. The embedding dimension was kept fixed at 512 throughout the layers. For our custom GPN (human) model, we created training datasets using the human reference genome (hg38 [104]). The genome was split into contigs and filtered for a minimum length of 512 nucleotides, with chromosome 8 held out as test set. During training, 15% of the nucleotide positions were masked and the model is tasked to predict the nucleotide probabilities for each masked location. The model was trained for 2 million steps with a constant learning rate of 0.001 using ADAM [105].

HyenaDNA

The HyenaDNA model is a gLM pre-trained on the human reference genome, with context lengths up to 1 million tokens at the single nucleotide-resolution [21]. Architecturally, it adopts a decoder-only, sequence-to-sequence configuration, organized into a succession of blocks each encompassing a Hyena operator [59], followed by a feed-forward neural network. The model weights and representation extraction code was acquired through the Hugging Face repository [106]. For all experiments in this study, we used the "hyenadna-tiny-1k-seqlen-d256" model due to the sequence length limitation of the functional genomics datasets.

DNABERT2

DNABERT2, a second generation version of the original DNABERT model [24], is constructed on the BERT architecture, comprising 12 Transformer blocks. In this new iteration, the authors improved the model by replacing learned positional embeddings with Attention with Linear Biases (ALiBi) and utilizing Flash Attention to increase computation and memory efficiency [26]. In the context of this study, analyses were done with the representations generated by the last Transformer block. The model was acquired through the Hugging Face repository, using the "DNABERT-2-117M" model.

Pre-trained supervised models

Sei

The Sei model is composed of three sequential modules: (1) a convolutional network with dual linear and nonlinear paths; (2) residual dilated convolution layers; (3) spatial basis function transformation and output layers. Sei was trained to take as input 4 kb length sequences and predict 21,907 TF binding, histone marks and DNA accessibility from peak data of *cis*-regulatory profiles. For this study, we extracted our representations after the spline basis function transformation, before inputting into fully connected layers. The pre-trained Sei model was acquired through zenodo from the original study [74].

RBP

Our custom RBP model was trained using eCLIP-seq [83] data of 120 RBPs in K562 from ENCODE [107]. The dataset was organized into a multi-task binary classification format. The model has a ResidualBind-like structure:

1. 1D convolution (96 filters, size 19, batch-norm, exponential)
   dropout (0.1)
2. Dilated residual block [108]
   convolution (96 filters, size 3, batch-norm, ReLU)
   dropout (0.1)
   convolution (96 filters, size 3, batch-norm, dilation rate 2)
   dropout (0.1)
   convolution (196 filters, size 3, batch-norm, dilation rate 4)
   dropout (0.1)
   skip connection to input
   ReLU activation
   max-pooling (size 10)
   dropout(0.1)
3. 1D convolution (192 filters, size 7, batch-norm, ReLU)
   dropout (0.1)
   global average-pooling
4. flatten
5. fully-connected (512 units, batch-norm, ReLU)
   dropout (0.5)
6. output layer (120 units, sigmoid)

Enformer

A LASSO regression was fit to the lentiMPRA data based on predictions from Enformer [76]. Each sequence was padded to a target length of 196,608 base pairs using zero-padding on both ends. The padded sequences were then passed through Enformer to generate predictions. We extracted the predictions corresponding to the central bin (bin = 448, 1-based index). Lasso regression was employed using scikit-learn. The training data was split into training (80%) and validation (20%) sets. LassoCV with 5-fold cross-validation was employed to select the optimal regularization parameter (alpha) from 200 candidates. The model was trained with a maximum of 20,000 iterations and a tolerance of 1e−2. Performance was evaluated using the Pearson correlation coefficient on the test set. Unlike previously [73], the predictions were based on a single model trained on one fold and only considering the forward strand.

Data

lentiMPRA

The lentiMPRA dataset for K562 and HepG2 cell lines was acquired from the Supplementary Tables in Ref. [73]. The HepG2 library consists of 139,984 sequences, each 230 nucleotides long, and the K562 library contains 226,253 sequences. Each sequence is paired with a target scalar value that represents the transcriptional activity. Each cell line was treated independently as a single-task regression. For each dataset, we randomly split the training, validation, and test sets according to the fractions 0.7, 0.1, and 0.2, respectively. Unlike the original study, we treated reverse-complement sequences separately; they were not aggregated or augmented during test time. The results represent the performance over a single fold.

CAGI dataset

The CAGI5 challenge dataset [78] was used to evaluate the performance of the models on zero-shot single-nucleotide variant effect generalization as following the same procedure as Ref. [69]. We only considered MPRA experiments in HepG2 (LDLT, SORT1, F9) and K562 (PKLR). We extracted 230 nucleotide sequences from the reference genome centered on each regulatory region of interest. Alternative alleles are then substituted correspondingly to construct the CAGI test sequences. Pearson correlation was calculated between the varient effect scores by the model and experimentally measured effect size per experiment. For HepG2 performances, we report the average Pearson's r across the three experiments.

ChIP-seq

Ten transcription factor (TF) chromatin immunoprecipitation sequencing (ChIP-seq) datasets were acquired from the zenodo repository of Ref. [84]. The prediction task is a binary classification of whether 200 nt input DNA sequences are associated with a ChIP-seq peak (positive label) versus sequences from DNase I hypersensitive sites from the same cell type (i.e., GM12878) that do not overlap with any ChIP-seq peaks (negative label). The number of negative sequences were randomly down-sampled to exactly match the number of positive sequences to ensure balanced classes. The dataset was

split randomly into training, validation, and test set according to the fractions 0.7, 0.1, and 0.2, respectively.

Alternative splicing data

Data was acquired from direct correspondence with the authors of Ref. [81] Briefly, 61,823 cassette exons from ASCOT was split into a training, validation, and test set. The training set consisted of 38,028 exons from chromosomes 4, 6, 8, 10–23, and the sex chromosomes. The 11,955 exons from chromosomes 1, 7, and 9 were used as the validation set, and the remaining 11,840 exons were used as the test set (chromosomes 2, 3, and 5). Models are evaluated based on their performance on the test set. The prediction task takes as input two sequences—a sequence with 300 nt upstream of the acceptor and 100 nt downstream of the acceptor and a sequence with 100 nt upstream of the donor and 300 nt downstream of the donor—and the goal is to predict PSI across 56 tissues as a multi-task regression.

INSERT-seq

INSERT-seq data was obtained from Ref. [82]. INSERT-seq measures the impact of transcribed sequences on the RNA polymerase II elongation potential and expression in mouse embryonic stem cells. 11,417 insert sequences of length 173 nt long were used as inputs and the goal is to predict the totalRNA output, which measures the relative abundance in RNA relative to genomic DNA, as a regression task. Training, validation, and test sets were split according to the fractions 0.8, 0.1, and 0.1, resulting in 9131, 1149, and 1137 sequences, respectively.

eCLIP datasets

The in vivo eCLIP-based datasets were downloaded from the ENCODE. For each RBP experiment, the bed narrowPeaks (two replicates) and the bam file for the corresponding mock inputs experiment were downloaded. For each replicate, we removed peaks with a signal value less than 1 and a log-p value greater than 3. Using bedtools, the remaining peaks that share at least one nucleotide across the two replicates were selected as positive peaks. A correlation filter across the replicates was applied: $(2(s_i^1 - s_i^2)/(s_i^1 + s_i^2))^2 < 1.0$, where s_i^j represent the signal value for the ith peak in replicate j. The median peak size was about 50 nt with a positive tail that exceeded 200 nt in some cases. Positive sequences were generated by extracting 200 nucleotide sequences about the center position of the peak coordinates. Sequences with undefined nucleotides were filtered out. Negative peaks were generated by employing Piranha peak caller on the bam file of the mock inputs with a bin size of 20 and a p value threshold of 0.01. We then removed negative peaks which overlap with any unfiltered peaks from each replicate. Negative peaks were generated by extracting 200 nt sequences about the center position of the remaining negative peak coordinates. Because the negative peaks usually had more entries compared to the positive peaks, we randomly selected a similar number of negative peaks as positive peaks. All sequences were given a corresponding label 1 for sequences which contain a positive peak and 0 for sequences which contain a negative peak. All sequences were then randomly split into a training set, validation set, and test set according to the fractions 0.7, 0.1, and 0.2, respectively.

Models for downstream tasks

Linear models

Linear models with L2 regularization (i.e., Ridge) serve as the baseline, representing a simple downstream model. The inputs of the model were based on the embeddings of the CLS token or the average embedding across sequences for Nucleotide Transformer models. For regression and classification tasks, the linear model was a linear regression or logistic regression, respectively. The strength of the L2 regularization was set to 1e−3.

MLP

A multi-layer perceptron model was used to train on CLS token embeddings or the average embedding across sequences for Nucleotide Transformer models. The model is constructed by two fully connected blocks. The first block includes a fully connected layer with 512 units and ReLU ativation, followed by batch-normalization and a dropout rate of 0.5. The second block consists of a fully connected layer with 256 units and the same activation, batch-normalization, and dropout layers. The model was trained on lentiMPRA dataset with Adam optimizer, learning rate of 0.0001, mean-squared error loss function, learning rate decay with a patience of 5 epochs and a decay factor of 0.2, and early stopping patience of 10 epochs.

MPRAnn for lentiMPRA

MPRAnn is a convolutional based model with a total of 4 convolutional and 3 dense layers trained on the lentiMPRA dataset. It takes 230 nt one-hot encoded sequences including the adapters as input to predict the mean log2(RNA/DNA) values from forward and reverse strands. We augmented the batches using the reverse-complement of the 200 nt target sequence, while keeping the two 15 bp adapters fixed. To fit the model, we used a learning rate of 0.001, an early stopping criterion with patience of 10 on 100 epochs, and the Adam optimizer with a mean square error loss function. Model structure and training parameters obtained from Github directory of original publication [73].

Baseline CNN for lentiMPRA

We designed a baseline CNN model with the following structure:

1. batch-norm (optional)
2. 1D convolution (196 filters, size 1) (optional)
3. 1D convolution (196 filters, size 7, batch-norm, exponential)
   dropout (0.2)
   max-pooling (size 5)
4. 1D convolution (256 filters, size 7, batch-norm, ReLU)
   dropout (0.2)
   max-pooling (size 4)
5. flatten
6. fully-connected (512 units, batch-norm, ReLU)
   dropout (0.5)
7. fully-connected (256 units, batch-norm, ReLU)
   dropout (0.5)

8. output layer (1 unit, linear)

CNN models were trained with Adam optimizer, mean-squared error loss function, learning rate of 0.0001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs for both one-hot sequence and language model embedding-based training on the lentiMPRA dataset. For one-hot sequences, batch-norm and the convolution with kernel 1 were not employed.

ResidualBind for lentiMPRA

We designed the ResidualBind model by adding a dilated residual block after the first convolutional layer of the baseline CNN model, according to:

1. 1D convolution (196 filters, size 19, batch-norm, Silu)
   dropout (0.2)
2. Dilated residual block
   convolution (196 filters, size 3, batch-norm, Silu)
   dropout (0.1)
   convolution (196 filters, size 3, batch-norm, Silu, dilation rate 2)
   dropout (0.1)
   convolution (196 filters, size 3, batch-norm, Silu, dilation rate 4)
   dropout (0.1)
   convolution (196 filters, size 3, batch-norm, Silu, dilation rate 8)
   dropout (0.1)
   convolution (196 filters, size 3, batch-norm, Silu, dilation rate 16)
   dropout (0.1)
   convolution (196 filters, size 3, batch-norm, dilation rate 32)
   skip connection to input
   Silu activation
   max-pooling (size 5)
   dropout(0.2)
3. 1D convolution (256 filters, size 7, batch-norm, Silu)
   dropout (0.2)
   max-pooling (size 5)
4. fully-connected (256 units, batch-norm, Silu)
   dropout (0.5)
   average-poolint (size 2)
5. flatten
6. fully-connected (256 units, batch-norm, Silu)
   dropout (0.5)
7. output layer (1 unit, linear)

ResidualBind was trained with Adam optimizer, mean-squared error loss function, learning rate of 0.001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs.

Baseline CNN for ChIP-seq and CLIP-seq

We designed a baseline CNN model with the following structure:

1. batch-norm (optional)
2. 1D convolution (512 filters, size 1) (optional)
3. 1D convolution (64 filters, size 7, batch-norm, ReLU)
   max-pooling (size 4)
   dropout (0.2)
4. 1D convolution (96 filters, size 5, batch-norm, ReLU)
   max-pooling (size 4)
   dropout (0.2)
5. 1D convolution (128 filters, size 5, batch-norm, ReLU)
   max-pooling (size 2)
   dropout (0.2)
6. flatten
7. fully-connected (256 units, batch-norm, ReLU)
   dropout (0.5)
8. output layer (1 unit, linear)

CNN models were trained with Adam optimizer, binary cross-entropy loss function, learning rate of 0.001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs for both one-hot sequence and language model embedding-based training on the lentiMPRA dataset. For one-hot sequences, batch-norm and the convolution with kernel 1 were not employed.

Insert-seq model

For the RNA pol II elongation potential dataset, we developed a residual convolutional network structure and used it for all embedding and one-hot-based models. The model was trained using mean square error loss function, Adam optimizer, learning rate of 0.0001, learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping patience of 10 epochs.

1. convolution(48 filters, size 1) (optional)
2. convolution (96 filters, size 19, batch-norm, exponential)
   dropout (0.1)
3. dilated residual block
   convolution (96 filters, size 3, batch-norm, ReLU)
   dropout (0.1)
   convolution (96 filters, size 3, batch-norm, dilation rate 2)
   dropout (0.1)
   convolution (96 filters, size 3, batch-norm, dilation rate 4)
   skip connection to block input
   ReLU activation
   max-pooling (size 10)
   dropout(0.1)

4. convolution (128 filters, size 7, batch-norm, ReLU)
   global average-pooling
   dropout (0.1)
5. fully-connected layer (128 units, ReLU)
   dropout (0.5)
6. output layer (1 unit, linear)

CNN models were trained with Adam optimizer, mean-squared error loss function, learning rate of 0.0001 with a learning rate decay patience of 5 epochs with a decay rate of 0.2, and early stopping with patience of 10 epochs for both one-hot sequence and language model embedding-based training on the lentiMPRA dataset. For one-hot sequences, the convolution with kernel 1 was not employed.

Zero-shot variant effect prediction methods

For Nucleotide Transformer, we derived the zero-shot predictions using cosine similarity as suggested in the original study [23]. For each variant, we passed the sequences with the centered reference allele and the alternative allele through the model to extract embeddings. The cosine similarity between the two complete sequence embeddings was calculated and used as the zero-shot score. A negative correlation is expected between the score and effect size. Since this distance-based zero-shot score only reflects the magnitude, not the direction, of function change, we calculated the Pearson correlation using the absolute value of the effect size.

For GPN, we followed a similar procedure as the original study [20]. First, we input sequences with the center variant loci masked and acquired the predicted allele probabilities for the masked loci. Then, we calculate the zero-shot prediction score as the log-likelihood ratio between the alternate and reference alleles. Again, since the likelihood ratio does not reflect the direction of function change associated with the variants, we calculated the correlation score using the absolute value of effect size.

Finally, for the embedding-based and one-hot based models, we used the difference in predictions between the alternative and reference allele sequence as the zero-shot prediction score. For Enformer, we use the cell-type agnostic approach of averaging the effect size across all DNase-seq tracks, as it was previously shown to have a similar performance as a cell type-matched approach [76]. To reduce predictions to scalars, we summed across the profile predictions.

Attribution methods

For CNN models, the attribution analysis was based on grad-times-input with saliency maps. The gradients of the prediction were calculated with respect to the input sequence to yield an $L \times A$ map, where L is the length of the sequence and A is 4 (one for each nucleotide). By subtracting the position-wise average saliency scores from this map and then multiplying by the one-hot encoded sequence, the method isolates the sensitivity of each observed nucleotide at every position, enhancing interpretability by pinpointing nucleotide-specific contributions to predictions.

For gLMs, the analysis involved sequentially masking each token of the input sequence and predicting the probability of the masked token by the model. The entropy of the probability distribution for each position was computed to quantify the information content represented by the gLM. Given that lower entropy signifies a higher information level, the saliency score was derived as the difference between the maximum entropy value and the entropy at each position, ensuring that a higher saliency score reflects greater information retention.

Sequence logos were visualized using Logomaker [109].

Global importance analysis

Global importance analysis was carried out according to Ref. [75]. A example sequence was selected from the LentiMPRA (K562) dataset. We sampled 300 dinucletoide shuffled versions of the sequence to be used as background sequences. The shuffling aims to preserve the dinucleotide frequency while destroying any coherent patterns. Predictions were given by the baseline CNN model trained on the LentiMPRA dataset using one-hot sequences. Predictions for the shuffled sequences are considered to be the baseline for predicted CRE activity. The top three positive motif patterns identified separately in the One-hot-CNN and GPN-CNN saliency maps (Fig. 6c) were inserted into the corresponding position of the shuffled sequences, creating two experiment sequence sets. The One-Hot-CNN model was used to make predictions for the motif embedded sequences. The difference in prediction for the three sets of sequences reflects the global importance of these motif patterns to the CNN model.

Model fine-tuning on lentiMPRA data

We fine-tuned DNABERT2, HyenaDNA, and Nucleotide Transformer on lentiMPRA data derived from HepG2 and K562 cell lines. Each model was adapted for sequence regression, predicting a single continuous value corresponding to CRE activity.

Nucleotide Transformer

We used the 500M parameter version pre-trained on 1000 Genomes data. Fine-tuning employed the LoRA technique, applied to the query and value matrices of the self-attention mechanism. The LoRA rank was set to 1, with a scaling factor of 32 and dropout rate of 0.1. An AdamW optimizer with a learning rate of 5e−4 was used, training for 1000 steps or 2 epochs (whichever came first) with a batch size of 64.

HyenaDNA

The pre-trained `hyenadna-tiny-1k-seqlen` model was used for fine-tuning. The model performed a mean pool of the penultimate representation across sequence length and then used a linear layer to transform the pooled representation to a single output. We employed a character-level tokenizer for DNA bases (A, C, G, T, N), with a maximum sequence length of 230 tokens and left-side padding. Training used PyTorch with an AdamW optimizer, learning rate of 6e−4, and weight decay of 0.1 for 100 epochs with a batch size of 256. The final model checkpoint after 100 epochs was used for evaluation.

DNABERT2

We utilized the `DNABERT-2-117M` model, fine-tuning it using the Hugging Face Transformers library. An AdamW optimizer with a learning rate of $3e-5$ was used, training for 5 epochs with batch sizes of 8 and 16 for training and evaluation, respectively.

For all models, mean squared error was used as the loss function. The lentiMPRA datasets for both cell lines were preprocessed and stored in HDF5 format. Model performance was evaluated using mean squared error, Pearson correlation coefficient, and Spearman correlation coefficient. All experiments were conducted using CUDA-enabled GPUs, with the best-performing model for each combination selected based on the lowest validation loss.

Supplementary information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-025-03674-8.

Additional file 1. Supplementary Tables S1-S3 and Figures S1-S6.

Acknowledgements
The authors would like to thank Evan Seitz for providing Enformer variant effect predictions and other members of the Koo Lab for helpful comments on the manuscript.

Peer review information
Tim Sands was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team. The peer-review history is available in the online version of this article.

Authors' contributions
ZT and PKK conceived of the method and designed the experiments. ZT developed code, ran the experiments, and analyzed the majority of the results. YY and NS performed the fine-tuning analysis for gLMs for Task 1 and Task 3. NS performed the Enformer analysis for Task 1. ZT and PKK interpreted the results and contributed to writing the paper.

Funding
Research reported in this publication was supported in part by the National Human Genome Research Institute of the National Institutes of Health under Award Number R01HG012131 and the National Institute Of General Medical Sciences of the National Institutes of Health under Award Number R01GM149921. ZT was also supported by the Elisabeth Sloan Livingston Fellowship. Funding for NS was provided by the National Institute of Health Postbaccalaureate Research Experience Program at Cold Spring Harbor Laboratory (NIGMS PREP award R25GM144246). This work was performed with assistance from the US National Institutes of Health Grant S10OD028632-01.

Data availability
Processed data and model weights can be found at Ref. [110] (https://doi.org/10.5281/zenodo.8279715). Datasets include lentiMPRA (Task 1) [111], ChIP-seq (Task 2) [112], MPRAs for zero-shot single-nucleotide generalization (Task 3) [113], alternative splicing (Task 4) [114], INSERT-seq (Task 5) [115], and eCLIP-seq (Task 6) [112].
Open-source code to reproduce this study is available on GitHub (https://github.com/amberT15/LLM_eval [116]) under a MIT License as well as Ref. [110] (https://doi.org/10.5281/zenodo.8279715) under a Creative Commons Attribution 4.0 International License.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 20 December 2024 Accepted: 27 June 2025
Published online: 14 July 2025

References

1. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv. 2018;1810.04805.
2. OpenAI. Gpt-4 technical report. arXiv. 2023;2303.08774.
3. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, et al. Llama: open and efficient foundation language models. arXiv. 2023;2302.13971.

4. Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, Yogatama D, Bosma M, Zhou D, Metzler D, et al. Emergent abilities of large language models. arXiv. 2022;2206.07682.
5. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, Guo D, Ott M, Zitnick C, Ma J, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci. 2021;118(15), e2016239118.
6. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, Gibbs T, Feher T, Angerer C, Steinegger M, et al. Prottrans: toward understanding the language of life through self-supervised learning. IEEE Trans Pattern Anal Mach Intell. 2021;44(10):7112–7127.
7. Madani A, McCann B, Naik N, Keskar NS, Anand N, Eguchi RR, Huang P-S, Socher R. Progen: language modeling for protein generation. arXiv. 2020;2004.03497.
8. Bepler T, Berger B. Learning the protein language: evolution, structure, and function. Cell Syst. 2021;12(6):654–69.
9. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, Santos Costa A, Fazel-Zarandi M, Sercu T, Candido S, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. BioRxiv. 2022;2022:500902.
10. Chowdhury R, Bouatta N, Biswas S, Floristean C, Kharkar A, Roy K, Rochereau C, Ahdritz G, Zhang J, Church GM, et al. Single-sequence protein structure prediction using a language model and deep learning. Nat Biotechnol. 2022;40(11):1617–23.
11. Wu R, Ding F, Wang R, Shen R, Zhang X, Luo S, Su C, Wu Z, Xie Q, Berger B, et al. High-resolution de novo structure prediction from primary sequence. bioRxiv. 2022;2022.07.21.500999.
12. Brandes N, Goldman G, Wang CH, Ye CJ, Ntranos V. Genome-wide prediction of disease variant effects with a deep protein language model. Nat Genet. 2023;55(9):1512–22.
13. Meier J, Rao R, Verkuil R, Liu J, Sercu T, Rives A. Language models enable zero-shot prediction of the effects of mutations on protein function. Adv Neural Inf Process Syst. 2021;34:29287–303.
14. Madani A, Krause B, Greene ER, Subramanian S, Mohr BP, Holton JM, Olmos JL Jr, Xiong C, Sun ZZ, Socher R, et al. Large language models generate functional protein sequences across diverse families. Nat Biotechnol. 2023;8:1–8.
15. Ferruz N, Höcker B. Controllable protein design with language models. Nat Mach Intell. 2022;4(6):521–32.
16. Hie BL, Shanker VR, Xu D, Bruun TU, Weidenbacher PA, Tang S, Wu W, Pak JE, Kim PS. Efficient evolution of human antibodies from general protein language models. Nat Biotechnol. 2024;42(2):275–283.
17. Hie BL, Yang KK, Kim PS. Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins. Cell Syst. 2022;13(4):274–85.
18. Zhang Z, Wayment-Steele HK, Brixi G, Wang H, Dal Peraro M, Kern D, Ovchinnikov S. Protein language models learn evolutionary statistics of interacting sequence motifs. bioRxiv. 2024;2024.01.30.577970.
19. Consens ME, Dufault C, Wainberg M, Forster D, Karimzadeh M, Goodarzi H, Theis FJ, Moses A, Wang B. To transformers and beyond: large language models for the genome. arXiv. 2023;2311.07621.
20. Benegas G, Batra SS, Song YS. DNA language models are powerful predictors of genome-wide variant effects. Proc Natl Acad Sci. 2023;120(44):2311219120.
21. Nguyen E, Poli M, Faizi M, Thomas A, Birch-Sykes C, Wornow M, Patel A, Rabideau C, Massaroli S, Bengio Y, et al. Hyenadna: long-range genomic sequence modeling at single nucleotide resolution. arXiv. 2023;2306.15794.
22. Lal A, Biancalani T, Eraslan G. regLM: Designing realistic regulatory DNA with autoregressive language models. NeurIPS 2023 Generative AI and Biology (GenBio) Workshop. 2023.
23. Dalla-Torre H, Gonzalez L, Mendoza-Revilla J, Carranza NL, Grzywaczewski AH, Oteri F, Dallago C, Trop E, Sirelkhatim H, Richard G, et al. The nucleotide transformer: building and evaluating robust foundation models for human genomics. bioRxiv. 2023;2023.01.11.523679.
24. Ji Y, Zhou Z, Liu H, Davuluri RV, Zhou Z, Liu H, Davuluri RV, Zhou Z, Liu H, Davuluri RV. Dnabert: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. Bioinformatics. 2021;37(15):2112–2120.
25. Zhang D, Zhang W, He B, Zhang J, Qin C, Yao J. Dnagpt: a generalized pretrained tool for multiple DNA sequence analysis tasks. bioRxiv. 2023;2023.07.11.548628.
26. Zhou Z, Ji Y, Li W, Dutta P, Davuluri R, Liu H. Dnabert-2: efficient foundation model and benchmark for multi-species genome. arXiv. 2023;2306.15006.
27. Sanabria M, Hirsch J, Joubert PM, Poetsch AR. DNA language model grover learns sequence context in the human genome. Nat Mach Intell. 2024;6(8):911–923.
28. Karollus A, Hingerl J, Gankin D, Grosshauser M, Klemon K, Gagneur J. Species-aware DNA language models capture regulatory elements and their evolution. Genome Biol. 2024;25(1):83.
29. Chu Y, Yu D, Li Y, Huang K, Shen Y, Cong L, Zhang J, Wang M. A 5'UTR language model for decoding untranslated regions of mRNA and function predictions. Nat Mach Intell. 2024;6(4):449–460.
30. Chen K, Zhou Y, Ding M, Wang Y, Ren Z, Yang Y. Self-supervised learning on millions of pre-mRNA sequences improves sequence-based RNA splicing prediction. bioRxiv. 2023;2023.01.31.526427.
31. Shen X, Li X. Omnina: a foundation model for nucleotide sequences. bioRxiv. 2024;2024.01.14.575543.
32. Zaheer M, Guruganesh G, Dubey KA, Ainslie J, Alberti C, Ontanon S, Pham P, Ravula A, Wang Q, Yang L, et al. Big bird: transformers for longer sequences. Adv Neural Inf Process Syst. 2020;33:17283–97.
33. Fishman V, Kuratov Y, Petrov M, Shmelev A, Shepelin D, Chekanov N, Kardymon O, Burtsev M. Gena-lm: a family of open-source foundational models for long DNA sequences. bioRxiv. 2023;2023.06.12.544594.
34. Benegas G, Albors C, Aw AJ, Ye C, Song YS. Gpn-msa: an alignment-based DNA language model for genome-wide variant effect prediction. bioRxiv. 2023;2023.10.10.561776.
35. Hallee L, Rafailidis N, Gleghorn JP. cdsbert-extending protein language models with codon awareness. bioRxiv. 2023;2023.09.15.558027.
36. Li S, Moayedpour S, Li R, Bailey M, Riahi S, Kogler-Anele L, Miladi M, Miner J, Zheng D, Wang J, et al. Codonbert: large language models for mRNA design and optimization. bioRxiv. 2023;2023.09.09.556981.
37. Gündüz HA, Binder M, To X-Y, Mreches R, Bischl B, McHardy AC, Münch PC, Rezaei M. A self-supervised deep learning method for data-efficient training in genomics. Commun Biol. 2023;6(1):928.

38. Yang M, Huang L, Huang H, Tang H, Zhang N, Yang H, Wu J, Mu F. Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution. Nucleic Acids Res. 2022;50(14):81.
39. Chen J, Hu Z, Sun S, Tan Q, Wang Y, Yu Q, Zong L, Hong L, Xiao J, Shen T, et al. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. arXiv. 2022;2204.00300.
40. Zvyagin M, Brace A, Hippe K, Deng Y, Zhang B, Bohorquez CO, Clyde A, Kale B, Perez-Rivera D, Ma H, et al. Genslms: genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. Int J High Perform Comput Appl. 2023;37(6):683–705.
41. Levy B, Xu Z, Zhao L, Kremling K, Altman R, Wong P, Tanner C. Florabert: cross-species transfer learning with attention-based neural networks for gene expression prediction. Research Square. 2022.
42. Liang C, Bai W, Qiao L, Ren Y, Sun J, Ye P, Yan H, Ma X, Zuo W, Ouyang W. Rethinking the bert-like pretraining for DNA sequences. arXiv. 2023;2310.07644.
43. Gu A, Dao T. Mamba: linear-time sequence modeling with selective state spaces. arXiv. 2023;2312.00752.
44. Liu H, Zhou S, Chen P, Liu J, Huo K-G, Han L. Exploring genomic large language models: bridging the gap between natural language and gene sequences. bioRxiv. 2024;2024.02.26.581496.
45. Outeiral C, Deane CM. Codon language embeddings provide strong signals for use in protein engineering. Nat Mach Intell. 2024;6(2):170–9.
46. Schiff Y, Kao C-H, Gokaslan A, Dao T, Gu A, Kuleshov V. Caduceus: bi-directional equivariant long-range DNA sequence modeling. arXiv. 2024;2403.03234.
47. Zhai J, Gokaslan A, Schiff Y, Berthel A, Liu Z-Y, Miller ZR, Scheben A, Stitzer MC, Romay MC, Buckler ES, et al. Cross-species modeling of plant genomes at single nucleotide resolution using a pre-trained DNA language model. bioRxiv. 2024;2024.06.04.596709.
48. Mendoza-Revilla J, Trop E, Gonzalez L, Roller M, Dalla-Torre H, Almeida BP, Richard G, Caton J, Lopez Carranza N, Skwark M, et al. A foundational large language model for edible plant genomes. Commun Biol. 2024;7(1):835.
49. Trotter MV, Nguyen CQ, Young S, Woodruff RT, Branson KM. Epigenomic language models powered by cerebras. arXiv. 2021;2112.07571.
50. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. Language models are unsupervised multitask learners. OpenAI blog. 2019;1(8):9.
51. Joshi M, Chen D, Liu Y, Weld DS, Zettlemoyer L, Levy O. Spanbert: improving pre-training by representing and predicting spans. Trans Assoc Comput Linguist. 2020;8:64–77.
52. Clark K, Luong M-T, Le QV, Manning CD. Electra: pre-training text encoders as discriminators rather than generators. arXiv. 2020;2003.10555.
53. Zhang Z, Wu Y, Zhao H, Li Z, Zhang S, Zhou X, Zhou X. Semantics-Aware BERT for Language Understanding. Proc AAAI Conf Artif Intell. 2020;34:9628–9635.
54. Cui Y, Che W, Liu T, Qin B, Yang Z. Pre-training with whole word masking for Chinese bert. IEEE/ACM Trans Audio Speech Lang Process. 2021;29:3504–14.
55. Zhang Z, Liu J, Razavian N. Bert-xml: large scale automated ICD coding using bert pretraining. arXiv. 2020;2006.03685.
56. Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units. arXiv. 2015;1508.07909.
57. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inf Process Syst. 2017;30.
58. Dao T, Fu D, Ermon S, Rudra A, Ré C. Flash attention: fast and memory-efficient exact attention with IO-awareness. Adv Neural Inf Process Syst. 2022;35:16344–59.
59. Poli M, Massaroli S, Nguyen E, Fu DY, Dao T, Baccus S, Bengio Y, Ermon S, Ré C. Hyena hierarchy: towards larger convolutional language models. arXiv. 2023;2302.10866.
60. Penić RJ, Vlašić T, Huber RG, Wan Y, Šikić M. Rinalmo: general-purpose RNA language models can generalize well on structure prediction tasks. arXiv. 2024;2403.00043.
61. Consortium GP, et al. A global reference for human genetic variation. Nature. 2015;526(7571):68.
62. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. Lora: low-rank adaptation of large language models. 2021. arXiv 2106.09685.
63. Zhan H, Wu YN, Zhang Z. Efficient and scalable fine-tune of language models for genome understanding. arXiv. 2024;2402.08075.
64. Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning. arXiv. 2021;2104.08691.
65. Liu H, Tam D, Muqeeth M, Mohta J, Huang T, Bansal M, Raffel CA. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. Adv Neural Inf Process Syst. 2022;35:1950–65.
66. Marin FI, Teufel F, Horrender M, Madsen D, Pultz D, Winther O, Boomsma W. Bend: benchmarking DNA language models on biologically meaningful tasks. arXiv. 2023;2311.12570.
67. Robson ES, Ioannidis NM. Guanine v1. 0: benchmark datasets for genomic AI sequence-to-function models. Machine Learning in Computational Biology. PMLR. 2024;250–266.
68. Vilov S, Heinig M. Investigating the performance of foundation models on human 3'UTR sequences. bioRxiv. 2024;2024.02.09.579631.
69. Toneyan S, Tang Z, Koo PK. Evaluating deep learning for predicting epigenomic profiles. Nat Mach Intell. 2022;4(12):1088–1100.
70. Nair S, Ameen M, Sundaram L, Pampari A, Schreiber J, Balsubramani A, Wang YX, Burns D, Blau HM, Karakikes I, et al. Transcription factor stoichiometry, motif affinity and syntax regulate single-cell chromatin dynamics during fibroblast reprogramming to pluripotency. bioRxiv. 2023;2023.10.04.560808.
71. Shlyueva D, Stampfel G, Stark A. Transcriptional enhancers: from properties to genome-wide predictions. Nat Rev Genet. 2014;15(4):272–86.
72. Zeitlinger J. Seven myths of how transcription factors read the cis-regulatory code. Curr Opin Syst Biol. 2020;23:22–31.

73. Agarwal V, Inoue F, Schubach M, Martin B, Dash P, Zhang Z, Sohota A, Noble W, Yardimci G, Kircher M, et al. Massively parallel characterization of transcriptional regulatory elements in three diverse human cell types. Nature. 2025;1–10.
74. Chen KM, Wong AK, Troyanskaya OG, Zhou J. A sequence-based global map of regulatory activity for deciphering human genetics. Nat Genet. 2022;54(7):940–9.
75. Koo PK, Majdandzic A, Ploenzke M, Anand P, Paul SB. Global importance analysis: an interpretability method to quantify importance of genomic features in deep neural networks. PLoS Comput Biol. 2021;17(5):1008925.
76. Avsec Ž, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, Assael Y, Jumper J, Kohli P, Kelley DR. Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods. 2021;18(10):1196–1203.
77. Ren B, Robert F, Wyrick JJ, Aparicio O, Jennings EG, Simon I, Zeitlinger J, Schreiber J, Hannett N, Kanin E, et al. Genome-wide location and function of DNA binding proteins. Science. 2000;290(5500):2306–9.
78. Kircher M, Xiong C, Martin B, Schubach M, Inoue F, Bell RJ, Costello JF, Shendure J, Ahituv N. Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. Nat Commun. 2019;10(1):3583.
79. Shigaki D, Adato O, Adhikari AN, Dong S, Hawkins-Hooker A, Inoue F, Juven-Gershon T, Kenlay H, Martin B, Patra A, et al. Integration of multiple epigenomic marks improves prediction of variant impact in saturation mutagenesis reporter assay. Hum Mutat. 2019;40(9):1280–91.
80. Ling JP, Wilks C, Charles R, Leavey PJ, Ghosh D, Jiang L, Santiago CP, Pang B, Venkataraman A, Clark BS, et al. Ascot identifies key regulators of neuronal subtype-specific splicing. Nat Commun. 2020;11(1):137.
81. Cheng J, Çelik MH, Kundaje A, Gagneur J. Mtsplice predicts effects of genetic variants on tissue-specific splicing. Genome Biol. 2021;22:1–19.
82. Vlaming H, Mimoso CA, Field AR, Martin BJ, Adelman K, Mimoso CA, Field AR, Martin BJ, Adelman K. Screening thousands of transcribed coding and non-coding regions reveals sequence determinants of RNA polymerase II elongation potential. Nat Struct Mol Biol. 2022;29(6):613–620.
83. Van Nostrand EL, Pratt GA, Shishkin AA, Gelboin-Burkhart C, Fang MY, Sundararaman B, Blue SM, Nguyen TB, Surka C, Elkins K, et al. Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced clip (eclip). Nat Methods. 2016;13(6):508–14.
84. Majdandzic A, Rajesh C, Koo PK. Correcting gradient-based interpretations of deep neural networks for genomics. Genome Biol. 2023;24(1):109.
85. Rafi AM, Kiyota B, Yachie N, Boer C. Detecting and avoiding homology-based data leakage in genome-trained sequence models. bioRxiv. 2025;2025.01.22.634321.
86. Brixi G, Durrant MG, Ku J, Poli M, Brockman G, Chang D, Gonzalez GA, King SH, Li DB, Merchant AT, et al. Genome modeling and design across all domains of life with Evo 2. bioRxiv. 2025;2025.02.18.638918.
87. Nguyen E, Poli M, Durrant MG, Thomas1 AW, Kang B, Sullivan J, Ng MY, Lewis A, Patel A, Lou A, Ermon S, Baccus SA, Hernandez-Boussard1 T, Re C, Hsu PD, Hie BL. Sequence modeling and design from molecular to genome scale with Evo. Science. 2024;386(6723):eado9336.
88. Shao B. A long-context language model for deciphering and generating bacteriophage genomes. Nat Commun. 2024;15(1):9392.
89. Eddy SR. The c-value paradox, junk DNA and encode. Current Biol. 2012;22(21):898–9.
90. Niu D-K, Jiang L. Can encode tell us how much junk DNA we carry in our genome? Biochem Biophys Res Commun. 2013;430(4):1340–3.
91. Graur D, Zheng Y, Price N, Azevedo RB, Zufall RA, Elhaik E. On the immortality of television sets: "function" in the human genome according to the evolution-free gospel of encode. Genome Biol Evol. 2013;5(3):578–90.
92. Avsec Ž, Weilert M, Shrikumar A, Krueger S, Alexandari A, Dalal K, Fropf R, McAnany C, Gagneur J, Kundaje A, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. Nat Genet. 2021;53(3):354–66.
93. Linder J, Srivastava D, Yuan H, Agarwal V, Kelley DR. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. Nat Genet. 2025;1–13.
94. Koo PK, Eddy SR. Representation learning of genomic sequence motifs with convolutional neural networks. PLoS Comput Biol. 2019;15(12):1007560.
95. Koo PK, Ploenzke M. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. Nat Mach Intell. 2021;3(3):258–66.
96. Almeida BP, Reiter F, Pagani M, Stark A. Deepstarr predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. Nat Genet. 2022;54(5):613–24.
97. Seitz EE, McCandlish DM, Kinney JB, Koo PK. Interpreting cis-regulatory mechanisms from genomic deep neural networks using surrogate models. Nat Mach Intell. 2024;6(6):701–713.
98. Clauwaert J, Menschaert G, Waegeman W. Explainability in transformer models for functional genomics. Brief Bioinforma. 2021;22(5):060.
99. Sanabria M, Hirsch J, Poetsch AR. Distinguishing word identity and sequence context in DNA language models. BMC Bioinformatics. 2024;25(1):301.
100. Zhang Y-Z, Bai Z, Imoto S. Investigation of the bert model on nucleotide sequences with non-standard pre-training and evaluation of different k-mer embeddings. Bioinformatics. 2023;39(10):617.
101. Patel A, Singhal A, Wang A, Pampari A, Kasowski M, Kundaje A. Dart-eval: a comprehensive DNA language model evaluation benchmark on regulatory DNA. arXiv. 2024;2412.05430.
102. Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, Casas DDL, Hendricks LA, Welbl J, Clark A, et al. Training compute-optimal large language models. arXiv. 2022;2203.15556.
103. Li F-Z, Amini AP, Yue Y, Yang KK, Lu AX. Feature reuse and scaling: understanding transfer learning with protein language models. bioRxiv. 2024;2024.02.05.578959.

104. Schneider VA, Graves-Lindsay T, Howe K, Bouk N, Chen H-C, Kitts PA, Murphy TD, Pruitt KD, Thibaud-Nissen F, Albracht D, et al. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. Genome Res. 2017;27(5):849–64.
105. Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014. arXiv 1412.6980.
106. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, et al. Huggingface's transformers: state-of-the-art natural language processing. arXiv. 2019;1910.03771.
107. The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012;489(7414):57–74.
108. Yu F, Koltun V, Funkhouser T. Dilated residual networks. Proc IEEE Conf Comput Vis Pattern Recognit. 2017;472–480.
109. Tareen A, Kinney JB. Logomaker: beautiful sequence logos in Python. Bioinformatics. 2020;36(7):2272–4.
110. Tang Z, Somia N, Yu Y, Koo PK. Dataset: genome language model probing task. Zenodo. Datasets. 2023. https://doi.org/10.5281/zenodo.8279715
111. Agarwal V, Inoue F, Schubach M, Martin B, Dash P, Zhang Z, Sohota A, Noble W, Yardimci G, Kircher M, Shendure J. Dataset: processed lentiMPRA data from massively parallel characterization of transcriptional regulatory elements in three diverse human cell types. ENCODE. Datasets. ENCODE accessions: ENCSR463IRX, ENCSR460LZI, ENCSR022GQD, ENCSR382BVV, ENCSR244FWB, ENCSR405QCT, ENCSR203UFY, ENCSR336MKI. 2023. https://www.encodeproject.org/.
112. Majdandzic A, Rajesh C, Koo PK. Dataset: ChIP-seq and CLIP-seq data from correcting gradient-based interpretations of deep neural networks for genomics. Zenodo. Datasets. 2023. https://doi.org/10.5281/zenodo.7011631.
113. Kircher M, Xiong C, Martin B, Schubach M, Inoue F, Bell RJ, Costello JF, Shendure J, Ahituv N. Dataset: saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. Gene Expression Omnibus. Datasets. 2019. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE126550.
114. Ling JP, Wilks C, Charles R, Leavey PJ, Ghosh D, Jiang L, Santiago CP, Pang B, Venkataraman A, Clark BS, et al. Dataset: ASCOT identifies key regulators of neuronal subtype-specific splicing. ASCOT Database. Datasets. 2020. http://ascot.cs.jhu.edu/.
115. Vlaming H, Mimoso CA, Field AR, Martin BJ, Adelman K. Dataset: screening thousands of transcribed coding and non-coding regions reveals sequence determinants of RNA polymerase II elongation potential. Gene Expression Omnibus. Datasets. 2022. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE178230.
116. Tang Z, Somia N, Yu Y, Koo PK. Code: genome language model probing analysis. GitHub. 2023. https://github.com/amberT15/LLM_eval.

Publisher's Note