Trends in Genetics

CellPress

Review

Genomic language models: opportunities and challenges

Gonzalo Benegas[1,4], Chengzhong Ye[2,4], Carlos Albors[1,4], Jianan Canal Li[1,4], and Yun S. Song[1,2,3,*]

Large language models (LLMs) are having transformative impacts across a wide range of scientific fields, particularly in the biomedical sciences. Just as the goal of natural language processing is to understand sequences of words, a major objective in biology is to understand biological sequences. Genomic language models (gLMs), which are LLMs trained on DNA sequences, have the potential to significantly advance our understanding of genomes and how DNA elements at various scales interact to give rise to complex functions. To showcase this potential, we highlight key applications of gLMs, including functional constraint prediction, sequence design, and transfer learning. Despite notable recent progress, however, developing effective and efficient gLMs presents numerous challenges, especially for species with large, complex genomes. Here, we discuss major considerations for developing and evaluating gLMs.

Highlights

Following their remarkable recent success in protein biology, language models are making their way into the field of genomics.

Genomic language models (gLMs) trained on DNA sequences have achieved state-of-the-art results on genome-wide variant effect prediction.

gLMs can also be used to design novel DNA sequences and to improve downstream prediction tasks in genomics via transfer learning.

We review the key opportunities and challenges for gLMs and outline important considerations for their development and evaluation to benefit the genomics community.

Motivation for developing language models for DNA

Recent advances in artificial intelligence (AI)/machine learning (ML) have profoundly impacted a wide range of scientific disciplines, revolutionizing approaches to modeling, data analysis, interpretation, and discovery. One of the key pillars of this development is self-supervised learning, in which training on massive amounts of unlabeled data enables the learning of complex features and their interactions. This paradigm has particularly transformed natural language processing (NLP), allowing AI models to match human performance on several challenging tasks, including translation [1], speech recognition [2], and even answering questions from standardized professional and academic exams [3].

Just as the aim of NLP is to understand sequences of natural language, a major aim of computational biology is to understand biological sequences. As such, there has been intense recent interest in adapting modern techniques from NLP for biological sequences (DNA, RNA, proteins). In particular, protein sequence databases (e.g., UniProt [4]) have grown exponentially over the past decade, and protein language models (pLMs) trained on these immense data have achieved impressive performance on complex problems such as structure prediction [5] and variant effect prediction [6,7], to name just a few examples (see [8,9] for reviews on pLMs and their applications). This success aligns with the intuition that billions of years of evolution have explored portions of the protein sequence space that are relevant to life, so large unlabeled datasets of protein sequences are expected to contain significant biological information.

In a similar vein, LLMs trained on DNA sequences have the potential to transform genomics, but training an effective model for genomes presents several additional challenges. For instance, unlike proteins, which are functionally important units and relatively small in size, most genomes are much larger and often contain vast amounts of complex, non-functional regions that overshadow the amount of functional elements. In addition, the number of available whole-genome

[1] Computer Science Division, University of California, Berkeley, CA, USA
[2] Department of Statistics, University of California, Berkeley, CA, USA
[3] Center for Computational Biology, University of California, Berkeley, CA, USA
[4] These authors contributed equally to this work

*Correspondence:
yss@berkeley.edu (Y.S. Song).

https://doi.org/10.1016/j.tig.2024.11.013

sequences across the tree of life is minuscule compared with the hundreds of millions of protein sequences, limiting the diversity of functionally important DNA elements in training data. Despite these issues, we believe that language models trained on genomes – referred to as gLMs – hold great promise for biology. In this article, we review some of the key opportunities and challenges in this domain and outline major considerations that should be addressed to develop and evaluate gLMs that would be useful to the genomics community.

Applications

The general language model framework is summarized in Box 1. Later, we elaborate on three main application areas of gLMs: functional constraint prediction, sequence design, and transfer learning.

Functional constraint prediction

An intriguing application of gLMs is the prediction of functional constraint on a genomic locus without any supervision on the task. A significant benefit of this approach is its independence from labels, such as whether a variant is disease-causing, which are often limited and subject to biases. The underlying idea is that reference genomes, typically derived from healthy individuals, are depleted of deleterious variants. Consequently, models trained on these data are predisposed

Box 1. General language model framework

At a high level, a language model is trained to learn the conditional probability distribution of the form $\mathbb{P}[X_i|X_{-\text{Masked}}]$ for $i \in$ Masked [in masked language modeling (MLM)] or $\mathbb{P}[X_k|X_{1:k-1}]$ [in causal language modeling (CLM)], where $X = (X_1, X_2, \ldots)$ denotes a sequence of 'tokens' (e.g., nucleotides or amino acids) and 'Masked' denotes a collection of masked positions. The key to recent advances in NLP is that, instead of fitting a simple parametric model of context dependency that one designs by hand, one lets the data speak for themselves and fit more complex models as more data are observed, by leveraging powerful deep learning architectures. Figure I depicts the language modeling framework for DNA. While the model is trained to predict the nucleotide at each masked site using information from unmasked sites, it will learn position-specific contextual representation (called embedding, a high-dimensional vector in $\mathbb{R}^n$), which then gets converted into a probability distribution over {A,C,G,T}. These embeddings and probability distributions, both of which are position-specific, can be applied to many problems in genomics.

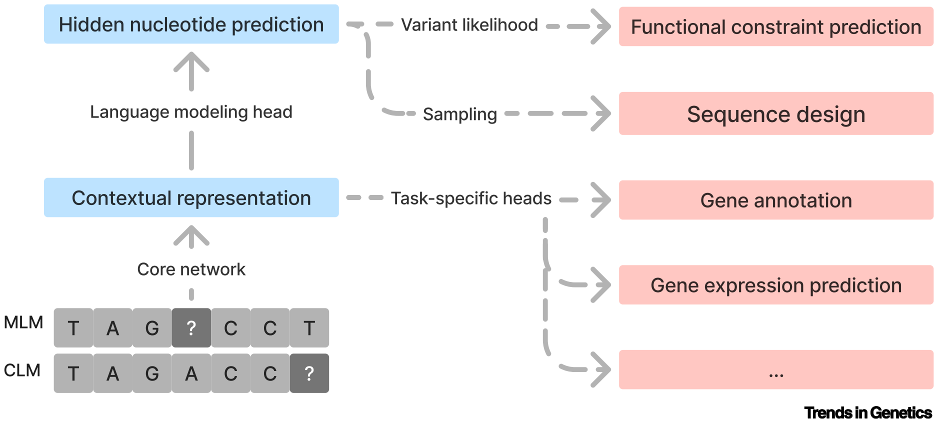
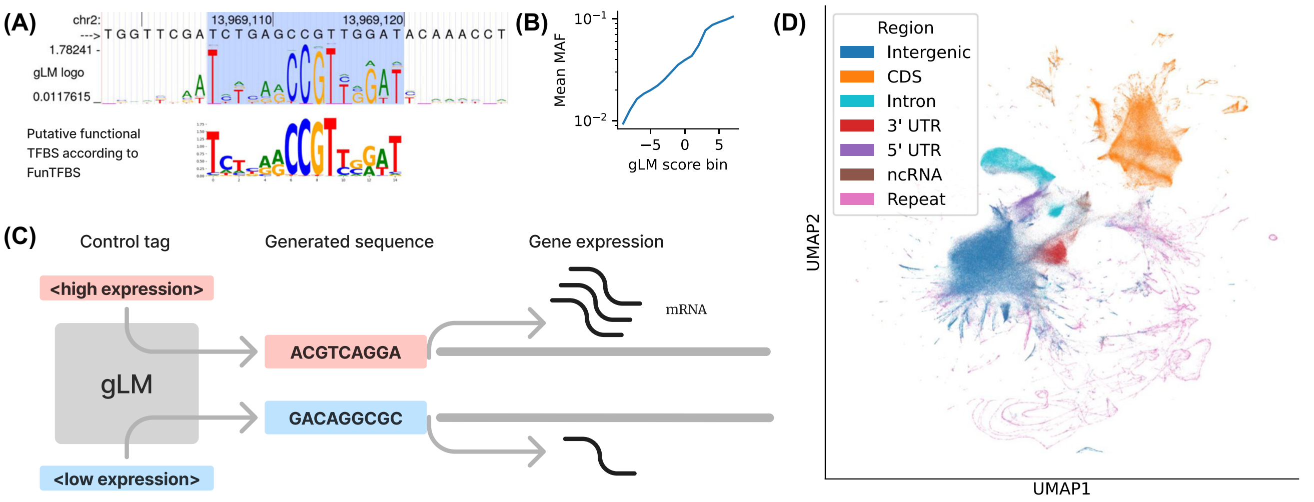


Figure I. Training and applications of genomic language models (gLMs). The schematic on the left-hand side illustrates gLM training. The log-likelihood ratio (LLR) between two alleles (specifically, $\log\left[\mathbb{P}(X_i = a|X_{-i})/\mathbb{P}(X_i = b|X_{-i})\right]$) is a good unsupervised predictor of functional constraint (see section 'Functional constraint prediction' in the main text). New sequences can be generated by sampling from the learned probability distribution (see section 'Sequence design'). A vector representation, called embedding, of each token in the input sequence can be extracted and adapted for different downstream tasks (see section 'Transfer learning'). Abbreviations: CLM, causal language modeling; MLM, masked language modeling.

to assigning lower probabilities to harmful variants. This observation underpins the strategy of using the log-likelihood ratio (LLR) between two alleles (i.e., $\log[\mathbb{P}(X_i = a|X_{-i})/\mathbb{P}(X_i = b|X_{-i})]$) to estimate their relative fitness.

Functional constraint prediction using the LLR was initially introduced in the context of protein sequence models, leading to outstanding results in predicting the effects of missense variants [6,10–12]. Expanding this approach, genome-wide functional constraint prediction using a gLM was first undertaken by Genomic Pre-trained Network (GPN) [13], achieving state-of-the-art results in the model plant *Arabidopsis thaliana*. To illustrate how a gLM might be able to predict functional constraint, we note that gLMs can learn transcription factor binding site (TFBS) motifs, understanding which positions are under constraint and which are not (Figure 1A). In addition, GPN's LLR score is correlated with allele frequencies in natural *A. thaliana* populations, even though the model was only trained on a single genome from this species (Figure 1B). Subsequently, AgroNT [14] and PlantCaduceus [15] have also obtained excellent results in other plant species. For the human genome, however, the LLR from the Nucleotide Transformer (NT) [16] fell short of existing baselines. Meanwhile, GPN-MSA [17], leveraging a whole-genome multiple sequence alignment (MSA) across diverse vertebrate species, was able to attain state-of-the-art performance (see 'Learning objective' section for further MSA considerations). It should be noted that the observed nucleotide distribution is driven not only by functional constraint but also by mutational biases; explicitly incorporating this information into functional constraint prediction is a promising avenue of future research.

For a single nucleotide polymorphism (SNP), the LLR can be computed in a single query to an MLM with the variant position masked, but in two queries to a CLM on the reference and alternate sequences. A CLM can just as easily handle multiple substitutions, insertions, and deletions, while an MLM must resort to a more expensive pseudo-LLR [12,18]. Scores other than the LLR have been proposed for functional constraint prediction, such as the distance in embedding space [14,16] or the change in nucleotide probabilities in positions around a mutation [19].



Trends in Genetics

Figure 1. Application examples. (A) Genomic language model (gLM) predicted logo plot (top) at a promoter, highlighting a motif (bottom logo) that matches a putative functional transcription factor binding site (TFBS). (B) Correlation between variant minor allele frequency (MAF) and gLM score (log-likelihood ratio). (C) A gLM can be prompted with different control tags to design promoter sequences driving high or low expression in a given cell type. (D) Visualization of gLM embeddings for different classes of genomic windows, illustrating that the learned representations contain useful information such as gene regions. (A, B, D) were generated using the GPN model. Abbreviations: CDS, coding DNA sequence; GPN, Genomic Pre-trained Network; ncRNA, noncoding RNA; UTR, untranslated region.

While the LLR is widely used in both the pLM and gLM communities, it is important to understand better in which contexts these alternative scores are useful.

There are two main kinds of variant effect predictors in genomics: functional constraint predictors, including gLMs and traditional conservation scores [20,21], and activity predictors, such as the gene expression predictor Enformer [22] or the splicing predictor SpliceAI [23]. These two kinds of models are related in the sense that if a variant at a locus is under selection, it induces a change in activity in some context (e.g., change in transcription of a certain gene during limb development), ultimately affecting a high-level trait (e.g., polydactyly). Functional constraint models cover all possible mechanisms and contexts that affect the overall organismal fitness, while activity models reflect only those they are explicitly trained on (some data, such as protein expression in the developing human brain, are just hard to obtain). However, activity models can nominate a specific mechanism and context through which a variant acts, while functional constraint models do not offer a mechanistic interpretation.

With regard to functional variant prioritization, there are some additional considerations. An activity model typically gives similar scores to two variants that induce a similar expression fold-change but in different genes, even if there is a vast difference in physiological tolerance to their expression levels. However, a trait not under detectable selection could still be of scientific or medical interest. In this case, a functional constraint model would have limited power to prioritize variants affecting it, especially if they have small effect sizes, as is the case in complex trait genome-wide association studies (GWAS). However, while a gLM's LLR might not have high power in this setting, gLM's learned embeddings (Box 1) could still provide value with additional supervision on labeled data [24].

Sequence design

Designing novel biological sequences is of great interest to both the academic and industry research communities due to its immense potential in drug discovery and delivery; agricultural improvement; bioremediation; and the development of biological research tools. We here describe sequence generation with a CLM (Box 1) as it is the most common approach (see 'Learning objective' section for generation with MLMs). Specifically, the sequence generation task is decomposed into a series of next-token prediction problems. Starting with a given sequence fragment (referred to as prompts [25], or control tags [26]), the language model can predict the next token recursively and generate a whole new sequence. pLMs have been shown to be powerful tools for protein design [26–29]. Going beyond coding sequences, designing noncoding sequences is also crucial due to its applications such as gene and cell therapies [30], as well as synthetic biology [31]. Such design tasks have previously been addressed using supervised activity models [32–34], but more recently several works have explored the use of gLMs to tackle this challenge as described next.

The model regLM [30] was built upon the causal gLM HyenaDNA [35] and used to perform *de novo* generation of promoter and enhancer sequences. HyenaDNA models are trained or fine-tuned on regulatory sequences with control tags prepended. The trained model is then used to generate new regulatory sequences with given tags (Figure 1C). The authors performed *in silico* evaluation of the diversity and activity of the generated sequences in yeast and human cell lines and demonstrated the sequences to have desired functionality as well as realistic and diverse sequence features.

gLMs have the unique potential for multi-modal design tasks such as generating protein–RNA complexes by unifying them as DNA sequence design. For instance, EVO, a gLM trained on prokaryote genomes, was used to design novel CRISPR-Cas systems [31]. The model was fine-tuned using a dataset of CRISPR-Cas sequences with Cas subtype-specific prompt prepended. The fine-tuned model was able to generate novel CRISPR-Cas sequences that matched the subtype prompt and had predicted structures that resemble naturally existing systems.

Additionally, gLMs can be potentially used to design organized, functional DNA sequences at the chromosome or genome-scale. Recently, two gLMs, MegaDNA and EVO, have explored such design tasks for prokaryote genomes [31,36]. EVO was used to generate 20 sequences of approximately 650 Mbp. The generated sequences were found to have realistic coding sequence density, protein sequences with predicted secondary structure and globular folds, as well as plausible tRNA sequences. MegaDNA was used to generate full bacteriophage genomes up to 96 kbp. Apart from validating coding sequences, the author further identified functional regulatory elements including promoters and ribosome binding sites in the generated sequences. Yet, such mega-scale DNA sequence design tasks remain challenging. The generated sequences by EVO were found to lack highly conserved marker genes that typically exist in functional prokaryote genomes, and the predicted protein structures have limited matches to natural protein databases. A recent independent evaluation [37] revealed that the sequence composition of MegaDNA-generated genomes is still largely dissimilar to natural genomes. Therefore, further work is needed to refine the methods to enable *de novo* design of fully functional genomes with gLMs.

Transfer learning

Neural networks trained to predict annotations from functional genomics experiments have been widely utilized to interpret the functions of genomic elements. A significant application has been predicting variant effects on molecular phenotypes, such as gene expression [22,38–41] and splicing [23,42]. The ability of neural networks to interpret complex interactions between genomic sites has made them essential tools for tackling these important problems, but suitable training data are often difficult to collect and consequently limited. To generalize on prediction tasks, models need to be capable of identifying the broad set of functionally important sequence elements, which may require substantial data and computation. To overcome the limitations of insufficient data for individual tasks, developers have employed transfer learning methods – techniques that leverage knowledge gained from training models on one task to improve performance on related tasks. Specifically, most neural networks trained to predict functional annotations have been trained to predict a wide array of annotations simultaneously, forcing these models to learn a single unifying representation. This, in turn, has improved their generalization performance.

Language models may also be utilized for transfer learning. (See Box 2 for a discussion of the utility of transfer learning for NLP.) One technique is feature extraction: while learning to predict the context-dependent distribution of nucleotides, gLMs transform input genomic sequences into intermediate vector representations (Box 1). These representations may distill relevant information and, therefore, be utilized as features for another model. For example, visualization of gLM

Box 2. Transfer learning in NLP

For NLP models to generalize on most tasks (including typical tasks, such as sentiment analysis, question answering, and part-of-speech tagging, to name only a few), models need to understand grammar and meaning. However, data specific to these tasks are limited. Utilizing LLMs trained on raw text data (sourced from articles, books, and websites) for transfer learning has enabled breakthrough progress on these problems [74]. Today, virtually every state-of-the-art NLP model is adapted from an LLM.

Transfer learning techniques have underpinned the recent boom in natural language models. In particular, the availability of pretrained models that are broadly adaptable to downstream tasks – termed 'foundation models' – has yielded a major shift in how machine learning (ML) models are developed [112].

embeddings reveals that, without any supervision, the model has learned to distinguish different classes of genomic elements such as coding sequence and untranslated regions [13] (Figure 1D). Embeddings from different layers can provide information useful for different tasks [43]. Another way to utilize language models for transfer learning is to use them as pretrained models: that is, to continue training them on downstream tasks. This technique is called fine-tuning. Fine-tuning a pretrained neural network on a task implicitly regularizes its parameters such that the network's predictions synthesize knowledge from both tasks. As a result, pretraining neural networks tends to improve their generalization performance on downstream tasks. In recent work, SegmentNT, a model developed by fine-tuning the NT gLM [16] to the task of annotating genes and *cis-regulatory* elements, achieved state-of-the-art performance on this task [44]. Utilizing a pretrained model was shown to be critical to its success. Similarly, AgroNT [14], another model of the NT family, was pretrained on diverse plant species and then fine-tuned to predict chromatin accessibility and gene expression on select crop species. DNABERT-S [45] applied contrastive learning with pretrained DNABERT-2 [46] embeddings to perform metagenomics binning. IsoFormer [47] is an example of multi-modal transfer learning between DNA and pLMs for the task of predicting transcript isoform expression. These recent successes suggest that fine-tuned gLMs may make meaningful progress on diverse genome interpretation tasks.

Two recent studies evaluated several gLMs in prediction tasks in the human genome and found that they generally did not outperform non-gLM baselines [48,49]. These results were based on frozen embeddings; evaluating full fine-tuning would provide additional insights. While gLMs are already well suited to demonstrate the value of transfer learning in less-studied organisms, further innovation may be required for them to offer significant value in human genetics, where high-quality labeled data and carefully crafted models already exist. An important question is how far the scaling hypothesis holds for gLMs (i.e., how much increasing unlabeled data and computation will keep improving model performance). A recent pLM study found that scaling improved only protein structure prediction but not most other tasks such as function or property prediction [50], so gLM tasks should also be held to the same scrutiny.

Development

We now describe the key components of developing useful gLMs; a schematic diagram summarizing the development pipeline is illustrated in Figure 2. We first describe the importance of selecting and preparing training data and then discuss architectural and training decisions. We then consider interpreting and benchmarking gLMs. Our aim is to provide insights into the methodologies and challenges encountered in developing gLMs that are both effective and efficient. To provide a comprehensive view of the current landscape in the field, we list in Table 1 some of the existing gLMs that we are aware of and summarize their design decisions.

Training data

The performance of an ML model is significantly influenced by both its architecture and its training data. Various model architectures such as convolutional neural networks (CNNs), transformers, and state-space models (SSMs) have been successfully adapted to a wide range of domains, including natural language, images, audio, proteins, and genomics. However, selecting suitable data for pretraining requires a deep understanding of the specific domain, especially in genomics where there is no universally accepted, curated dataset comparable with those in NLP (e.g., the Pile [51,52]) or protein biology (e.g., UniProt [4]).

A key consideration is data quality. For example, in NLP this may refer to data sources that have undergone editing or peer review, such as scientific articles or books [52]. In the case of proteins,

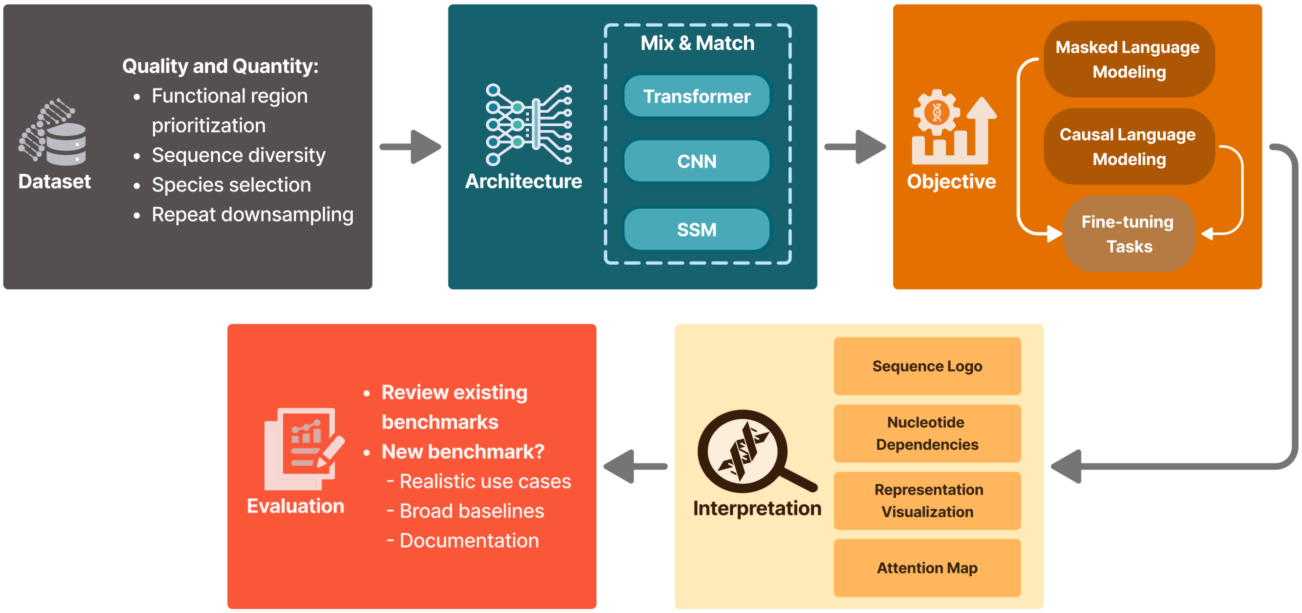


Figure 2. Development pipeline. This figure illustrates the general genomic language model (gLM) development pipeline described in this review, from model conception to deployment. We begin with the selection and preparation of the training dataset, emphasizing the importance of data quality and quantity (see section ‘Training data’). Subsequently, in the sections ‘Model architecture’ and ‘Learning objective’, we explore the various choices for designing and training gLMs, discussing the strengths and weaknesses of different approaches. We also examine how hybrid models combine elements from multiple architectures to mitigate specific limitations. In ‘Interpretation’, we discuss methods for analyzing and interpreting the outputs of gLMs. Finally, in ‘Evaluation’, we present evaluation methods through current benchmarks, emphasizing the complexities in aligning model performance with actual biological functions. Abbreviations: CNN, convolutional neural network; SSM, state-space model.

quality control involves removing predicted pseudogenes or truncated proteins that are no longer functional [4]. However, a recent study found only 3.3% of the bases in the human reference genome, the most popular gLM training dataset (Table 1), to be significantly constrained and likely functional [53]. Importantly, a typical genomic sequence used for training a gLM will contain a mix of functional and non-functional sites and one cannot always separate training examples into high versus low quality. A proposed solution is to have a base-pair-level weighting of the training loss according to the evidence for functionality [17].

It is standard in NLP and proteins to filter out duplicated sequences, which improves training efficiency and reduces memorization [54]. Despite the fact that a staggering 50% of the human genome is repetitive (a high proportion across eukaryotes), very few gLM studies propose a solution (downweighting [13,15] or downsampling [55,56]), let alone acknowledge the issue. It would be insightful if studies of language model perplexity [24,35] would also report it separately for non-repetitive regions, to distinguish improvements due to generalization versus memorization.

Another key question is how to ensure that the amount of data is enough. It is likely that a single genome might not be enough to train a large model, especially if non-functional regions are downsampled or downweighted. One approach is to add sequence variants from the same species [16]. However, in many species, including humans, there is relatively little variation between individuals. A more common approach is to train across multiple species (Table 1), as typically done for pLMs. As species become more distant, regulatory logic diverges faster than proteins [57]. One proposed approach is to explicitly add a species identifier as an extra input to the model [58]. Notwithstanding, it is plausible that a large enough model, with enough genomic

Table 1. A summary of existing gLMs[a,b]

Model name	Pretraining data sources	Task	Architecture	Tokenization	Notes
BigBird [65]	Human	MLM	Transformer	BPE	
DNABERT [93]	Human	MLM	Transformer	Overlapping *k*-mer	
GeneBERT [84]	Human	MLM	Transformer	Overlapping *k*-mer	Trained to also predict chromatin accessibility ATAC-seq data
Epigenomic BERT [85]	Human	MLM	Transformer	Non-overlapping *k*-mer	DNA sequences are paired with associated epigenetic state information (IDEAS) [114] during training
LookingGlass [115]	Bacteria + archaea	CLM	Recurrent neural network	Nucleotide-level	Metagenomic sequences from diverse environments rather than assembled genomes are used for training
LOGO [67]	Human	MLM	CNN + transformer	Overlapping *k*-mer	
ViBE [116]	Virus	MLM	Transformer	Overlapping *k*-mer	
GPN [13]	*Arabidopsis thaliana* + 7 related Brassicales genomes	MLM	CNN	Nucleotide level	
FloraBERT [117]	Several hundred plants + selected maize genomes	MLM	Transformer	BPE	Only 1-kb promoter sequences are used in training
INHERIT [118]	Bacteria + bacteriophage	MLM	Transformer	Overlapping *k*-mer	
GenSLMs [119]	Prokaryotic gene sequences + SARS-CoV-2 genomes	CLM	Transformer	Non-overlapping *k*-mer	Pretrain on prokaryotic genes and fine-tune on SARS-CoV-2 genomes
NT [16]	Human + 1000 Genomes Project + multi-species	MLM	Transformer	Non-overlapping *k*-mer	
SpliceBERT [55]	Human + 71 vertebrate genomes	MLM	Transformer	Nucleotide level	Only RNA transcripts are used in training
SpeciesLM Fungi [58]	1500 Fungal genomes	MLM	Transformer	Overlapping *k*-mer	Only 5′ and 3′ UTR regions are used in training: the 5′ species LM and 3′ species LM
GENA-LM [69]	Human + multi-species	MLM	Transformer	BPE	
DNABERT-2 [46]	Human + multi-species	MLM	Transformer	BPE	
HyenaDNA [35]	Human	CLM	SSM	Nucleotide level	
GROVER [120]	Human	MLM	Transformer	BPE	
DNAGPT [121]	Human + multi-species	CLM	Transformer	Non-overlapping *k*-mer	
GPN-MSA [17]	Human + multiple sequence alignment (MSA) with 100 vertebrate genomes	MLM	Transformer	Nucleotide level	
UTR-LM [122]	Human + 4 vertebrate genomes	MLM	Transformer	Nucleotide level	Only 5′ UTR regions are used in training. Trained also to predict mRNA minimum free energy and secondary structures calculated by ViennaRNA [123]
hgT5 [56]	Human	T5 [124]	Transformer	Unigram model [125]	
AgroNT [14]	48 Plant genomes focusing on edible plant species	MLM	Transformer	Non-overlapping *k*-mer	
MegaDNA [36]	~100K bacteriophage genomes	CLM	Transformer	Nucleotide level	
regLM [30]	Human + yeast	CLM	SSM	Nucleotide level	Human enhancer and yeast promoter sequences are used to fine-tune/pretrain separate HyenaDNA [35] models
EVO [31]	Bacteria + archaea + virus + plasmid	CLM	SSM + transformer	Nucleotide level	
Caduceus [24]	Human	MLM	SSM	Nucleotide level	

Table 1. (continued)

Model name	Pretraining data sources	Task	Architecture	Tokenization	Notes
ChatNT [89]	Genomic sequences + English instructions	CLM	Transformer	Overlapping *k*-mer	Combines the pretrained gLM NT [16] and the English LM Vicuna [126]. Trained to perform all supervised genomics prediction tasks as text-to-text tasks
LucaOne [86]	Genomic and protein sequences from 169 861 species	MLM	Transformer	Nucleotide and amino acid level	Mixed pretraining with DNA, RNA, and protein sequences. Trained also to predict 8 types of selected annotations
PlantCaduceus [15]	16 Angiosperm genomes	MLM	SSM	Nucleotide level	
CD-GPT [87]	Genomic and protein sequences of 14 organisms	CLM	Transformer	BPE	Mixed pretraining with DNA, RNA, and protein sequences, followed by targeted DNA–protein and mRNA–protein paired pretraining.
SpeciesLM Metazoa [19]	494 metazoan genomes	MLM	Transformer	overlapping *k*-mer	Only trained on 2 kb upstream of start codons
gLM2 [88]	Metagenomes and genomes from IMG [127] and MGnify [128]	MLM	Transformer	BPE for nucleotides, amino acid level for proteins	Pretraining with a mixed-modality dataset, comprising interleaved protein-coding (amino acid) and intergenic (nucleotide) sequences

[a] An overview of various gLMs is provided, highlighting their pretraining datasets, tasks, architectures, tokenization methods, and unique features. The models are listed in the order of their public release dates.

[b] Abbreviations: ATAC-Seq, assay for transposase-accessible chromatin with sequencing; BPE, byte-pair encoding; CLM, causal language modeling; CNN, convolutional neural network; LM, language model; MLM, masked language modeling; SARS-CoV-2, severe acute respiratory syndrome coronavirus 2; SSM, state-space model.

context, might be able to naturally model distant genomes, similarly to how LLMs handle multilingual datasets.

As mentioned earlier, in prokaryotes, there exist models (MegaDNA and EVO) that take an entire genome as context [31,36]. This is currently infeasible for eukaryotes and therefore leads to the question of how to partition the genome into context windows to be separately modeled. Many interactions are restricted to nearby positions, such as TFBS motifs, motivating the development of models with a relatively small context (<6 kb) (Table 1). However, there are obvious long-range interactions, such as between exons of the same gene or between enhancers and promoters (up to 1 Mb) [59]. Such long context lengths introduce computational and statistical challenges and efforts have been made to overcome them [24,31,35,36]. Regardless of the chosen context length, it is still not easy to partition the genome into independent units (similarly to how proteomes are separated by protein). For instance, the enhancer of a gene can be located inside the intron of another gene [59] and multiple genes can be controlled by the same enhancer [60]. Avoiding potential data leakage due to orthology and paralogy is quite challenging, especially when training across species.

The choice of training data may significantly influence gLMs' outputs and learned representations. DNA sequences observed in nature are the outcome of various evolutionary processes, the foremost of which are mutation and selection [61]. For certain applications, it may be desirable to curate training data such that one of these processes is more manifest than the other. For example, for the sake of functional constraint prediction, it may be desirable to exclude/downweight hypermutable sites (such as CpG sites) and non-functional regions (such as certain classes of repetitive elements).

Model architecture

CNN models [38–41] have been widely used in genomics for supervised tasks prior to the emergence of the Transformer architecture [1]. CNNs are particularly effective at capturing local dependencies and motifs within genomic sequences through their ability to apply filters across the input data. These models have been successful in predicting DNA–protein binding sites, regulatory elements, and TFBS. GPN [13], the aforementioned gLM for genome-wide variant effect prediction in *A. thaliana*, took inspiration from the success of language models with modified CNN layers in NLP [62] and protein modeling [63] and replaced self-attention layers in a Transformer encoder with dilated CNN layers.

Transformer models have revolutionized various ML domains, particularly in NLP [1], and have recently been widely adopted for genomics modeling. The self-attention mechanism allows each token to attend to all positions in the input sequence simultaneously, enabling the model to dynamically focus on relevant parts of the sequence. This capability has led to significant advancements in detecting regulatory mechanisms for supervised gene expression tasks [22,64].

Despite their strengths, Transformer models face several challenges unique to genomic modeling. One significant issue is that Transformers have weak or no inductive biases regarding the locality of interactions [65,66], making them less data-efficient at modeling local motifs such as TFBS. This motivated the development of CNN-Transformer hybrids such as LOGO [67], following supervised models such as Enformer [22].

Another challenge is the context length: the self-attention mechanism results in computational time and memory scaling quadratically with the input sequence length, making it impractical to apply Transformers to very long genomic sequences [68]. Consequently, the longest input length that conventional attention-based gLMs can handle so far is 12 kb for NT-v2 [16]. To address this limitation, several Transformer-based gLMs have implemented approximate attention or hierarchical attention methods that sacrifice full pairwise attention between all tokens. These methods include the use of sparse attention [65] in GENA-LM [69], which extends the context length to 36 kb, and the MEGABYTE sub-quadratic hierarchical self-attention [70] employed in MegaDNA [36], achieving a context length of 96 kb.

To overcome the quadratic scaling issues of self-attention, various SSMs [71–73] have been proposed for gLMs as efficient alternatives to Transformers, offering nearly linear scaling with sequence length. HyenaDNA [35], based on the Hyena Hierarchy [72], can support input contexts as long as one million nucleotides. EVO [31], a hybrid model combining Hyena and Transformer architectures, is pretrained with 8 kb sequences and later fine-tuned with 131 kb sequences during the context extension stage. Caduceus [24], built on the Mamba-based SSM [73], is trained on 131 kb sequences while incorporating reverse-complement equivariance.

Learning objective

As described in Box 1, the MLM task (sometimes also called ‘masked token prediction’) involves predicting the identities of tokens randomly omitted from sequences with a predetermined probability (a common choice is 15%) given the remaining tokens. This framework has been used to train the seminal LLM BERT [74] and pLM ESM-1b [75] and has since been widely used for training gLMs. The CLM task (also referred to as ‘autoregressive language modeling’ or ‘next token prediction’) involves predicting the identities of tokens in sequences given their preceding tokens; it has been used to train the GPT series of LLMs [25]. In this task, the model predicts the next token given the previous tokens in a unidirectional, left-to-right order. A commonality between these two tasks is that they require models to predict components of data given other components as context. To generalize on these tasks, models must learn low-dimensional representations of the data. This capability enables the gLMs to understand and generate genomic sequences by capturing the underlying patterns and dependencies within the genome. In protein modeling, MLM tends to achieve better representations and transfer learning capabilities than CLM [76]. However, CLMs are the traditional choice for generation tasks, but excellent results have been recently obtained with MLMs via progressive unmasking [77,78].

To reduce input sequence length and model longer context, both *k*-mer and byte-pair encoding [79] (BPE) tokenizations create artificially defined nucleotide vocabularies larger than the natural nucleotide vocabularies of {A, C, G, T}. However, single-nucleotide tokenization simplifies model interpretation and attribution and enhances the model's ability to handle genomic variations more effectively.

Several modifications to the training objective have been explored to provide additional signal and boost performance. For instance, GPN-MSA [17] enhances MLM training on the human reference genome with a whole-genome MSA [80,81] of vertebrate species, leveraging conservation across related species for additional context. A limitation is that whole-genome MSAs have only been generated for certain species and might require further development to be effective in plants [82]. Additionally, *cis*-regulatory elements tend to diverge fast in sequence space even if they have conserved activity, which limits the orthology information that can be extracted via alignment [83]. Species LM [58] directly integrates species information by assigning a dedicated token for each yeast species and appending the species token to the input sequence during training and inference. Pretraining on nucleotide sequences has been expanded to enable crosstalk with additional modalities such as epigenetics [84,85], RNA [86,87], proteins [86–88], and natural language [89].

Interpretation

Deep learning models, while having achieved remarkable performance in various prediction tasks, typically lack interpretability and are often used as 'black boxes'. However, understanding how these models generate such predictions is crucial for enabling broader applications and advancing model development. As a result, a series of methods have been developed to interpret deep learning models, including those specific to genomics [90–92]. While the interpretation of gLMs is still an emerging line of research, several models have been shown to have learned meaningful biological patterns.

The sequence embeddings extracted from language models are commonly used as representations that capture rich contextual information and sequence features. Unsupervised clustering of the encoded sequence embeddings from gLMs has shown distinct clusters of input sequences that correspond to different genomic classes such as coding DNA sequence (CDS), intronic, untranslated region (UTR), etc. [13,16,35,93] (Figure 1D). Additionally, unsupervised clustering of SpliceBERT embeddings of canonical splice sites and non-splice GT/AG sites reveals distinct clusters that correspond to the two groups [55]. These results suggest that the models have learned to capture key contextual patterns that characterize functional elements in the genome.

The attention mechanism in the Transformer model is designed to capture the pattern of interaction between input tokens. Thus, interpreting the attention weights or the attention map for a given input sequence can reveal genomic features learned by the model. In SpliceBERT [55], attention weights between splice donors and acceptors are significantly higher than those between random pairs of sites; also, the strength of interaction tends to be higher within true donor–acceptor pairs compared with other combinations of donor and acceptor sites. These findings suggest that the model has learned the relationship between functionally interacting sites.

The nucleotide reconstruction approach has also been used in several gLMs to discover sequence motifs learned by the models. Specifically, individual positions of the input sequence are masked one at a time and the probability distribution of the nucleotides is predicted by the trained model given the genomic context. The obtained distribution at each site can reveal motifs learned by the model. This approach has been used in GPN to find notable patterns in the distribution of the reconstructed nucleotides. In particular, the model's predictions are generally more confident in functionally important sites. For example, coding sequences and splice donor/ acceptor sites are typically predicted with higher confidence than deep intronic sites. Moreover, within coding sequences, the third nucleotide position of a codon, the least determinant of the translated amino acid, is typically predicted with lower confidence than the first two nucleotide positions. Adapting TF-MoDISco [94], a dedicated tool to identify novel TFBS using model predictions, the authors also found sequence motifs that match known ones in TFBS databases and relevant literature [13] (Figure 1A). Similarly, the reconstructed sequence motifs from Species LM [58] also match the binding sites of known DNA- and RNA-binding proteins in species that are unseen during training, with the fidelity of motif reconstruction depending on the context and genomic regions that correctly reflect the *in vivo* binding sites. Furthermore, the reconstructed motifs' composition, existence, and location exhibit species-specific patterns, which suggests gLM as a potentially powerful tool for investigating the evolution of sequence motifs and regulatory code.

More recently, the dependency between genomic positions learned by a gLM was studied by introducing point mutations at a position and quantifying the changes in nucleotide probabilities at other positions [19]. Nucleotide dependency analysis revealed learned interactions within and across functional elements such as TFBS, splice sites, and RNA, including known secondary and tertiary structure contacts. Notably, nucleotide dependency analysis was able to detect bound TFBS more robustly than the previous approach based on predicted marginal probability distributions.

Evaluation

In this section, we discuss how models' performance can be benchmarked in regards to the three application areas described earlier: predicting functional constraints on alleles, generating novel viable sequences, and transfer learning.

There are various types of data that may be used to evaluate how well variant effect predictors can identify sites that are deleterious – or, in other words, that are under negative selection. One type of data are assays that couple functional differences between genetic variants to readouts (such as the expression of a reporter gene or cell growth) [95–97]. These readouts may be used to rank variants by their functionality, and since variants that affect function also tend to be under selection, we should expect that these ranks should correlate with ranks obtained from models' predictions. One source for these data is ProteinGym, a widely-used collection of experimental data that may be used to benchmark missense variant effect predictors [98]. Another type of data are clinical labels indicating whether variants have evidence of pathogenicity – that is, can elevate the risk for diseases. Pathogenic variants may affect fecundity and, therefore, be deleterious. As a result, we can benchmark variant effect predictors by evaluating them as pathogenicity classifiers. In human genetics, primary sources of clinical labels for variants include the ClinVar [99], HGMD [100], and OMIM [101] databases. A third type of data are variant frequencies. Since common variants are unlikely to be highly deleterious [102], their predicted level of constraint should be relatively higher than those of rare variants. Therefore, we may benchmark predictors based on how well they identify common variants. This evaluation is recommended by the authors of CADD [103]. A primary source of data on human allele frequencies in various ancestry groups is the gnomAD database [104]. Altogether, these data may be used as separate lines of evidence for models' generalization performance.

An issue with the evaluation of variant effect predictors is that the relationship between validation data and functional constraint can be murky. As a consequence, models can excel at benchmarks by exploiting the ways in which data fail to capture functional constraint. For example, a critical issue with using clinical labels is that variants are classified based on whether there is ample evidence that they are benign or pathogenic [105]. Since predictors can also utilize this evidence, their benchmarked performance on labeled variants may not reflect their true performance on unlabeled variants. (See Box 3 for a brief discussion of generalization performance.) There are also critical issues with using allele frequency data. For one, in addition to the direct action of natural selection, allele frequencies are influenced by factors such as mutation rates, drift, background selection, and hitchhiking [106]. As a result, predictors may perform well on benchmarks by predicting the effects of these processes instead of functional constraints. These issues highlight a need to carefully interpret the causes of predictors' performance and they have led to calls for greater transparency on which data and methods are used to train predictors [107]. Additionally, the variant labels obtained from these data typically do not account for the possibility that the deleteriousness of a variant can be influenced by epistasis and dominance.

There are a separate set of challenges with the evaluation of generative sequence models. A basic way to evaluate language models' generative capabilities is to compare their perplexities on sets of valid sequences. However, to evaluate models' capability to design novel sequences, it is necessary to gauge whether they can identify sequences that are both viable and novel. For this reason, models' perplexities on test sets may not reliably indicate their utility for design. Instead, a holistic approach that examines a broad range of properties of generated sequences may be warranted. For instance, Polygraph [108], a recent benchmark for regulatory sequence design, proposes a series of analyses that investigate sequence composition, motif patterns, and predicted functional activity. For whole-genome or chromosome design tasks, it may also be necessary to evaluate the existence and positioning of essential genes and functional regulatory elements, as well as the interactions between them. Ultimately, the designed sequences should be experimentally evaluated to determine if they perform their desired functions.

Last, there is a unique challenge with the evaluation of gLMs for transfer learning: any set of benchmarks – perhaps, in conjunction – must reliably indicate a model's performance on relevant tasks. A type of data that may be fashioned into a set of tasks broadly informative of models' adaptability to genome interpretation are functional genomics data (such as those from the ENCODE [109] or Roadmap Epigenomics [110] projects), which may be used to annotate genomic regions and variants. We should expect that a model's performance on predicting these annotations from genome sequences after adaptation is indicative of their capability to identify functionally similar genomic elements. To facilitate comparison between models, these annotations have been consolidated into various standardized sets of training and test data [16,48,56,111].

Box 3. Evaluating generalization performance

The purpose of evaluating predictive models is to build trust in their capability to generalize – that is, to make satisfactory predictions for unlabeled data. A straightforward and standard way to estimate the generalization performance of a model is to evaluate its accuracy on a 'test set' of labeled data that are representative of unlabeled data of interest [129]. This approach is the basis of most machine learning (ML) benchmarks.

Importantly, for this evaluation to be a reliable indicator of generalization performance, models must not be provided any information that may be used to differentiate test set data from the data they will ultimately be deployed on. Otherwise, they may decrease their test set error at the expense of their generalization performance. For this reason, ML contests that withhold their test data from participants are routinely organized [130–132].

As transfer learning benchmarks help highlight limitations of current models and establish criteria for publication, they are likely to be important assets for gLM developers and users. However, despite differences in current benchmarks' choice of tasks and methodologies, they provide seemingly redundant insight into gLMs' capabilities. Moving forward, it will be incumbent on the computational genomics community to develop standardized and extensible benchmarks that are widely trusted.

Concluding remarks and future perspectives

In an age of a vast and growing number of genomic sequences, gLMs are emerging as powerful tools to extract complex patterns useful for numerous applications, including functional constraint estimation, sequence design, and transfer learning, but much work lies ahead. Often, gLMs are claimed to be 'foundation models', a term coined to describe models trained on broad data that can be adapted to a wide range of downstream tasks [112]. The introduction of this term has been criticized since 'foundation' has the connotation that pretraining is necessary for solid performance, which is an empirical question, not an inherent property of all pretraining schemes in all domains [113]. This criticism rings even louder in new domains such as genomics, where establishing adequate benchmarks is likely to take some time. We are optimistic that gLMs can achieve a foundational role in future genomic applications, but further efforts are needed to develop suitable models and demonstrate their capabilities and utility rigorously.

While earlier gLMs tend to be more or less direct adaptations from NLP models, we expect that further contextualization with deep genomics expertise will reap the highest rewards. We note that evaluating the capabilities of gLMs is challenging because metrics may be misleading, especially when over-optimized. A boon for NLP is that humans are experts in natural language and, therefore, can calibrate benchmarks to match their expertise. In genomics, however, we must rely on data and expert domain knowledge to falsify models. This aspect of the problem makes it especially challenging and may highlight a need for engagement with subject-matter experts and deliberate experimentation for the sake of developing benchmarks. We conclude this review with a few research directions (see Outstanding questions) that we believe warrant further investigation.

Outstanding questions

How can we best model patterns across a wide range of scales, from motifs to genes to whole genomes?

For which applications is it important to model long-range interactions and how does one determine a suitable size of the receptive field?

How can we incorporate structural variations into gLMs?

What is the best way to utilize population genetic data when training gLMs?

How can we best integrate gLMs with other complex modalities, such as transcriptomic and epigenetic data?

For developing gLMs, can we better understand what makes some genomes harder to model than others?

Will the scaling hypothesis hold for gLMs and for how long? Are there really that much data available, considering that most may be non-functional?

Acknowledgments

This work is supported in part by an National Institutes of Health (NIH) grant R35-GM134922, a grant from the Koret-UC Berkeley-Tel Aviv University Initiative in Computational Biology and Bioinformatics, and a grant from the Noyce Initiative UC Partnerships in Computational Transformation Program.

Declaration of interests

The authors declare no competing interests.

References

1. Vaswani, A. *et al.* (2017) Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30) (Guyon, I. *et al.*, eds), Curran Associates
2. Gulati, A. *et al.* (2020) Conformer: Convolution-augmented transformer for speech recognition. *arXiv*, Published online May 16, 2020. https://arxiv.org/abs/2005.08100
3. Achiam, J. *et al.* (2023) GPT-4 technical report. *arXiv*, Published online March 4, 2024. https://arxiv.org/abs/2303.08774
4. Bateman, A. *et al.* (2023) UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Res.* 51, D523–D531
5. Lin, Z. *et al.* (2023) Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 379, 1123–1130
6. Meier, J. *et al.* (2021) Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems* (Vol. 34) (Ranzato, M. *et al.*, eds), pp. 29287–29303, Curran Associates
7. Truong Jr., T. and Bepler, T. (2023) PoET: a generative model of protein families as sequences-of-sequences. In *Advances in Neural Information Processing Systems* (Vol. 36) (Oh, A. *et al.*, eds), pp. 77379–77415, Curran Associates
8. Bepler, T. and Berger, B. (2021) Learning the protein language: evolution, structure, and function. *Cell Syst.* 12, 654–669
9. Ruffolo, J.A. and Madani, A. (2024) Designing proteins with language models. *Nat. Biotechnol.* 42, 200–202
10. Riesselman, A.J. *et al.* (2018) Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* 15, 816–822
11. Frazer, J. *et al.* (2021) Disease variant prediction with deep generative models of evolutionary data. *Nature* 599, 91–95

12. Brandes, N. *et al.* (2023) Genome-wide prediction of disease variant effects with a deep protein language model. *Nat. Genet.* 55, 1512–1522
13. Benegas, G. *et al.* (2023) DNA language models are powerful predictors of genome-wide variant effects. *Proc. Natl. Acad. Sci. U. S. A.* 120, e2311219120
14. Mendoza-Revilla, J. *et al.* (2024) A foundational large language model for edible plant genomes. *Commun. Biol.* 7, 835
15. Zhai, J. *et al.* (2024) Cross-species plant genomes modeling at single nucleotide resolution using a pre-trained DNA language model. *bioRxiv*, Published online August 22, 2024. https://doi.org/10.1101/2024.06.04.596709
16. Dalla-Torre, H. *et al.* (2023) Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nat. Methods*, Published online November 28, 2024. https://doi.org/10.1038/s41592-024-02523-z
17. Benegas, G. *et al.* (2025) A DNA language model based on multispecies alignment predicts the effects of genome-wide variants. *Nat. Biotechnol.*, In press
18. Hsu, C. *et al.* (2022) Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.* 40, 1114–1122
19. Tomaz da Silva, P. *et al.* (2024) Nucleotide dependency analysis of DNA language models reveals genomic functional elements. *bioRxiv*, Published online July 27, 2024. https://doi.org/10.1101/2024.07.27.605418
20. Siepel, A. *et al.* (2005) Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res.* 15, 1034–1050
21. Pollard, K.S. *et al.* (2010) Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.* 20, 110–121
22. Avsec, Ž. *et al.* (2021) Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* 18, 1196–1203
23. Jaganathan, K. *et al.* (2019) Predicting splicing from primary sequence with deep learning. *Cell* 176, 535–548
24. Schiff, Y. *et al.* (2024) Caduceus: Bi-directional equivariant long-range DNA sequence modeling. *arXiv*, Published online June 5, 2024. https://doi.org/10.48550/arXiv.2403.03234
25. Brown, T. *et al.* (2020) Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (Vol. 33) (Larochelle, H. *et al.*, eds), pp. 1877–1901, Curran Associates
26. Madani, A. *et al.* (2023) Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.* 41, 1099–1106
27. Ingraham, J. *et al.* (2019) Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems* (Vol. 32) (Wallach, H. *et al.*, eds), Curran Associates
28. Hsu, C. *et al.* (2022) Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning International Conference on Machine Learning*, pp. 8946–8970, PMLR
29. Shin, J.-E. *et al.* (2021) Protein design and variant prediction using autoregressive generative models. *Nat. Commun.* 12, 2403
30. Lal, A. *et al.* (2024) regLM: designing realistic regulatory DNA with autoregressive language models. In *International Conference on Research in Computational Molecular Biology*, pp. 332–335, Springer
31. Nguyen, E. *et al.* (2024) Sequence modeling and design from molecular to genome scale with Evo. *Science* 386, eado9336
32. Wang, Y. *et al.* (2020) Synthetic promoter design in *Escherichia coli* based on a deep generative network. *Nucleic Acids Res.* 48, 6403–6412
33. Jores, T. *et al.* (2021) Synthetic promoter designs enabled by a comprehensive analysis of plant core promoters. *Nat. Plants* 7, 842–855
34. de Almeida, B.P. *et al.* (2022) DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. *Nat. Genet.* 54, 613–624
35. Nguyen, E. *et al.* (2023) HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution. In *Advances in Neural Information Processing Systems* (Vol. 36) (Oh, A. *et al.*, eds), pp. 43177–43201, Curran Associates
36. Shao, B. (2023) A long-context language model for deciphering and generating bacteriophage genomes. *Nat. Commun.* 15, 9392
37. Ratcliff, J.D. (2024) Transformer model generated bacteriophage genomes are compositionally distinct from natural sequences. *NAR Genom. Bioinform.* 6, lqae129
38. Alipanahi, B. *et al.* (2015) Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat. Biotechnol.* 33, 831–838
39. Zhou, J. and Troyanskaya, O.G. (2015) Predicting effects of noncoding variants with deep learning–based sequence model. *Nat. Methods* 12, 931–934
40. Kelley, D.R. *et al.* (2016) Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* 26, 990–999
41. Kelley, D.R. *et al.* (2018) Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res.* 28, 739–750
42. Zeng, T. and Li, Y.I. (2022) Predicting RNA splicing from DNA sequence using Pangolin. *Genome Biol.* 23, 103
43. West-Roberts, J. *et al.* (2024) Diverse genomic embedding benchmark for functional evaluation across the tree of life. *bioRxiv*, Published online July 16, 2024. https://doi.org/10.1101/2024.07.10.602933
44. de Almeida, B.P. *et al.* (2024) SegmentNT: annotating the genome at single-nucleotide resolution with DNA foundation models. *bioRxiv*, Published online March 15, 2024. https://doi.org/10.1101/2024.03.14.584712
45. Zhou, Z. *et al.* (2024) DNABERT-S: Learning species-aware dna embedding with genome foundation models. *arXiv*, Published online October 22, 2024 https://doi.org/10.48550/arXiv.2402.08777
46. Zhou, Z. *et al.* (2023) DNABERT-2: Efficient foundation model and benchmark for multi-species genome. *arXiv*, Published online June 26, 2023. https://doi.org/10.48550/arXiv.2306.15006
47. Garau-Luis, J.J. *et al.* (2024) Multi-modal transfer learning between biological foundation models. *arXiv*, Published online June 20, 2024. https://doi.org/10.48550/arXiv.2406.14150
48. Marin, F.I. *et al.* (2024) BEND: benchmarking DNA language models on biologically meaningful tasks. *arXiv*, Published online April 9, 2024. https://doi.org/10.48550/arXiv.2311.12570
49. Tang, Z. and Koo, P.K. (2024) Evaluating the representational power of pre-trained DNA language models for regulatory genomics. *bioRxiv*, Published online September 25, 2024. https://doi.org/10.1101/2024.02.29.582810
50. Li, F.-Z. *et al.* (2024) Feature reuse and scaling: understanding transfer learning with protein language models. *bioRxiv*, Published online February 14, 2024. https://doi.org/10.1101/2024.02.05.578959
51. Gao, L. *et al.* (2020) The Pile: an 800GB dataset of diverse text for language modeling. *arXiv*, Published online December 31, 2020. https://doi.org/10.48550/arXiv.2101.00027
52. Longpre, S. *et al.* (2024) The responsible foundation model development cheatsheet: a review of tools & resources. *arXiv*, Published online https://doi.org/10.48550/arXiv.2406.16746
53. Sullivan, P.F. *et al.* (2023) Leveraging base-pair mammalian constraint to understand genetic variation and human disease. *Science* 380, eabn2937
54. Lee, K. *et al.* (2022) Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Muresan, S. *et al.*, eds), pp. 8424–8445, Association for Computational Linguistics
55. Chen, K. *et al.* (2024) Self-supervised learning on millions of primary RNA sequences from 72 vertebrates improves sequence-based RNA splicing prediction. *Brief. Bioinform.* 25, bbae163
56. Robson, E.S. and Ioannidis, N.M. (2023) GUANinE v1. 0: Benchmark Datasets for Genomic AI Sequence-to-Function Models. *bioRxiv*, Published online March 7, 2024. https://doi.org/10.1101/2023.10.12.562113
57. Carroll, S.B. (2005) Evolution at two levels: on genes and form. *PLoS Biol.* 3, e245
58. Karollus, A. *et al.* (2024) Species-aware DNA language models capture regulatory elements and their evolution. *Genome Biol.* 25, 83
59. Schoenfelder, S. and Fraser, P. (2019) Long-range enhancer–promoter contacts in gene expression control. *Nat. Rev. Genet.* 20, 437–455

60. Karnuta, J.M. and Scacheri, P.C. (2018) Enhancers: bridging the gap between gene control and human disease. *Hum. Mol. Genet.* 27, R219–R227
61. King, J.L. and Jukes, T.H. (1969) Non-darwinian evolution. *Science* 164, 788–798
62. Tay, Y. *et al.* (2021) Are pretrained convolutions better than pretrained transformers? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4349–4359, Association for Computational Linguistics
63. Yang, K.K. *et al.* (2024) Convolutions are competitive with transformers for protein sequence pretraining. *Cell Syst.* 15, 286–294
64. Linder, J. *et al.* (2023) Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. *bioRxiv*, Published online September 1, 2023. https://doi.org/10.1101/2023.08.30.555582
65. Zaheer, M. *et al.* (2020) Big Bird: Transformers for Longer Sequences. In *Advances in Neural Information Processing Systems* (Vol. 33) (Larochelle, H. *et al.*, eds), pp. 17283–17297, Curran Associates
66. Su, J. *et al.* (2024) Roformer: enhanced transformer with rotary position embedding. *Neurocomputing* 568, 127063
67. Yang, M. *et al.* (2022) Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution. *Nucleic Acids Res.* 50, e81
68. Dai, Z. *et al.* (2019) Transformer-XL: attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (Korhonen, A. *et al.*, eds), pp. 2978–2988, Association for Computational Linguistics
69. Fishman, V. *et al.* (2023) GENA-LM: a family of open-source foundational models for long DNA sequences. *bioRxiv*, Published online August 23, 2024. https://doi.org/10.1101/2023.06.12.544594
70. Yu, L. *et al.* (2023) MEGABYTE: Predicting million-byte sequences with multiscale transformers. In *Advances in Neural Information Processing Systems* (Vol. 36) (Oh, A. *et al.*, eds), pp. 78808–78823, Curran Associates
71. Gu, A. *et al.* (2022) Efficiently modeling long sequences with structured state spaces. *arXiv*, Published online August 5, 2022, https://doi.org/10.48550/arXiv/2111.00396
72. Poli, M. *et al.* (2023) Hyena Hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078, PMLR
73. Gu, A. and Dao, T. (2023) Mamba: Linear-time sequence modeling with selective state spaces. *arXiv*, Published online May 31, 2024. https://doi.org/10.48550/arXiv.2312.00752
74. Devlin, J. *et al.* (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Burstein, J. *et al.*, eds), pp. 4171–4186, Association for Computational Linguistics
75. Rives, A. *et al.* (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.* 118, e2016239118
76. Cheng, X. *et al.* (2024) Training compute-optimal protein language models. *bioRxiv*, Published online June 9, 2024. https://doi.org/10.1101/2024.06.06.597716
77. Samuel, D. (2024) BERTs are generative in-context learners. *arXiv*, Published online October 31, 2024. https://doi.org/10.48559/arXiv.2406.04823
78. Hayes, T. *et al.* (2024) Simulating 500 million years of evolution with a language model. *bioRxiv*, Published online July 2, 2024. https://doi.org/10.1101/2024.07.01.600583
79. Sennrich, R. *et al.* (2016) Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Erk, K. and Smith, N.A., eds), pp. 1715–1725, Association for Computational Linguistics
80. Blanchette, M. *et al.* (2004) Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.* 14, 708–715
81. Armstrong, J. *et al.* (2020) Progressive Cactus is a multiple-genome aligner for the thousand-genome era. *Nature* 587, 246–251
82. Song, B. *et al.* (2024) New whole-genome alignment tools are needed for tapping into plant diversity. *Trends Plant Sci.* 29, 355–369
83. Phan, M.H. *et al.* (2024) Conservation of regulatory elements with highly diverged sequences across large evolutionary distances. *bioRxiv*, Published online May 14, 2024. https://doi.org/10.1101/2024.05.13.590087
84. Mo, S. *et al.* (2021) Multi-modal self-supervised pre-training for large-scale genome data. *arXiv*, Published online November 3, 2021. https://doi.org/10.48550/arXiv.2110.05231
85. Trotter, M.V. *et al.* (2021) Epigenomic language models powered by Cerebras. *arXiv*, Published online December 14, 2021. https://doi.org/10.48550/arXiv.2112.07571
86. He, Y. *et al.* (2024) LucaOne: generalized biological foundation model with unified nucleic acid and protein language. *bioRxiv*, Published online May 14, 2024. https://doi.org/10.1101/2024.05.10.592927
87. Zhu, X. *et al.* (2024) CD-GPT: a biological foundation model bridging the gap between molecular sequences through central dogma. *bioRxiv*, Published online June 28, 2024. https://doi.org/10.1101/2024.06.24.600337
88. Cornman, A. *et al.* (2024) The OMG dataset: an Open MetaGenomic corpus for mixed-modality genomic language modeling. *bioRxiv*, Published online August 17, 2024. https://doi.org/10.1101/2024.08.14.607850
89. Richard, G. *et al.* (2024) ChatNT: multimodal conversational agent for DNA, RNA and protein tasks. *bioRxiv*, Published online September 11, 2024. https://doi.org/10.1101/2024.04.30.591835
90. Linardatos, P. *et al.* (2020) Explainable AI: a review of machine learning interpretability methods. *Entropy* 23, 18
91. Zhang, Y. *et al.* (2021) A survey on neural network interpretability. *IEEE Trans. Emerg. Topics Comput. Intell.* 5, 726–742
92. Talukder, A. *et al.* (2021) Interpretation of deep learning in genomics and epigenomics. *Brief. Bioinform.* 22, bbaa177
93. Ji, Y. *et al.* (2021) DNABERT: pre-trained bidirectional encoder representations from Transformers model for DNA-language in genome. *Bioinformatics* 37, 2112–2120
94. Shrikumar, A. *et al.* (2018) Technical note on transcription factor motif discovery from importance scores (TF-MoDISco) version 0.5. 6.5. *arXiv*, Published online October 31, 2018. https://doi.org/10.1101/arxiv.1811.00416
95. Fowler, D.M. *et al.* (2023) An atlas of variant effects to understand the genome at nucleotide resolution. *Genome Biol.* 24, 147
96. Kircher, M. *et al.* (2019) Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. *Nat. Commun.* 10. https://doi.org/10.1038/s41467-019-11526-w
97. Findlay, G.M. *et al.* (2018) Accurate classification of BRCA1 variants with saturation genome editing. *Nature* 562, 217–222
98. Notin, P. *et al.* (2023) ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design ProteinGym: large-acale benchmarks for protein fitness prediction and design. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, NeurIPS
99. Landrum, M.J. *et al.* (2016) ClinVar: public archive of interpretations of clinically relevant variants. *Nucleic Acids Res.* 44, D862–D868
100. Stenson, P.D. *et al.* (2017) The Human Gene Mutation Database: towards a comprehensive repository of inherited mutation data for medical research, genetic diagnosis and next-generation sequencing studies. *Hum. Genet.* 136, 665–677
101. Amberger, J.S. *et al.* (2015) OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic Acids Res.* 43, D789–D798
102. Pritchard, J.K. and Cox, N. (2002) The allelic architecture of human disease genes: common disease–common variant…or not? *Hum. Mol. Genet.* 11, 2417–2423
103. Rentzsch, P. *et al.* (2019) CADD: predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Res.* 47, D886–D894

104. Karczewski, K.J. *et al.* (2020) The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* 581, 434–443
105. Grimm, D.G. *et al.* (2015) The evaluation of tools used to predict the impact of missense variants is hindered by two types of circularity. *Hum. Mutat.* 36, 513–523
106. Hartl, D.L. *et al.* (1997) *Principles of Population GeneticsVol. 116 Vol. 116. Sinauer Associates*
107. Livesey, B.J. *et al.* (2024) Guidelines for releasing a variant effect predictor. *arXiv*, Published online April 16, 2024. https://arXiv:2404.10807v1
108. Gupta, A. *et al.* (2023) Polygraph: a software framework for the systematic assessment of synthetic regulatory DNA elements. *bioRxiv*, Published online November 27, 2023. https://doi.org/10.1101/2023.11.27.568764
109. The ENCODE Project Consortium (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57–74
110. Kundaje, A. *et al.* (2015) Integrative analysis of 111 reference human epigenomes. *Nature* 518, 317–330
111. Grešová, K. *et al.* (2023) Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data* 24, 25
112. Bommasani, R. *et al.* (2021) On the opportunities and risks of foundation models. *arXiv*, Published online August 16, 2021. https://doi.org/10.48550/arxiv.2108.07258
113. Helfrich, G. (2024) The harms of terminology: why we should reject so-called "frontier AI". *AI Ethics* 4, 699–705
114. Zhang, Y. *et al.* (2016) Jointly characterizing epigenetic dynamics across multiple human cell types. *Nucleic Acids Res.* 44, 6721–6731
115. Hoarfrost, A. *et al.* (2022) Deep learning of a bacterial and archaeal universal language of life enables transfer learning and illuminates microbial dark matter. *Nat. Commun.* 13, 2606
116. Gwak, H.-J. and Rho, M. (2022) ViBE: a hierarchical BERT model to identify eukaryotic viruses using metagenome sequencing data. *Brief. Bioinform.* 23, bbac204
117. Levy, B. *et al.* (2022) FloraBERT: cross-species transfer learning withattention-based neural networks for gene expression prediction. *Res. Sq.*, Published online August 30, 2022. https://doi.org/10.21203/rs.3.rs-1927200/v1
118. Bai, Z. *et al.* (2022) Identification of bacteriophage genome sequences with representation learning. *Bioinformatics* 38, Btac509
119. Zvyagin, M. *et al.* (2023) GenSLMs: genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. *Int. J. High Perform. Comput. Appl.* 37, 683–705
120. Sanabria, M. *et al.* (2023) The human genome's vocabulary as proposed by the DNA language model GROVER. *bioRxiv*, Published online July 19, 2023. https://doi.org/10.1101/2023.07.19.549677
121. Zhang, D. *et al.* (2023) DNAGPT: a generalized pretrained tool for multiple DNA sequence analysis tasks. *bioRxiv*, Published online July 12, 2023. https://doi.org/10.1101/2023.07.11.548628
122. Chu, Y. *et al.* (2024) A 5' UTR language model for decoding untranslated regions of mRNA and function predictions. *Nat. Mach. Intell.* 6, 449–460
123. Lorenz, R. *et al.* (2011) ViennaRNA Package 2.0. *Algorithm. Mol. Biol.* 6, 1–14
124. Raffel, C. *et al.* (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 1–67
125. Kudo, T. (2018) Subword regularization: improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Gurevych, I. and Miyao, Y., eds), pp. 66–75, Association for Computational Linguistics
126. Chiang, W.-L. *et al.* (2023) *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*, UC Berkeley Sky Computing
127. Markowitz, V.M. *et al.* (2012) IMG: the integrated microbial genomes database and comparative analysis system. *Nucleic Acids Res.* 40, D115–D122
128. Richardson, L. *et al.* (2023) MGnify: the microbiome sequence data analysis resource in 2023. *Nucleic Acids Res.* 51, D753–D759
129. Vapnik, V.N. (1999) *The Nature of Statistical Learning Theory*, Springer
130. Russakovsky, O. *et al.* (2015) ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis. (IJCV)* 115, 211–252
131. Moult, J. *et al.* (2018) Critical assessment of methods of protein structure prediction (CASP)—round XII. *Proteins Struct. Funct. Bioinform.* 86, 7–15
132. Johnson, A.D. *et al.* (2017) CAGI: the critical assessment of genome interpretation. *Genome Biol.* 18, 1–5