nature portfolio

https://doi.org/10.1038/s41586-026-10176-5

Supplementary information

Genome modelling and design across all domains of life with Evo 2

In the format provided by the authors and unedited

Genome modeling and design across all domains of life with Evo 2

Supplementary Information

Garyk Brixi[*,1,2,3], Matthew G. Durrant[*,1,2], Jerome Ku[*,1,2], Mohsen Naghipourfar[*,1,2,7], Michael Poli[*,2,3,5], Gwanggyu Sun[*,1,2], Greg Brockman[2,6,§], Daniel Chang[1,2,3], Alison Fanton[1,2], Gabriel A. Gonzalez[1,2], Samuel H. King[1,2,3], David B. Li[1,2,3], Aditi T. Merchant[1,2,3], Eric Nguyen[2,3], Chiara Ricci-Tam[1,2], David W. Romero[2,4], Jonathan C. Schmok[1,2], Ali Taghibakhshi[2,4], Anton Vorontsov[2,4], Brandon Yang[2,6], Myra Deng[8], Liv Gorton[8], Nam Nguyen[8], Nicholas K. Wang[8], Michael T. Pearce[8], Elana Simon[8], Etowah Adams[9], Zachary J. Amador[11], Euan A. Ashley[3], Stephen A. Baccus[3], Haoyu Dai[3], Steven Dillmann[3], Stefano Ermon[3], Daniel Guo[1,3], Michael H. Herschl[1,7], Rajesh Ilango[1], Ken Janik[4], Amy X. Lu[7], Reshma Mehta[1], Mohammad R.K. Mofrad[7], Madelena Y. Ng[3], Jaspreet Pannu[12], Christopher Ré[3], John St. John[4], Jeremy Sullivan[1], Joseph Tey[1,3], Ben Viggiano[3], Kevin Zhu[7], Greg Zynda[4], Daniel Balsam[8], Patrick Collison[1], Anthony B. Costa[4], Tina Hernandez-Boussard[3], Eric Ho[8], Ming-Yu Liu[4], Thomas McGrath[8], Kimberly Powell[4], Sudarshan Pinglay[2,11], Dave P. Burke[‡,1,2], Hani Goodarzi[‡,1,2,10], Patrick D. Hsu[‡,†,1,2,7], Brian L. Hie[‡,†,1,2,3]

[1]Arc Institute, Palo Alto, CA, USA; [2]Evo 2 Core Team, Palo Alto, CA, USA; [3]Stanford University, Stanford, CA, USA; [4]NVIDIA, Santa Clara, CA, USA; [5]Liquid AI, San Francisco, CA, USA; [6]Independent Researcher, San Francisco, USA; [7]University of California, Berkeley, Berkeley, CA, USA; [8]Goodfire, San Francisco, CA, USA; [9]Columbia University, New York, NY, USA; [10]University of California, San Francisco, San Francisco, CA, USA; [11]University of Washington, Seattle, WA, USA; [12]Johns Hopkins University, Baltimore, MD, USA

[*]These authors contributed equally to this work.
[‡]These authors jointly supervised this work.
[§]Present address: OpenAI, San Francisco, USA.
[†]Corresponding authors: P.D.H. (patrick@arcinstitute.org), B.L.H. (brianhie@stanford.edu, brian.hie@arcinstitute.org).

Supplementary Information Index

A. Methods

A.1. Training Evo 2

Evo 2 is trained using next-token-prediction on the byte-tokenized OpenGenome2 dataset. We train Evo 2 in two phases: a pretraining phase at 8192 token context focused more on functional elements and midtraining phase during which we extend up to 1M token context length with more entire genomes in the data mix. Evo 2 40B's pretraining stage is split into two stages, first training at 1024 context for 6.6T tokens before extending to 8192 context for 2.1T tokens. Additionally, we train and release a smaller, Evo 2 1B base at 8192 context length for 1T tokens. For efficiency, Evo 2 is trained using sequence packing. We trained Evo 2 40B on a cluster of 2048 NVIDIA H100 GPUs for over three months, with a training cost of approximately 2.25×10^{24} floating point operations (FLOPs). In addition to the code, we provide an extensive description of algorithms and engineering efforts that we developed to train long-context models at scale in [60].

Supplementary Table 4 provides an estimate of training FLOPS for Evo 2 40B and other large models spanning application domains of biology and natural language. We include Pythia Large [61], OLMo 2 Large [62], Evo 1, Falcon 1 180B [63], StarCoder 2 [64], DeepSeek V3 [65], Llama 3.1 Large [66], ESM3 Large [67] and xTrimo [68].

A.1.1. Model architecture

Evo 2 uses StripedHyena 2 [60], the first multi-hybrid architecture based on input-dependent convolutions [69, 70]. Multi-hybrids combine various different operators to balance model quality with training and inference efficiency, in line with findings of [71] and [72]. **Extended Data Fig. 1b** provides a schematic of each new convolutional operator in the architecture. Self-attention layers use rotary positional embeddings [73].

Model size information and hyperparameters used for Evo 2 models are shown in **Supplementary Table 1**. Each Evo 2 model uses a pattern of Hyena-SE, Hyena-MR and Hyena-LI, and attention, with the number of repetitions scaling with model size. Hyena-SE uses inner filters of length 7, Hyena-MR length 128. GELU activations are only used for the first layer, followed by no activations. We use weight tying between embedding and final projection layers.

	Evo 2 40B	Evo 2 7B	Evo 2 1B base
Parameters	40.3B	6.5B	1.1B
Total Layers	50	32	25
Hidden Size	8,192	4,096	1,920
FFN Size	22,528	11,264	5,120
Num Heads	64	32	15
Total Tokens	9.3T	2.4T	1T

Supplementary Table 1 | Model architecture configurations for Evo 2 models

A.1.2. Loss function

Evo 2 is trained with a reweighted cross entropy loss, which weighs the loss contribution of repetitive portions of DNA by 0.1. This affects the genomic window and whole genome portions of the data which contain these annotations. This loss has been found in other DNA models to improve performance on downstream tasks and better calibrate likelihoods between repetitive and nonrepetitive DNA [74], which we found to be true for downstream tasks in a controlled comparison (**Appendix C.2**). The loss is

$$\ell_{\text{wCE}} = \frac{1}{Z} \sum_t w_t \ell_{\text{CE}}(t)$$

with weighting

$$w_t = \begin{cases} 0.1 & \text{if position } t \text{ is in repetitive region} \\ 1.0 & \text{otherwise} \end{cases}$$

$$Z = 0.1\ N_{\texttt{repeat}} + N_{\texttt{non\_repeat}}$$

where w_t is the weight applied to each position, $N_{\texttt{repeat}}$ represents the number of positions in repetitive regions within a batch and $N_{\texttt{non\_repeat}}$ is the number of non repetitive regions, and Z is the normalization factor that ensures consistent loss scaling regardless of the proportion of repetitive regions.

For any base pair, the model is always tasked with predicting the uppercase character. For the first 3T tokens of pretraining, lowercase tokens are input to the model to add information on which portions of DNA are repetitive. This was done to further help learn different representations for interspersed repeats, which are very common in many eukaryotic genomes. For additional pretraining and for all midtraining, all inputs to the model are uppercase. Loss is masked on special tokens used to condition the model that we do not want to generate, including the stitch tokens '@' and '#' , as well as the multi-token phylogenetic tags used during midtraining.

A.1.3. Pretraining infrastructure

Evo 2 was trained on Savanna (https://github.com/Zymrael/savanna), custom training infrastructure built with components from DeepSpeed, GPT-NeoX [75], and Transformer Engine. Our stack supports efficient pretraining of multi-hybrid models and new context parallel algorithms. We train our largest models with mixed precision, using a 3D mesh of data, tensor, and context parallelism, combined with ZeRO-3 [76]. During training, we use Transformer Engine's FP8 implementation for linear layers and RMSNorms.

A.1.4. Pretraining phase

Supplementary Table 2 provides information on our pretraining configuration. We refer to the models after pretraining but before context extension as Evo 2 40B and 7B base.

	Evo 2 40B base	Evo 2 7B base	Evo 2 1B base
Learning Rate	2.0e-4	3.0e-4	1.5e-4
Training Batch	16.8M	4.2M	2.1M
Total Iterations	516K	500K	490K
Pretraining Tokens	8.7T	2.1T	1T
Sequence length	1024 (6.6T), 8192 (2.1T)	8192	8192

Supplementary Table 2 | Training hyperparameters for Evo 2 models. Each model uses the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and cosine learning rate decay.

A.1.5. Midtraining phase: Context extension

We follow a multi-stage midtraining procedure, gradually extending the context length while keeping the same batch size as pretraining, adjusting model parallelism accordingly. Midtraining was performed on an adjusted data composition, including more whole genomes and with longer average sequence length (**Fig. 1, Supplementary Table 9**).

We explore two different rotary embedding-based methods to adapt to longer sequences: positional interpolation by down scaling the positional index of tokens [77] and increasing the base frequency of the RoPE embedding [78]. We decide on a combined approach using both together, with a 10× increase to the base frequency for every doubling of the context length. **Supplementary Table 3** provides information on our midtraining protocol.

After extension stages, model performance was evaluated using loss, performance on short sequence DMS tasks, and performance on our long context needle-in-a-haystack evaluation to evaluate effectiveness of the extension. We did not find significant differences between different extension protocols. Based on the success-

ful context extension results for the 7B, we reduced the number of stages and increase the number of tokens for the 40B, seeing 600B tokens during midtraining.

Context Length	Base Frequency	Scale Factor	Evo 2 7B Training Tokens	Evo 2 40B Training Tokens
32.8K	1.0×10^{6}	4	50B	-
65.5K	1.0×10^{7}	8	50B	-
131.1K	1.0×10^{8}	16	50B	200B
262.1K	1.0×10^{9}	32	50B	200B
524.3K	1.0×10^{10}	64	50B	-
1048.6K	1.0×10^{11}	128	50B	200B
			300B	600B

Supplementary Table 3 | Context extension protocol and number of tokens for Evo 2 7B and 40B

A.1.6. Midtraining phase: Needle-in-a-haystack evaluation

We developed a novel synthetic evaluation to assess the ability of DNA language models to identify and utilize a specific sequence pattern in its context to make predictions on a repeated sequence with the same pattern. This "needle-in-haystack" evaluation quantifies a model's capacity to retrieve sequence patterns within different context lengths. For each evaluation, we generated a random DNA sequence of 100 base pairs (bp) to serve as the "needle" sequence. A background sequence ("haystack") was constructed by sampling DNA base pairs uniformly at random at varying lengths following powers of two, ranging from 512 bp to 1,048,576 bp. The needle sequence was systematically inserted at different relative positions within each haystack, specifically at depths corresponding to 10% through 90%, at intervals of 10%, of the total haystack length. A "query" sequence, which is an exact duplicate of the needle sequence, was then placed at the suffix of the haystack sequence.

The evaluation methodology employs a modification of the "categorical Jacobian" analysis, as originally proposed by [79], to measure the model's use of the needle sequence to predict the query sequence. At a high-level, we mutate the needle and measure the effects on model predictions for the query as a way to assess retrieval. More formally, let $\mathbf{C}$ denote the categorical Jacobian matrix using the notation from [80], where the entry $\mathbf{C}[i, j]$ indicates the Euclidean magnitude of the difference in logits at position j when mutating position i to all tokens in the vocabulary. We compute a retrieval score

$$r = \frac{1}{N_{\text{needle}}} \sum_{i \in [0, N_{\text{needle}}-1]} \mathbf{C}[a_{\text{needle}} + i, a_{\text{query}} + i]$$

where $N_{\text{needle}} = 100$ is the length of the needle (and of the query) sequence, a_{needle} is the starting position of the needle and a_{query} is the starting position of the query. High values of r indicate greater retrieval strength. We use a threshold of $r \geq 0.8$ to determine successful retrieval, which was obtained by manually inspecting categorical Jacobian matrices of synthetic sequences containing repeated motifs. We computed the retrieval score for various haystack lengths and when inserting the needle at various depths into the haystack.

To quantify the statistical significance of our retrieval-score threshold of 0.8, we computed a null distribution of scores by randomly shuffling the needle sequence and computing the resulting retrieval score using the same methods described above. We performed this analysis at haystack lengths of 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536; at depths into context of 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the haystack sequence length; and 13 repeated shuffles. This resulted in a null distribution containing $8 \times 10 \times 13 = 1040$ values, of which the highest valued score is 0.037 (**Extended Data Fig. 1e**). At a retrieval score of 0.8 (our threshold of determining retrieval success), we can reject the null hypothesis of random retrieval with $P < 0.001$.

A.1.7. Inference infrastructure

Evo 2 inference runs on Vortex (https://github.com/Zymrael/vortex). Vortex contains infrastructure and efficient implementation for autoregressive generation with StripedHyena 2. For the new convolution

operators with finite inner filters, we adopt a caching strategy similar to KV caching in self-attention. For long filters, we switch to a recurrent form. All convolution operators in the architecture can generate autoregressively with a constant memory footprint.

A.2. OpenGenome2 training data

We significantly expanded upon the OpenGenome pretraining dataset, used to train Evo 1, to create OpenGenome2, increasing the total number of nucleotides from 300 billion to 8.84 trillion. This included a 33% expansion of representative prokaryotic genomes from 85,205 to 113,379 (357 billion nucleotides), a total of 6.98 trillion nucleotides from eukaryotic genomes, 854 billion nucleotides of non-redundant metagenomic sequencing data, 2.82 billion nucleotides of organelle genomes, and 602 billion nucleotides of subsets of eukaryotic sequence data to focus on likely functional regions of the genomes by focusing on different windows around coding genes. We introduced these data augmentations to prioritize genes and regions around genes to improve performance on downstream tasks (**Appendix C.3**).

A.2.1. Data curation

Reused datasets. Our previously published OpenGenome dataset was used in its entirety as part of the training data for this study [80]. This included representative prokaryotic genomes available through GTDB release v214.1, and curated phage and plasmid sequences retrieved through IMG/VR and IMG/PR. As previously described, the OpenGenome dataset was curated to exclude genomic sequences of viruses which infect eukaryotic hosts.

Updated prokaryotic genomes. New prokaryotic reference genomes made available through the GTDB release 220.0 [81] update were added to the training data for this study. New genomes were identified by selecting all species' reference genomes that had no previously published (release 214.1) genomes within their species cluster, resulting in 28,174 additional prokaryotic genomes.

Eukaryotic reference genomes. All available eukaryotic reference genomes were downloaded from NCBI on May 31, 2024, excluding atypical genomes, metagenome-assembled genomes, and genomes from large multi-isolate projects. This resulted in 16,704 genomes including an estimated 10.7 trillion nucleotides. Only contigs that were annotated as 'Primary Assembly', 'non-nuclear', or 'aGasCar1.hap1' (an aberrant annotation that applied only to GCA_027917425.1) were retained. Mash sketch was run on each individual genome with the flag "-s 10000" and the mash distance was calculated between all genomes as an estimate for their pairwise 1-ANI (average nucleotide identity) [82]. All genomes with a mash distance < 0.01 were joined with edges in a graph [83], and clusters were identified by finding connected components. One representative genome per cluster was chosen, prioritizing genomes with a higher assembly level and genomes with longer total sequence length. This clustering resulted in 15,148 candidate genomes. Genomes were further filtered by removing ambiguous nucleotides at the termini of each contig, by removing regions annotated as "centromere" in an available GFF file, and by removing contigs that were less than 10 kb in total length. Finally, contigs that were composed of more than 5% ambiguous nucleotides were removed. This final filtered set included 15,032 genomes and 6.98 trillion nucleotides.

Metagenomes. A previously described metagenomics dataset [84] was further curated as part of the training data. This included 41,253 metagenomes and metagenome-assembled genomes from NCBI, JGI IMG [85], MGnify [86], MG-RAST [87], Tara Oceans samples [88], and Youngblut et al. animal gut metagenomes [89]. All contigs were split at consecutive stretches of ambiguous nucleotides of length 5 bp or longer, the split contigs were filtered by a minimum sequence length of 1 kb, and only contigs with at least one open reading frame as predicted by Prodigal [90] were kept. Contig-encoded proteins were previously clustered at 90% identity using MMseqs [84, 91]. To further remove redundant sequences, contigs were sorted by descending length, and each contig was only retained if at least 90% of its respective protein clusters were not already in the sequence collection (determined using a bloom filter from the pybloom package with a capacity of 1614960255 and an error rate of 1e-6).

Eukaryotic organelle genomes. 33,457 organelle genomes were identified and downloaded using the "NCBI Organelle" web resource. Ambiguous nucleotides at the terminal ends of the organelle genome sequences were removed. Sequences that had over 25 ambiguous nucleotides were removed. This resulted in 32,240 organelle genomes that were used for training, including 17,613 mitochondria, 12,856 chloroplasts, 1,751 plastids, 18 apicoplasts, 1 cyanelle, and 1 kinetoplast.

mRNA and ncRNA transcripts. Transcripts were extracted using GTF files that were available through NCBI for 4,390 reference genomes. All transcripts were extracted using these coordinates, and the longest representative transcript per gene was selected to limit sequence redundancy. Transcripts from each representative genome were then filtered to be at least 64 nucleotides in length and less than 100 kb in length, and transcripts that had consecutive stretches of ambiguous nucleotides that were 5 bp or longer were removed. Transcripts were clustered with mmseqs at 90% identity for each genome to reduce redundancy. Transcripts were then split into "mRNA" and ncRNA (not-mRNA) by using the gbkey field of the GTF file. All mRNA and ncRNA transcripts across all species were then grouped together, and separately clustered again using mmseqs at 90% identity.

Noncoding RNAs. ncRNA sequences were obtained from Ensembl (release 112), Rfam, and RNAcentral. For Ensembl, we used sequences explicitly annotated as "ncrna" across 338 reference genomes. For Rfam and RNAcentral, we downloaded all available sequences in each database. All ncRNA sequences were then combined into a single FASTA file and then clustered with mmseqs easy-linclust with parameters –min-seq-id 0.9, -c 0.8, and –cov-mode 0 to produce the final set of sequences for training.

Eukaryotic promoters (EPDnew). Sequences from 15 organisms consisting of 600 bp representing positions −499 to 100 relative to the transcription start sites, which correspond to experimentally validated promoter sequences, were obtained from the EPDnew database. These sequences were then clustered with mmseqs easy-linclust with parameters –min-seq-id 0.9, -c 0.8, and –cov-mode 0 to produce the final set of sequences for training.

A.2.2. Data processing and tokenization

Datasets were preprocessed differently for pretraining and midtraining to reflect the different lengths of training. We augment the data in order to focus model pretraining on more conserved, information dense regions around genes, inspired by previous approaches [92]. To enable data augmentations and stitching of multiple contigs together, we introduce two special tokens. The '#' token is used to join sequences from the same species with uncertain distance to each other, while the '@' token is used for sequences that are from the same contig/strand and are near each other. We performed a data ablation to test the effectiveness of our data composition compared to one with fewer augmentations (**Appendix C.3**).

For pretraining, we performed the following additional filtering, processing, and augmentation strategies. We used both ncRNA and mRNA transcript annotations to generate the augmented transcripts and gene windows data portions.

GTDB and IMG/PR. We reverse complement 50% of the sequences. A random set of 100 sequences was used for validation.
Transcripts. Sequences with 3 continuous 'N' nucleotides were removed, and all uracil is replaced with thymine. We reverse complement 50% of the sequences. A random set of 1000 sequences is held out for validation.
Augmented transcripts: Eukaryote promoter, exon, and splice overhangs. As a separate data augmentation, the same set of clustered mRNA and ncRNA transcripts (see **Section A.2.1**) were modified to include an additional 1,024 bp of starting sequence and an additional 32 bp around each exon for additional splice site information. These "stitched" transcript sequences were then combined together for each transcript using a special "@" token.
Eukaryotic genic regions. We use windows around annotated genes to enrich functional coding and cis regulatory elements in the training data. Using the same GTF gene coordinates previously retrieved from NCBI (see **Section A.2.1**), we created an augmented collection of eukaryotic sequences that were

enriched around coding and noncoding exons. All transcripts that were retained after all filtering steps of the mRNA and ncRNA transcript curation process (see **Section A.2.1**) were identified, with 5000 bp from both sides of each exon coordinate. These coordinates were then merged using bedtools [93] in a strand-agnostic manner, and contiguous stretches of sequence were then extracted from each respective genome sequence file. All extracted sequences were then split at consecutive stretches of ambiguous nucleotides of length 5 bp or longer, and the remaining sequences were filtered to be at least 1,000 nucleotides in length. The '@' token is used to join sequences from the same contig. 50% of entire joined sequences are reverse complemented.

For midtraining, we increased the effective length of sequences to take advantage of the model's extended context window. We achieved this by stitching together sequences from the same accession with different strategies for each dataset, leading to effective sequence lengths of millions of base pairs for prokaryotic and eukaryotic genomes so that the model sees relevant sequence in its entire window during training.

GTDB. Sequences from the same organism are joined together with a special '#' token at the gaps. This increases the median sequence length from 12 kb to 2 million base pairs. Phylogenetic tags are added every 131 kb to help condition the model.
IMG/VR. Phylogenetic tags are added at the beginning of every IMG/VR sequence
Eukaryotic genomes. Sequences from the same organism are joined together. '@' is used for sequences from the same contig, while the '#' is used to join sequences from different contigs. This increases the median sequence length from 15 kb to millions.

Transcripts, augmented transcripts, genomic windows, and IMGPR remain the same as for the pretraining phase.

Phylogenetic tags are included help condition the model during midtraining, and loss is ignored for these tokens. Phylogenetic tags are formatted Greengenes-style lineage strings which concatenate all taxa starting domain to species, with all uppercasing separated by semicolons. '|' tokens are added at the start and end, similar to in Evo 1 [80]. For example, the tag for *E. coli* would be:

```
|D__BACTERIA;P__PSEUDOMONADOTA;C__GAMMAPROTEOBACTERIA;
O__ENTEROBACTERALES;F__ENTEROBACTERIACEAE;G__ESCHERICHIA;
S__ESCHERICHIA|
```

A.2.3. UMAP visualization of whole genomes

We created a UMAP visualization of all prokaryotic and eukaryotic representative genomes used in the training data as an illustrative representation of their abundance and diversity. K-mer frequencies were calculated for all prokaryotic and eukaryotic genomes using jellyfish [94] and k values 1 through 6. These were combined into a single data frame, and scikit-learn [95] was used to scale the data. To better separate the domains, k-mer composition vectors were multiplied by 2 for archaeal species and by 3 for eukaryotic species. The umap.UMAP function was used to then calculate the UMAP with parameters n_neighbors=15, min_dist=0.5, and default parameters otherwise [96].

A.3. Prediction evaluations

A.3.1. Effect of mutations on Evo 2 likelihoods around start codons

Reference genome sequences and their annotations for 20 prokaryotic and 16 eukaryotic species were obtained from NCBI. For each genome, 1,024 protein-coding genes were randomly selected, with the exception of *N. equitans*, for which all of its 536 protein-coding genes were selected. For each of these sampled genes, we selected genomic coordinates ranging from 20 bp upstream to 20 bp downstream from the first base of the start codon, and mutated the wildtype base of each position to each of the three alternative bases to introduce SNVs. Then, using the Evo 2 7B model, we calculated the difference in the likelihoods of the SNVs to their respective wildtype sequences, both of which included the genomic context of a 8,192nt-long window centered around the mutated nucleotide. In calculating the log likelihoods of both wildtype sequences and their SNVs,

we used the average likelihoods of the original sequence and its reverse complement. For each position, these delta likelihoods were averaged across the three SNVs, and across the 1,024 sampled genes (3,072 SNVs in total). The same process was used for the variant effects around stop codons. The phylogenetic trees for both prokaryotic and eukaryotic species were constructed using existing literature [97, 98, 99].

A.3.2. Effect of prokaryotic mutations on Evo 2 likelihoods

We systematically introduced mutations across different genomic regions in prokaryotic genomes. We obtain reference genomes of 20 prokarytoic species shown below through NCBI for this analysis.

- *Escherichia coli* (GCF_000005845.2)
- *Bacillus subtilis* (GCF_000009045.1)
- *Synechocystis sp. PCC* (GCF_000019485.1)
- *Mycobacterium tuberculosis* (GCF_002357975.1)
- *Bacteroides thetaiotaomicron* (GCF_001314975.1)
- *Methanococcus maripaludis* (GCF_002945325.1)
- *Nitrosopumilus maritimus* (GCF_000018465.1)
- *Acinetobacter baumannii* (GCF_001628795.1)
- *Enterococcus faecium* (GCF_009734005.1)
- *Klebsiella pneumoniae* (GCF_000968155.1)
- *Neisseria gonorrhoeae* (GCF_023822665.1)
- *Neisseria meningitidis* (GCF_015679665.1)
- *Pseudomonas aeruginosa* (GCF_000626655.2)
- *Staphylococcus aureus* (GCF_003609855.1)
- *Methanocaldococcus jannaschii* (GCF_000091665.1)
- *Sulfolobus solfataricus* (GCF_000968435.2)
- *Haloferax volcanii* (GCF_010692905.1)
- *Thermococcus kodakarensis* (GCF_028471865.1)
- *Nanoarchaeum equitans* (GCF_000008085.1)
- *Streptomyces coelicolor* (GCF_008931305.1)

For each species, 5,000 positions were randomly sampled from bases annotated as coding regions, and 2,000 positions each were sampled from bases within loci annotated as rRNAs, tRNAs, and ncRNAs, respectively. Positions not annotated as CDS, rRNA, tRNA, or ncRNA were considered to be intergenic regions, from which 2,000 positions were sampled for each species. Centered around each sampled position, a 8,192 bp window was used as the genomic context for the calculation of likelihoods with Evo 7B. For each sampled position in the CDS, the wildtype base was mutated to each of the three alternative bases to introduce SNVs, and deleted for the 1 bp deletion mutant. For each sampled position in rRNA, tRNA, ncRNA, or intergenic regions, the 10 bp window surrounding the position was deleted for the 10 bp deletion variant. For the deletion variants, the 8,192 bp window was extended into the neighboring sequences to match the length of the wildtype sequence (8,192 bp) to avoid any biases in the likelihoods that arise from differences in sequence lengths. Whether an SNV in the coding region was synonymous, missense, or a nonsense mutation was determined using the standard codon table for all species.

A.3.3. Effect of eukaryotic mutations on Evo 2 likelihoods

We systematically introduced artificial mutations across different genomic regions in eukaryotic genomes. Gene annotations were extracted from GFF3 and GTF files obtained through Ensembl [100] or NCBI. Reference genome sequences for 16 eukaryotic species were used:

- *Arabidopsis thaliana* (GCF_000001735.4)
- *Caenorhabditis elegans* (GCA_000002985.3)
- *Callithrix jacchus* (GCA_011100555.1)
- *Chlamydomonas reinhardtii* (GCA_000002595.3)
- *Drosophila melanogaster* (GCF_000001215.4)

- *Danio rerio* (GCA_000002035.4)
- *Homo sapiens* (GCF_000001405.40)
- *Macaca mulatta* (GCF_003339765.1)
- *Mus musculus* (GCF_000001635.27)
- *Nicotiana attenuata* (GCA_001879085.1)
- *Oryza sativa* (GCA_001433935.1)
- *Paramecium tetraurelia* (GCA_000165425.1)
- *Saccharomyces cerevisiae* (GCA_000146045.2)
- *Thalassiosira pseudonana* (GCA_000149405.2)
- *Tetrahymena thermophila* (GCA_000189635.1)
- *Xenopus tropicalis* (GCF_000004195.4)

Mutations were generated in coding sequences, including synonymous substitutions, nonsynonymous substitutions, premature stop codons, and single nucleotide deletions (frameshift). Noncoding regions include 5′ UTRs, 3′ UTRs, introns, intergenic sequences, and noncoding RNA exons annotated as lncRNA (lncRNA, lincRNA, or lnc_RNA), snoRNA, miRNA (miRNA, pre_miRNA), snRNA, tRNA (tRNA or source=trnascan), rRNA, or ncRNA (ncRNA or ncRNA_gene). Noncoding regions were mutated with 10 bp deletions. 8,192 bp of sequence context around each mutation position were used, and additional flanking sequence was appended to the ends of the sequence in the case of deletions to keep the length of the sequences consistent. Species-specific codon tables were used to identify appropriate premature stop codons and substitutions. Up to 200,000 sequences were sampled across sequence types to maintain balanced representation. For each unique region coordinate (i.e. a specific exon or intron) up to 20 distinct mutations were retained. For each region and mutation type, the median change in likelihood was calculated as the representative change in likelihood.

A.3.4. Stop codon analysis across genetic codes

To test the ability of the Evo 2 model to understand variations in the genetic code usage across different species, we introduced premature stop codons into coding sequences across 5 species: *Arabidopsis thaliana* (GCF_000001735.4, standard code), *Homo sapiens* (GCF_000001405.40, standard code), *Mycoplasma pneumoniae* (GCF_900660465.1, mycoplasma code), *Thalassiosira pseudonana* (GCA_000149405.2, ciliate code), and *Tetrahymena thermophila* (GCA_000189635.1, ciliate code).

For each species we used GFF3 and GTF annotation files to identify coding sequences. Coding sequences and mutation positions were randomly selected, filtering out pseudogenes and sequences shorter than a minimum length threshold of 10. For each selected coding sequence, randomly selected codons were replaced with seven different stop codons (TAA, TAG, TGA, AGG, TCA, AGA, TTA). Sequences were randomly subsampled to at most 200,000 per genome. Using the Evo 2 7B model, we calculate the likelihood of reference and mutated sequences using sequence context windows ranging from 100 to 8,192 base pairs centered on the mutation site. We calculate the change in likelihood for each mutation, and the median of these values for each codon mutation was calculated for each species. These median values were z-score standardized across all 7 tested codons.

To distinguish between memorization versus in-context recognition, we test the effect of recoding all stop codons to a different code on Evo 2 predictions. We recode the genomes of *Thalassiosira pseudonana* (GCA_000149405.2, ciliate code), and *Tetrahymena thermophila* (GCA_000189635.1, ciliate code) to use codon table 1 by mutating all ciliate TAA and TAG glutamine codons into randomly selected standard code glutamine codons and repeat the compare the likelihood of reference and mutated sequences.

A.3.5. Translational codon ramp analysis

The translational codon ramp pattern was investigated across the genomes of *H. sapiens* (GCF_000001405.40), *S. cerevisiae* (GCF_000146045.2), *A. pernix* (GCF_000011125.1), and *E. coli* (GCF_000005845.2). For each codon in each species, a previously calculated tRNA-adaptation index (tAI) was used as a single metric to summarize the availability of complementary tRNAs for each codon, accounting for tRNA abundance and wobble position redundancy [101]. We computed the average change in log-likelihood across a random sample of genes and codon positions in each species' genome. These log-likelihood changes were separated according to synonymous codon mutations with a higher tAI than the reference sequence and synonymous codon muta-

tions with a lower tAI than the reference sequence and then was averaged at each position. We then applied a position-wise smoothing with a rolling 5-codon average.

A.3.6. Zero-shot protein fitness prediction

We conducted zero-shot fitness prediction for protein and ncRNA sequences as described in [80]. We previously compiled nine datasets of prokaryotic DMS datasets from ProteinGym in which the original study authors had provided readily accessible nucleotide and protein sequences: a β-lactamase DMS by Firnberg et al., a β-lactamase DMS by Jacquier et al., a CcdB DMS, a multiprotein thermostability dataset, an IF-1 DMS, an Rnc DMS, an HaeIII DMS, a VIM-2 DMS, and an APH(3′)II DMS. See ProteinGym [102] for a list of references to these studies. We also compiled six datasets of human proteins from [103] in which the original study authors had provided readily accessible nucleotide and protein sequences: a CBS DMS, a GDI1 DMS, a PDE3A DMS, a P53 DMS by Kotler et al., a P53 DMS by Giacomelli et al., and a BRCA1 DMS. See [103] for references to these studies.

For this study, we also compiled a set of 18 DMS datasets corresponding to proteins from viruses that infect humans: an HCV polymerase DMS by Qi et al., an influenza hemagglutinin DMS by Wu et al., an influenza nucleoprotein DMS and a PB1 DMS by Doud et al., an influenza PA DMS by Wu et al., an influenza neuraminidase DMS by Jiang et al., an HIV-1 TAT DMS and an HIV-1 REV DMS by Fernandes et al., an HIV-1 envelope DMS by Haddox et al., an HIV-1 envelope DMS by Duenas-Decamp et al., an influenza hemagglutinin DMS by Lee et al., an HIV-1 envelope DMS by Haddox et al., an influenza PB2 DMS by Soh et al., a Zika envelope DMS by Sourisseau et al., a SARS-CoV-2 spike RBD DMS by Starr et al., a coxsackievirus capsid DMS by Mattenberger et al., an AAV2 capsid DMS by Sinai et al., a SARS-CoV-2 Mpro DMS by Flynn et al., a dengue virus NS5 DMS by Suphatrakul et al., and an influenza PB1 DMS by Li et al. See ProteinGym [102] for a list of references to these studies.

For comparing nucleotide and protein language models on bacterial and human DMS datasets, we utilized the complete set of unique nucleotide sequences and their corresponding fitness values as reported in the original studies. When discrepancies arose between fitness values reported for nucleotide sequences versus their protein counterparts, we used the fitness values for the nucleotide sequences; in these cases, we evaluated the protein language models using the translated sequence. For mutations involving stop codons, which were reported in some studies, we included these sequences when evaluating the nucleotide language models but excluded them from the protein language model benchmark. For human viral proteins, we used the wildtype protein sequence and substitutions as reported by ProteinGym to evaluate protein language models. To evaluate the nucleotide language models on human viral proteins, we manually retrieved the nucleotide sequence of each protein from the GenBank entry associated with each protein's UniProt entry. For each amino acid substitution in ProteinGym, we used the most frequently used nucleotide codon in the human codon table corresponding to the mutant amino acid value.

To assess model performance, we calculated the absolute Spearman correlation between the experimental fitness values and the model-derived sequence scores. For autoregressive language models, we used sequence likelihood as the score, while for masked language models, we used sequence pseudolikelihood. We compared the Evo 2 40B, Evo 2 7B, Evo 1, GenSLM [104], Nucleotide Transformer (2.5b-multi-species) [105], and RNA-FM [106] nucleotide language models and the CARP-640M [107], ESM-1v [108], ESM-2 650M, ESM-2 3B [109], ProGen2-large, and ProGen2-xlarge [110] protein language models.

A.3.7. Zero-shot ncRNA fitness prediction

We previously compiled seven DMS datasets of ncRNA [80]: a ribozyme DMS by Kobori et al., a ribozyme DMS by Andreasson et al., a tRNA DMS by Domingo et al., a tRNA DMS by Guy et al., a ribozyme DMS by Hayden et al., a ribozyme DMS by Pitt et al., and a rRNA mutagenesis study by Zhang et al. See [80] for a list of references to these studies. To assess model performance, we calculated the absolute Spearman correlation between the experimental fitness values and the model-derived sequence scores. For autoregressive language models, we used sequence likelihood as the score, while for masked language models, we used sequence pseudolikelihood. We compared the Evo 2 40B, Evo 2 7B, Evo 1, GenSLM, Nucleotide Transformer (2.5b-multi-species), RNA-FM, RNAErnie [111], and RiNALMo [112] nucleotide language models.

A.3.8. Zero-shot mRNA decay evaluation

For this evaluation, we used a human cell line dataset that leverages metabolic RNA labeling to estimate mRNA decay rates across the transcriptome [113]. We used the average values across the reported lines as a measure of overall mRNA decay rates. We then used the Evo 2 40B and 7B models, along with Nucleotide Transformer (2.5b-multi-species), RNA-FM, and RiNALMo to compare sequence scores to mRNA decay rates. For inputs to each model, we either used the full-length mRNA or the longest context allowed by the model from the end of the transcript. Since model scores may be confounded by sequence length, we first applied LOESS (locally estimated scatterplot smoothing) regression to correct for variations in transcript length.

A.3.9. Exon/intron classification

To evaluate model embeddings' ability to classify genomic positions as exonic or intronic across diverse eukaryotes, we selected 94 available eukaryotic species from PANTHER19.0 [114]. We partitioned organisms into training (80%), hyperparameter optimization (10%), and test (10%) sets, with *Homo sapiens*, *Mus musculus*, and *Danio rerio* manually assigned to the test set after random partitioning. *Trichomonas vaginalis* and *Leishmania major*, originally in the test set, were excluded from evaluation due to insufficient intronic annotations. All computations were performed on a single NVIDIA H100 GPU. Model versions used were Evo 2 7B base, evo-1-8k-base for Evo 1 [80], and nucleotide-transformer-2.5b-multi-species for Nucleotide Transformer [105].

For each species, we randomly sampled positions from NCBI RefSeq-annotated [115] protein-coding genes in an unbiased manner. The latest version of RefSeq FASTA and GTF files were downloaded on November 18, 2024. All exon annotations from RefSeq GTF files were included, without distinguishing between constitutive and alternative exons. For each position, we extracted both forward and reverse strand sequences from reference genomes up to each model's maximum length (8192 bp for Evo 2 and Evo 1; 5994 bp for Nucleotide Transformer), with the target position at the 3′ end of each. We concatenated the forward and reverse strand embeddings as classifier input after keeping all embedding dimensions and only the final sequence position.

We evaluated both linear and single-hidden layer classifiers. For initial optimization, we sampled 150 positions per species and extracted model embeddings from each model's top-level layers. Using weighted binary cross-entropy loss and Adam optimization, we trained classifiers for all layers and selected the best-performing layer based on validation accuracy. We then optimized hyperparameters (number of hidden layers (0 or 1), hidden layer dimension, learning rate, batch size, and weight decay) using Tree-structured Parzen Estimators (TPE) on the selected layer, choosing the configuration that maximized validation accuracy. For Evo 2 - layer: blocks.26, number of hidden layers: 1, hidden layer dimension: 1024, learning rate: 5×10^{-5}, batch size: 16, and weight decay: 2×10^{-4}. For Nucleotide Transformer - layer: 24, number of hidden layers: 1, hidden layer dimension: 1024, learning rate: 5×10^{-4}, batch size: 64, and weight decay: 2×10^{-7}. For Evo 1 - layer: blocks.3, number of hidden layers: 0, learning rate: 5×10^{-5}, batch size: 64, and weight decay: 1.5×10^{-6}. Final classifiers were trained on 1500 positions per species using these optimal parameters.

SegmentNT predictions were obtained using the publicly released multispecies model `InstaDeepAI/segment_nt_multi_species` available through HuggingFace [116]. For each organism in the test set, we collected the same input sequences used for Evo 2 evaluation and tokenized them with the model's provided tokenizer. For each sequence, we extracted the softmax probabilities for the model's genomic features, and recorded the exon probability at the midpoint, which was used to calculate AUROC values.

AUGUSTUS (v3.5.0) [117] predictions were obtained using the closest available species parameter sets for each organism: Homo sapiens→human, Mus musculus→human, Danio rerio→zebrafish, Theobroma cacao→cacao, Hordeum vulgare subsp. vulgare→wheat, Vitis vinifera→arabidopsis, Spinacia oleracea→tomato, and Selaginella moellendorffii→arabidopsis. We provided the same sequence context as the Evo 2-based classifier ($2 \times 8192 - 1 = 16{,}383$ bp) centered on each position and wrote multi-FASTA files. AUGUSTUS was run *ab initio* with the following parameters - genemodel: *partial*, introns: *on*, sample: 100. From GFF outputs, we computed a midpoint exon probability using the reported posterior score at the center and used this to calculate AUROC scores. If no exon/intron was detected at the center position, we set `p_exon_center` = 0.5.

GC-content was calculated in a 20-bp window around each position in the test set. PhyloP scores were extracted from conservation tracks from the UCSC Genome Browser [118] where available data matched assemblies used in exon classifier training (hg38.phyloP100way.bw and mm39.phyloP35way.bw).

A.3.10. Prokaryotic and phage gene essentiality

We conducted zero-shot gene essentiality prediction as previously described [80], while newly adding data from archaea. We obtained binary essentiality data (labeled as "essential" or "nonessential") for 56 bacterial genomes and 2 archaeal genomes from the DEG database [119]. Additionally, we incorporated genome-wide essentiality data for two phage genomes, lambda and P1, from [120], using the binary labels that the study authors assigned based on their CRISPRi screen results.

To conduct our *in-silico* gene essentiality screen, we accessed the complete bacterial genomes using DEG-provided RefSeq IDs. For the phage genomes, we used RefSeq: NC_001416 (lambda phage) and RefSeq: NC_005856 (P1 phage) as reference genomes. We provided the model with gene sequence plus symmetric context totaling 8,192 bp (equal distribution on both sides). For genes exceeding 8,192 bp in length, we utilized only the first 8,192 bp of the sequence.

We calculated scores by determining the difference in log-likelihoods between mutated and wildtype sequences. Our mutation strategy involved inserting multiple stop codons ("TAATAATAATAGTGA") at a 12-nucleotide offset into the coding sequence. We evaluated the Evo 2 40B, Evo 2 7B, Evo 1 131k, and megaDNA (`megaDNA_phage_145M`) [121] models using this strategy. We also used the gene's linear position in the reference genome as a predictive value for essentiality to control for potential positional bias in the model's predictions. We assessed gene conservation as another control, as previously described in [80]. We first extracted all protein sequences from the OpenGenome1 dataset, performed an all-by-all sequence search (using mmseqs easy-search with default parameters) between proteins in a given genome and OpenGenome proteins, counted the number of significant hits (E-value threshold of 1×10^{-2}), and used higher hit counts as an approximation of greater conservation and potential essentiality. We evaluated the predictive power of the log-likelihood changes (and control experiments) for binary gene essentiality using the AUROC score.

A.3.11. Zero-shot eukaryotic gene essentiality

For the eukaryotic gene essentiality experiment, we obtained CRISPR-Cas9 screening data from the Cancer Dependency Map (DepMap) Public 24Q4 release [122]. Specifically, we downloaded the file `ScreenGeneEffect.csv`, which contains per-screen (across 1401 cell-lines) gene effect estimates processed by the Chronos, a computational algorithm that corrects for copy number alterations and other confounding factors in CRISPR screens. Gene-level and screen-level metadata were extracted from `Gene.csv` and `AchillesScreenQCReport.csv` files. Gene transcript and coding sequence (CDS) coordinates were obtained from Ensembl release 114 [123], and corresponding genomic sequences were extracted from the human reference genome (GRCh38).

To derive a single essentiality metric per gene, we computed the overall essentiality score as the arithmetic mean of gene effect scores across all profiled screens:

$$E_g = \frac{1}{N} \sum_{c=1}^{N} r_{cg} \,, \tag{A.1}$$

where r_{cg} represents the copy-number corrected Chronos score for gene g in cell line c, and N denotes the total number of cell lines ($N = 1,401$ cell lines). More negative values of E_g indicate higher gene essentiality. The final preprocessed dataset contained essentiality scores for 16,434 human protein-coding genes. To conduct our *in-silico* gene essentiality screen, we provided each language model with the gene's coding sequence without any additional flanking genomic context or padding tokens. For genes exceeding the 8,192 base pair context window limit of the models, we truncated sequences to include only the first 8,192 bp, starting from the start codon.

We quantified each model's gene essentiality predictions by computing the difference in log-likelihoods between reference and scrambled gene sequences. The scrambled control sequence was generated by randomly permuting the nucleotides of the reference sequence while preserving di-nucleotide composition. We evaluated Evo 2 40B and Evo 2 7B, Evo 1 131k (predecessor model), Nucleotide Transformer [105] (transformer based 2.5B multi-species model), CodonBert [124] (BERT-style model focused on codon usage), Encodon [125] (encoder-only architecture), and Decodon [125] (decoder-only architecture). All models were evaluated using their publicly available pre-trained weights without fine-tuning. To establish baseline performance, we implemented four sequence-based conservation metrics. The first two baselines computed GC-content percentages for gene transcript and coding sequences, respectively. The other two baselines used evolutionary

conservation scores from phyloP [126], specifically computing mean phyloP scores across all possible single nucleotide variants (SNVs) within each gene's coding sequence.

A.3.12. Zero-shot variant scoring

We compare different models' ability to score mutations, zero-shot, by taking the delta between the predicted mutant and reference log likelihoods. DNA models use the DNA sequence around the variant while protein models are scored using the amino acid sequence of the gene. For non-SNV variants, we also normalized to the reference sequence, and for DNA models to keep the change in likelihood invariant to length we maintain the total length of sequence scored to be the same regardless of the length of the variant by centering on the insertion or deletion and adding or removing context nucleotides on the edges of the window. Variants assigned a more negative log-likelihood change from reference are considered to be more deleterious. Across the various evaluations, we use different models for comparisons: statistical models of conservation (PhyloP variants [126]); unsupervised language models of proteins, RNA, and DNA (ESM [127, 109], Nucleotide Transformer [105], CodonBert [124], Decodon [125], EnCodon [125], RNA-FM [128]); state-of-the-art human variant effect models (AlphaMissense [129], GPN-MSA [92]); supervised splicing prediction models (SpliceAI [130], Pangolin [131]); and state-of-the-art ensembles for human VEP (CADD [132]).

To score a variant with Evo 2, we take a genomic window of length 8,192 around the variant and calculate the log-likelihood of the variant sequence normalized by the log-likelihood of the reference sequence at the same position. For encoder models scoring single variants, we applied position-wise masking to calculate the pseudolikelihood of the reference sequence normalized to pseudolikelihood of the reference alleles, known as masked marginal likelihood [108]. To score non-SNVs using encoder models ESM-1b, ESM-2, and Encodon we use the change in pseudolikelihood [133], but cannot include non-SNVs for models with fixed coordinates like GPN-MSA or AlphaMissense. Pangolin predicts the largest increase and decrease in splice site usage at positions relative to the initially mutated site. For each variant, we reported the maximum of the absolute values of these predicted changes as the final Pangolin score. For PhyloP, we compare PhyloP100, PhyloP241, PhyloP447, and PhyloP470. To score non-SNV variants, we take the average of the PhyloP scores at each affected position in the reference. For each benchmark, we score the same variants with all models.

A.3.13. ClinVar variant effect prediction

We use ClinVar (October 2024 release), a database of expert annotated human disease variants [134]. We remove variants which affect more than 64 base pairs of reference or alternate allele sequence and remove variants of unknown significance from the evaluation. We include only variants on the nuclear genome, with a review status of at least two stars (i.e., at least multiple submitters provided evidence or an expert panel reviewed the variant), and subset to loci with matched `transcript_ids` in GRCh38.p14 GTF file.

Using model scores, we classify Pathogenic/Likely Pathogenic variants from Benign/Likely Benign variants, evaluating using AUROC and AUPRC. To enable comprehensive evaluation and fair comparison across different variant effect predictors, we stratified performance analysis across multiple dimensions. First, we separated variants by genomic context into coding (exonic) and noncoding (intronic, UTR, and intergenic). Second, we distinguished between single nucleotide variants (SNVs) and complex/non-SNV variants (insertions, deletions, and multi-nucleotide variants) to assess model performance across different mutation types. This stratification allows direct comparison with coding-specific predictors (e.g., AlphaMissense, ESM models) and SNV-specialized models (e.g. GPN-MSA) that cannot handle complex variants.

To investigate the relationship between evolutionary conservation and model predictive accuracy, we stratified ClinVar variants according to their phylogenetic conservation scores. We utilized phyloP 100-way vertebrate conservation scores [126], which quantify the degree of evolutionary constraint at each genomic position based on a multiple sequence alignment of 100 vertebrate species. Variants were categorized into four conservation bins based on their phyloP 100 scores: highly conserved (phyloP $>$ 2), moderately conserved ($0.5 \leq$ phyloP ≤ 2, suggesting mild evolutionary constraint), weakly conserved ($0 \leq$ phyloP < 0.5, near-neutral evolution), and accelerated evolution (phyloP < 0, indicating positive selection or relaxed constraint).

We implemented additional stratification based on variant proximity to known transcript splice sites. Using RefSeq gene annotations for the GRCh38 reference genome, we extracted splice donor and acceptor sites for all annotated transcripts represented in the ClinVar dataset. A variant was classified as "Near to splice site" if

located within 3 base pairs of the nearest splice donor site or within 5 base pairs of the nearest splice acceptor site, reflecting the asymmetric importance of donor versus acceptor consensus sequences. To further assess Evo 2's performance on variants with low predicted effects on splicing, we separately ran our ClinVar analyses for SNVs and non-SNVs after filtering out all variants with either a SpliceAI score ≥ 0.1 or a Pangolin score ≥ 0.1 (i.e., keeping variants with max(SpliceAI score, Pangolin score) < 0.1) using published thresholds from [130] and [131].

A.3.14. SpliceVarDB variant effect prediction

We use SpliceVarDB, a database of experimentally validated splice variants in humans [135], to classify mutations that cause aberrant splicing based on zero-shot prediction of various models. We exclude variants labeled as 'low frequency' in SpliceVarDB.

A.3.15. BRCA1/2 zero-shot classification

For all *BRCA1* and *BRCA2* SNVs with reported functional scores and classifications, we parsed sequences of a 8,192 bp window around the variant site from the human reference genome used by the respective original studies [136, 137]. Zero-shot likelihoods were calculated for the reference and variant sequences for this 8,192 bp window, and their deltas were used to classify the variants. For both genes, we used the classification of SNVs made in the original studies to label the SNVs. For *BRCA1*, SNVs labeled as "LOF" in the original study were classified as loss of function variants ($N = 823$), while SNVs labeled as "FUNC" ($N = 2{,}821$) or "INT" ($N = 249$) were labeled as functional/intermediate variants (total $N = 3{,}070$). In the original study, these categories were determined by running a two-component Gaussian mixture model on bimodally distributed "function scores" of each SNV that were measured experimentally [136]. The INT category was grouped together with the FUNC category because of its relatively smaller size. For *BRCA2*, SNVs labeled as "P strong", "P moderate", and "P supporting" were classified as loss of function variants ($N = 1{,}156$), while SNVs labeled as "B strong", "B moderate", "B supporting" were classified as functional/intermediate variants ($N = 5{,}681$). We further separated z-shot prediction of BRCA1 variant pathogenicity for noncoding SNVs according to "intronic" ($N = 489$) and "splice site" variants ($N = 559$) based on whether the variant site is located more than 8 bp away from the intron-exon boundary using labels supplied by [136].

A.3.16. BRCA1 supervised classification

To train the supervised classifier, the 3,893 *BRCA1* SNVs were first split into train and test sets. To prevent SNVs on the same genomic position from being split into the train and test sets, we first randomly split the 1,326 genomic positions of *BRCA1* that were covered by these SNVs into 1,060 (80%) positions for the train set, and 266 (20%) positions for the test set. This resulted in a total of 3,104 SNVs for the training set and 789 SNVs for the test set that each belong to these two sets of genomic positions.

For all *BRCA1* SNVs in the train and test sets, we again parsed sequences of a 8,192bp window around the mutation site using the human reference genome used by the original study [136]. Evo 2 embeddings were extracted for each of these reference and variant sequences, for both the original sequence and its reverse complement. To identify the best layer of Evo 2 40B for this task, we extracted embeddings from the pre-normalization layer of each block of the Evo 2 40B model for all four sequences (reference sequence, reverse complement of the reference sequence, variant sequence, reverse complement of the variant sequence). For each of these embeddings, we also applied a range of pooling functions that take the mean-pooled embeddings over tokens, but with different window sizes (16, 32, 64, 128, 256, 512, 1,024, 2,048, 4,098, and 8,192bp) around the mutation site. For instance, for the pooling function with window size 32, we calculated the mean-pooled embeddings for only the bases that lie within the 32bp window surrounding the mutation site. For each SNV, these pooled vectors from the embeddings of the reference sequence, the reverse complement of the reference sequence, the variant sequence, and the reverse complement of the variant sequence were concatenated to serve as inputs to train a ridge regression model.

The training data was randomly split into 5 folds using the genomic positions of the SNVs, similar to how the train and test sets were split. For each combination of embedding layer and pooling function window size, ridge regression models with the regularization parameter (α) set as 0.1, 1, 10, 100, 1,000, 10,000, 100,000, and 1,000,000 were tested using 5-fold cross-validation. The AUROCs of the 5 folds were averaged to find the

optimal combination of embedding layer (layer 20), pooling function window size (128nt), and the value of the ridge regression regularization parameter (1,000). This final model was then trained on the entire train set and tested on the test set, from which we report the final AUROC and AUPRC.

A.3.17. Noncoding regulatory sequence evaluations (DART-Eval)

For zero-shot evaluations in DART-Eval [138], we assessed model performance on Task 1 (CCRE), Task 2 (TF Motif), and Task 5 (Variant Effect Prediction). Tasks 1 and 2 focused on likelihood-based evaluations: Task 1 involved distinguishing candidate cis-regulatory elements (cCREs) from shuffled control sequences, using a dataset of over 2.3 million sequences derived from experimentally validated regulatory regions, while Task 2 aimed to identify transcription factor (TF) motifs by differentiating true TF binding sites from control sequences, utilizing a dataset of approximately 577,000 sequences from TF footprinting experiments.

Task 5 evaluated variant effects through both likelihood-based and embedding-based analyses. This task included two datasets: one examining chromatin accessibility QTLs (CaQTLs) across African populations, with over 219,000 variant sites, and another focusing on Yoruba dynamic sequence QTLs (dsQTLs), with approximately 28,000 sites. For the embedding-based analysis, Evo 2 embeddings were first extracted from Block 13, which was randomly selected for both the 7B and 40B models. Additionally, Block 27 was then chosen for the Evo 2 7B model while Block 20 was selected for the Evo 2 40B model due to their superior performance on the *BRCA1* variant supervised classification task (see **Section A.3.16**). We selected these tasks to emphasize Evo 2's zero-shot capabilities for regulatory sequence prediction. Results across all three tasks were compared to pre-computed results from testing various models in the DART-Eval paper, including GENA-LM (bert-large-t2t), HyenaDNA (large-1m), DNABERT-2 (117M), and NT 500M (v2-500m-multi-species).

A.4. Mechanistic interpretability with sparse autoencoders

A.4.1. SAE training and dataset composition

We trained a BatchTopK sparse autoencoder [139], a variant of the TopK sparse autoencoder [140, 141] on activations from Evo 2 7B residual stream following layer 26 (a Hyena-MR block) based on the sparsity of features and performance of classification on an evaluation set of genome annotations. This layer position (26/32) is consistent with findings from language model interpretability where abstract concepts emerge in later layers [142]. We used an intermediate checkpoint of Evo 2 7B after extension to 262,144 context length. Sparse autoencoders take as input model activations x and autoencode them into a sparse feature vector f, before predicting a reconstruction of the inputs $\hat{x}$. The conventional SAE architecture (which we also make use of here) is a very wide MLP with one hidden layer:

$$f = \sigma(W_e x + b_e)$$
$$\hat{x} = W_d f + b_d.$$

The nonlinearity $\sigma(\cdot)$ that we use is the Batch-TopK activation function. The Batch-TopK activation function is a version of the TopK activation function, which zeros out all but the k largest elements of a vector. The BatchTopK activation function with an equivalent value of k and a batch of size B zeros out all but the kB largest elements of the input batch. This allows the SAE to flexibly allocate capacity to higher-complexity inputs. Evo 2 activations have dimensionality $d_{\text{model}} = 4,096$ and our SAE has feature dimensionality $d_{\text{feature}} = 8 \times 4,096 = 32,768$. We used $k = 64$ for our SAE training.

Batch-TopK SAEs are trained with a loss $\mathcal{L} = \mathcal{L}_{\text{recon}} + \alpha \mathcal{L}_{\text{aux}}$ that combines a reconstruction loss $\mathcal{L}_{\text{recon}}$ and auxiliary loss $\mathcal{L}_{\text{aux}}$. The reconstruction loss is simply the mean squared reconstruction error:

$$\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|_2^2,$$

and the auxiliary loss $\mathcal{L}_{\text{aux}}$ predicts the residual $\varepsilon = x - \hat{x}$ using only "dead features" (see [141] and [139] for details). A dead feature is a feature f_i which has not fired (i.e., $f_i = 0$) for some large number of inputs during training. We call a feature dead if it has not fired for 10 million training inputs.

Our main SAE (which we refer to as the mixed-data SAE) was trained on a dataset of prokaryotic and eukaryotic genomes, representing a small subset of OpenGenome2. For prokaryotes, genomes were collected from GTDB release 220.0, filtering for genomes annotated as GTDB and NCBI representative genomes, and

assembly level annotated as "complete", totaling 2752 genomes and 10.97 billion base pairs. For eukaryotic genomes, 16 representative species were selected (see **Section** A.3.3), and only regions with high gene content (genic regions, see **Section** A.2.2) were used for SAE training, totaling 4.91 billion base pairs.

We subsampled this dataset to 1 billion activations from sequence chunks of length 16,384 to provide SAE training data. These activations were then globally shuffled so that activations from the same input sequence were unlikely to occur in the same SAE training batch. We trained the SAE using the Adam optimizer with a learning rate of 5×10^{-5}, default values of β_1 and β_2, and a batch size of 16,384 for a single epoch. We used a trapezoidal learning rate schedule, ramping up from zero for the first 5% of training and ramping down to zero for the last 5%. We also trained an SAE solely on eukaryotic data (which we refer to as the eukaryotic data SAE). The eukaryotic data SAE was identically trained to the mixed-data SAE, except that instead of subsampling 1B tokens from the combined eukaryotic and prokaryotic datasets, we subsampled only from the eukaryotic dataset described above.

We evaluated the reconstruction quality of the SAE using normalized mean squared error (nMSE), defined as:

$$\text{nMSE} = \frac{\text{MSE}}{\text{Var}(X)} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}, \tag{A.2}$$

where N is the total number of data points, x_i represents the original activation value at position i, $\hat{x}_i$ represents the SAE-reconstructed activation value at position i, $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the mean of the original activation values, and $\text{Var}(X)$ denotes the activation variance. The MSE measures the average squared difference between original and reconstructed activations. This normalization expresses the reconstruction error relative to the natural variation in the data, allowing for meaningful comparison across different scales. Our BatchedTop-K SAE reached a normalized mean square error (nMSE) of 0.29 after ~65 thousand steps, with only 2 dead features out of 32,768. The SAE reconstruction explained 75% of activation variance across the entire E. coli genome and 72% on a broader dataset comprising a random 5% sample of the mixed prokaryotic-eukaryotic training corpus. An nMSE of 0.29, as achieved by our mixed prokaryotic-eukaryotic BatchedTop-K SAE after approximately 65 thousand training steps, indicates that the reconstruction error is 29% of the variance in the original activations.

A.4.2. SAE metrics and feature embedding calculation

To compute some basic statistics for the trained mixed-data SAE, activations were computed for all features across the *E. coli* K12 MG1655 genome (NCBI reference sequence NC_000913.3) and a length-matched segment of human chromosome 17 (NCBI reference sequence NC_000017.11, bases 40,019,967-44,661,618) in 50 kb non-overlapping sequence chunks. Activation density for each feature was then computed as the fraction of non-zero activations and the mean non-zero activation computed as expected across all sequence chunks from the relevant genome for both prokaryotes and eukaryotes. To visualize the features in an embedding, the (4096×32768) weights matrix was column-normalized and then embedded using UMAP [96] with 2 components and random seed 1 and visualized by coloring each point by the difference in prokaryotic and eukaryotic activation density for each feature.

A.4.3. Prophage feature

To identify a prophage-associated feature, we used contrastive feature search on 100 kb sequences that are centered on geNomad-annotated [143] phage regions and include flanking bacterial regions. These sequences were from 100 randomly selected genomes from GTDB. The feature with the highest mean differential activation (f/19746) was selected. To evaluate the predictive value of this feature, we analyzed its mean activation values in prophage regions. We then compared these values against length-matched bacterial (non-phage) sequences from the same genome, sampled without replacement.

To analyze this feature at the genome scale, we computed feature activations on 50 kb non-overlapping chunks of the *E. coli* K12 MG1655 genome, and compared them with RefSeq-annotated phage regions. In the genome-scale plot in **Figure 4b**, we de-noise feature activations by setting a position's activation to 0 if it does not have a neighboring position with a nonzero activation. From the dataset of 100 randomly selected GTDB genomes, we selected 3 loci with the highest f/19746 activation values outside of phage-annotated regions for visualization in **Extended Data Fig. 7d**.

As feature f/19746 also activated on the region downstream of the last CRISPR direct repeat in *E. coli*, we sought to determine whether Evo 2 is learning to associate sequences downstream of CRISPR direct repeats (like CRISPR spacers) with phage sequences or directly memorizing phage sequences. We investigated this by performing ablations on the sequence. Using the same CRISPR locus displayed in **Fig. 4b**, we first computed feature activations on a synthetic sequence where all spacer sequences in the CRISPR array were independently scrambled. Observing that this did not result in a change in the feature activation pattern, we then computed feature activations on a synthetic sequence in which all the CRISPR direct repeats were replaced with a constant scrambled sequence, while keeping the spacer sequences unchanged from the natural sequence. Lastly, we repeated the previous test but replaced CRISPR direct repeats with nonconstant, scrambled sequences.

A.4.4. E. coli genomic loci features

Features reflecting genome organization were identified using a combination of contrastive feature search and manual examination of features with large total activations over sequence chunks of the *E. coli* K12 MG1655 genome (NCBI reference sequence NC_000913.3), as annotated by RefSeq. Feature activations for the 100 kb segment displayed in the figures were computed over the full 100 kb segment as one batch.

To quantify mean activations over annotations, activations were computed over non-overlapping 50 kb chunks of the *E. coli* K12 MG1655 genome. Each nucleotide in the genome was grouped into ORF, tRNA, rRNA, or intergenic annotations based on existing RefSeq annotations, with the intergenic annotations including all sequences that were not categorized as ORF, tRNA, or rRNA. ORF annotations were further categorized into (+) ORF and (-) ORF depending on their directionality. For each annotation, mean activation for each feature was computed by extracting and averaging the activations across all nucleotides within the annotation.

A.4.5. Protein secondary structure features

We identified α-helix and β-sheet associated features using contrastive feature search on all coding sequences in the *E. coli* K12 MG1655 genome. Secondary structure annotations were obtained using the DSSP algorithm [144, 145] on structures from the AlphaFold Protein Structure Database [146]. For each feature, we computed the Pearson correlation between its activation pattern on a coding sequence and the secondary structure annotations of the encoded protein, for all coding sequences. The features with the highest mean Pearson correlation with α-helix regions (f/28741) and β-sheet regions (f/22326) were selected.

Structures of EF-Tu in complex with thrT tRNA, and of RpoB and RpoC in complex, were obtained using AlphaFold 3 [147] through the AlphaFold server. To project the features onto the structures, we first preprocessed the feature activation values. Except for thrT tRNA, we first mean collapsed feature activations per codon, as feature activations were computed at the genome level. Feature activations were then smoothed using a Gaussian kernel with a window size of 10. We then overlaid the smoothed feature activations on the structures, using linear interpolation coloring and clipping values at a maximum threshold of 0.2, visualized using ChimeraX [148].

A.4.6. Frameshift and premature stop feature

Using the same set of artificially mutated coding sequences generated previously for the human genome, contrastive feature search was used with the eukaryotic data SAE to identify features that seemed to specifically correspond with frameshift and premature stop codon mutations, but not synonymous or nonsynonymous substitutions. A subset of 100 loci was first used to calculate average feature activations across all mutation types in 8192 bp windows. The difference in mean feature activations between the mutated and reference sequence was used to prioritize features. Top 20 features for each mutation type were identified. Features that were among the top 20 predictors for both frameshift and premature stop codon mutations were identified. Features that were also in the top 20 for synonymous or nonsynonymous substitutions were then excluded from this subset. This resulted in 3 remaining features: f/24278, f/29870, and f/18585. f/24278 was then selected for further analysis. A separate subset of 100 sequences, each of length 8192 bp, was used to analyze 24,278 feature activations on a per-nucleotide, and this subset was used to determine the average feature activation pattern near the mutation site. A separate subset of 500 sequences was then used to estimate the precision, recall, and the F1 score of the feature. For calculating these metrics, any non-zero activation of the feature within 100 bp after the mutation position was considered a true positive.

A.4.7. Transcription factor motif features

From the GRCh38 human reference genome [149], a random sample of 50,000 promoter sequences, defined as 1 kb upstream of a transcription start site using GENCODE v46 annotations, was selected and passed through the model to extract the SAE feature activations. We focus on the eukaryotic-only SAE. For each feature, a maximum of 10,000 feature activations above the sparsity threshold of 1×10^{-4} were stored in sparse matrices from these activations. Following this selection, 31 bp windows centered on each activation position were extracted. We then construct a Position Weight Matrix (PWM) for each SAE feature, using these 31 bp windows.

As a baseline, we evaluated HOMER [150], a widely used unsupervised motif discovery framework. We performed de novo motif discovery on the same 50k promoter sequences, extracting 2,000 motifs each at lengths of 8 bp, 10 bp, and 12 bp. The discovered motifs were then compared to the HOCOMOCO database using TOMTOM under the same parameters, and recall was computed across both the full and promoter-enriched subsets. While HOMER serves as a reasonable baseline, we note that it represents an earlier generation of motif discovery tools and may not reflect recent methodological advances.

To assess whether Evo 2's SAE features recapitulate known transcription factor binding motifs, we compared each PWM derived from Evo 2 to the "Human and Mouse (HOCOMOCO v12 CORE)" motif database [151] using the TOMTOM motif comparison tool from the MEME suite (v5.5.7) [152]. A match was considered significant if it achieved a q-value < 0.01 under the Sandelin-Wasserman similarity metric. We report the percentage of HOCOMOCO motifs recalled by Evo 2 across the full database, as well as a promoter-enriched subset. Promoter-enriched motifs were identified using the Simple Enrichment Analysis (SEA) [153] tool from the MEME suite, which detected transcription factor motifs significantly enriched in 50,000 promoter sequences relative to a background of 50,000 randomly selected genic regions.

It is important to acknowledge the limitations of the TOMTOM motif comparison tool: a statistically significant hit does not always correspond to clear visual motif similarity. To mitigate false positives, we apply a more stringent threshold ($q < 0.01$) and use the Sandelin-Wasserman similarity metric [152]. Nonetheless, TOMTOM remains susceptible to spurious matches. Additionally, we observe that multiple SAE features often match the same known motif. This is consistent with the feature-splitting phenomenon [154], wherein increasing the number of SAE features leads to finer-grained representations—such as sub-motifs—that result in multiple closely related features mapping to the same canonical motif. To mitigate this, future work may involve reducing the number of SAE features, or using intelligent motif clustering techniques, such as TF-MoDISco [155].

A.4.8. Exon-intron features

SAE features associated with eukaryotic coding sequences, introns, and exon-intron boundaries were found through contrastive feature search on the annotated sequence of chromosome 1 of the GRCh38 human reference genome [149]. 50 exon-rich segments of length 8,192 bp were sampled from this reference sequence, and feature activations were collected from each segment. Then, we searched for features with the top-20 highest values of the weighted sum

$$\sum_i f_i(4m_i - 3),$$

where f_i is the activation at base i, and m_i is 1 if base i is annotated as the given genomic component (CDS, intron, exon start, exon end), and 0 if not.

For features associated with exon starts and ends, the first 50 bases and the last 50 bases of each exon were used as the loci where m_i is 1, respectively. The features that are presented most frequently in these 50 segments were visually inspected to determine the features that activate the most consistently at the respective loci, and a representative feature was chosen for CDS, introns, exon starts, and exon ends. To calculate the F1, precision, and recall scores for each feature, we sampled 1,000 genes with 5 or more exons from the human genome, and counted the bases where the features have a nonzero activation on or off the genomic regions corresponding to each feature. The reference genome annotations for CDS and introns were considered to be ground truth in calculating the numbers of true/false positives/negatives. For the exon start feature (f/1050),

we considered the first 25 bases of each exon following an intron to be the ground truth, in consideration of the firing patterns of the feature. For the exon end feature (f/25666), we considered the last one base of each exon followed by an intron to be the ground truth label. To calculate the mean activations of features for each type of genomic loci, we used the same set of 1,000 genes and calculated the average activations of each feature for each individual genomic locus identified by the ground truth labels: a single exon, a single intron, first 25 bases of a single exon, and the last base of a single exon.

The activations for the same set of features were collected for the *PDK3* gene loci of the woolly mammoth genome [156]. The exon coordinates of the mammoth genome were derived from homologs to the *Loxodonta africana* genome. Note that while the *Loxodonta africana* genomic sequence was part of the training corpus for Evo 2, it was not part of the training data used to collect activations for the SAE.

A.4.9. SAE feature ablations

To assess whether the SAE-identified features contribute causally to the model's computation, we performed targeted latent ablations and quantified the effect on cross-entropy loss. For each feature, we mapped the model's intermediate activations into the sparse autoencoder (SAE) latent space, set the corresponding latent coefficient to zero at all token positions ("SAE feature ablation"), decoded the perturbed latents back to the activation space via the SAE decoder, and substituted these reconstructed activations into Evo 2 before resuming the forward pass. We then measured the change in cross-entropy relative to an unperturbed run.

A.4.10. SAE feature visualization web viewer

We manually curated 104 prokaryotic genomes of microbiological, medical, or agricultural relevance from NCBI. These genomes were annotated using the Bakta v1.10.3 pipeline [157], putative prophage regions were annotated using geNomad [143], and secondary structure annotations were obtained using the DSSP algorithm [144, 145] on structures from the AlphaFold Protein Structure Database [146]. Activations were computed for all features without using the Batch-TopK activation function, and the top 50 largest activations per token were retained. Building on igv.js [158], features identified as mapping to prokaryotic concepts are plotted and interactively displayed at https://arcinstitute.org/tools/evo/evo-mech-interp.

A.5. Unconstrained generation with Evo 2

A.5.1. Gene completion

DNA sequences were collected around genes from *Haloferax volcanii*, *Escherichia coli*, *Saccharomyces cerevisiae*, *Chlamydomonas reinhardtii*, *Arabidopsis thaliana*, and *Homo sapiens*, covering archaeal, prokaryotic, and eukaryotic species. For each gene, a prompt of 1,000 base pairs upstream of the gene and the first 500 bp (*Haloferax volcanii*, *Escherichia coli*, *Saccharomyces cerevisiae*) or 1000 bp (*Chlamydomonas reinhardtii*, *Arabidopsis thaliana*, and *Homo sapiens*) of the gene sequence were input into the model. The language model generated the remaining sequence using a temperature of 0.7 and top-k of 4. We performed 10 generations for each prompt and evaluated results by converting to amino acids, aligning using BioPython, and calculating the percent protein recovery. We compared Evo 1 and Evo 1 131k with the different Evo 2 models.

A.5.2. Mitochondrial genome generation

We prompted with 3,000 base pairs of the human mitochondrial genome (NCBI Reference Sequence NC_012920.1) and generated over 250 unique mitochondrial sequences (16 kb each) using top-k of 4 and temperature of 1.0 or 0.7 with Evo 2 40B and 7B. Prompts consisted of positions 0-3,000 of the human mitochondrial genome ("forward-prompt") or the reverse complement of positions 13,500-16,500 ("reverse-prompt"). Generated sequences were annotated using MitoZ [159] to assess rRNA, tRNA, and CDS counts in Evo 2 40 B and 7B forward- and reverse-prompt generations. Synteny analysis was performed using LoVis4u on the Evo 2 40B reverse-prompt generations [160]. To visualize all genes in each generated sequence by LoVis4u, including tRNAs and rRNAs, the MitoZ-predicted genes and their positions were extracted from MitoZ's output tables and each feature was labeled as a 'CDS' in a GFF3-compatible file format as input to LoVis4u. Sequence diversity of the Evo 2 40B and 7B forward-prompt generations was evaluated through BLASTn searches against the core_nt database with default settings, to quantify query coverage and sequence identity. We calculated

codon usage bias on the Evo 2 40B forward-prompt generations as the proportion of each codon used out of the total frequency of all codons across all generated coding sequences predicted by MitoZ, which we compared with the codon usage bias of the reference *Homo sapiens* mitochondrion. Anticodons were predicted for each tRNA gene by MitoZ on the Evo 2 40B forward-prompt generations and their frequencies per generated sequence were averaged. For structural analysis, we selected generated sequences containing the complete human complement of mitochondrial ND, CO, and ATP genes. Protein complexes were predicted using AlphaFold 3 through the AlphaFold server, with identical analysis performed on human mitochondrial genes for comparison. Generated structures were assessed through alignment against wild-type structures. Gene homology was analyzed using BLASTp with default settings, with percent sequence identity calculated as the product of query coverage and percent identity. Protein structures were visualized using ChimeraX [148].

A.5.3. Mycoplasma genitalium genome generation

Generation was initiated using the first 10.5 kb of the reference *M. genitalium* genome as prompt. Using Evo 2 40B, we generated 35 sequences of 580 kb each with temperature 1.0 and top-k of 4. We compared these against previously generated *M. genitalium* sequences from Evo 1 131k and the reference natural *M. genitalium* genome. Protein-coding regions were predicted using Prodigal [90] and analyzed for homology using hmmscan [161] against the Pfam database [162]. The density of significant hits (E-value < 0.001) was calculated for generations from each model. Proteins with significant Pfam hits were folded using ESMfold [109] and filtered to exclude any proteins with pLDDT < 0.2. Protein quality metrics evaluated included pLDDT scores, which were derived from ESMfold, and secondary structure distributions, which were obtained using DSSP [144]. As a point of comparison, annotated proteins from the reference *M. genitalium* genome were folded using ESMFold and compared to those from Evo 2 generated *M. genitalium* using the same metrics and filtering. Protein structures were visualized using ChimeraX [148].

A.5.4. Yeast chromosome generation

Sequences were generated using the first 10.5 kb of *S. cerevisiae* chromosome III as prompt, with temperature 1.0 and top-k of 4 settings in Evo 2 40B. Twenty artificial chromosomes of 330 kb were generated. Gene prediction was performed using GeneMark-S with Prot-hint based on the Fungi proteins from OrthoDB v12.0. As with the prokaryotic generations, predicted protein sequences were analyzed for homology against natural proteins using hmmscan (E-value < 0.001) against the Pfam database. Proteins containing significant Pfam hits were subsequently folded using ESMFold, filtered to remove hits with pLDDT < 0.2, and evaluated for pLDDT and secondary structure (DSSP). To predict tRNAs and promoters, generated DNA sequences were annotated using the Yeast Genome Annotation Pipeline [163] and Promoter 2.0 [164] respectively (likelihood score > 0.8). Feature density analysis included quantification of predicted tRNAs, promoters, and genes with intronic structure. As a point of comparison, this process was repeated for the reference *S. cerevisiae* chromosome III genome. To evaluate the composition of these Evo 2 generated sequences against *S. cerevisiae*, we calculate the tetranucleotide usage deviations (TUDs) for the whole sequence, for CDS, and for the 500bps upstream of CDS to represent promoters. We compare the TUDs of all Evo 2 7B and Evo 2 40B generated genomes against *S. cerevisiae* chromosome III, with *S. cerevisiae* chromosome IV and *S. pombe* chromosome III as positive controls. We use the Ensembl annotations and fasta for the natural genomes compared against our annotated generated genomes.

A.6. Generative epigenomics via inference-time guidance

A.6.1. Beam search algorithm

We implemented a beam search algorithm for efficiently sampling a sequence autoregressively while guided by a scoring function. At a high level, we sample autoregressively from a language model to obtain multiple chunks of tokens that are all continuations of the same prompt. We then use a given scoring function to select the best chunks, which are appended to the prompt for the next round of sampling.

More formally, let us denote a sequence as $\mathbf{x} = \{x_1, \ldots, x_L\} \in \mathcal{X}^L$ where L is the sequence length and $\mathcal{X}$ is the vocabulary (e.g., DNA base pairs). Let

$$\hat{\mathbf{x}}[a, b] \sim p(x_a, x_{a+1}, \ldots, x_b \mid \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{a-1}) = p(\mathbf{x}[a, b] \mid \tilde{\mathbf{x}}[1, a-1])$$

denote a sampled sequence from a distribution p, which we parameterize with an autoregressive language model (e.g., Evo 2), and where $a, b \in [L]$ define the start and stop indices for a sampled sequence chunk such that $a < b$. We define $C = b - a + 1$ as the length of a sampled sequence chunk. We can obtain samples from p via standard autoregressive decoding. On each iteration t of the beam search algorithm, we sample K chunks

$$\hat{\mathbf{x}}^{(k)}[Ct, C(t+1)-1] \sim p(\mathbf{x}[Ct, C(t+1)-1] \,|\, \tilde{\mathbf{x}}[1, Ct-1]), \quad k \in [K]$$

off the prompt $\tilde{\mathbf{x}}[1, Ct-1]$.

Now, we are also given a scoring function $f : \mathcal{X}^L \to \mathbb{R}_{\geq 0}$ that takes in a sequence and outputs a nonnegative score, where a lower value of the score is better. When we pick the prompt for iteration $t+1$, we can pick the chunk with the lowest score to append to the prompt at iteration t, i.e.,

$$\tilde{\mathbf{x}}[Ct, C(t+1)-1] = \arg\min_{k \in [K]} \left\{ f\left(\hat{\mathbf{x}}^{(k)}[1, C(t+1)-1] \right) \right\}$$

where

$$\hat{\mathbf{x}}^{(k)}[1, C(t+1)-1] = \tilde{\mathbf{x}}[1, Ct-1] + \hat{\mathbf{x}}^{(k)}[Ct, C(t+1)-1]$$

and + denotes a string concatenation operator.

Note that instead of taking only the best chunk, we can instead use the best $K' \leq K$ chunks as prompts for sampling at the next iteration, i.e.,

$$\hat{\mathbf{x}}^{(j,k)}[Ct, C(t+1)-1] \sim p(\mathbf{x}[Ct, C(t+1)-1] \,|\, \tilde{\mathbf{x}}^{(j)}[1, Ct-1]), \quad k \in [K], \quad j \in [K']$$

where $\tilde{\mathbf{x}}^{(j)}[1, Ct-1]$ includes one of the best-K' chunks (i.e., with the lowest scores) according to f obtained in the previous iteration. The algorithm iterates until all tokens L have been sampled. For the first chunk, we sample off a sequence $\tilde{\mathbf{x}}_{\text{prompt}}$. We assume that C is constant and that L is an integer multiple of C, i.e., all chunks sampled throughout the procedure, including the final chunk, are of the same length.

A.6.2. Generative epigenomics in mouse cells: Beam search

For the mouse genome design runs shown in **Fig. 6f-h** and **Extended Data Fig. 10a**, we used a chunk length $C = 128$, a total designed sequence length of $L = 19,968$, used either $K = 15$ (for the "ARC" design) or $K = 42$ (for all other designs) chunks per prompt, and retained $K' = 2$ prompts per iteration (i.e., we sampled $2 \times 42 = 84$ chunks per iteration). We used Evo 2 7B to parameterize p, from which we sample using standard autoregressive decoding with a temperature of 1.0 and top-k of 4. For $\tilde{\mathbf{x}}_{\text{prompt}}$, we used a sequence of length 40,960 upstream of the genomic region chrX: 52,051,929–52,123,468 in the mm39 reference genome.

For our scoring function, we used an ensemble of Enformer [165] and Borzoi [166], where Borzoi itself contains an ensemble of four replicate models. We used the implementation of Enformer at https://github.com/lucidrains/enformer-pytorch and the Flashzoi reimplementation of Borzoi [167].

We note that Enformer takes in a sequence $\mathbf{x} \in \mathcal{X}^{196,608}$ and outputs scores $\tilde{\mathbf{y}}_{\text{Enformer}} \in \mathbb{R}^{896}$, where each dimension of $\tilde{\mathbf{y}}_{\text{Enformer}}$ corresponds to 128 bp (representing output predictions for $896 \times 128 = 114,688$ bp centered in the input), thereby not making predictions for 40,960 bp of left and right flanking context. Our scoring function requires a user-defined binary pattern $\mathbf{y}_{\text{Enformer}} \in \{0,1\}^{896}$, which specifies, for example, a Morse code peak pattern. To score with Enformer, we simply divide all entries in $\tilde{\mathbf{y}}_{\text{Enformer}}$ by the maximum value in $\tilde{\mathbf{y}}_{\text{Enformer}}$ (i.e., so all values are in the interval $[0,1]$), which we denote $\hat{\mathbf{y}}_{\text{Enformer}}$, and report the score as the ℓ_1 norm

$$f_{\text{Enformer}}(\mathbf{x}) = \|\mathbf{y}_{\text{Enformer}} - \hat{\mathbf{y}}_{\text{Enformer}}\|_1.$$

As input to Enformer, we concatenate: (i) 40,960 bp of sequence upstream of chrX: 52,051,929–52,123,468, (ii) the designed sequence, and (iii) enough DNA sequence downstream of chrX: 52,051,929–52,123,468 to achieve a total input sequence to Enformer of length 196,608. As beam search progresses, we mask the loss such that it is defined only on the sequence designed by Evo 2.

Borzoi makes predictions similar to Enformer but with longer sequence lengths and higher output resolution, taking in a sequence $\mathbf{x} \in \mathcal{X}^{524,288}$ and outputting scores $\tilde{\mathbf{y}}^{(i)}_{\text{Borzoi}} \in \mathbb{R}^{6,144}$, where $i \in [4]$ refers to one Borzoi model replicate in an ensemble of four and each dimension corresponds to 32 bp (representing output predictions for $6,144 \times 32 = 196,608$ bp centered in the input), thereby not making predictions for 163,840 bp of

left and right flanking context. We modify $\mathbf{y}_{\text{Enformer}}$ to have higher resolution by expanding its dimensionality to 6,144 by repeating each entry 128/32 = 4 times to produce a vector $\mathbf{y}_{\text{Borzoi}} \in \{0,1\}^{6,144}$. We normalize the Borzoi predictions $\tilde{\mathbf{y}}^{(i)}_{\text{Borzoi}}$ by dividing by the maximum value across all dimensions and across all predictions in the ensemble, to produce $\hat{\mathbf{y}}^{(i)}_{\text{Borzoi}}$. We then combine scores across the Borzoi ensemble by computing a "lower confidence bound"

$$(\hat{y}_{\text{Borzoi}})_j = \left[\frac{1}{4}\sum_{i=1}^{4}(\hat{y}^{(i)}_{\text{Borzoi}})_j\right] - \left[\text{std}_{i\in[4]}(\hat{y}^{(i)}_{\text{Borzoi}})_j\right], \quad j \in [6144]$$

where $(\hat{y}^{(i)}_{\text{Borzoi}})_j$ indicates the jth entry in $\hat{\mathbf{y}}^{(i)}_{\text{Borzoi}}$ and std$(\cdot)$ is the standard deviation. Our Borzoi loss is then defined as

$$f_{\text{Borzoi}}(\mathbf{x}) = \|\mathbf{y}_{\text{Borzoi}} - \hat{\mathbf{y}}_{\text{Borzoi}}\|_1.$$

Analogous to the Enformer setting, our input sequence to Borzoi is a concatenation of: (i) 163,840 bp of sequence upstream of chrX: 52,051,929–52,123,468, (ii) the designed sequence, and (iii) enough DNA sequence downstream of chrX: 52,051,929–52,123,468 to achieve a total input sequence to Borzoi of length 524,288. As beam search progresses, we mask the loss such that it is defined only on the sequence designed by Evo 2.

Our final scoring function simply takes the average of the Enformer and Borzoi scoring functions

$$f(\mathbf{x}) = \frac{1}{2}\left(f_{\text{Enformer}}(\mathbf{x}) + f_{\text{Borzoi}}(\mathbf{x})\right).$$

We use this scoring function to guide the beam search algorithm as described above.

Importantly, our implementation of beam search stores the full inference cache (including the attention kv-cache and hyena recurrent state) alongside each sequence. Rather than, for example, reprompting the model on each iteration with the generated sequence, this approach enables a time-efficient implementation of the beam search algorithm within practical amounts of memory (our sampling runs leverage an NVIDIA H100 GPU with 80 GB of memory).

A.6.3. Generative epigenomics in mouse cells: Design patterns and tasks

We designed six peak patterns:

- **Long square wave:** Alternating between 1664 bp of open chromatin and 1664 bp of closed chromatin.
- **Medium square wave:** Alternating between 768 bp of open chromatin and 768 of closed chromatin.
- **Short rectangular wave:** Alternating between 384 bp of open chromatin and 1152 bp of closed chromatin.
- **"LO" morse code:** Encoding LO (`.-.. ---`) in Morse code. One dot length corresponds to 768 bp.
- **"ARC" morse code:** Encoding ARC (`.- .-. -.-.`) in Morse code. One dot length corresponds to 384 bp.
- **"EVO2" morse code:** Encoding EVO2 (`. ...- --- ..---`) in Morse code. One dot length corresponds to 384 bp.

We follow the Morse code specification in which a dash is three times the width of a dot, dots and dashes within a character are separated by a single dot length, and characters within a word are separated by three dot lengths. Dots and dashes are encoded by open chromatin; spaces are encoded by closed chromatin.

A.6.4. Generative epigenomics in mouse cells: Inference-time scaling experiments

We also conducted scaling analyses in which we vary the number of chunks sampled per beam-search iteration, thereby increasing the amount of inference-time compute required to complete a single design run. For each of the six patterns above, we ran beam search with the following configurations:

- $K' = 1$, $K = 1$ (1 tok/bp)
- $K' = 1$, $K = 2$ (2 tok/bp)
- $K' = 1$, $K = 3$ (3 tok/bp)

- $K' = 1, K = 6$ (6 tok/bp)
- $K' = 1, K = 9$ (9 tok/bp)
- $K' = 1, K = 12$ (12 tok/bp)
- $K' = 1, K = 15$ (15 tok/bp)
- $K' = 2, K = 15$ (30 tok/bp)
- $K' = 2, K = 30$ (60 tok/bp)

where tok/bp indicates the number of tokens sampled per base pair designed. All other design parameters were kept the same. We assessed the quality of a design with the AUROC metric, in which the "true" values were the desired peak patterns $\mathbf{y}_{\text{Enformer}}$ and $\mathbf{y}_{\text{Borzoi}}$, the "predicted" values were the outputs $\hat{\mathbf{y}}_{\text{Enformer}}$ and $\hat{\mathbf{y}}_{\text{Borzoi}}$, and the AUROC compares the true and predicted values separately for Enformer and Borzoi. The final reported AUROC score for a design is the average of the Enformer and Borzoi AUROCs. We conducted these scaling laws using either Evo 2 7B or a model that samples uniformly over the nucleotide vocabulary as the proposal distribution p.

We also assessed the agreement between the single Enformer prediction and the four aligned Borzoi predictions as another quality metric, for which we computed the Intraclass Correlation Coefficient (ICC). Specifically, we used a two-way mixed-effects model across the five prediction tracks, which represent a fixed set of observers over which we measure agreement. We calculated the ICC for absolute agreement to ensure that systematic differences in the magnitude of predictions between models would be penalized. The final reported score is for the reliability of the average of the $k = 5$ tracks, corresponding to the ICC(2,k) form. This analysis was performed at the 32 bp resolution after upsampling the Enformer track, using the pingouin statistical package in Python.

A.6.5. Generative epigenomics in mouse cells: Experimental validation

Designed sequences were synthesized by a commercial vendor (Ansa) with flanking heterotypic Bxb1 recombinase sites to enable site-specific, single-copy integration into a pre-installed landing pad at the Hprt locus in hybrid Bl6 x Castaneus mouse embryonic stem cells (mESCs). The landing pad harbors a negative selection marker (PIGA). Methods for mESC culture and targeted integration have been described previously [168, 169, 170].

Briefly, synthetic DNA was purified from *E. coli* using a ZymoPURE midiprep kit, sequence-verified via whole plasmid sequencing, and nucleofected into 1 million landing pad mESCs (LP-mESCs). Each reaction included 2 μg of construct DNA and 2 μg of a Bxb1 recombinase expression plasmid [170], delivered using a Lonza 4D nucleofector according to the manufacturer's protocol. To select for successful payload integration that overwrote the landing pad, cells were treated with 2 nM proaerolysin (Aerohead Scientific) seven days post-nucleofection. Individual resistant clones were picked, genotyped via PCR across the construct, and positive clones were selected for ATAC-seq analysis.

ATAC-seq was performed on ~50,000 cells per clone using the Zymo-Seq ATAC Library Kit (Zymo D5458), following the manufacturer's instructions. Libraries were sequenced on an Illumina NextSeq 2000 using the following read structure: Read 1: 59 cycles, Read 2: 59 cycles, Index 1: 10 cycles, Index 2: 10 cycles.

Custom mm10 reference genomes were generated using the reform tool (https://reform.bio.nyu.edu/), with the designed sequence replacing mm10 chrX: 52,963,052–53,034,591 (the landing pad region). FASTQ files were aligned to the custom reference using Bowtie2 [171]. Resulting BAM files were sorted and indexed with Samtools [172]. Finally, bedGraph files containing RPKM-normalized coverage across the construct coordinates (bin size = 1) were generated using deepTools bamCoverage [173] and used for downstream visualization.

A.6.6. Generative epigenomics in mouse cells: Downstream bioinformatic analysis

To compare the sequence properties of regions meant to contain accessible chromatin with those of regions meant to be inaccessible, we analyzed three categories of sequences: (i) our experimentally validated 20-kb Morse code designs generated by Evo 2 ("ARC", "EVO2", and "LO"), (ii) control sequences for the same Morse code designs but generated with a uniform or with a bigram proposal, and (iii) peak/non-peak regions from natural mouse genomes. For the uniform and bigram designs, we selected the best design for each peak pattern

("ARC", "EVO2", and "LO") by AUROC across all design runs.

To obtain peak/non-peak regions from the native mouse genome, reproducible DNASE-seq peaks were identified by intersecting two replicate mm10 narrowPeak files from ES-E14 mESCs (ENCODE accessions: ENCFF048DWN, ENCFF565YPL) using pybedtools. The resulting peaks were filtered to retain those with a width between 384 bp and 2,304 bp (the width range of our designed peaks). The non-peak regions were defined as the genomic intervals immediately downstream of each peak region on the forward strand, extending up to a maximum of 10,000 bp or until the start of the next peak.

For each defined region, we calculated the following sequence-based features:

- **TF motif density:** TF binding motifs were identified using FIMO from the MEME Suite (v5.5.8) [174] with the HOCOMOCO v12 motif database [151] subsetted to mouse motifs. Motif density was calculated as the number of motif hits per 1 kb of sequence. This was calculated at two stringencies: a standard p-value threshold of $p < 1 \times 10^{-4}$ and a strict q-value threshold of $q < 0.01$.
- **GC content:** The percentage of guanine (G) and cytosine (C) bases in the sequence.
- **CpG density:** The number of CpG dinucleotides per 100 bp, calculated as $(N_{\text{CpG}}/L) \times 100$, where ($N_{\text{CpG}}$ is the number of CpG dinucleotides and L is the length of the sequence.
- **AA/TT periodicity:** The rotational positioning signal of AA/TT dinucleotides was quantified using a Fast Fourier Transform (FFT) implemented with `scipy.fft`. A binary signal was generated along the sequence, with a value of 1 at the start of AA or TT dinucleotides and 0 otherwise. The maximum power of the resulting spectrum within the 10.0–10.5 bp period range was reported as the periodicity score.

The distributions of these features were compared between peak and non-peak regions using violin and strip plots. The statistical significance of the difference between the two groups for each feature was determined using a two-sided Welch's t-test, which does not assume equal variance. For visualization of feature trends across larger loci, values were calculated in a 128 bp sliding window and plotted as genomic tracks. For the designed sequences, we also computed the AUROC for each metric's ability to classify peak versus non-peak regions, as well as the Spearman correlation between each metric and the experimental accessibility coverage values if available.

To determine if the designed transcription factor motifs were enriched for TFs actively expressed in mESCs, we performed a hypergeometric enrichment analysis. The background set of all possible mouse TFs (M) was obtained by parsing the HOCOMOCO v12 annotation file (`H12CORE_annotation.jsonl`) to extract all unique Entrez Gene IDs associated with mouse TF motifs. The set of expressed TFs in mESCs (n) was determined from biological replicate RNA-seq datasets from ES-E14 mESCs (ENCODE accessions: ENCFF898TDI, ENCFF827OZU). The average transcript per million (TPM) value was calculated for each gene across the two replicates. Genes with an average $\log(\text{TPM} + 1) > 1$ were considered expressed (**Extended Data Fig. 10i**). The Ensembl IDs of these expressed genes were converted to Entrez Gene IDs using the mygene Python library. The final set of expressed TFs was obtained by intersecting these Entrez IDs with the TF universe. The set of TFs with binding sites present in the designed "LO" sequence was obtained via the FIMO analysis described above. All motif hits with $q < 0.01$ were considered significant. The HOCOMOCO motif IDs from these hits were mapped to their corresponding Entrez Gene IDs to generate the set of FIMO-identified TFs (N). The statistical significance of the overlap between the set of expressed TFs and the set of FIMO-identified TFs (k) was calculated using a one-tailed hypergeometric test. The test evaluated the probability of observing an overlap of size k (or greater) by chance, given the universe size M, the number of expressed TFs n, and the number of FIMO-identified TFs N. A P-value less than 0.05 was considered statistically significant.

A.6.7. Generative epigenomics in human cells: Prompt selection and generated patterns

Generated DNA sequences were designed to encode four different chromatin accessibility patterns across two human cell lines, HEK293T and K562. These patterns included: (1) accessible in HEK293T and inaccessible in K562 (HEK+/K562−), (2) inaccessible in HEK293T and accessible in K562 (HEK−/K562+), (3) uniformly inaccessible (HEK−/K562−), and (4) uniformly accessible (HEK+/K562+). To condition the Evo 2 on native sequence context, 40,960 bp of genomic sequence upstream of the NEBL locus was used. Specifically, the complement sequence from chr10:21089445c-21130405c in the human reference genome (GRCh38.p14) was appended to the complement of the insertion sequence from pAF0977 to facilitate insertion of the generated

sequence at chr10:21130405c. This sequence was provided as the initial prompt to condition the generation of both the 1 kb and 4 kb sequence designs.

A.6.8. Generative epigenomics in human cells: Sequence generation and decoding strategy

Autoregressive generation was performed using Evo 2 7B. Beam search was applied with a chunk size of $C = 128$ bp and total designed length L of either 1024 or 4996 bp. At each decoding iteration, $K = 96$ candidate chunks were sampled using a temperature of 0.8 and top-k of 4. The $K' = 5$ lowest-loss sequences were retained and appended to the prompt for the next iteration. This process was repeated until the full target region was generated. To construct model inputs for scoring, each candidate sequence was inserted between the upstream 40,960 bp context and sufficient downstream human sequence (starting at chr10:21130405c) to reach the full required length of 196,608 bp for Enformer and 524,288 bp for Borzoi.

A.6.9. Generative epigenomics in human cells: Cell-type specific scoring function

To identify sequences capable of eliciting predefined accessibility patterns specific to one or both cell types, candidate sequences were evaluated based on their predicted chromatin accessibility profiles in HEK293T and K562. Sequence evaluation was performed using two independent predictive models, Enformer [165] and Borzoi [166], both of which generate epigenomic signal predictions from raw genomic DNA sequence.

Each candidate sequence was evaluated using input sequences of 196,608 bp for Enformer and 524,288 bp for Borzoi, corresponding to the required input lengths of each model. Inputs were constructed by concatenating 40,960 bp of genomic sequence upstream of the NEBL locus, the progressively generated sequence at the current beam search step, and sufficient downstream genomic sequence to meet the total input length. Model predictions were then extracted over the entire designed region, which was centrally positioned within the model's prediction window. For each model, DNase-seq output tracks corresponding to HEK293T and K562 were selected based on known indices (HEK293T: 23-ENCFF148BGE; K562: 625-ENCFF971AHO, 123-ENCFF565YDB, 122-ENCFF868NHV, 121-ENCFF413AHU). The output resolution was 128 bp for Enformer and 32 bp for Borzoi, resulting in 8 or 32 bins for a 1024 bp generated region, and 32 or 128 bins for a 4096 bp generated region, respectively.

A binary target vector was defined for each design pattern and cell type, encoding the desired presence (1) or absence (0) of chromatin accessibility in each bin. Model predictions were normalized to the [0, 1] interval by dividing all scores within the relevant output segment by the maximum predicted value. For Borzoi, predictions were obtained from an ensemble of four independently trained replicates. Final Borzoi predictions for each bin were computed by subtracting the standard deviation from the mean across replicates, and selecting the minimum value across models to produce a conservative estimate of accessibility.

For HEK293T, predictions were extracted directly from the corresponding output channel. For K562, predictions from the four individual tracks were averaged element-wise across the designed region to obtain a single representative accessibility profile following normalization.

For each combination of model and cell type, the deviation between predicted and target accessibility was calculated using the ℓ_1 norm,

$$f(\mathbf{x}) = \sum_{i=1}^{N} |\hat{y}_i - y_i| \,,$$

where $\hat{y}_i$ denotes the normalized prediction for bin i, and y_i is the binary target. The final loss associated with each candidate sequence was defined as the sum of the four component-losses. We used different weights for each class of sequence pattern being generated:

1. Accessible in HEK293T and inaccessible in K562 (HEK+/K562−):

$$f_{\text{total}} = f_{\text{Enformer, HEK293T}} + f_{\text{Borzoi, HEK293T}} + 0.5 f_{\text{Enformer, K562}} + 0.5 f_{\text{Borzoi, K562}}$$

2. Accessible in HEK293T and inaccessible in K562 (HEK+/K562−):

$$f_{\text{total}} = 0.5 f_{\text{Enformer, HEK293T}} + 0.5 f_{\text{Borzoi, HEK293T}} + f_{\text{Enformer, K562}} + f_{\text{Borzoi, K562}}$$

3. Inaccessible in both (HEK−/K562−) and accessible in both (HEK+/K562+):

$$f_{\text{total}} = f_{\text{Enformer, HEK293T}} + f_{\text{Borzoi, HEK293T}} + f_{\text{Enformer, K562}} + f_{\text{Borzoi, K562}}$$

This loss value served as the primary metric to guide beam search, favoring candidate sequences that conformed most closely to the combined specified cell-type-specific accessibility profiles.

A.6.10. Generative epigenomics in human cells: Cell line generation

Generated sequences were synthesized by Integrated DNA Technologies (1 kb sequences) or Twist Biosciences (4 kb sequences) with Gibson overhangs and Gibson cloned into pAF0977. This plasmid expresses puromycin and mCherry driven by the SV40 promoter and contains the optimized Dn29 e-attP attachment site sequence with donor binding guide sequence as described in [175]. Co-delivery of these donor plasmids with the LSR variant hifiDn29-dCas9 enables LSR-mediated integration into the NEBL locus at chr10:21130405.

HEK293T transfection. One day before transfection, ~12,000 HEK293T cells were plated per well of a 96-well poly-d-lysine coated plate (Corning, Catalog: 356516). After 24 hours, 513 ng donor plasmid and 212 ng hifiDn29-dCas9 plasmid (a 5:1 molar ratio) were transfected with Lipofectamine 2000 (Thermo Fisher). Transfection duplicates were cultured for four weeks in DMEM with 10% FBS (Gibco) supplemented with 1× Penicillin-Streptomycin (Thermo Fisher) and 1 μg/mL puromycin, and passaged every 2-3 days upon reaching 80-90% confluency.

K562 electroporation. 715 ng donor plasmid and 285 ng hifiDn29-dCas9 plasmid (a 5:1 molar ratio) were electroporated into 200,000 K562 cells using the Lonza 4D nucleofector and the SF Cell Line 96-well Nucleofector Kit (Catalog V4SC-2096), following manufacturers instructions and using pulse code FF-120. Cells were cultured for four weeks in RPMI with 10% FBS (Gibco) supplemented with 1× Penicillin-Streptomycin (Thermo Fisher) and 2 μg/mL puromycin, and passaged every 2-3 days maintaining cells between 10^5 and 10^6 cells/mL.

Cells were monitored for mCherry expression on the Attune Flow Cytometer (Thermo Fisher), and integrations at the correct on-target locus were validated by ddPCR. Complete selection and episomal plasmid dilution was monitored by including a donor only (no LSR) control. After the four weeks of culture, ~100,000 cells were frozen in base media with 10% DMSO.

A.6.11. Generative epigenomics in human cells: ATAC-seq library generation and sequencing

Cells were processed in batches of 24 samples utilizing the Omni-ATAC-seq protocol as described in [176]. Frozen cell aliquots were thawed in a water bath, centrifuged at 300× g for 5 minutes at 4°C, and washed once with cold PBS. Next, the cells were centrifuged at 300× g for 5 minutes at 4°C, and the supernatant was carefully aspirated using P1000 and P200 pipettes. Cells were resuspended in 100 μL Lysis Buffer, consisting of a fresh solution of 0.1% (w/v) NP-40 alternative (Millipore Cat: 492016), 0.1% (w/v) Tween-20 (Teknova Cat: T0710), and 0.01% (w/v) digitonin (Thermo Fisher Cat: BN2006) in cold ATAC-RSB (10 mM Tris-HCl pH 7.5, 10 mM NaCl and 3 mM $MgCl_2$ in water, sterile filtered). After three minutes of incubation on ice, 1 mL of ATAC-seq wash buffer was added, containing 0.1% (w/v) Tween-20 in cold ATAC-RSB, and inverted five times to mix. Next, the nuclei were pelleted at 500× g for 10 minutes at 4°C, and the supernatant was carefully aspirated with a P1000 and P200 pipette. The nuclei were then resuspended in 50 μL of freshly made Transposition Mix and pipette mixed six times. For each sample, the transposition mix is composed of 25 μL 2× tagmentation buffer (Diagenode, Cat: C01019043), 16.5 μL PBS, 5 μL UltraPure distilled water, 0.5 μL 1% (w/v) digitonin, 0.5 μL 10% (w/v) Tween-20, and 2.5 μL Tn5 transposase (Diagenode, Cat: C01070012). Reactions were incubated at 37°C for 30 minutes in a thermomixer with 1000 rpm mixing. After 30 minutes, the transposition reaction was cleaned with the DNA Clean and Concentrator-5 Kit (Zymo).

The tagmented genomic DNA was next indexed and amplified using a unique combination of Adapter 1 and Adapter 2 sequences as described in Supplementary Table 2 of [176]. Each PCR reaction contained 20 μL of transposed sample, 25 μL of NEBNext Ultra II Q5 2× Master Mix, 2.5 μL 5 μM Adapter 1, and 2.5 μL 5 μM Adapter 2. K562 cells were amplified for 8 cycles and HEK293T cells were amplified for 7 cycles, as determined by a preliminary PCR cycle optimization. Finally, the libraries were cleaned with 1.3× AmpureXP beads, quantified by Qubit and TapeStation (Agilent), and sequenced at a depth of 20 million reads per sample on a NovaSeqX+ with a read length of 101 bp.

A.6.12. Generative epigenomics in human cells: ATAC-seq analysis pipeline

ATAC-seq paired-end reads were processed using the PEPATAC pipeline [177], which included quality trimming, sorting, deduplication, and alignment to custom chromosome 10 reference genomes using BWA [178]. Parallel processing of samples was performed using GNU Parallel [179]. Library quality was validated through examining fragment size distribution and by calculating TSS enrichment scores as described in [176].

To account for variation in plasmid integration efficiency and copy number between samples, we implemented a custom normalization strategy using the signal in the integrated plasmid backbone as an internal control. All reads mapping to the plasmid backbone region were counted, and a Scale Factor of 1000/plasmid_backbone_reads was calculated, normalizing signal to reads per thousand plasmid backbone reads. The scale factor was applied to generate normalized coverage tracks using bedtools genomecov, and normalized bedGraph files were converted to BigWig format using UCSC bedGraphTOBigWig tools.

A.6.13. Generative epigenomics in human cells: Downstream bioinformatic analysis

A hypergeometric enrichment analysis was performed to determine if TF motifs in designed human sequences were enriched for TFs expressed in K562 and HEK293T cell lines. For this analysis, we restricted to designs with a K562/HEK293T fold change in coverage greater than 2: B7, B10, B11, and B12. The background universe of all human TFs (M) was defined using the HOCOMOCO v12 annotation database for the HUMAN species. The set of expressed TFs for each cell line was determined by analyzing replicate RNA-seq data. Genes were considered "expressed" if their average TPM across replicates exceeded a specific threshold. These thresholds were set independently for each cell line to reflect their distinct expression profiles (**Extended Data Fig. 11i,j**). For K562 cells, expressed genes were defined as those with $\log(1 + \text{TPM}) > 1.5$, where TPM is the average across two replicates (ENCODE accessions: ENCFF003XKT and ENCFF928NYA). For HEK293T cells, expressed genes were defined using a TPM cutoff of $\log(1 + \text{TPM}) > 2.15$, where TPM is the average across two replicates (Gene Expression Omnibus accessions: GSM3734787 and GSM3734788). For both cell lines, the Ensembl IDs of expressed genes were converted to Entrez Gene IDs and intersected with the human TF universe to create the final set of expressed TFs (n). The set of TFs with binding sites in the designed sequences (N) was identified by running FIMO on the four designed sequences (B7, B10, B11, and B12). We restricted analysis to only the motif hits with a $q < 0.01$ and located entirely within the first half of the sequence (i.e., the designed peak). The unique Entrez IDs corresponding to this filtered set of motifs were aggregated across all four sequences to form the final set of FIMO-identified TFs. Finally, a one-tailed hypergeometric test was conducted separately for each cell line to assess the statistical significance of the overlap (k) between the set of cell-line-expressed TFs (n) and the set of FIMO-identified TFs (N), relative to the total human TF universe (M). An enrichment with a P-value less than 0.05 was considered statistically significant.

B. Supplementary Text

B.1. Codon usage analysis

Since we observed that less efficiently translated codons had consistently lower likelihoods than more efficient codons, we sought to assess if Evo 2 had learned about the 'codon ramp' pattern in which coding sequences begin with lower efficiency codons [101]. For four different species, we performed synonymous codon mutations and compared the per position likelihood changes of the mutations between higher versus lower tRNA adaptation index (tAI) mutations A.3.2. We did not observe the expected 'codon ramp' pattern, although higher tAI codons were favored (**Extended Data Fig. 3d,e**).

B.2. mRNA decay rate

Beyond evolutionary conservation, we hypothesized that Evo 2 has learned sequence features that contribute to molecular features of RNA molecules. We conducted a zero-shot evaluation by comparing the model-derived likelihoods for human messenger RNAs (mRNAs) against their average decay rates measured in an mRNA stability dataset [113]. Unlike other gLMs, Evo 2 likelihoods show a negative correlation with mRNA decay prediction and outperform conservation, with the 40B model showing a stronger negative correlation than the 7B model. However the correlation falls behind Borzoi, a model supervised with RNA expression information

(**Extended Data Fig. 3g**). Establishing this association across a broader range of cell types will be important to rigorously assess this relationship.

B.3. ClinVar variants stratified

We assessed Evo 2's performance on ClinVar variants when stratified by conservation. Evo 2 maintains competitive performance among unsupervised models for noncoding variants and has the overall best performance on non-SNVs in both coding and noncoding regions (**Extended Data Fig. 4a**). Evo 2 also maintains its competitive performance when separately evaluating noncoding variants that are near or far from a splice site (**Extended Data Fig. 4b**) or when only considering variants with low predicted effects on splicing (**Extended Data Fig. 4c**).

B.4. SAE feature ablations

To assess whether these learned features contribute to functional sequence generation, we performed ablation experiments on SAE features with high F1 scores for biological elements (**Methods**). For example, ablating the (+)ORF feature (f/11734) increased the cross entropy (CE) of the model predictions by 1.0% in (+)ORF annotated regions. Meanwhile, the average ratio of ablated to original CE increased by 34%, indicating larger relative changes for nucleotides starting with low CE (**Extended Data Fig. 7h**). These results suggest that some learned features are causally relevant to model logit predictions within the annotations, and suggest further research is needed to understand the structure of features learned by DNA language training.

B.5. SAE limitations

While our SAE analysis reveals biologically interpretable features, there are limitations to current dictionary learning approaches for mechanistic interpretability. For example, SAE feature absorption is a known issue [154] in which parent hierarchical features can be "absorbed" into their children and fail to fire as expected. Better disentangling such composite features remains an important direction for future work.

B.6. Morse Code Design Sequence Analysis

Further analysis of our Morse code designs revealed that designed peak regions contain significantly higher predicted TF motif density [174] than non-peak regions (two-sided Welch's t-test $P = 3.6 \times 10^{-7}$) (**Extended Data Fig. 10f-i, 11a**). Interestingly, our "LO" design has especially high TF motif density that is positioned precisely within designed peaks (**Extended Data Fig. 10g**). These motifs are also significantly enriched for mESC-expressed TFs (one-sided hypergeometric $P = 2.0 \times 10^{-4}$) (**Extended Data Fig. 10h,i**). While GC content and CpG density [180] also correlate with peak positions at levels similar to natural sequences, Fourier analysis did not detect strong AA/TT periodicity in any designed non-peak regions [181] (**Extended Data Fig. 10f**). While designs produced by uniform and bigram proposals largely recapitulate the GC content of natural peak/non-peak regions, they contain significantly fewer predicted TF motifs than the Evo 2 proposal (two-sided Welch's t-test $P = 0.0013$ comparing Evo 2 to a bigram proposal) (**Extended Data Fig. 10f**). Notably, Evo 2 was not explicitly conditioned to generate motif-rich sequences.

References

[60] Jerome Ku, Eric Nguyen, David Romero, Garyk Brixi, Brandon Yang, Anton Vorontsov, Ali Taghibakhshi, Amy Lu, David Burke, Greg Brockman, Stefano Massaroli, Christopher Re, Patrick Hsu, Brian Hie, Stefano Ermon, and Michael Poli. Systems and algorithms for convolutional multi-hybrid language models at scale. 2025.

[61] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[62] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

[63] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The Falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[64] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*, 2024.

[65] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[66] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[67] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, eads0018, 2025.

[68] Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, et al. xTrimoPGLM: unified 100b-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv:2401.06199*, 2024.

[69] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pages 28043–28078. PMLR, 2023.

[70] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in Neural Information Processing Systems*, 36, 2024.

[71] Michael Poli, Armin W Thomas, Eric Nguyen, Pragaash Ponnusamy, Björn Deiseroth, Kristian Kersting, Taiji Suzuki, Brian Hie, Stefano Ermon, Christopher Ré, et al. Mechanistic design and scaling of hybrid architectures. *arXiv preprint arXiv:2403.17844*, 2024.

[72] Armin W Thomas, Rom Parnichkun, Alexander Amini, Stefano Massaroli, and Michael Poli. STAR: Synthesis of tailored architectures. *arXiv*, 2411.17800, 2024.

[73] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[74] Gonzalo Benegas, Carlos Albors, Alan J Aw, Chengzhong Ye, and Yun S Song. A DNA language model based on multispecies alignment predicts the effects of genome-wide variants. *Nature Biotechnology*, pages 1–6, 2025.

[75] Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, Phil Wang, and Samuel Weinbach. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch, 9 2023.

[76] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

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

[78] Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*, 2023.

[79] Zhidian Zhang, Hannah K Wayment-Steele, Garyk Brixi, Haobo Wang, Dorothee Kern, and Sergey Ovchinnikov. Protein language models learn evolutionary statistics of interacting sequence motifs. *Proceedings of the National Academy of Sciences*, 121(45):e2406285121, 2024.

[80] Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, Jeremy Sullivan, Madelena Y Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D Hsu, and Brian L Hie. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, November 2024.

[81] Donovan H Parks, Maria Chuvochina, Christian Rinke, Aaron J Mussig, Pierre-Alain Chaumeil, and Philip Hugenholtz. GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic Acids Res.*, 50(D1):D785–D794, January 2022.

[82] Brian D Ondov, Todd J Treangen, Páll Melsted, Adam B Mallonee, Nicholas H Bergman, Sergey Koren, and Adam M Phillippy. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.*, 17(1):132, June 2016.

[83] Gábor Csárdi, Tamás Nepusz, Kirill Müller, Szabolcs Horvát, Vincent Traag, Fabio Zanini, and Daniel Noom. igraph for R: R interface of the igraph library for graph theory and network analysis, December 2024.

[84] Matthew G Durrant, Nicholas T Perry, James J Pai, Aditya R Jangid, Januka S Athukoralage, Masahiro Hiraizumi, John P McSpedon, April Pawluk, Hiroshi Nishimasu, Silvana Konermann, and Patrick D Hsu. Bridge RNAs direct programmable recombination of target and donor DNA. *Nature*, 630(8018):984–993, June 2024.

[85] I-Min A Chen, Ken Chu, Krishnaveni Palaniappan, Anna Ratner, Jinghua Huang, Marcel Huntemann, Patrick Hajek, Stephan Ritter, Neha Varghese, Rekha Seshadri, Simon Roux, Tanja Woyke, Emiley A Eloe-Fadrosh, Natalia N Ivanova, and Nikos C Kyrpides. The IMG/M data management and analysis system v.6.0: new tools and advanced capabilities. *Nucleic Acids Res.*, 49(D1):D751–D763, January 2021.

[86] Alex L Mitchell, Alexandre Almeida, Martin Beracochea, Miguel Boland, Josephine Burgin, Guy Cochrane, Michael R Crusoe, Varsha Kale, Simon C Potter, Lorna J Richardson, Ekaterina Sakharova, Maxim Scheremetjew, Anton Korobeynikov, Alex Shlemov, Olga Kunyavskaya, Alla Lapidus, and Robert D Finn. MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Res.*, 48(D1):D570–D578, January 2020.

[87] F Meyer, D Paarmann, M D'Souza, R Olson, E M Glass, M Kubal, T Paczian, A Rodriguez, R Stevens, A Wilke, J Wilkening, and R A Edwards. The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9(1):386, September 2008.

[88] Shinichi Sunagawa, Luis Pedro Coelho, Samuel Chaffron, Jens Roat Kultima, Karine Labadie, Guillem Salazar, Bardya Djahanschiri, Georg Zeller, Daniel R Mende, Adriana Alberti, Francisco M Cornejo-Castillo, Paul I Costea, Corinne Cruaud, Francesco d'Ovidio, Stefan Engelen, Isabel Ferrera, Josep M Gasol, Lionel Guidi, Falk Hildebrand, Florian Kokoszka, Cyrille Lepoivre, Gipsi Lima-Mendez, Julie Poulain, Bonnie T Poulos, Marta Royo-Llonch, Hugo Sarmento, Sara Vieira-Silva, Céline Dimier, Marc Picheral, Sarah Searson, Stefanie Kandels-Lewis, Tara Oceans coordinators, Chris Bowler, Colomban de Vargas, Gabriel Gorsky, Nigel Grimsley, Pascal Hingamp, Daniele Iudicone, Olivier Jaillon, Fabrice Not, Hiroyuki Ogata, Stephane Pesant, Sabrina Speich, Lars Stemmann, Matthew B Sullivan, Jean Weissenbach, Patrick Wincker, Eric Karsenti, Jeroen Raes, Silvia G Acinas, and Peer Bork. Structure and function of the global ocean microbiome. *Science*, 348(6237):1261359, May 2015.

[89] Nicholas D Youngblut, Jacobo de la Cuesta-Zuluaga, Georg H Reischer, Silke Dauser, Nathalie Schuster, Chris Walzer, Gabrielle Stalder, Andreas H Farnleitner, and Ruth E Ley. Large-scale metagenome assembly reveals novel animal-associated microbial genomes, biosynthetic gene clusters, and other genetic diversity. *mSystems*, 5(6), 2020.

[90] Doug Hyatt, Gwo-Liang Chen, Philip F Locascio, Miriam L Land, Frank W Larimer, and Loren J Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11:119, March 2010.

[91] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, 35(11):1026–1028, November 2017.

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

[93] Aaron R Quinlan and Ira M Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, March 2010.

[94] Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011.

[95] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[96] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv*, 1802.03426, 2018.

[97] Laura A Hug, Brett J Baker, Karthik Anantharaman, Christopher T Brown, Alexander J Probst, Cindy J Castelle, Cristina N Butterfield, Alex W Hernsdorf, Yuki Amano, Kotaro Ise, Yohey Suzuki, Natasha Dudek, David A Relman, Kari M Finstad, Ronald Amundson, Brian C Thomas, and Jillian F Banfield. A new view of the tree of life. *Nature Microbiology*, 1(5):16048, April 2016.

[98] Stefanie Hartmann, Dihui Lu, Jason Phillips, and Todd J Vision. Phytome: a platform for plant comparative genomics. *Nucleic Acids Research*, 34(Database issue):D724–30, January 2006.

[99] S Blair Hedges. The origin and evolution of model organisms. *Nature Reviews Genetics*, 3(11):838–849, November 2002.

[100] Peter W Harrison, M Ridwan Amode, Olanrewaju Austine-Orimoloye, Andrey G Azov, Matthieu Barba, If Barnes, Arne Becker, Ruth Bennett, Andrew Berry, Jyothish Bhai, Simarpreet Kaur Bhurji, Sanjay Boddu, Paulo R Branco Lins, Lucy Brooks, Shashank Budhanuru Ramaraju, Lahcen I Campbell, Manuel Carbajo Martinez, Mehrnaz Charkhchi, Kapeel Chougule, Alexander Cockburn, Claire Davidson, Nishadi H De Silva, Kamalkumar Dodiya, Sarah Donaldson, Bilal El Houdaigui, Tamara El Naboulsi, Reham Fatima, Carlos Garcia Giron, Thiago Genez, Dionysios Grigoriadis, Gurpreet S Ghattaoraya, Jose Gonzalez Martinez, Tatiana A Gurbich, Matthew Hardy, Zoe Hollis, Thibaut Hourlier, Toby Hunt, Mike Kay, Vinay Kaykala, Tuan Le, Diana Lemos, Disha Lodha, Diego Marques-Coelho, Gareth Maslen, Gabriela Alejandra Merino, Louisse Paola Mirabueno, Aleena Mushtaq, Syed Nakib Hossain, Denye N Ogeh, Manoj Pandian Sakthivel, Anne Parker, Malcolm Perry, Ivana Piližota, Daniel Poppleton, Irina Prosovetskaia, Shriya Raj, José G Pérez-Silva, Ahamed Imran Abdul Salam, Shradha Saraf, Nuno Saraiva-Agostinho, Dan Sheppard, Swati Sinha, Botond Sipos, Vasily Sitnik, William Stark, Emily Steed, Marie-Marthe Suner, Likhitha Surapaneni, Kyösti Sutinen, Francesca Floriana Tricomi, David Urbina-Gómez, Andres Veidenberg, Thomas A Walsh, Doreen Ware, Elizabeth Wass, Natalie L Willhoft, Jamie Allen, Jorge Alvarez-Jarreta, Marc Chakiachvili, Bethany Flint, Stefano Giorgetti, Leanne Haggerty, Garth R Ilsley, Jon Keatley, Jane E Loveland, Benjamin Moore, Jonathan M Mudge, Guy Naamati, John Tate, Stephen J Trevanion, Andrea Winterbottom, Adam Frankish, Sarah E Hunt, Fiona Cunningham, Sarah Dyer, Robert D Finn, Fergal J Martin, and Andrew D Yates. Ensembl 2024. *Nucleic Acids Res.*, 52(D1):D891–D899, January 2024.

[101] Tamir Tuller, Asaf Carmi, Kalin Vestsigian, Sivan Navon, Yuval Dorfan, John Zaborske, Tao Pan, Orna Dahan, Itay Furman, and Yitzhak Pilpel. An evolutionarily conserved mechanism for controlling the efficiency of protein translation. *Cell*, 141(2):344–354, April 2010.

[102] Pascal Notin, Aaron W Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Hansen Spinner, Nathan Rollins, Ada Shaw, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Rose Orenbuch, Yarin Gal, and Debora S Marks. ProteinGym: Large-scale benchmarks for protein design and fitness prediction. *bioRxiv*, page 2023.12.07.570727, 1 2023.

[103] Benjamin J Livesey and Joseph A Marsh. Updated benchmarking of variant effect predictors using deep mutational scanning. *Molecular Systems Biology*, 19, 8 2023.

[104] Maxim Zvyagin, Alexander Brace, Kyle Hippe, Yuntian Deng, Bin Zhang, Cindy Orozco Bohorquez, Austin Clyde, Bharat Kale, Danilo Perez-Rivera, Heng Ma, Carla M. Mann, Michael Irvin, Defne G. Ozgulbas, Natalia Vassilieva, James Gregory Pauloski, Logan Ward, Valerie Hayot-Sasson, Murali Emani, Sam Foreman, Zhen Xie, Diangen Lin, Maulik Shukla, Weili Nie, Josh Romero, Christian Dallago, Arash Vahdat, Chaowei Xiao, Thomas Gibbs, Ian Foster, James J. Davis, Michael E. Papka, Thomas Brettin, Rick Stevens, Anima Anandkumar, Venkatram Vishwanath, and Arvind Ramanathan. GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. *The International Journal of High Performance Computing Applications*, 37:683–705, 11 2023.

[105] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. Nucleotide Transformer: Building and evaluating robust foundation models for human genomics. *Nature Methods*, pages 1–11, 2024.

[106] Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, Irwin King, and Yu Li. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions, 2022.

[107] Kevin K Yang, Nicolo Fusi, and Alex X Lu. Convolutions are competitive with transformers for protein sequence pretraining. *Cell Systems*, 15(3):286–294, 2024.

[108] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 2021.

[109] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379:1123–1130, 3 2023.

[110] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. ProGen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

[111] Ning Wang, Jiang Bian, Yuchen Li, Xuhong Li, Shahid Mumtaz, Linghe Kong, and Haoyi Xiong. Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning. *Nature Machine Intelligence*, pages 1–10, 2024.

[112] Rafael Josip Penić, Tin Vlašić, Roland G Huber, Yue Wan, and Mile Šikić. RiNALMo: General-purpose RNA language models can generalize well on structure prediction tasks. *arXiv*, 2403.00043, 2024.

[113] Heather Karner, Tabea Mittmann, Vicky W. Chen, Ashir A. Borah, Andreas Langen, Hassan Yousefi, Lisa Fish, Balyn W. Zaro, Albertas Navickas, and Hani Goodarzi. Integrative analysis of mrna stability regulation uncovers a metastasis-suppressive program in breast cancer. *bioRxiv*, 2025.

[114] Huaiyu Mi, Anushya Muruganujan, John T Casagrande, and Paul D Thomas. Large-scale gene function analysis with the PANTHER classification system. *Nature Protocols*, 8(8):1551–1566, 2013.

[115] Nuala A O'Leary, Mathew W Wright, J Rodney Brister, Stacy Ciufo, Diana Haddad, Rich McVeigh, Bhanu Rajput, Barbara Robbertse, Brian Smith-White, Danso Ako-Adjei, et al. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1):D733–D745, 2016.

[116] Bernardo P. de Almeida, Hugo Dalla-Torre, Guillaume Richard, Christopher Blum, Lorenz Hexemer, Maxence Gélard, Javier Mendoza-Revilla, Ziqi Tang, Frederikke I. Marin, David M. Emms, Priyanka Pandey, Stefan Laurent, Marie Lopez, Alexandre Laterre, Maren Lang, Uğur Şahin, Karim Beguir, and Thomas Pierrot. Annotating the genome at single-nucleotide resolution with dna foundation models. *bioRxiv*, 2025.

[117] Mario Stanke, Rasmus Steinkamp, Stephan Waack, and Burkhard Morgenstern. Augustus: a web server for gene finding in eukaryotes. *Nucleic acids research*, 32(suppl_2):W309–W312, 2004.

[118] Gerardo Perez, Galt P Barber, Anna Benet-Pages, Jonathan Casper, Hiram Clawson, Mark Diekhans, Clay Fischer, Jairo Navarro Gonzalez, Angie S Hinrichs, Christopher M Lee, Luis R Nassar, Brian J Raney, Matthew L Speir, Marijke J van Baren, Charles J Vaske, David Haussler, W James Kent, and Maximilian Haeussler. The UCSC Genome Browser database: 2025 update. *Nucleic Acids Research*, 53(D1):D1243–D1249, 10 2024.

[119] R. Zhang. DEG: a database of essential genes. *Nucleic Acids Research*, 32:271D–272, 1 2004.

[120] Denish Piya, Nicholas Nolan, Madeline L. Moore, Luis A. Ramirez Hernandez, Brady F. Cress, Ry Young, Adam P. Arkin, and Vivek K. Mutalik. Systematic and scalable genome-wide essentiality mapping to identify nonessential genes in phages. *PLOS Biology*, 21:e3002416, 12 2023.

[121] Bin Shao and Jiawei Yan. A long-context language model for deciphering and generating bacteriophage genomes. *Nature Communications*, 15(1):9392, 2024.

[122] Broad DepMap. DepMap 24Q4 public, December 2024.

[123] Sarah C Dyer, Olanrewaju Austine-Orimoloye, Andrey G Azov, Matthieu Barba, If Barnes, Vianey Paola Barrera-Enriquez, Arne Becker, Ruth Bennett, Martin Beracochea, Andrew Berry, Jyothish Bhai, Simarpreet Kaur Bhurji, Sanjay Boddu, Paulo R Branco Lins, Lucy Brooks, Shashank Budhanuru Ramaraju, Lahcen I Campbell, Manuel Carbajo Martinez, Mehrnaz Charkhchi, Lucas A Cortes, Claire Davidson, Sukanya Denni, Kamalkumar Dodiya, Sarah Donaldson, Bilal El Houdaigui, Tamara El Naboulsi, Oluwadamilare Falola, Reham Fatima, Thiago Genez, Jose Gonzalez Martinez, Tatiana Gurbich, Matthew Hardy, Zoe Hollis, Toby Hunt, Mike Kay, Vinay Kaykala, Diana Lemos, Disha Lodha, Nourhen Mathlouthi, Gabriela Alejandra Merino, Ryan Merritt, Louisse Paola Mirabueno, Aleena Mushtaq, Syed Nakib Hossain, José G Pérez-Silva, Malcolm Perry, Ivana Piližota, Daniel Poppleton, Irina Prosovetskaia, Shriya Raj, Ahamed Imran Abdul Salam, Shradha Saraf, Nuno Saraiva-Agostinho, Swati Sinha, Botond Sipos, Vasily Sitnik, Emily Steed, Marie-Marthe Suner, Likhitha Surapaneni, Kyösti Sutinen, Francesca Floriana Tricomi, Ian Tsang, David Urbina-Gómez, Andres Veidenberg, Thomas A Walsh, Natalie L Willhoft, Jamie Allen, Jorge Alvarez-Jarreta, Marc Chakiachvili, Jitender Cheema, Jorge Batista da Rocha, Nishadi H De Silva, Stefano Giorgetti, Leanne Haggerty, Garth R Ilsley, Jon Keatley, Jane E Loveland, Benjamin Moore, Jonathan M Mudge, Guy Naamati, John Tate, Stephen J Trevanion, Andrea Winterbottom, Bethany Flint, Adam Frankish, Sarah E Hunt, Robert D Finn, Mallory A Freeberg, Peter W Harrison, Fergal J Martin, and Andrew D Yates. Ensembl 2025. *Nucleic Acids Res.*, 53(D1):D948–D957, January 2025.

[124] Sizhen Li, Saeed Moayedpour, Ruijiang Li, Michael Bailey, Saleh Riahi, Milad Miladi, Jacob Miner, Dinghai Zheng, Jun Wang, Akshay Balsubramani, Khang Tran, Minnie Zacharia, Monica Wu, Xiaobo Gu, Ryan Clinton, Carla Asquith, Joseph Skalesk, Lianne Boeglin, Sudha Chivukula, Anusha Dias, Fernando Ulloa Montoya, Vikram Agarwal, Ziv Bar-Joseph, and Sven Jager. CodonBERT: Large language models for mRNA design and optimization. *bioRxiv*, 2023.

[125] Mohsen Naghipourfar, Siyu Chen, Mathew Howard, Christian Macdonald, Ali Saberi, Timo Hagen, Mohammad Mofrad, Willow Coyote-Maestas, and Hani Goodarzi. A suite of foundation models captures the contextual interplay between codons. *bioRxiv*, October 2024.

[126] Katherine S Pollard, Melissa J Hubisz, Kate R Rosenbloom, and Adam Siepel. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Research*, 20(1):110–21, 2010.

[127] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from

scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118:e2016239118, 2021.

[128] Tao Shen, Zhihang Hu, Siqi Sun, Di Liu, Felix Wong, Jiuming Wang, Jiayang Chen, Yixuan Wang, Liang Hong, Jin Xiao, et al. Accurate RNA 3d structure prediction using a language model-based deep learning approach. *Nature Methods*, pages 1–12, 2024.

[129] Jun Cheng, Guido Novati, Joshua Pan, Clare Bycroft, Akvilė Žemgulytė, Taylor Applebaum, Alexander Pritzel, Lai Hong Wong, Michal Zielinski, Tobias Sargeant, et al. Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science*, 381(6664):eadg7492, 2023.

[130] Kishore Jaganathan, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F McRae, Siavash Fazel Darbandi, David Knowles, Yang I Li, Jack A Kosmicki, Juan Arbelaez, Wenwu Cui, Grace B Schwartz, et al. Predicting splicing from primary sequence with deep learning. *Cell*, 176(3):535–548, 2019.

[131] Tony Zeng and Yang I Li. Predicting RNA splicing from DNA sequence using Pangolin. *Genome Biology*, 23(1):103, 2022.

[132] Max Schubach, Thorben Maass, Lusiné Nazaretyan, Sebastian Röner, and Martin Kircher. CADD v1.7: using protein language models, regulatory cnns and other nucleotide-level scores to improve genome-wide variant predictions. *Nucleic Acids Research*, 52(D1):D1143–D1154, 01 2024.

[133] Nadav Brandes, Grant Goldman, Charlotte H. Wang, Chun Jimmie Ye, and Vasilis Ntranos. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, 55(9):1512–1522, September 2023.

[134] Melissa J. Landrum, Jennifer M. Lee, George R. Riley, Wonhee Jang, Wendy S. Rubinstein, Deanna M. Church, and Donna R. Maglott. ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Research*, 42(D1):D980–D985, 11 2013.

[135] Patricia J. Sullivan, Julian M.W. Quinn, Weilin Wu, Mark Pinese, and Mark J. Cowley. SpliceVarDB: A comprehensive database of experimentally validated human splicing variants. *The American Journal of Human Genetics*, 111(10):2164–2175, October 2024. Publisher: Elsevier.

[136] Gregory M. Findlay, Riza M. Daza, Beth Martin, Melissa D. Zhang, Anh P. Leith, Molly Gasperini, Joseph D. Janizek, Xingfan Huang, Lea M. Starita, and Jay Shendure. Accurate classification of *BRCA1* variants with saturation genome editing. *Nature*, 562:217–222, 10 2018.

[137] Huaizhi Huang, Chunling Hu, Jie Na, Steven N Hart, Rohan David Gnanaolivu, Mohamed Abozaid, Tara Rao, Yohannes A Tecleab, Tina Pesaran, et al. Functional evaluation and clinical classification of *BRCA2* variants. *Nature*, pages 1–10, 2025.

[138] Aman Patel, Arpita Singhal, Austin Wang, Anusri Pampari, Maya Kasowski, and Anshul Kundaje. DART-Eval: A comprehensive DNA language model evaluation benchmark on regulatory DNA. *Advances in Neural Information Processing Systems*, 37:62024–62061, 2024.

[139] Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. *arXiv*, 2412.06410, 2024.

[140] Alireza Makhzani and Brendan Frey. k-sparse autoencoders. *arXiv*, 2014.

[141] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv*, 2024.

[142] Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer El Showk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, June 2022.

[143] Antonio Pedro Camargo, Simon Roux, Frederik Schulz, Michal Babinski, Yan Xu, Bin Hu, Patrick S G Chain, Stephen Nayfach, and Nikos C Kyrpides. Identification of mobile genetic elements with geNomad. *Nat. Biotechnol.*, 42(8):1303–1312, August 2024.

[144] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

[145] Patrick Kunzmann, Tom David Müller, Maximilian Greil, Jan Hendrik Krumbach, Jacob Marcel Anter, Daniel Bauer, Faisal Islam, and Kay Hamacher. Biotite: new tools for a versatile python bioinformatics library. *BMC Bioinformatics*, 24(1):236, June 2023.

[146] Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, et al. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596, 2021.

[147] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 2024.

[148] Elaine C. Meng, Thomas D. Goddard, Eric F. Pettersen, Greg S. Couch, Zach J. Pearson, John H. Morris, and Thomas E. Ferrin. UCSF ChimeraX: Tools for structure building and analysis. *Protein Science*, 32(11):e4792, 2023.

[149] Valerie A. Schneider, Tina Graves-Lindsay, Kerstin Howe, Nathan Bouk, Hsiu-Chuan Chen, Paul A. Kitts, Terence D. Murphy, Kim D. Pruitt, Françoise Thibaud-Nissen, Derek Albracht, Robert S. Fulton, Milinn Kremitzki, Vincent Magrini, Chris Markovic, Sean McGrath, Karyn Meltz Steinberg, Kate Auger, William Chow, Joanna Collins, Glenn Harden, Timothy Hubbard, Sarah Pelan, Jared T. Simpson, Glen Threadgold, James Torrance, Jonathan M. Wood, Laura Clarke, Sergey Koren, Matthew Boitano, Paul Peluso, Heng Li, Chen-Shan Chin, Adam M. Phillippy, Richard Durbin, Richard K. Wilson, Paul Flicek, Evan E. Eichler, and Deanna M. Church. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Research*, 27(5):849–864, May 2017. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.

[150] Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C Lin, Peter Laslo, Jason X Cheng, Cornelis Murre, Harinder Singh, and Christopher K Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular Cell*, 38(4):576–589, 2010.

[151] Ilya E Vorontsov, Irina A Eliseeva, Arsenii Zinkevich, Mikhail Nikonov, Sergey Abramov, Alexandr Boytsov, Vasily Kamenets, Alexandra Kasianova, Semyon Kolmykov, Ivan S Yevshin, Alexander Favorov, Yulia A Medvedeva, Arttu Jolma, Fedor Kolpakov, Vsevolod J Makeev, and Ivan V Kulakovskiy. HOCOMOCO in 2024: a rebuild of the curated collection of binding models for human and mouse transcription factors. *Nucleic Acids Research*, 52(D1):D154–D163, November 2023.

[152] Shobhit Gupta, John A. Stamatoyannopoulos, Timothy L. Bailey, and William Stafford Noble. Quantifying similarity between motifs. *Genome Biology*, 8(2):R24, February 2007.

[153] Timothy L Bailey and Charles E Grant. SEA: simple enrichment analysis of motifs. *BioRxiv*, pages 2021–08, 2021.

[154] David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv*, 2409.14507, 2025.

[155] Avanti Shrikumar, Katherine Tian, Žiga Avsec, Anna Shcherbina, Abhimanyu Banerjee, Mahfuza Sharmin, Surag Nair, and Anshul Kundaje. Technical note on transcription factor motif discovery from importance scores (TF-MoDISco) version 0.5. 6.5. *arXiv preprint arXiv:1811.00416*, 2018.

[156] Marcela Sandoval-Velasco, Olga Dudchenko, Juan Antonio Rodríguez, Cynthia Pérez Estrada, Marianne Dehasque, Claudia Fontsere, Sarah ST Mak, Ruqayya Khan, Vinícius G Contessoto, Antonio B Oliveira Junior, et al. Three-dimensional genome architecture persists in a 52,000-year-old woolly mammoth skin sample. *Cell*, 187(14):3541–3562, 2024.

[157] Oliver Schwengers, Lukas Jelonek, Marius Alfred Dieckmann, Sebastian Beyvers, Jochen Blom, and Alexander Goesmann. Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. *Microb. Genom.*, 7(11), November 2021.

[158] James T Robinson, Helga Thorvaldsdottir, Douglass Turner, and Jill P Mesirov. igv.js: an embeddable javascript implementation of the Integrative Genomics Viewer (IGV). *Bioinformatics*, 39(1):btac830, 12 2022.

[159] Guanliang Meng, Yiyuan Li, Chentao Yang, and Shanlin Liu. MitoZ: a toolkit for animal mitochondrial genome assembly, annotation and visualization. *Nucleic Acids Research*, 47(11):e63–e63, 2019.

[160] Artyom A. Egorov and Gemma C. Atkinson. LoVis4u: Locus visualisation tool for comparative genomics. *bioRxiv*, 2024.

[161] Robert D. Finn, Jody Clements, and Sean R. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, 39(suppl_2):W29–W37, 05 2011.

[162] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik LL Sonnhammer, Silvio CE Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, et al. Pfam: The protein families database in 2021. *Nucleic acids research*, 49(D1):D412–D419, 2021.

[163] Estelle Proux-Wéra, David Armisén, Kevin P Byrne, and Kenneth H Wolfe. A pipeline for automated annotation of yeast genome sequences by a conserved-synteny approach. *BMC Bioinformatics*, 13:1–12, 2012.

[164] Steen Knudsen. Promoter2.0: for the recognition of PolII promoter sequences. *Bioinformatics (Oxford, England)*, 15(5):356–361, 1999.

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

[166] Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R Kelley. Predicting RNA-seq coverage from dna sequence as a unifying model of gene regulation. *Nature Genetics*, pages 1–13, 2025.

[167] Johannes C Hingerl, Alexander Karollus, and Julien Gagneur. Flashzoi: An enhanced Borzoi model for accelerated genomic analysis. *bioRxiv*, pages 2024–12, 2024.

[168] Sudarshan Pinglay, Milica Bulajić, Dylan P Rahe, Emily Huang, Ran Brosh, Nicholas E Mamrak, Benjamin R King, Sergei German, John A Cadley, Lila Rieber, et al. Synthetic regulatory reconstitution reveals principles of mammalian Hox cluster regulation. *Science*, 377(6601):eabk2820, 2022.

[169] Ran Brosh, Jon M Laurent, Raquel Ordoñez, Emily Huang, Megan S Hogan, Angela M Hitchcock, Leslie A Mitchell, Sudarshan Pinglay, John A Cadley, Raven D Luther, et al. A versatile platform for locus-scale genome rewriting and verification. *Proceedings of the National Academy of Sciences*, 118(10):e2023952118, 2021.

[170] Sudarshan Pinglay, Jean-Benoît Lalanne, Riza M Daza, Sanjay Kottapalli, Faaiz Quaisar, Jonas Koeppel, Riddhiman K Garge, Xiaoyi Li, David S Lee, and Jay Shendure. Multiplex generation and single-cell analysis of structural variants in mammalian genomes. *Science*, 387(6733):eado5978, 2025.

[171] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, 2012.

[172] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[173] Fidel Ramírez, Devon P Ryan, Björn Grüning, Vivek Bhardwaj, Fabian Kilpert, Andreas S Richter, Steffen Heyne, Friederike Dündar, and Thomas Manke. deepTools2: A next generation web server for deep-sequencing data analysis. *Nucleic Acids Research*, 44(Web Server issue):W160, 2016.

[174] Charles E Grant, Timothy L Bailey, and William Stafford Noble. FIMO: Scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

[175] Alison Fanton, Liam J. Bartie, Juliana Q. Martins, Vincent Q. Tran, Laine Goudy, Matthew G. Durrant, Jingyi Wei, April Pawluk, Silvana Konermann, Alexander Marson, Luke A. Gilbert, and Patrick D. Hsu. Site-specific DNA insertion into the human genome with engineered recombinases. *bioRxiv*, 2024.

[176] Fiorella C Grandi, Hailey Modi, Lucas Kampman, and M Ryan Corces. Chromatin accessibility profiling by ATAC-seq. *Nature Protocols*, 17(6):1518–1552, 2022.

[177] Jason P Smith, M Ryan Corces, Jin Xu, Vincent P Reuter, Howard Y Chang, and Nathan C Sheffield. PEPATAC: an optimized pipeline for ATAC-seq data analysis with serial alignments. *NAR Genomics and Bioinformatics*, 3(4):lqab101, 11 2021.

[178] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 05 2009.

[179] Ole Tange. *GNU Parallel 2018*. Ole Tange, April 2018.

[180] Daniel Beck, Millissia Ben Maamar, and Michael K Skinner. Genome-wide CpG density and DNA methylation analysis method (MeDIP, RRBS, and WGBS) comparisons. *Epigenetics*, 17(5):518–530, 2022.

[181] J Widom. Short-range order in two eukaryotic genomes: relation to chromosome structure, 1996.

C. Appendix

C.1. Model training FLOPS comparison

Model	Model Size	Tokens	FLOPS est.
Fully open: data, infrastructure, weights			
Evo 2 40B	40.3B	9.3T	2.25×10^{24}
Pythia Large	12B	300B	2.16×10^{22}
OLMo 2 Large	13B	5.6T	4.37×10^{23}
Open data, weights			
Evo 1	7B	300B	1.26×10^{22}
Falcon 180B	180B	3.5T	3.78×10^{24}
StarCoder 2 15B	15B	4.3T	3.87×10^{23}
Open weights only			
DeepSeek V3	37B (671B total)	14.8T	3.28×10^{24}
Llama 3.1 Large	405B	15T	3.64×10^{25}
Closed			
ESM 3 Large	98B	771B	1.07×10^{24}
xTrimo Large	100B	1T	6.00×10^{23}

Supplementary Table 4 | Comparison of training FLOPs across flagship language and biology models, showing Evo 2 as the largest fully open model. FLOPs are estimated without accounting for mixed-precision or pre-training context length.

C.2. Repeat Down Weighting Ablation

We conducted experiments to assess the impact of repeat loss reweighting on downstream performance using two Striped Hyena 1 models (550M parameters, 8192 sequence length). Both models were trained on a differently weighted ablation dataset, which was enriched for complete eukaryotic genomes to better evaluate the effects of repeat down-weighting (**Supplementary Table 9**). The models differed only in their treatment of repetitive sequences: one applied a loss reweighting factor of 0.1 to lowercase sequences, while the other used no reweighting.

Performance was evaluated using the ClinVar pathogenic versus benign classification benchmark. The model without loss reweighting achieved an AUROC of 0.63 at 40,000 training steps, compared to 0.73 for the model with reweighting. Training of the non-reweighted model was discontinued after we observed no improvement in ClinVar performance beyond 30,000 steps, while the reweighted model continued to show improvements until the experiment concluded at 100,000 steps, reaching 0.82 AUROC. Based on these results, we adopted a loss reweighting factor of 0.1 for repetitive sequences in subsequent experiments and the final models.

Supplementary Table 5 | ClinVar AUROC at 40,000 Steps

Model	AUROC
No reweighting	0.63
Repeat reweighting 0.1	0.73

C.3. Data Composition Effect on Downstream Tasks

We compare Evo 2 7B base with a 7B parameter StripedHyena 2 model trained on an ablation data composition (**Supplementary Table 9**) at 8192 context length. This ablation dataset includes mRNA with the same weight as the final pretraining dataset, but used whole genomes instead of the eukaryotic genomic windows and augmented mRNA. We trained for 1.9T tokens until the loss plateaued. We compare the data ablation model with Evo 2 7B base model on zero-shot prediction of ClinVar, SpliceVarDB, and *BRCA1*, following the same protocol as before for these evaluations (**Section A.3.12**).

Zero-shot evaluations demonstrated improved performance for the final pretraining data composition compared to the data ablation, suggesting that focusing pretraining on functionally enriched regions around genes can significantly improve downstream performance. Zero-shot AUROC improves for *BRCA1* variants from 0.793 to 0.891, ClinVar SNV from 0.933 to 0.957, ClinVar non-SNV from 0.897 to 0.939, and SpliceVarDB (intronic and exonic) from 0.791 to 0.826. We improvements for all subsets except for ClinVar coding SNVs, which slightly decrease. While the data ablation has more weight to noncoding regions, the high effect noncoding variants considered by these benchmarks are better modeled by the Evo 2 7B base model which was pretrained on datasets focused around genes. Importantly, the final model learned to better calibrate these proximal noncoding and coding effects as shown by the improvements when considering coding and noncoding together. At the same time, Evo 2 7B base performs within 0.01 at DART-eval zero shot evaluations of the data ablation 7B model (task 1: 0.97 vs 0.98, task 2: 0.63 vs 0.64), suggesting that added pretraining on whole genomes did not help substantially even for tasks on enhancer regions, consistent with findings from recent analysis of DNA language models [138]. These results highlight the importance of data engineering in DNA language models.

Supplementary Table 6 | Comparing Evo 2 7B base and data ablation model on zero-shot ClinVar classification prediction

Metric	Subset	7B data ablation	Evo 2 7B base	Difference
AUROC	All SNV	0.933	0.957	+0.024
AUPRC	All SNV	0.910	0.927	+0.017
AUROC	All non-SNV	0.897	0.939	+0.042
AUPRC	All non-SNV	0.846	0.898	+0.052
AUROC	Coding SNV	0.851	0.829	-0.022
AUPRC	Coding SNV	0.908	0.883	-0.025
AUROC	Coding non-SNV	0.893	0.918	+0.025
AUPRC	Coding non-SNV	0.946	0.954	+0.008
AUROC	Noncoding SNV	0.943	0.976	+0.033
AUPRC	Noncoding SNV	0.911	0.956	+0.045
AUROC	Noncoding non-SNV	0.867	0.929	+0.062
AUPRC	Noncoding non-SNV	0.726	0.825	+0.099

Supplementary Table 7 | Comparing Evo 2 7B base and data ablation model on zero-shot SpliceVarDB classification

Metric	Subset	7B data ablation	Evo 2 7B base	Difference
AUROC	All	0.791	0.826	+0.035
AUPRC	All	0.845	0.873	+0.028
AUROC	Exonic	0.666	0.687	+0.021
AUPRC	Exonic	0.510	0.540	+0.030
AUROC	Intronic	0.893	0.923	+0.030
AUPRC	Intronic	0.956	0.968	+0.012

Supplementary Table 8 | *BRCA1* zero-shot comparison between models

Metric	**Subset**	**7B data ablation**	**Evo 2 7B base**	**Difference**
Spearman *r*	Overall	0.358	0.513	+0.155
AUROC	Overall	0.793	0.891	+0.098
AUPRC	Overall	0.543	0.713	+0.170
AUROC	Coding	0.769	0.797	+0.028
AUPRC	Coding	0.522	0.534	+0.012
AUROC	Noncoding	0.867	0.962	+0.095
AUPRC	Noncoding	0.658	0.869	+0.211

Supplementary Table 9 | Pretraining, midtraining, and data ablation experiment data composition

Dataset	Pretraining (%)	Midtraining (%)	Ablation (%)
GTDB + IMG/PR	18	24	9
Metagenomics (MGD DB)	24	5	15
IMG/VR	3	2	1.7
Organelles	0.5	0.25	0
Euk Promoters	0.02	0.01	0.02
ncRNA	2	1	2
Euk mRNAs	9	4.5	9
Euk augmented transcripts	9	4.5	0
Euk 5 kb windows	35	5	0
NCBI: Animalia	0	36	45
NCBI: Plantae	0	12	8.6
NCBI: Fungi	0	4	8
NCBI: Protista	0	0.8	0.8
NCBI: Chromista	0	0.8	0.6
hg38	0	0	0.3

Supplementary Table 10 | BLASTp results for genes from Evo 2-generated mitochondria (**Fig. 5e**). The amino acid percent sequence ID is obtained by multiplying the query cover with the sequence identity.

Annotation	Sequence ID %	Nearest Species
mt-Co1	86.94	*Muntiacus reevesi*
mt-Atp8	77.27	*Myotis albescens*
mt-Nd4	84.95	*Semilabeo notabilis*
mt-Nd1	84.59	*Phocoena phocoena*
mt-Co3	77.31	*Alces alces*
MT-ND4L	86.73	*Bos grunniens*
mt-Nd5	78.99	*Schizothorax argentatus*
MT-ND2	68.45	*Rusa unicolor*
mt-Co2	88.54	*Ovis aries*
mt-Atp6	91.59	*Alcelaphus buselaphus*
mt-Nd3	87.11	*Bos indicus*
mt-Cytb	75.08	*Davidijordania poecilimon*
mt-Nd6	94.43	*Neolissochilus hendersoni*

Supplementary Table 11 | DART-eval task 2: Quantiles of motif identification accuracies for each model

Model	0%	25%	50%	75%	100%
Evo 2 40B	0.020	0.475	0.670	0.8300	1.000
Evo 2 7B	0.035	0.480	0.670	0.8300	1.000
Evo 2 7B Base	0.035	0.490	**0.675**	**0.8250**	1.000
Nucleotide Transformer	**0.200**	0.465	0.565	0.6575	0.995
HyenaDNA	0.000	0.420	0.645	0.8200	0.995
GENA-LM	0.055	0.475	0.620	0.7400	1.000
DNABERT-2	0.145	**0.495**	0.590	0.6850	1.000