Sequence modeling and design from molecular to genome scale with Evo

Eric Nguyen[1,2], Michael Poli[†,3], Matthew G. Durrant[*2], Armin W. Thomas[1], Brian Kang[1], Jeremy Sullivan[2], Madelena Y. Ng[1], Ashley Lewis[1], Aman Patel[1], Aaron Lou[1], Stefano Ermon[1,4], Stephen A. Baccus[1], Tina Hernandez-Boussard[1], Christopher Ré[1], Patrick D. Hsu[†,2,5], and Brian L. Hie[†,1,2]
*Equal contribution.
†Corresponding author. B.L.H. (brianhie@stanford.edu); P.D.H. (patrick@arcinstitute.org).

Abstract

The genome is a sequence that completely encodes the DNA, RNA, and proteins that orchestrate the function of a whole organism. Advances in machine learning combined with massive datasets of whole genomes could enable a biological foundation model that accelerates the mechanistic understanding and generative design of complex molecular interactions. We report Evo, a genomic foundation model that enables prediction and generation tasks from the molecular to genome scale. Using an architecture based on advances in deep signal processing, we scale Evo to 7 billion parameters with a context length of 131 kilobases (kb) at single-nucleotide, byte resolution. Trained on 2.7M prokaryotic and phage genomes, Evo can generalize across the three fundamental modalities of the central dogma of molecular biology to perform zero-shot function prediction that is competitive with, or outperforms, leading domain-specific language models. Evo also excels at multi-element generation tasks, which we demonstrate by generating synthetic CRISPR-Cas molecular complexes and entire transposable systems for the first time. Using information learned over whole genomes, Evo can also predict gene essentiality at nucleotide resolution and can generate coding-rich sequences up to 650 kb in length, orders of magnitude longer than previous methods. Advances in multi-modal and multiscale learning with Evo provides a promising path toward improving our understanding and control of biology across multiple levels of complexity.
基因组是一个序列，完整编码了协调整个生物体功能的 DNA、RNA 和蛋白质。结合机器学习的进展与大规模全基因组数据集，有望实现一个生物学基础模型，从而加速对复杂分子相互作用机制的理解与生成式设计。我们提出了 Evo，这是一个支持从分子到基因组尺度预测与生成任务的基因组基础模型。该模型采用基于深度信号处理进展的架构，具备 131 千碱基（kb）上下文长度，在单核苷酸、字节级分辨率下扩展至 70 亿参数规模。Evo 在 270 万个原核生物和噬菌体基因组上进行训练，能够在分子生物学中心法则的三种基本模态之间实现泛化，执行零样本功能预测，并与领先的领域专用语言模型媲美或超越它们。Evo 同样擅长多元件生成任务，我们首次展示了其生成合成 CRISPR-Cas 分子复合物和完整转座系统的能力。Evo 利用对整个基因组的学习信息，还可在单碱基分辨率下预测基因必需性，并能生成最长达 650 kb 的富编码序列，其长度比以往方法高出多个数量级。Evo 在多模态与多尺度学习方面的进展，为我们在多个复杂层级上提升对生物系统的理解与控制提供了有前景的路径。

Introduction 引言

DNA is the fundamental layer of biological information that is responsible for transmitting the results of evolution across generations of life (Morgan, 1910; Watson and Crick, 1953; Nirenberg and Matthaei, 1961). Evolutionary variation in genome sequences is a reflection of adaptation and selection for biological function at the phenotypic level (Dobzhansky, 1951). Rapid advances in DNA sequencing technologies have enabled the systematic mapping of this evolutionary diversity at the whole-genome scale.
DNA 是生物信息的基本层，负责跨代传递进化的成果（Morgan, 1910；Watson 与 Crick, 1953；Nirenberg 与 Matthaei, 1961）。基因组序列中的进化变异反映了表型层面对生物功能的适应与选择（Dobzhansky, 1951）。DNA 测序技术的快速发展，使我们能够在全基因组尺度上系统地描绘这种进化多样性。

A machine that learns this breadth of information across genomes could model the function of DNA, RNA, and proteins, as well as their diverse interactions that orchestrate complex biological functions, mediate disease, or create a complete organism. Modern machine learning algorithms combined with massive datasets of genomic sequences could enable a general biological foundation model that learns the intrinsic logic of whole genomes.
一个能够学习全基因组范围信息的机器模型，可以模拟 DNA、RNA 和蛋白质的功能，以及它们之间协调复杂生物学功能、介导疾病或构建完整生物体的多样化相互作用。将现代机器学习算法与海量基因组序列数据相结合，有望实现一个通用的生物学基础模型，从而掌握完整基因组的内在逻辑。

However, current efforts to model molecular biology with machine learning have been focused on creating modality-specific models that are specialized to proteins, regulatory DNA, or RNA (Jumper et al., 2021; Rives et al., 2021; Avsec et al., 2021; Theodoris et al., 2023). In addition, generative applications in biology have been limited to the design of single molecules, simple complexes (Watson et al., 2023; Madani et al., 2023;
然而，目前使用机器学习建模分子生物学的工作多集中于开发针对特定模态的模型，这些模型通常专注于蛋白质、

调控性 DNA 或 RNA（Jumper 等，2021；Rives 等，2021；Avsec 等，2021；Theodoris 等，2023）。此外，生物学中的生成式应用也主要局限于单分子或简单复合物的设计（Watson 等，2023；Madani 等，2023）。

Results **结果**

Modeling long sequences at nucleotide resolution with the StripedHyena architecture **使用** StripedHyena **架构在单碱基分辨率下建模长序列**

Evo is a genomic foundation model with 7B parameters trained with a context length of up to 131 k tokens, using single-nucleotide, byte-level tokenization. To model long sequences at nucleotide resolution efficiently, which we demonstrate by generating sequences over 650 k tokens, we leveraged the StripedHyena architecture (Poli et al., 2023b) (Figure ??A) that builds on emerging techniques in deep signal processing (Li et al., 2020; Gu et al., 2021; Orvieto et al., 2023; Massaroli et al., 2024). The model is a hybrid of 29 layers of data-controlled convolutional operators (hyena layers) interleaved with 3 layers (10%) of multi-head attention equipped with rotary position embeddings (RoPE) (Su et al., 2024) (Methods).

Evo 是一个具有 70 亿参数的基因组基础模型，训练时上下文长度可达 131 千标记（token），并采用单核苷酸、字节级的标记方式。为高效地在单碱基分辨率下建模长序列（我们展示其可生成超过 65 万标记的序列），我们采用了基于深度信号处理新兴技术的 StripedHyena 架构（Poli 等，2023b）（见图 ??A），该技术融合了多项最新研究（Li 等，2020；Gu 等，2021；Orvieto 等，2023；Massaroli 等，2024）。该模型由 29 层数据控制的卷积算子（hyena 层）与 3 层（占比 10%）多头注意力机制交错组成，注意力层使用了旋转位置编码（RoPE）（Su 等，2024）（见方法）。

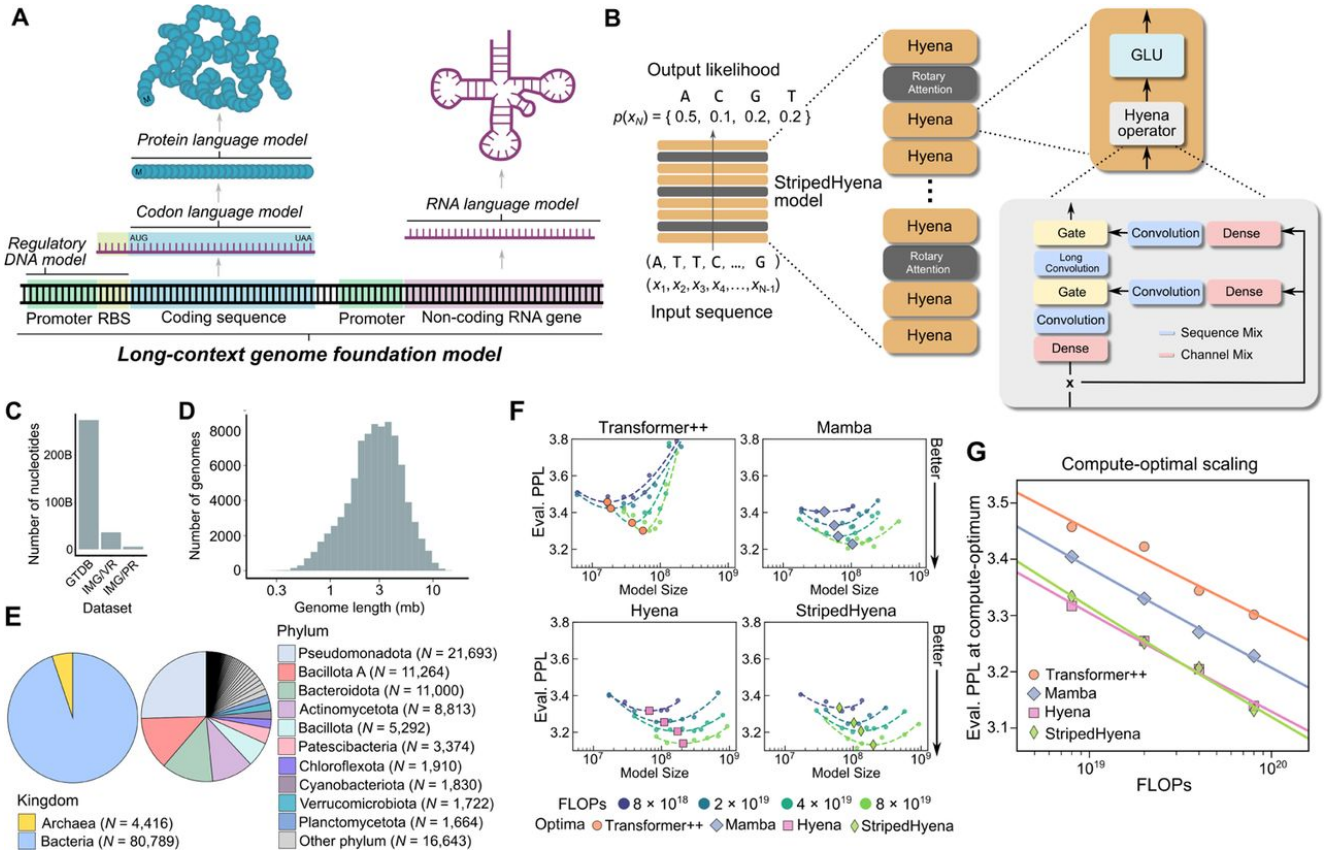

Figure 1: Pretraining a genomic foundation model across prokaryotic life. (A) A model of genome sequences at single-nucleotide resolution could learn all of the information encoded in regulatory DNA and in the sequences of the other modalities within the central dogma (proteins, coding RNA, and noncoding RNA). Even further, it could learning covariation involving multiple genes and regulatory elements. The status of DNA as the fundamental layer of biological information makes it a productive modality at which to develop a biological foundation model. (B) A model that predicts the likelihood of the next token given a sequence of tokens, referred to as autoregressive modeling, can learn complex patterns underlying DNA sequences. StripedHyena is a deep signal processing architecture for long sequences, obtained by hybridizing attention and hyena operators. (C) We pretrained Evo, a 7B parameter model with the StripedHyena architecture, on bacterial genome sequences from GTDB and IMG/PR and viral sequences from IMG/VR, excluding sequences from viruses that infect eukaryotic hosts. (D) A histogram depicting the sequence length of the genomes in GTDB. mb: megabases. (E) Pie charts depicting the taxonomic makeup of GTDB based on the kingdom (left) and phylum (right). (F) Results from a first-of-its-kind scaling laws analysis for large-scale DNA pretraining. Models improve monotonically with scale, with significant differences between architectures. Eval. PPL: evaluation perplexity. (G) To determine optimal architecture and scaling for Evo, we compared scaling rates of different models pretrained on the compute–optimal frontier, i.e., with optimal allocation of compute between dataset size and model size. Eval. PPL: evaluation perplexity. FLOPs: Floating point operations. Ingraham et al., 2023), or short DNA sequences (DaSilva et al., 2024; Lal et al., 2024). In contrast, complex biological processes, such as gene regulation, CRISPR immunity, or genetic transposition, rely on many interactions involving molecules across multiple modalities.

Model hybridization, first proposed to address shortcomings of state-space models (Ma et al., 2022; Fu et al., 2022; Pilault et al., 2024) has recently been shown to improve scaling performance on language modeling of both standalone Hyena and Transformer architectures (Poli et al., 2023b). StripedHyena is designed to benefit from the specialization of each of its layer types, with hyena layers implementing the bulk of the computation required for sequence processing and attention layers supplementing the ability to recall information from the context of an input.

模型混合最初为解决状态空间模型的局限性而提出（Ma 等，2022；Fu 等，2022；Pilault 等，2024），近期的研究表明，这种方法可显著提升 Hyena 与 Transformer 架构在语言建模任务中的扩展性能（Poli 等，2023b）。StripedHyena 的设计旨在发挥各类层结构的专长：hyena 层执行大部分序列处理所需计算，注意力层则增强模型对输入上下文中信息的回溯能力。

Hyena layers process sequences in an input-dependent manner via compositions of short and long convolution filters (Figure ??B), making the layer especially effective at filtering noisy patterns that can occur in DNA and at aggregating individual nucleotides into motifs. Compared to HyenaDNA (Nguyen et al., 2023), a previous

generation of DNA models leveraging a Hyena architecture (Poli et al., 2023a), Evo is based on an improved hybrid design and scaled to 1000× larger model size and 100× more data.
Hyena 层通过组合短程与长程卷积核，以输入依赖的方式处理序列（见图 ??B），在过滤 DNA 中常见的噪声模式以及将单个碱基聚合为功能基序方面尤其有效。相较于基于 Hyena 架构的前一代 DNA 模型 HyenaDNA（Nguyen 等，2023；Poli 等，2023a），Evo 在架构上实现了混合设计的改进，并在模型规模上扩大了 1000 倍、训练数据量上提升了 100 倍。

Training Evo at scale on OpenGenome 在 OpenGenome 上大规模训练 Evo

We compiled a large genome dataset called OpenGenome (Methods) with over 80,000 bacterial and archaeal genomes, and millions of predicted prokaryotic phage and plasmid sequences, covering 300B nucleotide tokens (Figures ??C and S1) (Parks et al., 2022; Camargo et al., 2023, 2024). For safety considerations, we excluded viral genomes that infect eukaryotic hosts. Like most language models, Evo is pretrained via a next-token prediction objective on raw genome sequences with no explicit supervision or annotations. In order to predict the next token given a sequence of tokens, the model must learn the distribution of the genome data and be aware of the biological sequence motifs found in the collected genomes. Pretraining involves 2 stages: the first stage uses a context length of 8 k tokens, while the second context extension stage uses 131 k tokens as context. Depending on the downstream task, we select a base model from one of the two stages to finetune on smaller datasets of interest for generation.
我们构建了一个名为 OpenGenome 的大规模基因组数据集（见方法），其中包含超过 80,000 个细菌和古菌的基因组，以及数百万条预测的原核噬菌体和质粒序列，覆盖总计 3000 亿个核苷酸标记（见图 ??C 与附图 S1）（Parks 等，2022；Camargo 等，2023，2024）。出于安全考虑，我们排除了感染真核生物的病毒基因组。与大多数语言模型类似，Evo 通过对原始基因组序列执行下一个标记预测目标进行预训练，无需显式监督或注释。为了预测下一个标记，模型必须学习基因组数据的分布，并识别收集到的基因组中的生物序列基序。预训练包括两个阶段：第一阶段使用 8k 标记的上下文长度，第二阶段扩展上下文至 131k 标记。根据下游任务，我们从这两个阶段中选择一个基础模型，在较小的数据集上进行微调以用于生成。

StripedHyena demonstrates favorable scaling laws on DNA sequence data 在 DNA 序列建模中展现优越的扩展规律

Aiding our model design, we performed the first scaling laws analysis (to our knowledge) for DNA sequence modeling. The main objective of this type of analysis is to determine the relationship between training, architectural details, and performance metrics via a systematic experimental protocol (Hoffmann et al., 2022; Kaplan et al., 2020). Once a set of scaling laws is obtained, it can then be used as a guide to optimally scale training to larger models and datasets.
为辅助模型设计，我们首次（据我们所知）对 DNA 序列建模进行了扩展规律分析。该类分析的主要目标是通过系统化的实验协议，揭示训练量、模型架构细节与性能指标之间的关系（Hoffmann 等，2022；Kaplan 等，2020）。一旦得到扩展规律，就可以用作指导，将训练规模优化至更大的模型与数据集。

Here, we compare different classes of architectures via a compute-optimal protocol, aimed at evaluating results on the compute-optimal frontier (Methods). We trained over 300 models across four architectures: Transformer++, Mamba, Hyena, and StripedHyena. Transformer++ is a state-of-the-art Transformer, and Mamba is a modern architecture using data-controlled state-space models (Gu and Dao, 2023).
在此，我们使用计算最优协议对多种架构进行了比较，旨在评估其在计算最优前沿的表现（见方法）。我们在四种架构（Transformer++、Mamba、Hyena 和 StripedHyena）上共训练了超过 300 个模型。Transformer++ 是先进的 Transformer 架构，而 Mamba 是一种采用数据控制状态空间模型的新型架构（Gu 与 Dao，2023）。

We found Transformer++ to yield significantly worse perplexity (a measure of next token prediction quality) at all compute budgets (Figures ??G), a symptom of the inefficiency of the architecture at the byte resolution. State-space and deep signal processing architectures are observed to improve on the scaling rate over Transformer++, with Hyena and StripedHyena resulting in the best scaling rate. We observed stable training for StripedHyena throughout all the studied model sizes and learning rates during the scaling analysis.
我们发现，无论在何种计算预算下，Transformer++ 的困惑度（衡量下一个标记预测质量）都显著较差（见图 ??G），表明其在字节分辨率下的建模效率不足。相比之下，状态空间与深度信号处理架构在扩展率上均优于 Transformer++，其中 Hyena 与 StripedHyena 表现最佳。在所有测试的模型尺寸与学习率条件下，StripedHyena 的训练过程均表现出高度稳定性。

We also compare architecture performance outside the compute-optimal frontier, namely with allocations of the computational budget that may be suboptimal. Performance outside the compute-optimal frontier is important in practice, as most models (including Evo) are trained for more tokens than recommended by compute-optimal scaling laws. We estimate 250 billion to be the compute-optimal number of tokens for Evo 7B given the FLOP budget, meaning the model was trained at a 17% offset from the compute-optimal model size during the initial 8192 sequence length pretraining phase of 300 billion tokens. Both Transformer++ and Mamba experienced numerical instability during training, and suffered from a sharper performance degradation of the scaling rate outside the compute-optimal frontier, in contrast to StripedHyena (further analysis in Figure S3). These findings motivate the choice of StripedHyena as the architecture for Evo.

我们还比较了在非计算最优前沿条件下各架构的性能，即在计算预算分配不理想时的表现。实际中，这种情形尤为重要，因为大多数模型（包括 Evo）在训练中使用的 token 数量远超计算最优扩展规律的建议值。我们估算，在当前的 FLOP 预算下，对于 Evo 7B，计算最优的训练 token 数为 2500 亿，而模型实际在预训练阶段（序列长度为 8192）使用了 3000 亿 token，相当于偏离最优模型大小约 17%。Transformer++ 与 Mamba 在训练过程中均出现数值不稳定性，且在非最优条件下的扩展率下降更为剧烈，相比之下 StripedHyena 更具稳定性（更详尽分析见图 S3）。这些结果支持将 StripedHyena 作为 Evo 架构的决策。

Evo performs zero-shot function prediction across DNA, RNA, and protein modalities
Evo **实现** DNA、RNA **和蛋白质模态的零样本功能预测**

Predicting mutational effects on protein function **预测突变对蛋白质功能的影响**

Beyond evaluating perplexity, we investigated the model's zero-shot performance on biologically relevant downstream tasks. For example, language models specifically trained on large corpuses of protein sequences or nucleotide coding sequences have demonstrated an impressive ability to predict mutational effects on protein function (Meier et al., 2021; Notin et al., 2022; Benegas et al., 2023) without any task-specific finetuning or supervision. Because Evo is trained on long genomic sequences that contain protein coding sequences, we tested whether the model would also learn the protein language well enough to perform zero-shot protein function prediction.
除了评估困惑度外，我们还研究了该模型在与生物学相关的下游任务中的零样本性能。例如，专门在大规模蛋白质序列或核苷酸编码序列语料上训练的语言模型，已展现出无需任务特定微调或监督即可预测突变对蛋白质功能影响的能力（Meier 等，2021；Notin 等，2022；Benegas 等，2023）。由于 Evo 是在包含蛋白质编码序列的长基因组序列上进行训练的，我们测试了该模型是否也能充分掌握蛋白质语言，从而实现零样本蛋白质功能预测。

Following work in evaluation of protein language models, we leveraged deep mutational scanning (DMS) studies which introduce an exhaustive set of mutations to a protein coding sequence and then experimentally measure the effects of these mutations on various definitions of fitness (fitness is a study-specific metric quantifying how well a protein performs a certain function) (Notin et al., 2022, 2023; Livesey and Marsh, 2023). The language-model likelihood or pseudolikelihood (Methods) of the amino acid sequence is used to predict the experimental fitness score (Figure ??A). To adapt this task to nucleotide sequences, we use the wild-type coding sequence and nucleotide mutations reported in the original DMS studies (Methods).
借鉴蛋白质语言模型的评估方法，我们利用深度突变扫描（DMS）研究数据，这些研究在蛋白质编码序列中引入详尽的突变集，并通过实验测量这些突变对不同定义的适应度的影响（适应度是一个研究特定的指标，衡量蛋白质执行某项功能的能力）（Notin 等，2022，2023；Livesey 与 Marsh，2023）。我们使用氨基酸序列的语言模型似然值或伪似然值（见方法）来预测实验测得的适应度得分（图 ??A）。为适配核苷酸层面的任务，我们采用了原始 DMS 研究中报告的野生型编码序列及突变信息（见方法）。

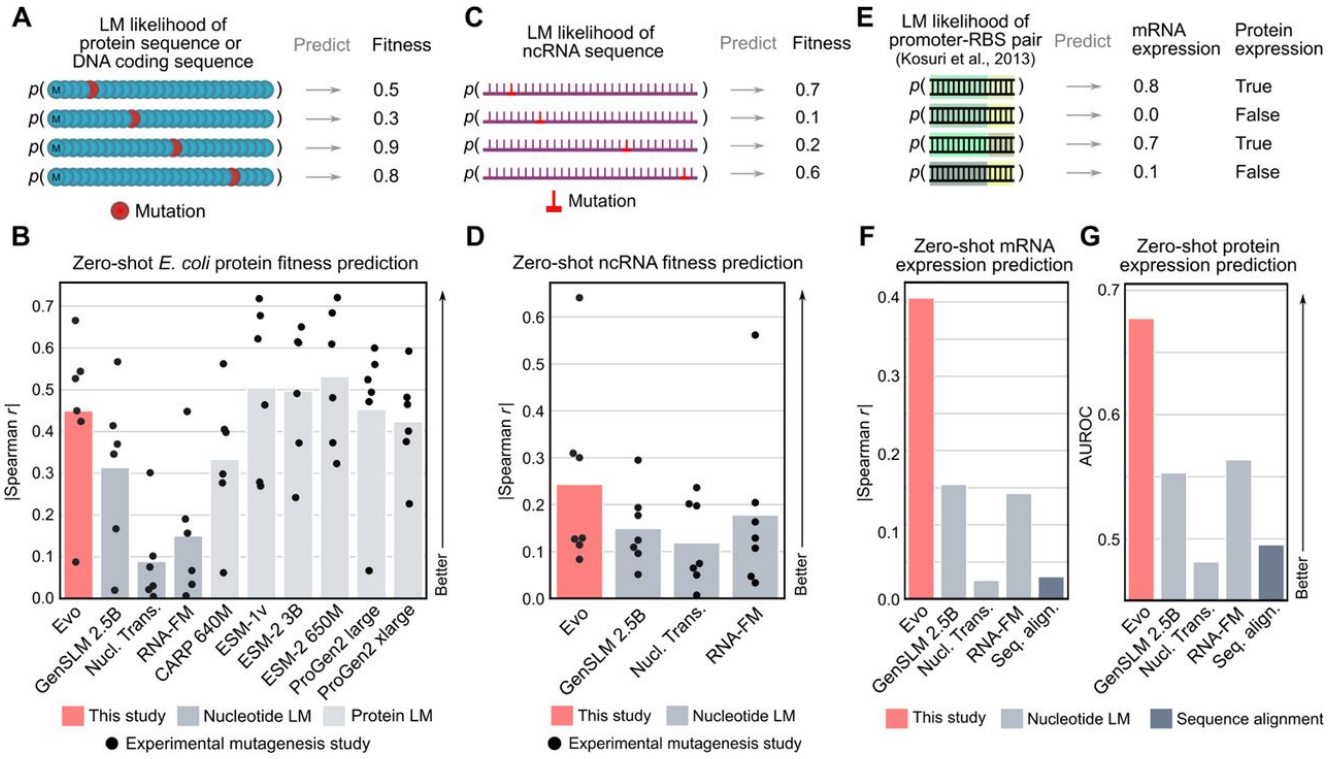

Figure 2: Evo performs zero-shot function prediction for proteins, non-coding RNAs, and regulatory DNA. (A) We obtained deep mutational scanning (DMS) datasets in which many mutations are made to a protein and a corresponding fitness score is experimentally measured for each protein variant. On the same set of mutated sequences, we compute its likelihood (or pseudolikelihood) under a protein language model or a nucleotide language model (LM). We then correlated these likelihoods with the experimental fitness measurements and used the strength of the correlation to measure the performance of zero-shot function prediction. (B) Evo has comparable predictive performance, measured via Spearman correlation, to state-of-the-art protein language models and higher performance than nucleotide language models. Bar height indicates the mean; each dot indicates a different DMS study. LM: language model; Nucl. Trans.: Nucleotide Transformer. (C) We obtained datasets in which many mutations are made to a ncRNA and a corresponding fitness score is experimentally measured. Predictive performance is measured as in the method described in (A). (D) Evo exhibits higher performance than nucleotide language models at predicting mutational effects on ncRNA function. Bar height indicates the mean; each dot indicates a different DMS study. LM: language model; Nucl. Trans.: Nucleotide Transformer. (E) We obtained a dataset in which Kosuri et al. (2013) measured mRNA and protein expression of a gene downstream of $\sim$ 12k promoter-RBS pairs in E. coli. For each promoter-RBS pair, we computed the likelihood of the sequence under a language model or a score indicating the frequency with which a promoter-RBS pair is observed in bacterial genomes. (F, G) Evo has higher predictive performance of mRNA and protein expression compared to nucleotide language models and to methods for computing the frequency of promoter-RBS pairs based on sequence alignment ("Seq. align."). Bar height indicates the mean; each dot indicates a different DMS study. LM: language model; Nucl. Trans.: Nucleotide Transformer.

Evo 可对蛋白质、非编码 RNA 以及调控性 DNA 执行零样本功能预测。(A) 我们获取了深度突变扫描（DMS）数据集，其中对蛋白质进行了多种突变，并为每种蛋白质变体实验测量了相应的适应性评分。我们在相同的一组突变序列上计算其在蛋白质语言模型或核苷酸语言模型（LM）下的似然（或伪似然），然后将这些似然与实验测得的适应性进行相关分析，相关强度即为零样本功能预测性能的衡量指标。(B) Evo 的预测性能（用 Spearman 相关系数衡量）与最先进的蛋白质语言模型相当，并优于其他核苷酸语言模型。柱高表示平均值；每个点表示一个不同的 DMS 研究。LM: language model；Nucl. Trans.: Nucleotide Transformer。(C) 我们获取了一些数据集，对 ncRNA 进行了多种突变，并测量了相应的适应性评分。预测性能的衡量方式与 (A) 所述方法相同。(D) Evo 在预测突变对 ncRNA 功能影响方面表现优于其他核苷酸语言模型。柱高表示平均值；每个点表示一个不同的 DMS 研究。LM: language model；Nucl. Trans.: Nucleotide Transformer。(E) 我们使用了 Kosuri 等人 (2013) 的数据集，其中测量了约 12,000 个启动子-RBS（核糖体结合位点）对在大肠杆菌中对下游基因的 mRNA 和蛋白质表达影响。对于每个启动子-RBS 对，我们计算了其在语言模型下的序列似然，或其在细菌基因组中出现频率的得分。(F, G) Evo 对 mRNA 和蛋白质表达的预测性能高于其他核苷酸语言模型，以及基于序列比对方法计算启动子-RBS 对频率的方法（"Seq. align."）。柱高表示平均值；每个点表示一个不同的 DMS 研究。LM: language model；Nucl. Trans.: Nucleotide Transformer。

When we evaluated Evo's zero-shot ability to predict mutational effects on protein function using DMS datasets

of E. coli proteins, we found that it outperformed all other nucleotide models tested (Figure ??B), including GenSLM (Zvyagin et al., 2023), a model explicitly trained only on coding sequences with a codon vocabulary (Figure ??A). Evo also reaches competitive performance with leading protein-specific language models (Yang et al., 2024; Meier et al., 2021; Lin et al., 2023; Madani et al., 2023) at this task (Figure ??B). Previous work has shown that improvement beyond this performance range is very difficult for protein language models with self-supervised pretraining alone (Li et al., 2024), indicating that Evo is already competitive with state-of-the-art protein language modeling on bacterial proteins. Notably, Evo is trained on long-context genomic sequences without any explicit coding sequence annotations. On DMS datasets of human proteins, Evo is unable to predict mutational effects on fitness (Figure S6A), most likely because the pretraining dataset only contains prokaryotic sequences. However, we observed a strong association between language-model perplexity on the wildtype sequence and fitness prediction performance (Figure S6B), indicating that additional finetuning or future pretraining on mammalian coding sequences could improve Evo's performance beyond bacterial proteins.
我们在大肠杆菌蛋白质的 DMS 数据集上评估了 Evo 的零样本突变功能预测能力，结果显示其性能优于所有其他测试的核苷酸语言模型（图 ??B），包括专门基于密码子词汇并仅在编码序列上训练的 GenSLM（Zvyagin 等，2023）（图 ??A）。在该任务中，Evo 也达到了与顶尖蛋白质语言模型（Yang 等，2024；Meier 等，2021；Lin 等，2023；Madani 等，2023）相当的性能（图 ??B）。先前研究表明，仅依赖自监督预训练的蛋白质语言模型很难超越该性能区间（Li 等，2024），这表明 Evo 在细菌蛋白建模上已具备与最先进方法竞争的能力。值得注意的是，Evo 的训练数据为长上下文的基因组序列，并未提供任何显式的编码序列注释。在人类蛋白的 DMS 数据集中，Evo 无法预测突变对适应度的影响（图 S6A），这可能是因为预训练数据集中仅包含原核生物序列。但我们观察到模型在野生型序列上的困惑度与适应度预测性能之间存在强关联（图 S6B），说明通过额外微调或在哺乳动物编码序列上的未来预训练，有望进一步提升 Evo 在非细菌蛋白上的表现。

Predicting mutational effects on ncRNA function **预测突变对非编码** RNA **功能的影响**

Next, we tested whether the same pretrained model could learn functional information about noncoding RNAs (ncRNA), such as tRNAs, rRNAs, and ribozymes. ncRNAs are encoded in the genome in a similar manner to proteins and they serve a variety of essential functions, including in protein synthesis and gene regulation. We collected ncRNA DMS (Methods), which are conceptually similar to protein DMS datasets but where mutations are made to the ncRNA sequence instead. We likewise evaluated Evo's ability to perform zero-shot ncRNA fitness prediction using the results of experimental ncRNA DMS studies as the ground truth score (Figure ??C).
接着，我们测试同一个预训练模型是否也能学习非编码 RNA（ncRNA，如 tRNA、rRNA 和核酶）的功能信息。ncRNA 在基因组中的编码方式与蛋白质类似，并在蛋白质合成和基因调控等方面发挥多种关键作用。我们收集了 ncRNA 的 DMS 数据集（见方法），这类数据集与蛋白 DMS 数据集在概念上类似，只是突变发生在 ncRNA 序列上。我们以实验 DMS 数据为真实分数，评估 Evo 在零样本 ncRNA 适应度预测中的能力（图 ??C）。

We found that Evo again outperforms all other tested nucleotide language models at this task, including RNA-FM (Chen et al., 2022), an RNA language model that is explicitly trained on ncRNA sequences (Figure ??D). We observed especially strong predictive performance on a study that measured the effects of mutations to the SS ribosomal RNA on the growth rate of E. coli (Spearman $r = 0.64$, two-sided t-distributed $P = 7.3 \times 10^{-4}$) (Zhang et al., 2009). Together with our results for protein sequences, these results indicate that Evo is able to learn from its prokaryotic genome training data to predict functional properties across different molecular modalities.
我们发现，Evo 在该任务中再次优于所有其他测试的核苷酸语言模型，包括专门在 ncRNA 序列上训练的 RNA-FM（Chen 等，2022）（图 ??D）。特别是在一项研究中，评估了 SS 核糖体 RNA 突变对大肠杆菌生长速率的影响，Evo 展现出极强的预测性能（Spearman $r = 0.64$，双尾 t 分布检验 $P = 7.3 \times 10^{-4}$）（Zhang 等，2009）。结合我们在蛋白质序列预测中的结果，说明 Evo 能够利用其原核基因组训练数据，预测不同分子模态下的功能属性。

Predicting gene expression from regulatory DNA **从调控性** DNA **预测基因表达**

Given that Evo is also trained on prokaryotic regulatory DNA sequences in addition to sequences that encode proteins or ncRNA, we investigated whether it is able to learn aspects of DNA regulatory grammar. To this end, we leveraged a dataset in which Kosuri et al. (2013) constructed approximately 12k combinations of common promoters and ribosome binding sites (RBSs) and measured the corresponding mRNA and protein expression of a reporter gene for each promoter-RBS pair in E. coli (Figure ??E). We find that the model likelihood scores that Evo assigns to promoter-RBS sequences is significantly correlated with mRNA expression (Spearman $r = 0.41$, two-sided t-distributed $P < 1 \times 10^{-5}$) (Figure ??F) and predictive of binarized protein expression (area under the receiver operating characteristic curve [AUROC] = 0.68, permutation-based $P < 1 \times 10^{-5}$) (Figure ??G) (Methods). Evo's predictive performance is also substantially higher than those of other nucleotide language models, though none of these baseline language models have been trained on datasets containing regulatory elements.
鉴于 Evo 训练数据中除蛋白质和 ncRNA 编码序列外，还包括原核生物调控性 DNA 序列，我们进一步探讨其是否能学习 DNA 的调控语法。为此，我们使用了 Kosuri 等（2013）构建的数据集，该数据集中包含约 12,000 种启动子与核糖体结合位点（RBS）的组合，并测量每对启动子-RBS 对应的报告基因的 mRNA 与蛋白表达水平（图 ??E）。我们发现，Evo 为启动子-RBS 序列赋予的模型似然得分与 mRNA 表达量显著相关（Spearman $r = 0.41$，双尾 t 分布检验 $P < 1 \times 10^{-5}$）（图 ??F），并能预测二值化的蛋白表达情况（受试者工作特征曲线下面积 AUROC

$= 0.68$，基于排列检验的 $P < 1 \times 10^{-5}$）（图 ??G）（见方法）。Evo 的预测性能也远高于其他核苷酸语言模型，尽管这些基准模型均未在包含调控元件的数据集上进行训练。

To predict gene expression from the promoter-RBS sequence alone, Evo most likely uses its knowledge of natural regulatory sequences learned during pretraining, analogous to how protein language models can predict functional changes based on natural variation in protein sequences (Meier et al., 2021; Notin et al., 2023). To this end, as an additional benchmark, we aligned the promoter-RBS pairs to our model's pretraining dataset of prokaryotic sequences. While we hypothesized that the number or the strength of these alignments would be predictive of gene expression, we found that computing these alignments using standard bioinformatic tools resulted in poor or nonexistent sequence matches (Methods). Of all techniques we attempted, none were predictive of gene expression (Figures ??F and ??G), indicating that Evo can distill non-obvious functional information of regulatory DNA directly from large genomic sequence databases.

Evo 能够仅通过启动子-RBS 序列预测基因表达，可能依赖其在预训练过程中对天然调控序列所学到的知识，这类似于蛋白质语言模型根据天然变异预测功能变化的方式（Meier 等，2021；Notin 等，2023）。为此，我们额外将启动子-RBS 配对序列与模型的原核生物预训练数据集进行比对。尽管我们原本假设比对数量或强度可能与基因表达相关，但实际使用标准生物信息学工具计算这些比对时，发现匹配结果较差或几乎不存在（见方法）。我们尝试的所有方法均无法预测基因表达（图 ??F 和 ??G），这说明 Evo 能够直接从大规模基因组序列中提炼出非显性的调控 DNA 功能信息。

Overall, we show how a single foundation model of prokaryotic genomes can perform tasks that have previously been accomplished by different, domain-specific models (protein language models, RNA language models, and regulatory DNA models). Despite being trained on long genomic crops without explicit sequence annotations, Evo still demonstrates an understanding of the constitutive protein-coding sequences, ncRNA sequences, and regulatory elements.

综上所述，我们展示了一个单一的原核基因组基础模型如何完成过去需由不同领域特定模型（蛋白语言模型、RNA 语言模型、调控 DNA 模型）分别完成的任务。尽管 Evo 的训练数据为无显式序列注释的长基因组片段，其仍表现出对蛋白质编码序列、ncRNA 序列与调控元件的理解能力。

Generative design of CRISPR-Cas molecular complexes CRISPR-Cas **分子复合物的生成式设计**

Next, we reasoned that Evo should be able to generate functional complexes that involve interactions between distinct molecular modalities. In prokaryotes, functionally related genes are generally located next to each other on the linear genome sequence. Because Evo learns covariation patterns involving any genetic element within its context window, the model should understand interactions between encoded protein and ncRNA molecules. To demonstrate this capability, we finetuned Evo on a dataset of genomic loci containing CRISPR-Cas sequences: molecular machines that consist of one or more protein components and one or more ncRNA components that, together, direct adaptive immunity against viral infection (Wang et al., 2022).

随后，我们推断 Evo 应该能够生成具有不同分子模态间相互作用的功能性复合物。在原核生物中，功能相关的基因通常在基因组线性序列中彼此邻近。由于 Evo 能够学习上下文窗口中任何遗传元素之间的协变模式，它应该具备理解编码蛋白质与 ncRNA 分子之间相互作用的能力。为展示该能力，我们在一组包含 CRISPR-Cas 序列的基因组区域数据集上对 Evo 进行了微调。CRISPR-Cas 是一种分子机器，包含一个或多个蛋白质组分以及一个或多个 ncRNA 组分，它们协同作用实现针对病毒感染的适应性免疫（Wang 等，2022）。

The DNA-targeting Cas9 nuclease is typically encoded within 3,000 to 4,800 bp of coding sequence and found in close genomic proximity to its cognate CRISPR array (Hsu et al., 2014). Transcription from the CRISPR array generates non-coding CRISPR RNA (crRNA) molecules that are bound by the Cas protein to generate a functional defense complex that is required for sequence-specific DNA-targeting (Figure ??A). For Cas9 in particular, a second trans-activating CRISPR RNA (tracrRNA) forms a duplex with the crRNA to create a full guide RNA (gRNA). Diverse families of CRISPR-Cas systems are found throughout bacterial and archaeal life, such as Cas12- or Cas13-based systems that target DNA and RNA, respectively (Koonin and Makarova, 2019).

靶向 DNA 的 Cas9 核酸酶通常由 3,000 至 4,800 bp 的编码序列编码，并与其对应的 CRISPR 阵列在基因组中邻近存在（Hsu 等，2014）。CRISPR 阵列转录后生成非编码 CRISPR RNA（crRNA），这些 crRNA 被 Cas 蛋白结合，形成功能性防御复合物，实现对 DNA 的序列特异性识别与切割（图 ??A）。对于 Cas9，另一个转激活 CRISPR RNA（tracrRNA）与 crRNA 配对形成双链，构成完整的向导 RNA（gRNA）。在细菌和古菌中广泛存在多种 CRISPR-Cas 系统家族，例如靶向 DNA 的 Cas12 系统或靶向 RNA 的 Cas13 系统（Koonin 与 Makarova，2019）。

In the finetuning step, we trained the model on 82,430 CRISPR-Cas loci extracted from public metagenomic and genomic sequences, adding special prompt tokens for Cas9, Cas12, and Cas13 that were prepended to the beginning of each training sequence (Figure ??B). During sampling, these tokens allow us to guide generation of a specific CRISPR-Cas system type by prompting with the corresponding special token. Strikingly, sampling 8 kb sequences using each of the three Cas token prompts resulted in coherent generations. Depending on the prompt token used, $15 - 45\%$ of generations contained Cas coding sequences as long as 5 kb as detected by Cas subtype profile HMMs (Methods). We also observed that prompting with a specific Cas subtype token typically produced a sample with the expected subtype, demonstrating that Evo can be tuned to generate sequences with both proteins of interest as well as associated non-coding elements such as CRISPR arrays (Figure ??C). Sequence alignment

with the training dataset revealed that Evo is capable of highly unique Cas protein generations, as some of the predicted ORFs exhibited less than 40% protein sequence identity to their respective closest match (Figures ??D and S7).
在微调步骤中，我们使用从公共宏基因组与基因组序列中提取的 82,430 个 CRISPR-Cas 基因组区域对模型进行训练，并为 Cas9、Cas12 和 Cas13 添加了特殊提示词，插入于每条训练序列的开头（图 ??B）。在采样过程中，这些提示词可用于引导模型生成特定类型的 CRISPR-Cas 系统。令人惊讶的是，使用三种提示词分别生成的 8 kb 序列结果都具有连贯性。根据所用提示词不同，有 $15-45\%$ 的生成序列中包含长达 5 kb 的 Cas 编码序列（由 Cas 亚型 HMM 模型检测，见方法）。我们还观察到，使用特定 Cas 亚型提示词通常能生成对应亚型的样本，表明 Evo 可调节生成目标蛋白及相关非编码元素（如 CRISPR 阵列）的能力（图 ??C）。将生成序列与训练数据集比对显示，Evo 可生成高度独特的 Cas 蛋白，其中一些预测的开放阅读框（ORF）与最相似的已知序列的蛋白质相似度低于 40%（图 ??D 和附图 S7）。

To evaluate the quality of Cas generation with Evo, we focused on Cas9 generations and evaluated AlphaFold2 structure predictions of the sampled Cas9 coding sequence and non-coding RNA complexes against experimentally determined structures of the Streptococcus pyogenes Cas9 protein and its tracrRNA:crRNA duplex. Selected structure predictions of sampled Cas9 sequences show that even low-identity generations bear resemblance to natural Cas9 structures in key domains such as the RuvC nuclease and protospacer adjacent motif (PAM)-interacting domains (Figure ??E). Similarly, Evo-generated crRNA:tracrRNA duplexes form predicted RNA secondary structures resembling the canonical crRNA:tracrRNA duplexes found in naturally occurring Cas9 systems (Figure ??F) (Gasiunas et al., 2020).
为评估 Evo 在 Cas 生成上的质量，我们聚焦于 Cas9 生成，并使用 AlphaFold2 对采样得到的 Cas9 编码序列及其非编码 RNA 复合物进行结构预测，与链球菌 Cas9 蛋白及其 tracrRNA:crRNA 双链的实验结构进行对比。部分预测结构表明，即便是低相似度生成的序列，也在关键结构域（如 RuvC 核酸酶域与原间序列邻接基序（PAM）识别域）中与天然 Cas9 结构相似（图 ??E）。同样，Evo 生成的 crRNA:tracrRNA 双链也预测形成类似于天然 Cas9 系统中标准双链结构的 RNA 二级结构（图 ??F）（Gasiunas 等，2020）。

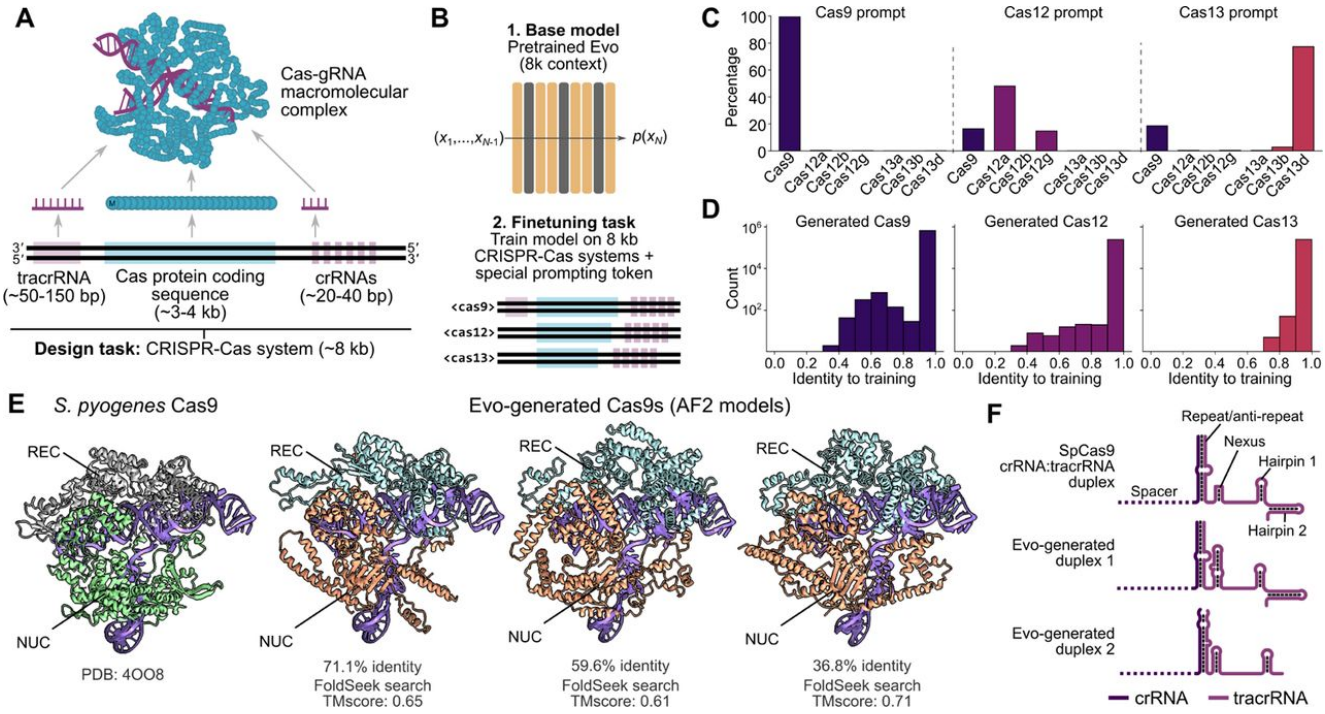

Figure 3: Finetuning on CRISPR-Cas sequences enables generative design of protein-RNA complexes. (A) CRISPR-Cas defense nucleases comprise a large macromolecular complex involving an effector protein bound to a noncoding guide RNA (gRNA) that is derived from a CRISPR RNA (crRNA). For some CRISPR types, a trans-activating CRISPR RNA (tracrRNA) is combined with the crRNA to create the final gRNA. Our design task is to produce sequences that contain these protein and noncoding RNA components. (B) We finetuned Evo, following its initial 8 k pretraining phase, on 8 kb-length genomic sequences containing CRISPR-Cas systems. During finetuning, we prepended a special conditioning token ("cas9", "cas12", or "cas13") to the beginning of each sequence, indicating the general type of Cas protein encoded in the sequence. (C) A prompting token enables controllability over Evo generations. When prompting with the token for a given type of Cas protein, the most common Cas protein found in the resulting generated sequences corresponds to that token prompt (for example, prompting with a "cas9" token typically produces Cas9 sequences). (D) Histograms representing the distribution of percentage identity of a generated Cas protein sequence to any Cas protein sequence in the training dataset. This distribution is computed across sampling runs involving all three prompts. (E) Representative generations of Cas proteins alongside the S. pyogenes Cas9 crystal structure (PDB: 4OO8). AF2: AlphaFold2.
在 CRISPR-Cas 序列上进行微调可实现蛋白质-RNA 复合物的生成式设计。(A) CRISPR-Cas 防御核酸酶是一个大型大分子复合物，包含一个与非编码向导 RNA（gRNA）结合的效应蛋白。该 gRNA 来源于 CRISPR RNA（crRNA），在某些 CRISPR 类型中，crRNA 会与转录激活 CRISPR RNA（tracrRNA）形成最终的 gRNA。我们的设计任务是生成包含这些蛋白质和非编码 RNA 组分的序列。(B) 我们在 Evo 初始 8k 上下文预训练阶段之后，使用长度为 8 kb 的包含 CRISPR-Cas 系统的基因组序列对其进行微调。在微调过程中，我们在每条序列的开头添加一个特殊的条件 token（"cas9"、"cas12" 或"cas13"），以指示序列中编码的 Cas 蛋白类型。(C) 使用提示 token 可以对 Evo 的生成进行可控性引导。当使用特定类型的 Cas 蛋白提示 token 进行提示时，生成的序列中最常见的 Cas 蛋白类型通常与提示 token 相对应（例如，使用"cas9" token 通常会生成 Cas9 蛋白序列）。(D) 直方图表示生成的 Cas 蛋白序列与训练数据集中任一 Cas 蛋白序列之间的百分比相似性分布。该分布是在包含所有三种提示条件的采样运行中计算得到的。(E) 生成的代表性 Cas 蛋白结构与 S. pyogenes Cas9 晶体结构（PDB: 4OO8）进行对比。AF2：AlphaFold2。

When finetuned on CRISPR-Cas systems, Evo can coherently generate diverse samples that resemble naturally occurring Cas systems in both sequence and structure. Designing new Cas systems has historically relied on mining sequence databases for homologous proteins, where natural evolution provides functional diversity. Generative modeling with Evo provides an alternative design methodology that can be harnessed
在 CRISPR-Cas 系统数据上进行微调后，Evo 能够连贯地生成与天然 Cas 系统在序列和结构上均相似的多样化样本。传统上，设计新型 Cas 系统主要依赖于从序列数据库中挖掘同源蛋白，由自然进化提供功能多样性。利用 Evo 进行生成式建模为这一过程提供了一种可用的替代设计方法。

Generative design of transposable biological systems **可转座生物系统的生成式设计**

In addition to molecular complexes, Evo can learn patterns underlying multi-gene systems. An example of minimal replicating systems are mobile genetic elements (MGEs), which are found throughout all domains of life.

Their opportunistic spread provides a fundamental force driving sequence variation, new gene function, and even speciation (Chandler et al., 2020). Insertion sequence (IS) elements are compact MGEs that generally encode only the components that are required for transposition. The IS605 group is widely distributed across prokaryotes and consists of three key components: a TnpA transposase that catalyzes peel-and-paste transposition next to an RNA-guided TnpB nuclease and its cognate ω RNA that bias the selfish inheritance of the transposable element (Figure ??A) (Meers et al., 2023; Karvelis et al., 2021; Altae-Tran et al., 2021). The IS605 group belongs to the greater IS200/IS605 family, which includes IS200 group elements that lack the TnpB endonuclease. Improving our understanding of their biological function and generating new MGEs with desired properties could lead to more effective genome engineering tools.

除了分子复合物，Evo 还能够学习多基因系统中所蕴含的模式。最小的复制系统之一是可动遗传元件（MGEs），它们广泛存在于生命的各个领域。这些元件的机会主义传播是驱动序列变异、新基因功能甚至物种形成的基本力量（Chandler 等，2020）。插入序列（IS）元件是紧凑型 MGE，通常仅编码转座所需的关键组分。IS605 类群在原核生物中广泛分布，由三个主要组分构成：催化" 剥离-粘贴" 式转座的 TnpA 转座酶、RNA 引导的 TnpB 核酸酶及其配对的 ω RNA，后者共同推动该转座元件的" 自私性" 遗传（图 ??A）（Meers 等，2023；Karvelis 等，2021；Altae-Tran 等，2021）。IS605 类属于更大的 IS200/IS605 家族，其中的 IS200 类元件缺少 TnpB 内切酶。深入理解这些元件的生物学功能，并设计具有特定属性的新型 MGE，有望促进更高效的基因组工程工具的发展。

We finetuned Evo on 10,720 IS605 elements and 219,867 IS200 elements in their natural sequence context and used the model to generate novel IS200/IS605 elements (Figure ??B) (Methods). Focusing on generated sequences that contained both a predicted TnpA and TnpB coding sequence, we successfully detected many sequences that encoded proteins that diverged substantially from the training set, with 22.5% of TnpA proteins being $< 50\%$ identical to the training set, and 90.1% of TnpB proteins (Figures ??C and S8). We found that 87.6% of generated TnpA proteins folded well with ESMFold pLDDT > 70, compared to 25.5% of TnpB proteins, which may be due to the greater abundance of TnpA proteins in the training set. Through annotation and inspection of individual examples, we found that some diverse loci encoded coherent transposase and nuclease proteins that folded well using ESMFold, closely matched experimentally determined structures of homologous proteins, and also contained predicted ω RNA sequences (cmsearch E-value $= 5.6 \times 10^{-12}$; Figure ??D).

我们在 10,720 个 IS605 元件和 219,867 个 IS200 元件的自然序列上下文上对 Evo 进行了微调，并利用该模型生成了新型 IS200/IS605 元件（图 ??B）（见方法）。我们聚焦于那些同时包含预测的 TnpA 和 TnpB 编码序列的生成样本，发现其中有大量编码的蛋白与训练集中的序列差异显著：22.5% 的 TnpA 蛋白与训练集中最近匹配的序列相似度低于 50%，而 TnpB 更甚，高达 90.1% 的样本都属于低相似度生成（图 ??C 和附图 S8）。我们发现，87.6% 的 TnpA 生成蛋白在 ESMFold 结构预测中折叠良好（pLDDT > 70），而 TnpB 的成功率为 25.5%，这可能是因为训练集中 TnpA 蛋白的数量更多。通过注释和分析个别样本，我们发现某些多样化的基因位点能够编码结构连贯的转座酶和核酸酶蛋白，它们可通过 ESMFold 良好折叠，且与同源蛋白的实验结构高度一致，同时还包含预测的 ω RNA 序列（cmsearch E-value $= 5.6 \times 10^{-12}$；见图 ??D）。

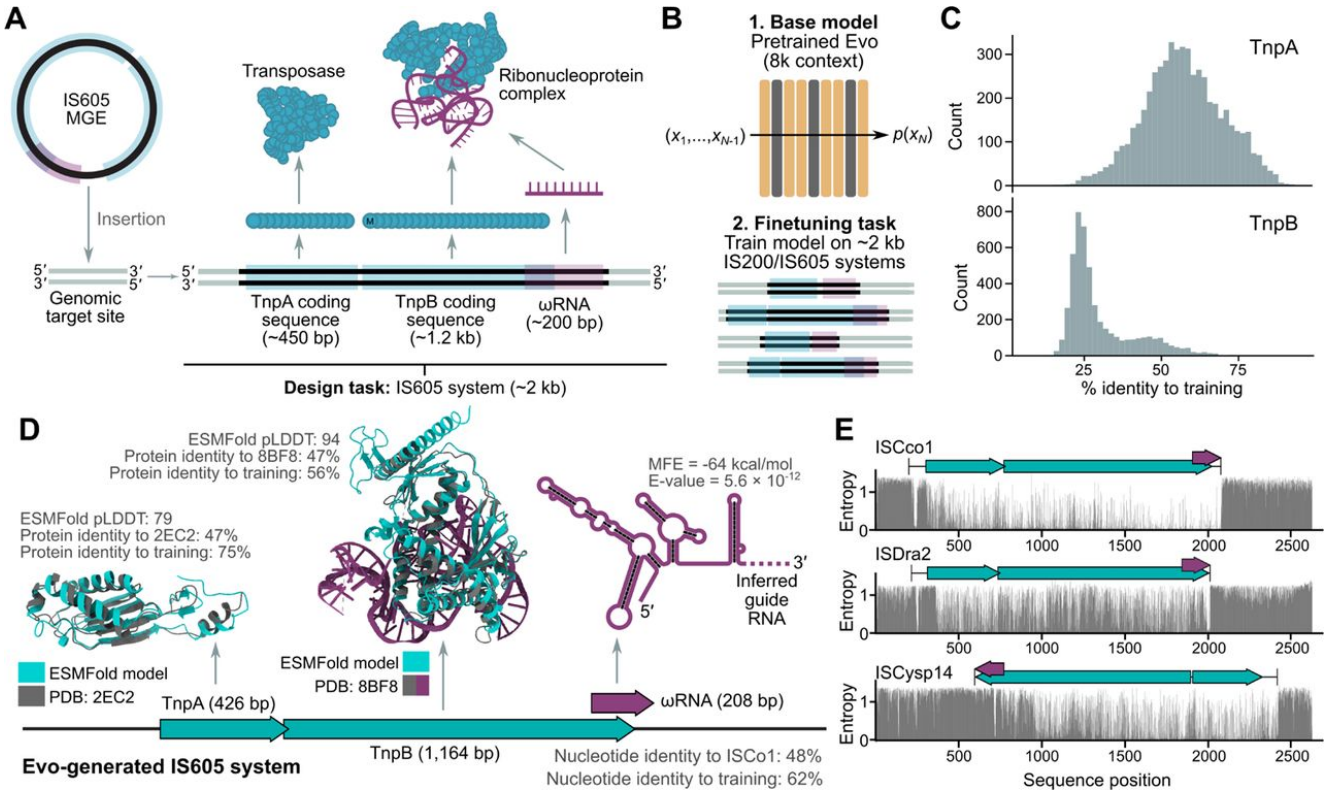

Figure 4: Finetuning on IS200/IS605 sequences enables generative design of transposable biological systems. (A) IS605 group systems are a group of MGEs that belong to the IS200/IS605 family and encode a Y1-HuH transposase encoded by the TnpA coding sequence and a TnpB-ω RNA complex that performs DNA cleavage. Our design task is to produce sequences that contain these protein and ncRNA components. (B) We finetuned Evo, following its initial 8k pretraining phase, on 2 kb-length sequences containing IS200/IS605 systems. (C) Histograms representing the distribution of percentage identity of generated loci that contained a predicted TnpA and TnpB coding sequences. The closest matching member of the training set as identified by MMseqs2 was compared with the generated sequence by MAFFT alignment. (D) Example of a generated IS605 element. Showing TnpA and TnpB protein structures as predicted by ESMFold (blue) aligned to homologous PDB structures (gray) and the secondary structure of a predicted ω RNA. (E) Showing the entropy of the conditional probabilities at each position across three natural IS605 loci. TnpA (short) and TnpB (long) coding sequences are shown in blue, predicted ω RNA boundaries are shown in purple, and the start and end of the complete element is shown with black bars.

MGEs are highly abundant and evolve rapidly, making it difficult to systematically identify the precise boundaries of the elements in their natural sequence context (Durrant et al., 2020). Using the finetuned model, we calculated the entropy of the conditional probabilities at each position across natural IS605 loci (Figures ??E and S8). Although the model was trained without any explicit labeling of MGE boundaries, the entropy signal indicates that the model is learning a representation of these boundaries, with a sharp and sustained increase in entropy corresponding with the 3′ end of the element in particular. Taken together, these results indicate that the finetuned model can generate diverse IS605 systems with coherent protein and RNA sequences, and that the model is learning important features of these elements that could be repurposed for improved functional annotation.
由于 MGEs 数量庞大且进化迅速，在自然序列上下文中系统识别其精确边界极具挑战（Durrant 等，2020）。我们使用微调后的模型计算了天然 IS605 基因位点中每个位点条件概率的熵值（图 ??E 与附图 S8）。尽管模型训练中未使用任何关于 MGE 边界的显式标签，熵信号表明模型正在学习这些边界的内在表示，尤其在转座元件 3′ 末端出现了明显且持续的熵值上升。综上，这些结果说明微调后的模型不仅能生成具有连贯蛋白与 RNA 序列的多样化 IS605 系统，还学习了可用于改进功能注释的重要特征。

Predicting gene essentiality with long genomic context 基于长基因组上下文预测基因必需性

Beyond the molecular or systems level, we designed Evo to be capable of analyzing whole genomes. We conducted a second stage of pretraining using the 8 k-pretrained Evo model as the base model, training it on sequences of 131 k tokens (Figure ??A) with prepended species-level special tokens. This pretraining stage used data from GTDB and a subset of IMG/VR that excludes eukaryotic viruses (Figures ??C and S1). See Methods for additional details related to context extension. Importantly, Evo maintains single-nucleotide resolution at its 131 k context size, which

is important because changes involving small numbers of base pairs can still dramatically affect a whole organism's phenotype. For example, even a single-nucleotide mutation in an essential gene can be incompatible with life if it disrupts that gene's expression or function. Identifying these essential genes is important for understanding the fundamental biology of an organism and for identifying genes in pathogenic organisms that could be the targets of inhibitory drugs (Rocha and Danchin, 2003).
除了分子或系统层面的任务，Evo 还被设计为具备分析整个人类基因组的能力。我们在已经使用 8k 上下文预训练过的 Evo 模型基础上，进行了第二阶段的预训练，将上下文长度扩展至 131k token，并在输入序列前添加了物种级别的特殊提示符（图 ??A）。本阶段训练所使用的数据来自 GTDB 和 IMG/VR 的一个子集，其中不包含感染真核生物的病毒（图 ??C 和附图 S1）。关于上下文扩展的更多细节见" 方法" 部分。重要的是，Evo 在 131k 的上下文规模下仍保持单核苷酸分辨率，这一点至关重要，因为即使是极少数碱基对的改变也可能显著影响整个生物体的表型。例如，若某个关键基因中的单个核苷酸突变破坏了该基因的表达或功能，则该突变可能导致生命不兼容。识别这些必需基因对于理解生物体的基本生物学特性，以及寻找病原体中可作为药物靶点的基因，具有重要意义（Rocha 与 Danchin，2003）。

To this end, we evaluated Evo's ability to predict gene essentiality solely based on mutations to the genome sequence. We conducted an experiment in which we inserted premature stop codons at the beginning of each coding sequence in a given organism's genome and measured the effects of these changes on Evo's likelihood with respect to the likelihood of the wildtype sequence (Figure ??B). When computing the changes to the mutant versus wildtype sequences, we evaluated Evo on the gene sequence alone ("gene only context"), or the gene sequence with flanking context up to a total of 8 k tokens ("8 k context") or 66 k tokens ("66 k context") (Methods). We hypothesized that mutations to essential genes would result in larger, more negative changes in log-likelihood compared to mutations to non-essential genes, allowing us to predict gene essentiality.
为此，我们评估了 Evo 是否能够仅依据基因组序列上的突变预测基因的必需性。我们设计了一个实验，在某一给定生物体的全基因组中为每个编码序列开头插入提前终止密码子，并比较突变序列与对应野生型序列在 Evo 预测中的对数似然变化（图 ??B）。我们在三种上下文设定下评估 Evo 的预测表现：仅输入目标基因序列（" 仅基因上下文"）、包含上下游序列组成总长度为 8k token 的上下文（"8k 上下文"）、以及 66k 上下文（"66k 上下文"）（见方法）。我们的假设是，对必需基因的突变将导致更大、方向更负的对数似然变化，从而使得模型能够区分必需基因与非必需基因。

On a dataset of 56 whole-genome essentiality studies in bacteria from the DEG database (Zhang, 2004) and two whole-genome essentiality studies in phage from Piya et al. (2023), we observed that the changes in Evo log-likelihood with 66 k context are significantly predictive (Bonferroni-corrected permutation-based $P < 0.05$) of gene essentiality in 43 out of 58 genomes. We also observed that providing the model with additional genomic context beyond the gene sequence results in a substantial improvement in performance, especially from gene only context to 8 k context. From 8 k to 66 k context, the average predictive performance is essentially equivalent, but the range does increase due to improvement in outlier examples (Figures ??C and S9A). With 8 k context, the model most likely has access to enough of the genome to improve its prediction of mutational effects on organism function, whereas 66 k context provides new, helpful information in only some cases. For a few genomes, the zero-shot performance with 66 k context is notably strong, with an AUROC of 0.86 on lambda phage essentiality data (Piya et al., 2023) and an AUROC of 0.81 on Pseudomonas aeruginosa essentiality data (Turner et al., 2015) (Figure ??D).
在来自 DEG 数据库的 56 个细菌全基因组必需性研究以及 Piya 等（2023）提供的 2 个噬菌体全基因组研究中，我们发现，在 66k 上下文下，Evo 预测的对数似然变化在 58 个基因组中有 43 个显著预测了基因必需性（Bonferroni 校正后的排列检验 $P < 0.05$）。此外，我们还观察到，在输入超出目标基因本身的上下文信息后，模型性能有显著提升，尤其是从" 仅基因" 到"8k 上下文" 的跃升最为显著。从 8k 到 66k 上下文，平均预测性能变化不大，但由于部分极端样本表现改善，总体分布范围有所扩展（图 ??C 和附图 S9A）。8k 上下文可能已经为模型提供了足够的基因组信息以优化突变效应预测，而 66k 上下文则仅在某些特定情况下额外提供有价值的信息。在个别基因组中，66k 上下文下的零样本预测表现尤为强劲，例如在 噬菌体数据中 AUROC 达 0.86（Piya 等，2023），在铜绿假单胞菌数据中 AUROC 达 0.81（Turner 等，2015）（图 ??D）。

Evo is also able to predict essentiality when using different in-silico mutagenesis strategies, such as varying the number of stop codons inserted, or deleting the gene sequence entirely (Figure S9B; Methods), though we did not attempt an exhaustive search of the best prompting strategy for this task. GenSLM, a codon language model that had mild predictive performance of mutational effects on single-gene protein function (Figure ??B), could not perform this zero-shot prediction task (Figure ??C). We also observed that a gene's position in the genome is not predictive of essentiality, indicating that trivial positional biases do not contribute to prediction performance (Figure S9B). Together, these results demonstrate that Evo can predict mutational effects at a whole-organism level across many bacterial and phage species, without any explicit genome annotations, task-specific training data, or functional labels. In contrast to protein or codon language models, Evo enables an understanding of gene function within a broader genomic context.
Evo 也能适应不同的计算突变策略来预测基因必需性，例如改变插入终止密码子的数量，或直接删除整个基因序列（附图 S9B；见方法），尽管我们并未对该任务进行最佳提示策略的穷尽性搜索。与此形成对比的是，GenSLM（一个以密码子为单位的语言模型），虽然在预测单基因蛋白功能突变方面表现一般（图 ??B），却无法完成此类零样本预测任务（图 ??C）。我们还发现，基因在染色体上的位置并不能预测其必需性，表明预测性能并未受到显而易见的位置偏倚影响（图 S9B）。综上结果表明，Evo 能够跨多个细菌与噬菌体物种，在无需任何显式基因组注释、

任务特定训练数据或功能标签的前提下，预测突变对整个生物体的影响。与蛋白或密码子语言模型不同，Evo 能够在更广泛的基因组上下文中理解基因功能。

Generating DNA sequences at genome scale **生成基因组尺度的** DNA **序列**

Given Evo's generative capabilities, we were interested in testing its generation quality at long sequence lengths without additional finetuning. By doing so, we can better understand the patterns and the level of detail learned by the model, which helps us determine the model's capabilities and limitations. We used Evo to sample twenty sequences each containing $\sim$ 650 kb, representing about five times the model's context length of 131 kb. For comparison, the smallest "minimal" bacterial genomes are about 580 kb in length (Blanchard and Bébéar, 2011). We prompted the model to generate bacterial genomes using the species-level tokens in the training dataset (Figure ??A). To analyze how well these generations recapitulate natural genomes, we used CheckM (Parks et al., 2015), a tool originally developed to assess the quality of bacterial DNA sequenced from nature. CheckM calculates statistics such as the density of coding sequences in the genome and the presence of key marker genes that are found in nearly all prokaryotes, which we used to determine how well our generated sequences mirror key characteristics of natural genomes.

鉴于 Evo 的生成能力，我们希望在不进行额外微调的前提下，测试其在长序列长度下的生成质量。通过这样做，我们可以更好地理解模型所学习到的模式及其细节水平，从而明确其能力与局限性。我们使用 Evo 采样生成了 20 条序列，每条长度约为 $\sim$ 650 kb，相当于模型 131 kb 上下文长度的 5 倍。作为对比，最小的" 极简" 细菌基因组长度约为 580 kb（Blanchard 和 Bébéar，2011）。我们通过训练集中使用的物种级别提示词引导模型生成细菌基因组（图 ??A）。为分析生成序列与天然基因组的相似程度，我们使用了 CheckM 工具（Parks 等，2015），该工具最初用于评估从自然样本中测得的细菌 DNA 的质量。CheckM 会计算如编码序列密度及常见关键标记基因等统计特征，以判断所生成序列是否复现了天然基因组的核心特性。

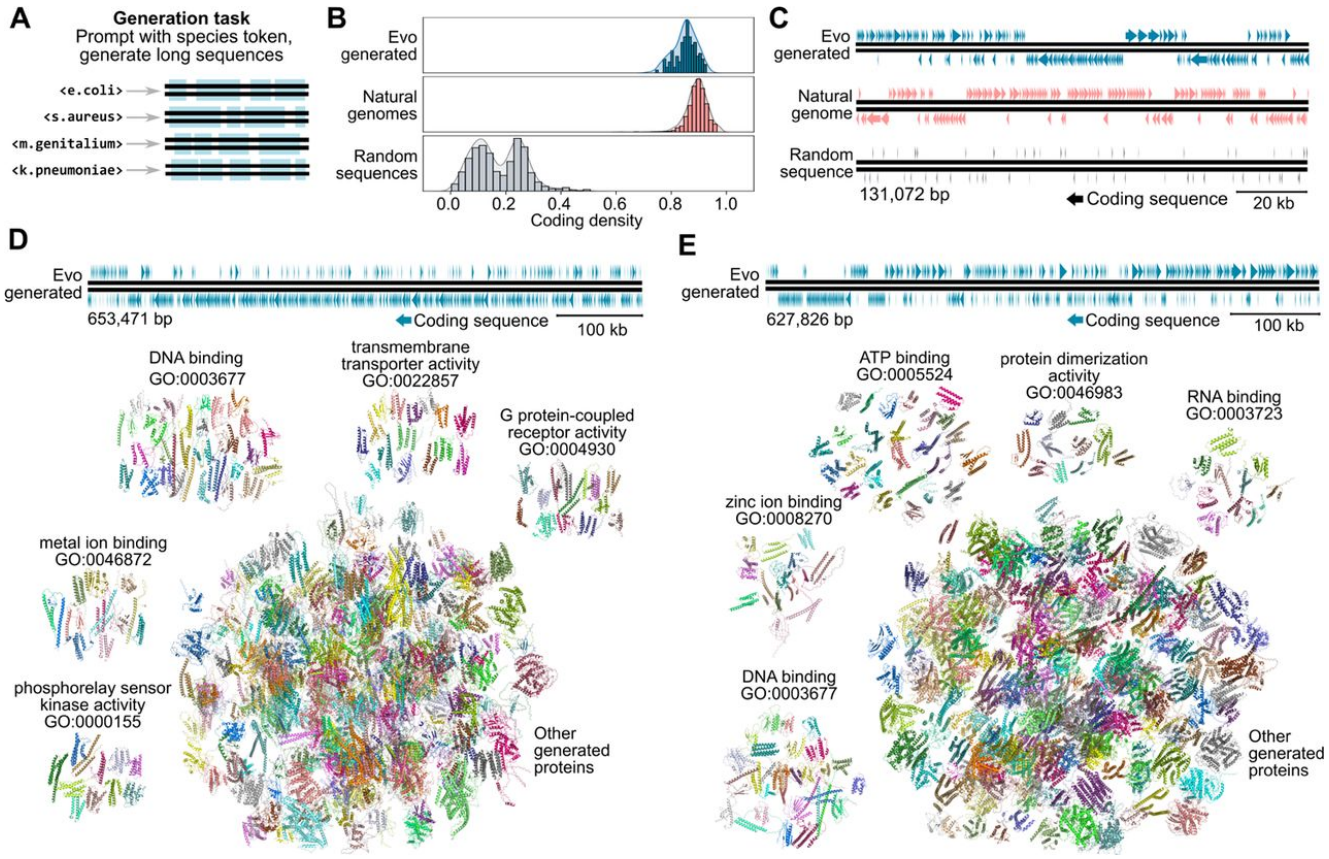

Figure 5: Evo generates genome-scale sequences with dense coding architecture. (A) We prompted Evo with species-level tokens used during the second pretraining stage. We use bacterial species prompts and generate sequences of 650 kb in length.
(A) 我们使用第二阶段预训练中引入的物种级别提示词对 Evo 进行提示，利用细菌物种提示生成长度约为 650 kb 的序列。
(B) Histograms depicting the distribution of coding density scores among 131 kb crops of sequences generated by Evo ("Evo generated"), sequences from natural bacteria ("natural genomes"), or sequences in which the four base pairs were sampled uniformly at random ("random sequences").
(B) 柱状图展示了 131 kb 片段中编码密度得分的分布，分别来自 Evo 生成的序列（"Evo generated"）、天然细菌基因组序列（"natural genomes"）以及四种碱基均匀随机采样的序列（"random sequences"）。
(C) Arrow plots depicting the organization of coding sequences on an example 131 kb sequence generated by Evo, derived from a natural genome, or sampled randomly. Coding sequences are depicted as arrows in which the horizontal length of the arrow corresponds to the genomic interval and the direction of the arrow indicates the strand. The top and bottom rows of arrows indicate the 5 -to-3 and 3 -to-5 strands, respectively, and the Evo-generated sequence was designated as the 5 -to-3 strand. Both Evo-generated and natural genomes exhibit operon-like structure in which clusters of co-located genes are on the same strand.
(C) 箭头图展示了编码序列在一个 131 kb 序列中的分布，该序列分别由 Evo 生成、来自天然基因组，或由随机采样产生。编码序列以箭头表示，箭头的水平长度表示基因组片段长度，箭头方向表示所在链的方向。上方和下方的箭头分别表示 5 -to-3 和 3 -to-5 链，Evo 生成的序列被指定为 5 -to-3 链。Evo 生成的序列与天然基因组均显示出操纵子样结构，其中位于相近位置的基因倾向于位于同一链上。
(D, E) Example generated sequences are represented as arrow plots as in (C).
(D, E) 示例生成序列以与 (C) 相同的箭头图形式展示。

Notably, the coding density of sequences generated by Evo is nearly as high on average as the density of coding sequences found in natural genomes, and is substantially higher than the coding density of random sequences (Figure ??B). Importantly, when visualized, both natural and generated sequences display similar patterns of coding organization (Figure ??C), with sequences in close proximity typically found with the same strand orientation. In bacteria, these closely linked groups of coding sequences typically correspond to functionally tied gene clusters or operons. When using ESMFold to obtain protein structure predictions corresponding to these coding sequences, almost all showed some predicted secondary structure and globular folds (Figures ??D, ??E, and S10). Some proteins also showed structural similarity to natural proteins involved in known molecular functions as annotated by Gene Ontology (GO) (Figures ??D and ??E). However, many of these structure predictions are of low confidence and have limited structural matches to any entry in a representative database of naturally occurring proteins (Figure

S10). The generated sequences also do not contain many highly conserved marker genes that typically indicate complete genomes. Across all of our generated sequences representing ∼ 13 megabases, Evo sampled 18 tRNA sequences (compared to 35 tRNAs in the ∼ 580 kb genome of M. genitalium) and no rRNAs, as detected by the programs tRNAscan-SE (Chan and Lowe, 2019) and barrnap (Seemann, 2018), respectively.
值得注意的是，Evo 所生成序列的编码密度平均接近于天然基因组中的编码序列密度，且明显高于随机序列的编码密度（图 ??B）。更重要的是，当进行可视化时，天然与生成序列都展现出相似的编码组织模式（图 ??C），邻近的序列通常具有相同的链方向性。在细菌中，这类紧密链接的编码序列通常对应功能相关的基因簇或操纵子。利用 ESMFold 对这些编码序列对应的蛋白结构进行预测时，几乎所有蛋白都显示出某种形式的二级结构及球状折叠（图 ??D、??E 及附图 S10）。部分蛋白结构也显示出与已知分子功能相关的天然蛋白具有结构相似性（根据 Gene Ontology 进行注释）（图 ??D 和 ??E）。然而，许多结构预测结果的置信度较低，且与天然蛋白代表数据库中的条目匹配度有限（图 S10）。此外，所生成的序列中也缺乏许多高度保守的标记基因，这些基因通常表征完整基因组。在总计约 ∼ 13 Mb 的生成序列中，Evo 共生成了 18 条 tRNA 序列（相比之下，M. genitalium 的 ∼ 580 kb 基因组中含有 35 条 tRNA）且未检测到任何 rRNA，分别通过 tRNAscan-SE（Chan 与 Lowe，2019）和 barrnap（Seemann，2018）识别得出。

These results suggest that Evo can generate genome sequences containing plausible high-level genomic organization at an unprecedented scale without extensive prompt engineering or finetuning. These samples represent a "blurry image" of a genome that contains key characteristics but lacks the finer-grained details typical of natural genomes. This is consistent with findings involving generative models in other domains, such as natural language or image generation. For example, directly sampling from a large natural language model typically produces sequences that are grammatically correct yet locally biased toward simpler sentence constructions and that are globally incoherent, especially at long lengths. Promisingly, in these domains, algorithmic techniques have emerged to improve the quality of generations compared to sampling from the pretrained model alone (Wei et al., 2022; Ouyang et al., 2022; Rafailov et al., 2024). The baseline generation quality observed for pretrained Evo suggests that Evo is also amenable to these techniques.
上述结果表明，Evo 能够在无需大量提示词工程或微调的情况下，生成具备可信高阶基因组结构特征的基因组级序列，达到了前所未有的生成规模。这些样本可被视为基因组的一种" 模糊图像"，其包含了关键特征，但缺乏天然基因组中常见的细粒度细节。这一现象与其它领域的生成模型研究结果一致，如自然语言或图像生成。例如，从大型语言模型中直接采样通常会生成语法正确但局部结构偏向简单句型、全局逻辑不连贯的文本，特别是在序列较长时尤为明显。值得期待的是，在这些领域中，已经发展出多种算法技术来提高生成质量，相比仅依赖预训练模型的直接采样效果更佳（Wei 等，2022；Ouyang 等，2022；Rafailov 等，2024）。我们观察到的 Evo 预训练模型的基础生成质量表明，它也同样适用于这些优化方法。

Discussion **讨论**

Evo is a genomic foundation model trained on hundreds of billions of DNA tokens across the evolutionary diversity of prokaryotic life, capable of prediction and generation tasks at the scale of individual molecules, molecular complexes, biological systems, and even whole genomes. Based on a state-of-the art hybrid model architecture, StripedHyena, Evo enables single-nucleotide resolution language modeling at a context length of 131 k. We conducted the first scaling laws analysis of DNA pretraining across several architectures, where we observed StripedHyena outperforming several baseline architectures, including the Transformer architecture, at each level of scale. Evo accurately performed zero-shot prediction across diverse fitness or expression prediction tasks on proteins, ncRNAs, or regulatory DNA that matches or outperforms specialized models, while also understanding which genes are essential to organismal fitness. Evo is also a generative model, which we leverage to sample CRISPR-Cas proteins and their noncoding guide RNAs, multi-gene transposable systems, and ∼ 650 kb sequences that recapitulate the coding organization of real genomes. We make open-source code and models for Evo publicly available at https://github.com/evo-design/evo.
Evo 是一个基因组基础模型，在涵盖原核生物进化多样性的数千亿 DNA token 上进行了训练，具备从单个分子、分子复合物、生物系统直至整个人类基因组的预测与生成能力。Evo 基于最先进的混合架构 StripedHyena，能够以 131k 上下文长度进行单核苷酸分辨率的语言建模。我们首次对多种架构进行了 DNA 预训练的缩放规律分析，结果显示 StripedHyena 在各个规模下均优于包括 Transformer 在内的多个基线模型。Evo 能够在蛋白质、ncRNA 及调控 DNA 上执行多种适应度或表达预测任务的零样本推理，其表现与专业模型持平或更优，并能识别对生物体适应性至关重要的基因。作为生成模型，Evo 还可用于采样 CRISPR-Cas 蛋白及其非编码向导 RNA、多基因转座系统、以及 ∼ 650 kb 的序列，后者重现了真实基因组的编码组织。我们已将 Evo 的开源代码与模型发布于 https://github.com/evo-design/evo。

A model capable of genome-scale design holds great potential to advance therapeutic discovery, sustainability, and our understanding of fundamental biology, but simultaneously raises biosafety and ethical considerations. The Global Alliance for Genomics and Health (GA4GH) (Rehm et al., 2021) has developed principles for the oversight of genetic engineering technologies and could provide a robust foundation for transparency, accountability, and shared responsibility. Such a framework is essential to foster international cooperation that benefits all humanity. A proactive discussion involving the scientific community, security experts, and policymakers is imperative to prevent misuse and to promote effective strategies for mitigating existing and emerging threats. Furthermore, investment

in education and capacity building, especially in under-resourced communities, will sustainably democratize access and use of tools like Evo. We open source the model to promote transparency and begin a dialogue with the broader scientific community and shareholders. We also apply the precaution of excluding eukaryotic viruses from our pretraining dataset. We include an extended discussion on ethical considerations in a supplementary Safety and ethics discussion in which we assessed the risks of tools like Evo, including their potential to be misused, contribute to social and health inequity, or disrupt the natural environment. Clear, comprehensive guidelines that delineate ethical practices for the field are required for the responsible development and use of genome-scale language models.
一个具备基因组尺度设计能力的模型具有巨大的潜力，可推动治疗发现、可持续发展以及对基础生物学的理解，但同时也带来了生物安全与伦理方面的考量。全球基因组与健康联盟（GA4GH）（Rehm 等，2021）已制定了一套用于监管基因工程技术的原则，有望为实现透明、问责与共享责任提供坚实基础。这一框架对促进造福全人类的国际合作至关重要。科学界、安全专家与政策制定者之间必须展开积极对话，以防止模型被滥用，并制定有效策略应对现有及新兴风险。此外，对教育与能力建设的投入，特别是在资源匮乏地区，有助于可持续地普及像 Evo 这样的工具的获取与应用。我们开源该模型，以促进透明化，并与更广泛的科研群体及利益相关者展开对话。我们还采取了预防措施，在预训练数据中排除了真核病毒。我们在补充材料中的" 安全与伦理讨论" 部分详细阐述了对类似 Evo 工具的风险评估，包括其被滥用、加剧社会与健康不平等、或破坏自然环境的可能性。制定明确、全面的伦理实践指南对于基因组级语言模型的负责任开发与应用至关重要。

Despite the remarkable capabilities of this first-generation DNA foundation model, a number of technical limitations and challenges remain. We pretrained Evo on a dataset of 300B prokaryotic tokens which represents a miniscule portion of petabytes of publicly available genomic data. Because our model is trained only on prokaryotic data, our ability to predict functional effects of mutations on human protein fitness is limited. Natural language models often struggle to maintain coherent and diverse generation over long sequences, and Evo can demonstrate similar properties. For example, we observed that more novel CRISPR-Cas sequences were sampled at relatively low frequency and prompting on special tokens had moderate controllability, occasionally generating a Cas9 protein when prompted with a Cas12 token. At the genome-scale, Evo generates hundreds of kilobases that demonstrate a high-level understanding of genome organization, but struggles to include key marker genes such as full tRNA-encoding repertoires. These limitations mirror the constraints of natural language models, which have been improved over time with increased scale, labeled data, prompt engineering, and alignment with human preferences (Kaplan et al., 2020; Ouyang et al., 2022; Wei et al., 2022; Kojima et al., 2022; Rafailov et al., 2024). We expect a similar trajectory for models of DNA.
尽管这一代 DNA 基础模型展现出了卓越能力，但仍存在诸多技术限制与挑战。Evo 预训练使用的 3000 亿个原核 token，仅占公开可用基因组数据（以 PB 计）的极小一部分。由于模型仅使用原核数据进行训练，其预测人类蛋白突变功能效应的能力有限。与自然语言模型一样，Evo 在长序列上也面临生成连贯性与多样性下降的问题。例如，我们观察到新型 CRISPR-Cas 序列的采样频率较低，且对特殊提示词的可控性中等——例如在以 Cas12 为提示词时偶尔生成 Cas9 蛋白。在基因组尺度下，Evo 能够生成数十万碱基对的序列并展现出对高阶基因组结构的理解，但在生成诸如完整 tRNA 编码组这类关键标记基因时存在困难。这些限制反映了自然语言模型的通病，而这些问题在语言建模领域已通过扩大模型规模、引入标注数据、提示工程及对齐人类偏好等手段得到逐步改善（Kaplan 等，2020；Ouyang 等，2022；Wei 等，2022；Kojima 等，2022；Rafailov 等，2024）。我们预期 DNA 模型将遵循类似的发展路径。

DNA modeling at this scale and resolution lays the groundwork for a host of research directions. We expect that Evo will benefit from additional scale, longer context length, and more diverse pretraining data. Given the success of language-model-guided directed evolution of proteins (Hie et al., 2024), genomic language models may also help guide the directed evolution of multi-gene biological systems. Similarly, the co-evolutionary information contained in these models could improve molecular structure prediction in a multi-gene context (Jumper et al., 2021; Lin et al., 2023). Properties of systems biology may emerge as these models improve, such as fitness effects of combinatorial gene interactions or the prediction of functional operon linkages. With better conditioning or prompt engineering, Evo could form the basis of a next-generation sequence search algorithm by enabling metagenomic mining at a relational or a semantic level rather than extracting literal sequences from existing organisms. Beyond prokaryotes, the incorporation of eukaryotic genomes into Evo will need to consider the far higher complexity of these genomes and require substantial resource investment in engineering, compute, and safety-related model alignment. Combined with advances in large-scale genome modification (Durrant et al., 2024), Evo helps expand the scope of biological engineering and design to the scale of whole genomes.
如此规模与分辨率的 DNA 建模为众多研究方向奠定了基础。我们预期，Evo 将从更大规模、更长上下文长度以及更丰富的预训练数据中获益。鉴于语言模型引导的蛋白质定向进化已获得成功（Hie 等，2024），基因组语言模型或许也可用于指导多基因生物系统的定向进化。同样，这些模型中蕴含的协同进化信息也可望改善在多基因背景下的分子结构预测（Jumper 等，2021；Lin 等，2023）。随着模型性能的提升，系统生物学属性也可能逐步显现，如组合基因相互作用对适应度的影响，或功能性操纵子联结的预测。借助更强的条件建模能力或提示工程，Evo 有望成为下一代序列搜索算法的基础，不再仅依赖于提取现存生物中的字面序列，而是实现关系层级或语义层级的宏基因组挖掘。对于超出原核生物范畴的应用，将真核基因组纳入 Evo 将面临其更高复杂性所带来的挑战，并需要在工程、算力与安全对齐等方面投入大量资源。结合大规模基因组改造技术的进展（Durrant 等，2024），Evo 有助于将生物工程与设计的范围拓展至完整基因组的层面。

Code and data availability

Code and models related to this study are publicly available at https://github.com/evo-design/evo. We used the following datasets for pretraining:

- Bacterial and archaeal genomes from the Genome Taxonomy Database (GTDB) v214.1 (Parks et al., 2015).
- Curated prokaryotic viruses from the IMG/VR v4 database (Camargo et al., 2023).
- Plasmid sequences from the IMG/PR database (Camargo et al., 2024).

In addition to the above datasets, we also used portions of the following datasets for finetuning:

- NCBI RefSeq (O'Leary et al., 2016).
- UHGG (Almeida et al., 2021).
- JGI IMG (Chen et al., 2021).
- The Gut Phage Database (Camarillo-Guerrero et al., 2021).
- The Human Gastrointestinal Bacteria Genome Collection (Forster et al., 2019).
- MGnify (Mitchell et al., 2020).
- Youngblut et al. (2020) animal gut metagenomes.
- MGRAST (Meyer et al., 2008).
- Tara Oceans samples (Sunagawa et al., 2015).

Additional details on these datasets are provided in Methods.

Acknowledgements

We thank Elijah Chanakira, Dave Driggers, Richard Dugan, Helmut Fritz, Marco Iskender, Adeesh Jain, Mike LaPan, Sean Marrs, Sigalit Perelson, Randy Rizun, Jason Rojas, and Delaney Ugelstad for assistance with computational infrastructure. We thank Samuel Sternberg and Chance Meers for providing covariance models to identify diverse ω RNAs. We thank Jessica Adkins, Joana Carvalho, Dan Fu, Jared Dunnmon, Yunha Hwang, Julia Kazaks, Gautam Machiraju, April Pawluk, Christina Theodoris, Ben Viggiano, and Alden Woodrow for helpful discussions and assistance with manuscript preparation. P.D.H. acknowledges funding support from Arc Institute, Rainwater Foundation, Curci Foundation, Rose Hill Innovators Program, V. and N. Khosla, S. Altman, and anonymous gifts to the Hsu Lab. B.L.H. acknowledges funding support from Arc Institute, Varun Gupta, and Raymond Tonsing.

Author Contributions

E.N., P.D.H., and B.L.H conceived the project. P.D.H. and B.L.H. supervised the project. E.N., M.P., and A.W.T. designed the model architecture. M.G.D. and B.L.H. curated and processed the pretraining and finetuning datasets. M.P. implemented the optimized training and generation infrastructure. E.N., A.W.T., and B.L.H. contributed to the optimized training and generation infrastructure. E.N., M.P., and A.W.T. implemented and carried out the scaling laws analysis. E.N., M.P., A.W.T., and B.L.H. evaluated the pretrained model. E.N. and B.L.H. conducted model finetuning. B.K. and P.D.H. sampled or analyzed CRISPR-Cas generations. M.G.D. and B.L.H. sampled or analyzed the IS200/IS605 generations. B.L.H. conducted the gene essentiality analysis. M.P. and B.L.H. conducted genome-scale sampling and analysis. M.P., A.W.T., and B.L.H. implemented the public Evo codebase. M.Y.N., A.L., and T.H-B. conducted the ethics and safety investigation and discussion. E.N., M.P., M.G.D., P.D.H., and B.L.H wrote the first draft of the manuscript. All authors wrote the final draft of the manuscript.

Competing Interests

M.P. is an employee of TogetherAI. M.G.D. acknowledges outside interest in Stylus Medicine. C.R. acknowledges outside interest in Factory and Google Ventures. P.D.H. acknowledges outside interest in Stylus Medicine, Spotlight Therapeutics, Circle Labs, Arbor Biosciences, Varda Space, Vial Health, and Veda Bio, where he holds various roles including as co-founder, director, scientific advisory board member, or consultant. B.L.H acknowledges outside interest in Prox Biosciences as a scientific co-founder. All other authors declare no competing interests.

Supplementary Materials

A. Safety and ethics discussion **安全性与伦理讨论**

The introduction of powerful generative genomic foundation models such as Evo enables the rapid deciphering of complex genetic information, which can be used for genetic engineering and therapeutic development. Evo is the first of its kind to predict and generate DNA sequences at the whole-genome scale with single-nucleotide resolution, albeit only for prokaryotes in this version. As future capability increases are likely achievable with the class of large-scale DNA models enabled by Evo, we provide an extended ethical discussion on potential risks and precautionary measures. While Evo is limited in its current form, the molecular design, synthesis, manipulation, and dissemination of new synthetic genetic materials could pose concerns to individuals, society, and the environment. Through a responsible AI lens (Badal et al., 2023), we forecast three salient ethical implications and identify mediating solutions.
强大的生成式基因组基础模型（如 Evo）的引入，使得复杂遗传信息的快速解码成为可能，可广泛用于基因工程与治疗开发。Evo 是首个能够以单核苷酸分辨率在整个基因组尺度上进行 DNA 序列预测与生成的模型，尽管当前版本仅限于原核生物。考虑到 Evo 所代表的大规模 DNA 模型体系在未来可能实现能力跃升，我们在此提供对潜在风险及预防措施的扩展伦理讨论。尽管当前形态下的 Evo 仍具有限性，然而围绕新型合成遗传材料的分子设计、合成、操控及传播，仍可能对个人、社会乃至环境造成影响。从" 负责任 AI" 视角出发（Badal 等，2023），我们预测该技术涉及三大显著伦理影响，并提出相应的缓解措施。

A.1. Safety and ethical implications **安全与伦理影响**

A.1.1. Whole-genome foundation models have the potential for misuse **全基因组基础模型存在被滥用的风险**

There are concerns that the dual-use (or misuse) of genomic foundation models by malevolent actors could pose a threat to biosafety and biosecurity (Baker and Church, 2024). Tools like Evo serve to enhance queries of the existing genomic knowledge base and identify genetic regions of interest for editing or experimentation. The ability to discern fitness associated with certain sequences can assist in the discovery of novel biomarkers or therapeutic targets, but can also catalyze the development of harmful synthetic microorganisms that more easily bypass the body's natural defenses, are resistant to current treatments, or cause more severe disease. Fortunately, even with optimal synthetic genomic designs, the ability to create viable organisms is limited by high barriers to entry, including a substantial amount of technical resources and expertise needed to carry out genome synthesis and expression, which is further compounded by the unpredictability of biological mechanisms. Nevertheless, as genetic engineering tools become more readily available, guardrails (for example, access controls, usage audits) should be agreed upon by shareholders to limit unfettered queries for harmful genetic sequences. Clear definitions of what constitutes "dual-misuse" are also needed to draw the line for researchers, policy makers, and other shareholders.
人们担忧，恶意行为者对基因组基础模型的" 双重用途"（或滥用）可能对生物安全与生物安保构成威胁（Baker 与 Church，2024）。像 Evo 这样的工具有助于深入查询现有的基因组知识库，识别可供编辑或实验的遗传区域。识别与适应性相关的序列的能力可以加速新型生物标志物或治疗靶点的发现，但同样可能助长有害合成微生物的开发，例如更易规避人体天然免疫、更具耐药性或致病性更强的微生物。幸运的是，即使在最优设计条件下，创建可存活有机体的能力仍受限于较高门槛，包括完成基因组合成与表达所需的大量技术资源与专业知识，同时也受到生物机制不确定性的影响。尽管如此，随着基因工程工具的普及，相关利益方应就访问控制、使用审计等防护机制达成共识，以限制对有害基因序列的无限制查询。同时，也需要清晰界定" 恶意双重用途" 的范畴，为研究者、政策制定者及其他利益相关方提供明确的指导界限。

A.1.2. Whole-genome foundation models could contribute to social and health inequity **全基因组基础模型可能加剧社会与健康不平等**

Given the high barriers to entry, access and capability inequality with tools such as Evo can lead to inadvertent societal harms. Evo is open source to promote transparency and reproducible research. However, those who can most effectively use, and hence benefit the most from, the tool are entities with coordinated biotechnical resources and expertise, such as biotechnology and pharmaceutical corporations. These companies may accelerate research in a direction that prioritizes returns-on-investment over the global disease burden or health equity (Morin et al., 2023). Along the same line, wealthier nations or more well-funded institutions also stand to better leverage Evo to accelerate their research agendas, further widening the gap between high- and low-resource settings.
由于进入门槛较高，Evo 等工具在获取与能力上的不均衡可能在无意中带来社会性伤害。Evo 被开源以促进透明化与研究可复现性，然而最能有效使用并最大受益的仍是具备整合型生物技术资源与专业能力的机构，例如生物技术或制药公司。这些企业可能将研究优先方向偏向投资回报最大化，而非全球疾病负担或健康公平（Morin 等，2023）。类似地，经济发达国家或资金充足的研究机构也更能利用 Evo 推进其研究议程，从而进一步加剧高低资源环境间的差距。

The use of generative tools in biology also raises complex intellectual property concerns. Biological foundation models such as Evo may enable an organization to bypass current intellectual property limitations on biological

therapeutics or other materials. In some cases, this may lead to a monopolization of treatments for certain conditions. Such an entity could then use these rights to set prohibitively high prices and make treatments inaccessible to most patients (for example, those in low-income countries), thus further exacerbating health disparities (Peek, 2021). In other cases, bypassing intellectual property protections could discourage further investment into therapeutic innovation. Overall, we argue that an entity that uses and benefits from open-source tools such as Evo has a duty to return value to the public and contribute to social and health equity. Intellectual property law should also evolve as generative models increasingly automate the biological discovery and design process.
生成式工具在生物学中的应用也带来了复杂的知识产权问题。Evo 等生物基础模型可能使某些组织绕过现有生物治疗产品或其他材料的知识产权限制。在某些情况下，这可能导致对特定疾病治疗手段的垄断，从而通过定价权利将治疗费用设定得高不可及，使大多数患者（尤其是低收入国家的患者）无法获得治疗，进一步加剧健康不平等（Peek，2021）。在其他情况下，绕过知识产权保护也可能抑制对治疗创新的进一步投资。总体而言，我们认为，凡是使用并受益于 Evo 等开源工具的实体，都有责任回馈公众并推动社会与健康公平。同时，随着生成模型越来越多地自动化生物发现与设计流程，知识产权法律也应相应演进。

A.1.3. Whole-genome foundation models could contribute to disruptions to the natural environment **全基因组基础模型可能对自然环境造成干扰**

Although Evo does not directly manipulate any genetic material, it may enhance the efficiency of genetic engineering projects. There are concerns with how the capabilities of genetic engineering technologies may disrupt the environment and cause ecological uncertainty (for example, the release of altered organisms), leading to a loss of biodiversity or the emergence of new, potentially harmful species (Macfarlane et al., 2022). Although the ecological impacts of training whole-genome foundation models remain unknown, more immediately, it is also important to consider the carbon footprint associated with increasing infrastructure and computational demands (Nature Computational Science, 2023). The capabilities of tools such as Evo, alongside other technologies for genome editing and ecological engineering, add to complex debates about the extent to which science should intervene in evolution. As we push the boundaries of scientific capabilities with tools such as Evo, it becomes imperative to reflect on the interactions and boundaries between our inventions and natural evolutionary processes, aiming to preserve ecological balance, maintain environmental sustainability, and uphold ethical standards.
尽管 Evo 并不直接操纵任何遗传物质，但它可能提升基因工程项目的效率。公众担忧基因工程技术能力的增强可能扰乱自然环境、引发生态不确定性（例如释放被改造的生物体），从而导致生物多样性丧失，甚至催生出新的潜在有害物种（Macfarlane 等，2022）。虽然训练全基因组基础模型对生态系统的长期影响尚不明确，但更紧迫的是应考虑伴随基础设施与计算需求提升而带来的碳足迹问题（Nature Computational Science，2023）。Evo 等工具的能力，加上基因组编辑与生态工程等其他技术，进一步引发了" 科学应在多大程度上干预进化过程" 的复杂伦理争论。随着我们通过工具如 Evo 不断突破科学能力的边界，我们亟需思考人类发明与自然进化过程之间的互动与界限，力求在维护生态平衡、环境可持续性与伦理规范之间实现协调。

A.2. The path forward **未来的发展路径**

The path forward for the responsible use and development of tools like Evo is anchored in the establishment of clear, comprehensive guidelines that delineate ethical practices. These guidelines serve as a responsible AI framework, ensuring that all shareholders—researchers, developers, and users—have a common understanding of the safety and ethical dimensions inherent in genetic engineering. Coupled with robust oversight mechanisms, this approach aims to monitor and manage the application of Evo to prevent misuse and ensure its alignment with ethical standards. Furthermore, promoting transparency regarding the use of these technologies and fostering open dialogue among all parties will enhance trust and collaboration within the scientific community and beyond.
负责任地使用和开发 Evo 等工具的未来路径，应以建立清晰且全面的伦理操作指南为基础。这些指南应构成" 负责任 AI" 的框架，确保所有利益相关方——包括研究者、开发者与用户——都对基因工程所涉及的安全性与伦理维度拥有共同理解。结合强有力的监管机制，此路径旨在持续监督与管理 Evo 的应用，以防止其被滥用，并确保其符合伦理标准。此外，通过提升这些技术使用的透明度并促进各方之间的开放对话，将有助于增强科研界及更广泛社会之间的信任与合作。

To address disparities in access and capabilities, particularly in low-income countries, the strategy includes forging community partnerships and international collaborations. By offering targeted training and support, these partnerships can democratize access to advanced tools like Evo, enabling a broader spectrum of scientists and researchers to contribute to and benefit from genetic engineering innovations. At the policy level, investing in education and capacity building emerges as a pivotal element, equipping the next generation of scientists with the ethical acumen and technical skills to navigate the complexities of genetic research responsibly.
为应对获取与能力方面的不平等，尤其是在低收入国家，此策略还应包括建立社区合作伙伴关系及国际协作网络。通过提供有针对性的培训与支持，这些合作关系可推动如 Evo 之类先进工具的民主化获取，使更广泛的科研人员群体能参与并受益于基因工程创新。在政策层面，加大对教育与能力建设的投入是关键一步，能够使下一代科学家掌握所需的伦理意识与技术能力，以负责任的方式应对基因研究的复杂性。

Central to sustaining ethical innovation is the creation of a dynamic feedback loop that engages all shareholders in a continuous dialogue. By setting up mechanisms to collect and integrate feedback from those involved in or

impacted by Evo's applications, the process ensures that guidelines, policies, and practices are regularly refined in response to evolving ethical challenges and societal expectations. Collaborating with organizations such as the Global Alliance for Genomics and Health (GA4GH) (Rehm et al., 2021) to develop and update genetic engineering guidelines further solidifies this commitment to ethical excellence. This multifaceted approach not only addresses immediate concerns but also lays the groundwork for a future where genetic engineering advances in harmony with ethical principles and societal values.
实现伦理可持续创新的核心，是构建一个能让所有利益相关者参与其中的动态反馈机制。通过建立机制收集并整合来自 Evo 应用相关者及受其影响者的反馈信息，可确保操作指南、政策与实践在面对不断演变的伦理挑战与社会期望时持续更新。与全球基因组与健康联盟（GA4GH）（Rehm 等，2021）等组织合作，共同制定并更新基因工程指导方针，将进一步巩固这一伦理卓越承诺。这种多维度的方法不仅可应对当前的紧迫问题，也为未来基因工程在伦理原则与社会价值观引导下的发展奠定了坚实基础。

B. Methods

B.1. StripedHyena architecture

Evo is based on StripedHyena (Poli et al., 2023a), a state-of-the-art hybrid model architecture for sequence modeling. Evo comprises 32 blocks at a model width of 4096 dimensions. Each block contains a sequence mixing layer, tasked with processing information along the sequence dimension, and a channel mixing layer, focused on processing information along the model width dimension. In the sequence mixing layers, Evo employs 29 hyena layers, interleaved with 3 rotary (Su et al., 2024) self-attention layers at equal intervals. We parametrize convolutions in hyena operators using the modal canonical form described in (Massaroli et al., 2024). For the channel mixing layers, Evo employs gated linear units (Dauphin et al., 2017; Shazeer, 2020). Evo further normalizes the inputs to each layer using root mean square layer normalization (Zhang and Sennrich, 2019).
Evo 基于 StripedHyena（Poli 等，2023a），这是一种最先进的用于序列建模的混合模型架构。Evo 包含 32 个模块块，模型宽度为 4096 维。每个模块包括一个序列混合层（用于处理序列维度上的信息）和一个通道混合层（用于处理模型宽度维度上的信息）。在序列混合层中，Evo 使用了 29 个 Hyena 层，并以等间距插入了 3 个旋转自注意力层（Su 等，2024）。Hyena 操作中的卷积通过模态标准形式进行参数化（Massaroli 等，2024）。在通道混合层中，Evo 使用 gated linear units（门控线性单元）（Dauphin 等，2017；Shazeer，2020）。此外，Evo 对每层输入使用均方根层归一化（Zhang 和 Sennrich，2019）。

Hyena layers Hyena (Poli et al., 2023a) is a sequence mixer implementing an input-dependent (data-controlled) operator via a composition of short convolutions, long convolutions and gating (Figure ??B). Hyena belongs to the class of deep signal processing primitives (Poli et al., 2023a; Fu et al., 2024; Massaroli et al., 2024), designed for efficient, input-dependent computation in large-scale sequence models. Input-dependence enables an architecture built with deep signal processing layers to adapt computation based on the input, enabling in-context learning (Arora et al., 2023; Bhattamishra et al., 2023). These layers rely on structured operators compatible with fast multiplication algorithms and can thus be evaluated in subquadratic time using, e.g., Fast Fourier Transforms for convolutions. The operators are parametrized implicitly, e.g., learning a map from positional embeddings, or the input, to the parameters of the operator. Typical choices of implicit parametrizations are linear projections, hypernetworks (Romero et al., 2021; Poli et al., 2023a) or linear state-space models in modal or companion form (Gupta et al., 2022; Gu et al., 2022; Massaroli et al., 2024; Orvieto et al., 2023). The blueprint of a hyena operator forward pass is summarized below.
Hyena 层是 Hyena（Poli 等，2023a）提出的一种序列混合器，它通过短卷积、长卷积与门控机制的组合，实现输入依赖（数据控制）操作（图 ??B）。Hyena 属于深度信号处理原语类方法（Poli 等，2023a；Fu 等，2024；Massaroli 等，2024），专为大规模序列模型中的高效、输入依赖计算设计。输入依赖性使得基于深度信号处理层构建的架构能够根据输入自适应调整计算，实现上下文学习能力（Arora 等，2023；Bhattamishra 等，2023）。这些层依赖于结构化操作符，兼容快速乘法算法，例如使用快速傅里叶变换进行卷积，从而能以亚二次时间复杂度进行计算。该类操作符采用隐式参数化方式，例如从位置嵌入或输入中学习映射至操作符参数。常见的隐式参数化方式包括线性投影、超网络（Romero 等，2021；Poli 等，2023a）或模态/伴随形式下的线性状态空间模型（Gupta 等，2022；Gu 等，2022；Massaroli 等，2024；Orvieto 等，2023）。以下为 Hyena 操作符前向传递的蓝图：

Algorithm 1 ConvProjection

Require: Input sequence $u \in R^{L \times D}$
Require: Inner dimension D_v
1: In parallel across L: $\hat{z} = \text{Linear}(u)$, Linear : $R^D \to R^{3Dv}$
2: In parallel across D_z: $z = \text{DepthwiseConv1d}(h, \hat{z})$, h is a short convolution filter
3: Reshape and split z into q, k, v. Dimensions of one element are $q \in R^{D_z \times L}$
4: return q, k, v

Self-attention layers Self-attention is the core sequence mixing operator of Transformer models. Self-attention constructs the output sequence as a weighted combination of the input elements, where the weights themselves are

Algorithm 2 Forward pass

Require: Input sequence $u \in R^{L \times D}$
Require: Order N, model width D, sequence length L, inner dimension D_z
$q, k, v = \text{ConvProjection}(u)$
$h = \text{ImplicitFilter}(L)$
In parallel: $v \leftarrow k \cdot v$
In parallel across D_z: $v_t \leftarrow \text{FFTConv}(h, v)_t$
In parallel: $v \leftarrow q \cdot v$
return $y = v$

input-dependent. Given an input sequence, the forward pass of an (unnormalized) self-attention layer is:

$$(q, k, v) \mapsto A(q, k)v, \quad A(q, k) = \text{softmax}\left(qk^T\right)$$

where queries $q \in R^{L \times D}$ and keys $k \in R^{L \times D}$ and values $v \in R^{L \times D}$ are obtained through a linear transformation of the input e.g., $v = uW_v$. The softmax is applied to rows of A. The query, key, value terminology is borrowed from databases, where keys are used to index stored values. Conceptually, the values of the attention matrix $A(q, k)$ measure the similarity between queries and keys akin to matching queries to keys in a database.
自注意力层是 Transformer 模型中核心的序列混合操作符。它通过对输入元素进行加权组合构建输出序列，权重本身依赖于输入。对于一个输入序列，自注意力层的（未归一化）前向计算表达式为：

$$(q, k, v) \mapsto A(q, k)v, \quad A(q, k) = \text{softmax}\left(qk^T\right)$$

其中查询 q、键 k 和值 v 均来自对输入 u 的线性变换（例如 $v = uW_v$），$q, k, v \in R^{L \times D}$。Softmax 应用于矩阵 A 的行。术语 query、key、value 来源于数据库领域，表示按 key 检索 value 的操作；在此，attention 矩阵 $A(q, k)$ 的值可视为衡量查询与键之间相似度的指标，类似于查询匹配。

Positional embeddings By itself, the self-attention operator does not have any notion of the different positions of the input embeddings in an input sequence. For this reason, it is generally supplemented with a positional encoding mechanism. The attention layers of StripedHyena utilize a rotary position embedding mechanism (RoPE) to model relative positional information (Su et al., 2024). Position information is encoded by rotating the query and key token vectors of the attention operator. Specifically, RoPE implements a rotation to queries and keys, with the rotation magnitude defined as a function of their relative position in the sequence.
位置嵌入由于自注意力操作本身并不具备输入序列中各位置的位置信息，因此通常需配合位置编码机制加以补充。StripedHyena 中的注意力层采用旋转位置编码（RoPE）来建模相对位置信息（Su 等，2024）。RoPE 通过对查询与键向量进行旋转来编码位置，其旋转幅度为该位置相对于序列中其他位置的函数。

To extend the context window length from 8 k to 131 k during our second pretraining stage, we apply linear position interpolation to extend the rotary position embedding applied in the first pretraining stage at 8 k sequence length (for details, see Chen et al. (2023)). Interpolating enables the model to continue leveraging its learned representations when applied to longer sequences than it was originally trained on. We also tested other position interpolation methods but found that they performed slightly worse than linear interpolation on our data.
为将第二阶段预训练的上下文长度从 8k 扩展至 131k，我们对第一阶段 8k 时使用的 RoPE 编码进行了线性位置插值（详见 Chen 等，2023）。插值方法允许模型在面对超出训练长度的新输入时仍能利用已学得的表征能力。我们也测试了其他位置插值方法，但结果略逊于线性插值。

Tokenization In language modeling, tokens describe the smallest unit of semantic information that is used by a model to process language. For example, tokens can indicate individual words of a vocabulary or even lower-level semantic information such as individual characters. Tokenization describes the process of mapping these semantic language units, such as words or characters, to unique integer values, each indicating an entry in a lookup table. These integer values are mapped by embedding layers to vectors, which are then processed by the model in an end-to-end fashion. Evo tokenizes DNA sequences at single-nucleotide resolution, using the UTF-8 encoding implemented in Python. During pretraining, Evo uses an effective vocabulary of four tokens, one per base, from a total vocabulary of 512 characters. We use the additional characters to enable prompting with special tokens during generation with finetuned models.
分词（Tokenization）在语言建模中，token 表示模型处理语言时的最小语义单元，例如词汇表中的单词，甚至是字符级的语义单位。分词过程即为将这些语义单位（如单词或字符）映射为唯一整数值，作为查找表中的索引。嵌入层随后将这些整数映射为向量，供模型端到端处理。Evo 以单核苷酸分辨率对 DNA 序列进行分词，使用 Python 中的 UTF-8 编码。在预训练过程中，Evo 使用了 512 字符的总词表，其中有效 DNA 碱基词汇仅为 4 个（每种碱基一个）。额外字符用于提示词构造等目的。

B.2. OpenGenome datasets

The OpenGenome pretraining dataset (S3 for summary statistics) was compiled from three different sources: 1) Bacterial and archaeal genomes from the Genome Taxonomy Database (GTDB) v214.1 (Parks et al., 2015), 2)

curated prokaryotic viruses from the IMG/VR v4 database (Camargo et al., 2023), and 3) plasmid sequences from the IMG/PR database (Camargo et al., 2024). For GTDB, representative genomes for each species were retained to reduce data redundancy.
OpenGenome 预训练数据集（统计摘要见 S3）汇总自三个不同来源：1）来自 Genome Taxonomy Database（GTDB）v214.1 的细菌和古菌基因组（Parks et al., 2015）；2）来自 IMG/VR v4 数据库的经人工整理的原核病毒序列（Camargo et al., 2023）；3）来自 IMG/PR 数据库的质粒序列（Camargo et al., 2024）。对于 GTDB，为了减少数据冗余，仅保留每个物种的代表性基因组。

For IMG/PR, only one representative per plasmid taxonomic unit (PTU) was kept. For IMG/VR, sequences were retained only if they were labeled as "High-confidence" according to the database metadata, and only one representative per viral operational taxonomic unit (vOTU) was kept. These sequences were further curated to remove potential eukaryotic viruses by keeping only sequences whose assigned taxonomic classification was found within a prokaryotic host at least twice. Next, the remaining taxonomic classifications were inspected and further filtered to exclude all viruses assigned to any of 19 families (Adenoviridae, Caliciviridae, Coronaviridae, Filoviridae, Flaviviridae, Hantaviridae, Hepadnaviridae, Herpesviridae, Orthomyxoviridae, Papillomaviridae, Poxviridae, Reoviridae, Retroviridae, Rhabdoviridae, Circoviridae, Geminiviridae, Picobirnaviridae) or 12 orders (Amarillovirales, Durnavirales, Geplafovirales, Herpesvirales, Lefavirales, Ortervirales, Orthopolintovirales, Piccovirales, Picornavirales, Priklausovirales, Cirlivirales, Mulpavirales). Next, viruses with poor taxonomic specificity were excluded, including those with no assigned realm at all, and those only assigned up to the level of r:Riboviria, r:Monodnaviria, k:Heunggongvirae, k:Bamfordvirae, p:Preplasmiviricota, p:Cressdnaviricota, p:Pisuviricota, or c:Tectiliviricetes.
对于 IMG/PR，每个质粒分类单元（PTU）只保留一个代表序列。对于 IMG/VR，仅保留元数据标注为“高置信度”的序列，并对每个病毒操作分类单元（vOTU）只保留一个代表序列。这些序列进一步被整理以剔除潜在的真核病毒，仅保留在原核宿主中被发现至少两次的分类序列。随后，对剩余的分类标签进行人工检查，进一步排除属于以下 19 个病毒科或 12 个病毒目中的病毒：Adenoviridae、Caliciviridae、Coronaviridae、Filoviridae、Flaviviridae、Hantaviridae、Hepadnaviridae、Herpesviridae、Orthomyxoviridae、Papillomaviridae、Poxviridae、Reoviridae、Retroviridae、Rhabdoviridae、Circoviridae、Geminiviridae、Picobirnaviridae，以及 Amarillovirales、Durnavirales、Geplafovirales、Herpesvirales、Lefavirales、Ortervirales、Orthopolintovirales、Piccovirales、Picornavirales、Priklausovirales、Cirlivirales、Mulpavirales。接着，进一步过滤掉分类分辨率低的病毒，包括没有被分配到任何界别的序列，以及仅被分类到如 r:Riboviria、r:Monodnaviria、k:Heunggongvirae、k:Bamfordvirae、p:Preplasmiviricota、p:Cressdnaviricota、p:Pisuviricota 或 c:Tectiliviricetes 等高级分类标签的病毒。

The CRISPR/Cas and IS200/IS605 finetuning datasets were compiled from a previously described custom database gathered from multiple sources (Wei et al., 2023). Briefly, this custom database includes genomic and metagenomic sequence data from NCBI RefSeq O'Leary et al. (2016), UHGG (Almeida et al., 2021), JGI IMG (Chen et al., 2021), the Gut Phage Database (Camarillo-Guerrero et al., 2021), the Human Gastrointestinal Bacteria Genome Collection (Forster et al., 2019), MGnify (Mitchell et al., 2020), Youngblut et al. (2020) animal gut metagenomes, MGRAST (Meyer et al., 2008), and Tara Oceans samples (Sunagawa et al., 2015).
CRISPR/Cas 和 IS200/IS605 微调数据集是从一个先前描述的自定义数据库中整理而来（Wei et al., 2023）。简而言之，该数据库包含来自多个来源的基因组和宏基因组序列数据，包括：NCBI RefSeq（O'Leary et al., 2016）、UHGG（Almeida et al., 2021）、JGI IMG（Chen et al., 2021）、肠道噬菌体数据库（Camarillo-Guerrero et al., 2021）、人类胃肠道细菌基因组集合（Forster et al., 2019）、MGnify（Mitchell et al., 2020）、Youngblut 等（2020）动物肠道宏基因组、MGRAST（Meyer et al., 2008）以及 Tara Oceans 项目样本（Sunagawa et al., 2015）。

To compile the CRISPR/Cas genomic loci, this custom database was searched using profile HMM models and the HMMER software package to identify Cas9, Cas12, and Cas13 sequences (Finn et al., 2011). Several pHMMs were collected from the CRISPRCasTyper annotation tool (Russel et al., 2020), and a recent computational survey of TnpB and Cas12 (Altae-Tran et al., 2023). Custom Cas13 pHMMs that were previously generated by our group were also used (Wei et al., 2023). These models were searched against our large custom database using hmmsearch and the parameter "Z 1000000". All hits that met $E < 1 \times 10^{-6}$ with at least one pHMM were kept. Only hits that were at least 300 aa long and covered over 80% of the pHMM were kept. For all hits to a given pHMM, only proteins that were within the middle 99% of the size distribution were kept. Corresponding genetic loci were extracted from the database, including 8,192 nucleotides of flanking sequence on both the $5'$ and $3'$ ends of the Cas effector CDS. The tool minced was used to identify CRISPR arrays in the flanking sequences using the parameters "-minRL 18 -maxRL 50 -minSL 18 -maxRL 50" (?). Only loci with both a predicted Cas effector and a CRISPR array were retained. The final CRISPR/Cas loci were extracted by first identifying the subsequence that covered both the Cas effector and the CRISPR array, and then including additional flanking nucleotides on both sides up until 8,192 were retained for finetuning purposes. Only 1 locus per 90% identity Cas cluster was retained, clustered using the MMseqs2 command "easy-cluster –cluster-reassign –cluster-mode 0 –cov-mode 0 -c 0.7 –min-seq-id 0.9" (Steinegger and Söding, 2017).
为了整理 CRISPR/Cas 基因组位点，我们使用 profile HMM 模型和 HMMER 软件包（Finn et al., 2011）搜索该自定义数据库，以识别 Cas9、Cas12 和 Cas13 序列。所使用的多个 pHMM 模型来自 CRISPRCasTyper 注释工具（Russel et al., 2020）以及最近对 TnpB 和 Cas12 的计算研究（Altae-Tran et al., 2023）。我们还使用了实验室先前构建的 Cas13 自定义 pHMM 模型（Wei et al., 2023）。这些模型使用 hmmsearch 和参数"Z 1000000" 搜索我们的大型数据库。保留所有 E 值小于 1×10^{-6} 且命中至少一个 pHMM 的序列。仅保留长度至少为 300 个氨基酸并覆盖超过 pHMM 80% 的命中项。对于某一特定 pHMM 的所有命中，仅保留位于该蛋白质长度分布中间 99%

的蛋白序列。随后从数据库中提取相关基因位点，包括在 Cas 效应子 CDS 两端各 8,192 个碱基的侧翼序列。使用 minced 工具在这些侧翼序列中识别 CRISPR array，参数为"-minRL 18 -maxRL 50 -minSL 18 -maxRL 50"（参数来源未明）。仅保留同时含有预测的 Cas 效应子和 CRISPR array 的位点。最终的 CRISPR/Cas 位点通过首先识别覆盖 Cas 效应子和 CRISPR array 的子序列，并在两端补充碱基直到总长达 8,192 nt 用于微调。每个 90% 相似性的 Cas 簇仅保留一个位点，使用 MMseqs2 命令"easy-cluster –cluster-reassign –cluster-mode 0 –cov-mode 0 -c 0.7 –min-seq-id 0.9" 进行聚类（Steinegger and Söding, 2017）。

To compile the IS200/IS605 loci, this custom database was searched using a Pfam Y1 HUH Transposase pHMM model (Pfam ID: PF01797). This pHMM identifies IS200/IS605 TnpA proteins. All matches meeting E-value $< 1 \times 10^{-6}$ that covered at least 80% of the pHMM and were less than 400 aa were kept. 8,196 nt of CDS-flanking sequence was then extracted for each hit. Loci that also contained TnpB coding sequences were identified using previously compiled pHMMs (Altae-Tran et al., 2023), and a custom pHMM compiled using jackhmmer and the ISDra2 TnpB as an initial query against the MGnify protein database, followed by a MAFFT alignment of hits and pHMM construction with HMMER (Finn et al., 2011; Mitchell et al., 2020; Katoh et al., 2002). Hits that were between 250 and 650 aa in length were retained, and only loci where the distance between the beginning and end of the TnpA and TnpB sequences was less than 2500 nt were retained. For TnpA-only loci, up to 300 nt of flanking sequence were added to either side of the CDS. For TnpA+TnpB loci, up to 300 nt were added to the TnpA side of the IS200/IS605 element, while 600 nt were added to the TnpB side (to account for the presence of an ω RNA). Only 1 locus per 90% identity TnpA cluster was retained.
为了整理 IS200/IS605 位点，使用 Pfam 中的 Y1 HUH 转座酶 pHMM 模型（Pfam ID: PF01797）搜索自定义数据库。该模型可识别 IS200/IS605 TnpA 蛋白。保留所有 E 值小于 1×10^{-6}，覆盖 pHMM 至少 80%，且长度小于 400 aa 的命中项。随后提取每个命中项的 8,196 个碱基 CDS 侧翼序列。使用先前构建的 pHMM（Altae-Tran et al., 2023）以及我们使用 jackhmmer 和 ISDra2 TnpB 作为初始查询在 MGnify 蛋白数据库中建立的自定义 pHMM 来识别含 TnpB 编码序列的位点。随后使用 MAFFT 对命中项进行比对，并使用 HMMER 构建新的 pHMM（Finn et al., 2011; Mitchell et al., 2020; Katoh et al., 2002）。保留长度在 250 到 650 aa 之间的命中序列，仅保留 TnpA 和 TnpB 起始与终止位置之间距离小于 2500 nt 的位点。对于仅含 TnpA 的位点，在 CDS 两侧分别添加最多 300 nt 的侧翼序列；对于同时含有 TnpA 和 TnpB 的位点，在 TnpA 一侧添加最多 300 nt，在 TnpB 一侧添加最多 600 nt（以考虑 ω RNA 的存在）。每个 90% 相似性 TnpA 聚类中仅保留一个位点。

B.3. Training procedure **训练过程**

We pretrain Evo in two stages, first with a context size of 8 k tokens, followed by a second stage where we increase the context size to 131 k tokens. Multi-stage sequence length pretraining has been shown to reduce the overall number of compute hours required to train long context models (Xiong et al., 2023).
Evo 的预训练分为两个阶段：第一阶段使用 8k 上下文长度，第二阶段将上下文长度扩展至 131k。多阶段的序列长度预训练已被证实能显著减少训练长上下文模型所需的总计算时间（Xiong 等，2023）。

In total, we trained Evo in stage 1 on 64 Nvidia H100 GPUs and on 128 Nvidia A100 GPUs in stage 2. In total, Evo was trained on approximately 340B tokens, using approximately 2×10^{22} FLOPS.
第一阶段训练在 64 张 Nvidia H100 GPU 上完成，第二阶段在 128 张 Nvidia A100 GPU 上完成。Evo 总共在约 3400 亿个 token 上完成训练，总计算量约为 2×10^{22} FLOPS。

Dataloading We use sequence packing to generate training samples, where multiple DNA sequences are appended until the context length (8 k or 131 k) is reached. Individual DNA sequences are separated by end-of-sequence (EOS) tokens. Depending on the dataset or task, we additionally prepend a special class (CLS) token to condition the model, for example, to steer its generations through prompting.
数据加载方面，我们使用序列打包方式生成训练样本：将多个 DNA 序列拼接至上下文长度（8k 或 131k）为止，单个 DNA 序列间以 EOS（结束）标记分隔。根据具体任务或数据集，有时还会在开头添加 CLS（类别）标记，用于通过提示条件引导模型生成。

Hyperparameter tuning and direct model comparisons Before training Evo, we carried out hyperparameter tuning on partially trained 7B Transformer++ (see B.4) models and compared to similarly sized Hyena and StripedHyena models. In particular, we swept batch size, learning rate and other architectural details. Even when controlling for training iterations instead of compute (FLOPS), Transformer++ performance is substantially worse than StripedHyena (see S4). Out of all the baselines, we find that StripedHyena achieves the overall lowest perplexity at the 7B scale, consistent with the scaling rates presented in Figure ??G.
超参数调优与模型比较：在训练 Evo 之前，我们对部分训练的 7B 参数规模 Transformer++ 模型进行了超参数调优，并与同等规模的 Hyena 和 StripedHyena 模型进行了对比。调参内容包括 batch size（批次大小）、学习率和架构参数等。即使控制训练轮次而非 FLOPS，Transformer++ 的性能仍远逊于 StripedHyena（详见 S4）。在所有基线模型中，StripedHyena 在 7B 规模下实现了最低的困惑度（perplexity），这一结果也与图 ??G 所展示的缩放率一致。

B.4. Scaling laws **缩放法则**（Scaling Laws）

We compare different classes of architectures via a compute-optimal protocol, aimed at evaluating results on the compute-optimal frontier. Compute-optimal analysis studies the best performance of a pretraining run given a

compute budget, typically indicated in floating point operations (FLOPs), and achieved by optimally allocating portions of the compute budget to model size and dataset size. Architecture types differ in compute efficiency, as well as how they allocate this compute budget.
我们采用计算最优协议来比较不同类别的架构，目的是评估其在计算最优前沿上的表现。计算最优分析研究在给定计算预算（通常以 FLOPs 浮点操作数衡量）下预训练的最佳性能，并通过对模型规模与数据集规模的最优分配来实现。不同架构在计算效率及预算分配策略方面存在差异。

We started by tuning hyperparameters such as learning rate and batch size for Transformer++ with a grid search, then used the same values for all architectures except in settings where numerical instability was observed. To address instability, we lowered the learning rate gradually and repeated the experiment until convergence. In all experiments, we trained models with 8,192 tokens in context length. For each compute budget defined by a total FLOP count, we varied the model sizes (6 million to 1 billion parameters) and the number of tokens trained. To measure model performance, we use the perplexity metric, which indicates how well an autoregressive model performs at predicting the next token of a sequence and is highly correlated with performance on downstream tasks. A lower perplexity value indicates better performance.
我们首先通过网格搜索对 Transformer++ 模型的学习率和 batch size 等超参数进行调优，并在无数值不稳定情况下将其应用于其他架构。若发现不稳定，则逐步降低学习率并重复实验直至收敛。在所有实验中，模型上下文长度均为 8192 个 token。对于每个由 FLOPs 定义的计算预算，我们在模型参数量（600 万至 10 亿）和训练 token 数量之间进行变化。模型性能通过困惑度（perplexity）衡量，该指标反映自回归模型预测下一个 token 的能力，并与下游任务性能高度相关。困惑度越低，表示性能越好。

Scaling laws procedure We provide a summary of the steps involved in our scaling laws analysis. Quantifying scaling rates allows us to predict performance as model size, dataset size, and compute grow.
缩放法则流程以下为我们缩放法则分析的步骤概述。量化缩放率可帮助预测在模型规模、数据量及计算量扩展下的性能变化趋势。

1. Define a set of compute budgets to study. We use $8 \times 10^{18}, 2 \times 10^{19}, 4 \times 10^{19}$ and 8×10^{19} FLOPS.
   定义一组待研究的计算预算值。本研究使用的 FLOPs 分别为 $8 \times 10^{18}, 2 \times 10^{19}, 4 \times 10^{19}$ 与 8×10^{19}。

2. Calculate the FLOPS (floating point operations) required to process a fixed input size for the model architecture of interest (i.e. the "cost" of using the model).
   计算给定架构在处理固定输入大小时所需的 FLOPs（即使用该模型的" 成本"）。

3. Identify the model's compute-optimal allocation for each compute budget
   对于每个计算预算，识别该模型的计算最优分配方案：

   (a) Select a wide range of possible model sizes, and calculate for each model size the corresponding number of tokens that need to be processed to reach the compute budget. Other hyperparameters are chosen according to Table S1. We generally observe minor changes to model topology (depth, width) to only minimally affect perplexity, aligning our results with the findings presented by Kaplan et al. (2020) for Transformers.
   选取多种模型规模，并计算在该规模下需要训练的 token 数以匹配预算；其他超参数参见表 S1。我们发现模型拓扑（如深度与宽度）的微调对困惑度影响较小，与 Kaplan 等（2020）对 Transformer 的结论一致。

   (b) Train a model of each size and record its performance (e.g., in terms of perplexity).
   对每种规模的模型进行训练，并记录其性能（如困惑度）。

   (c) Identify the optimal compute allocation: Following prior analysis, we fit a second-order polynomial as a function from (log) model size to perplexity, and extract obtained the compute-optimal point as its minimum. The compute-optimal point identifies the optimal allocation of model size and training tokens at the given compute budget.
   找到最优计算分配：参照以往方法，对 log 模型规模与困惑度之间关系拟合二次多项式，其极小值即为该计算预算下的最优点，即模型规模与训练 token 的最优组合。

After deriving the compute-optimal scaling rates (Figure ??G), we compare architectures and compute optimal allocation of tokens and model size (Figure S5). In Figure S3, we also show rates for compute-suboptimal model sizes by architecture. In particular, we quantify the effect on perplexity scaling caused by a suboptimal allocation of compute budget to model or dataset size (e.g., training a smaller model for more tokens). We estimate the compute-optimal model size for each compute budget, then reduce it by a percentage (the offset). The corresponding perplexity is obtained via the IsoFLOP curves (Figure ??F). Transformer++ perplexity scaling rapidly degrades outside the compute-optimal frontier, in contrast to Hyena and StripedHyena. Architecture details of models trained for our scaling law analysis provided in Table S1.
得出计算最优缩放率后（图 ??G），我们对不同架构下的最优 token 分配与模型规模进行了比较（图 S5）。图 S3 展示了不同架构在非最优模型规模下的缩放性能。我们评估在将计算预算偏离最优点（如将模型缩小、训练更多 token）时困惑度的变化，依据 IsoFLOP 曲线获得相关指标（图 ??F）。结果显示，Transformer++ 架构在远离最优前沿时困惑度明显恶化，而 Hyena 与 StripedHyena 更为稳定。各模型架构的训练细节列于表 S1。

Transformer++ We use a modern decoder-only Transformer architecture with rotary position embeddings (Su et al., 2024), pre-norm with root mean square layer normalization, and SwiGLU as channel mixer. The inner width of the SwiGLU is 4/3 the model width. We experimented with grouped-query attention (GQA) (Ainslie et al., 2023) and found minimal differences in final loss, suggesting the technique may be suited to DNA sequence modeling, in order to further reduce memory footprint during inference. All scaling results with Transformer++ do not use GQA.
Transformer++ 架构采用现代的仅解码器式 Transformer，配备旋转位置编码（RoPE，Su 等，2024）、前置归一化（pre-norm）与均方根层归一化，并使用 SwiGLU 作为通道混合器，其内部宽度为模型宽度的 4/3。我们测试了 grouped-query attention（GQA）（Ainslie 等，2023），但最终所有缩放实验均未使用 GQA，因其影响有限。

Hyena The Hyena baseline is designed with the same architecture improvements applied to the Transformer++ model. We replace all multi-headed self-attention layers with hyena layers, and use a modal canonical parametrization for the long convolution, with state dimension 8.
Hyena 架构基线在保留 Transformer++ 的架构改进基础上，将全部多头自注意力层替换为 Hyena 层，并使用模态标准形式（modal canonical form）参数化长卷积，状态维度设为 8。

Mamba We use the implementation of Mamba as provided by the authors in the public repository.
Mamba 架构采用作者在公开仓库提供的实现版本。

B.5. Generating DNA sequences with Evo **使用 Evo 生成 DNA 序列**

We sample sequences from Evo using standard top-k and temperature-based methods for autoregressive models. Evo benefits from the fast recurrent mode of hyena layers, enabling lower latency and memory cost (Massaroli et al., 2024; Poli et al., 2023b).
我们采用标准的 top-k 与温度采样策略，从 Evo 中生成 DNA 序列。得益于 Hyena 层的快速递归模式，Evo 拥有更低的延迟与内存成本（Massaroli 等，2024；Poli 等，2023b）。

In particular, we use the recurrent form of the modal canonical form as shown in (Massaroli et al., 2024), first processing the prompt with a Fast Fourier Transform modified to return output and state. We use a cache for the states of short convolutions. Evo can generate sequences of up to 650 k nucleotides on a single 80GB GPU, in contrast to other long context methods for dense Transformers requiring a larger number of nodes. We use standard kv-caching for rotary attention layers in StripedHyena.
我们具体使用 modal canonical form 的递归形式，在处理提示词时结合快速傅里叶变换（FFT）以输出状态并缓存短卷积状态。

Controllable generation. We follow standard language model prompting techniques that condition generation on a given prefix. For class-conditional generation we prompt with a single token, representing the desired class, or genomic sequence type (e.g. Cas system, IS200/605). The model can also be steered by prompting on desired DNA subsequences. 可控生成：我们采用标准的语言模型提示策略，通过给定前缀条件化生成。例如，对于类别条件生成任务，我们用单个标记表示目标类别或基因组类型（如 Cas 系统或 IS200/605 元件）。此外，模型还可通过提示特定 DNA 子序列进行定向生成。

B.6. Multimodal evaluations **多模态评估**

B.6.1. Protein function prediction **蛋白质功能预测**

We used DMS datasets to benchmark protein and nucleotide language models in their ability to predict mutational effects on protein function. In all cases, we used the nucleotide sequences reported by the original study authors. We limited our analysis to E. coli and human proteins, where E. coli protein information is contained in the Evo training dataset but where human proteins are not.
我们使用 DMS（深度突变扫描）数据集对蛋白质语言模型和核苷酸语言模型在预测突变对蛋白质功能的影响方面进行基准测试。所有情况下，我们均使用原始研究作者报告的核苷酸序列。我们的分析仅限于大肠杆菌和人类蛋白质，其中大肠杆菌蛋白质信息包含在 Evo 的训练数据集中，而人类蛋白质则不包括在内。

To compile the nucleotide information from E. coli DMS studies, we used all of the datasets listed in the ProteinGym benchmark for which we could also find nucleotide-level information reported by the original study authors. This resulted in six studies: a β-lactamase DMS by Firnberg et al. (2014), a β-lactamase DMS by Jacquier et al. (2013), a CcdB DMS by Adkar et al. (2012), a multi-protein thermostability dataset by Tsuboyama et al. (2023), an IF-1 DMS by Kelsic et al. (2016), and an Rnc DMS by Weeks and Ostermeier (2023).
为整理来自大肠杆菌 DMS 研究的核苷酸信息，我们使用了 ProteinGym 基准中列出的所有数据集，并确保这些研究也由原始作者提供了核苷酸级别的信息。最终共选取了六项研究：Firnberg 等人（2014）关于 β-内酰胺酶的 DMS，Jacquier 等人（2013）关于 β-内酰胺酶的 DMS，Adkar 等人（2012）关于 CcdB 的 DMS，Tsuboyama 等人（2023）提供的多蛋白质热稳定性数据集，Kelsic 等人（2016）关于 IF-1 的 DMS，以及 Weeks 和 Ostermeier（2023）关于 Rnc 的 DMS。

To compile the nucleotide information from human DMS studies, we narrowed the scope of the set of datasets used in our human benchmark to the human datasets used by Livesey and Marsh (2023) to benchmark mutational effect predictors. We also limited our analysis to studies where we could also find nucleotide-level information reported by the original study authors. This resulted in six studies: a CBS DMS by Sun et al. (2020), a GDI1

DMS by Silverstein et al. (2022), a PDE3A DMS by Garvie et al. (2021), a P53 DMS by Kotler et al. (2018), a P53 DMS by Giacomelli et al. (2018), and a BRCA1 DMS by Findlay et al. (2018).
为整理人类 DMS 研究中的核苷酸信息，我们将数据集范围限定为 Livesey 和 Marsh（2023）在评估突变效应预测器时使用的人类数据集，并进一步筛选出可从原始研究中获取核苷酸级别信息的研究。最终选取六项研究：Sun 等人（2020）关于 CBS 的 DMS，Silverstein 等人（2022）关于 GDI1 的 DMS，Garvie 等人（2021）关于 PDE3A 的 DMS，Kotler 等人（2018）和 Giacomelli 等人（2018）关于 P53 的 DMS，以及 Findlay 等人（2018）关于 BRCA1 的 DMS。

We compared Evo (pretrained with 8 k context) to three nucleotide language models: GenSLM 2.5B, which was trained with a codon vocabulary on sets of genes from prokaryotic organisms (Zvyagin et al., 2023); Nucleotide Transformer 2B5_multi_species, which was trained with a 6-mer nucleotide vocabulary on genome sequences from prokaryotic and eukaryotic species (Dalla-Torre et al., 2023); and RNA-FM, which was trained on a single-nucleotide vocabulary on short ncRNA sequences (Chen et al., 2022). We also compared Evo to several protein language models trained on non-redundant, generic corpuses of protein sequences: CARP 640M (Yang et al., 2024), ESM-1v (Meier et al., 2021), ESM-2 650M, ESM-2 3B (Lin et al., 2023), ProGen2 large, and ProGen 2 xlarge (Madani et al., 2023). For studies that provide models with multiple parameter sizes, we selected the largest size on which we could perform inference with an 80 GB Nvidia H100 GPU on sequences from all of our benchmarked studies without exceeding GPU memory. We also included ESM2 650M and ProGen2 large given that these models have sometimes shown better performance at function prediction than larger variants of these models (Notin et al., 2023).
我们将 Evo（使用 8k 上下文预训练）与三种核苷酸语言模型进行了比较：GenSLM 2.5B（Zvyagin 等人，2023），该模型在原核生物的基因集上使用密码子词汇表进行训练；Nucleotide Transformer 2B5_multi_species（Dalla-Torre 等人，2023），该模型在原核与真核生物的基因组序列上使用 6-mer 核苷酸词汇表训练；以及 RNA-FM（Chen 等人，2022），该模型在短 ncRNA 序列上使用单核苷酸词汇表训练。我们还将 Evo 与几个在非冗余通用蛋白质序列语料库上训练的蛋白质语言模型进行了比较：CARP 640M（Yang 等人，2024）、ESM-1v（Meier 等人，2021）、ESM-2 650M 和 ESM-2 3B（Lin 等人，2023）、ProGen2 large 和 ProGen2 xlarge（Madani 等人，2023）。对于具有多个参数规模的模型，我们选择在 80GB Nvidia H100 GPU 上能在所有基准研究中完成推理的最大模型版本。此外，由于某些情况下 ESM2 650M 和 ProGen2 large 在功能预测上的表现优于其更大版本（Notin 等人，2023），我们也将其包括在内。

To compare nucleotide and protein language models, we used all unique nucleotide sequences and their corresponding fitness values as reported by the original studies. Occasionally, we observed that the fitness values reported for nucleotide sequences differed from fitness values reported for protein sequences; in such cases, we used the fitness values reported for nucleotide sequences and evaluated the protein language models using the translated sequence. In cases where there are multiple nucleotide sequences for a single protein sequence due to different codon usage, the nucleotide language models were evaluated on each unique nucleotide sequence and the protein language models were evaluated on the coding sequence corresponding to each unique nucleotide sequence; this means that a protein language model could have been evaluated on the same protein sequence multiple times for a given study. Some studies report fitness values for mutations that involve stop codons; in such cases, we evaluated the nucleotide language model on the sequence containing the stop codon and excluded these examples from the protein language model benchmark.
为了比较核苷酸和蛋白质语言模型，我们使用原始研究报告的所有唯一核苷酸序列及其相应的适应度值。在个别情况下，核苷酸序列的适应度值与对应蛋白质序列的不同，此时我们采用核苷酸序列的适应度，并将蛋白质语言模型评估应用于翻译后的蛋白质序列。若多个核苷酸序列因密码子不同而对应于同一蛋白质序列，则对每个唯一的核苷酸序列单独评估核苷酸模型，同时蛋白质模型评估则使用与每条核苷酸序列对应的编码序列。这意味着在某一研究中，蛋白质模型可能对相同蛋白质序列被评估多次。部分研究报告的突变涉及终止密码子，我们对包含终止密码子的序列评估核苷酸模型，但将这些序列从蛋白质模型评估中排除。

We computed the Spearman correlation between the experimental fitness values and the sequence likelihood (for autoregressive language models) or the sequence pseudolikelihood (for masked language models). We assessed statistical significance of the Spearman correlation coefficient under a null hypothesis that the correlation coefficient is drawn from a t-distribution with $N-2$ degrees of freedom, where N is the number of samples over which we compute the correlation. We used this null distribution to compute a P value based on the observed correlation. We used the scipy Python library (https://scipy.org/) to compute these values.
我们计算实验适应度值与序列似然性（对于自回归语言模型）或序列伪似然性（对于掩码语言模型）之间的 Spearman 相关系数。我们在零假设下评估该相关系数的统计显著性，假设其服从自由度为 $N-2$ 的 t 分布，其中 N 是参与相关性计算的样本数量。我们根据观察到的相关系数值使用该分布计算 P 值。所有统计值使用 scipy Python 库（https://scipy.org/）计算完成。

B.6.2. ncRNA function prediction 非编码 RNA 功能预测

We used DMS datasets to benchmark protein and nucleotide language models based on their ability to predict mutational effects on ncRNA function. Given that no well established benchmark datasets exist for ncRNAs function prediction, we curated the literature for examples of ncRNA mutational scanning experiments. We obtained the following datasets: a ribozyme DMS by Kobori et al. (2015), a ribozyme DMS by Andreasson et al.

(2020), a tRNA DMS by Domingo et al. (2018), a tRNA DMS by Guy et al. (2014), a ribozyme DMS by Hayden et al. (2011), a ribozyme DMS by Pitt and Ferré-D'Amaré (2010), and a rRNA mutagenesis study by Zhang et al. (2009).
我们使用 DMS 数据集对蛋白质和核苷酸语言模型在预测突变对 ncRNA 功能的影响方面进行基准测试。由于目前尚无公认的 ncRNA 功能预测基准数据集，我们从文献中整理了有关 ncRNA 突变扫描实验的研究。我们获得了以下数据集：Kobori 等人（2015）的核酶 DMS，Andreasson 等人（2020）的核酶 DMS，Domingo 等人（2018）的 tRNA DMS，Guy 等人（2014）的 tRNA DMS，Hayden 等人（2011）的核酶 DMS，Pitt 和 Ferré-D'Amaré（2010）的核酶 DMS，以及 Zhang 等人（2009）的 rRNA 突变研究。

We compared Evo (pretrained with 8 k context) to the nucleotide language models described above. Similar to the methods applied to protein coding sequences above, we compiled experimental fitness values for each ncRNA variant. We computed the Spearman correlation between the experimental fitness values and the sequence likelihood (for autoregressive language models) or the sequence pseudolikelihood (for masked language models). Correlation coefficients and associate P values were computed as described above.
我们将使用 8k 上下文预训练的 Evo 与上述核苷酸语言模型进行了比较。与蛋白质编码序列的评估方法类似，我们为每个 ncRNA 突变体整理了其实验适应度值，并计算了实验适应度值与序列似然性（用于自回归语言模型）或伪似然性（用于掩码语言模型）之间的 Spearman 相关系数。相关系数及其 P 值的计算方法与前述一致。

B.6.3. Gene expression prediction from regulatory DNA 来自调控 DNA 的基因表达预测

To evaluate the model's ability to learn properties of regulatory DNA, we used a dataset reported by Kosuri et al. (2013) in which a set of E. coli promoters and a set of E. coli RBSs were combinatorially paired and the promoter-RBS pairs were experimentally tested for their effect on downstream mRNA and protein expression. We computed the sequence likelihood (for autoregressive language models) or the sequence pseudolikelihood (for masked language models) for each promoter-RBS pair, where we concatenated the sequence of the promoter directly with the sequence of the RBS.
为评估模型学习调控 DNA 属性的能力，我们使用了 Kosuri 等人（2013）报告的数据集。该数据集中，一组大肠杆菌启动子与一组大肠杆菌核糖体结合位点（RBS）进行了组合配对，并通过实验测量了各启动子-RBS 对对下游 mRNA 和蛋白质表达的影响。我们计算了每组启动子-RBS 配对的序列似然性（自回归模型）或伪似然性（掩码语言模型），其中我们将启动子序列与 RBS 序列直接连接。

We computed these likelihoods using Evo (pretrained with 8 k context) and the three other nucleotide language models described above. We used these likelihoods to predict continuous mRNA expression values and binarized protein expression values as reported in the original study. Protein expression was binarized using a cutoff at which expression values above 100,000 were treated as positive and values below 100,000 were treated as negative, where this cutoff was based on the bimodal distribution of protein expression values reported in the original study. We used the Spearman correlation coefficient to quantify the predictive performance for mRNA expression and the AUROC to quantify predictive performance for protein expression. We assessed statistical significance of the Spearman correlation coefficient with a t-distributed P-value as described above. We assessed the statistical significance of the AUROC with a permutation-based method in which a null distribution is constructed by permuting the binary labels and recomputing the subsequence AUROC. We performed 100,000 permutations to construct this null distribution.
我们使用 Evo（8k 上下文预训练）以及前述三种核苷酸语言模型计算了这些似然性，并用于预测原始研究中报告的连续 mRNA 表达值和二值化的蛋白质表达值。我们使用 100,000 为阈值将蛋白表达进行二值化，即表达值大于 100,000 视为阳性，小于 100,000 视为阴性，该阈值基于原始研究中报告的双峰分布。我们使用 Spearman 相关系数评估对 mRNA 表达的预测性能，使用 AUROC 评估对蛋白表达的预测性能。Spearman 相关系数的显著性检验方法如前所述，AUROC 的显著性检验采用置换法构建零分布，通过置换标签并重新计算 AUROC，进行 100,000 次置换以获得统计显著性。

We also attempted to quantify how well a given promoter-RBS pair was represented in the Evo pretraining data, as we hypothesized that promoter-RBS pairs that are seen more often in nature, and would thereby have higher language model likelihood, are also pairs that are more likely to lead to higher gene expression. We attempted to align the promoter-RBS sequences to bacterial genomic sequences using three methods. First, we constructed a BLAST database over the full GTDB using the makeblastdb command with default parameters (Ye et al., 2006). We then used the blastn command with default parameters where for each promoter-RBS pair we queried the database for significant hits. We used the number of returned BLAST hits to score each promoter-RBS pair (having no hits was scored as zero).
我们还尝试量化每组启动子-RBS 配对在 Evo 预训练数据中出现的频率，因为我们假设在自然界中更常见的启动子-RBS 组合可能在语言模型中具有更高的似然性，也更可能导致更高的基因表达。为此我们采用三种方法将启动子-RBS 序列与细菌基因组进行比对。首先，我们使用 makeblastdb 默认参数构建了一个包含完整 GTDB 的 BLAST 数据库（Ye 等人，2006），然后使用 blastn 默认参数对每个启动子-RBS 序列进行查询。我们使用返回的显著匹配数量作为每组配对的评分（无匹配计为零）。

Second, we used mmseqs to construct databases over the set of promoter-RBS pairs and over the full GTDB using the mmseqs createdb command (–dbtype 2) to create nucleotide databases. We then used mmseqs createindex (–search-type 3) to create a nucleotide search index. We then conducted an all-by-all search (–cov-mode 2) to search

for sequences in GTDB that aligned to the promoter-RBS queries. We used the number of significant alignments to score each promoter-RBS pair (having no alignments was scored as zero).
其次，我们使用 mmseqs 构建了启动子-RBS 配对序列数据库及完整 GTDB 数据库，使用 mmseqs createdb 命令（–dbtype 2）创建核苷酸数据库，随后使用 mmseqs createindex（–search-type 3）建立搜索索引，并通过 all-by-all 搜索（–cov-mode 2）在 GTDB 中寻找与启动子-RBS 查询序列匹配的序列。我们使用匹配数量为每组配对评分（无匹配计为零）。

Third, we attempted to align promoter-RBS sequences to the E. coli reference genome (RefSeq: GCF_000005845.2) using bowtie2. We used bowtie2-build with default parameters to build an index over the E. coli genome. We then treated promoter-RBS pairs as unpaired reads in a FASTA file and enabled multimapping with the -a flag to bowtie2. We used the number of alignments to score each promoter-RBS pair (having no alignments was scored as zero). We used the Spearman correlation coefficient to quantify predictive performance for mRNA expression and the AUROC to quantify predictive performance for protein expression and report the highest correlation values across the three attempted methods.
第三，我们尝试将启动子-RBS 序列与大肠杆菌参考基因组（RefSeq: GCF_000005845.2）进行比对，使用 bowtie2-build 默认参数构建索引，并将启动子-RBS 配对作为 FASTA 文件中的单端读取进行比对，同时使用 -a 参数启用多重比对。我们使用比对次数为每组配对评分（无比对计为零）。最后我们使用 Spearman 相关系数评估 mRNA 表达的预测性能，并使用 AUROC 评估蛋白表达的预测性能，并报告三种比对方法中相关性最高的结果。

B.7. CRISPR-Cas finetuning, generation, and downstream analysis **微调、生成与后续分析**

To generate CRISPR-Cas systems, we finetuned Evo by continuing to train the 8 k-context pretrained model on a dataset of CRISPR-Cas sequences, which was curated as described above. We retained most of the hyperparameters used during pretraining but set the batch size to 524 k tokens and an initial learning rate of 0.00009698, which was the learning rate at the final step of pretraining. During pretraining, we prepended a single class token corresponding to the type of Cas protein (Cas9, Cas12, or Cas13), which was identified as described above; this class token was then followed by the nucleotide sequence.
为了生成 CRISPR-Cas 系统，我们对 Evo 进行了微调，在 8k 上下文预训练模型的基础上继续训练上文所述的 CRISPR-Cas 序列数据集。我们保留了大部分预训练时使用的超参数，但将批大小设为 524k 个 token，初始学习率设为 0.00009698（即预训练最后一步的学习率）。在预训练中，我们在核苷酸序列前添加了一个类别 token，对应于 Cas 蛋白的类型（Cas9、Cas12 或 Cas13），其识别方式如上所述。

We also modified the dataloader such that each sample provided to the model during training would begin with the first token of the CRISPR-Cas sequence and, if a sequence was shorter than the context length, we padded the sequence to the remaining context (where padding did not contribute to the loss computation). This ensured that each training sample would correspond to a single CRISPR-Cas sequence. We finetuned the model for approximately 10 epochs.
我们还修改了数据加载器，使每个训练样本都以其 CRISPR-Cas 序列的首 token 开始。如果序列短于上下文长度，则使用 padding 填充剩余部分（padding 不计入损失计算），从而确保每个训练样本仅对应一个 CRISPR-Cas 序列。我们对模型微调了大约 10 个 epoch。

We prompted the model with a given class token for each sequence generation. We performed standard temperature-based and top-k autoregressive sampling (Chang and Bergen, 2023). In our generations, we performed an exhaustive sweep consisting of temperatures of $0.1, 0.3, 0.5, 0.7, 0.9, 1.0$, and 1.3, and top-k values of 2 and 4. All sampled sequences were then combined into a single file and used for downstream extraction and analysis of candidate CRISPR systems.
在生成序列时，我们使用指定的类别 token 进行提示。我们采用标准的基于 temperature 和 top-k 的自回归采样方法（Chang 和 Bergen, 2023）。我们进行了全面的采样测试，temperature 取值为 $0.1, 0.3, 0.5, 0.7, 0.9, 1.0$ 和 1.3，top-k 值为 2 和 4。所有生成的序列被合并到一个文件中，并用于后续候选 CRISPR 系统的提取和分析。

The in silico Cas evaluation pipeline consisted of an initial open reading frame (ORF) search using Prodigal (Hyatt et al., 2010) and subsequent profiling of the extracted ORFs using hidden markov model (HMM) profiles for each Cas subtype. Sampled sequences with a positive pHMM hit with an E-value under 1×10^{-3} and a sequence length above a given threshold were further analyzed using the MinCED package to identify possible CRISPR arrays (Bland et al., 2007). Generations containing both a Cas ORF and a CRISPR array were then clustered using MMSeqs2 at a sequence identity of 90% and minimum coverage length of 75% (Steinegger and Söding, 2017). Finally, representative sequences from the clustering analysis were aligned against Cas ORF sequences in the training data with MMSeqs2 to quantify divergence from the training dataset.
体外 Cas 评估流程包括使用 Prodigal（Hyatt 等，2010）进行开放阅读框（ORF）搜索，并使用各 Cas 亚型的隐马尔可夫模型（HMM）profile 对提取的 ORF 进行分析。采样序列中若存在 pHMM 匹配且 E-value 小于 1×10^{-3} 且序列长度超过阈值的，则进一步使用 MinCED 软件包识别可能的 CRISPR 阵列（Bland 等，2007）。同时包含 Cas ORF 和 CRISPR 阵列的生成序列使用 MMSeqs2 按照 90% 序列一致性和 75% 最小覆盖长度进行聚类（Steinegger 和 Söding，2017）。最终，聚类代表序列与训练集中的 Cas ORF 序列使用 MMSeqs2 进行比对，以评估与训练数据的差异性。

Candidate sequences were selected from the cluster representatives within various sequence identity deciles and processed using AlphaFold2 to manually inspect structural similarities between generations and a crystal structure

of wild-type SpCas9. Predicted Cas9 structures were aligned to SpCas9 (PDB: 4OO8) and its gRNA complex with ChimeraX and top candidates were chosen for further analysis. Possible recognition (REC) and nuclease (NUC) lobes of the sampled Cas9 structures were labeled by performing a multiple sequence alignment with the protein sequence of SpCas9. The MSA boundaries of the lobes in SpCas9 were used as boundaries for labeling the REC and NUC lobes in the sampled sequences. CRISPRtracrRNA was used to extract potential tracrRNA sequences from candidate generations and co-folded with the extracted crRNA sequence using RNAmultifold (Mitrofanov et al., 2022; Lorenz et al., 2011). Different combinations of tracrRNA and crRNA lengths were assessed as the resulting mature crRNA and tracrRNA sequences are not readily apparent from raw sequence data.
我们从不同序列一致性分位的聚类代表中选取候选序列，并使用 AlphaFold2 对其结构与野生型 SpCas9 晶体结构进行人工比对。预测得到的 Cas9 结构使用 ChimeraX 与 SpCas9（PDB: 4OO8）及其 gRNA 复合物进行比对，选出表现优异的候选结构进行进一步分析。我们通过与 SpCas9 蛋白序列的多序列比对，对生成的 Cas9 结构中的识别（REC）结构域和核酸酶（NUC）结构域进行标注。SpCas9 中这些结构域的 MSA 边界被用作生成序列中 REC 和 NUC 域的边界。使用 CRISPRtracrRNA 提取候选序列中的可能 tracrRNA 序列，并与提取的 crRNA 序列一起使用 RNAmultifold 进行共折叠（Mitrofanov 等，2022；Lorenz 等，2011）。我们评估了 tracrRNA 和 crRNA 的不同长度组合，因为成熟的 crRNA 和 tracrRNA 序列通常无法直接从原始序列中明确识别。

B.8. IS200/IS605 finetuning, generation, and downstream analysis
IS200/IS605 **微调、生成与后续分析**

To generate IS605 systems, we finetuned Evo by continuing to train the 8 k-context pretrained model on a dataset of IS200/IS605 sequences, which was curated as described above. We retained most of the hyperparameters used during pretraining but set the batch size to 524 k tokens and an initial learning rate of 0.00009698, which was the learning rate at the final step of pretraining. During pretraining, we prepended a generic start token to each sequence.
为了生成 IS605 系统，我们对 Evo 进行了微调，在 8k 上下文预训练模型的基础上继续训练上文所述的 IS200/IS605 序列数据集。我们保留了大部分预训练时使用的超参数，但将批大小设为 524k 个 token，初始学习率设为 0.00009698（即预训练最后一步的学习率）。在预训练中，我们在每个序列前添加了一个通用的起始 token。

We prompted the model with the start token for each sequence generation. We performed standard temperature-based and top-k autoregressive sampling (Chang and Bergen, 2023). In our generations, we performed an exhaustive sweep consisting of temperatures of $0.1, 0.3, 0.5, 0.7, 0.9, 1.0$, and 1.3, and top-k values of 2 and 4. We sampled a total of $1,004,850$ sequences.
我们在每次生成序列时用起始 token 提示模型。我们采用标准的基于 temperature 和 top-k 的自回归采样方法（Chang 和 Bergen，2023）。我们对 temperature（$0.1, 0.3, 0.5, 0.7, 0.9, 1.0, 1.3$）与 top-k（2 和 4）组合进行了全面测试，总共采样了 $1,004,850$ 条序列。

We analyzed generated sequences using prodigal to identify coding sequences and proteins (Hyatt et al., 2010), followed by hmmsearch (Z 1000000) using pHMMs to identify TnpA and TnpB sequences (Finn et al., 2011), and cmsearch (Z 4) using covariance models developed in a previous publication (Meers et al., 2023) to identify candidate ω RNAs (Nawrocki and Eddy, 2013). Candidate TnpA sequences were kept if they had an E-value $< 1 \times 10^{-3}$ to the pHMM and if they covered at least 50% of the pHMM. Candidate TnpB sequences were kept if they had an E-value $< 1 \times 10^{-3}$ to at least one pHMM, if they covered at least 50% of the pHMM, and if they were between 300 and 600 amino acids in length.
我们使用 prodigal 工具对生成的序列进行分析，以识别编码序列和蛋白质（Hyatt 等，2010），随后使用 hmmsearch（Z 1000000）结合 pHMM profile 识别 TnpA 和 TnpB 序列（Finn 等，2011），并使用 cmsearch（Z 4）结合先前研究构建的协方差模型识别候选 ω RNA（Meers 等，2023；Nawrocki 和 Eddy，2013）。TnpA 候选序列保留标准为：pHMM E-value 小于 1×10^{-3} 且覆盖度至少为 50%；TnpB 序列要求符合至少一个 pHMM E-value 小于 1×10^{-3}、覆盖度至少 50% 且氨基酸长度在 300 到 600 之间。

Predicted TnpA and TnpB protein sequences were aligned back to proteins in the training set using MMseqs2 (Steinegger and Söding, 2017). The top hit for each protein was extracted and separately aligned using the MAFFT-G-INS-I algorithm to estimate the amino acid identity across the full lengths of the two sequences (Katoh et al., 2002). To account for different start codons and to generate a more conservative percentage identity, these alignments were trimmed to the middle 80% of each sequence, end gaps were trimmed, and the amino acid identity was recalculated.
使用 MMseqs2 工具将预测得到的 TnpA 和 TnpB 蛋白序列与训练集中蛋白进行比对（Steinegger 和 Söding，2017）。我们提取每个蛋白的最佳命中序列，并使用 MAFFT-G-INS-I 算法对其进行比对，以估计完整序列之间的氨基酸相似度（Katoh 等，2002）。为了考虑不同起始密码子带来的偏差并得出更保守的相似度估计，我们将比对结果截断至每个序列的中间 80% 区段，同时修剪末端缺口，并重新计算氨基酸相似度。

For loci that contained both a TnpA and a TnpB coding sequence, we used ESMFold (Lin et al., 2023) to predict atomic-level structures for each protein sequence. We reported the mean backbone atom pLDDT as a measurement of ESMFold prediction confidence. Example TnpA and TnpB proteins were aligned to the 2EC2 and 8BF8 PDB structures, respectively, using the cealign algorithm in PyMOL (Schrödinger, LLC, 2015). RNAfold from the ViennaRNA package was used to fold the predicted ω RNA with parameters "d 3 P rna_langdon2018.par" (Gruber et al., 2008; Langdon et al., 2018).

对于同时包含 TnpA 和 TnpB 编码序列的位点，我们使用 ESMFold（Lin 等，2023）对每条蛋白序列进行原子级结构预测，并报告主链原子平均 pLDDT 值作为预测置信度指标。我们使用 PyMOL 中的 cealign 算法将 TnpA 和 TnpB 示例蛋白分别与 2EC2 和 8BF8 PDB 结构进行比对（Schrödinger, LLC, 2015）。使用 ViennaRNA 软件包中的 RNAfold 对预测的 ω RNA 进行结构折叠，参数为"d 3 P rna_langdon2018.par"（Gruber 等，2008；Langdon 等，2018）。

We also used Evo to calculate the entropy of the conditional probabilities at each position in a given sequence. For example, the entropy at position i was calculated using the likelihoods $p(x_i \mid x_1, \ldots, x_{i-1})$ over the entire vocabulary. We then visualized these entropies alongside the annotated sequence positions for several canonical IS200/IS605 systems.

我们还使用 Evo 计算每个序列位置的条件概率熵。例如，第 i 位的熵由其前缀 $x_1, \ldots, x_{i-1}$ 的条件概率 $p(x_i \mid x_1, \ldots, x_{i-1})$ 计算得到，并在整个词汇表上取总和。随后我们将这些熵值与多个典型 IS200/IS605 系统的注释序列位置一起进行可视化分析。

B.9. Gene essentiality prediction
基因必需性预测

We obtained binary genome-wide essentiality results for 56 bacterial genomes from the DEG database (Zhang, 2004) in which coding genes are labeled with "essential" or "nonessential" binary labels. We also obtained genome-wide essentiality results for two phage genomes, lambda and P1, from Piya et al. (2023) and used the binary labels assigned by the study authors based on the results of their CRISPRi screen.

我们从 DEG 数据库（Zhang，2004）中获取了 56 个细菌基因组的二元基因必需性标签，所有编码基因均被标注为" 必需" 或" 非必需"。我们还从 Piya 等人（2023）获取了两种噬菌体（lambda 和 P1）的全基因组必需性数据，并使用研究作者根据其 CRISPRi 筛选结果分配的标签。

To perform the in silico gene essentiality screen, we obtained the whole bacterial genome using the RefSeq IDs provided by DEG. We used RefSeq: NC_001416 as the reference genome for lambda phage and RefSeq: NC_005856 as the reference genome for P1 phage. We iterated over all genes annotated as protein coding and computed a score with a nucleotide language model for each gene.

为了进行基因必需性的计算机模拟预测，我们使用 DEG 提供的 RefSeq ID 获取了完整的细菌基因组。我们使用 RefSeq: NC_001416 作为 lambda 噬菌体参考基因组，RefSeq: NC_005856 作为 P1 噬菌体参考基因组。我们遍历所有注释为蛋白编码的基因，并使用核苷酸语言模型为每个基因计算一个评分。

To compute the score, we provided the language model with different levels of context: (1) the sequence of the gene only, (2) the sequence of the gene plus equally distributed context on both sides of the gene up to a total of 8 k tokens, or (3) the sequence of the gene plus equally distributed context on both sides of the gene up to a total of 65,536 tokens. If a gene extended beyond 8,192 bp, we used the first 8,192 bp of the gene sequences.

评分计算时，我们为语言模型提供了不同程度的上下文：(1) 仅提供基因序列；(2) 提供基因序列及其两侧等量的上下文，总长不超过 8k token；(3) 提供基因序列及其两侧等量的上下文，总长不超过 65,536 token。如果某基因长度超过 8,192 bp，则仅使用其前 8,192 bp。

We computed the score as the difference in log-likelihoods between a mutated sequence and the unmutated wildtype sequence. To mutate the sequence, we inserted multiple stop codons "TAATAATAATAGTGA" 12 nucleotides into the sequence; for the 8,192 and 65,536 bp context settings, we add context to both sides of the gene after the insertion.

评分定义为突变序列与野生型序列在语言模型下的对数似然差值。突变方式是将多个终止密码子"TAATAATAATAGTGA"插入基因的第 12 个核苷酸处；在 8,192 和 65,536 bp 上下文设置下，插入后对两侧补充上下文。

Additionally, for the 8,192 bp setting, we tested two other strategies: (1) inserting a single stop codon "TAA" 12 nucleotides into the sequence and (2) deleting the entire gene sequence (after which we provided 8,192 context centered on the deleted gene) (Figure S9). As an additional control, we also used the gene's linear position in the reference genome as the value with which to predict essentiality.

此外，在 8,192 bp 设置下，我们还测试了两种替代策略：(1) 在第 12 个核苷酸处插入单个终止密码子"TAA"；(2) 删除整个基因序列，并以其中心提供 8,192 bp 上下文（详见图 S9）。作为附加对照，我们还使用基因在线性参考基因组中的位置来预测其必需性。

We used the change in log-likelihoods to predict the binary gene essentiality labels and compute the strength of the prediction with the AUROC score. We assessed the statistical significance of the AUROC with a permutation-based method in which a null distribution is constructed by permuting the binary labels and recomputing the subsequence AUROC. We performed 100,000 permutations to construct this null distribution.

我们利用对数似然变化预测基因的二分类必需性标签，并用 AUROC 评分衡量预测强度。我们采用基于置换的检验方法评估 AUROC 的统计显著性：构造一个通过随机打乱标签并重复计算子 AUROC 所得到的零分布。该置换过程重复 100,000 次。

B.10. Genome-scale generation and evaluation **基因组尺度的生成与评估**

We used Evo pretrained at 131 k context to sample twenty sequences up to lengths $\sim$ 650 kb. We sampled with a temperature of 0.7 and a top-k value of 4 following a standard autoregressive sampling procedure (Chang and

Bergen, 2023). We prompted the model with four species-specific prompts, which were introduced during 131k pretraining, corresponding to the bacterial species Mycoplasma genitalium, Staphylococcus aureus, Klebsiella pneumoniae, and Escherichia coli. These prompts follow Greengenes-style lineage strings, which concatenate all taxa starting with the most ancestral and ending with the most current, separated by semicolons. A single character prefix is also added to each taxon indicating its rank. We sampled five sequences for each prompt, leading to a total of twenty sequences.
我们使用在 131k 上下文预训练的 Evo 模型采样了二十条序列，每条序列的长度约为 ~ 650 kb。我们采用标准的自回归采样方法（Chang 和 Bergen，2023），设置采样温度为 0.7，top-k 值为 4。我们用四个物种特定的提示词对模型进行提示，这些提示词是在 131k 预训练阶段引入的，对应的细菌物种分别为：生殖支原体（Mycoplasma genitalium）、金黄色葡萄球菌（Staphylococcus aureus）、肺炎克雷伯菌（Klebsiella pneumoniae）和大肠杆菌（Escherichia coli）。这些提示词采用 Greengenes 风格的谱系字符串形式，即将所有分类单元从最早祖先依次连接到当前物种，中间用分号隔开，并为每个分类单元添加表示其等级的单字符前缀。每个提示生成五条序列，共生成二十条序列。

We evaluated these generations with CheckM (Parks et al., 2015), a tool that computes basic genome quality metrics based on whether a given long DNA sequence has similar properties as known bacterial genomes. CheckM uses Prodigal (Hyatt et al., 2010) to identify coding sequences and computes the coding density as one metric of genome quality. CheckM will also search for the presence of genes that are highly conserved across much of prokaryotic diversity. We divided all of our generations into five discrete segments of up to 131,072 bp (a total of 100 sequences) and computed the distribution of CheckM coding densities across these crops. As a positive control, we randomly selected 100 bacterial genomes from GTDB and used CheckM to compute the coding densities for 131,072 bp crops from these genomes. As a negative control, we generated 1,000 sequences of length 131,072 in which the four DNA base pairs were sampled uniformly at random. We then used CheckM to compute the coding densities on this random sequence. We also used tRNAscan-SE to search for tRNA sequences in our generated sequences and we used barrnap to search for rRNA sequences.
我们使用 CheckM 工具（Parks 等，2015）对这些生成的序列进行了评估。CheckM 是一个用于评估长 DNA 序列是否具备与已知细菌基因组相似特征的基因组质量评估工具。CheckM 使用 Prodigal（Hyatt 等，2010）识别编码序列，并以编码密度作为衡量基因组质量的指标之一。CheckM 还会检测在多数原核生物中高度保守的基因是否存在。我们将所有生成序列分成最多 131,072 bp 的五个独立片段（共计 100 条序列），并计算这些片段的编码密度分布。作为阳性对照，我们从 GTDB 中随机选择了 100 个细菌基因组，并用 CheckM 计算这些基因组中 131,072 bp 片段的编码密度。作为阴性对照，我们随机生成了 1,000 条长度为 131,072 的序列，其碱基由四种 DNA 碱基均匀采样生成，并使用 CheckM 对这些随机序列计算编码密度。此外，我们使用 tRNAscan-SE 检测生成序列中的 tRNA 序列，使用 barrnap 检测 rRNA 序列。

We used ESMFold to obtain atomic-level structure predictions for all of the Prodigal-defined coding sequences in each of our generations. We limited ESMFold structure predictions to coding sequences between 100 and 1024 amino acids, inclusive. We computed the mean backbone pLDDT for all predicted structures. We used the biotite Python package to compute the percentages of secondary structure elements for all predicted structures. We used FoldSeek easy-search to perform efficient TM-based alignment (–alignment-type 1), and all other parameters set to their default values, to perform an all-by-all structural search between ESMFold structures corresponding to Evo-generated sequences and the structure predictions for UniRef50 provided in the AlphaFold Protein Structure Database (https://alphafold.ebi.ac.uk/). Structure alignments were scored as the average of the query TMscore and the target TMscore, where a score greater than 0.4 was considered a structural match. We used these structural matches, along with GO terms assigned to UniRef50 clusters, to infer GO terms for the Evo-generated proteins as well. We used PyMOL to visualize protein structures corresponding to the five GO "molecular function" terms with the most representation among the Evo generated proteins.
我们使用 ESMFold 对每个生成序列中 Prodigal 识别出的编码序列进行原子级结构预测。我们仅对长度在 100 至 1024 个氨基酸之间（包含边界值）的编码序列进行结构预测，并计算所有预测结构的主链平均 pLDDT 分数。我们使用 biotite Python 包来计算所有预测结构的二级结构元素占比。使用 FoldSeek 的 easy-search 功能（–alignment-type 1，其余参数使用默认值），我们在由 Evo 生成序列对应的 ESMFold 结构与 AlphaFold 蛋白结构数据库（https://alphafold.ebi.ac.uk/）中提供的 UniRef50 结构预测之间进行了结构比对。结构比对的评分为查询序列 TMscore 与目标序列 TMscore 的平均值，得分大于 0.4 被视为结构匹配。我们基于这些结构匹配结果以及分配给 UniRef50 聚类的 GO 术语，为 Evo 生成的蛋白质推断出对应的 GO 注释。最后，我们使用 PyMOL 可视化了五个在 Evo 生成蛋白中出现频率最高的 GO" 分子功能" 术语所对应的蛋白质结构。

Supplementary tables and figures

PARAMS (M)	D_MODEL	GLU_SIZE	KV_SIZE	N_HEADS	N_LAYERS	LEARNING RATE
1	128	336	64	2	4	9.77E-04
6	320	848	64	5	5	9.57E-04
17	448	1200	64	7	7	9.36E-04
29	512	1360	64	8	9	9.15E-04
40	576	1536	64	8	10	8.95E-04
59	640	1696	64	10	12	8.70E-04
69	640	1712	64	10	14	8.56E-04
84	704	1872	64	11	14	8.37E-04
99	768	2048	64	12	14	8.18E-04
114	768	2048	64	12	16	8.00E-04
121	768	2048	64	12	17	7.75E-04
135	768	2048	64	12	19	7.50E-04
158	832	2224	64	13	19	7.25E-04
175	832	2224	64	13	21	7.00E-04
203	896	2384	64	14	21	6.75E-04
232	896	2384	64	14	24	6.50E-04
266	960	2560	64	15	24	6.25E-04
303	1024	2736	64	16	24	6.00E-04
383	1152	3072	64	18	24	5.66E-04
473	1280	3408	64	20	24	5.33E-04
572	1408	3760	128	11	24	5.00E-04
680	1536	4096	128	12	24	4.75E-04
798	1664	4432	128	13	24	4.55E-04
926	1792	4784	128	14	24	4.33E-04
1063	1920	5120	128	15	24	4.15E-04
1209	1920	5120	128	15	25	4.11E-04

Table 1: Scaling laws model settings for Transformer++, Hyena, StripedHyena and Mamba. Layer number for Mamba is doubled (a single block corresponds to two Mamba layers and the dedicated channel mixer layer is removed, as described in (Gu and Dao, 2023)). Parameter counts vary slightly for each architecture.
Transformer++、Hyena、StripedHyena 和 Mamba 的 scaling laws 模型设置。Mamba 的层数加倍（一个 block 对应两个 Mamba 层，并移除了专用的通道混合器层，如 Gu 和 Dao（2023）中所述）。由于架构差异，每种模型的参数数量略有不同。

Noise level when training: 0.00 / 0.02	Modification	Number of Parameters in millions	PDB Test Accuracy	PDB Test Perplexity	AlphaFold Model Accuracy
Baseline model	None	1.381	41.2/40.1	6.51/6.77	41.4/41.4
Experiment 1	Add N, C α, C, Cβ, O distances	1.430	49.0/46.1	5.03/5.54	45.7/47.4
Experiment 2	Update encoder edges	1.629	43.1/42.0	6.12/6.37	43.3/43.0
Experiment 3	Combine 1 and 2	1.678	50.5/47.3	4.82/5.36	46.3/47.9
Experiment 4	Experiment 3 with random instead of forward decoding	1.678	50.8/47.9	4.74/5.25	46.9/48.5

Table 2: Single chain sequence design performance on CATH held out test split. Test accuracy (percentage of correct amino amino acids recovered) and test perplexity (exponentiated categorical cross entropy loss per residue) for models trained on the native backbone coordinates (left, normal font) and models trained with Gaussian noise (std=0.02) added to the backbone coordinates (right, bold font). Noise was only added during training and all test evaluations are with no added noise. The final column shows sequence recovery on 5,000 AlphaFold protein backbone models with average pLDDT > 80.0 randomly chosen from UniRef50 sequences.
在 CATH 保留测试集上进行单链序列设计性能评估。模型分别在天然主链坐标（左侧，正常字体）和添加了高斯噪声（标准差为 0.02 ）的主链坐标（右侧，加粗字体）上进行训练，并报告测试准确率（即正确氨基酸的百分比）和测试困惑度（每个残基的指数化分类交叉熵损失）。噪声仅在训练时添加，所有测试评估均未添加噪声。最后一列展示了在从 UniRef50 序列中随机选取、平均 pLDDT > 80.0 的 5,000 个 AlphaFold 蛋白质主链模型上的序列恢复率。

Dataset Name	Source	Subset	Total Genomes/Loci /Plasmids	Total Bases (M)	Avg Length (base)
Bacterial and Archaeal Genomes	GTDB		85,205	273,865	3, 214, 184
Prokaryotic Viruses	IMG/VR		2,653,046	36,236	13,658
Plasmids	IMG/PR		214,950	5,827	27,106
CRISPR/Cas Loci	Custom Database	Cas9	5,566	43	7,798
CRISPR/Cas Loci	Custom Database	Cas12	5,069	35	6,911
CRISPR/Cas Loci	Custom Database	Cas13	498	4	7,559
IS200/IS605 Loci	Custom Database	IS200 Loci	219,866	239	1,085
IS200/IS605 Loci	Custom Database	IS605 Loci	10,720	26	2,445

Table 3: Summary statistics for the OpenGenome datasets. See B.2 for further details on the dataset sources and curating process.
OpenGenome 数据集的汇总统计信息。有关数据集来源和整理流程的更多详细信息，请参见 B.2 节。

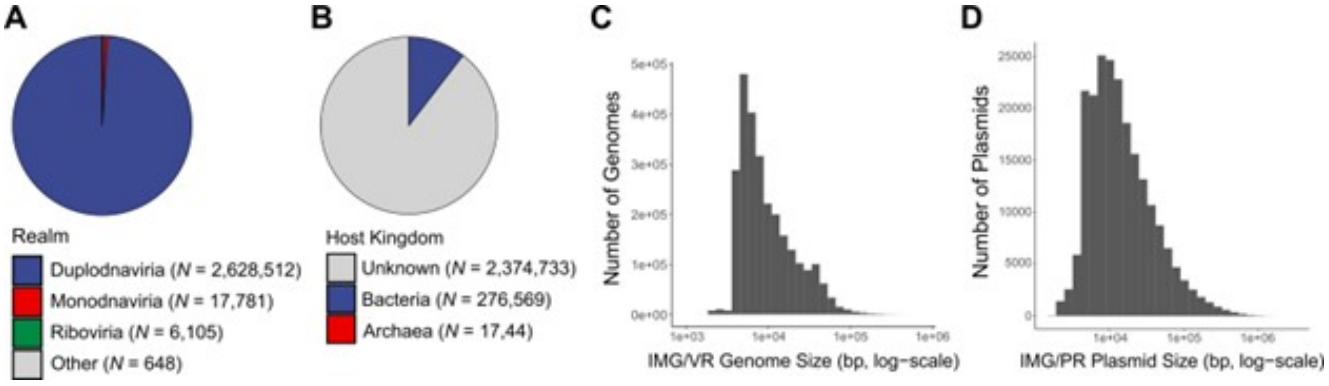

Figure 6: Pretraining data. Statistics of IMG/VR and IMG/PR. (A) A pie chart depicting the composition of viral realms in the IMG/VR subset of the pretraining dataset. (B) A pie chart depicting the composition of host kingdoms in the IMG/VR subset of the pretraining dataset. We excluded viruses that are likely to infect eukaryotic hosts (Methods). (C) The distribution of sequence lengths in the IMG/VR subset of the pretraining dataset. (D) The distribution of sequence lengths in the IMG/PR subset of the pretraining dataset.

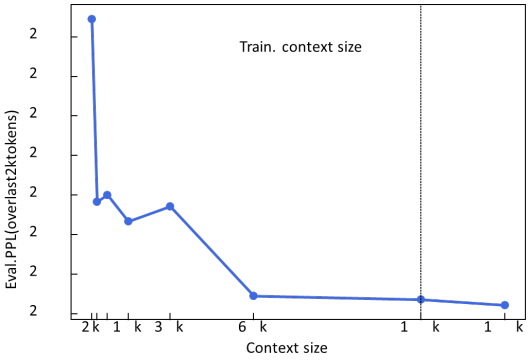

Figure 7: Perplexity scaling in context length. Perplexity on a subset of the OpenGenome validation set with Evo 131k as a function of sequence length, or context length. The perplexity is computed over the last 2048 nucleotides of each sequence, with increasing lengths of the prefix and thus of the context available to the model. We observe perplexity to continually decrease beyond the training context length at 131k, indicated by the vertical dashed line.

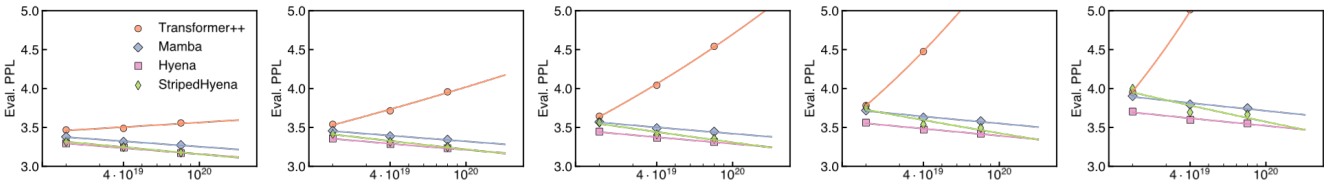

Figure 8: Scaling rates for compute-suboptimal model sizes by architecture. We quantify the effect on perplexity scaling caused by a suboptimal allocation of compute budget to model or dataset size (e.g., training a smaller model for more tokens). We estimate the compute-optimal model size (Figure S5) for each compute budget, then reduce it by a percentage (the offset). The corresponding perplexity is obtained via the IsoFLOP curves (Figure 1F). Transformer++ perplexity scaling rapidly degrades outside the compute-optimal frontier, in contrast to Hyena and StripedHyena.

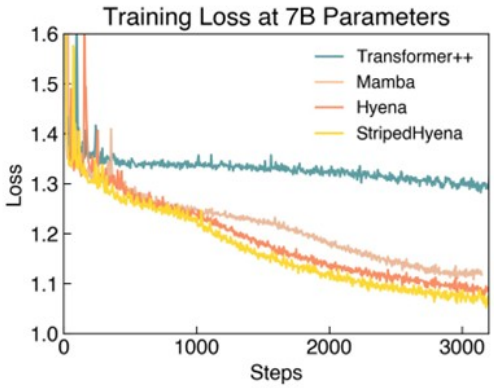

Figure 9: Direct comparison of training loss curves during hyperparameter tuning for 7B models. We tune several Transformer++ and Mamba models as baselines, sweeping learning rates, batch size, sequence lengths, and model depth vs. width ratio. In all settings, Hyena and StripedHyena outperformed Transformer++ and Mamba, where both baselines experienced instability during training.

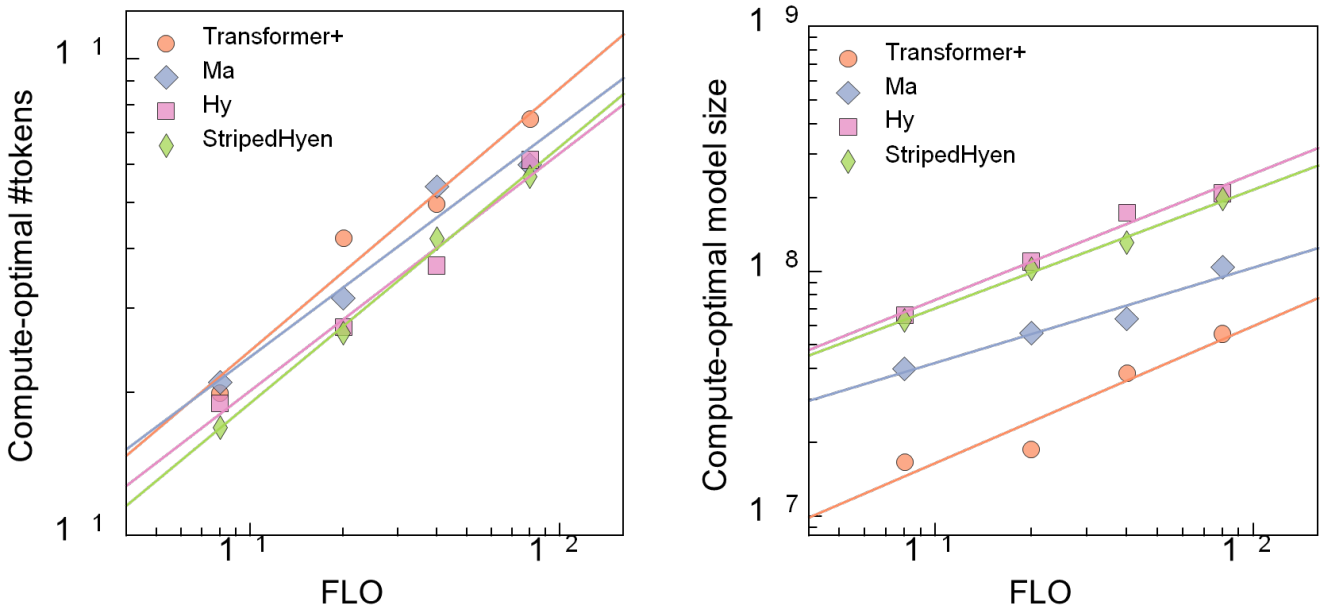

Figure 10: Compute-optimal tokens and model size by model. Compute-optimal allocation to dataset size (number of tokens) and model size (number of parameters) for each compute budget, measured in FLOPS.

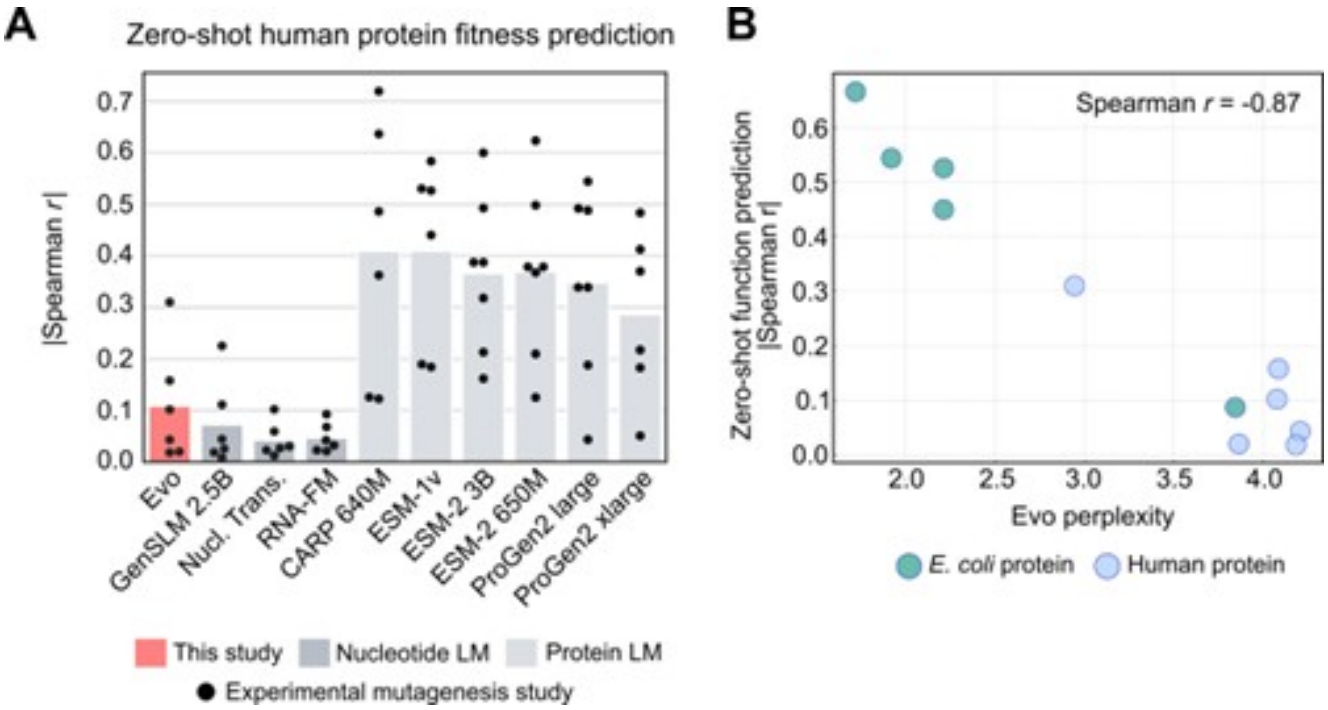

Figure 11: Performance of Evo on mutational effect prediction for human proteins. (A) Predictive performance of nucleotide and protein language models on mutational effect prediction for human proteins, measured via Spearman correlation. Bar height indicates the mean; each dot indicates a different DMS study. LM: language model; Nucl. Trans.: Nucleotide Transformer. Related to Figure 2B. (B) Relationship between the Evo perplexity of the wildtype nucleotide sequence (horizontal axis) and the ability for Evo to perform zero-shot mutational effect prediction for that protein as measured via Spearman correlation (vertical axis). Each dot corresponds to a different protein; dots are colored as to whether they are E. coli (teal) or human (blue) proteins. We observed a strong negative correlation (Spearman $R = -0.87$) between perplexity and zero-shot function prediction performance.

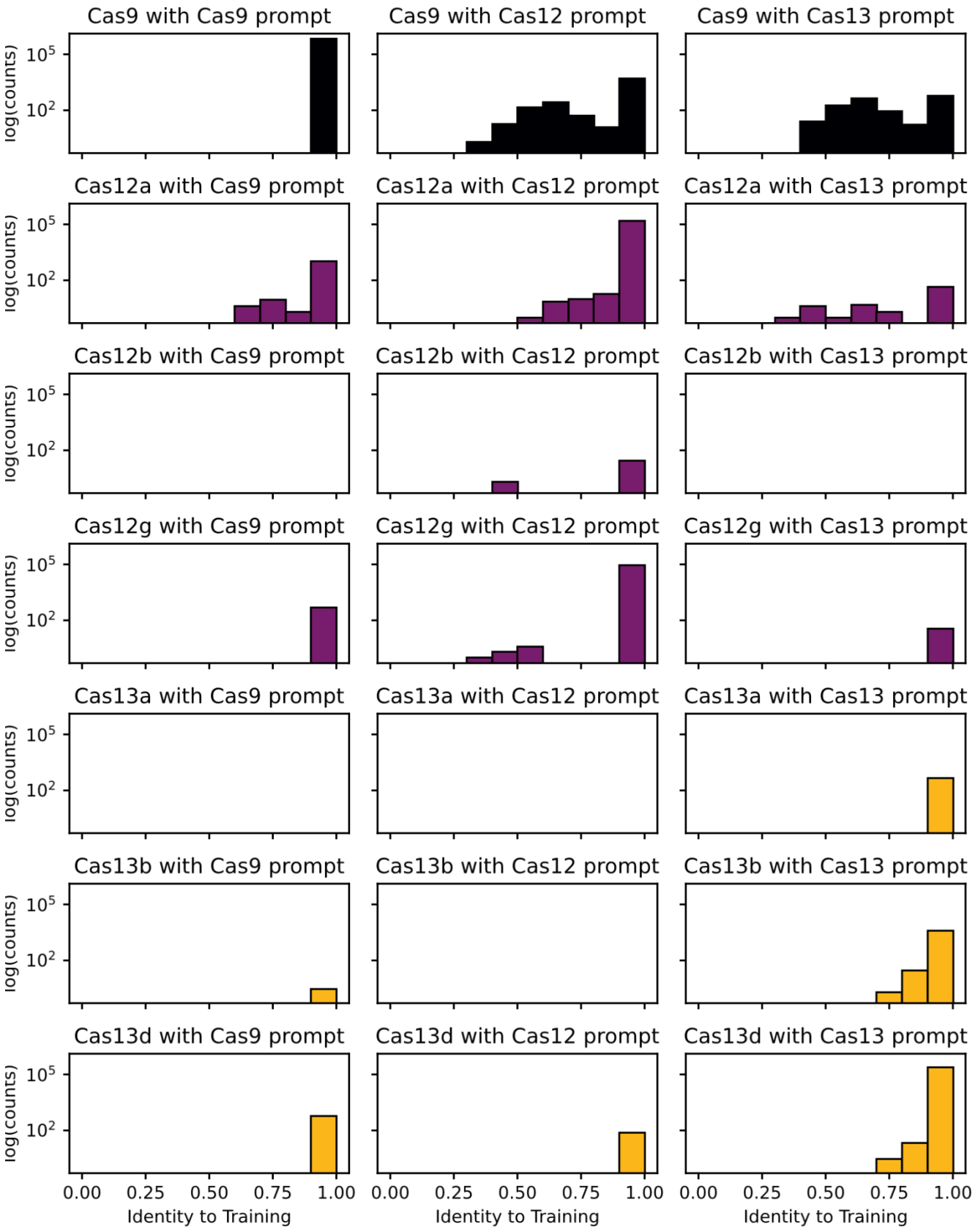

Figure 12: Breakdown of Cas generation diversity by prompt token. Evo can generate diverse Cas loci across subtypes through conventional prompting as well as through cross-type prompting. Evo was prompted with ”cas9”, ”cas12”, or ”cas13” special tokens and each resulting set of generations was analyzed using pHMMs for each Class 2 Cas subtype and compared against training data to observe divergence from the training dataset.

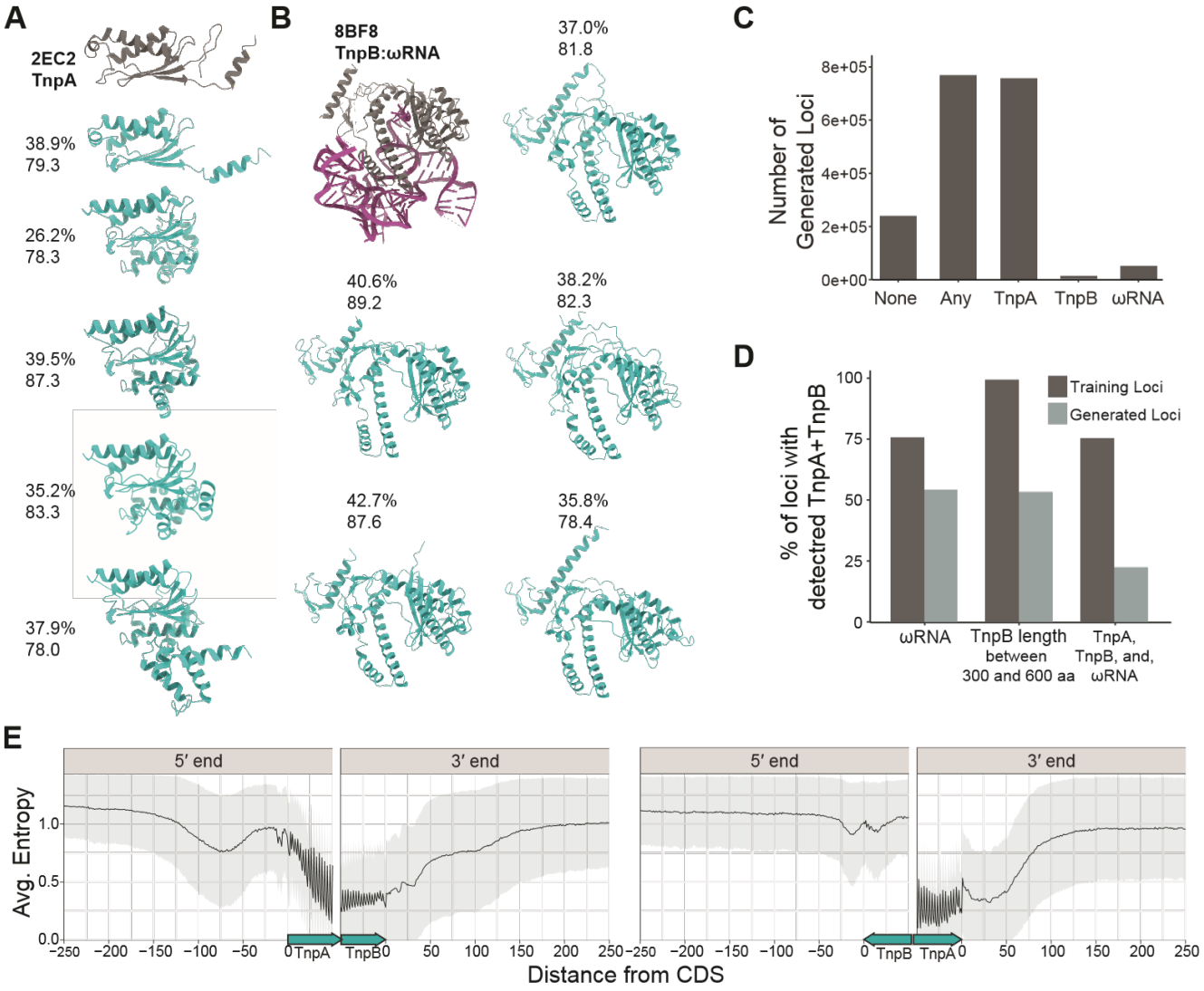

Figure 13: Additional analysis of generated IS200/IS605 sequences and the finetuned model. (A) Additional examples of diverse TnpA-like proteins with detected Y1 HuH domains and high mean backbone atom pLDDT. Labels indicate the percent amino acid identity to closest protein hit found in training set or NCBI nr and pLDDT. PDB structure 2EC2 shown for reference. (B) Examples of diverse generated TnpB proteins. PDB structure 8BF8 shown for reference. (C) A summary of the 1,004,850 sequences generated using the model finetuned on IS200/IS605 loci. Sequences have a detected TnpA, TnpB, or ω RNA ("Any"), a TnpA coding sequence ("TnpA"), a TnpB coding sequence ("TnpB"), or a ω RNA ("ω RNA"). (D) A comparison of the percentage in each category across the training set and generated sequences for sequences with a detected TnpA and TnpB coding sequence. Categories include sequences with a detected ω RNA ("ω RNA"); sequences encoding a TnpB protein between 300 and 600 aa in length; and sequences with a TnpA, a TnpB protein between 300 and 600 aa in length, and an ω RNA. (E) The average entropy within 250 nt of the 5 and 3 ends of IS605 coding sequences, including 50 nt of the CDS itself. The entropy was calculated at each position across IS605 sequences in the training set (ω = 10,419). Sequence positions were aligned with respect to the beginning and end of each respective CDS. (Left) All sequences with a TnpA followed by a TnpB. (Right) Sequences where the TnpB precedes the TnpA on the forward strand. Gray ribbon indicates the standard deviation of the entropy values.

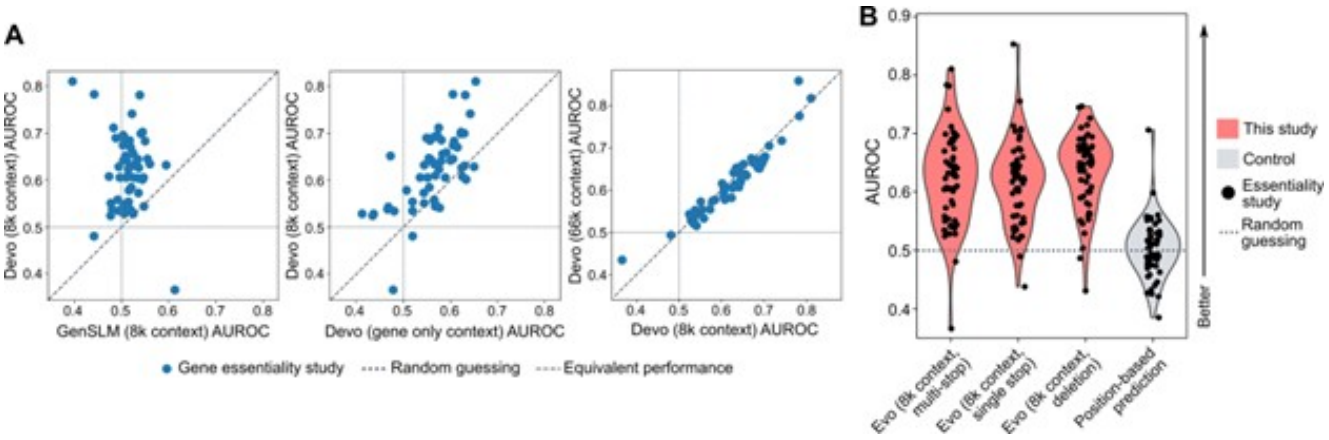

Figure 14: Gene essentiality prediction under different settings. (A) Scatterplots that compare the AUROC values between different models or different context windows. Axis labels are the same as the horizontal labels in Figure 5C. Each dot corresponds to a different whole-genome essentiality study. Related to Figure 5C. (B) Gene essentiality prediction performance for across 58 studies (each dot corresponds to a different study). We performed in silico mutagenesis of each coding sequence in a genome and commuted the change in Evo likelihood, which we used to predict gene essentiality. "Evo (8k context, multi-stop)" indicates a mutagenesis strategy that inserts multiple stop codons at the beginning of each coding sequence. "Evo (8k context, single stop)" indicates a mutagenesis that inserts a single stop codon at the beginning of each coding sequence. "Evo (8k context, deletion)" indicates a mutagenesis strategy that deletes the entire sequence of the gene. "Positionbased prediction" indicates a prediction strategy (not using Evo) in which we use the position of a gene in the reference genome annotation as the predictor variable for gene essentiality. See Methods for more details.

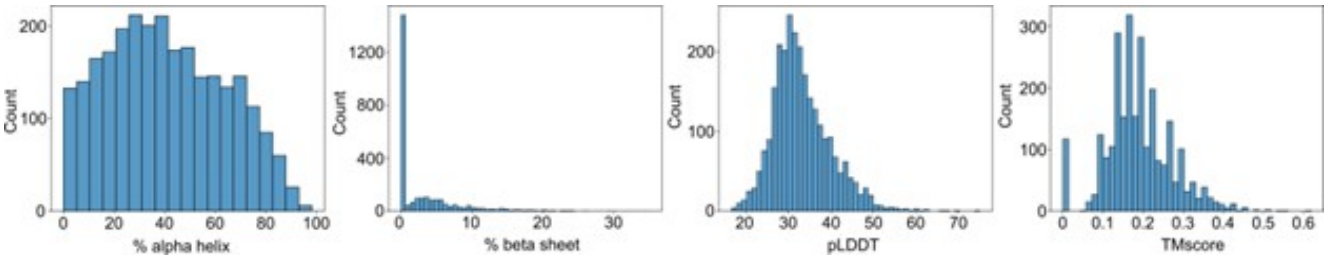

Figure 15: Statistics for ESMFold structure predictions of Evo-generated protein coding sequences. Histograms representing the distribution of statistics computed on ESMFold-predicted structures. These structures correspond to coding sequences found on five Evo-generated sequences, each of length 650 kb. These statistics are, from left to right: the percentage of residues in alpha helices, the percentage of residues in beta sheets, the mean backbone pLDDT, and the TMscore to the closest UniRef50 structure in the AlphaFold Protein Structure Database as determined by FoldSeek easy-search.