Computational Molecular Biology and Bioinformatics

Evo 2

Malay Bhattacharyya

Associate Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

October, 2025

1 Introduction

2 The Model

3 References

Foundation models

A foundation model is a large-scale AI model that serves as a foundation for the building of many specialized models. For example, GPT, BERT, CLIP, etc. are foundation models.

Foundation models learn general-purpose representations from diverse and massive amounts of data (often using self-supervised or weakly supervised learning) and can then be adapted or fine-tuned for a wide range of downstream tasks.

What is Evo 2?

Evo 2 is a state-of-the-art biological foundation model (precisely a DNA language model) for long context modeling and design [1]. It models DNA sequences at single-nucleotide resolution at up to 1 million base pair context length.

Evo 2 was trained autoregressively using Savanna [2] on OpenGenome2 [1], a dataset containing nearly 9 trillion base pairs of curated genomic atlas across all domains of life.

Evo 2 autonomously learns a breadth of biological features, including exon–intron boundaries, transcription factor binding sites, protein structural elements, and prophage genomic regions.

The OpenGenome2 dataset

OpenGenome2 includes nearly 9 trillion nucleotides [1], which is an increase of 33% from the OpenGenome dataset used in Evo [2].

This includes the following:

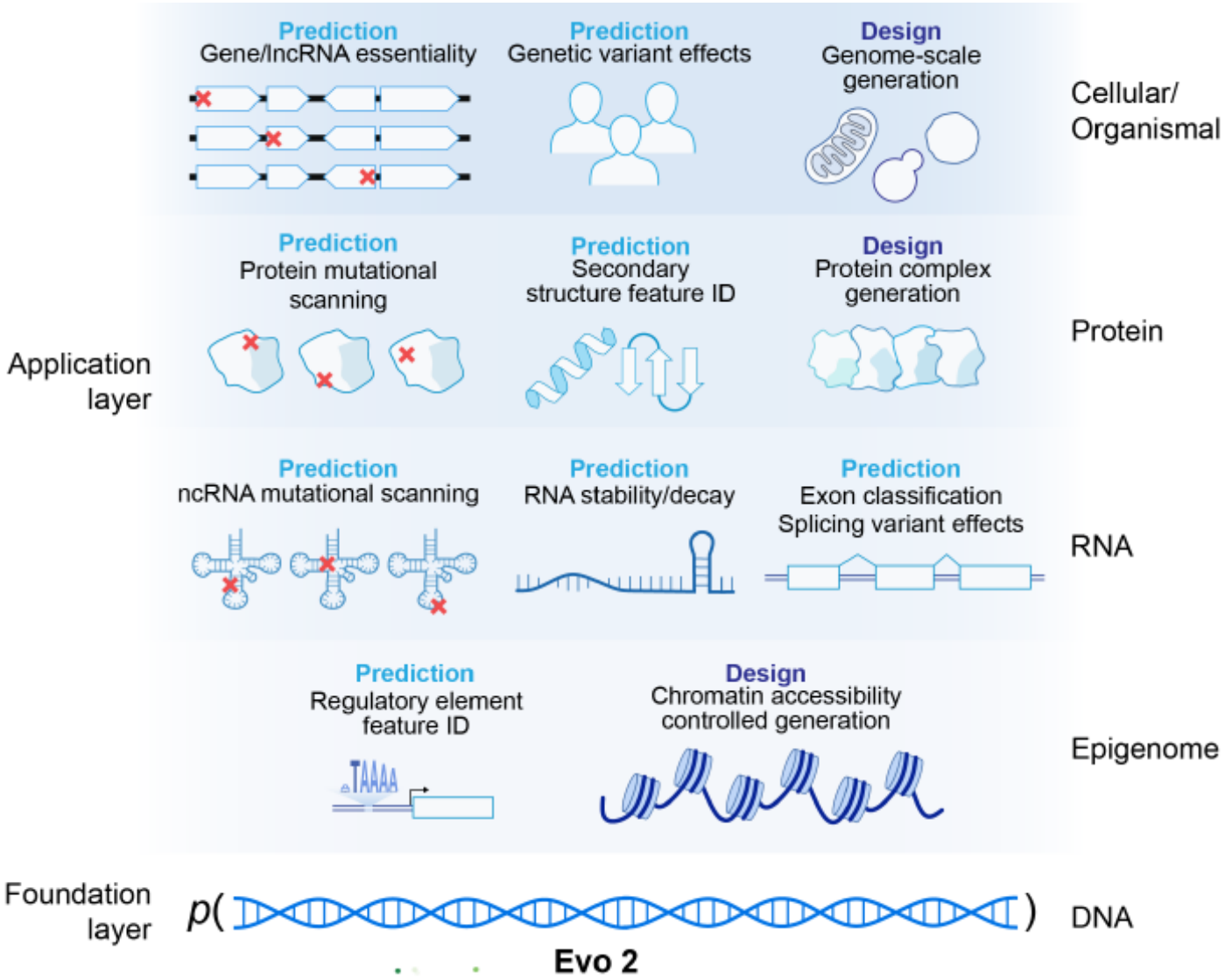
- 6.98 trillion nucleotides from eukaryotic genomes,
- 854 billion nucleotides of non-redundant metagenomic sequencing data,
- 2.82 billion nucleotides of organelle genomes, and
- 602 billion nucleotides of subsets of eukaryotic sequence data to focus on likely functional regions of the genomes by focusing on different windows around coding genes.

Data curation

The following sources were used to curate the database:

- Reused datasets like OpenGenome
- Updated prokaryotic genomes available through the GTDB release 220.0
- Eukaryotic reference genomes available from NCBI
- Metagenomes available from NCBI, JGI IMG, MGnify, MG-RAST, Tara Oceans samples, and Youngblut et al. animal gut metagenomes
- Eukaryotic organelle genomes available from NCBI
- mRNA and ncRNA transcripts available from NCBI
- Noncoding RNAs available from Ensembl (release 112), Rfam, and RNAcentral
- Eukaryotic promoters available from EPDnew

Applications of Evo 2



Training Evo 2

Evo 2 is trained using next-token-prediction on the byte-tokenized OpenGenome2 dataset. Two versions of Evo 2 were trained:

- A smaller model with 7 billion (7B) parameters trained on 2.4 trillion tokens.
- A full model with 40 billion (40B) parameters trained on 9.3 trillion tokens.

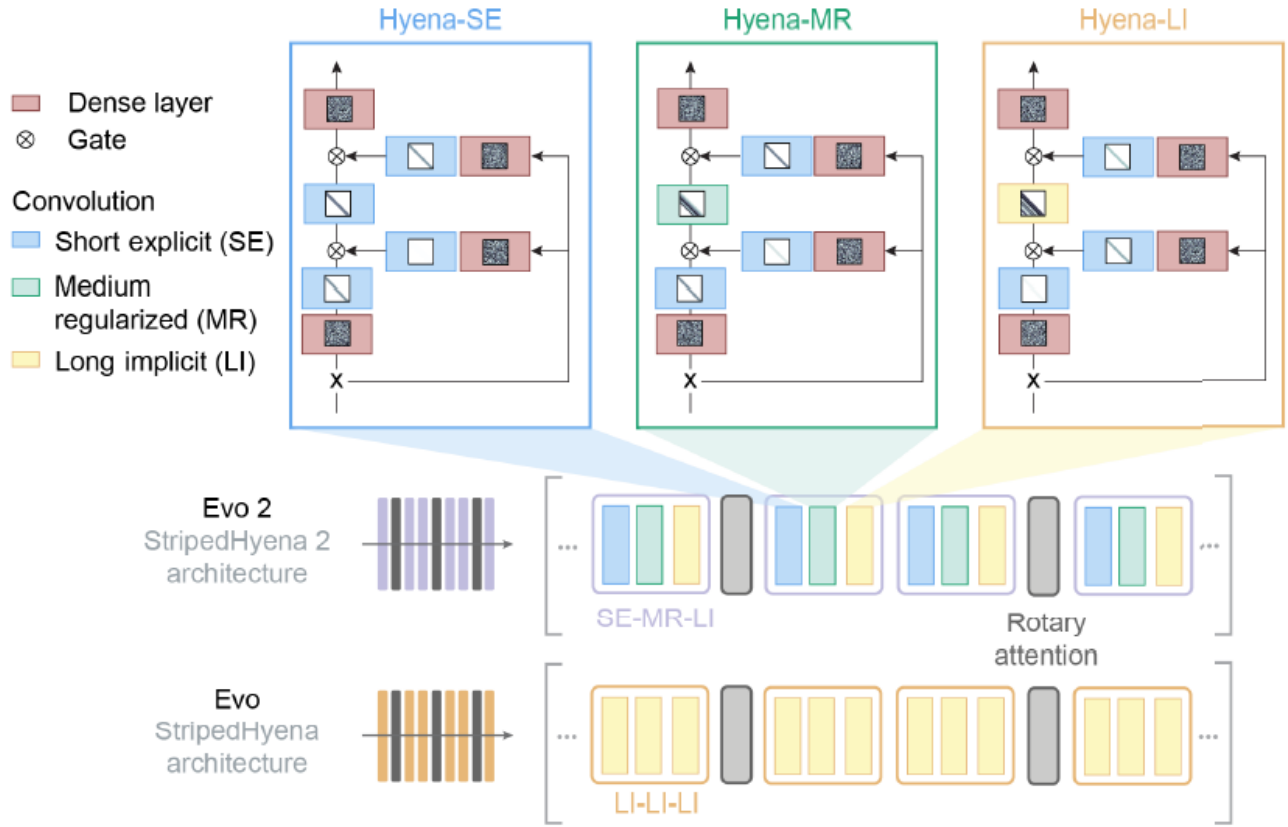
Evo 2 is trained in two phases:

1. a pretraining phase at 8192 token context focused more on functional elements
2. a midtraining phase during which we extend up to 1M token context length with more entire genomes in the data mix.

Evo 2 40B's pretraining stage is further split into two stages, first training at 1024 context for 6.6T tokens before extending to 8192 context for 1.1T tokens. For efficiency, Evo 2 is trained using sequence packing.

Model architecture

Evo 2 uses StripedHyena 2 [3], the first multi-hybrid architecture based on input-dependent convolutions [4, 5]. Multi-hybrids combine various different operators to balance model quality with training and inference efficiency, in line with earlier findings.



Loss function

Evo 2 is trained with a reweighted cross entropy loss, which weighs the loss contribution of repetitive portions of DNA by 0.1. This affects the genomic window and whole genome portions of the data which contain these annotations. This loss has been found in other DNA models to improve performance on downstream tasks and better calibrate likelihoods between repetitive and nonrepetitive DNA.

The loss is given by:

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

where $w_t = 0.1$ if position t is in repetitive region, 1.0 otherwise, and $Z = 0.1 * N_{repeat} + 1.0 * N_{non\_repeat}$.

Pretraining infrastructure

Evo 2 was trained on Savanna (see Section 6), custom training infrastructure built with components from DeepSpeed, GPT-NeoX (Andonian et al., 2023), and Transformer Engine. The stack supports efficient pretraining of multi-hybrid models and new context parallel algorithms.

The largest models were trained with mixed precision, using a 3D mesh of data, tensor, and context parallelism, combined with ZeRO-3 (Rajbhandari et al., 2020). During training, Transformer Engine's FP8 implementation was used for linear layers and RMSNorms.

Pretraining phase

For Evo 2 40B base model:

- Learning Rate = 2.0e-4
- Training Batch = 16.8M
- Total Iterations = 516K
- Pretraining Tokens = 8.7T
- Sequence length = 1024 (6.6T), 8192 (1.1T)

For Evo 2 7B base model:

- Learning Rate = 3.0e-4
- Training Batch = 4.2M
- Total Iterations = 500K
- Pretraining Tokens = 2.1T
- Sequence length = 8192

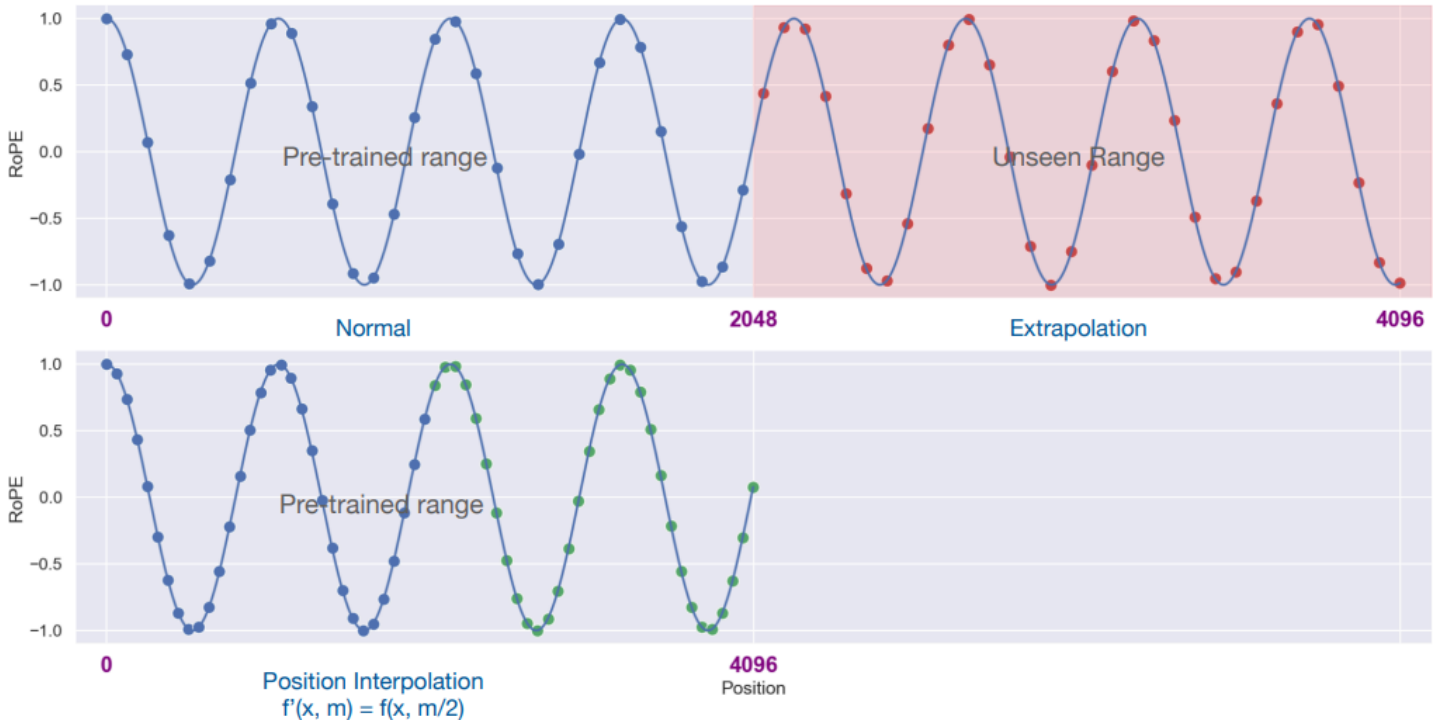
Midtraining phase – Context extension

A multi-stage midtraining procedure was followed, gradually extending the context length while keeping the same batch size as pretraining, adjusting model parallelism accordingly. Midtraining was performed on an adjusted data composition, including more whole genomes and with longer average sequence length.

Two different rotary embedding-based methods were applied to adapt to longer sequences: RoPE positional interpolation by downscaling the positional index of tokens [6] and increasing the base frequency of the RoPE embedding [7].

Midtraining phase – Context extension

Instead of performing length extrapolation where models are required to operate on unseen positions (red dots) up to 4096 context window length, the positional interpolation downscales the position indices (blue and green dots) themselves from [0, 4096] to [0, 2048] to force them to reside in the pretrained range.



Midtraining phase – Context extension

To extend the sequence length of a trained transformer, Position Interpolation (PI) parameterized with α, and Adjusted Base Frequency (ABF) parameterized with β correspond to the following embedding curves:

$$f^{RoPE+PI}(x,t)_j = (x_{2j} + ix_{2j+1})e^{i\alpha(b^{-\frac{2j}{d}})t},$$

and

$$f^{RoPE+ABF}(x,t)_j = (x_{2j} + ix_{2j+1})e^{i(\beta b)^{-\frac{2j}{d}}t}.$$

Note: The purpose of RoPE mapping is to help the attention module to separate the vectors corresponding to two instances of the same token that are situated at different positions in the input sequence.

Midtraining phase – Needle-in-a-haystack evaluation

We developed a novel synthetic evaluation to assess the ability of DNA languagemodels to identify and utilize a specific sequence pattern in its context to make predictions on a repeated sequence with the same pattern. This "needle-in-haystack" evaluation quantifies a model's capacity to retrieve sequence patterns within different context lengths.

The evaluation methodology employs a modification of the "categorical Jacobian" analysis, as originally proposed recently [8], to measure the model's use of the needle sequence to predict the query sequence.

Inference infrastructure

Evo 2 inference runs on Vortex that contains infrastructure and efficient implementation for autoregressive generation with StripedHyena 2. For the new convolution operators with finite inner filters, a caching strategy similar to KV caching in self-attention was adopted.

For long filters, it was switched to a recurrent form. All convolution operators in the architecture can generate autoregressively with a constant memory footprint.

References

1. Brixi, G., Durrant, M.G., Ku, J., Poli, M., Brockman, G., Chang, D., Gonzalez, G.A., King, S.H., Li, D.B., Merchant, A.T. and Naghipourfar, M., Genome modeling and design across all domains of life with Evo 2. bioRxiv, 2025.02.18.638918, 2025.
2. Nguyen, E., Poli, M., Durrant, M.G., Kang, B., Katrekar, D., Li, D.B., Bartie, L.J., Thomas, A.W., King, S.H., Brixi, G. and Sullivan, J., Sequence modeling and design from molecular to genome scale with Evo. Science, 386(6723):eado9336, 2024.
3. Ku, J., Nguyen, E., Romero, D.W., Brixi, G., Yang, B., Vorontsov, A., Taghibakhshi, A., Lu, A.X., Burke, D.P., Brockman, G. and Massaroli, S., Systems and algorithms for convolutional multi-hybrid language models at scale. arXiv preprint arXiv:2503.01868, 2025.

References

4. Poli, M., Massaroli, S., Nguyen, E., Fu, D.Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S. and Ré, C., 2023 Hyena hierarchy: Towards larger convolutional language models. In International Conference on Machine Learning, pp. 28043-28078. PMLR, 2023.
5. Nguyen, E., Poli, M., Faizi, M., Thomas, A., Birch-Sykes, C., Wornow, M., Patel, A., Rabideau, C., Massaroli, S., Bengio, Y. and Ermon, S., HyenaDNA: long-range genomic sequence modeling at single nucleotide resolution. arXiv. arXiv preprint arXiv:2306.15794, 2024.
6. Chen, S., Wong, S., Chen, L. and Tian, Y., Extending context window of large language models via positional interpolation, arXiv preprint arXiv:2306.15595, 2023.

References

7. Xiong, W., Liu, J., Molybog, I., Zhang, H., Bhargava, P., Hou, R., Martin, L., Rungta, R., Sankararaman, K.A., Oguz, B. and Khabsa, M., Effective long-context scaling of foundation models. arXiv preprint arXiv:2309.16039, 2023.
8. Zhang, Z., Wayment-Steele, H.K., Brixi, G., Wang, H., Kern, D. and Ovchinnikov, S., Protein language models learn evolutionary statistics of interacting sequence motifs. Proceedings of the National Academy of Sciences, 121(45), p.e2406285121, 2024.