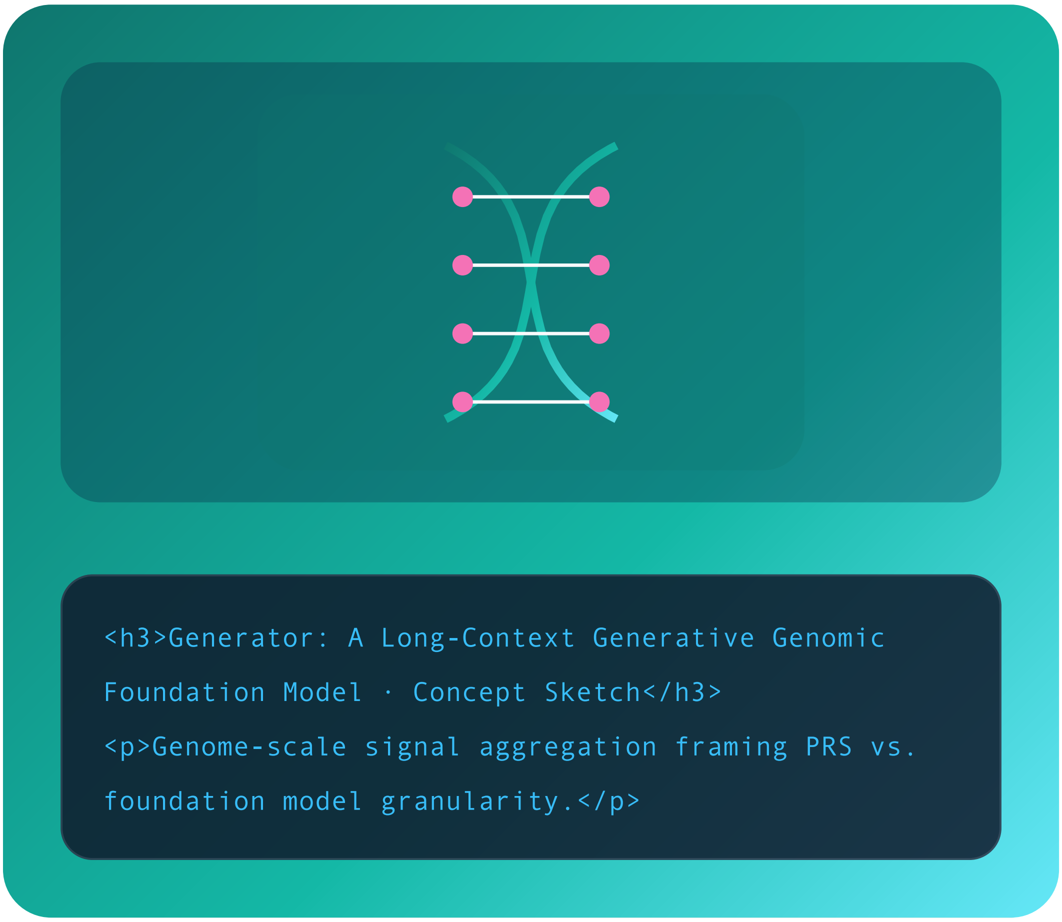

```
<h3>Generator: A Long-Context Generative Genomic
Foundation Model · Concept Sketch</h3>
<p>Genome-scale signal aggregation framing PRS vs.
foundation model granularity.</p>
```

Generator: A Long-Context Generative Genomic Foundation Model

Authors: Wei Wu, Qiuyi Li, Mingyang Li, Kun Fu, Fuli Feng, Jieping Ye, Hui Xiong, Zheng Wang

Year: 2025
Venue: arXiv (preprint)

1. Classification

- **Domain Category:**
  - Genomics FM. The paper develops a large generative language model over eukaryotic DNA sequences and evaluates it on a broad set of genomics prediction and design tasks.
- **FM Usage Type:**
  - Core FM development. The main contribution is a new 1.2B-parameter generative genomic foundation model with long context, specialized tokenization, and a carefully curated pretraining corpus.
- **Key Modalities:**
  - DNA sequence (eukaryotic genomes, gene regions vs whole-genome segments).
  - Derived protein sequences and enhancer activity measurements are used mainly as downstream evaluation targets.

2. Executive Summary

This paper introduces **Generator**, a large generative foundation model trained directly on eukaryotic DNA sequences to understand and design functional genomic sequences. The authors argue that existing genomic language models either lack generative ability, have short context, or are trained on relatively narrow datasets, which limits their usefulness for realistic genomic tasks. Generator is a 1.2B-parameter transformer decoder with a context length of about 98k base pairs, trained on 386 billion nucleotides from annotated gene regions of diverse eukaryotic species using next-token prediction and a 6-mer tokenizer. The model achieves state-of-the-art performance on standard genomic benchmarks (Genomic Benchmarks, Nucleotide Transformer tasks) and on new "Gener" tasks that probe long-range sequence understanding. Beyond benchmarks, Generator can generate DNA coding sequences whose translated proteins have realistic structure and statistics, and can design enhancer sequences with controllable activity levels. Overall, the work demonstrates that a large, long-context generative DNA model can serve as a versatile tool for genomic analysis and sequence design, pointing toward future applications in synthetic biology and precision genomics.

3. Problem Setup and Motivation

- **Scientific / practical problem:**
    - Build a **general-purpose generative model of eukaryotic DNA** that can both understand genomic sequences (for prediction tasks) and generate new, functional sequences (for design tasks).
    - Specifically, the model should handle **long genomic contexts** (tens of kilobases), capture the semantics of **gene regions and regulatory elements**, and support downstream tasks like classification, next-k-mer prediction, protein-coding sequence generation, and enhancer design.
- **Why existing approaches are limited:**
    - Many prior genomic models are **masked language models (MLMs)** trained with BERT-style objectives; they are strong for understanding but weaker or awkward for **generation**.
    - Context lengths are often only **hundreds to a few thousand base pairs**, which is too short for many realistic gene and regulatory contexts that span tens of thousands of base pairs.
    - Some generative models like HyenaDNA, megaDNA, and Evo are either limited to specific organism groups (e.g., bacteriophages, prokaryotes/viruses, or human-only) or use relatively small models or datasets, leaving a gap for large-scale **eukaryotic** generative models.
    - Naively training on entire genomes can flood the model with **non-gene, low-information regions**, potentially hurting downstream performance even if pretraining loss decreases.

- **Why this is hard (modeling and data challenges):**
  - DNA sequences are extremely long and **lack clear word boundaries**, making tokenization and context management nontrivial.
  - Computational cost scales badly with sequence length in standard transformers, so **long-context training** is expensive.
  - Functional regions (genes, promoters, enhancers) form only a **small fraction** of the genome; most bases are relatively redundant or low-entropy, so choosing what to train on matters.
  - Evaluating generative quality is difficult because there is not always a **ground truth** for “correct” generated DNA; one must rely on indirect metrics (e.g., downstream performance, protein structure statistics, enhancer activity predictions).

4. Data and Modalities

- **Datasets used (pretraining):**
  - DNA sequences are drawn from **all eukaryotic organisms** in the RefSeq database.
  - Two main strategies are compared:
    - **Gene Sequence Training (Scheme 1):** Uses annotated gene regions (including protein-coding genes, various RNAs, and regulatory elements like promoters and enhancers). This yields about **386 billion nucleotides** and is the configuration used for Generator.
    - **Whole Sequence Training (Scheme 2):** Mixes gene and non-gene regions from all eukaryotes, totaling about **2 trillion nucleotides**, producing the Generator-All variant.
- **Datasets used (downstream):**
  - **Nucleotide Transformer (NT) tasks:** A suite of genomics classification tasks (original and revised versions) covering promoters, enhancers, splice sites, chromatin marks, etc., across

many species.

  - **Genomic Benchmarks:** Primarily human-centric tasks such as enhancer vs non-enhancer, promoter vs non-promoter, regulatory element classification, and species discrimination.
  - **Gener tasks (proposed by this paper):**
    - **Gene classification:** Predict gene type from sequences of length 100–5000 bp.
    - **Taxonomic classification:** Predict taxonomic group from sequences of length 10,000–100,000 bp containing both gene and non-gene regions.
  - **Central dogma tasks:** Protein-coding DNA sequences for specific protein families (Histone and Cytochrome P450) from UniProt and RefSeq.
  - **Enhancer design:** The **DeepSTARR** dataset of enhancers with measured activity values (developmental vs housekeeping), including train/validation/test splits from the original DeepSTARR work.
- **Modalities and representations:**
  - Core input: **DNA sequence**, represented via a **6-mer tokenizer** (each token is a contiguous 6-base string).
  - Protein-level evaluation: DNA sequences are translated into **amino acid sequences** using the genetic code; protein language models and structure prediction tools are applied downstream.
  - Enhancer activity: numerical **activity values** (log2-transformed) from DeepSTARR, plus textual prompts `<high>` / `<low>` used as conditioning tokens in sequence design experiments.
- **Preprocessing / representation details:**
  - For gene sequence training, the authors **select annotated gene regions** and treat them as semantically rich de facto “sentences” for the model.
  - For whole sequence training, the model sees both gene and non-gene sequences directly.

- The 6-mer tokenizer is applied with a **random starting offset from 0 to 5** for each sample so that the model is not tied to a fixed phase of the genomic coordinate system.
- Long sequences are chunked into segments respecting the maximum token context.

5. Model / Foundation Model

- **Model Type:**
    - **Autoregressive transformer decoder**, broadly following the **LLaMA-style architecture** for causal language modeling, adapted to DNA.
- **New vs existing FM:**
    - This is a **new foundation model**, called **Generator**, designed specifically for eukaryotic DNA.
    - It is trained from scratch with DNA-specific choices for tokenization, data selection, and context length.
- **Key architectural configuration (approximate):**

Component	Value / Choice
Architecture	Transformer decoder (LLaMA-like)
Layers	26
Hidden size	2048
MLP / intermediate size	5632

Component	Value / Choice
Attention heads	32 (with 8 KV heads)
Vocabulary size	4128 (6-mer tokens)
Max token context	16,384 tokens
Max base-pair context	≈ 98,304 bp (because each token is 6 bp)
Positional encoding	RoPE (rotary position embeddings)
Activation	SiLU

- **Training objective and setup:**
  - Pretraining objective is **next-token prediction (NTP)** on 6-mer tokens, analogous to language modeling in NLP.
  - Batch size: about **2 million tokens** per batch.
  - Optimizer: **AdamW** with standard β values and weight decay, using **cosine learning rate schedule with warmup**.
  - Training spans **6 epochs** over 386B tokens (gene-only scheme), totaling ≈185k steps.
  - Implementation uses **FlashAttention** and **Zero Redundancy Optimizer (ZeRO)** to make long-context training efficient on GPUs (32 A100s).
- **Key components and innovations:**
  - **Data selection strategy:** Emphasis on **gene regions only** (Generator) versus gene + non-gene (Generator-All), showing that restricting to functional sequences can increase downstream performance even if pretraining loss is higher.

    - **Tokenization study:** Systematic comparison of single-nucleotide, k-mer, and BPE tokenizers for causal DNA LMs; finds that **6-mer tokenization** gives the best generative performance in NTP settings.
    - **Long-context capability:** By combining 6-mer tokens with a large context window (16k tokens ≈ 98k bp), the model can see **gene-scale and sub-chromosomal contexts** in a single forward pass.
    - **Generative downstream pipelines:**
        - Fine-tuning for **central dogma tasks**, i.e., generating protein-coding DNA whose translated proteins look realistic to protein language models and structure predictors.
        - Prompt-conditioned **enhancer design**, where `<high>` / `<low>` prompts steer the model toward sequences with desired activity.
- **Fine-tuning / downstream usage:**
    - For classification benchmarks, the model is fine-tuned using a linear head on top of embeddings (e.g., of an end-of-sequence token), with hyperparameter search over learning rates and batch sizes.
    - For central dogma and enhancer design, a **supervised fine-tuning (SFT)** stage adapts the model to particular protein families or enhancer datasets, after which autoregressive generation is used with temperature and nucleus sampling controls.

6. Multimodal / Integration Aspects (If Applicable)

- **Is this paper multimodal?**
    - The core model operates on **single-modality DNA sequence data**.

    - Protein sequences and enhancer activity measurements appear as **derived evaluation signals**, not as fully co-modeled input modalities in a single end-to-end multimodal architecture.
- **Relation to multimodal / integration ideas:**
    - Although Generator itself is not a multimodal model, its **embeddings and generative outputs** could be used in **late fusion pipelines** with other modalities (e.g., chromatin marks, expression, imaging) as described in the integration baseline plan, where per-modality representations are concatenated and fed into shallow models (logistic regression, GBDTs).
    - The emphasis on **semantically rich gene regions** is conceptually similar to the plan's principle of preserving **modality-specific signal**—here, "functional DNA segments" act as the high-value "modality" within the genome, and adding non-gene regions acts like injecting noise.
    - The careful evaluation practices (cross-validation, multiple benchmarks, explicit metrics) echo the baseline plan's focus on **robustness and disciplined evaluation**, even though the work is not explicitly an integration study.

7. Experiments and Results

- **Tokenizer and objective experiments (Next K-mer Prediction):**
    - The authors train multiple models with identical architecture but different tokenizers (single nucleotide, various k-mers, BPE with different vocabulary sizes) and evaluate on **next k-mer prediction** tasks.
    - They show that **k-mer tokenizers outperform BPE**, and within k-mers, the **6-mer tokenizer** gives the best accuracy across different input lengths.
    - They also compare a large **Mamba-based state space model** to transformer baselines and find that, despite its longer context and

efficiency, it does **not outperform** the transformer with 6-mer tokens as the input length grows.

- **Gene-only vs whole-genome training (Generator vs Generator-All):**
    - While the whole-sequence model (Generator-All) achieves **lower pretraining loss**, it underperforms Generator on almost all downstream tasks.
    - The authors argue that non-gene regions are often **redundant or non-functional**, so including them may dilute the high-information gene signal, effectively “contaminating” the data.
    - This supports the idea that **curated, semantically rich pretraining data** can be more valuable than raw volume.
- **Benchmark evaluations (Nucleotide Transformer tasks and Genomic Benchmarks):**
    - On both the **revised and original Nucleotide Transformer tasks**, Generator generally achieves **state-of-the-art or top-tier performance**, often surpassing Enformer, DNABERT-2, HyenaDNA, Nucleotide Transformer variants, Caduceus, and GROVER.
    - On **Genomic Benchmarks** (human-focused classification tasks), Generator again performs at or near the top; smaller specialized models like Caduceus sometimes come close but do not consistently dominate.
    - The **Gener tasks** (gene and taxonomic classification) further highlight Generator’s strengths, especially for **long-range sequence understanding**, where it reaches very high weighted F1 scores, including near-perfect performance on taxonomic classification.
- **Central dogma experiments (protein-coding sequence generation):**
    - After fine-tuning on DNA sequences encoding specific protein families (Histones and Cytochrome P450), Generator is used to generate new coding sequences.

    - Translating these sequences to proteins and analyzing them reveals:
        - Length distributions of generated proteins match those of natural families.
        - Protein language model perplexities (from Progen2) for generated sequences closely match those of natural proteins and differ from random shuffled controls.
        - **AlphaFold3** predictions and **Foldseek** structure search show many generated proteins with **high TM-scores (>0.8)** and high confidence, even when sequence identity is low (<0.3), indicating that the model is not simply memorizing training sequences.
- **Enhancer design experiments:**
    - A predictor fine-tuned from Generator on **DeepSTARR** enhancer data achieves **higher Pearson correlations** between predicted and measured enhancer activity than DeepSTARR itself and NT-multi, setting a new state of the art.
    - A further SFT stage adds `<high>` and `<low>` prompts to condition the model on desired activity levels.
    - Generated enhancers labeled `<high>` and `<low>` show **clear separation** in predicted activity distributions relative to each other and to natural enhancers, suggesting that Generator can perform **prompt-guided enhancer design**.
- **Overall empirical message:**
    - Generator is a **strong generalist** across many genomic classification benchmarks and **competent as a generative model** for both coding and regulatory sequences, with particular strengths stemming from its long context, curated pretraining data, and 6-mer tokenization.

8. Strengths, Limitations, and Open Questions

- **Strengths:**
    - **Long-context, large-scale generative model** tailored to eukaryotic DNA, filling a gap left by prior prokaryotic or short-context genomic models.
    - Careful **tokenization and data selection studies**, providing empirical guidance (e.g., 6-mer tokens, gene-only data) that can inform future genomic FMs.
    - Strong, consistent performance across a **wide range of benchmarks**, including newly proposed long-context tasks.
    - Demonstrated ability to **generate functional-like coding sequences** and to perform **prompt-conditioned enhancer design**, moving beyond pure prediction into actionable sequence design.
    - Extensive experimental detail (hyperparameter searches, cross-validation, ablations) that improves reproducibility and reliability.
- **Limitations:**
    - Pretraining focuses exclusively on **eukaryotic genomes**, leaving prokaryotic and viral DNA to other models like Evo; no unified genome-scale model across all domains of life is presented.
    - The model is **computationally heavy** (1.2B parameters, long sequences, many GPU hours), which may limit adoption in resource-constrained labs.
    - Most validations of generative quality are **in silico** (protein LMs, AlphaFold, enhancer predictors); there is limited or no wet-lab validation of generated sequences.
    - Despite long context, the model is still trained on **1D sequence alone**, without explicit modeling of 3D genome structure, epigenetic states, or cellular context.
    - As with other large models, interpretability of what the model has learned about regulatory grammar and long-range interactions remains challenging.

- **Open Questions and Future Directions:**
  - How would a **joint eukaryotic + prokaryotic + viral Generator** behave, and what design choices would be needed to balance these domains?
  - Can Generator's representations be combined with **other modalities** (chromatin accessibility, expression, epigenetics, single-cell data) in a systematic late-fusion or contrastive setup to improve downstream tasks like disease prediction?
  - What **interpretability tools** can be developed to probe the model's understanding of motifs, regulatory grammar, and long-range enhancer–promoter interactions?
  - How robust are Generator's predictions and designs across **different species and genomic contexts**, especially for non-model organisms with sparse annotations?
  - Can Generator be adapted for **clinical applications**, such as prioritizing noncoding variants in GWAS regions or designing therapeutic regulatory sequences, and what safety/ethics issues would arise?

9. Context and Broader Impact

- **Position in the landscape of foundation models in genomics:**
  - Generator sits alongside models like **Nucleotide Transformer**, **HyenaDNA**, **Caduceus**, and **Evo** as part of the emerging ecosystem of **genomic foundation models**.
  - Compared to large masked LMs (e.g., DNABERT-2, NT), Generator emphasizes **autoregressive generation, long context, and curated gene-only training data**, making it particularly suited for both understanding and designing DNA sequences.
  - It complements **Evo**, which targets prokaryotic and viral genomes, by focusing on the more complex **eukaryotic** genomic setting.

- **Conceptual analogies for intuition:**
    - You can think of Generator as a **GPT-style model for eukaryotic DNA**, where tokens are 6-mer substrings rather than words, and the context window spans whole genes or multi-gene regions instead of paragraphs.
    - The central dogma experiments are somewhat analogous to asking a text model to generate **syntactically valid and semantically coherent stories** that pass external quality checks, except here the “checkers” are protein LMs and structure predictors.
- **Relevance to integration and broader research programs:**
    - The model’s strong sequence-level representations could serve as a **genomic backbone** in larger multimodal systems that integrate DNA with other omics or imaging modalities via late fusion, as recommended in the integration baseline plan.
    - Its success supports the broader thesis that **domain-specific, long-context FMs** can provide robust building blocks for downstream applications, from basic gene regulation studies to clinical genomics and synthetic biology.

10. Key Takeaways (Bullet Summary)

- **Problem:** Existing genomic language models often lack **generative capability**, have **short context windows**, or are limited in organismal scope, constraining their usefulness for realistic eukaryotic genomics tasks.

- **Problem:** Training directly on whole genomes may emphasize low-information non-gene regions, potentially hurting performance even when pretraining loss looks better.

- **Method / model:** Generator is a **1.2B-parameter transformer decoder** with **≈98k bp context length**, trained with next-token prediction on **386B nucleotides** of eukaryotic gene regions.

- **Method / model:** A systematic study of **tokenizers** finds that **6-mer tokens** work best for causal DNA language modeling, beating both single-nucleotide and BPE tokenization.

- **Method / model:** A comparison between **gene-only** and **whole-genome** pretraining shows that focusing on **semantically rich gene regions** yields better downstream performance than including vast non-gene regions.

- **Results:** Generator achieves **state-of-the-art or near-SOTA performance** on Nucleotide Transformer tasks, Genomic Benchmarks, and newly proposed Gener tasks, particularly excelling in long-sequence understanding.

- **Results:** In **central dogma experiments**, Generator can generate protein-coding DNA whose translated proteins have realistic lengths, protein-LM perplexities, and 3D structures, indicating true generative competence rather than memorization.

- **Results:** For **enhancer design**, a Generator-based predictor surpasses previous models on DeepSTARR data, and prompt-conditioned generation produces enhancer sequences with clearly different predicted activity profiles.

- **Why it matters:** Generator demonstrates that a **large, long-context generative FM for eukaryotic DNA** can serve as a powerful tool for both **genomic analysis and sequence design**, opening doors to more sophisticated applications in synthetic biology, variant interpretation, and future multimodal integration with other biological data types.