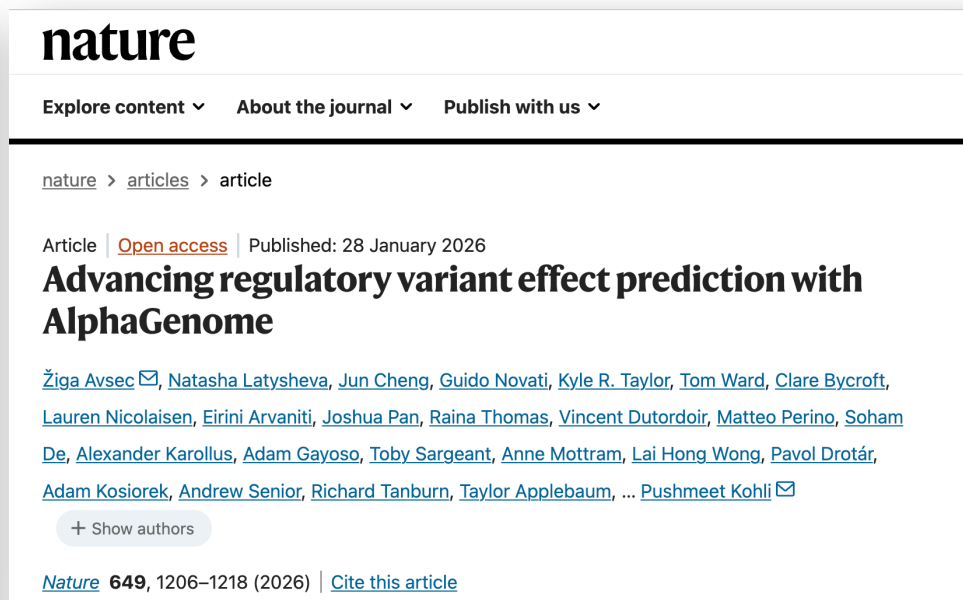


# ゲノム言語モデル研究動向と 開発進捗状況

東 光一

国立遺伝学研究所

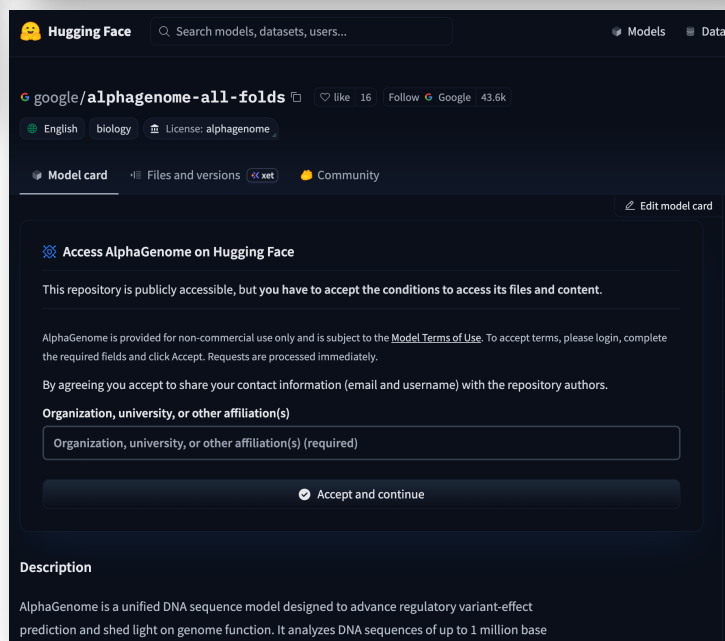
# AlphaGenome論文の公開 (Nature)



モデルについては半年前の研究会で議論したプレプリント版から大きく変更は無い。

変更点：  
いくつかの評価指標  
学習プロセスの詳細な説明

モデルパラメータの公開：  
これまで同様のAPIに加え、  
モデルのソースコードがGithubで、  
事前学習パラメータがHFで公開  
された。  
H100 1枚で動くらしい。



# 基盤モデル開発関連研究①

BGIが開発。

1.2B/10Bモデル

1塩基レベル、1Mbpコンテキスト

Transformer+MoE

Grouped-Query Attention, Flash-attention

8 expertsのうち、各塩基につき2つだけが活性化。

データセットはHuman中心

GRCh38, CHM13,

HPRC (Human Pangenome Reference Consortium)

HGSVC (Human Genome Structural Variation Consortium)

事前学習では遺伝子間領域を除外。

定番ベンチ (GB, NT, LRB) の他、

ENCODEデータを使ったファインチューニング  
によるCell type特異的発現量予測も。

Megatron-LMを利用、256GPUで学習。

5Dパラレル (PP, TP, DP, CP, Expert Pararell)

モデルソースコード、パラメータも公開。



(GIGA)<sup>n</sup>  
SCIENCE

GigaScience, 2025, 14, 1–13

DOI: 10.1093/gigascience/giaf132

Advance access publication date: 22 October 2025

Research

## Genos: a human-centric genomic foundation model

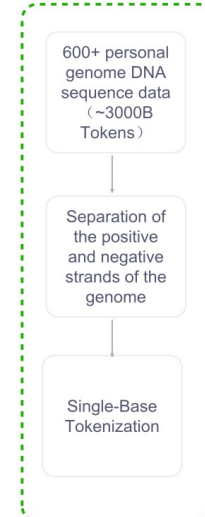
Adi Lin<sup>†</sup>, Bin Xie<sup>†</sup>, Cheng Ye<sup>†</sup>, Cheng Wang<sup>†</sup>, Duoyuan Chen<sup>†</sup>, Ercheng Wang<sup>†</sup>, Fanfeng Lu<sup>†</sup>, Guirong Xue<sup>†</sup>, Haiqiang Zhang<sup>†</sup>, Jiajie Zhan<sup>†</sup>, Jianfeng Zhang<sup>†</sup>, Jiangshuan Pang<sup>†</sup>, Jianqiang Liang<sup>†</sup>, Jiawei Lin<sup>†</sup>, Jiabin Ma<sup>†</sup>, Jie Hu<sup>†</sup>, Jing Ma<sup>†</sup>, Jinni Dong<sup>†</sup>, Jiongzheng Li<sup>†</sup>, Junchen Liu<sup>†</sup>, Junhong Chen<sup>†</sup>, Junyou Li<sup>†</sup>, Kai Ding<sup>†</sup>, Kaiwen Deng<sup>†</sup>, Kui Chen<sup>†</sup>, Lihui Wang<sup>†</sup>, Longqi Liu<sup>†</sup>, Ling Guo<sup>†</sup>, Liwen Xiong<sup>†</sup>, Luhao Yang<sup>†</sup>, Ming Cheng<sup>†</sup>, Nanning Chen<sup>†</sup>, Renzhong Chen<sup>†</sup>, Shanxin Sun<sup>†</sup>, Shaoshuai Li<sup>†</sup>, Shicheng Chen<sup>†</sup>, Shiping Liu<sup>†</sup>, Siwei Xie<sup>†</sup>, Suyan Liu<sup>†</sup>, Tao Zhou<sup>†</sup>, Wangyang Tang<sup>†</sup>, Weiqiang Zhang<sup>†</sup>, Xianyue Jiang<sup>†</sup>, Xianzhi Qi<sup>†</sup>, Xin Jin<sup>†</sup>, Xinjiang Tan<sup>†</sup>, Xinyue Hu<sup>†</sup>, Xun Xu<sup>†</sup>, Xuyang Feng<sup>†</sup>, Yafei Lu<sup>†</sup>, Yifan Gao<sup>†</sup>, Yong Shang<sup>†</sup>, Youzhe He<sup>†</sup>, Yue Yuan<sup>†</sup>, Yufan Wang<sup>†</sup>, Yuqi Liu<sup>†</sup>, Zhan Xiao<sup>†</sup>, Zhangyuan Meng<sup>†</sup>, Zhaorong Li<sup>†</sup>, Zhe Zhao<sup>†</sup>, Zheng Yang<sup>†</sup>, and Zilin Wang<sup>†</sup>

Genos Team, West Lake District, Hangzhou 310012, China

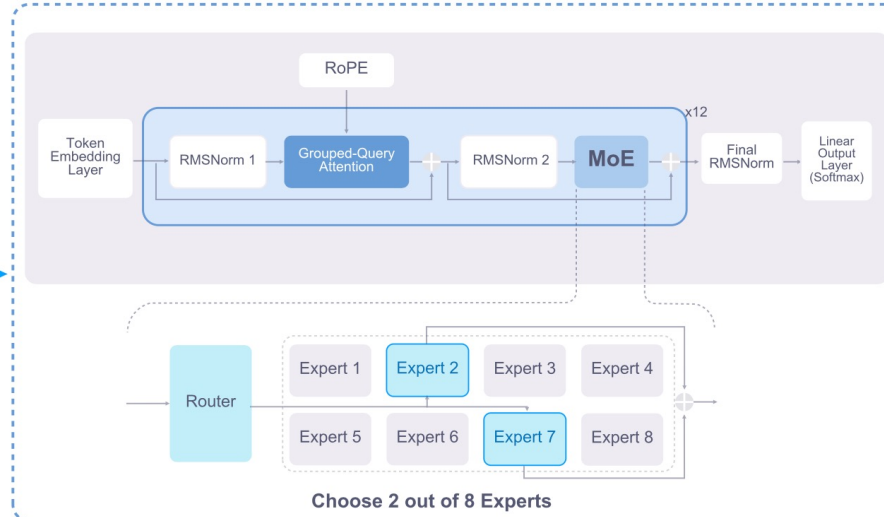
\*Correspondence address. Xun Xu, Genos Team, West Lake District, Hangzhou 310012, China. E-Mail: [xuxun@genomics.cn](mailto:xuxun@genomics.cn)

<sup>†</sup>All authors contributed equally. Authors are ranked in alphabetical order by their first names.

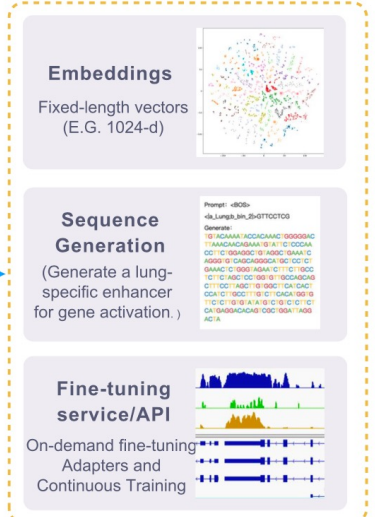
### Data Processing



### Model Architecture and Training(up to 1M bps context)



### Model Inference and Applications



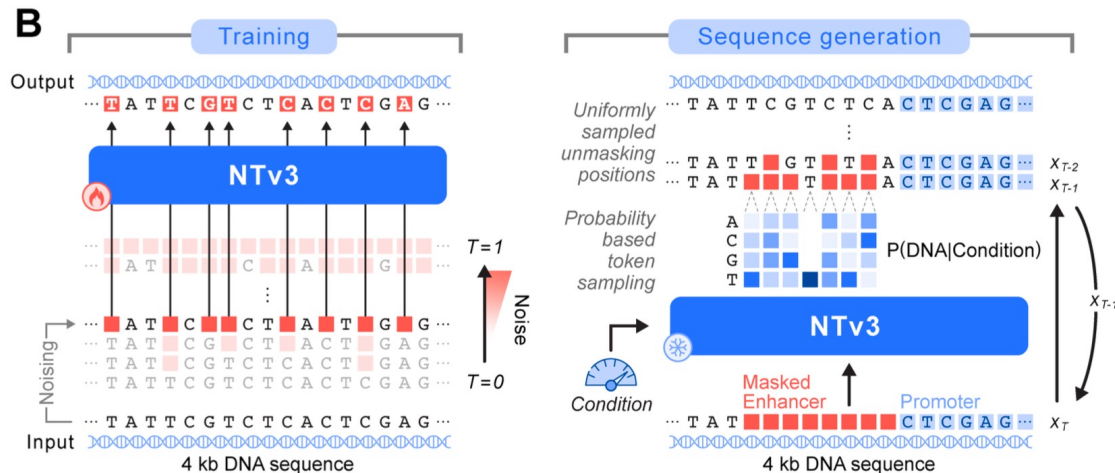
# 基盤モデル開発関連研究②

## NTv3 (Nucleotide Transformer v3)

畳み込み圧縮とTransformerレイヤで変換された表現を  
AlphaGenomeと類似したU-netライクな構造で  
一塩基解像度に戻す

1Mbpコンテキストを単一のA100/H100で推論可能。  
最大650Mパラメータ

AlphaGenome同様のマスク言語モデルだが、  
拡散モデル (MDLM) のDenoisingに使う配列生成ができる。  
500bp程度のエンハンサ生成を検証。



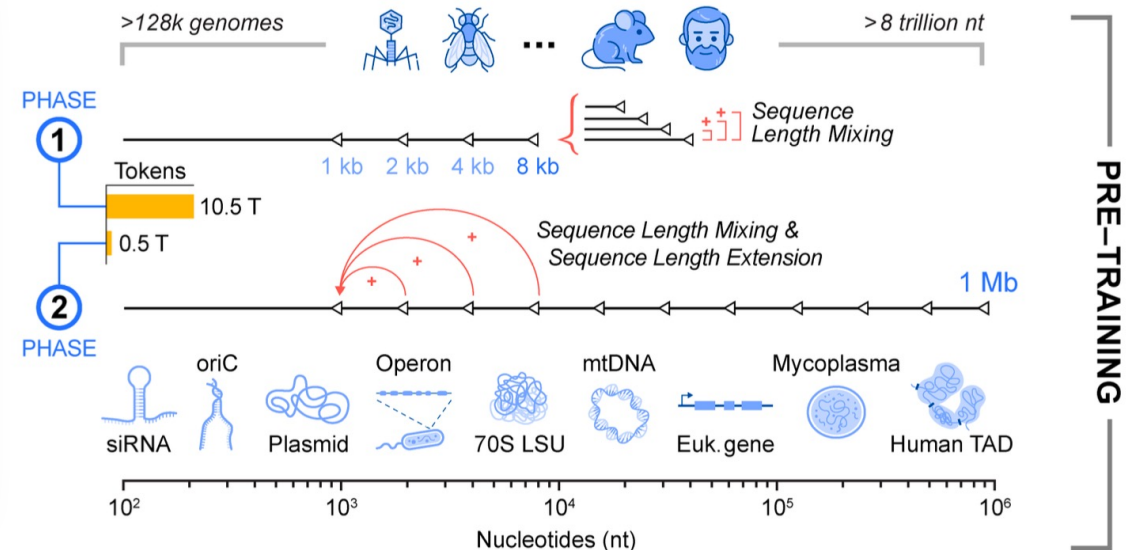
2026/1/30

## A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction

Sam Boshar<sup>\*1</sup>, Benjamin Evans<sup>\*1</sup>, Ziqi Tang<sup>\*1</sup>,

Armand Picard<sup>1</sup>, Yanis Adel<sup>1</sup>, Franziska K. Lorbeer<sup>2</sup>, Chandana Rajesh<sup>1</sup>, Tristan Karch<sup>1</sup>, Shawn Sidbon<sup>1</sup>, David Emms<sup>1</sup>, Javier Mendoza-Revilla<sup>1</sup>, Fatimah Al-Ani<sup>1</sup>, Evan Seitz<sup>1</sup>, Yair Schiff<sup>4</sup>, Yohan Bornachot<sup>1</sup>, Ariana Hernandez<sup>1</sup>, Marie Lopez<sup>1</sup>, Alexandre Laterre<sup>1</sup>, Karim Beguir<sup>1</sup>, Peter Koo<sup>5</sup>, Volodymyr Kuleshov<sup>4</sup>, Alexander Stark<sup>2,3</sup>, Bernardo P. de Almeida<sup>\*\*1</sup> and Thomas Pierrot<sup>\*\*1</sup>

<sup>\*</sup>Equal contributions, <sup>\*\*</sup>Corresponding authors: b.dealmeida@instadeep.com, t.pierrot@instadeep.com, <sup>1</sup>InstaDeep, London, UK, <sup>2</sup>Research Institute of Molecular Pathology (IMP), Vienna BioCenter (VBC), Vienna, Austria, <sup>3</sup>Medical University of Vienna, Vienna BioCenter (VBC), Vienna, Austria, <sup>4</sup>Cornell Tech, New York, USA, <sup>5</sup>Cold Spring Harbor Laboratory, New York, USA





# 基盤モデル開発関連研究③

EDEN (Environmentally-Derived Evolutionary Network)

構造：

Llama3.1ライクなDecoder-only transformer

パラメータサイズは 100M/1B/7B/28B (非公開)

コンテキストは8,192bp固定。

データ：

BaseData (メタゲノム、ファージ、MGE関連などが厚め)

コンティグ長や遺伝子密度で細かくフィルタリング

合計9.7T tokens

OpenGenome2 (Evo系の学習データ) と同条件で比較すると、  
小規模モデルでは大差ないが >7Bモデルではデータ品質が効いてくるらしい。

学習：

H200 x 1,008枚

Prompt-based programmable insertion

希望する挿入先の DNA配列 30bp だけを与えて、そこに挿入できるLSRを生成可能。  
ヒトゲノムの任意の領域 (疾患関連領域など) の “attっぽい配列” をプロンプトにして、  
そこをターゲットにするLSRを生成できた。

(実験的に検証しているがオフターゲットは不明)

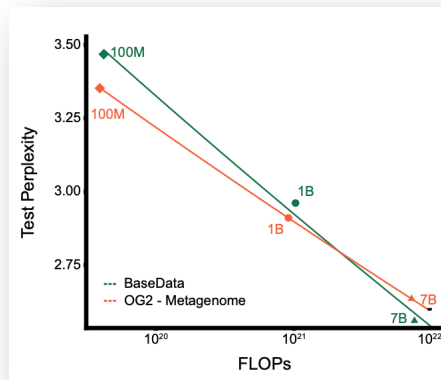
他、

「病原体タグで条件づけた抗菌ペプチド生成」

「マイクロバイオーーム全体の配列生成」 (生態系レベルのDNA配列確率分布の学習)

など。

2026/1/30



## Designing AI-programmable therapeutics with the EDEN family of foundation models

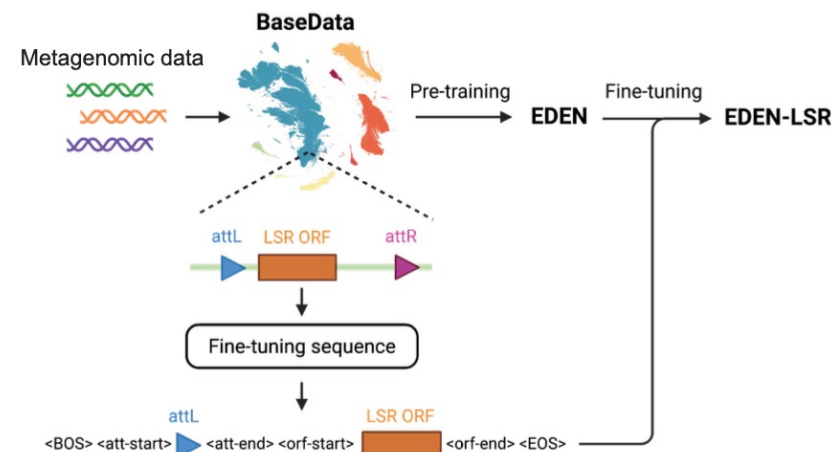
Geraldene Munsamy<sup>†1</sup>, Gavin Ayres<sup>†1</sup>, Carla Greco<sup>†1</sup>, Keith Kam<sup>†1</sup>, Gus Minto-Cowcher<sup>†1</sup>, John St John<sup>†2</sup>, Tanggis Bohnuud<sup>†1</sup>, Matthew Bakalar<sup>†1</sup>, William Chow<sup>†1</sup>, Robert Pecoraro<sup>†1</sup>, Marcello der Torossian Torres<sup>†3</sup>, Aaron Kollasch<sup>†1</sup>, Marcus Leung<sup>†1</sup>, Hassan Sirelkhathim<sup>†2</sup>, Francesco Farina<sup>†1,2</sup>, Connor McGinnis<sup>†1</sup>, Srijani Sridhar<sup>†1</sup>, Daniel Anderson<sup>†1</sup>, Francesco Oteri<sup>†1</sup>, Ali Taghibakhshi<sup>†2</sup>, Jeremie Dona<sup>†1</sup>, Tyler Shimko<sup>†2</sup>, Cedric Stenbeeke<sup>2</sup>, Alexandros Papadopoulos<sup>1</sup>, Malcolm Krolick<sup>1,4</sup>, Fabian Spoendlin<sup>1,5</sup>, Purba Gupta<sup>1</sup>, Sandeep Kumar<sup>1</sup>, Anne Bara<sup>1</sup>, Jared Wilbur<sup>2</sup>, Noelia Ferruz<sup>6</sup>, Timur Rvachov<sup>2</sup>, Fangping Wang<sup>3</sup>, Hanqun Cao<sup>3</sup>, Hyun-Su Lee<sup>3</sup>, Japan Mehta<sup>1</sup>, Raphael Chaleil<sup>1</sup>, Valerio Pereno<sup>1</sup>, Sid Potti<sup>1,7</sup>, Chris Emerson<sup>2</sup>, Roy Tal Dew<sup>2</sup>, Kevin K Yang<sup>8</sup>, Eric Nguyen<sup>7</sup>, Neha Tadimet<sup>2</sup>, Jillian F. Banfield<sup>9</sup>, Alicia Frame<sup>10</sup>, Emma Bolton<sup>1</sup>, David Ruau<sup>2</sup>, Rory Kelleher<sup>2</sup>, Anthony Costa<sup>2</sup>, Kimberley Powell<sup>2</sup>, Cesar de la Fuente-Nunez<sup>†3</sup>, Glen-Oliver Gowers<sup>1</sup>, Oliver Vince<sup>1</sup>, Jonathan Finn<sup>1</sup>, Philipp Lorenz<sup>††1</sup>

<sup>1</sup> Basecamp Research, <sup>2</sup> NVIDIA Corporation, <sup>3</sup> University of Pennsylvania,

<sup>4</sup> Johns Hopkins University, <sup>5</sup> Department of Statistics, University of Oxford,

<sup>6</sup> Centre for Genomic Regulation, Barcelona, <sup>7</sup> Stanford University,

<sup>8</sup> Microsoft Research, <sup>9</sup> University of California, Berkeley, <sup>10</sup> CoreAI, Microsoft Corporation



トウモロコシの系統Chang7-2を  
メタンスルホン酸エチル（変異原）で処理した  
ヘテロ接合のM1世代、1,716系統をWGS解析して  
変異を同定。

pLMやgLMdで予測した対数尤度低下が、実際の変異  
の有害性（自家受粉M2世代ホモ接合致死など）を予測  
できるかを検証。

利用したgLMはPlantCaduceus（CDSの変異はESMでも評価）  
→ Precision >97%、Recall 68.8%

・ 因果変異の同定

M1 (Wt/Mut)

M2 (WT/WT, WT/Mut, Mut/Mut)

M2世代25%に出現する表現型について、  
数千候補のホモ接合変異（アليل頻度100%）のうち、どれが  
その表現型に影響を持つのか、  
因果効果を尤度（適応度低下）で絞り込める

以下の表現型の因果変異を同定

- ・ アルビノ：ミスセンス変異
- ・ 光沢の低下（葉表面のワックス減少）：ミスセンス変異
- ・ 白色種子：スプライシング変異
- ・ 雄性不稔：ナンセンス変異（終始コドン）

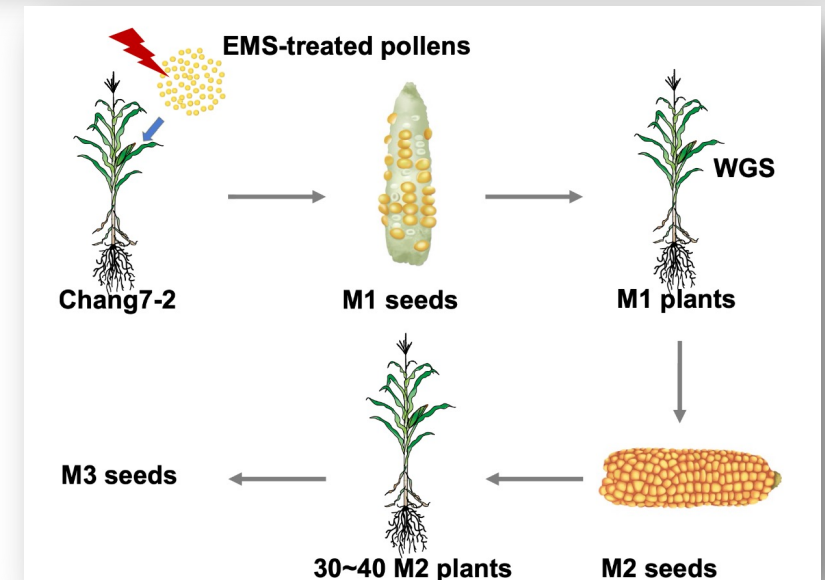
## RESEARCH

## Open Access



# Large DNA and protein language models enhance discovery of deleterious mutations in maize

Yiru Wang<sup>1†</sup>, Cheng He<sup>2,3,4†</sup>, Qiqi Wang<sup>1†</sup>, Rui Li<sup>1</sup>, Yang Qin<sup>1</sup>, Hongwu Wang<sup>1</sup>, Mingyue Gou<sup>5</sup>, Xuebin Zhang<sup>6</sup>, Huiyong Li<sup>7</sup>, Quansheng Huang<sup>8</sup>, Xinfang Liu<sup>9</sup>, Jinliang Yang<sup>10,11</sup>, Frank F. White<sup>12</sup>, Junjie Fu<sup>1</sup>, Sanzhen Liu<sup>4\*</sup> and Jun Zheng<sup>1\*</sup>



# gLM評価関連研究

相変わらず生成能力の評価研究は不足しているが、いくつか出始めている。

bioRxiv preprint doi: <https://doi.org/10.64898/2026.01.17.700093>; this version posted January 18, 2026. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

## Fundamental limitations of genomic language models for realistic sequence generation

Alexandros Tzanakakis<sup>1</sup>, Ioannis Mouratidis<sup>1,\*</sup>, Ilias Georgakopoulos-Soares<sup>1,\*</sup>,

<sup>1</sup> Division of Pharmacology and Toxicology, College of Pharmacy, The University of Texas at Austin, Dell Paediatric Research Institute, Austin, TX, USA.

この論文ではEvo2で生成された、さまざまな生物種の配列が現実のゲノムのパターンをどの程度正確に再現しているかを評価。

対象生物種：真核生物20種、細菌52種、古細菌9種、ウイルス129種

評価対象：

Evo2で各ゲノムについて300kbpを40個生成

（系統タグ+オリジナル3kbpをプロンプトとし、297kbpを生成）

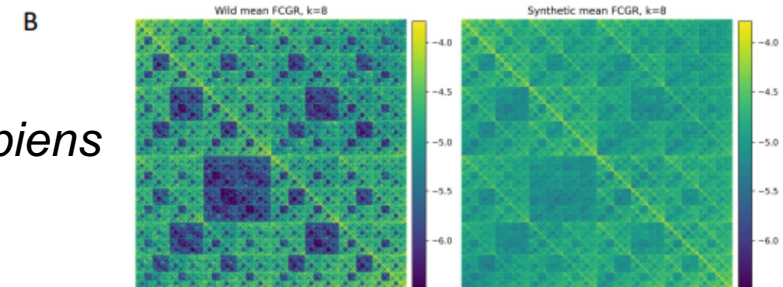
k-mer分布が一致しない。

カオスゲーム表現も歪む。

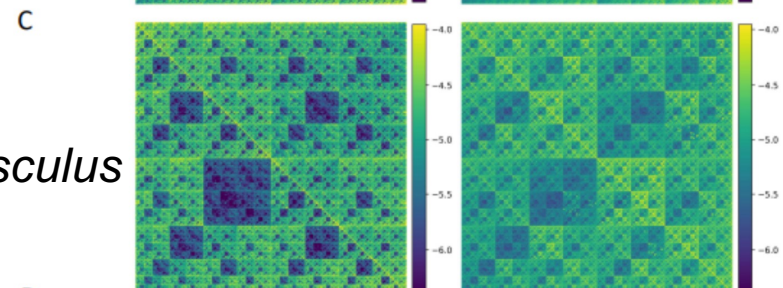
真核だとリピート配列を避ける傾向がある？逆に原核はリピート過多？

転写因子結合部位のゲノム上分布がおかしい？

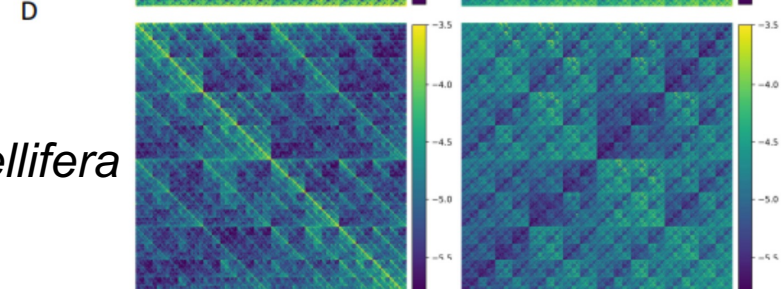
*Homo sapiens*



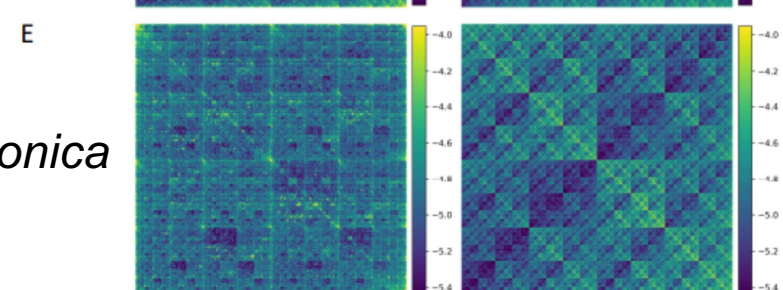
*Mus musculus*



*Apis mellifera*



*Saccharina japonica*



# ゲノム言語モデル開発目標

## Evo2 7B相当のモデルを独自データセットでフルスクラッチ学習する

- 計算基盤の構築と学習を通して、論文に書かれていないモデルや学習プロセスの「クセ」の知識を蓄積する。
- モデル性能の評価、応用開発の体制構築
- 本当に、この性能のモデルの構築にこの規模のパラメータが必要なのか？  
なんらかの帰納バイアスを入れて、もっとシンプルなモデルで同等の計算ができないのか。



# ゲノム言語モデル開発進捗状況

遺伝研スパコンB200の2ノード

（ノードあたりBlackwell B200 GPU x 8, 2ノード合計16枚）を利用して  
Evo2（StripedHyena2） 7B相当モデルのフルスクラッチ学習を実行中。

データセットは

GlobDB由来高品質MAG（Metagenome-assembled genomes）データセット  
約15万ゲノム、合計 約500B tokens

8,192bpチャンク分割、tokenization計算済み。

データリークしないようにゲノム単位で train/val/test 分割。

test/valやバッチに系統が偏らないように、

ANI距離に基づいてゲノムの並びを制御。

```
Input files      : 145486
Input contigs    : 18273440
Total bp (contigs) : 481,413,711,551
Kept chunks      : 49,941,497
Dropped (N-frac) : 18,465 (threshold=0.1)
Dropped (short)  : 0 (min_contig_len=1000)
```

BioNeMoで提供されている

Blackwell対応Megatron-LMコンテナイメージをベースに開発をしている。

# モデル開発チーム会議

遺伝研：東、黒川

科学大：廣田、増田、豊田

筑波大：鈴木

東大：笠原

第1回：2025年12月23日

第2回：2026年1月6日

第3回：2026年1月16日

第4回：2026年1月20日

第5回：2026年1月27日

# Megatron/Evo2 trainerの改造

LLM事前学習で使われているデータ構造（GPTDataset）をそのまま流用しているため、そのままだと、我々の学習方針とは相容れない特徴がある

- ドキュメント（DNA配列断片）をすべて連結して巨大なストリームにしてからチャンクに分割
- 分割後、トレーニングデータ全体でチャンクがランダムにシャッフルされてしまう

そのため、データセットビルダとローダは

- 提示したDNA配列サンプルを厳密に独立サンプルとして学習する
- データセット中のサンプル提示順をシャッフルしない

ように、追加で実装した。（Evo2OrderedChunkDatasetクラス）

```
301 class Evo2OrderedChunkDataset(Evo2Dataset):
302     """
303     - 1 IndexedDataset element (=1 doc) を 1 サンプルとして返す
304     - doc を跨がない
305     - 順序を保持（シャッフルしない）
306     - GPTDataset の index mapping cache (path_to_cache) 保存形式は踏襲する
307     """
308     # 必要なら設定: EOD を予測させない (labels==EOD を loss から外す)
309     MASK_LABEL_EOD: bool = True
310
311     def _build_document_sample_shuffle_indices(self) -> Tuple[numpy.ndarray, numpy.ndarray, numpy.ndarray]:
312         """
313         GPTDataset のキャッシュ機構 (path_to_cache) を踏襲しつつ、
314         ordered/no-cross-doc 用の “恒等/ダミー” index を作る。
315         """
316
317         # --- ここから下は GPTDataset のキャッシュ枠組みを踏襲 ---
318         path_to_cache = self.config.path_to_cache
```

# StripedHyena2 7B相当モデル事前学習の開始

B200 x 16 を利用し、バッチサイズ32 (DP=16/PP=1/TP=1/CP=1, マイクロバッチ=1, 勾配累積=2) で学習。

ハイブリッド構造は32層で以下の通り。全体で約6.5Bの学習可能パラメータ。

**SDH\*SDHSDH\*SDHSDH\*SDHSDH\*SDHSDH\***

(S = Short-range Hyena, D = Mid-range Hyena, H = Implicit Hyena, \* = Attention)

コンテキストサイズは8,192bp

```
Training epoch 0, iteration 0/180799 | lr: 0 | global_batch_size: 32 | global_step: 0 | reduced_train_loss: 5.708 | train_step_timing in s: 28.19
Training epoch 0, iteration 1/180799 | lr: 2.5e-07 | global_batch_size: 32 | global_step: 1 | reduced_train_loss: 5.892 | train_step_timing in s: 1.742 | consumed_samples: 64
Training epoch 0, iteration 2/180799 | lr: 5e-07 | global_batch_size: 32 | global_step: 2 | reduced_train_loss: 5.202 | train_step_timing in s: 2.567 | consumed_samples: 96
Training epoch 0, iteration 3/180799 | lr: 7.5e-07 | global_batch_size: 32 | global_step: 3 | reduced_train_loss: 3.511 | train_step_timing in s: 2.48 | consumed_samples: 128
Training epoch 0, iteration 4/180799 | lr: 1e-06 | global_batch_size: 32 | global_step: 4 | reduced_train_loss: 3.812 | train_step_timing in s: 2.531 | consumed_samples: 160
Training epoch 0, iteration 5/180799 | lr: 1.25e-06 | global_batch_size: 32 | global_step: 5 | reduced_train_loss: 4.127 | train_step_timing in s: 1.884 | consumed_samples: 192
Training epoch 0, iteration 6/180799 | lr: 1.5e-06 | global_batch_size: 32 | global_step: 6 | reduced_train_loss: 3.168 | train_step_timing in s: 1.849 | consumed_samples: 224
Training epoch 0, iteration 7/180799 | lr: 1.75e-06 | global_batch_size: 32 | global_step: 7 | reduced_train_loss: 6.785 | train_step_timing in s: 1.491 | consumed_samples: 256
Training epoch 0, iteration 8/180799 | lr: 2e-06 | global_batch_size: 32 | global_step: 8 | reduced_train_loss: 6.708 | train_step_timing in s: 1.493 | consumed_samples: 288
Training epoch 0, iteration 9/180799 | lr: 2.25e-06 | global_batch_size: 32 | global_step: 9 | reduced_train_loss: 5.712 | train_step_timing in s: 1.494 | consumed_samples: 320
Training epoch 0, iteration 10/180799 | lr: 2.5e-06 | global_batch_size: 32 | global_step: 10 | reduced_train_loss: 5.656 | train_step_timing in s: 1.49 | consumed_samples: 352
Training epoch 0, iteration 11/180799 | lr: 2.75e-06 | global_batch_size: 32 | global_step: 11 | reduced_train_loss: 5.995 | train_step_timing in s: 1.565 | consumed_samples: 384
Training epoch 0, iteration 12/180799 | lr: 3e-06 | global_batch_size: 32 | global_step: 12 | reduced_train_loss: 5.848 | train_step_timing in s: 1.472 | consumed_samples: 416
```

1ステップ (32サンプル) の学習に平均1.5秒かかる。なので、

$8,192 * 32 / 1.5 / 16 = 10,923 \text{ tokens/sec/GPU}$

2ノード (16GPU) を占有して学習を回し続けられるなら、

我々の学習データセット (総トークン数約500B) の1 epochの学習に約33日間かかる。



# トレーニング状況

約1,000ゲノム

(我々の学習データの約0.7%)

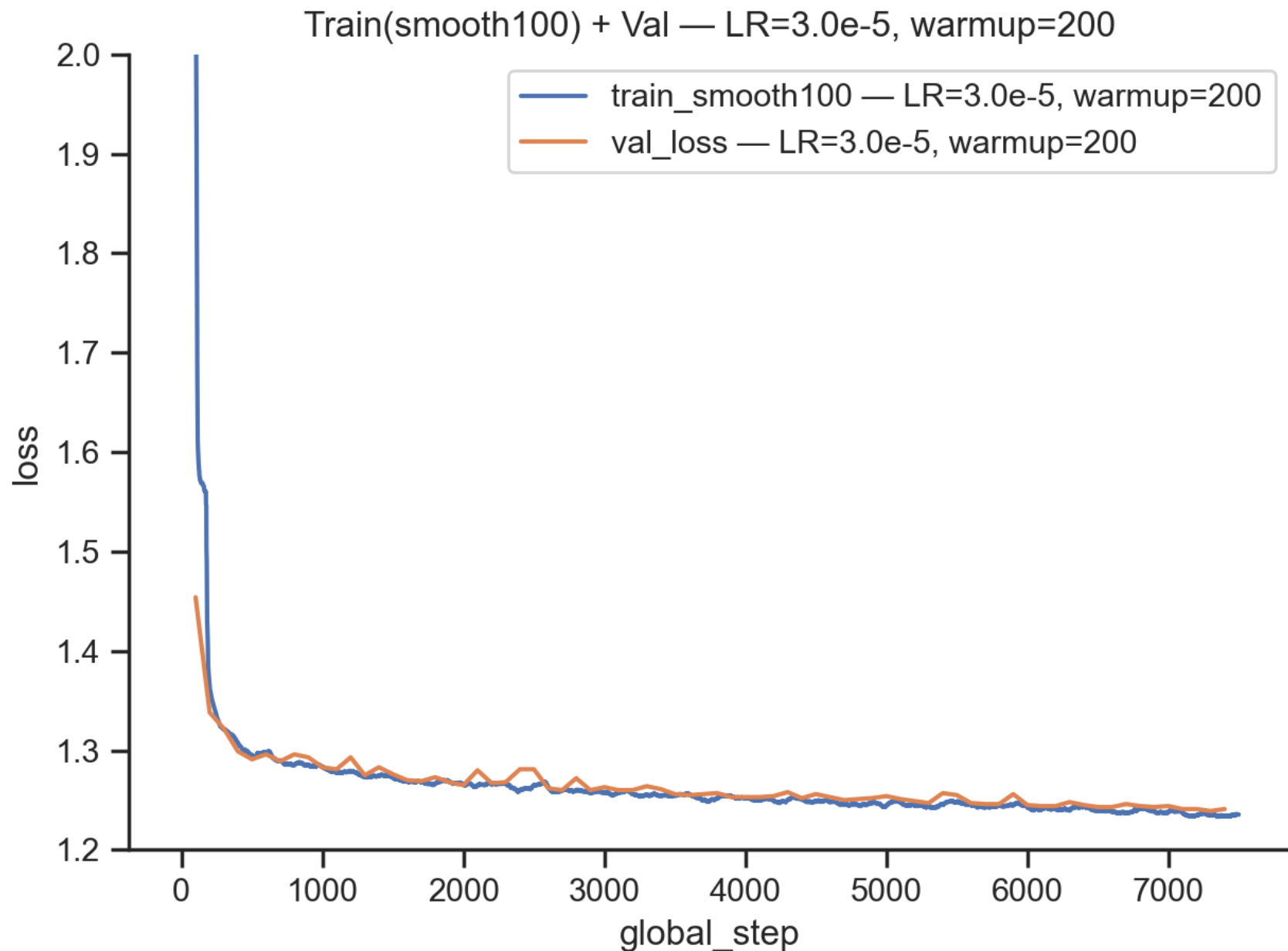
ぶんに該当するステップ

( $\approx 7,500$  steps)

までの train/val loss

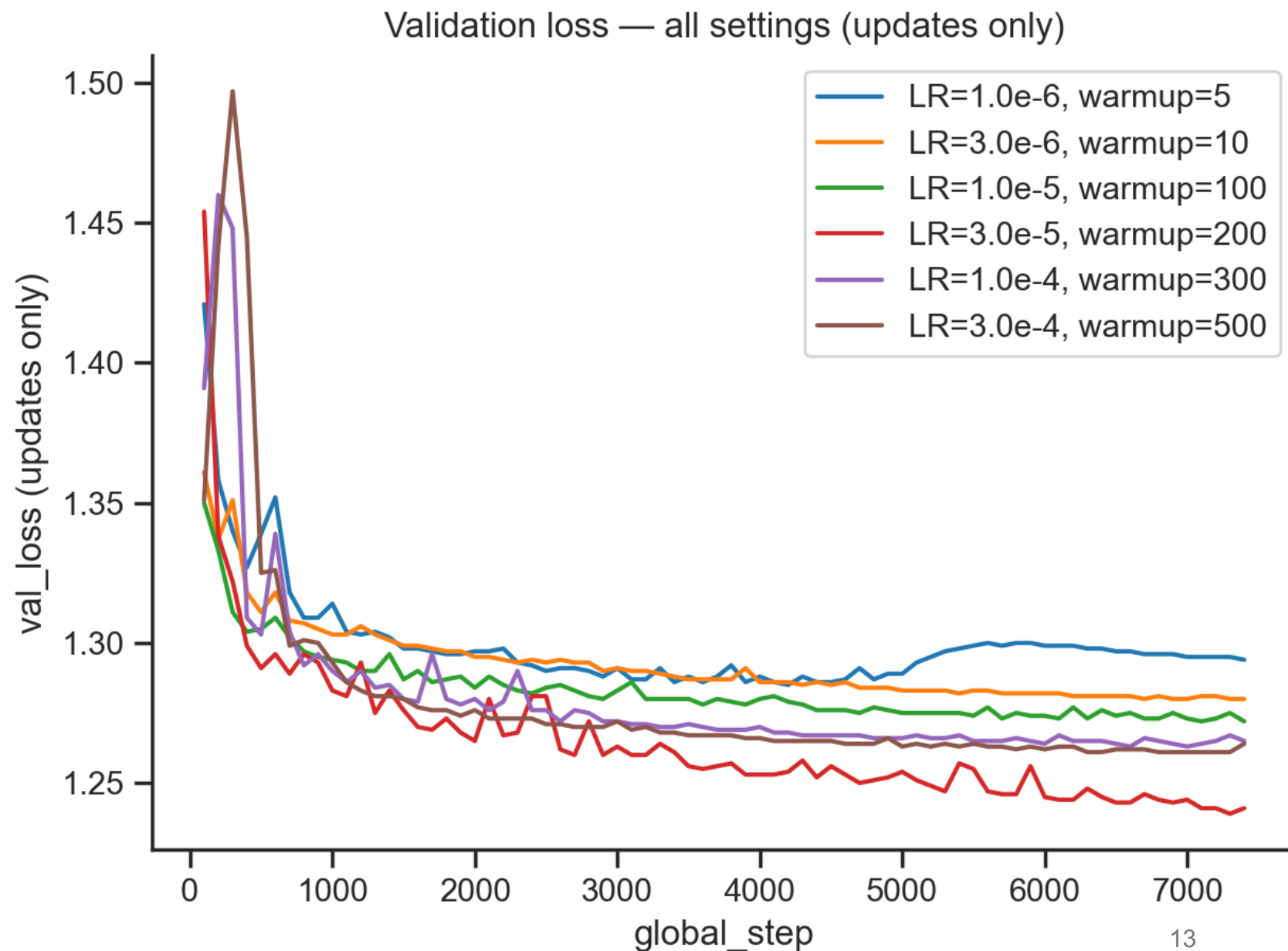
非常にゆっくりとだが、  
少しずつ下がってはいる。

なお、最新版EvoのNLLは  
だいたい0.95~1.0くらい



# ハイパーパラメータ探索

いずれも 7B モデルトレーニング  
の本番環境と同一条件で比較  
(DP/TP/PP/CP, batch size)



# 新規モデル開発

やはり進化系統による条件付けはちゃんと入れたい。

Evo2ではどうなっているか？

→「系統タグ」による条件付け

“|d\_\_\_\_;p\_\_\_\_;c\_\_\_\_;o\_\_\_\_;f\_\_\_\_;g\_\_\_\_;s\_\_\_\_|”

などの文字列。

ただし、Byte-level tokenのモデルで、特殊トークンも設定されていないため、これらの文字列はDNA (A, T, G, C) と同列にASCIIコードに変換され、Hyena集約やAttention演算が行われる。(loss 計算からはマスクされる)

データローダの設計上、系統タグがチャンクの頭につくとは限らない。

この設計の是非は？

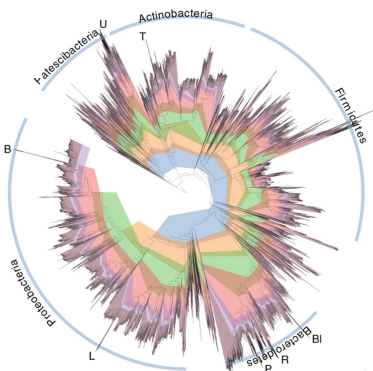
生物の「名前」は進化系統とは無関係。

(微生物だと形態や代謝特性、サンプリング環境など)

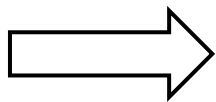
進化系統情報はもっとちゃんと入れられるのでは。

→系統樹（ツリートポロジー）上の「座標」をユークリッド空間に埋め込む手法の開発

GTDB  
Tree



N-dimensional  
vectors

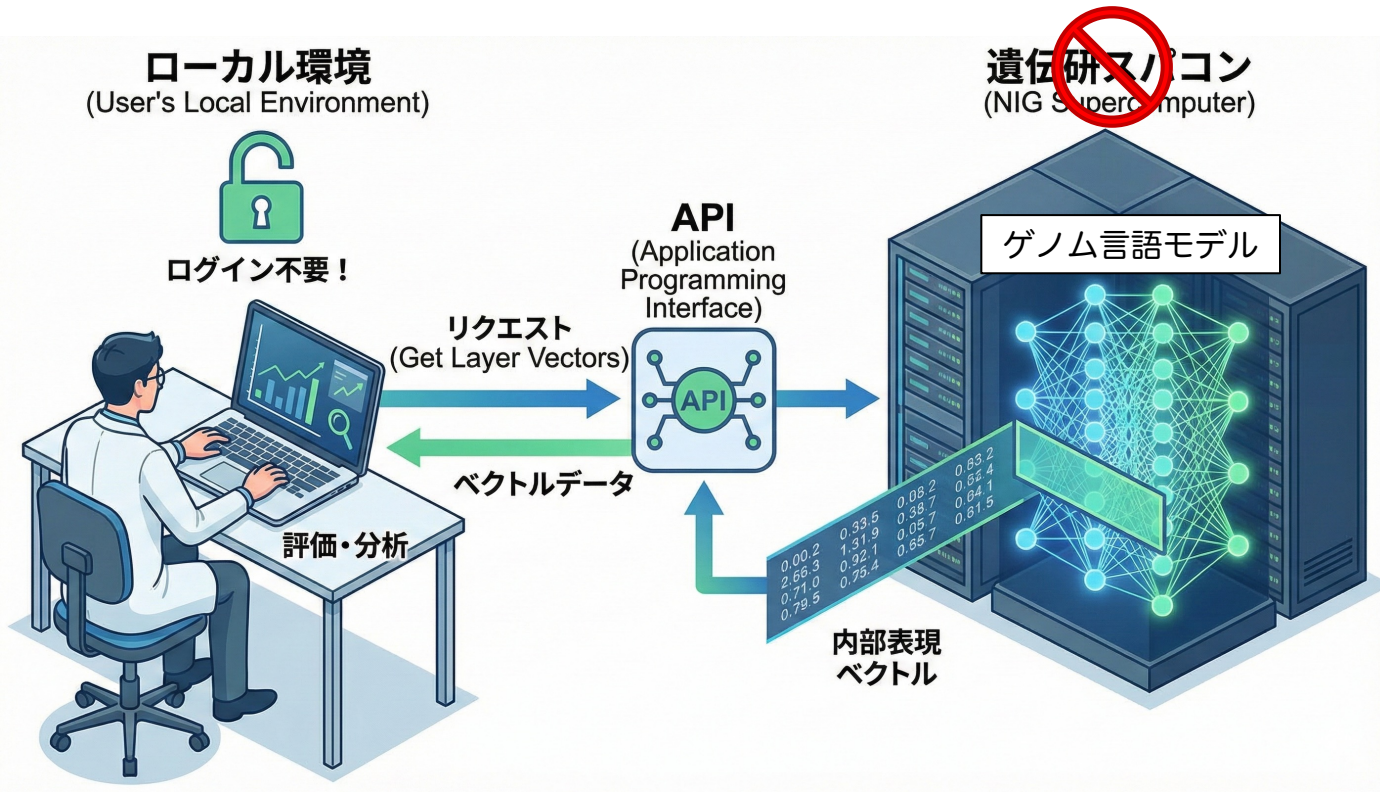


Evo2系統タグ処理コードの一部

```
220 # Process one EOD-free segment using the O1 logic.
221 def process_segment(seg_seq: torch.Tensor) -> torch.Tensor:
222     seg_len = seg_seq.size(0)
223     seg_mask = torch.ones(seg_len, device=device, dtype=torch.int)
224     # Identify positions of terminal tag (pipe)
225     pipe_pos = (seg_seq == terminal_tag_char).nonzero(as_tuple=True)[0].cpu().tolist()
226     if len(pipe_pos) == 0:
227         # If no pipe exists and any token is a known tag char or not valid DNA,
228         # mask the entire segment.
229         if not region_all_valid_or_control(seg_seq):
230             seg_mask.zero_()
231         return seg_mask
232
233     # Always mask the pipe positions.
234     seg_mask[pipe_pos] = 0
235
236     # Does tag start before the first pipe? This determines the starting state of our state machine.
237     first_pipe = pipe_pos[0]
238     if first_pipe >= 0 and first_pipe < seg_len - 1:
239         # fastest check is to look at the first token after the pipe, if it is a 'd' then the
240         # tag starts _after_ the pipe, otherwise it starts before.
241         next_tok = seg_seq[first_pipe + 1].item()
242         if next_tok == first_taxonomy_prefix_token:
243             # 'd' character for domain, which is the first part of a phylo tag.
244             # tag starts after the pipe.
245             is_tag = False
246         else:
247             # tag starts before the pipe.
248             is_tag = True
249     else:
250         # The sequence ends with a pipe, so just check everything before the pipe and return the seg mask
251         assert first_pipe == seg_len - 1
252         # The sequence ends with a pipe, so just check everything before the pipe.
253         if region_all_valid_or_control(seg_seq[:first_pipe]):
254             return seg_mask # Pipe pos has already been masked
255         else:
256             seg_mask[:first_pipe] = 0
257     return seg_mask
```

# モデル評価（性能計測、応用）チームの取り組み

## HPCクラウド（H100x16）



Evo2 40B / 7Bモデルの推論APIを開発し、評価チームで運用中。  
配列生成と、任意レイヤーを指定した内部表現の取得が可能。

抽出ブロック例：

```
blocks.0.pre_norm
blocks.0.post_norm
blocks.0.filter
blocks.0.projections
blocks.0.out_filter_dense
blocks.0.mlp
...
blocks.3.inner_mha_cls
blocks.3.inner_mha_cls.Wqkv
blocks.3.inner_mha_cls.inner_attn
...
blocks.49.mlp.act
blocks.49.mlp.l1
blocks.49.mlp.l2
blocks.49.mlp.l3
```

- ・ 下流タスク（遺伝子アノテーションなど）に適した内部表現は？
- ・ SAEによる解釈性研究は本当に機能するのか？何が表現されているのか？