ゲノム言語モデル
genome Language Model

バイオ生成AI（gLM）の研究動向

東　光一
国立遺伝学研究所

識別モデル（Discriminative model）

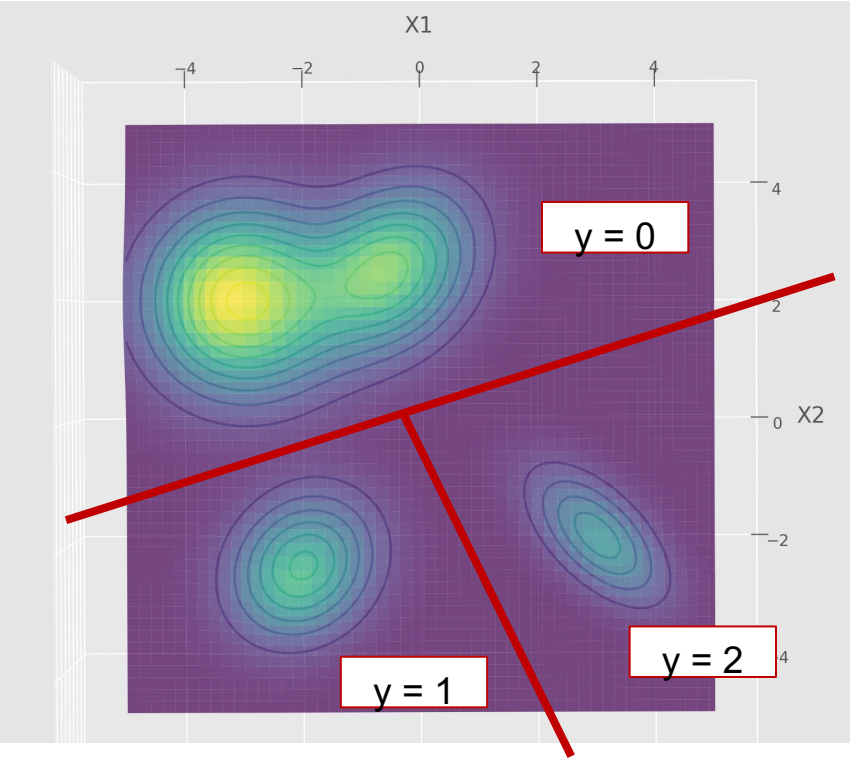
条件付き確率 P(y | X) のモデル化

分類、回帰、スコアリング

Encoder-only transformer (Masked language model)

尤度 P(X)の厳密計算は不可能

埋め込みの利用（表現学習）



生成モデル（Generative model）

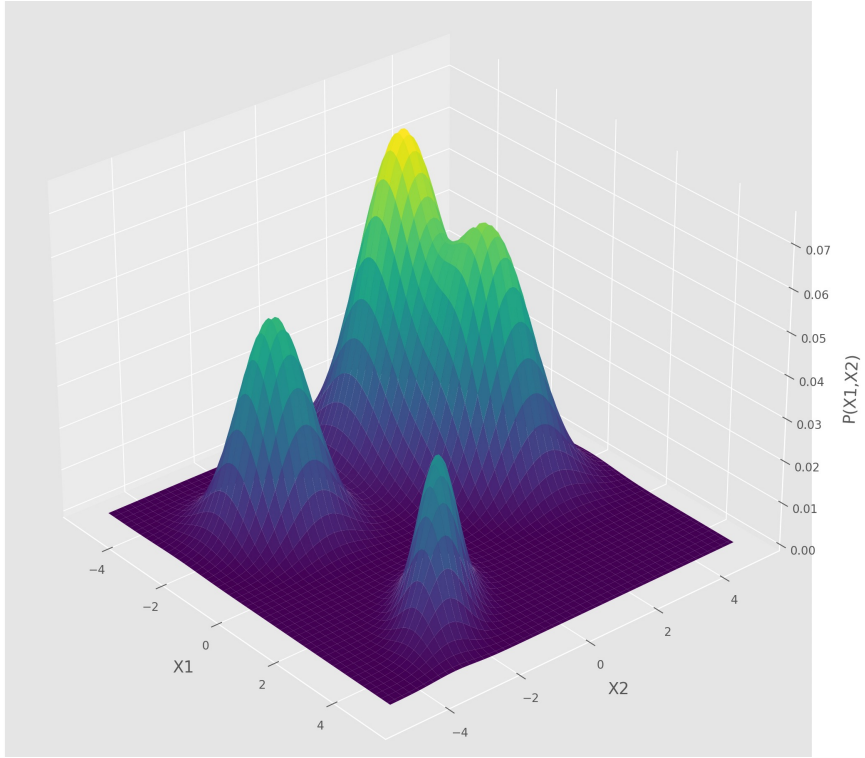
同時確率 P(X) のモデル化（密度推定）

データ分布からのサンプリングによる生成

Decoder-only (自己回帰), Encoder-Decoder (VAE, 拡散モデル)

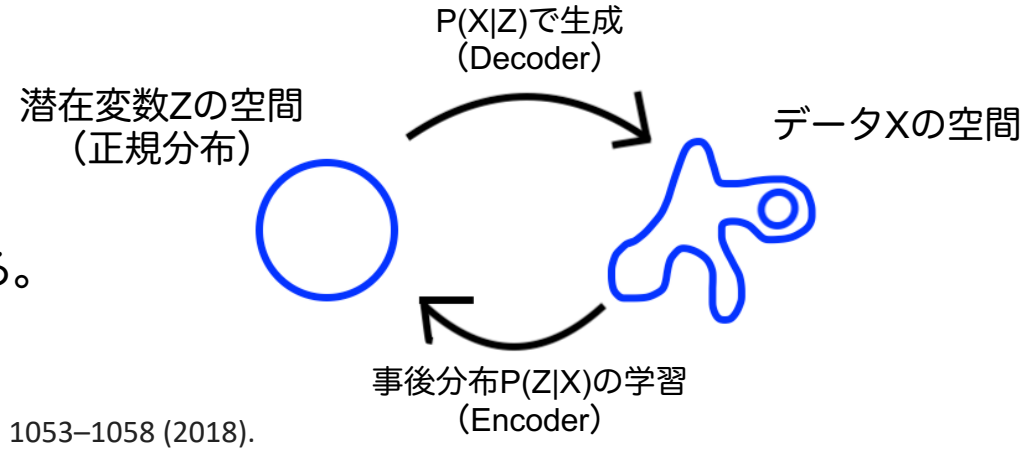
尤度 P(X) を計算可能（自己回帰は厳密、VAE・拡散はELBO近似）

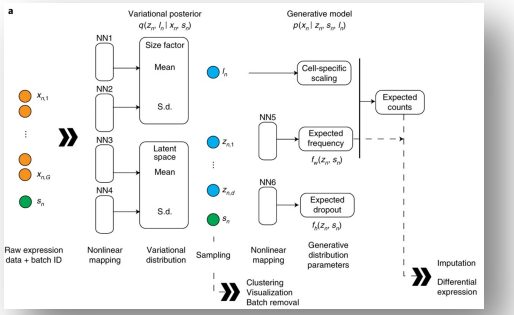
埋め込みの利用＋生成器としての活用



変分オートエンコーダ（Variational Autoencoder; VAE）

シングルセル解析でもはや必須の道具。
デノイジング、バッチ効果除去、データ統合。
scRNA-seqのUMAPは基本的に潜在変数Zで計算した距離空間で描かれる。
生成というより、表現学習、データ拡張のための利用が主目的。

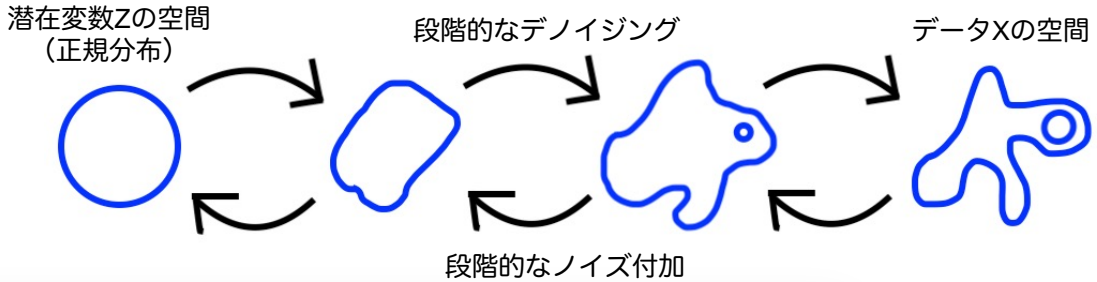


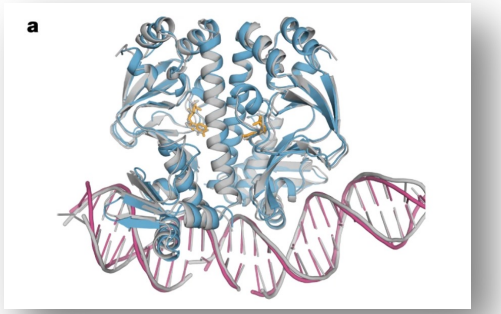


Lopez, R., Regier, J., Cole, M.B. *et al. Nat Methods* **15**, 1053–1058 (2018).
引用回数：1,921
scvi-tools関連ツール論文の総引用回数：3,198

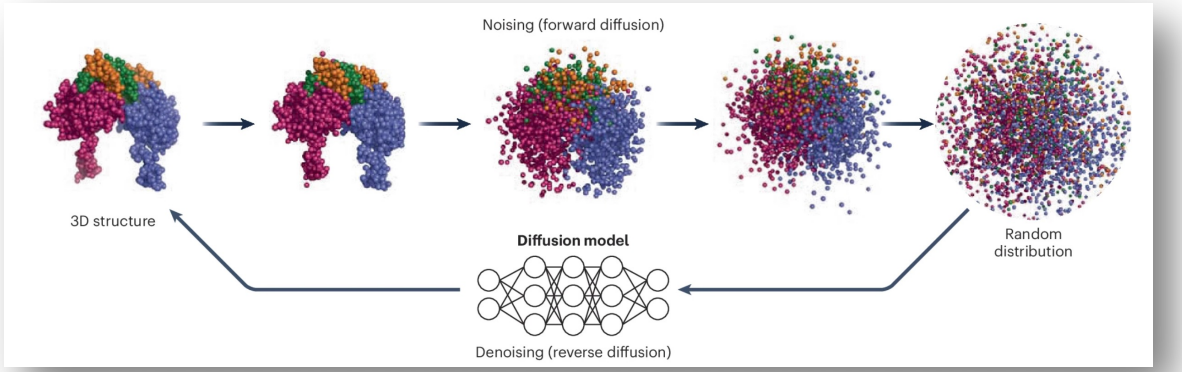
拡散モデル（Diffusion model）

AlphaFold3: Pairformer後、拡散モデルで座標生成。





Abramson, J., Adler, J., Dunger, J. *et al.* *Nature* **630**, 493–500 (2024).



Roy, R., Al-Hashimi, H.M. *Nat Struct Mol Biol* **31**, 997–1000 (2024).

自己回帰（Autoregressive）モデルによる生成プロセスのモデリング

LLM（大規模言語モデル）で採用されている手法。
P(X)を明示的に学習するのではなく、自己回帰的な生成プロセスを仮定して、次トークンの条件付き確率をモデル化。
5トークンの系列 x_1, x_2, x_3, x_4, x_5 だったら、同時確率は次のように条件確率の連鎖に分解できる。

$$P(x_1, x_2, x_3, x_4, x_5) = \mathrm{P}(x_1)\mathrm{P}(x_2|\, x_1)\, \mathrm{P}(x_3|\, x_1, x_2)\, \mathrm{P}(x_4|\, x_1, x_2, x_3)\, \mathrm{P}(x_5|\, x_1, x_2, x_3, x_4)$$

自然言語の場合、次トークンの生成確率が直前Nトークンに依存する N-gram language model

$$P(w_1, w_2, \ldots, w_k) = \prod_{i=1}^{k} P(w_i | w_{i-1}, \ldots, w_{i-N+1})$$

Context

User input (Initial Context)　Output

National Institute of Genetics, Japan is a

Context　Output

National Institute of Genetics, Japan is a prominent

Context　Output

National Institute of Genetics, Japan is a prominent research

Context　Output

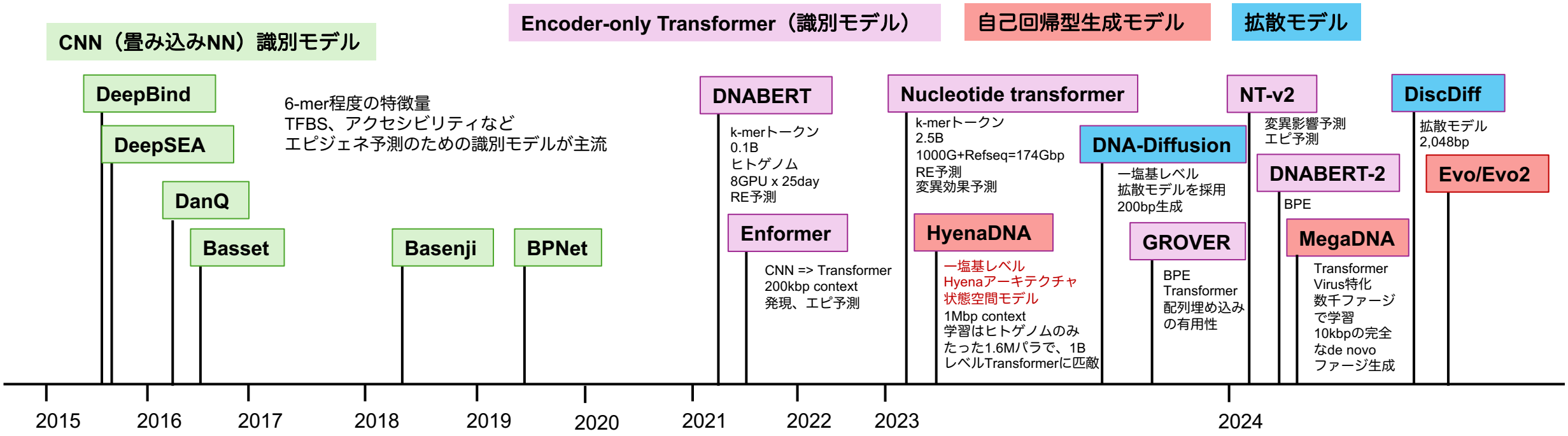
National Institute of Genetics, Japan is a prominent research institute

テキストの確率値は、トークン確率の積（トークン対数尤度の和）を計算すれば簡単に出せる。

逐次生成なので、文章全体の表現（潜在ベクトル）を持たない。
なのでLLMでテキスト埋め込みを計算するときは、
末尾トークンの最終レイヤー出力ベクトルまたはトークンごとの平均をとるなど、非自明でやや強引な計算が必要。

ゲノム基盤モデル：ゲノム配列（DNA配列）のモデリング

かつてはk-merを特徴量とした識別モデルが多かった。
近年、BPEまたはバイトレベル（一塩基レベル）生成モデルが次々に提案されている。



- RNNが主流
- word2vecなど単語埋め込みの研究
- Attentionの初提案（Bahdanau 2014）

- seq2seqによるニューラル翻訳
- AttentionベースのRNN

- Attention is all you need論文
- Transformerの提案

- BERT
- 大規模コーパスによる基盤モデル

- GPT-2発表

- GPT-3
- Language models are few-shot learners論文
- スケーリング則論文
- 基盤モデルの万能性

- 画像生成における拡散モデルの流行
- GPT-3.5

- ChatGPT
- GPT-4
- LLMの爆発的な普及

- マルチモーダル
- 推論モデルの登場（OpenAI o1/o3, DeepSeek-R1）

同時期のNLP（自然言語）モデルの状況

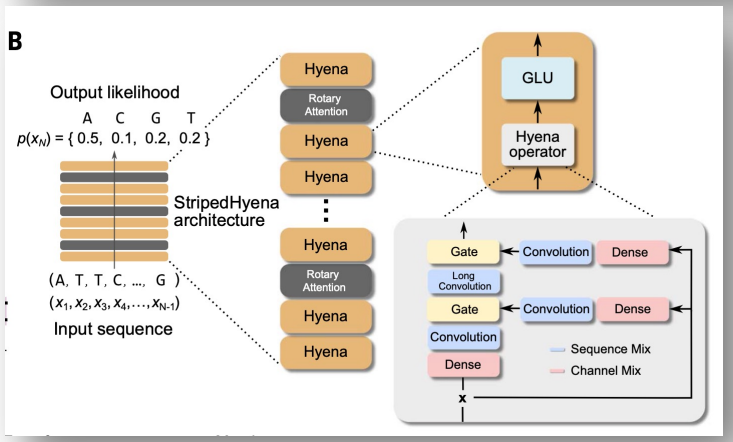
Evo：ゲノム基盤モデル（一塩基レベルDNA配列生成モデル）

RESEARCH ARTICLE

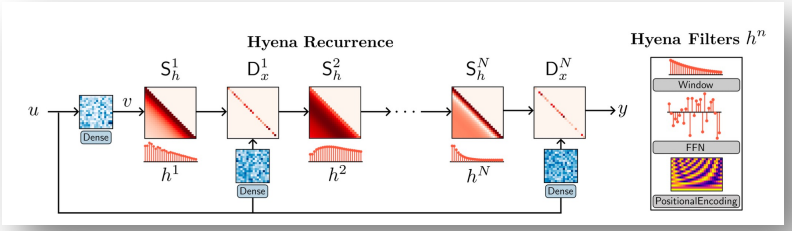
GENERATIVE GENOMICS

Sequence modeling and design from molecular to genome scale with Evo

Eric Nguyen[1,2]†, Michael Poli[3,4]†‡, Matthew G. Durrant[1]†, Brian Kang[1,2]†, Dhruva Katrekar[1]†, David B. Li[1,2]†, Liam J. Bartie[1], Armin W. Thomas[5], Samuel H. King[1,2], Garyk Brixi[1,6], Jeremy Sullivan[1], Madelena Y. Ng[7], Ashley Lewis[8], Aaron Lou[3], Stefano Ermon[3,9], Stephen A. Baccus[10], Tina Hernandez-Boussard[8], Christopher Ré[3], Patrick D. Hsu[1,11]*, Brian L. Hie[1,5,12]*



Nguyen *et al.* (2024)



Poli *et al.* (2023)

StripedHyenaアーキテクチャのバイトレベル生成
1トークン＝1塩基。
パラメータサイズ：7B
コンテキストサイズ：131kbp
つまり一塩基生成に直前13万bpのコンテキストを考慮できる。

自己回帰型生成モデルなので、モデルが最適化する対象は前述の通り。
実装に多くの工夫があり大規模ゲノムモデルを実現した。

(DNABERTなどの)表現学習に特化したモデルでは不可能で、
生成モデルであるEvoだからこそできる応用として、
1. DNA配列尤度差ΔlogPの計算（ΔlogPの適応度との同一視）
2. DNA配列生成
を紹介。

“Hyena”レイヤー
Attentionを明示的に計算せず、同等の計算を実現する仕組み。

Attentionは普通に計算すると計算量が$O(N^2)$なので、超長距離相関を扱えない。
Hyenaは、SSMにHiPPOを取り込んだH3、を一般化した計算手法。
RNNとCNNを組み合わせたような仕組みで、
RNNの系列処理能力とCNNの広大な受容野を同時に実現した。

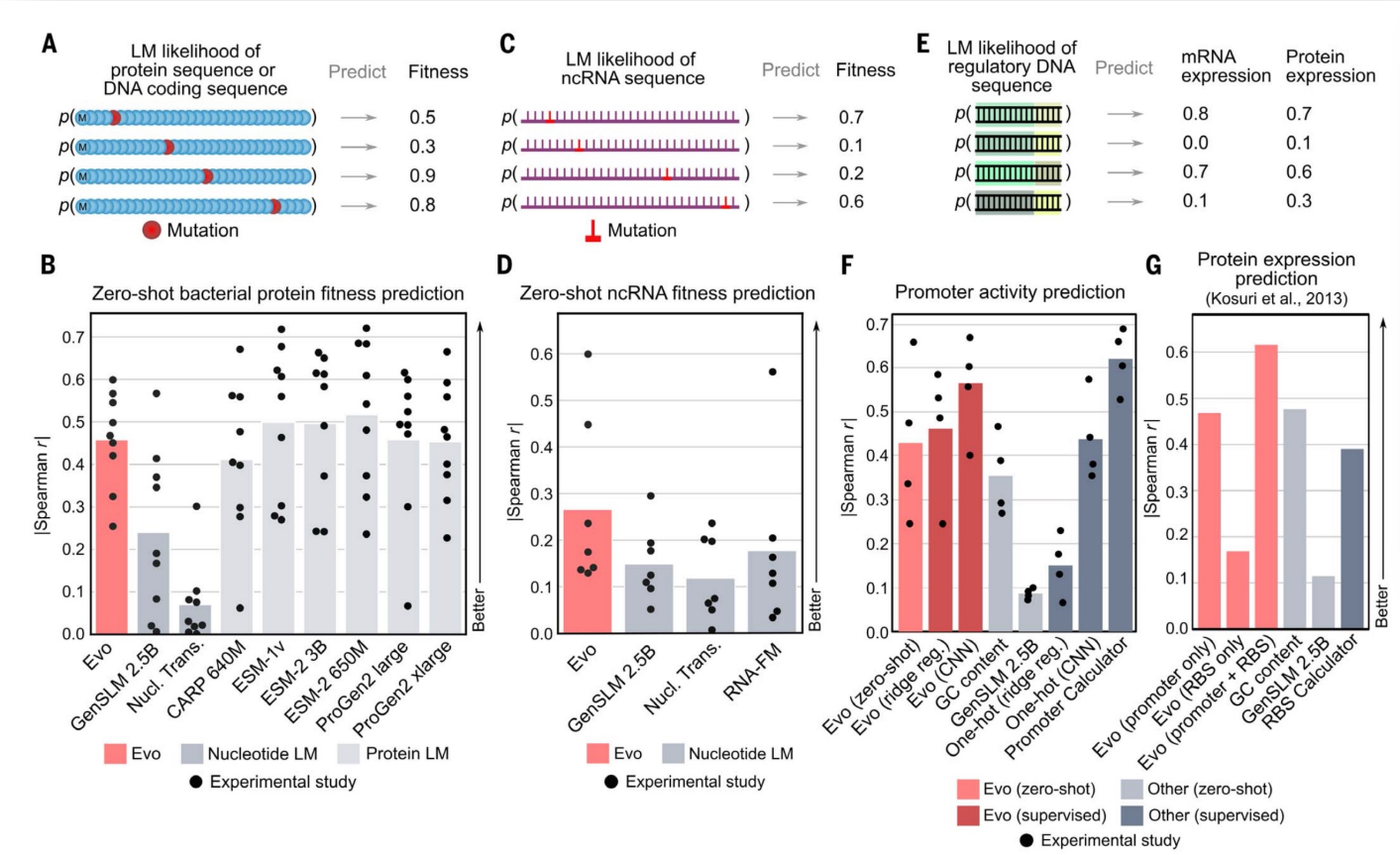
RNN/LSTMの歴史 → SSM（State space model）

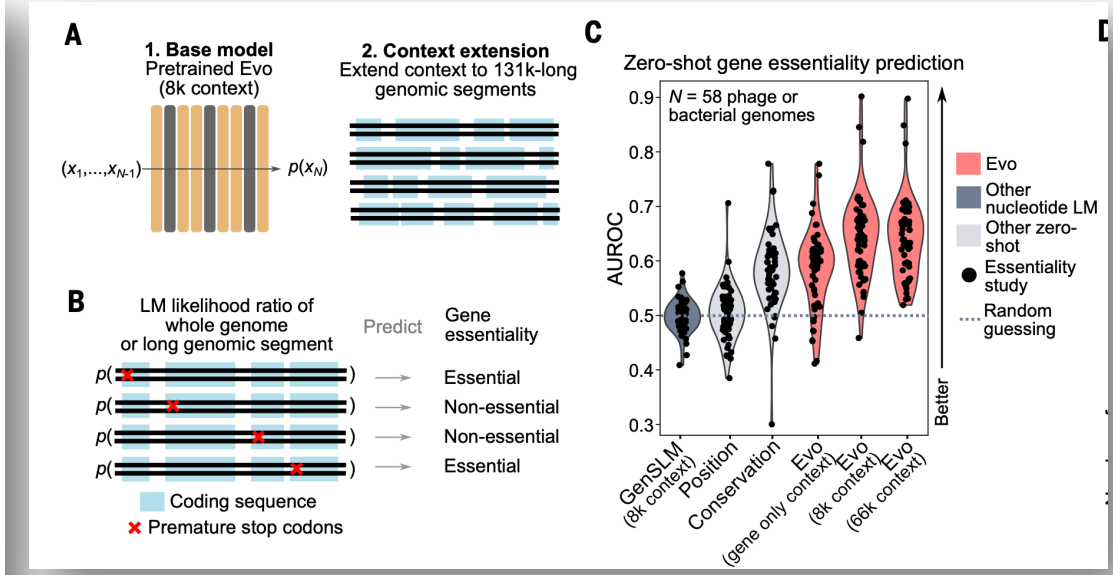
HiPPO
超長距離系列を「記憶」するために、過去のシグナル履歴をルジャンドル多項式で近似することで圧縮

SSM（State space model）＋HiPPO → H3
（Hungry hungry hippos）
HiPPO行列+Attention-likeな状態更新を計算するSSM

H3 → Hyena
HiPPO行列+HyenaFilter
Attention計算と実質同等な畳み込み演算を定式化
学習可能なフィルタで全トークン間計算を回避できる。畳み込みにFFTを使えるので計算量がO(NlogN)にできる。

Evo：結果・変異による適応度への影響、機能影響予測

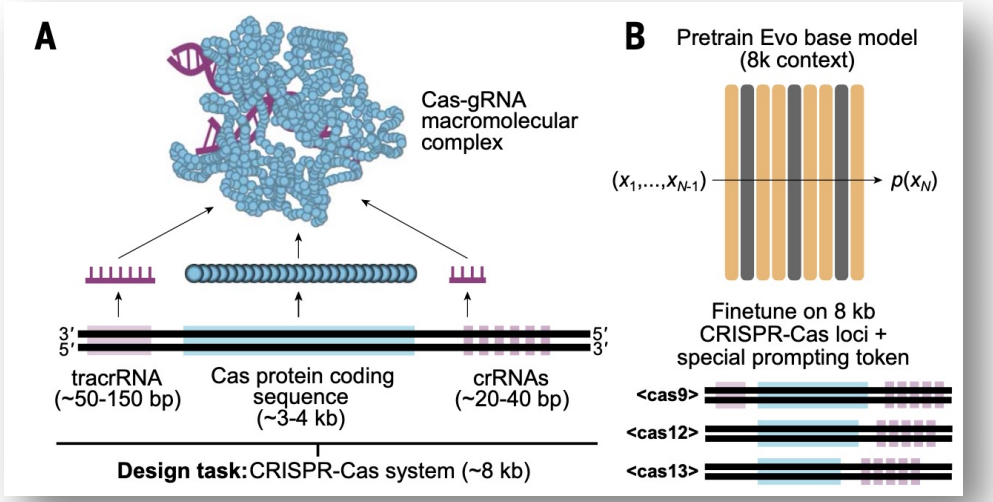




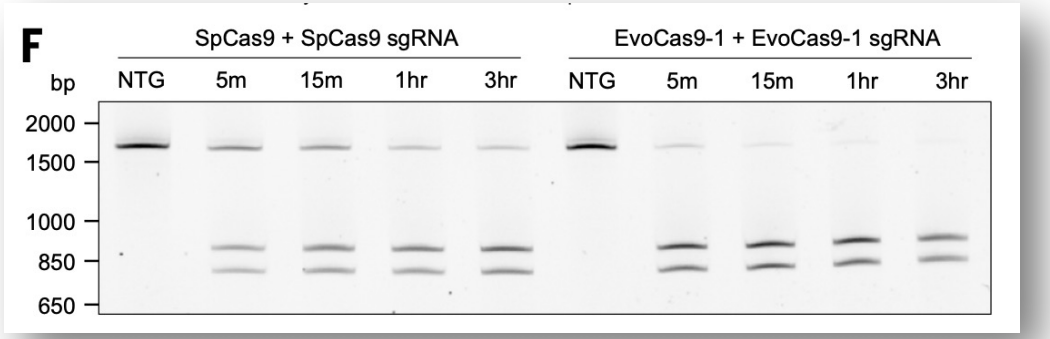
終始コドン置き換えによる尤度低下の程度によって、遺伝子の「必須性」を定量化できる。
進化的保存性による予測よりも、Evoによる尤度計算に基づく予測のほうが高精度。

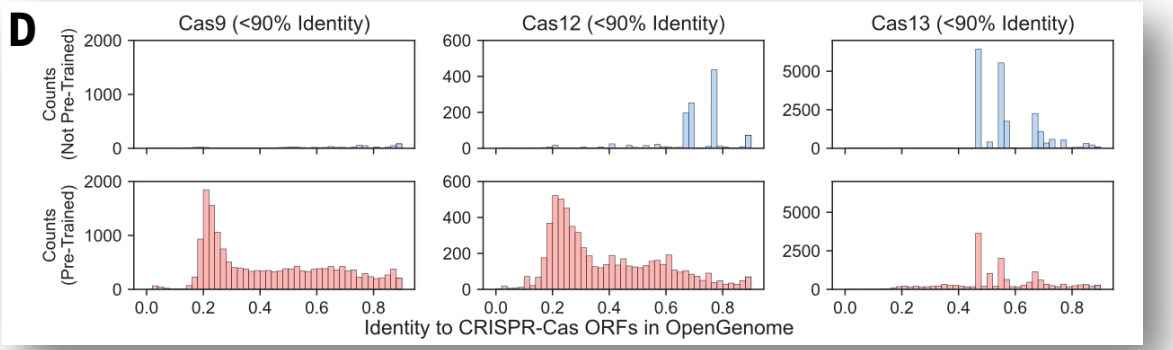
生成モデルは入力データの尤度を計算できる => 尤度を「適応度」の代替指標として活用
タンパク質適応度はDeep mutational scanning研究の値を正解として評価。（DMSコレクションの適応度評価はリガンド結合、薬剤耐性などさまざま）
プロモータ活性予測については、Evoに入力したトークンごとの潜在変数（配列長 x 埋め込み次元の行列）をインプットとして発現量を予測する、追加のリッジ回帰またはCNNレイヤーを学習。学習対象は約5,000ペアの実験データ。

Evo：結果・CRISPR-Casコンプレックスの生成

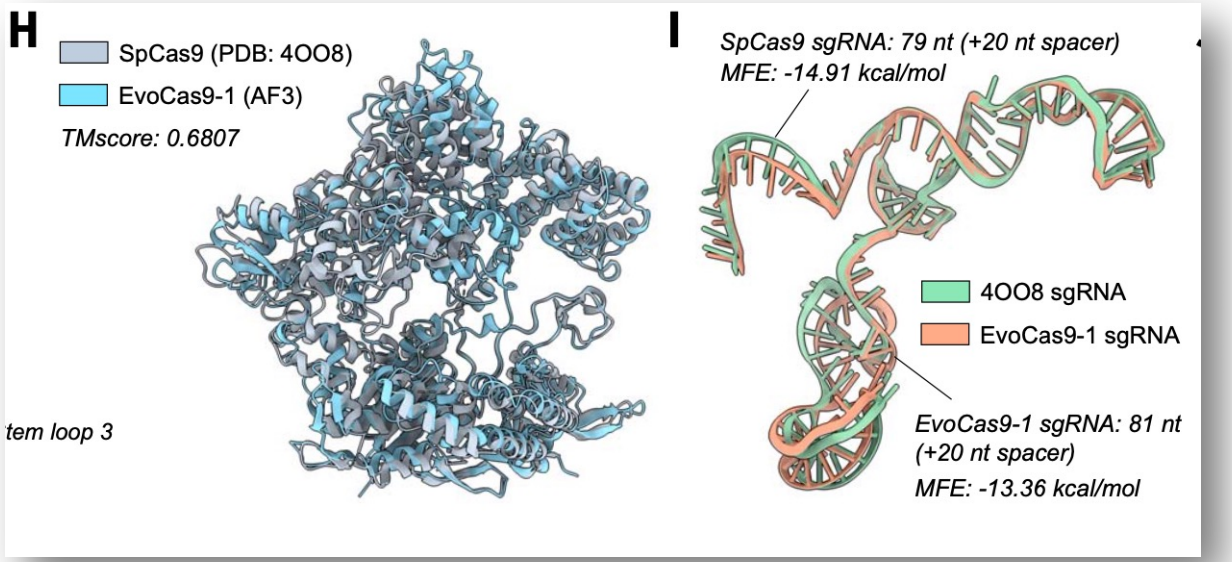


CRISPR-Cas領域（tracrRNA – Cas CDS - crRNAs）のデータセット（72,831配列）を構築してEvo-8kモデルをファインチューニング。
学習のために、Casクラスを識別する追加のクラストークン（<cas9>, <cas12>, <cas13>）を配列先頭に付加。





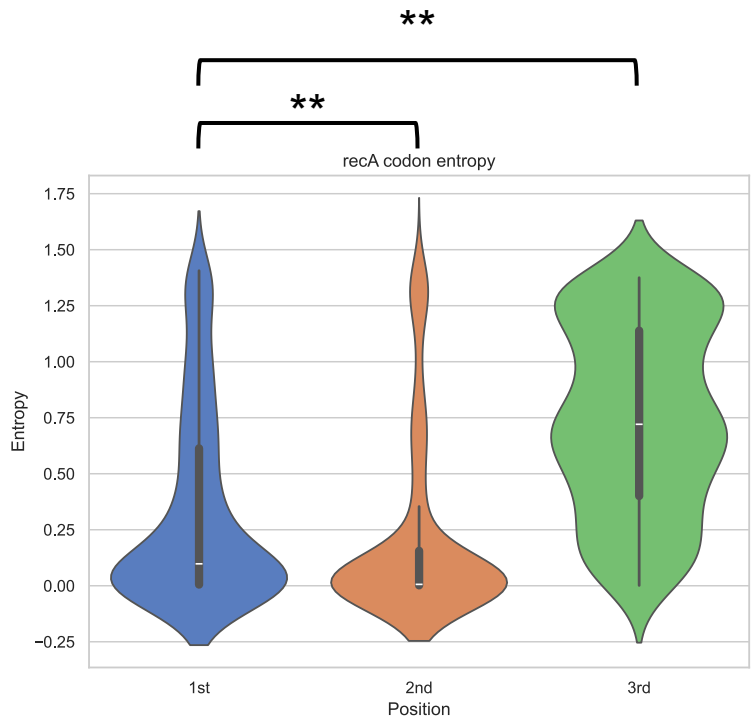
学習データとの配列類似性が40%以下の配列を含む、多様なCas配列を生成する。
Cas ORFのみの学習よりも、周辺RNAを含めた領域で学習したほうが多様になる。



生成された配列のひとつをEvoCas9-1と命名。既知のSpCas9とは73.1%の配列類似性。
特筆すべきは、同時に生成したgRNAと複合体を形成し、
化学合成したtarget DNAを対象としたin vitroの切断活性試験で活性を示す。

テスト：一塩基レベルエントロピー

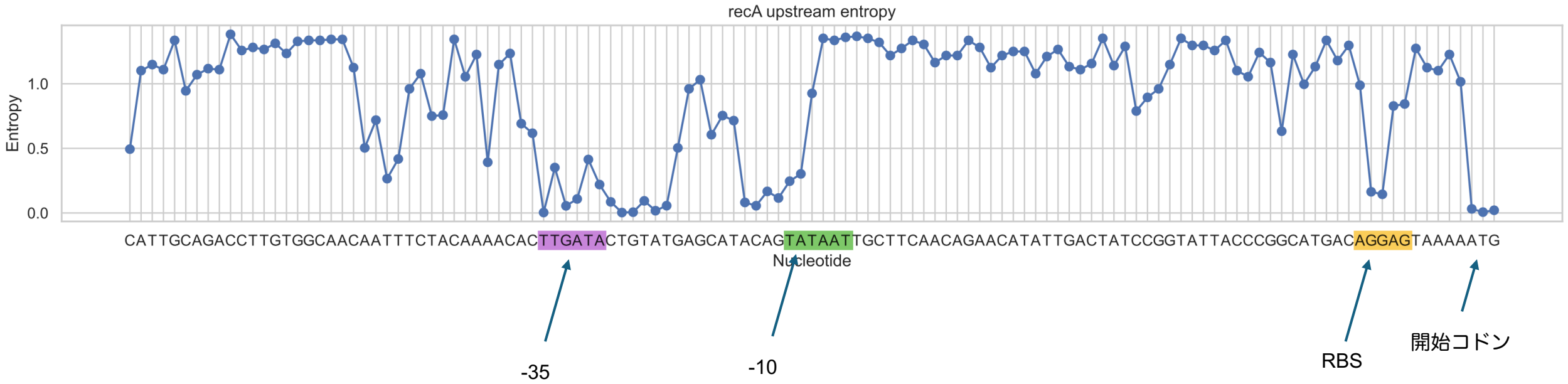
確率値が計算できるということは、その位置におけるATCG出力のエントロピーが計算できるということ。
つまり、ポジションごとの変異にかかる制約が定量できるということ。



遺伝子ごと、コドン1塩基目、2塩基目、3塩基目のエントロピーを集計。
2nd positionにおける強い制約、3rd positionにおける制約の緩和を再現している。
（アノテーション情報を与えていないことに注意）
いままで「比較ゲノム」の方法論でやってきた計算が全部このモデル上で計算できる！（かもしれない）

テスト：エントロピーによる上流制御配列検出

遺伝子開始コドン上流の制御因子（SD, -10, -35など）の進化制約をエントロピーが反映するか？

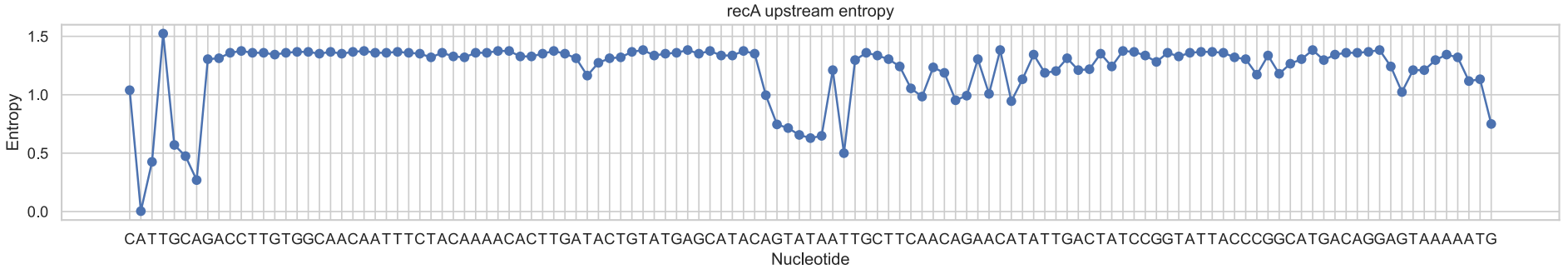


遺伝子間領域においても進化的制約を表現できてるっぽい。

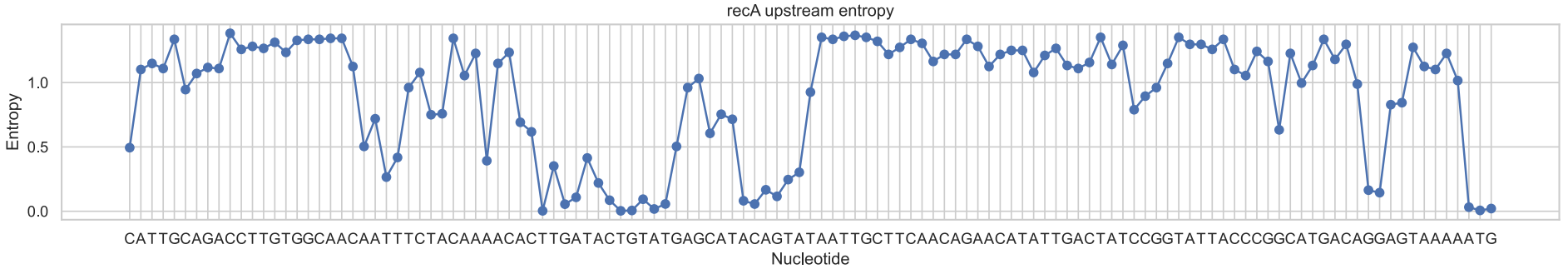
問題点：演算の方向性

確率値の推定は、「どの順番で塩基を提示するか」に依存する。
とくに遺伝子上流配列をCDSよりも先に提示する場合に顕著に値が影響を受ける。
Evoが「一方向自己回帰モデル」であることの弊害。
前述の例では実際は逆鎖を逆順に与え（逆のstrandを終始コドンから順番に提示）、
それによって終盤に登場する遺伝子上流配列が、すでに提示した遺伝子の制御に関わる、という推論ができた。
順番通り上流配列から提示してしまうと、その先にCDSが提示されるだろうことは予測できないので、
文脈の不足により上流制御配列の検出に失敗する。

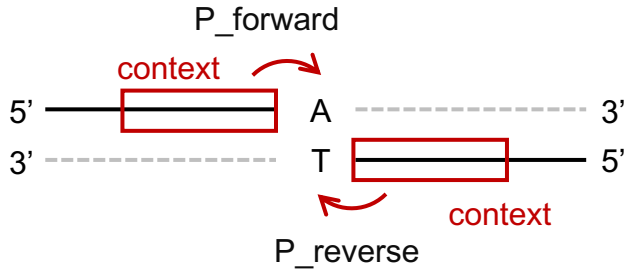
Forward演算



Reverse演算

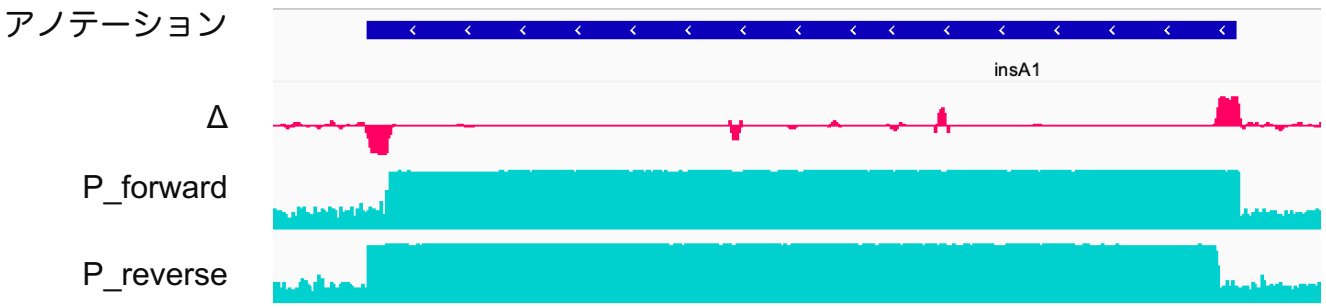


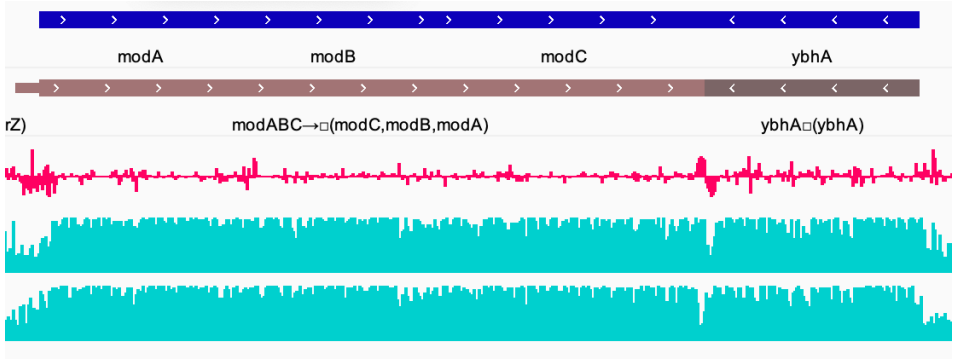
問題点：演算の方向性



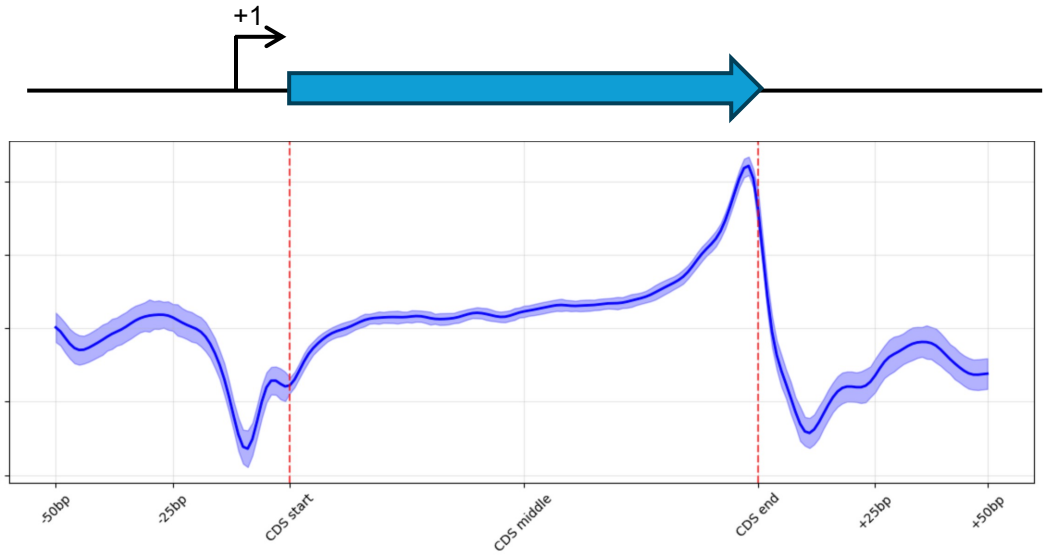
Δ = log(P_forward) – log(P_reverse)

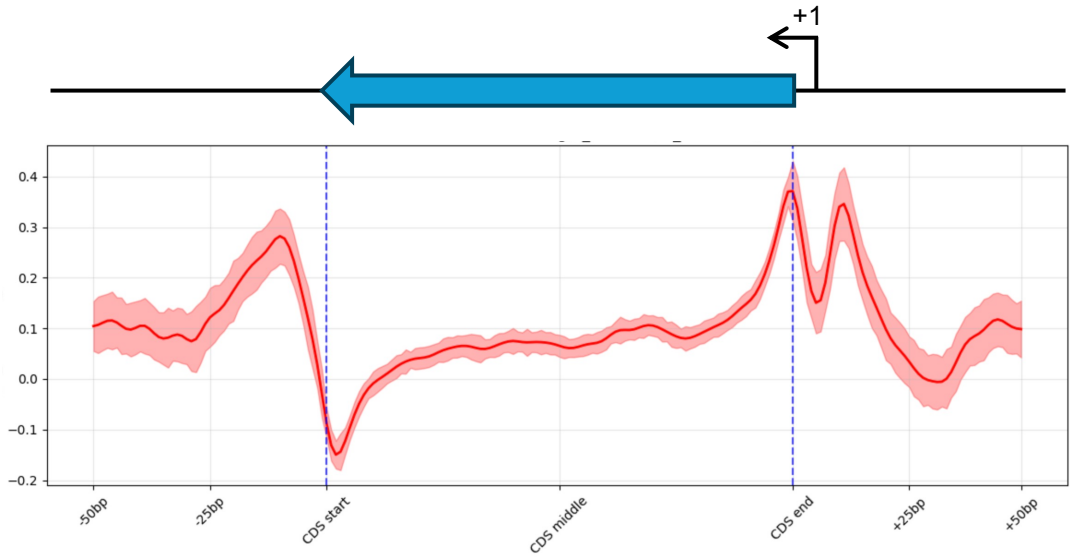
Δ>0：左方向に制約された塩基
Δ<0：右方向に制約された塩基





$log\left(\frac{P\_forward}{P\_reverse}\right)$





生成モデルを利用することの利点

Evo1/Evo2の場合

1. データ確率分布からのサンプリングで新規データを生成できる
   a. データ拡張。単純にN数を増やしてクラス不均衡を改善したり、普遍的特徴を追求できる。
   b. プライバシー保護。実データを秘匿して統計的に同等のデータセットを構成できる。
   c. データの正常な分布範囲を検討できる。
   d. 欠損値補完ができる。

- ゲノム生成
- 多様な機能タンパク質・RNAを生成

2. データを入力して、その確率値を計算できる
   a. 異常検知ができる。リアルタイムデータのモニタリング、外れ値除去によるデータクレンジングなど。
   b. モデル自体の性能評価ができる（複数モデルの対数尤度比較）
   c. データのセグメンテーションができる。低確率領域データにフォーカスした解析など。
   d. 介入の効果測定。データに修正を加えた際の確率値との比較によって、影響を予測できる。
   e. シミュレーションモデルのスコア関数として使う。

- 尤度に基づく適応度評価
- バリアント効果検証

3. データの埋め込みを使って解析できる
   a. 表現学習。データの（表面上のパターンではなく）本質的な特徴を自動抽出できる。
   b. 埋め込みを使った識別モデルのトレーニング。
   c. ノイズや変換に頑健な特徴の利用。scRNA-seqのデノイジング、バッチ効果除去。
   d. 連続値潜在空間における補間。離散データを連続値として扱える。大量の統計学的道具を使える。
   e. データ間の関係性の定量化。類似度などの関係性を潜在空間における距離やベクトルとして表現できる。
   f. マルチモダリティへの拡張。異なる種類のデータをすべて連続値のベクトルに変換して統合しやすくする。

- 潜在変数を土台にした回帰モデルで配列の性質を予測

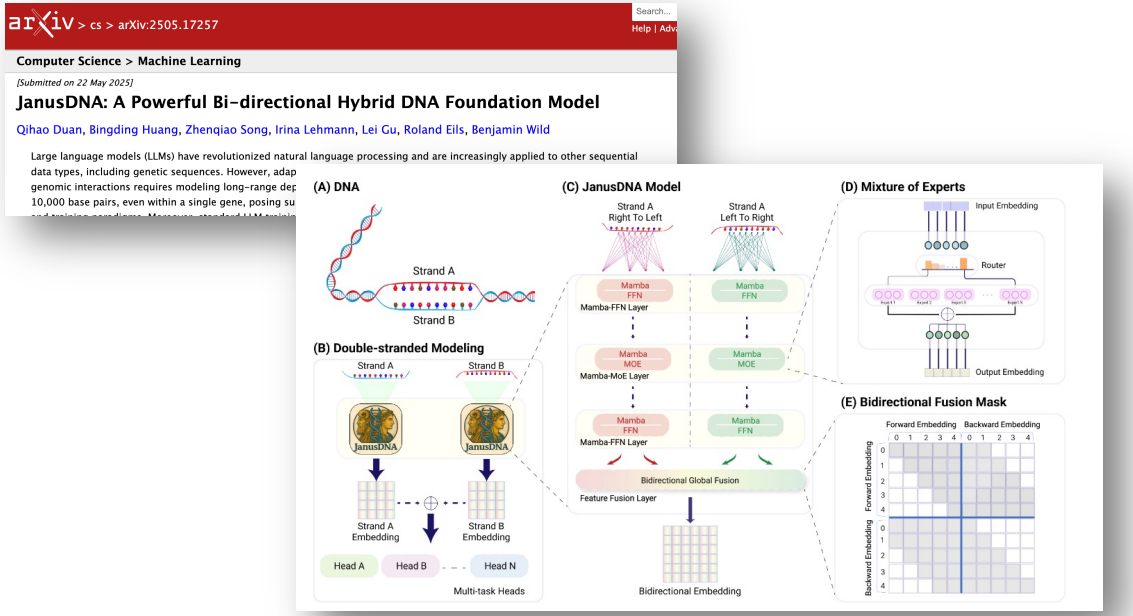
4. 研究目的に応じてモデルを拡張できる
   a. 条件付き生成への拡張
   b. 転移学習（ドメイン適応）。普遍的特徴を大規模データで学習し、サブセット固有の特徴を追加学習する。
   c. 解釈性向上のための拡張。Attentionを取り出してデータ間の関係性を可視化するなど。
   d. 人間によるフィードバック。実験結果からのフィードバックを取り込んだモデル改善など。

- 種トークン、クラストークンによる配列生成の制御
- スパースオートエンコーダによる解釈。ゲノムアノテーション

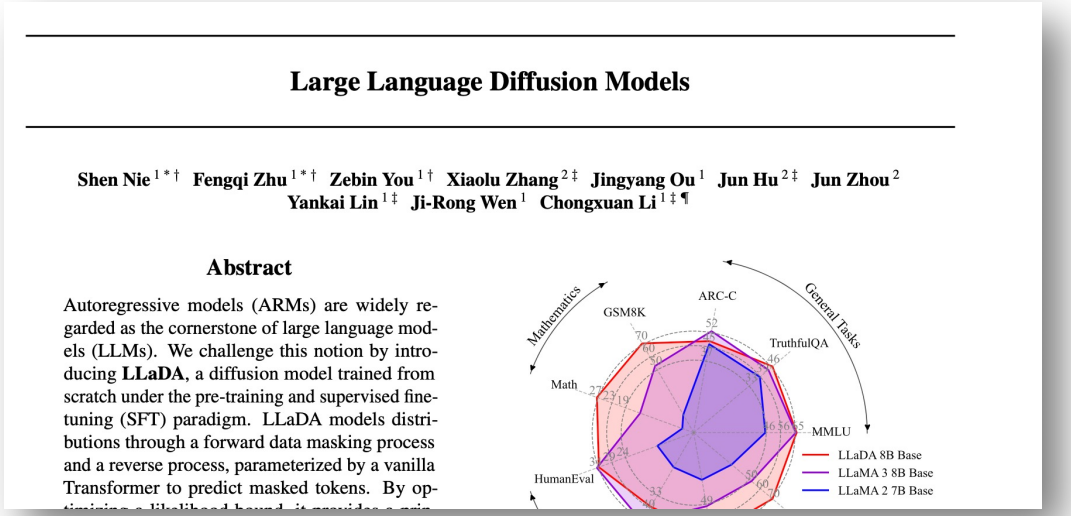
ゲノム言語モデルの開発・拡張の方向性

本当にDNAも自己回帰型モデルでいいのか？
5’ -> 3’ で複製されるとはいえ、
DNA上の情報は5’ -> 3’のみで規定されているわけではない。

両方向自己回帰などちょっと複雑な構成が必要？（cf. JanusDNA）



最近、拡散モデルを基盤のアーキテクチャとしてもLLMが
構成可能であることが示された（拡散言語モデル）

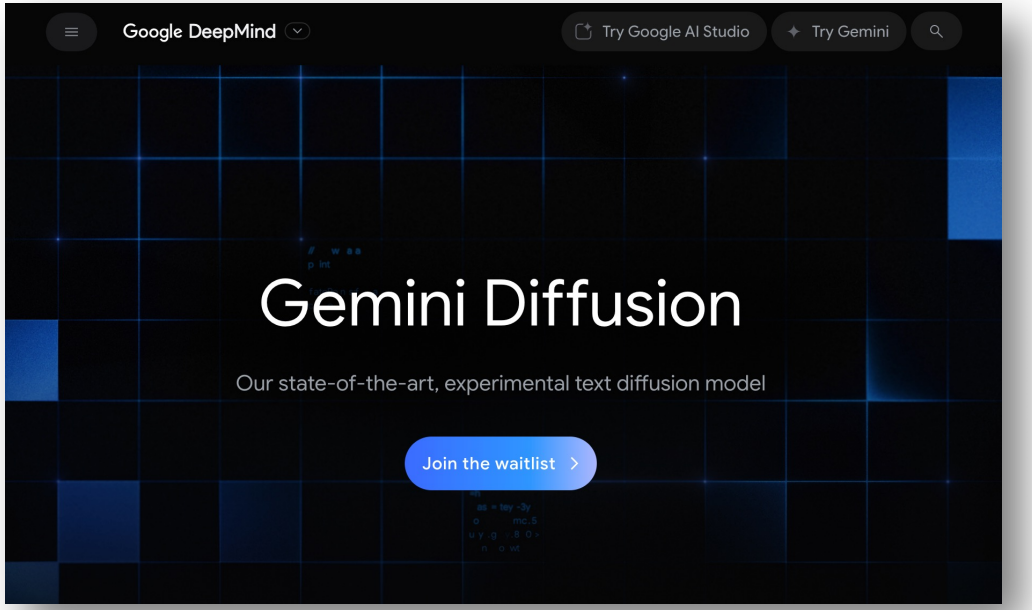


Large Language Diffusion Models

Shen Nie [1 * †] Fengqi Zhu [1 * †] Zebin You [1 †] Xiaolu Zhang [2 ‡] Jingyang Ou [1] Jun Hu [2 ‡] Jun Zhou [2]
Yankai Lin [1 ‡] Ji-Rong Wen [1] Chongxuan Li [1 ‡ ¶]

Abstract

Autoregressive models (ARMs) are widely regarded as the cornerstone of large language models (LLMs). We challenge this notion by introducing **LLaDA**, a diffusion model trained from scratch under the pre-training and supervised fine-tuning (SFT) paradigm. LLaDA models distributions through a forward data masking process and a reverse process, parameterized by a vanilla Transformer to predict masked tokens. By op-



Nie, Shen, et al. "Large Language Diffusion Models." *arXiv:2502.09992* (2025).

補足資料

Evo1：学習の概要

学習データ：著者らが構築した OpenGenome dataset

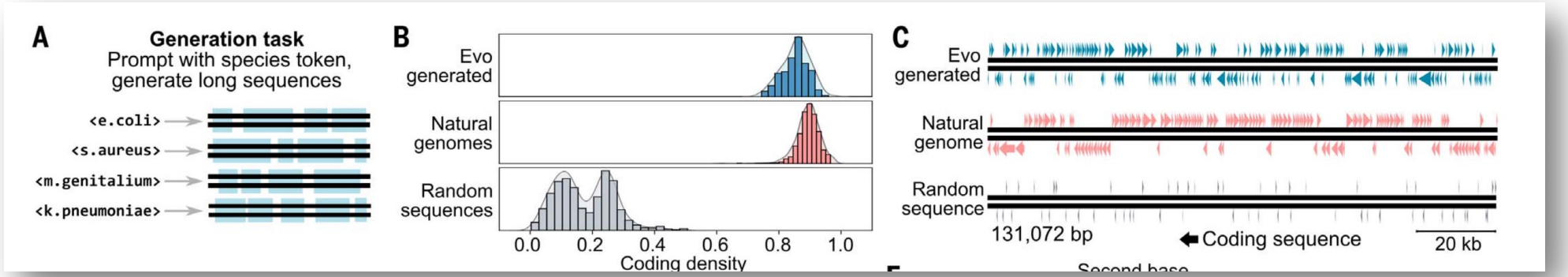
- Genome Taxonomy Database (GTDB) v.214 の代表ゲノム（1 / species）
- IMG/VR v4 database からキュレーションされたファージ配列（1 / PTU）
- IMG/PR databse からキュレーションされたプラスミド配列（1 / vOTU）
- 合計270万配列、3,000億塩基
- データはすべて生のDNA配列で、いっさいのアノテーションを含まない点に注意。

学習手順：2段階で学習

1. 1st stage: 8k サイズコンテキスト, 64枚 NVIDIA H100 で2週間の計算
2. 2nd stage: 131k サイズコンテキスト, 128枚 NVIDIA H100 で2週間の計算

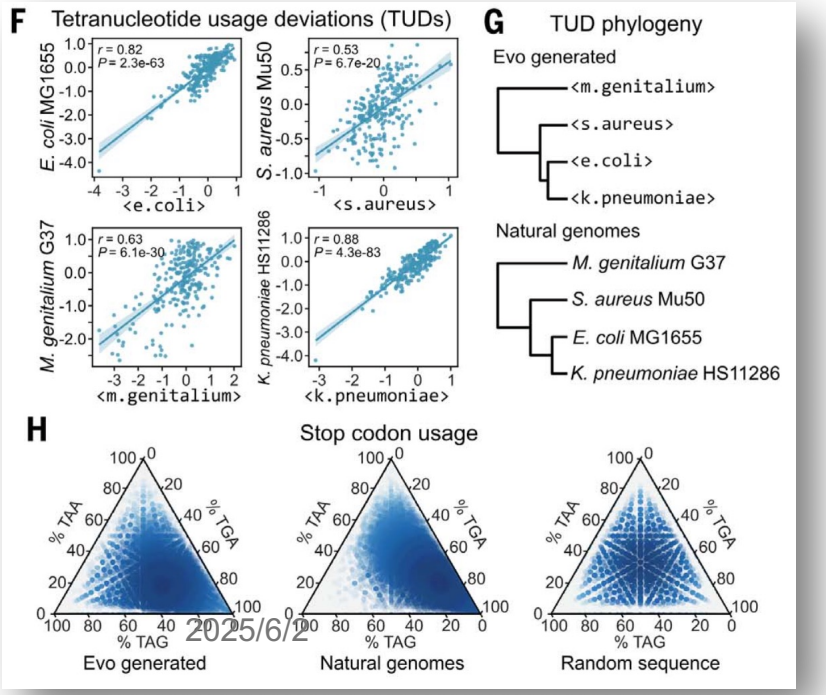
- いずれも、学習データセットに対しておよそ1.3エポックの計算となる

Evo1：バクテリアゲノム生成



事前学習時、GTDBの種ゲノムごとに以下のような「種トークン」を配列先頭に付加して、「系統プロンプトに基づく生成」を可能としておく。
|d__Bacteria;p__Pseudomonadota;c__Gammaproteobacteria;o__Enterobacterales;f__Enterobacteriaceae;g__Escherichia;s__Escherichia||

ゲノム生成の検証として 1Mbp の配列を生成させ、生成された配列について遺伝子予測、CheckMクオリティ評価を行った。
CDS密度は現実のゲノムと同様のパターンをとる。
生成されたタンパク質の多くが、一般的な細菌の持つ機能タンパク質と構造が類似。各ゲノムで、すべての標準アミノ酸に対応するtRNAも生成されていた。



4-mer頻度パターンや、終始コドンの使用パターンも現実のゲノムと類似。

一方で、存在するはずのシングルコピーマーカー遺伝子があまり含まれていない、
rrnオペロンが足りていないなど、完全ゲノムとしては不自然な特徴もある。

Evo2：全生物ドメインのゲノム基盤モデル

New Results

Follow this preprint Previous

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

Posted February 21, 2025.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A. Gonzalez, Samuel H. King, David B. Li, Aditi T. Merchant, Mohsen Naghipourfar, Eric Nguyen, Chiara Ricci-Tam, David W. Romero, Gwanggyu Sun, Ali Taghibakshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K. Wang, Etowah Adams, Stephen A. Baccus, Steven Dillmann, Stefano Ermon, Daniel Guo, Rajesh Ilango, Ken Janik, Amy X. Lu, Reshma Mehta, Mohammad R.K. Mofrad, Madelena Y. Ng, Jaspreet Pannu, Christopher Ré, Jonathan C. Schmok, John St. John, Jeremy Sullivan, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B. Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Thomas McGrath, Kimberly Powell, Dave P. Burke, Hani Goodarzi, Patrick D. Hsu, Brian L. Hie

Download PDF

Print/Save Options

Subject Area

パラメータサイズ：7B/40B
コンテキストサイズ：1Mbp

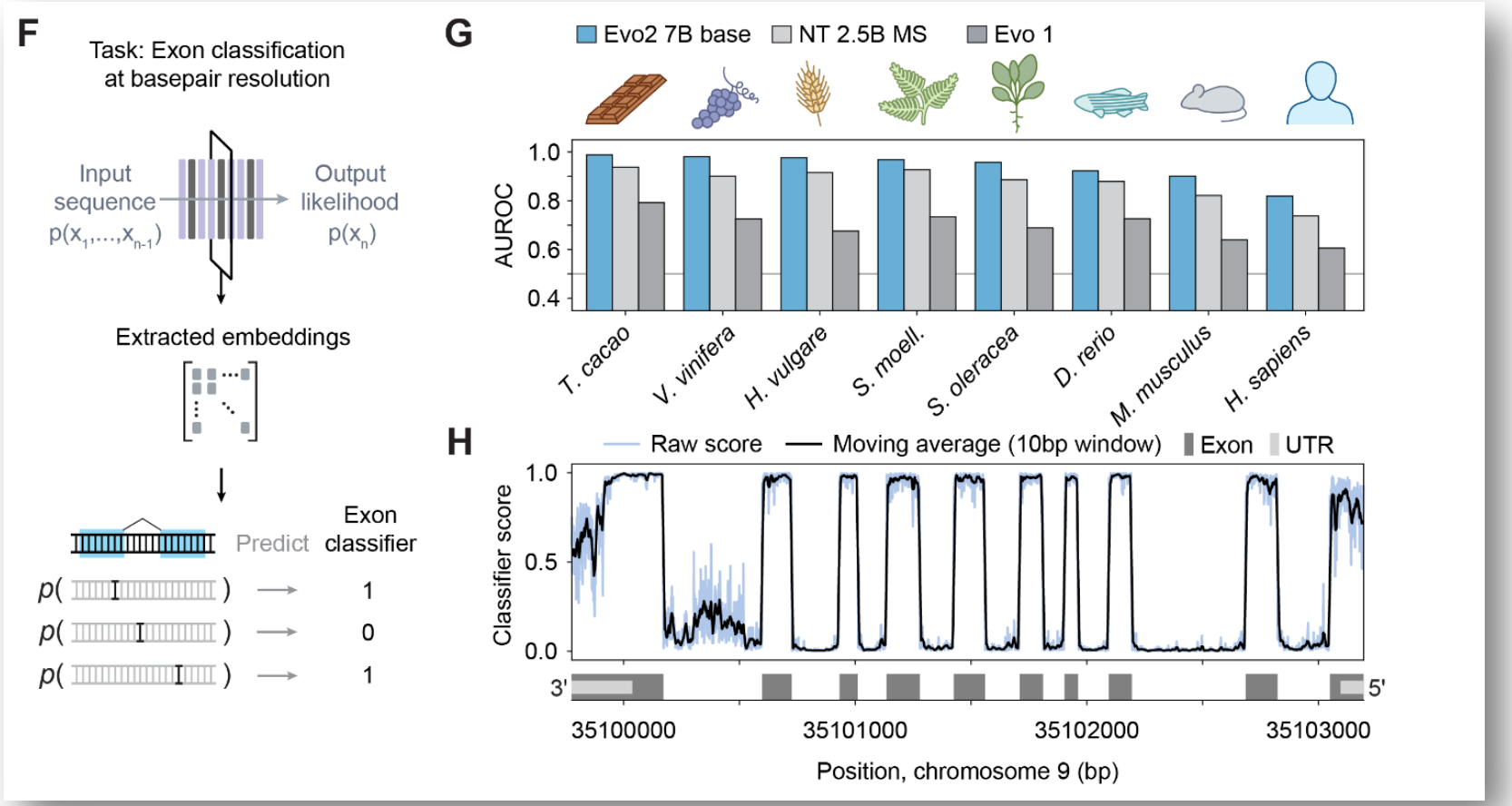
Evo1同様のHyenaアーキテクチャをベースに改良

学習データ：OpenGenome2 dataset

- Bacteria/Archaea: GTDB代表ゲノム約11万配列
- Eukaryotes: NCBI Genome. Mash距離でクラスタリングして代表ゲノムを抽出。追加のフィルタリングで最終的に15,032ゲノム
- Metagenome: NCBI, JGI IMG, MGnify, Tara Oceansなどのコンティグ、MAG
- オルガネラゲノム：NCBI Organella. 32,240ゲノム
- Evo1同様、配列先頭に系統タグを付加
- 合計8.84Tbp

NVIDIA DGX Cloudで数ヶ月の学習。
H100 GPU 2,000枚以上に相当。

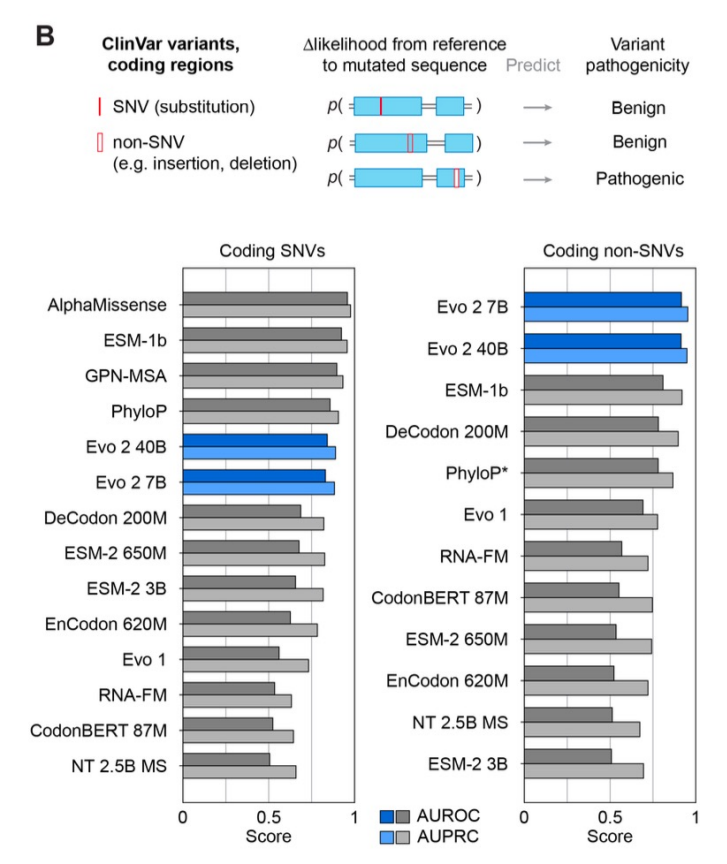
Evo2：結果・一塩基レベルエクソン/イントロン境界予測

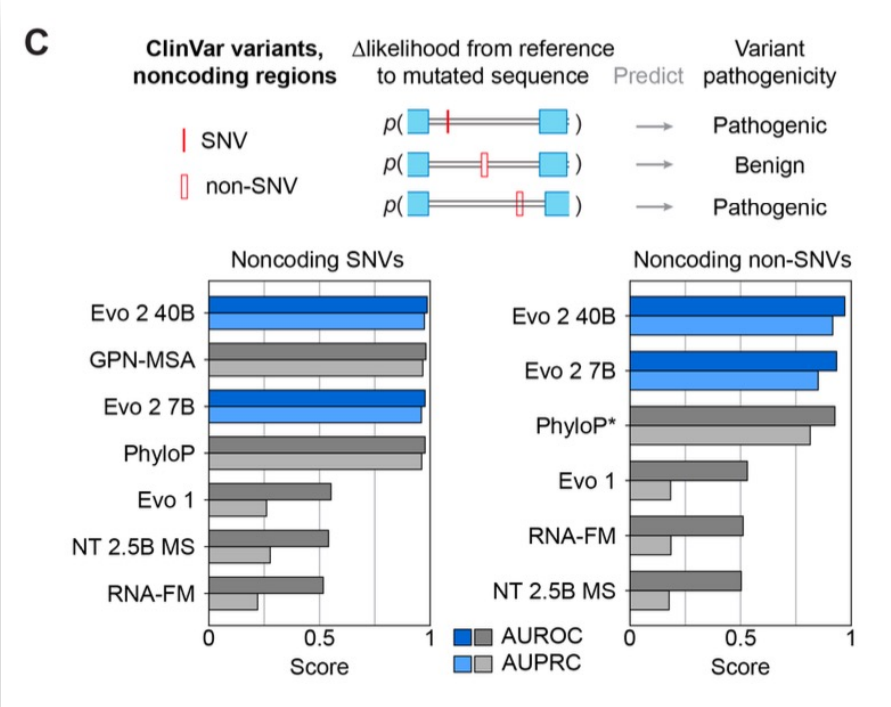


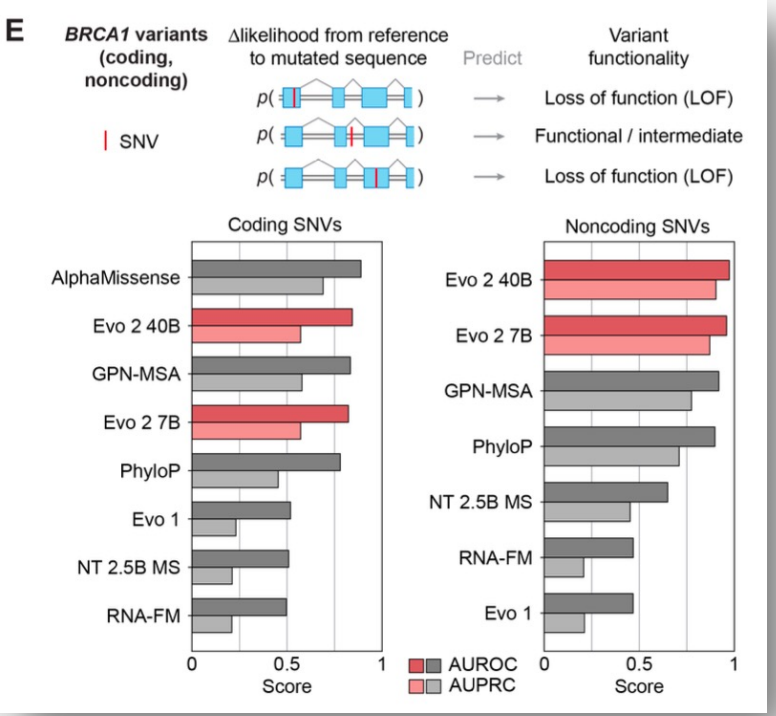
94種のゲノム配列とエクソン・イントロンアノテーション（RefSeq）を取得し、トレーニング、ハイパラ最適化、テストセットに分割。
Evo2のトークン潜在変数（塩基潜在ベクトル）を取り出して入力とし、アノテーションされたエクソン・イントロンラベルのクロスエントロピーを損失関数とするシングルレイヤーニューラルネットワーク（パーセプトロン）を構成。Hidden layerの次元は1,024.
各生物種について1,500のポジションラベルペアでトレーニング。

（論文で言及されていないが、そもそもEvo1で予測精度がそれなりに高いことがおもしろい）

Evo2：結果・ヒトバリアント影響予測



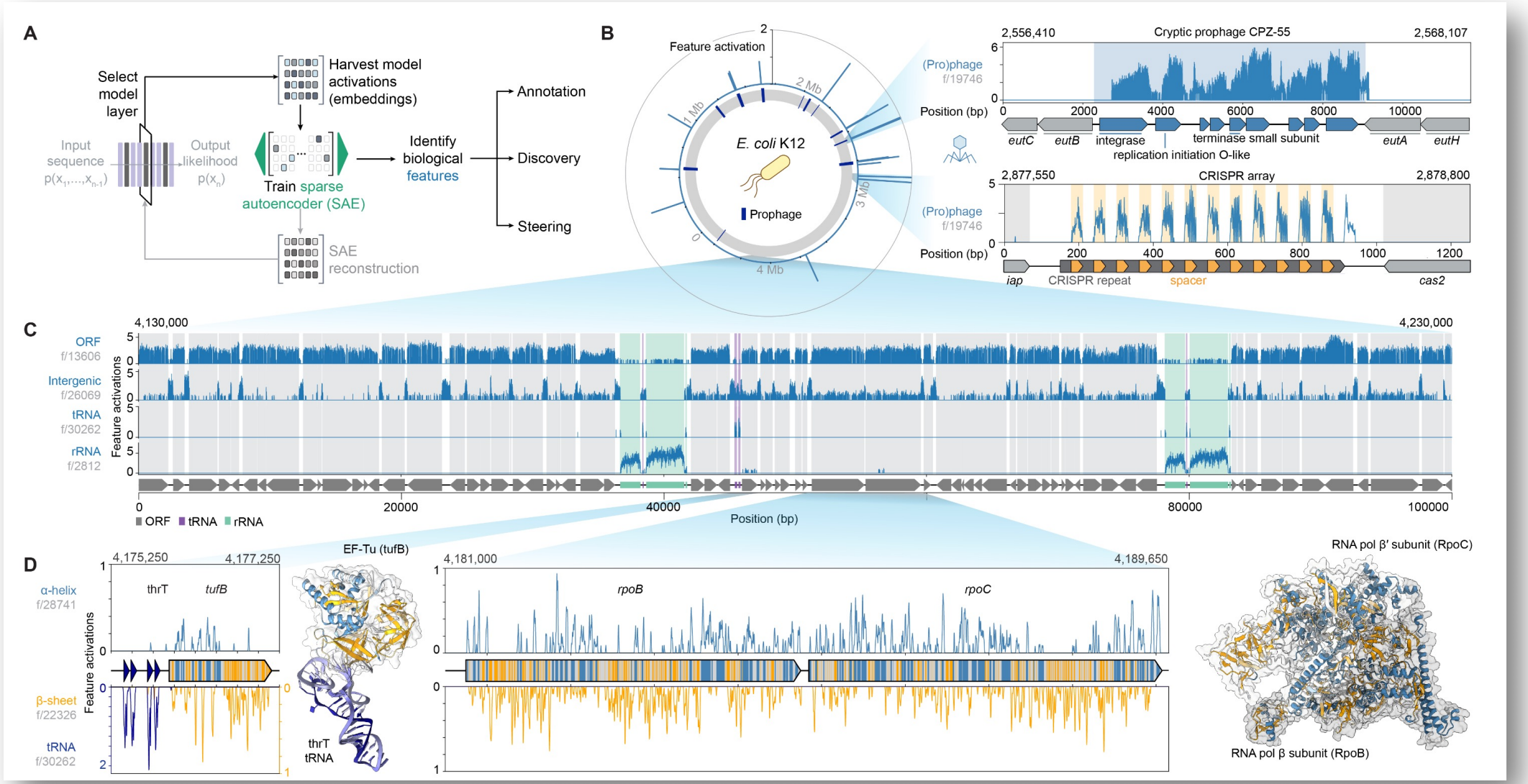




追加学習や回帰モデルの接続をいっさい使わない、ゼロショットの予測性能を評価。
つまり、DNA配列の尤度の数値のみで良性の変異か、有害な変異かを予測させる。（対数尤度差分をスコアとして利用）

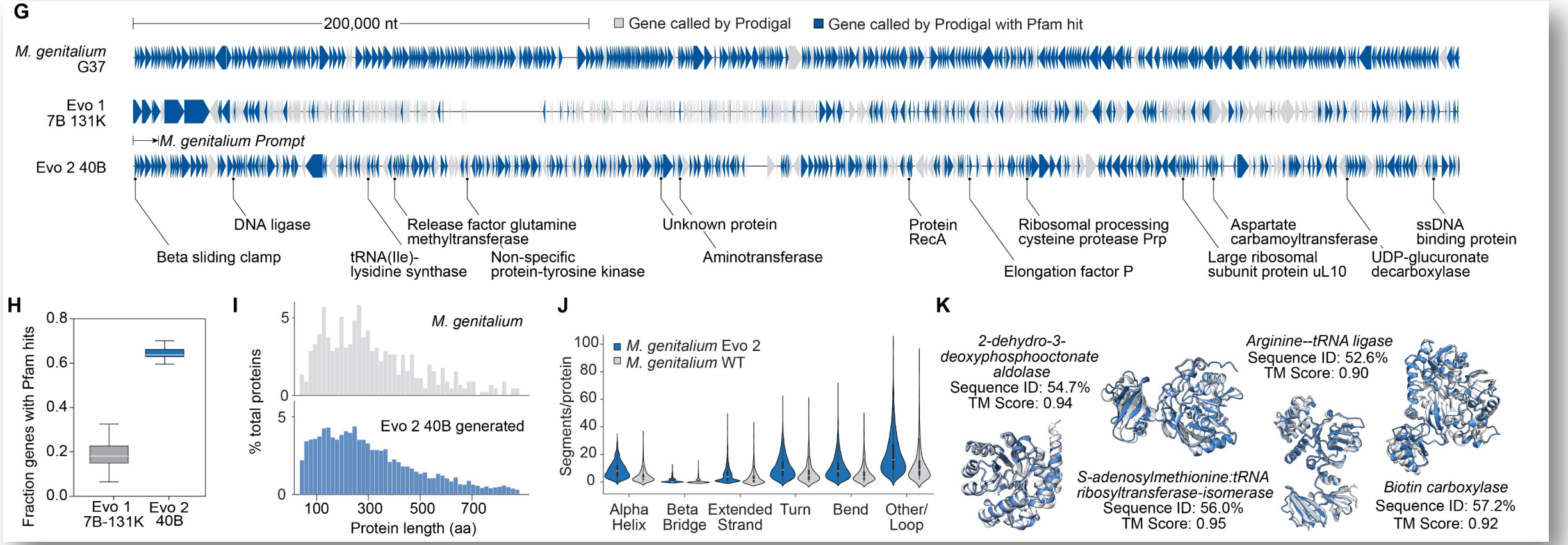
バリアント効果予測に特化したモデル（AlphaMissenseなど）に匹敵する性能。non-SNV（indel等）、Noncoding領域の予測性能はトップ。

Evo2：結果・潜在変数の解釈性



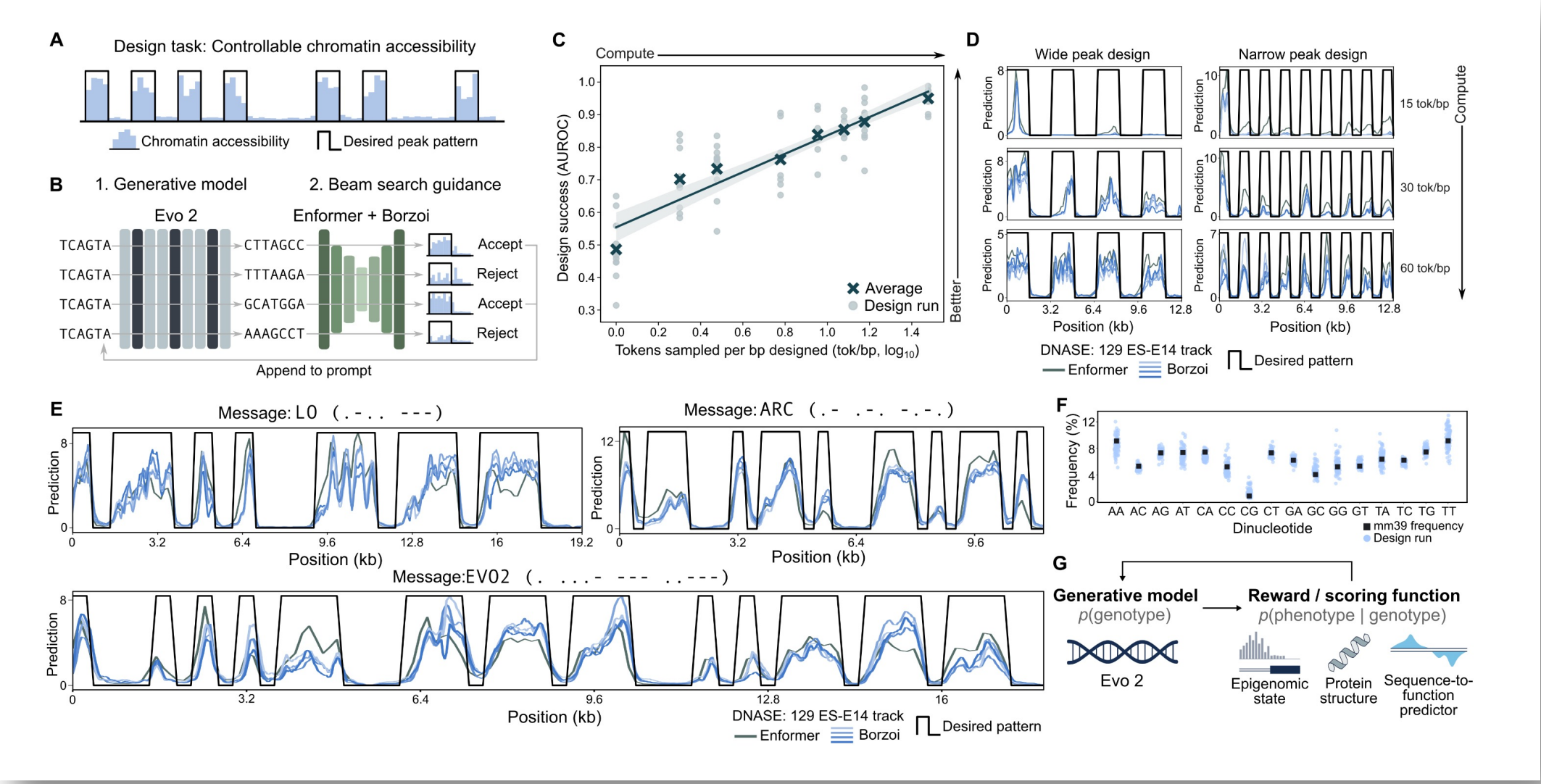
ゲノムを入力して埋め込まれた潜在変数を、スパースオートエンコーダで評価。
ゲノム上の領域ごとに、どのニューロンが発火するかを詳細に見ることができる。
多くのニューロンが既知のなんらかのゲノム上のパターンを表現している。
個別の特徴に特化して手作りしたHMMなどのモデルはもはや不要？

Evo2：結果・ゲノム生成



マイコプラズマゲノム、10kbを与えて、残りの500kbを生成。
Evo1と比較して、より「もっともらしい」ゲノムに。

Evo2：結果・補助条件によるゲノム生成の制御



Evo2によるゲノム生成と、EnformerによるAccesibilityパターン予測を組み合わせてぐるぐると回すことで、
望みのパターン制約を満たすゲノム配列を生成できる。
この例では（とくに意味はないが）Open/Close chromatinのパターンがモールス信号になる配列を生成。