JEPA-DNAゲノム基盤モデル技術分析報告書：ジョイント埋め込み予測アーキテクチャによるゲノム機能理解の革新

Gemini 3.6 Flash

NVIDIAがHugging FaceおよびGitHub上で公開した「JEPA-DNA」は、従来のゲノムと言語モデル（Genomic Language Models: gLMs）の限界を打破するために設計された、モデル非依存型（Model-agnostic）の継続事前学習フレームワークである[1]。本アーキテクチャは、従来の塩基配列レベルでの「生成型事前学習」と、自己監視表現学習の新パラダイムである「ジョイント埋め込み予測アーキテクチャ（Joint-Embedding Predictive Architecture: JEPA）」を高度に融合させている[3]。
DNAが持つ複雑な制御構造やエピゲノム的意味論（Semantics）を深く理解することを目的として開発された本モデルは、生物学ドメインにおける人工知能の応用範囲を劇的に拡張する可能性を秘めている[2]。本報告書では、Word文書への直接的な統合や実務的な技術移転を想定し、その理論的背景、数理モデル、具体的なシステム構築手順、および評価基盤について、学術的かつ実務的な観点から詳細に分析・解説する。

ゲノム言語モデルにおける「粒度の罠」とその技術的背景

ゲノム言語モデル（GFM）はこれまで、自然言語処理（NLP）における自己監視学習手法をほぼそのまま借用する形で発展してきた[2]。具体的には、ゲノム配列の一部をマスクして元の塩基を当てさせるマスク言語モデル（Masked Language Modeling: MLM）や、次の塩基を予測させる次トークン予測（Next-Token Prediction: NTP）が主流である[1]。これらの手法は、DNA配列の局所的な文法規則（コドンの境界やスプライス部位のモチーフなど）を学習するにはきわめて有効である[1]。
しかし、このような「トークン復元タスク」への過度な依存は、生物学的な意味理解において「粒度の罠（Granularity Trap）」と呼ばれる根本的な限界をもたらす[5]。ゲノムの真の機能（プロモーター、エンハンサー、クロマチン構造、遺伝子発現制御など）は、個々の塩基トークンの微視的な配置ではなく、数百から数万塩基対に及ぶマクロな配列コンテキストとその立体的な配置によって決定される[1]。
従来の生成パラダイムでは、モデルは局所的な「スペルチェック」に最適化されるあまり、配列全体が内包するグローバルな機能的意味論を学習し損ねる傾向があった[1]。この課題に対し、メタ（Meta）の元チーフAIサイエンティストであり、AMI Labsのヤン・ルカン（Yann LeCun）が提唱してきた「予測型世界モデル（Predictive World Models）」の概念が、生物学分野において具現化されたものがJEPA-DNAである[2]。
JEPA-DNAは、局所的な塩基のミクロな復元から、潜在空間（Latent Space）におけるグローバルな「セマンティック整列（Semantic Alignment）」へと学習シグナルを大転換させる[1]。これにより、ノイズに強く、下流タスクにおいて極めて高い線形分離可能性（Linear-separability）を誇るゲノム表現ベクトルが構築される[8]。

コアアーキテクチャと数理的定式化

JEPA-DNAの最大の特徴は、既存のゲノム言語モデルのバックボーンを変更することなく、その上に「潜在セマンティックグラウンディング（潜在的意味論の定着）」を後付け（継続事前学習）できる点にある[1]。

1. デュアルエンコーダ構造と情報の流れ

本フレームワークは、以下の要素から構成される非対称なジョイント埋め込み予測ネットワークである[8]。

- コンテキストエンコーダ（**Context Encoder**）: 処理対象となるDNA配列のうち、スパンベースで一部の領域がマスクされた不完全な配列を受け取り、潜在空間上へ写像する[8]。
- ターゲットエンコーダ（**Target Encoder**）: マスク処理が行われていない完全なDNA配列を受け取る[8]。学習の安定化と表現崩壊の防止を目的として、ターゲットエンコーダの重み θ_t はコンテキストエンコーダの重み θ_c の指数移動平均（Exponential Moving Average: EMA）として同期される[8]。
- 軽量予測器（**Predictor**）: 4層の軽量なTransformerアーキテクチャで構成され、コンテキストエンコーダの出力とマスク位置情報をもとに、ターゲットエンコーダが生成すべき「マスクされた領域の潜在表現」を予測する[8]。

2. 複合損失関数とVICRegによる崩壊防止

非対照学習（Non-contrastive Learning）アプローチにおける最大の技術的課題は、すべての入力に対してモデルが同一の定数ベクトルを出力してしまう「表現の崩壊（Representation Collapse）」である[9]。JEPA-DNAは、VICReg（Variance-Invariance-Covariance Regularization）フレームワークに準拠した正則化項を組み込むことで、ネガティブサンプルを必要とせずにこの問題を完全に回避している[9]。
最適化問題は、以下の複合損失関数の最小化として定義される[8]。

$$\min_{\theta,\phi} \mathcal{L}_{total} = \lambda_1 \mathcal{L}_{llm} + \lambda_2 \mathcal{L}_{jepa} + \lambda_3 \mathcal{L}_{var} + \lambda_4 \mathcal{L}_{cov}$$

[cite: 8]
各損失項の具体的な数理定義および機能的役割は以下の通りである[8]。

ネイティブ**LLM**損失（$\mathcal{L}_{llm}$）

バックボーンモデル本来の生成目的関数（MLMまたはNTP）であり、局所的な塩基の文法規則の維持に寄与する[8]。

$$\mathcal{L}_{llm} = -\sum_{i \in \text{masked}} \log P(x_i \mid x_{\setminus i})$$

JEPA潜在予測損失（$\mathcal{L}_{jepa}$）

コンテキストエンコーダ（予測器経由）の出力ベクトル z と、ターゲットエンコーダが出力する目的ベクトル z' のコサイン類似度損失である[8]。これにより、マスクされたセグメントの機能的意味論を直接予測する能力が鍛えられる[1]。

$$\mathcal{L}_{jepa} = 1 - \frac{z \cdot z'}{\|z\| \|z'\|}$$

[cite: 8, 9]

分散損失（$\mathcal{L}_{var}$）

埋め込み空間の各次元におけるバッチ内の多様性を担保し、全サンプルが同一の埋め込みに収束する（崩壊する）のを防止する[9]。

$$\mathcal{L}_{var} = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - \sigma(z_j))$$

[cite: 9] ここで、d は潜在表現の次元数、$\sigma(z_j)$ はバッチ内における第 j 次元の標準偏差、γ は目標閾値（通常 $\gamma = 1$）である[9]。

共分散損失（$\mathcal{L}_{cov}$）

次元間の相関を排除して、直交する独立した生物学的特徴を各次元が表現するように強制する[9]。

$$\mathcal{L}_{cov} = \frac{1}{d} \sum_{i \neq j} [C]_{i,j}^2$$

[cite: 9] ここで、$C \in \mathbb{R}^{d \times d}$ はバッチ内の中心化された埋め込みから算出される共分散行列であり、以下のように定義される[9]。

$$C = \frac{1}{B-1}(Z - \bar{Z})^T (Z - \bar{Z})$$

[cite: 9]（B はバッチサイズ、$Z \in \mathbb{R}^{B \times d}$ は埋め込み行列、$\bar{Z}$ はその平均ベクトルである[9]）。

リリースモデルファミリーと技術仕様

NVIDIAがHugging Face上でリリースしたJEPA-DNAモデル群は、既存の著名なGFMをバックボーンに採用し、ヒトゲノム基準配列「hg38」の約60万のシーケンスウィンドウを用いて継続事前学習が施されている[8]。
以下に、リリースされた3つのバリアントモデルの技術仕様を比較整理する[10]。

モデル識別子	バックボーンモデル	パラメータ数 / 特徴	入力仕様 / トークン化手法	主な適用領域
NV-JEPA-DNA-DNABERT2	DNABERT-2[2]	117M[2] / 高密度なAttention機構とMLMベースの学習履歴[2]。	BPEトークン化[2] / スパンベースマスキング[8]。	変異影響の高度な予測、発現調節因子の注釈[2]。
NV-JEPA-DNA-HyenaDNA	HyenaDNA 16k[8]	600K[8] / サブクアドラティック（準2次時間）動作、SSM / ロング畳み込み構造[8]。	単一塩基トークン化、最大8,192 bpの長文脈入力に対応[8]。	長距離のゲノム相互作用マッピング、配列摂動スコアリング[8]。
NV-JEPA-DNA-NTv3	Nucleotide Transformer v3[4]	多目的Transformer[4] / トークン予測と潜在セマンティクスの同時学習[2]。	マルチスケールK-merトークン化[4]。	進化保存領域の特定、複合制御部位の機能予測[4]。

実装・実行環境の構築プロトコル

JEPA-DNAをローカルシステムあるいは共有の高性能計算（HPC）クラスタ上に導入し、再現可能な事前学習およびベンチマークテストを実行するための手順を記述する[4]。Word文書での構成管理を考慮し、実行環境、依存パッケージ、および実行手順を構造化して提示する。

1. 動作環境と依存パッケージの設定

JEPA-DNAはPyTorch 2.6.0をベースに開発されており、NVIDIA GPU（Blackwell B300 SXM6やA10Gなど）による並列高速化をネイティブサポートしている[8]。
環境構築は、Anaconda（またはMiniconda）を用いたPython 3.10環境下でのセットアップ、もしくは標準の仮想環境（venv）を用いる手法が推奨される[4]。

```Bash
# オプションA: conda環境の作成とアクティベート
conda create -n jepa_dna_env python=3.10 -y
conda activate jepa_dna_env

# オプションB: pip (venv) 環境の作成とアクティベート
python -m venv jepa_dna_env
source jepa_dna_env/bin/activate

# 依存パッケージの一括インストール
pip install -r requirements.txt

# (GPU使用時の確認コマンド)
python -c "import torch; print(torch.cuda.is_available())"

# DNABERT-2でFlash Attentionを使用する場合の追加パッケージ (ビルド分離を無効化)
pip install flash-attn --no-build-isolation
```

2. 環境変数およびキャッシュパスの定義

Hugging Faceを介した大規模データのロードやチェックポイントの保存を行うにあたり、適切なストレージ領域へのキャッシュ設定が不可欠である[4]。

```Bash
export HF_HOME="/path/to/hf_home"
export HF_DATASETS_CACHE="/path/to/hf_datasets_cache"
```

3. GFMBench-APIの統合手順

JEPA-DNAによる表現モデルの品質評価を行うためには、標準評価ミドルウェアであるGFMBench-APIリポジトリを同一のワークスペース内にクローンし、PYTHONPATHを通す必要がある[4]。

```Bash
# GFMBench-apiリポジトリの取得
git clone https://github.com/NVIDIA/GFMBench-api.git

# PYTHONPATHの同時設定 (リポジトリのルートパスを結合)
cd /path/to/JEPA-DNA
export PYTHONPATH="/path/to/JEPA-DNA:/path/to/GFMBench-api"

# 正しくインポートされるか検証
python -c "import jepa_dna; import gfmbench_api"
```

4. チェックポイントのダウンロード

Hugging Face上のNVIDIA JEPA-DNA公式コレクションから必要なモデル重みを引き出す[4]。
技術的警告（警告事項）: huggingface_hub ライブラリのバージョンを1.0以上にアップグレードすると、依存関係にあるピン留めされた transformers が破損するおそれがある[4]。必要に応じて、以下の通り 0.36.2 に固定されたCLIエクストラを明示的にインストールすること[4]。

```Bash
# huggingface_hub[cli] のバージョン固定インストール
pip install "huggingface_hub[cli]==0.36.2"

# 各チェックポイントのダウンロード実行
# 1. DNABERT-2 backbone
hf download nvidia/NV-JEPA-DNA-DNABERT2 \
  --local-dir paper_examples/paper_checkpoints/NV-JEPA-DNA-DNABERT2

# 2. NTv3 backbone
hf download nvidia/NV-JEPA-DNA-NTv3 \
  --local-dir paper_examples/paper_checkpoints/NV-JEPA-DNA-NTv3

# 3. HyenaDNA backbone
hf download nvidia/NV-JEPA-DNA-HyenaDNA \
```

```
--local-dir paper_examples/paper_checkpoints/NV-JEPA-DNA-HyenaDNA
```

5. 事前学習（継続学習）の実行プロトコル

既存のGFMモデルをベースにJEPA-DNAの枠組みで追加の事前学習を回す場合、各設定（バックボーン選択、損失関数の係数など）を記述した params.json を準備した上で、以下のプレトレイナースクリプトをキックする[4]。

```Bash
# 実行ディレクトリ構成の設定
# 事前に /path/to/training_dir 内に params.json を配置
# (例: paper_examples/reproduce_params_files/params_dnabert2.json からのコピー)

python ./jepa_dna/run_jepa_pretrain.py \
  /path/to/training_dir \
  --root_data_dir /path/to/pretrain_data
```

GFMBench-APIによる性能評価とベンチマーク結果

JEPA-DNAの有効性を検証するため、17の異なるゲノム評価タスクを包含する標準ベンチマークライブラリ「GFMBench-API」を用いた実証検証が行われた[8]。このベンチマークは、モデル自体のパラメータを固定して単純な線形分類レイヤーのみを訓練する「線形プロービング（Linear Probing: 9タスク）」と、追加訓練を一切伴わない「ゼロショット評価（Zero-shot: 8タスク）」の2つのアプローチで構成されている[8]。
以下に、主要な検証結果および性能向上幅を要約する[8]。

バックボーン	評価プロトコル	評価タスクの概要	主要指標と性能向上の実績
HyenaDNA	線形プロービング (9タスク) & ゼロショット (8タスク)[8]	プロモーター予測、スプライス部位検出、転写因子結合活性、非コーディング領域の変異影響スコアリングなど[8]。	平均AUROCが 約 **3.28%** 向上[8]。 特定の下流タスクでは 最大 **+11.2%** の劇的な性能改善を達成[8]。

DNABERT-2	線形プロービング[9]	制御領域のセマンティック機能の識別、構造バリアント分類[1]。	生成タスクの本来の精密さを犠牲にすることなく、線形分離性能が有意に向上[8]。

この結果は、JEPA-DNAが塩基配列の局所文法を維持しつつ、従来モデルが取りこぼしていた広域的・高次の制御情報を埋め込み空間内に結晶化できていることを証明している[8]。

ゲノムAI分野における戦略的意義と今後の展望

JEPA-DNAの発表とオープンソースでの公開は、生命科学研究におけるAIの役割を「シーケンスアライメントおよび分類の支援ツール」から「ゲノムの物理的・生物学的ダイナミクスを理解する予測世界モデル」へと昇華させるマイルストーンとなる[2]。

1. 変異影響予測（Variant Effect Prediction）への即応

個別化医療や希少疾患のゲノム解析において、非コーディング領域（ノンコーディングDNA）に存在する一塩基バリアント（SNV）やインデル（挿入・欠失）がどのような機能変化をもたらすかの予測は最重要課題の一つである[5]。JEPA-DNAは、ゼロショットで配列変化に伴う潜在的な空間摂動スコアを安定して出力できるため、実験的に検証が困難な何百万ものバリアントに対する信頼性の高いフィルターとして機能する[8]。

2. 進化生物学と頑健性の両立

ゲノムデータは本質的にノイズが多く、シークエンシングのエラーや自然な個体差を多く含む[5]。JEPA-DNAの潜在空間におけるベクトル予測は、このような配列レベルの瑣末な差分（高周波ノイズ）を自然に無視し、生物学的な頑健表現（低周波の機能的な骨格）のみを捉えるように設計されている[9]。これにより、異なる祖先背景や系統のゲノムを解析する際にも偏りの少ない中立な解釈が可能となる[15]。

3. 他の生体モダリティとの融合

本アーキテクチャの基本概念は、RNAスプライシング、シングルセル転写ミクス、プロテオミクス言語モデルなど、あらゆる構造化された生体モダリティへと水平展開可能である[16]。今後は、ゲノムという1次元の制御コードから3次元のクロマチン立体構造、そして最終的な表現型（疾患や細胞状態）までを潜在空間上で一貫してモデリングし、シミュレーションする「マルチオミクス世界モデル」の構築へと発展することが予見される。

引用文献

1. JEPA-DNA: Grounding Genomic Foundation Models through Joint-Embedding Predictive Architectures - Hugging Face, https://huggingface.co/papers/2602.17162
2. Nvidia's new DNA model learns what token prediction misses - The New Stack, https://thenewstack.io/nvidia-jepa-dna-genomics/
3. Start Customizing NVIDIA Nemotron 3 Nano with Prime… - Develeap, https://www.develeap.com/news/start-customizing-nvidia-nemotron-3-nano-with-prime-intellec-fd26de3b/

4. NVIDIA-BioNeMo/JEPA-DNA - GitHub, https://github.com/NVIDIA-BioNeMo/JEPA-DNA
5. Genomic language models: opportunities and challenges - ResearchGate, https://www.researchgate.net/publication/387723991_Genomic_language_models_opportunities_and_challenges
6. [2602.17162] JEPA-DNA: Grounding Genomic Foundation Models through Joint-Embedding Predictive Architectures - arXiv, https://arxiv.org/abs/2602.17162
7. (PDF) BioToken and BioFM - Biologically-Informed Tokenization Enables Accurate and Efficient Genomic Foundation Models - ResearchGate, https://www.researchgate.net/publication/390401808_BioToken_and_BioFM_-_Biologically-Informed_Tokenization_Enables_Accurate_and_Efficient_Genomic_Foundation_Models
8. nvidia/NV-JEPA-DNA-HyenaDNA - Hugging Face, https://huggingface.co/nvidia/NV-JEPA-DNA-HyenaDNA
9. (PDF) JEPA-DNA: Grounding Genomic Foundation Models through Joint-Embedding Predictive Architectures - ResearchGate, https://www.researchgate.net/publication/400970473_JEPA-DNA_Grounding_Genomic_Foundation_Models_through_Joint-Embedding_Predictive_Architectures
10. nvidia - Hugging Face, https://huggingface.co/nvidia
11. nvidia - Hugging Face, https://huggingface.co/nvidia/collections
12. GFMBench-API: A Standardized Interface for Benchmarking Genomic Foundation Models - bioRxiv, https://www.biorxiv.org/content/10.64898/2026.02.19.706811v1.full.pdf
13. GFMBench-API offers a unified middleware that decouples genomic foundation model architectures from diverse tasks, standardizes data streams and metrics, & enables users to submit models through a single interface for reproducible evaluation and reporting · GitHub, https://github.com/NVIDIA/GFMBench-api
14. GFMBench-API: A Standardized Interface for Benchmarking Genomic Foundation Models, https://www.researchgate.net/publication/400979703_GFMBench-API_A_Standardized_Interface_for_Benchmarking_Genomic_Foundation_Models
15. Simon A. Lee - Google Scholar, https://scholar.google.com/citations?user=Hlj-rdQAAAAJ&hl=en
16. JEPA In Bioscience - a mmhamdy Collection - Hugging Face, https://huggingface.co/collections/mmhamdy/jepa-in-bioscience