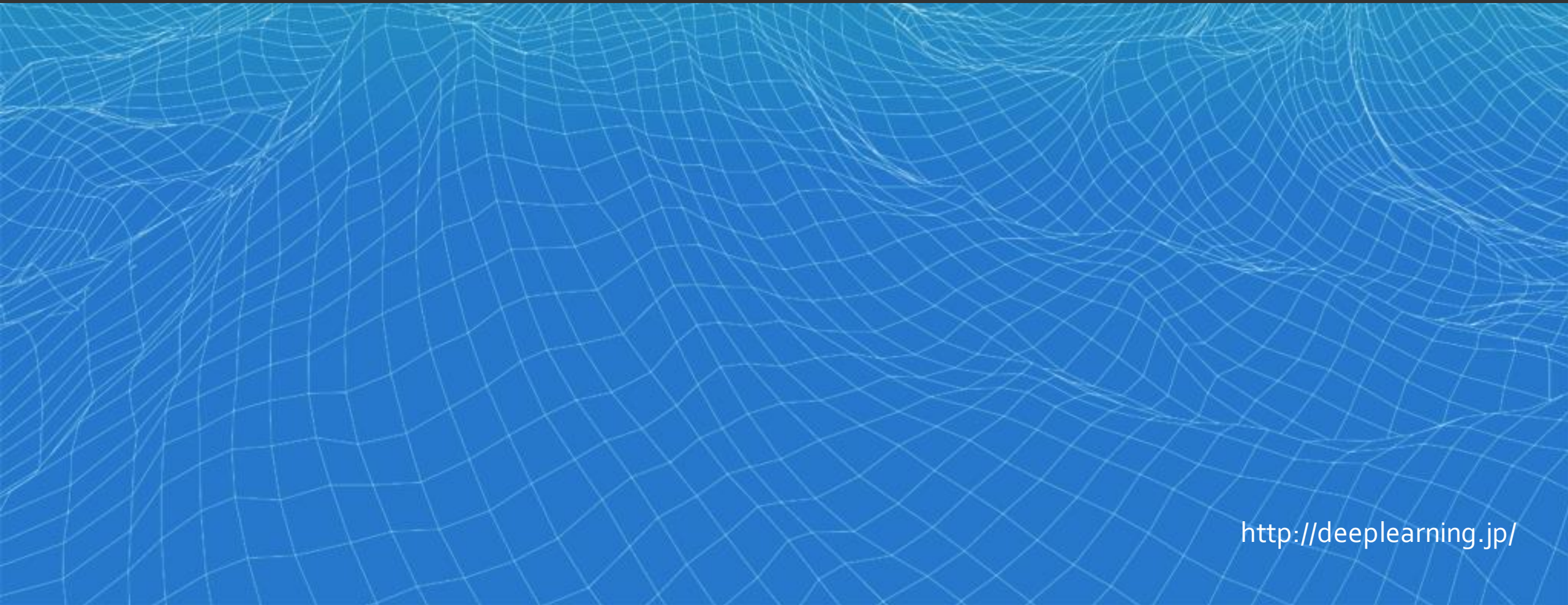
DEEP LEARNING JP
[DL Papers]

# GENERATING ALL-ATOM PROTEIN STRUCTURE FROM SEQUENCE-ONLY TRAINING DATA

Yuki Sato, Science Tokyo(Ohue lab)



http://deeplearning.jp/

# 書誌情報

## GENERATING ALL-ATOM PROTEIN STRUCTURE FROM SEQUENCE-ONLY TRAINING DATA

AmyX.Lu[1,2] Wilson Yan[1] Sarah A. Robinson[2] Kevin K. Yang[3] Vladimir Gligorijevic[2]
Kyunghyun Cho[2,4] Richard Bonneau[2] Pieter Abbeel[1] Nathan Frey[2]
[1]UC Berkeley [2]Prescient Design, Genentech [3]Microsoft Research [4]New York University

投稿先: bioRxiv(ICLR2025 under review)
投稿日: 2024/12/05
実装: https://github.com/amyxlu/plaid

選定理由

- 最近の研究テーマがタンパク質立体構造を扱うものであり最新の研究をサーベイしていたため
- タンパク質構造生成タスクでは一般に主鎖構造のみを生成する場合が多いが、この研究では側鎖も含めた全原子の生成を拡散モデルを用いて行っており興味があったため

※特に引用がない限り図表は本論文より引用

# 概要

- 既存のタンパク質立体構造生成モデルの多くははアミノ酸配列と立体構造を分離して扱っており、一貫性の欠如や生成構造の制御の難しさが問題であった。
- 本研究ではアミノ酸配列と立体構造を両方出力するマルチモーダルモデルPLAIDを提案した。PLAIDはアミノ酸配列と立体構造を同時に生成するため、このモデル1つのみで立体構造生成が可能である。

# 目次

- 研究目的
- 先行研究
- 提案手法
- 実験と結果
- 考察とまとめ
- 感想

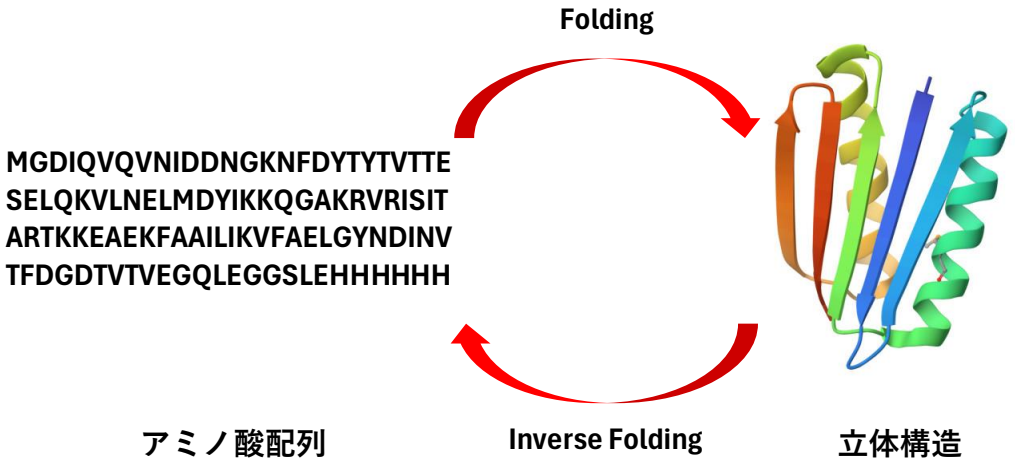
# タンパク質の立体構造

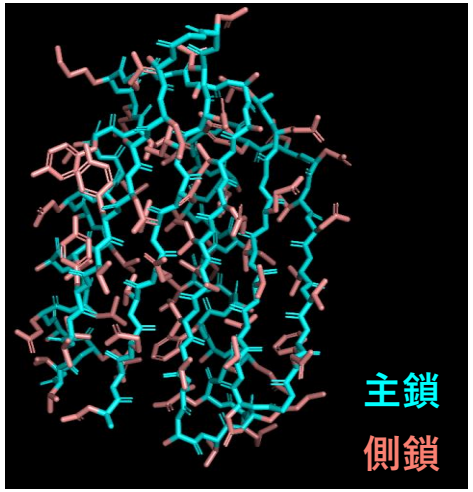
## タンパク質

- タンパク質は、一般に50以上のアミノ酸がペプチド結合してできる化合物のこと
- 共有結合や水素結合、静電気的相互作用によって複雑な立体構造を形成する
- アミノ酸で共通している-NH-CH-CO-の構造を主鎖、アミノ酸ごとに固有の構造を側鎖という。
- アミノ酸配列と立体構造には相関関係があることが知られており、アミノ酸配列から立体構造への変換をFolding、その逆はInverse Foldingと呼ばれる

## タンパク質の構造生成

- アミノ酸配列とタンパク質の立体構造を生成するタスク



# 研究目的

## 既存研究の問題点

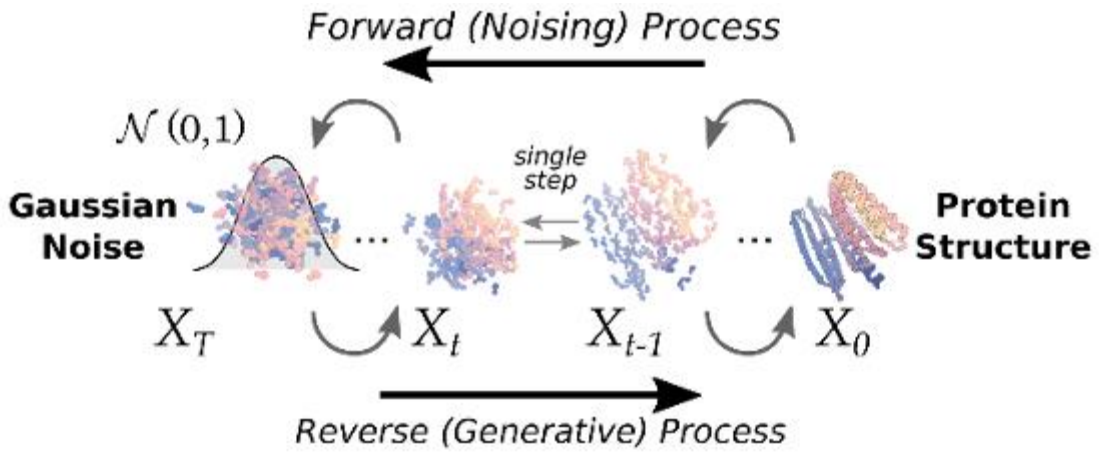
- データベースから生成を行う手法では生成される結晶化可能なタンパク質に偏りがある
- 全原子を扱う手法では、1.主鎖構造を生成→2.Inverse Foldingでアミノ酸配列を特定→3.構造予測モデルで全原子を予測、という流れが主流で配列と構造が別々に扱われていた
  - 最終的に出力される立体構造は構造予測モデルの出力であるため生成構造の制御が難しかった
- スケーラブルな学習を行うためのアーキテクチャに関する検討が不十分であった(?)

## 本研究の目的

- 1つのモデルでアミノ酸配列と対応する立体構造を同時に出力するマルチモーダルモデルの構築

# 先行研究: RFDiffusion[1]

- 拡散モデルを用いたタンパク質の主鎖構造生成モデル
- モデルは残基ごとの3次元座標を入力として、DDPMを応用した方法でノイズ除去を学習する手法
- サンプリング時はRFDiffusionで出力されたバックボーン構造をProteinMPNN[2]に入力してアミノ酸配列を出力、立体構造予測モデルに入力することでタンパク質の構造生成が可能




文献[1]より引用


1. Watson, Joseph L., et al. *Nature,* (2023).
2. Dauparas, Justas, et al. *Science*, (2022).

# 先行研究: 構造生成モデル

Protpardelle[3]

- タンパク質の全原子の3次元座標に対して拡散モデルを適用し学習
- 配列生成にはInverse Foldingモデルを使用

ProteinGenerator[4]

- アミノ酸配列に対して拡散モデルを適用し学習
- 立体構造はノイズ除去されたアミノ酸配列から予測し、損失関数にも利用

Multiflow[5]

- 離散フローベースモデルを用いてアミノ酸配列、アミノ酸ごとの並進・回転行列に対してノイズ除去を行い学習
- アミノ酸ごとの主鎖構造しか扱えず側鎖を生成することはできない


3. Chu, Alexander E., et al. PANS, (2024).
4. Lisanza, Sidney Lyayuga, et al. *bioRxiv,* (2023).
5. Campbell, Andrew, et al, *arXiv,* (2024).

# 提案手法: 全体像

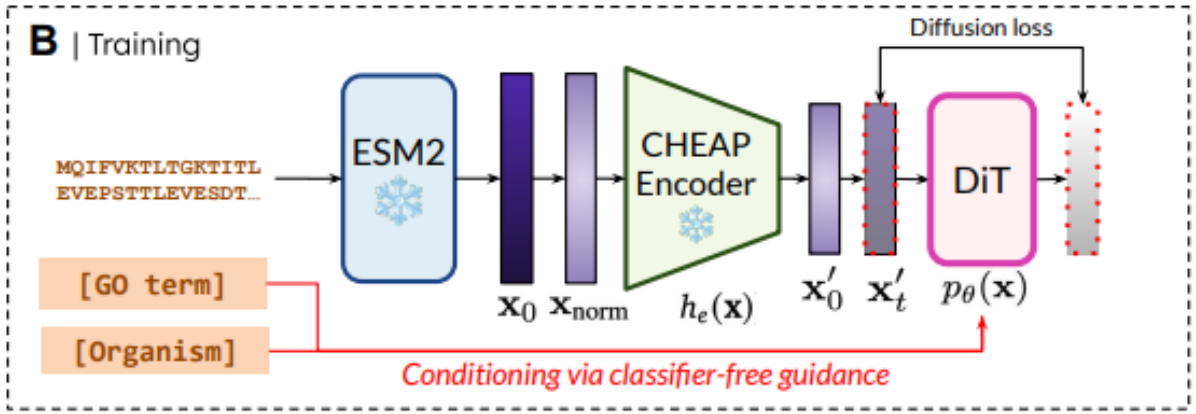
配列と構造を同一の潜在変数に埋め込み、配列と構造を生成するPLAID(Protein LAtent Induced Diffusion)を提案

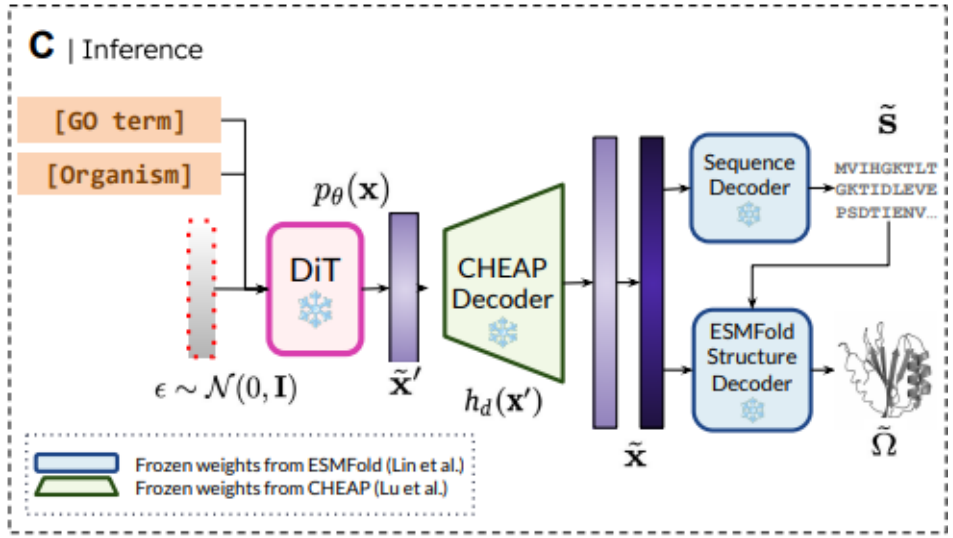
**学習時**

- アミノ酸配列からEncoderを用いて潜在変数に変換し、潜在変数に対して拡散モデルを適用して学習
- アミノ酸配列以外にも条件付けモジュールを追加して学習可能

**推論時**

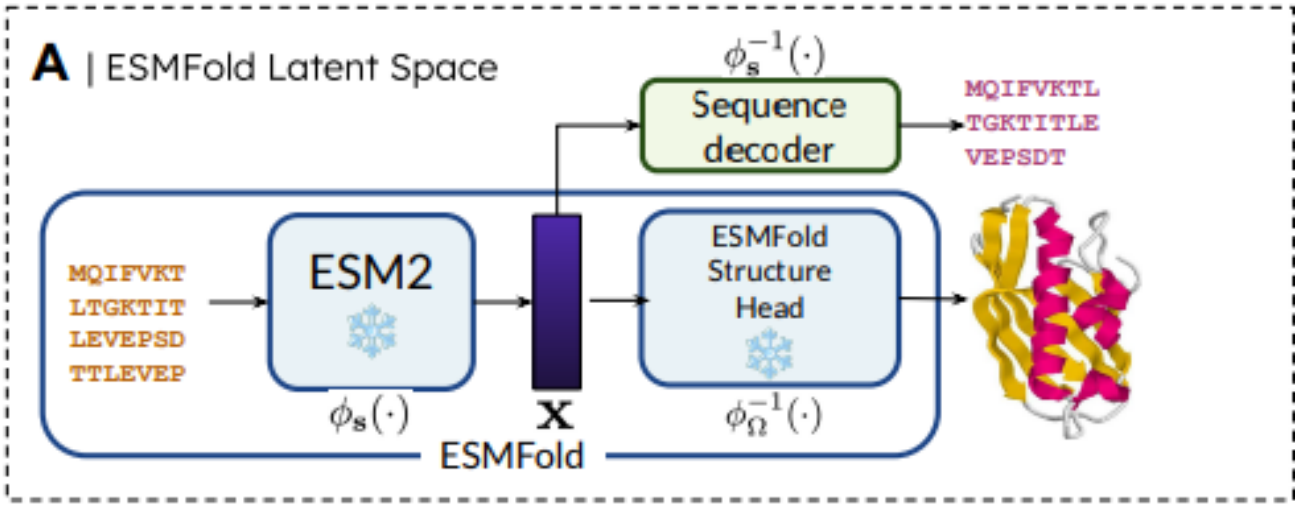
- ランダムなノイズを入力として拡散モデルから潜在変数を獲得し、配列・構造それぞれを生成するDecoderに入力して出力を得る。



# 提案手法: Encoder

- Encoderではアミノ酸配列を入力として潜在変数を出力する。
- Encoderは学習済みのESM2[6]とCHEAP Encoder[7]が用いられた。
- ESM2はESMFold[6]のEncoder部分で、masked language modelを大規模タンパク質配列データセットで学習したタンパク質言語モデルの1つである。
- CHEAP EncoderはESM2から得られる潜在変数を圧縮するために使用されるEncoder。CHAEPはESMFold内のESM2とStructure Decoderの間にEncoderとSequence Decoderを追加し構造と配列を予測するモデルを学習させることで、タンパク質の配列情報と構造データをEncoderの出力である低次元な潜在変数に埋め込むモデルである。

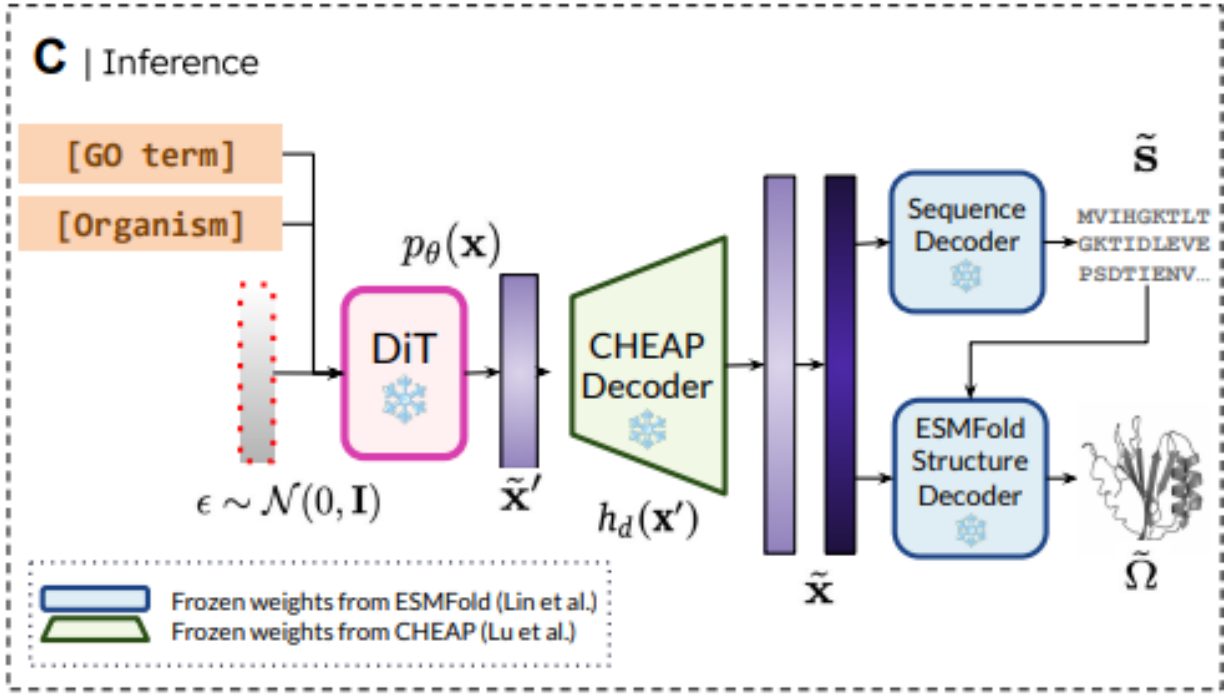




6. Zeming Lin *et al*. *Science*, (2023).
7. Lu, Amy X., et al. *bioRxiv,* (2024).

# 提案手法: Decoder

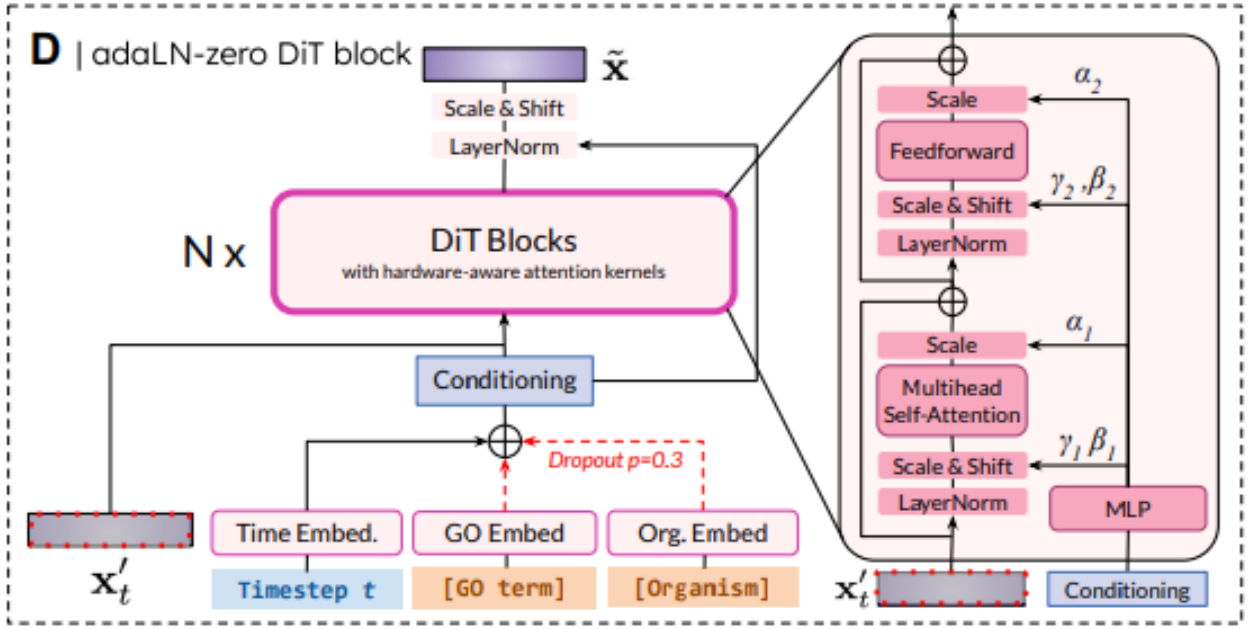
- Decoderでは拡散モデルの出力からアミノ酸配列と立体構造を出力する。
- アミノ酸配列の出力にはCHEAPで提案されているSequence Decoderを使用した。
- 立体構造の出力ではESMFoldで使用されているStructure Decoderを使用した。ESMFoldのStructure DecoderはAlphaFold2[8]のStructure Moduleが使用されている。




8. Jumper, John, et al. *nature*, (2021).

# 提案手法: 拡散モデル

- 拡散モデルにはDiffusion Transformer(DiT)[9]をベースとしたアーキテクチャを使用した。
- 提案手法では条件付けはGO言語(タンパク質の生物学的機能などを表す、全てのデータに振り分けられているわけではない)、Organism(生物種、ヒトなど)のみを想定しているため、DiTと異なりそれぞれEmbeddingし、classifier-free guidance[10]と同様に一定の確率でøとしてtime stepに足し合わせる。





9. Peebles, William, et al. *ICCV*, (2023).
10. Ho, Jonathan, et al. *arXiv,* (2022).

# 提案手法: 拡散モデルの学習

- 学習アーキテクチャにはDiTをベースとし以下の改良を加え、学習の安定性と精度向上を実現した。
  - v-diffusion[11,12] ノイズサンプリングの処理を見直し精度向上。
  - min-SNR rewerighting[13] 損失関数の重みをノイズに合わせて調整することで収束速度を向上。
  - sigmoid noise schedule[14]:ノイズスケジューリングの処理を改善し精度向上。
  - self-Conditioning[15,16] 学習時に一定の確率でモデルの出力で条件付けを行い学習することで精度向上。


11. Shanchuan Lin, et al. WACV, (2024).
12. Tim Salimans, et al. arXiv, (2022).
13. Tiankai Hang, et al. ICCV, (2023).
14. Ting Chen. arXiv, (2023).
15. Ting Chen, et al. arXiv, (2022).
16. Allan Jabri, et al. arXiv, (2022).

# データセット

Pfam[17]

- 2023年9月までにリリースされたデータ
- 20795ファミリー、57595205例の配列を含む
- データセットとして24637235例をサンプリングし、約15%を検証用に使用
- データセット内の約46.7%がGO用語を持ち、ユニークなラベルは2219例
- データセットに含まれるユニークな生物種は3617例

17. J. Mistry, et al. NAR, (2021).

# 実験設定

## 学習

- ESMFold,CHEAP autoencoderは学習済みモデルを使用
- タイムステップは1000
- 拡散モデルのパラメータ数は20億
- 学習ステップ数は80万

## 推論

- xformers[18]を使用して高速化・VRAM使用削減を行い実行
- DDIMサンプラーを用いて500タイムステップでサンプリング

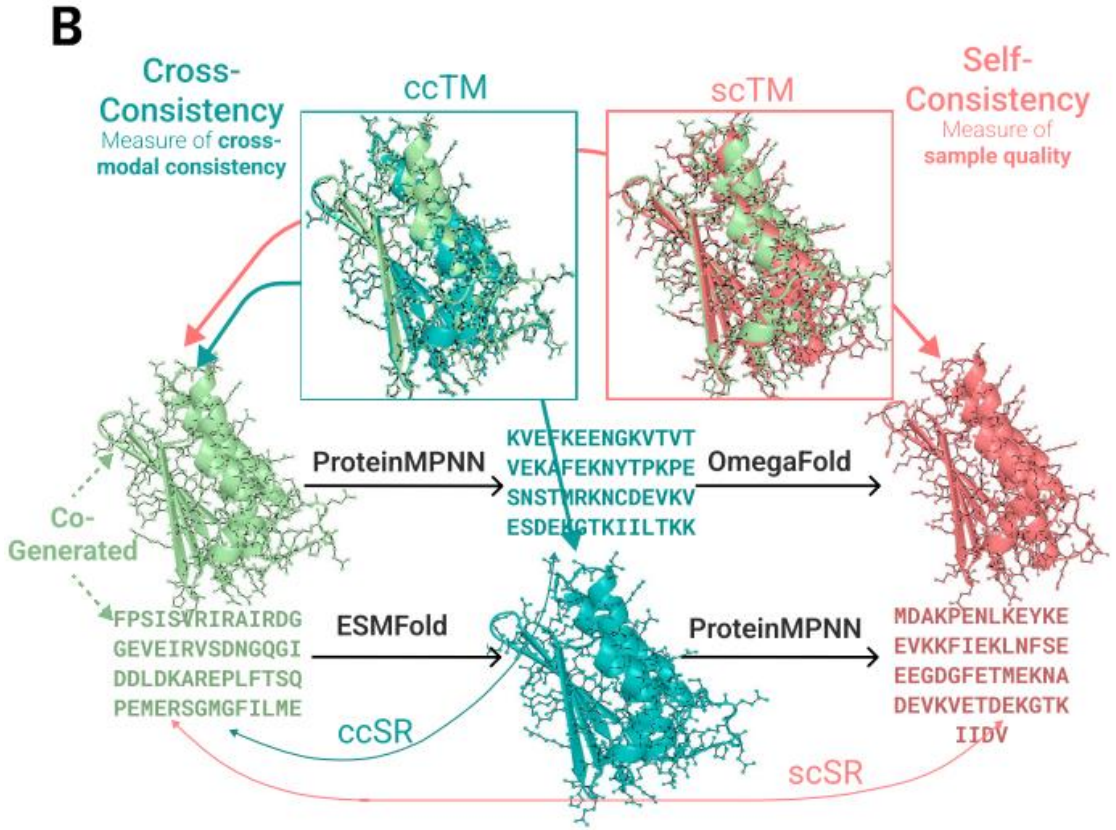
18. Benjamin Lefaudeux, et al. (2022).

# 評価指標: 条件なし生成

- **配列と構造の一貫性評価**:
  - 生成された配列に対応する立体構造と生成された立体構造が一致するかをTM-Score,RMSDで評価(ccTM,ccRMSD)
  - 生成された立体構造からProteinMPNNで予測した配列は生成された配列と一致するかをSequence recovery rateで評価(ccSR)
  - 生成された立体構造のccRMSD＜2Åを設計可能な構造と定義し、設計可能な構造の割合を評価
- **生成構造の評価**: 生成構造と生成構造からProteinMPNNで予測した配列をOmegaFoldで予測した構造が一致するかをTM-ScoreとRMSDで評価(scTM,scRMSD)
- **生成配列の評価**: 生成された配列から予測される構造をProteinMPNNで予測した配列が生成された配列と一致するかをSequence recovery rate(scSR)とタンパク質言語モデルを用いたトークンの発生確率(Ppl)で評価
- **構造の特性分布の適合率**: 生成構造と検証データの構造から生物物理学特性を計算しWasserstein距離を計算(Distributional conformity scores)
- **多様性**: 生成された配列・構造をそれぞれクラスタリングしクラスタ数を比較(# seq. clusters, # struct. clusters.)
- **新規性**: 生成構造と最も近い構造をPDB100から探索してTM-Scoreで評価(Foldseek TMScore)、生成配列と最も近いUniref90の配列との配列相同性を評価(MMseqs seq id)

# 評価指標: 条件なし・条件あり生成

- **GO用語による条件付けの評価:** GO用語でアノテーションして得られる潜在変数と検証データに含まれるタンパク質をEncoderに入力して得られる潜在変数のシンクホーン距離で評価
- **生物種の条件付けの評価:** 生物種で条件付けして得られる潜在変数をt-SNEを用いて2次元にプロットして分布を定性的に評価



# 実験

## 条件なし生成

- タンパク質長が$\{64,72,\ldots,512\}$の間でそれぞれ64例サンプリングし合計3648例のサンプルを出力、評価を行った

## 条件付き生成

- 条件付けして生成した配列・構造の定性評価を行った
- 検証データからランサムにサンプリングしたタンパク質とGO用語で条件付けした生成データとのシンクホーン距離での評価を行った

## 比較手法

- ProteinGenerator
- Protpardelle
- Multiflow

# 定量評価: 条件なし生成

## 配列と構造の一貫性評価

- ccSR以外の指標で最も良い精度を示した。

## 生成構造の評価

- ProteinGeneratorが最も高い精度を示したがデータセットに含まれるタンパク質も不完全である点に注意。

## 生成配列の評価

- データセットに含まれるタンパク質も含めて悪い精度を示した。

| | Cross-Modal Consistency | | | | Structure Quality | | | Sequence Quality | |
|---|---|---|---|---|---|---|---|---|---|
| | ccTM (↑) | ccRMSD (↓) | ccSR (↑) | % ccRMSD < 2Å (↑) | scTM (↑) | pLDDT (↑) | Beta sheet % (↑) | scSR (↑) | Ppl. (↓) |
| ProteinGenerator | 0.58 | 11.86 | **0.28** | 0.08 | **0.72** | **69.00** | 0.04 | 0.40 | **8.60** |
| Protpardelle | 0.44 | 24.28 | 0.22 | 0.00 | 0.57 | N/A | 0.11 | **0.44** | 8.86 |
| PLAID | **0.69** | **9.47** | 0.26 | **0.32** | 0.64 | 59.46 | **0.13** | 0.27 | 14.61 |
| *Natural* | *1.00* | *0.07* | *0.39* | *1.00* | *0.84* | 84.51 | 0.13 | 0.39 | 7.40 |

# 定量評価: 条件なし生成

構造の特性分布の適合率

- 全ての項目で最も良い精度を示した。

多様性

- 提案手法が最も良い精度を示した。

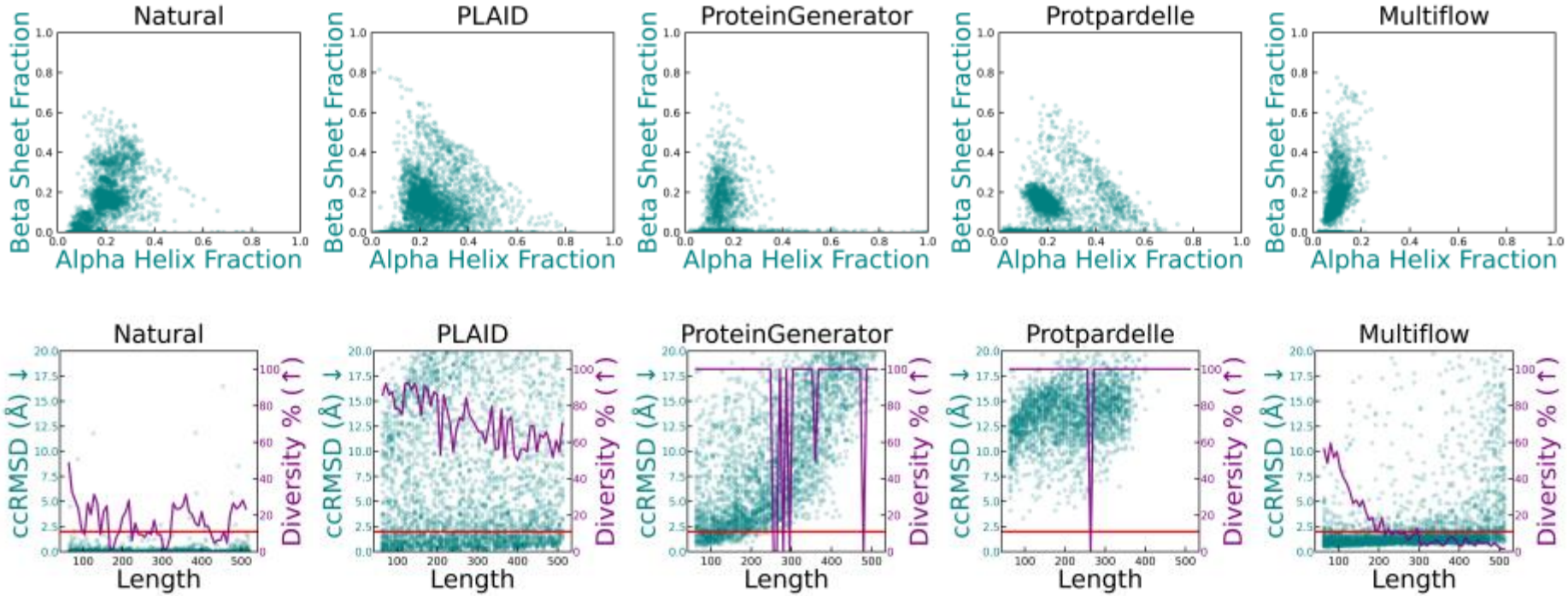
新規性

- 先行研究に劣る結果を示した。

| | Diversity | | | Novelty | | Distributional Conformity (Wasserstein Distance) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | # Des. (↑) | # Des. Seq. Clusts. (↑) | # Des. Struct. Clusts. (↑) | MMseqs Seq Id% (↓) | Foldseek TMScore (↓) | MW (↓) | Aroma-ticity (↓) | Dipeptide Instability Index (↓) | Iso-electricity (↓) | Hydro pathy (↓) | Charge at pH=7 (↓) |
| ProteinGenerator | 309 | 309 | 309 | 0.57 | **0.57** | 9.54 | 0.07 | 14.55 | 1.42 | 0.31 | 6.12 |
| Protpardelle | 0 | 0 | 0 | **0.56** | 0.72 | 10.4 | 0.07 | 8.61 | 1.99 | 0.37 | 8.58 |
| PLAID | **1171** | **809** | **522** | 0.60 | 0.67 | **0.62** | **0.01** | **1.98** | **0.49** | **0.28** | **2.71** |
| *Natural* | *3570* | *1362* | *600* | *0.81* | *0.87* | *0.0* | *0.0* | *0.0* | *0.0* | *0.0* | *0.0* |

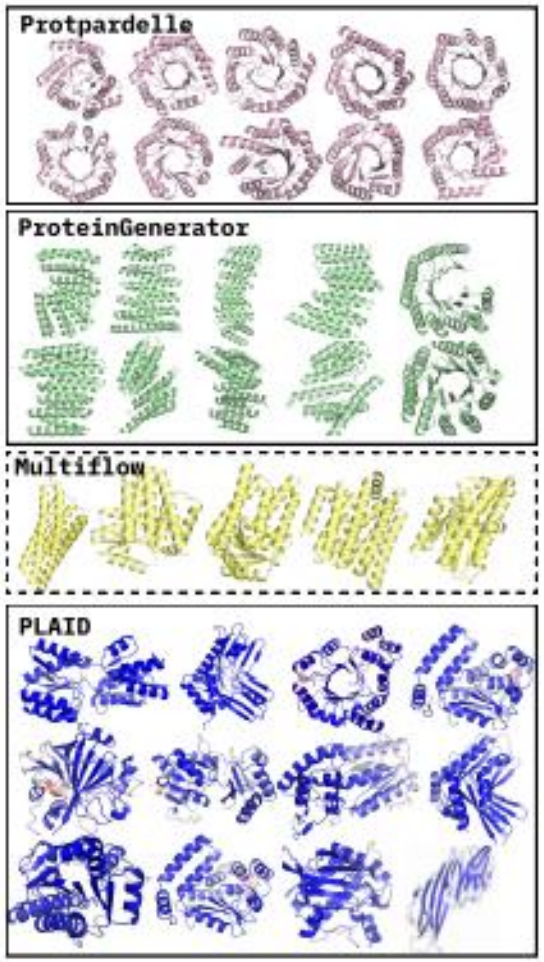
# 定性評価: 生成構造の解析

- 2次構造の分布をみるとPLAIDが最も自然のタンパク質の分布に近いことが分かった。
- 生成配列長と設計可能性のグラフを見るとPLAIDは配列長が長くなると一定精度は落ちるがその幅は小さかった。一方、Multiflowは大きく精度が低下し、他2つの手法では特異的な精度低下が確認された。



# 定量評価: 生成構造

- PLAIDでは多様な構造を生成できている一方、先行研究では部分構造を繰り返した構造が出力されていることが分かった。




**Protpardelle**
>len600_samp97
AGGGGGGGGGGGGGGGGGGGGGGGLGLGLLLPPAGL...
>len600_samp98
PPPPGGAGGGGAAAALAGGSPGGPPGGGGGGGGGGGG...
>len600_samp99
PPGPALPPSPGPGGVPPPPPLPPPPLPGGAPPAGGGLL...

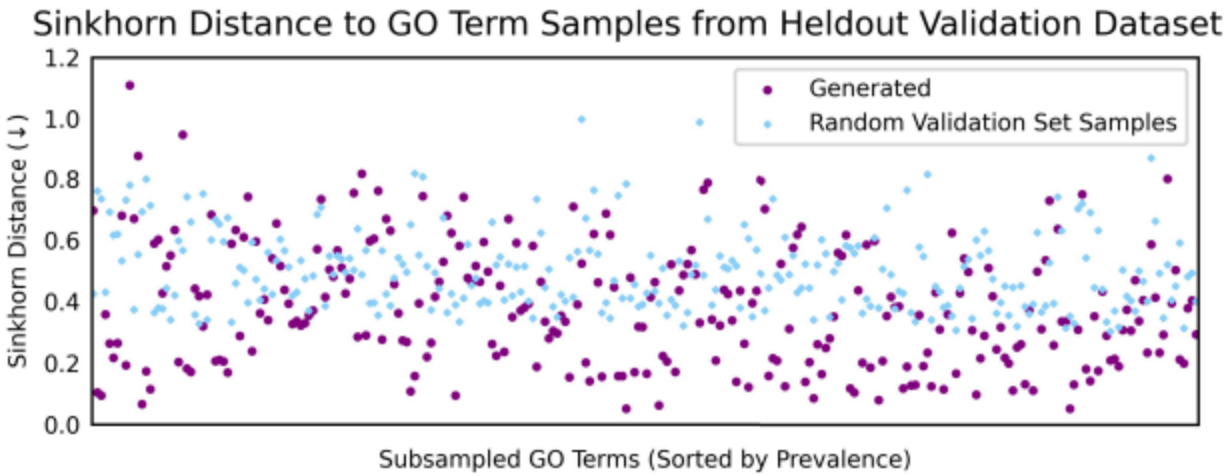
**ProteinGenerator**
>len600_000097
GAAGLTAAAAVVGAAAAAGAAAAAALAAAAGAGAAAAA...
>len600_000098
AGAAGAAAAAAAAAAAGAAAAGAGGGAGGAAAAAAAAAG...
>len600_000099
VAAAQAVQGAIAAAAALAATAALGLTAAGIAAPLLALV...

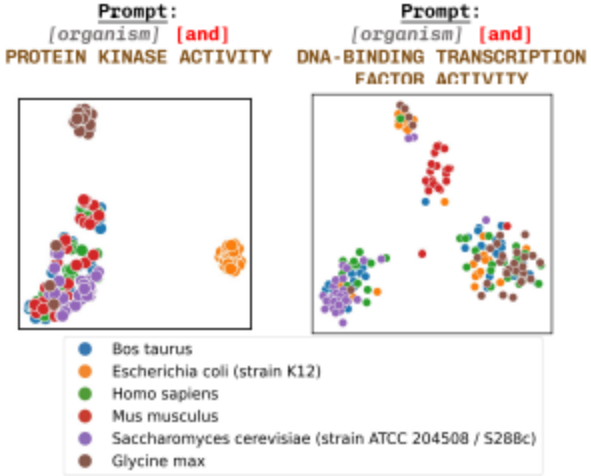
**Multiflow**
>len600_sample_97
LLGGLLGGLLGGAAGGAGAGAAAAGGGAVGVGVAGAVT...
>len600_sample_98
ADAATLTVGGGGTGGGGGAGGALGGAAAGGGGRVTLVV...
>len600_sample_99
AGGGAGLAGGAGGAGGAAAAAAAAAAAAAGAGGGAAAA...

**PLAID**
>len600_sample97
PDMGTVLGLAHSVGHLDFKTPDLSVADLETNLALLAAH...
>len600_sample98
FEMFDDKGGDLWERAASSGQLLIDVAYLANNGLRDGAT...
>len600_sample99
GNGGQARGTDDPLTHALQTLFQSAALDQSLQGDPENAV...

# 定量評価: 条件付き生成

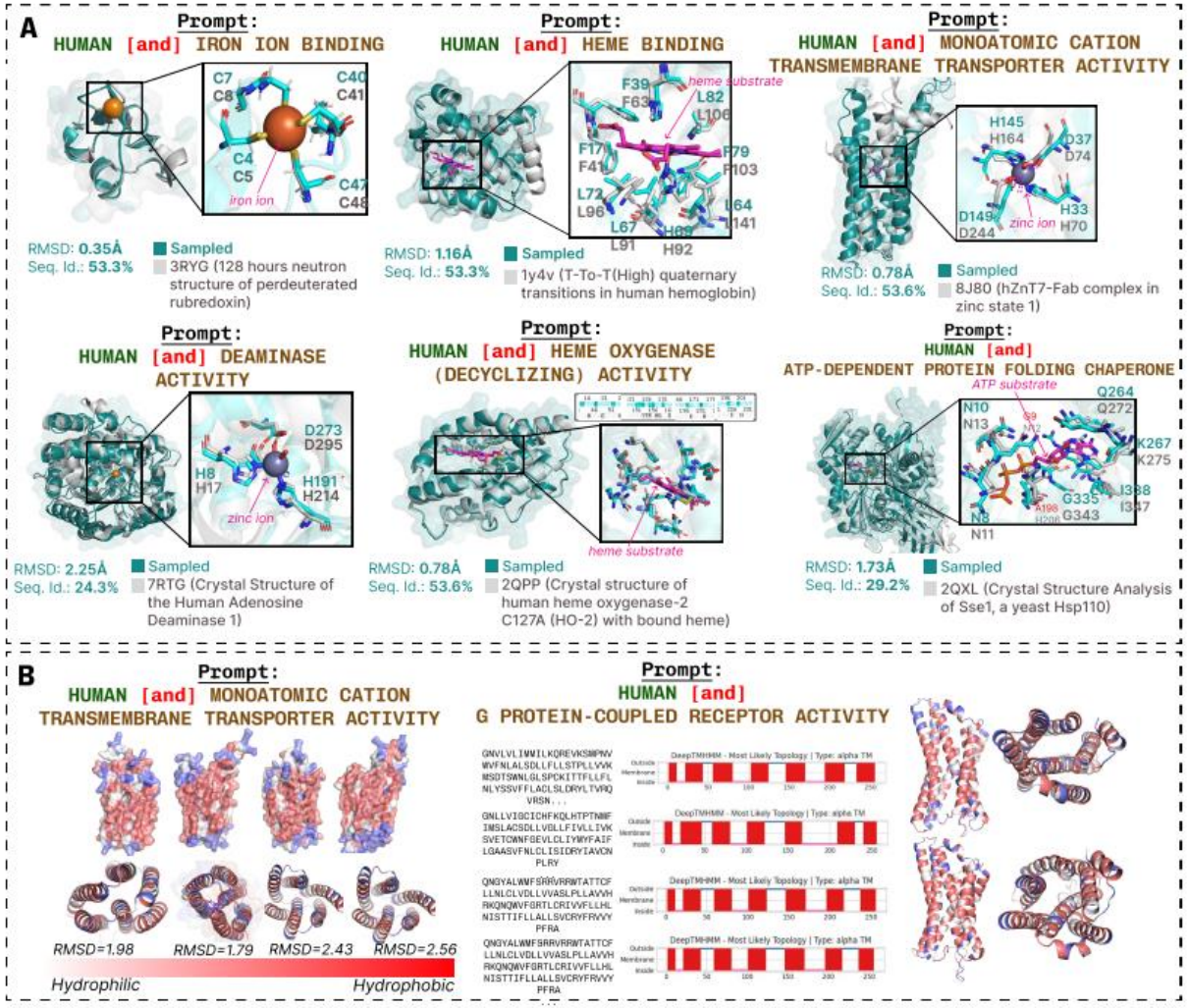
- GO用語を条件付けした場合のシンクホーン距離はランダムにサンプリングした場合と比較して小さいことが確認された。
- 生物種で条件付けして得られた潜在変数をt-SNEを用いて2次元にプロットしたところ、大腸菌やダイズはヒトやマウスなどの近縁の生物よりも明確なクラスタを形成していたことが確認された。



# 定量評価: 条件付き生成

- 配列相同性が低いにも関わらず条件付けされた機能を満たすような側鎖の配置が確認できた。
- また、膜貫通タンパク質を生成した例では膜貫通構造が生成されていることが確認された。



# 推論速度とAblation Study

- Nvidia A100を使用し、サンプリングステップ数100で配列長600のタンパク質を生成する時間とメモリ消費量を比較した。
- PLAIDは推論速度ではProtpardelleの次に速いことが確認された。
- 拡散モデルの学習時の工夫を取り入れたことによる精度向上が確認できた。

| | seconds/sample, batched | | seconds/sample, unbatched | |
|---|---|---|---|---|
| | Sample Latent | Decode | Sample Latent | Decode |
| Protpardelle | 11.21 | - | 17.16 | - |
| Multiflow* | 231.32 | - | 277.11 | - |
| ProteinGenerator* | 343.32 | - | 342.28 | - |
| **PLAID (100M)** | 1.64 | 15.12 | 27.63 | 1.07 |
| **PLAID (2B)** | 19.34 | 15.07 | 49.03 | 0.9 |

Table 1: Ablation results for metrics defined in Section 4.

| | Configuration | ccTM | scTM | Ppl. | Seq. Div. % | Struct. Div.% |
|---|---|---|---|---|---|---|
| **A** | cosine noise sched. & pred. noise | 0.54 | 0.55 | 16.97 | **0.98** | 0.86 |
| **B** | A + v-diffusion | 0.52 | 0.53 | 17.37 | **0.98** | **0.89** |
| **C** | A + MinSNR | 0.59 | 0.59 | 16.76 | 0.97 | 0.86 |
| **D** | A +B + C + sigmoid noise sched. | 0.56 | 0.58 | 16.88 | 0.92 | 0.86 |
| **E** | D + self-conditioning | **0.70** | **0.65** | **15.38** | 0.93 | 0.76 |
| **F** | E + no cond drop | 0.57 | 0.57 | 17.28 | 0.97 | 0.85 |

# 考察とまとめ

## 考察

- PLAIDが示した結果より条件付けした機能特性を持つ構造を保存しつつ相同性の低い配列と構造を出力できており、モデルが学習データをそのまま記憶することなくプロンプトに関連した生化学的特徴を学習できた可能性を示唆している。
- PLAIDは条件付けのアーキテクチャを変更することで複合体予測などへの応用が期待できる。
- PLAIDは学習済みDecoderを用いて構造生成を行うためDecoderの性能がボトルネックとなっており、これらも学習可能とすることで精度向上が期待される。

## まとめ

- 配列と側鎖も含めた立体構造を同時に出力するマルチモーダルモデルPLAIDを提案
- PLAIDは既存研究と比較して高速にかつ高い多様性の配列と構造を出力できる。
- PLAIDはGO用語と生物種について条件付けを行うことができ、条件に基づいた部分構造を出力できた。

# 感想

- 拡散モデルを用いて尤もらしい配列・構造が出るような潜在変数を生成できていることに驚いた。
  - 潜在変数に少しでもノイズが含まれているとDecoderが敏感に反応して出力構造がめちゃくちゃになりそうだと考えていた。
- 条件付けに関しても条件を出力構造に反映できているものが確認された点も興味深かった。
- 定量評価における評価指標に関してはより良い指標を設計すべきだと考える。
  - 評価の多くがInverse Foldingモデルや構造予測モデルの出力を使用しているがこれらモデルは予測モデルであり、本当に未知のタンパク質に対して適切に動作する保証がない。最も良い評価方法は生成結果を実際に生成できるか実験してみることだが一般の研究室では難しい。

# 参考文献


1. Watson, Joseph L., et al. "De novo design of protein structure and function with RFdiffusion." *Nature* 620.7976 (2023): 1089-1100.
2. Dauparas, Justas, et al. "Robust deep learning–based protein sequence design using ProteinMPNN." *Science* 378.6615 (2022): 49-56.
3. Chu, Alexander E., et al. "An all-atom protein generative model." *Proceedings of the National Academy of Sciences* 121.27 (2024): e2311500121.
4. Lisanza, Sidney Lyayuga, et al. "Joint generation of protein sequence and structure with RoseTTAFold sequence space diffusion." *bioRxiv* (2023): 2023-05.
5. Campbell, Andrew, et al. "Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design." *arXiv preprint arXiv:2402.04997* (2024).
6. Zeming Lin *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model.*Science*379,1123-1130(2023).DOI:10.1126/science.ade2574.
7. Lu, Amy X., et al. "Tokenized and Continuous Embedding Compressions of Protein Sequence and Structure." *bioRxiv* (2024): 2024-08.
8. Jumper, John, et al. "Highly accurate protein structure prediction with AlphaFold." *nature* 596.7873 (2021): 583-589.
9. Peebles, William, and Saining Xie. "Scalable diffusion models with transformers." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
10. Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." *arXiv preprint arXiv:2207.12598* (2022).
11. Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 5404–5411, 2024.
12. Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512, 2022.
13. Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via Min-SNR weighting strategy. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7441–7451, 2023.
14. Ting Chen. On the importance of noise scheduling for diffusion models. arXiv preprint arXiv:2301.10972, 2023.
15. Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. arXiv preprint arXiv:2208.04202, 2022.
16. Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. arXiv preprint arXiv:2212.11972, 2022.
17. J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G.A. Salazar, E.L.L. Sonnhammer, S.C.E. Tosatto, L. Paladin, S. Raj, L.J. Richardson, R.D. Finn, A. Bateman. "Pfam: The protein families database in 2021". Nucleic Acids Research (2021) doi: 10.1093/nar/gkaa913
18. Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xFormers: A modular and hackable Transformer modelling library. https://github.com/facebookresearch/xformers, 2022.