人工智能大模型与AlphaFold3应用

南京农业大学 人工智能学院

汇报人：朱一亨

2023年11月24日

目 录

01 生物大语言模型

02 AlphaFold3进展与应用

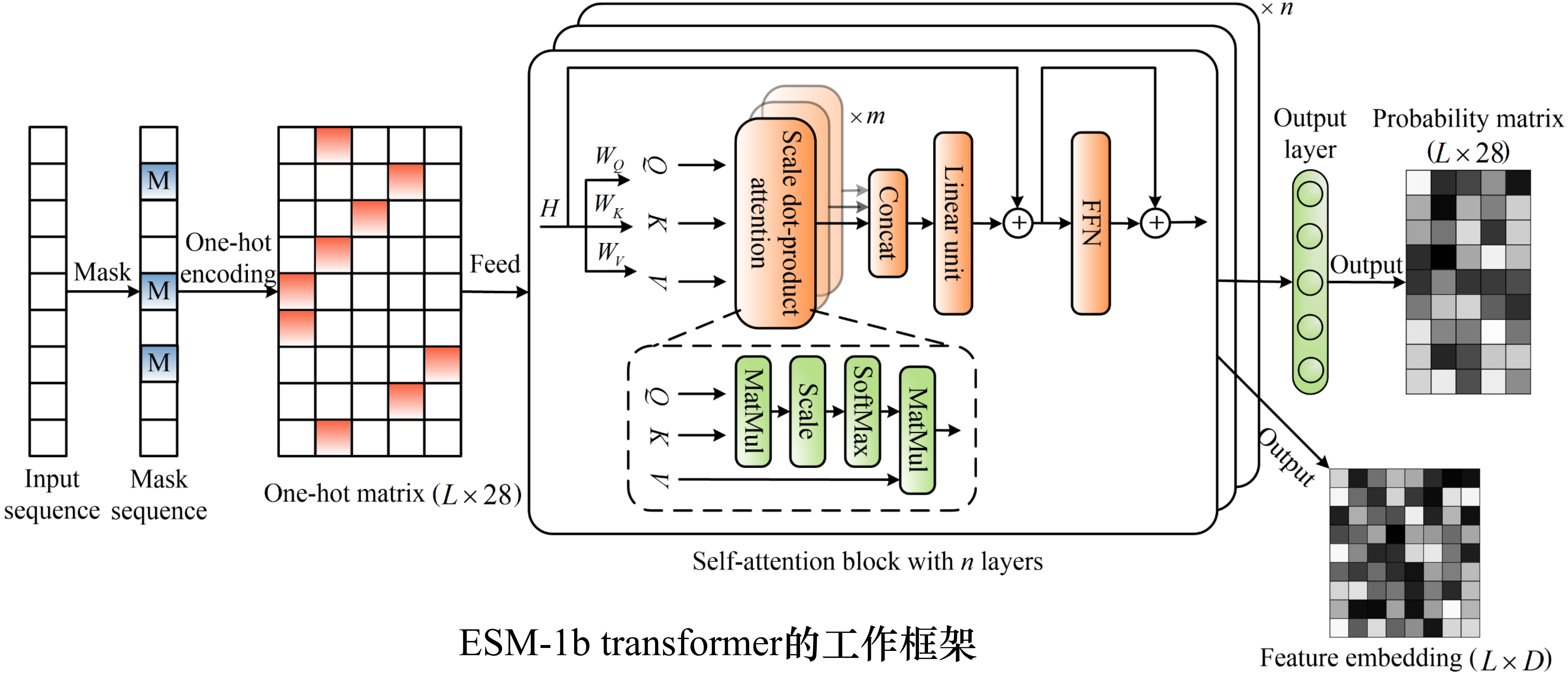
03 未来研究方向

01 Part one

生物大语言模型

01 蛋白质序列语言模型

■ ESM-1b transformer


Input
sequence
Mask
Mask
sequence
M
M
M
One-hot
encoding
One-hot matrix ($L \times 28$)
Feed
H
W_Q
W_K
W_V
Q
K
V
Scale dot-product attention
× m
Concat
Linear unit
FFN
× n
Q
K
V
MatMul
Scale
SoftMax
MatMul
Self-attention block with n layers
Output layer
Output
Probability matrix ($L \times 28$)
Output
Feature embedding ($L \times D$)

ESM-1b transformer的工作框架

Rives A et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences [J]. Proceedings of the National Academy of Sciences, 2021, 118(15): e2016239118. (谷歌学术引用量: 1188)

➢ 代码和模型下载地址

https://github.com/facebookresearch/esm

➢ 下游任务应用

（1） Protein structure prediction

ESMFold （Science, 2023）

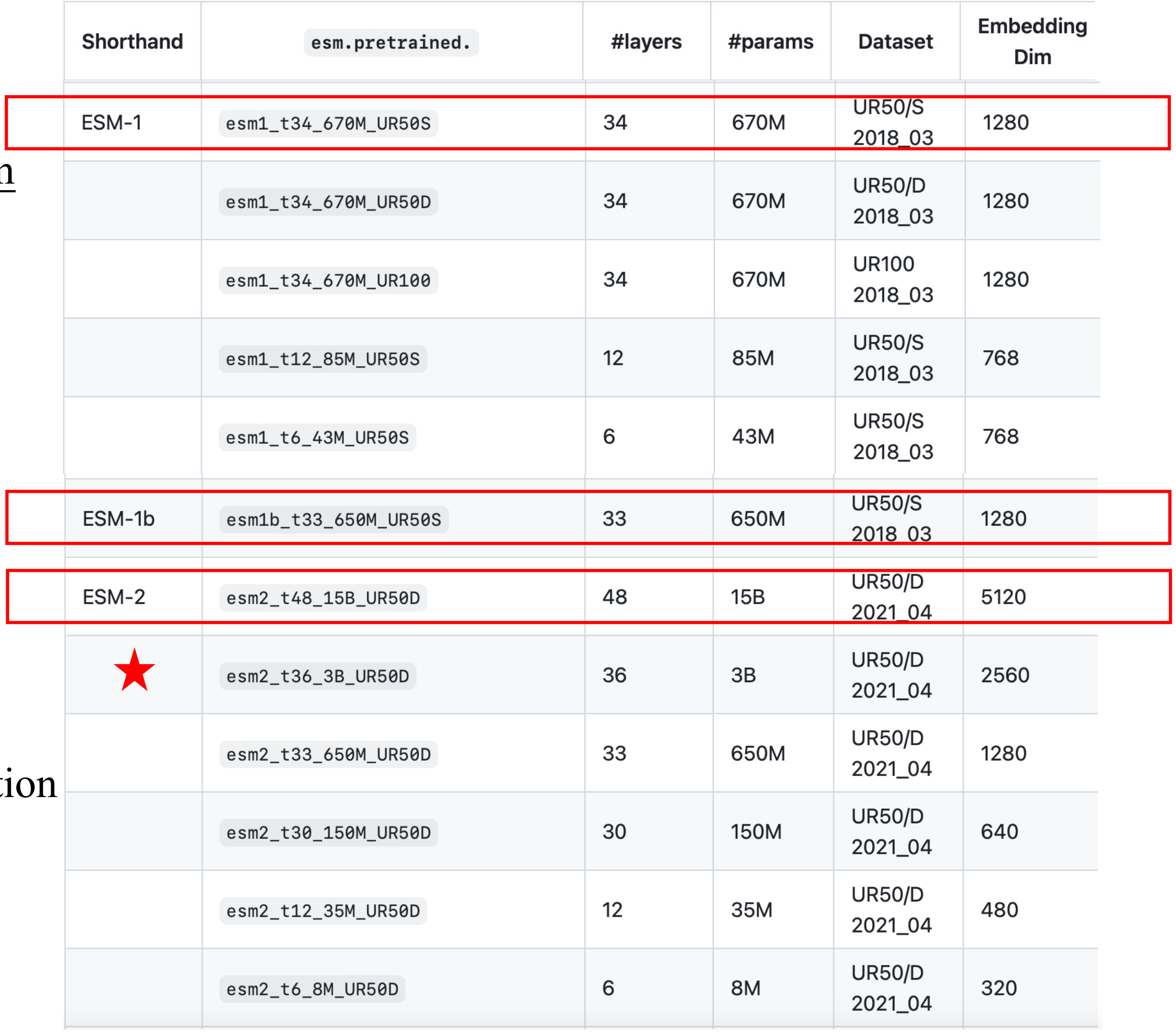
AlphaFold2 (Nature, 2022)

（2） Protein function prediction

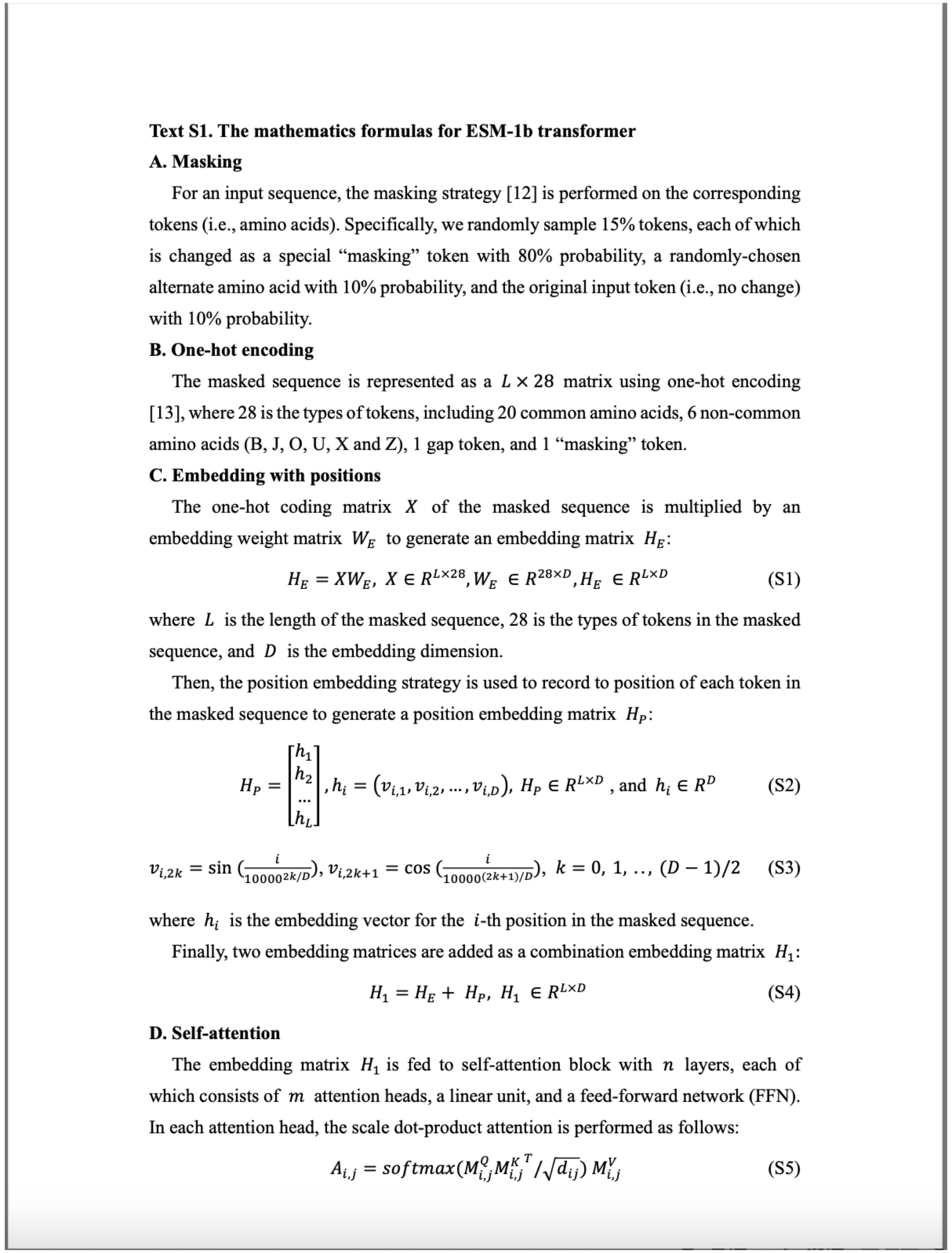
ATGO、NetGO 3.0、 HEAL

（3） Protein-ligand binding site prediction

ULDNA、 GraphSite、 NABind



Shorthand	esm.pretrained.	#layers	#params	Dataset	Embedding Dim
ESM-1	esm1_t34_670M_UR50S	34	670M	UR50/S 2018_03	1280
	esm1_t34_670M_UR50D	34	670M	UR50/D 2018_03	1280
	esm1_t34_670M_UR100	34	670M	UR100 2018_03	1280
	esm1_t12_85M_UR50S	12	85M	UR50/S 2018_03	768
	esm1_t6_43M_UR50S	6	43M	UR50/S 2018_03	768
ESM-1b	esm1b_t33_650M_UR50S	33	650M	UR50/S 2018_03	1280
ESM-2	esm2_t48_15B_UR50D	48	15B	UR50/D 2021_04	5120
★	esm2_t36_3B_UR50D	36	3B	UR50/D 2021_04	2560
	esm2_t33_650M_UR50D	33	650M	UR50/D 2021_04	1280
	esm2_t30_150M_UR50D	30	150M	UR50/D 2021_04	640
	esm2_t12_35M_UR50D	12	35M	UR50/D 2021_04	480
	esm2_t6_8M_UR50D	6	8M	UR50/D 2021_04	320



Text S1. The mathematics formulas for ESM-1b transformer

A. Masking

For an input sequence, the masking strategy [12] is performed on the corresponding tokens (i.e., amino acids). Specifically, we randomly sample 15% tokens, each of which is changed as a special “masking” token with 80% probability, a randomly-chosen alternate amino acid with 10% probability, and the original input token (i.e., no change) with 10% probability.

B. One-hot encoding

The masked sequence is represented as a $L \times 28$ matrix using one-hot encoding [13], where 28 is the types of tokens, including 20 common amino acids, 6 non-common amino acids (B, J, O, U, X and Z), 1 gap token, and 1 “masking” token.

C. Embedding with positions

The one-hot coding matrix X of the masked sequence is multiplied by an embedding weight matrix W_E to generate an embedding matrix H_E:

$$H_E = XW_E,\ X \in R^{L\times 28}, W_E\ \in R^{28\times D}, H_E\ \in R^{L\times D} \tag{S1}$$

where L is the length of the masked sequence, 28 is the types of tokens in the masked sequence, and D is the embedding dimension.

Then, the position embedding strategy is used to record to position of each token in the masked sequence to generate a position embedding matrix H_P:

$$H_P = \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_L \end{bmatrix}, h_i = \left(v_{i,1}, v_{i,2}, \dots, v_{i,D}\right),\ H_P \in R^{L\times D}\ \text{, and}\ h_i \in R^D \tag{S2}$$

$$v_{i,2k} = \sin\left(\frac{i}{10000^{2k/D}}\right), v_{i,2k+1} = \cos\left(\frac{i}{10000^{(2k+1)/D}}\right),\ k = 0,\ 1,\ ..,\ (D-1)/2 \tag{S3}$$

where h_i is the embedding vector for the i-th position in the masked sequence.

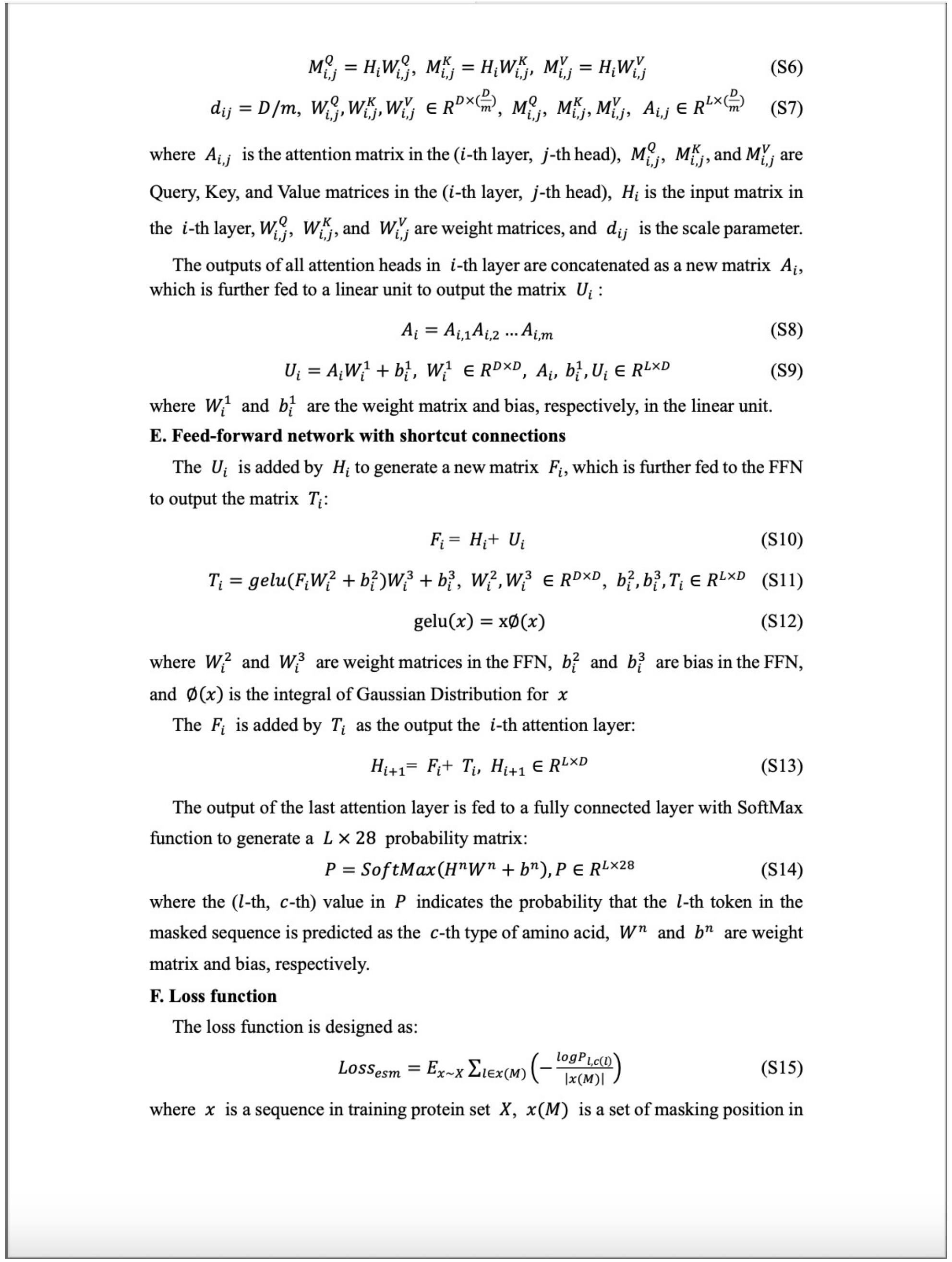
Finally, two embedding matrices are added as a combination embedding matrix H_1:

$$H_1 = H_E +\ H_P,\ H_1\ \in R^{L\times D} \tag{S4}$$

D. Self-attention

The embedding matrix H_1 is fed to self-attention block with n layers, each of which consists of m attention heads, a linear unit, and a feed-forward network (FFN). In each attention head, the scale dot-product attention is performed as follows:

$$A_{i,j} = softmax(M_{i,j}^{Q} {M_{i,j}^{K}}^{T} / \sqrt{d_{ij}})\ M_{i,j}^{V} \tag{S5}$$



$$M_{i,j}^{Q} = H_i W_{i,j}^{Q},\ M_{i,j}^{K} = H_i W_{i,j}^{K},\ M_{i,j}^{V} = H_i W_{i,j}^{V} \tag{S6}$$

$$d_{ij} = D/m,\ W_{i,j}^{Q}, W_{i,j}^{K}, W_{i,j}^{V}\ \in R^{D\times(\frac{D}{m})},\ M_{i,j}^{Q},\ M_{i,j}^{K}, M_{i,j}^{V},\ A_{i,j} \in R^{L\times(\frac{D}{m})} \tag{S7}$$

where $A_{i,j}$ is the attention matrix in the (i-th layer, j-th head), $M_{i,j}^{Q}$, $M_{i,j}^{K}$, and $M_{i,j}^{V}$ are Query, Key, and Value matrices in the (i-th layer, j-th head), H_i is the input matrix in the i-th layer, $W_{i,j}^{Q}$, $W_{i,j}^{K}$, and $W_{i,j}^{V}$ are weight matrices, and d_{ij} is the scale parameter.

The outputs of all attention heads in i-th layer are concatenated as a new matrix A_i, which is further fed to a linear unit to output the matrix U_i :

$$A_i = A_{i,1} A_{i,2} \dots A_{i,m} \tag{S8}$$

$$U_i = A_i W_i^1 + b_i^1,\ W_i^1\ \in R^{D\times D},\ A_i,\ b_i^1, U_i \in R^{L\times D} \tag{S9}$$

where W_i^1 and b_i^1 are the weight matrix and bias, respectively, in the linear unit.

E. Feed-forward network with shortcut connections

The U_i is added by H_i to generate a new matrix F_i, which is further fed to the FFN to output the matrix T_i:

$$F_i =\ H_i +\ U_i \tag{S10}$$

$$T_i = gelu(F_i W_i^2 + b_i^2) W_i^3 + b_i^3,\ W_i^2, W_i^3\ \in R^{D\times D},\ b_i^2, b_i^3, T_i \in R^{L\times D} \tag{S11}$$

$$\mathrm{gelu}(x) = \mathrm{x}\emptyset(x) \tag{S12}$$

where W_i^2 and W_i^3 are weight matrices in the FFN, b_i^2 and b_i^3 are bias in the FFN, and $\emptyset(x)$ is the integral of Gaussian Distribution for x

The F_i is added by T_i as the output the i-th attention layer:

$$H_{i+1} =\ F_i +\ T_i,\ H_{i+1} \in R^{L\times D} \tag{S13}$$

The output of the last attention layer is fed to a fully connected layer with SoftMax function to generate a $L \times 28$ probability matrix:

$$P = SoftMax(H^n W^n + b^n), P \in R^{L\times 28} \tag{S14}$$

where the (l-th, c-th) value in P indicates the probability that the l-th token in the masked sequence is predicted as the c-th type of amino acid, W^n and b^n are weight matrix and bias, respectively.

F. Loss function

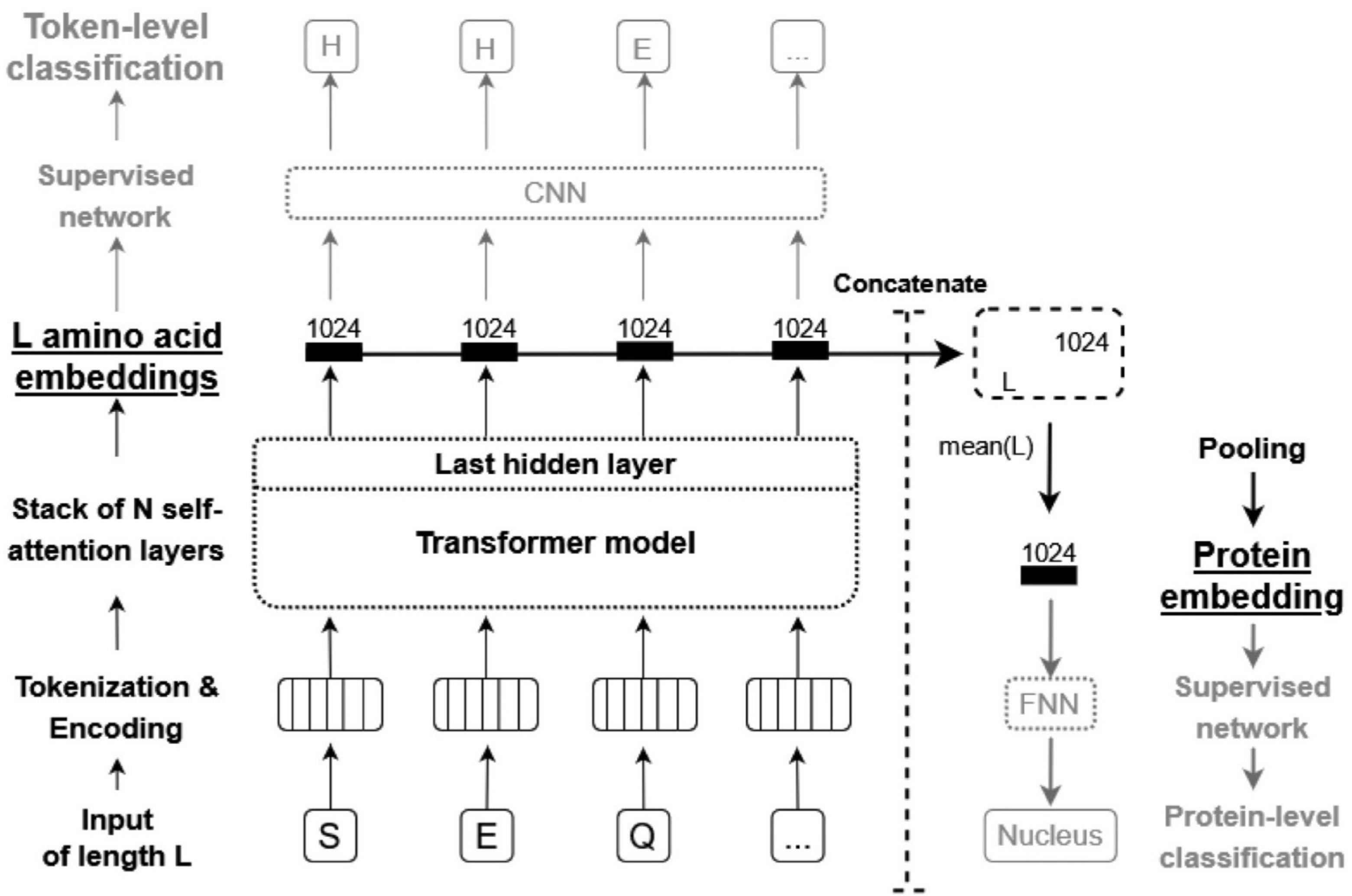
The loss function is designed as:

$$Loss_{esm} = E_{x\sim X} \textstyle\sum_{l\in x(M)} \left(-\frac{logP_{l,c(l)}}{|x(M)|}\right) \tag{S15}$$

where x is a sequence in training protein set X, $x(M)$ is a set of masking position in

https://yiheng-zhu.github.io/Yiheng/papers5.html

■ ProtTrans

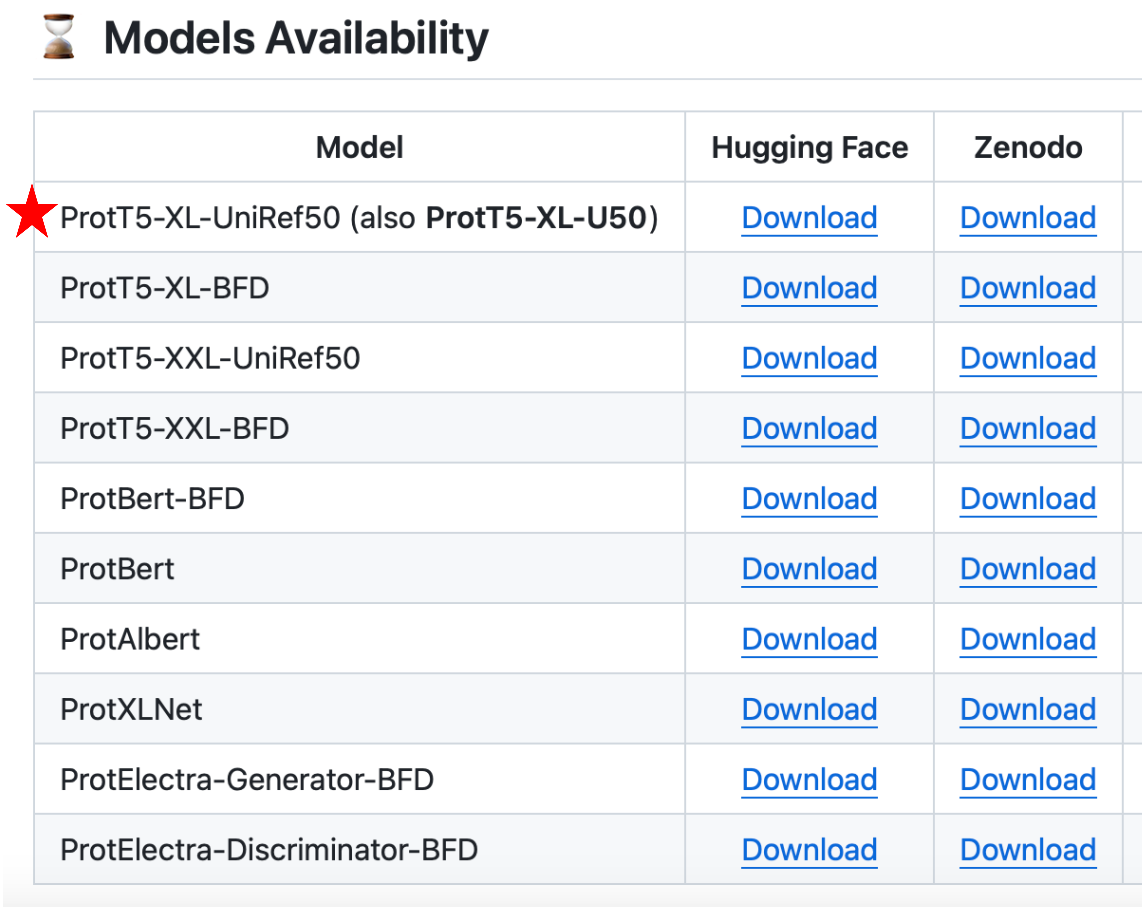


ProtTrans的工作框架

Elnaggar A et al. Prottrans: Toward understanding the language of life through self-supervised learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(10): 7112-7127. (谷歌学术引用量: 794)

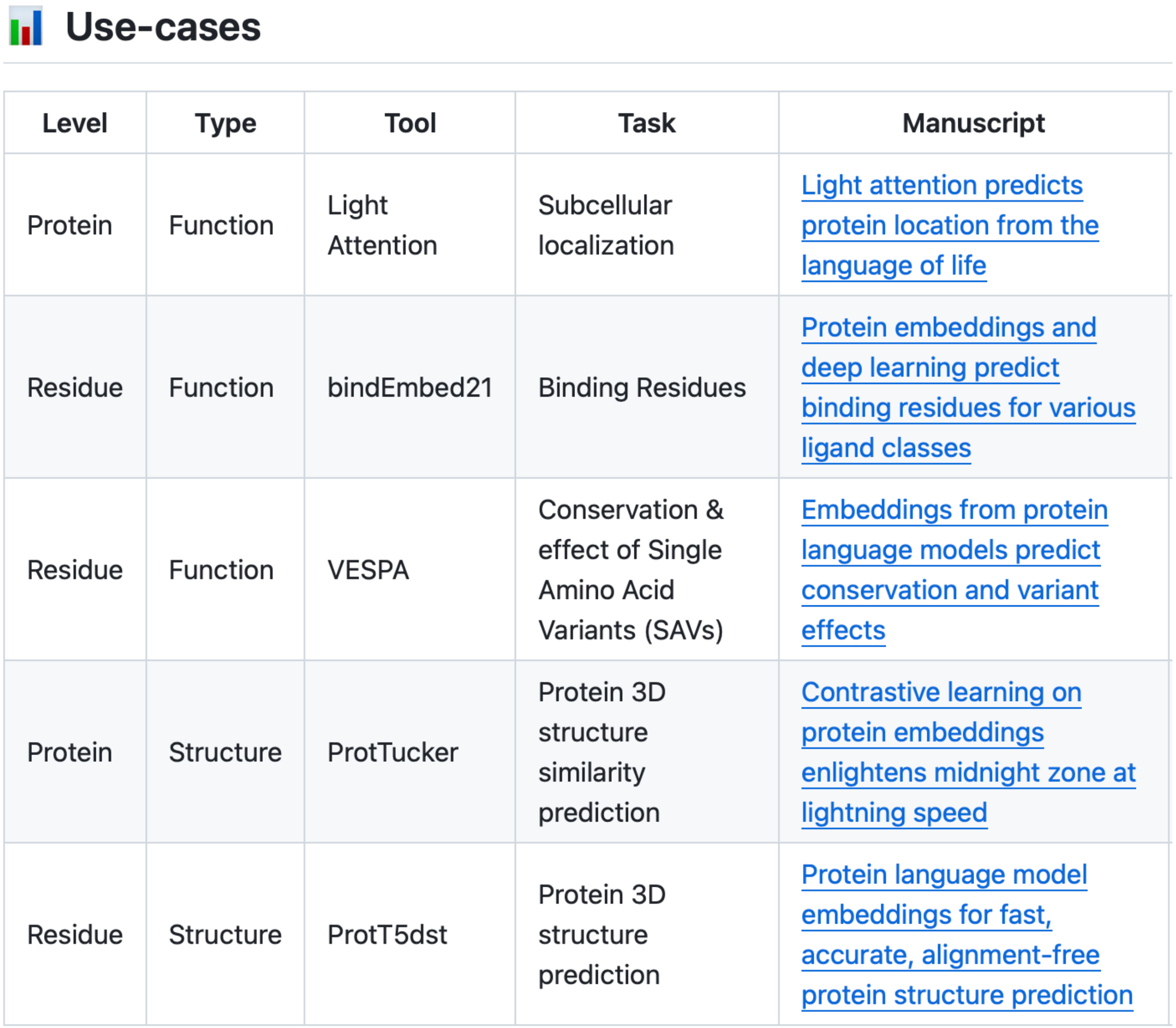
- 代码和模型下载地址

https://github.com/agemagician/ProtTrans



⌛ Models Availability

Model	Hugging Face	Zenodo
★ ProtT5-XL-UniRef50 (also **ProtT5-XL-U50**)	Download	Download
ProtT5-XL-BFD	Download	Download
ProtT5-XXL-UniRef50	Download	Download
ProtT5-XXL-BFD	Download	Download
ProtBert-BFD	Download	Download
ProtBert	Download	Download
ProtAlbert	Download	Download
ProtXLNet	Download	Download
ProtElectra-Generator-BFD	Download	Download
ProtElectra-Discriminator-BFD	Download	Download



📊 Use-cases

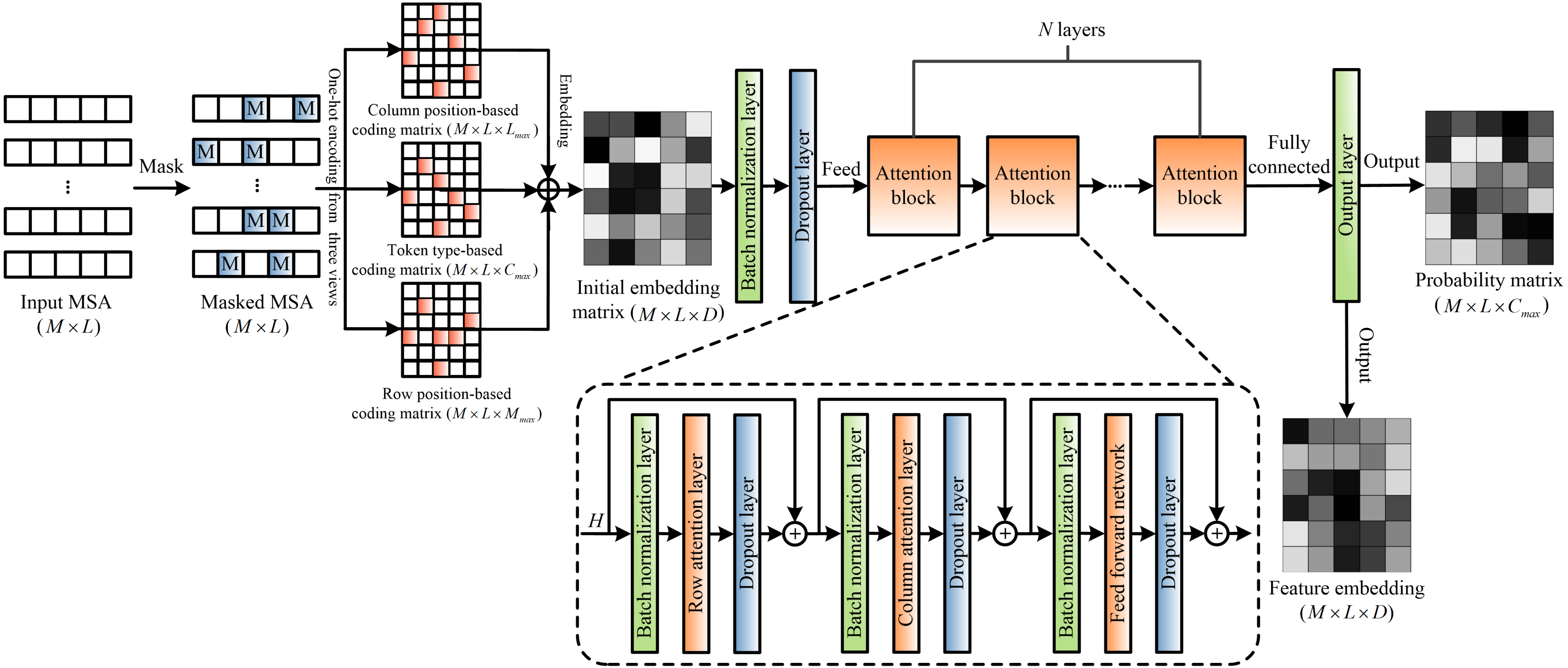
Level	Type	Tool	Task	Manuscript
Protein	Function	Light Attention	Subcellular localization	Light attention predicts protein location from the language of life
Residue	Function	bindEmbed21	Binding Residues	Protein embeddings and deep learning predict binding residues for various ligand classes
Residue	Function	VESPA	Conservation & effect of Single Amino Acid Variants (SAVs)	Embeddings from protein language models predict conservation and variant effects
Protein	Structure	ProtTucker	Protein 3D structure similarity prediction	Contrastive learning on protein embeddings enlightens midnight zone at lightning speed
Residue	Structure	ProtT5dst	Protein 3D structure prediction	Protein language model embeddings for fast, accurate, alignment-free protein structure prediction

- 其他蛋白质序列语言模型

(1) SeqVec (Heinzinger M et al, BMC Bioinformatics, 2019, 引用量: 387)
(2) TAPE (Rao R et al. Advances in Neural Information Processing Systems, 2019, 引用量: 633)
(3) Bepler & Berger’s approach (Tristan Bepler et al, ICLR, 2019,引用量: 278)

02 蛋白质多序列联配语言模型

■ ESM-MSA transformer

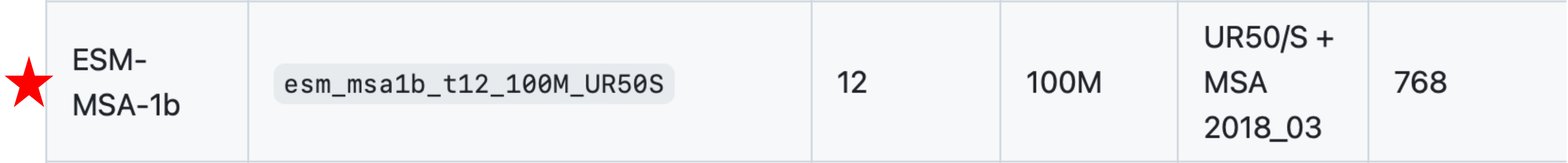

Input MSA
$(M \times L)$
Mask
M
Masked MSA
$(M \times L)$
One-hot encoding from three views
Column position-based coding matrix $(M \times L \times L_{max})$
Token type-based coding matrix $(M \times L \times C_{max})$
Row position-based coding matrix $(M \times L \times M_{max})$
Embedding
Initial embedding matrix $(M \times L \times D)$
Batch normalization layer
Dropout layer
Feed
N layers
Attention block
Attention block
Attention block
Fully connected
Output layer
Output
Probability matrix $(M \times L \times C_{max})$
Output
Feature embedding $(M \times L \times D)$
H
Batch normalization layer
Row attention layer
Dropout layer
Batch normalization layer
Column attention layer
Dropout layer
Batch normalization layer
Feed forward network
Dropout layer

ESM-MSA transformer 的工作框架

Rao R M et al. MSA transformer[C]//International Conference on Machine Learning. PMLR, 2021: 8844-8856. (谷歌学术引用量: 339)

➢ 代码和模型下载地址

https://github.com/facebookresearch/esm



★ ESM-MSA-1b	`esm_msa1b_t12_100M_UR50S`	12	100M	UR50/S + MSA 2018_03	768

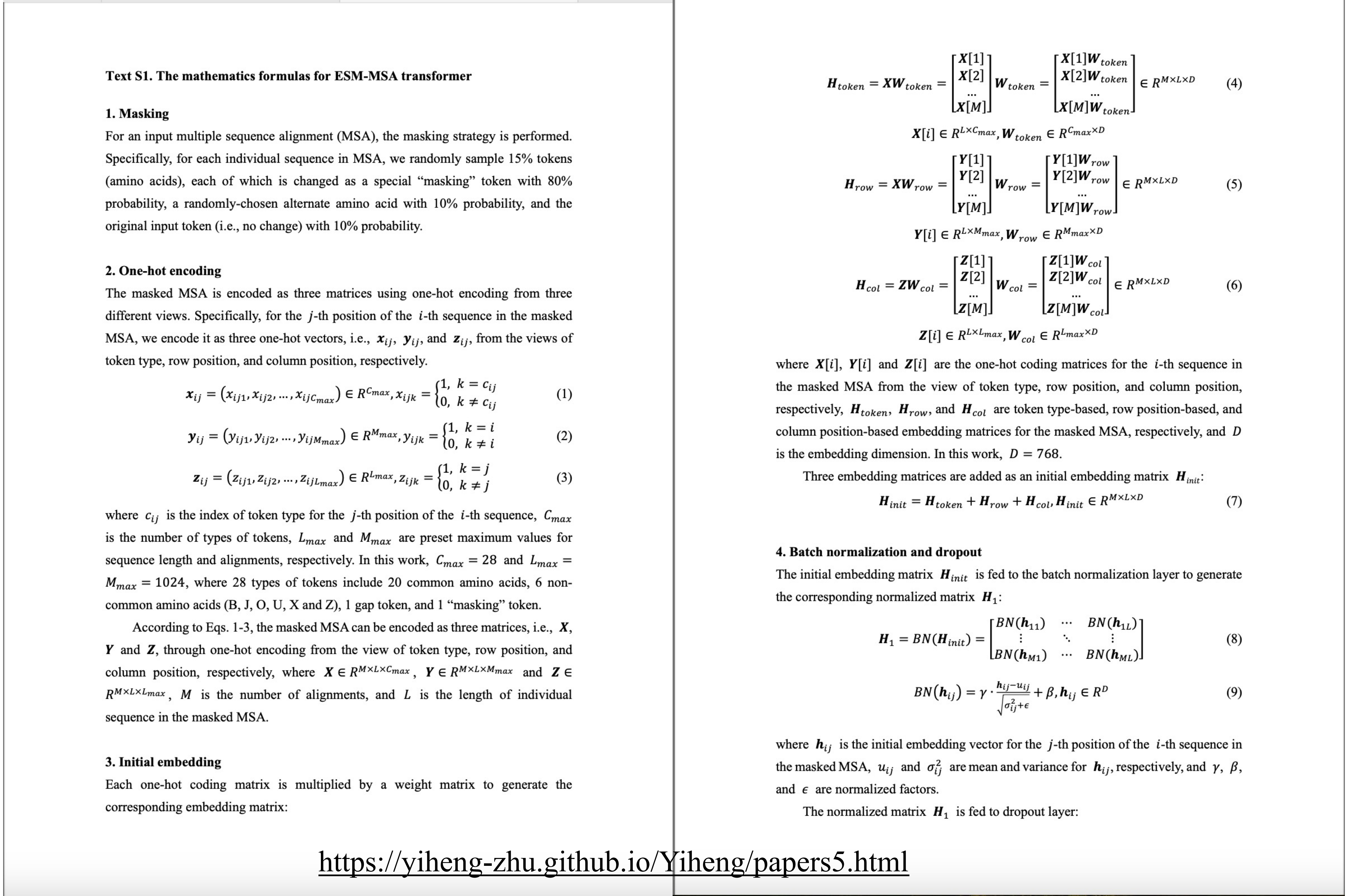
➢ 下游任务应用

(1) Protein contact prediction (single chain)

(2) Protein complex contact prediction (two chains)

DeepInter (Nature Machine Intelligence, 2023)

ICCPred (Computers in Biology and Medicine, 2023)



Text S1. The mathematics formulas for ESM-MSA transformer

1. Masking

For an input multiple sequence alignment (MSA), the masking strategy is performed. Specifically, for each individual sequence in MSA, we randomly sample 15% tokens (amino acids), each of which is changed as a special "masking" token with 80% probability, a randomly-chosen alternate amino acid with 10% probability, and the original input token (i.e., no change) with 10% probability.

2. One-hot encoding

The masked MSA is encoded as three matrices using one-hot encoding from three different views. Specifically, for the j-th position of the i-th sequence in the masked MSA, we encode it as three one-hot vectors, i.e., $\boldsymbol{x}_{ij}$, $\boldsymbol{y}_{ij}$, and $\boldsymbol{z}_{ij}$, from the views of token type, row position, and column position, respectively.

$$\boldsymbol{x}_{ij} = \left(x_{ij1}, x_{ij2}, \dots, x_{ijC_{max}}\right) \in R^{C_{max}}, x_{ijk} = \begin{cases} 1, & k = c_{ij} \\ 0, & k \neq c_{ij} \end{cases} \tag{1}$$

$$\boldsymbol{y}_{ij} = \left(y_{ij1}, y_{ij2}, \dots, y_{ijM_{max}}\right) \in R^{M_{max}}, y_{ijk} = \begin{cases} 1, & k = i \\ 0, & k \neq i \end{cases} \tag{2}$$

$$\boldsymbol{z}_{ij} = \left(z_{ij1}, z_{ij2}, \dots, z_{ijL_{max}}\right) \in R^{L_{max}}, z_{ijk} = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \tag{3}$$

where c_{ij} is the index of token type for the j-th position of the i-th sequence, C_{max} is the number of types of tokens, L_{max} and M_{max} are preset maximum values for sequence length and alignments, respectively. In this work, $C_{max} = 28$ and $L_{max} = M_{max} = 1024$, where 28 types of tokens include 20 common amino acids, 6 non-common amino acids (B, J, O, U, X and Z), 1 gap token, and 1 "masking" token.

According to Eqs. 1-3, the masked MSA can be encoded as three matrices, i.e., $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$, through one-hot encoding from the view of token type, row position, and column position, respectively, where $\boldsymbol{X} \in R^{M \times L \times C_{max}}$, $\boldsymbol{Y} \in R^{M \times L \times M_{max}}$ and $\boldsymbol{Z} \in R^{M \times L \times L_{max}}$, M is the number of alignments, and L is the length of individual sequence in the masked MSA.

3. Initial embedding

Each one-hot coding matrix is multiplied by a weight matrix to generate the corresponding embedding matrix:

$$\boldsymbol{H}_{token} = \boldsymbol{X}\boldsymbol{W}_{token} = \begin{bmatrix} \boldsymbol{X}[1] \\ \boldsymbol{X}[2] \\ \dots \\ \boldsymbol{X}[M] \end{bmatrix} \boldsymbol{W}_{token} = \begin{bmatrix} \boldsymbol{X}[1]\boldsymbol{W}_{token} \\ \boldsymbol{X}[2]\boldsymbol{W}_{token} \\ \dots \\ \boldsymbol{X}[M]\boldsymbol{W}_{token} \end{bmatrix} \in R^{M \times L \times D} \tag{4}$$

$$\boldsymbol{X}[i] \in R^{L \times C_{max}}, \boldsymbol{W}_{token} \in R^{C_{max} \times D}$$

$$\boldsymbol{H}_{row} = \boldsymbol{X}\boldsymbol{W}_{row} = \begin{bmatrix} \boldsymbol{Y}[1] \\ \boldsymbol{Y}[2] \\ \dots \\ \boldsymbol{Y}[M] \end{bmatrix} \boldsymbol{W}_{row} = \begin{bmatrix} \boldsymbol{Y}[1]\boldsymbol{W}_{row} \\ \boldsymbol{Y}[2]\boldsymbol{W}_{row} \\ \dots \\ \boldsymbol{Y}[M]\boldsymbol{W}_{row} \end{bmatrix} \in R^{M \times L \times D} \tag{5}$$

$$\boldsymbol{Y}[i] \in R^{L \times M_{max}}, \boldsymbol{W}_{row} \in R^{M_{max} \times D}$$

$$\boldsymbol{H}_{col} = \boldsymbol{Z}\boldsymbol{W}_{col} = \begin{bmatrix} \boldsymbol{Z}[1] \\ \boldsymbol{Z}[2] \\ \dots \\ \boldsymbol{Z}[M] \end{bmatrix} \boldsymbol{W}_{col} = \begin{bmatrix} \boldsymbol{Z}[1]\boldsymbol{W}_{col} \\ \boldsymbol{Z}[2]\boldsymbol{W}_{col} \\ \dots \\ \boldsymbol{Z}[M]\boldsymbol{W}_{col} \end{bmatrix} \in R^{M \times L \times D} \tag{6}$$

$$\boldsymbol{Z}[i] \in R^{L \times L_{max}}, \boldsymbol{W}_{col} \in R^{L_{max} \times D}$$

where $\boldsymbol{X}[i]$, $\boldsymbol{Y}[i]$ and $\boldsymbol{Z}[i]$ are the one-hot coding matrices for the i-th sequence in the masked MSA from the view of token type, row position, and column position, respectively, $\boldsymbol{H}_{token}$, $\boldsymbol{H}_{row}$, and $\boldsymbol{H}_{col}$ are token type-based, row position-based, and column position-based embedding matrices for the masked MSA, respectively, and D is the embedding dimension. In this work, $D = 768$.

Three embedding matrices are added as an initial embedding matrix $\boldsymbol{H}_{init}$:

$$\boldsymbol{H}_{init} = \boldsymbol{H}_{token} + \boldsymbol{H}_{row} + \boldsymbol{H}_{col}, \boldsymbol{H}_{init} \in R^{M \times L \times D} \tag{7}$$

4. Batch normalization and dropout

The initial embedding matrix $\boldsymbol{H}_{init}$ is fed to the batch normalization layer to generate the corresponding normalized matrix $\boldsymbol{H}_1$:

$$\boldsymbol{H}_1 = BN(\boldsymbol{H}_{init}) = \begin{bmatrix} BN(\boldsymbol{h}_{11}) & \cdots & BN(\boldsymbol{h}_{1L}) \\ \vdots & \ddots & \vdots \\ BN(\boldsymbol{h}_{M1}) & \cdots & BN(\boldsymbol{h}_{ML}) \end{bmatrix} \tag{8}$$

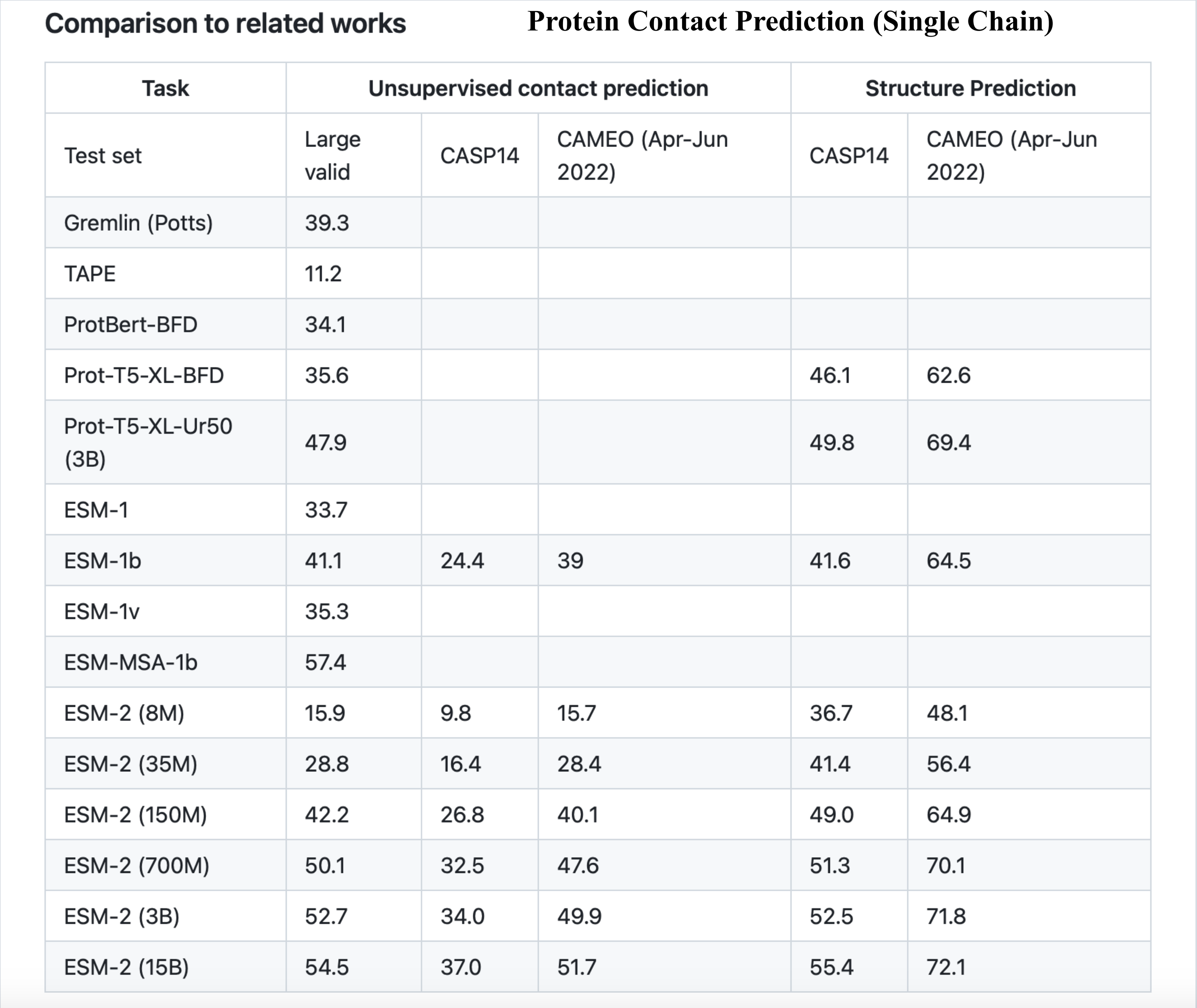
$$BN\left(\boldsymbol{h}_{ij}\right) = \gamma \cdot \frac{\boldsymbol{h}_{ij} - u_{ij}}{\sqrt{\sigma_{ij}^2 + \epsilon}} + \beta, \boldsymbol{h}_{ij} \in R^D \tag{9}$$

where $\boldsymbol{h}_{ij}$ is the initial embedding vector for the j-th position of the i-th sequence in the masked MSA, u_{ij} and σ_{ij}^2 are mean and variance for $\boldsymbol{h}_{ij}$, respectively, and γ, β, and ϵ are normalized factors.

The normalized matrix $\boldsymbol{H}_1$ is fed to dropout layer:

https://yiheng-zhu.github.io/Yiheng/papers5.html

■ 三种语言模型的在不同任务上的性能比较和分析

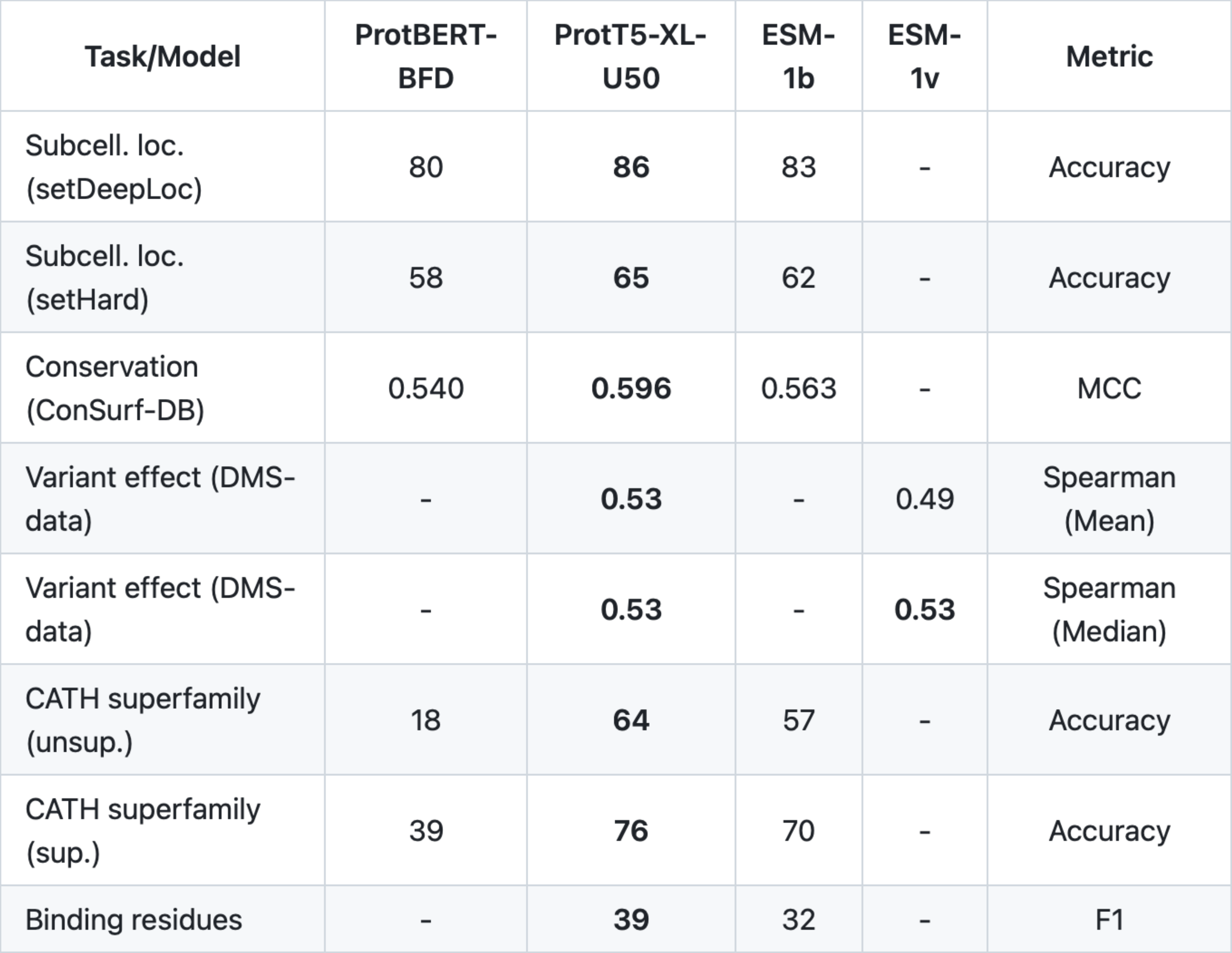


Comparison to related works

Protein Contact Prediction (Single Chain)

Task	Unsupervised contact prediction			Structure Prediction	
Test set	Large valid	CASP14	CAMEO (Apr-Jun 2022)	CASP14	CAMEO (Apr-Jun 2022)
Gremlin (Potts)	39.3				
TAPE	11.2				
ProtBert-BFD	34.1				
Prot-T5-XL-BFD	35.6			46.1	62.6
Prot-T5-XL-Ur50 (3B)	47.9			49.8	69.4
ESM-1	33.7				
ESM-1b	41.1	24.4	39	41.6	64.5
ESM-1v	35.3				
ESM-MSA-1b	57.4				
ESM-2 (8M)	15.9	9.8	15.7	36.7	48.1
ESM-2 (35M)	28.8	16.4	28.4	41.4	56.4
ESM-2 (150M)	42.2	26.8	40.1	49.0	64.9
ESM-2 (700M)	50.1	32.5	47.6	51.3	70.1
ESM-2 (3B)	52.7	34.0	49.9	52.5	71.8
ESM-2 (15B)	54.5	37.0	51.7	55.4	72.1

https://github.com/facebookresearch/esm



Task/Model	ProtBERT-BFD	ProtT5-XL-U50	ESM-1b	ESM-1v	Metric
Subcell. loc. (setDeepLoc)	80	**86**	83	-	Accuracy
Subcell. loc. (setHard)	58	**65**	62	-	Accuracy
Conservation (ConSurf-DB)	0.540	**0.596**	0.563	-	MCC
Variant effect (DMS-data)	-	**0.53**	-	0.49	Spearman (Mean)
Variant effect (DMS-data)	-	**0.53**	-	**0.53**	Spearman (Median)
CATH superfamily (unsup.)	18	**64**	57	-	Accuracy
CATH superfamily (sup.)	39	**76**	70	-	Accuracy
Binding residues	-	**39**	32	-	F1

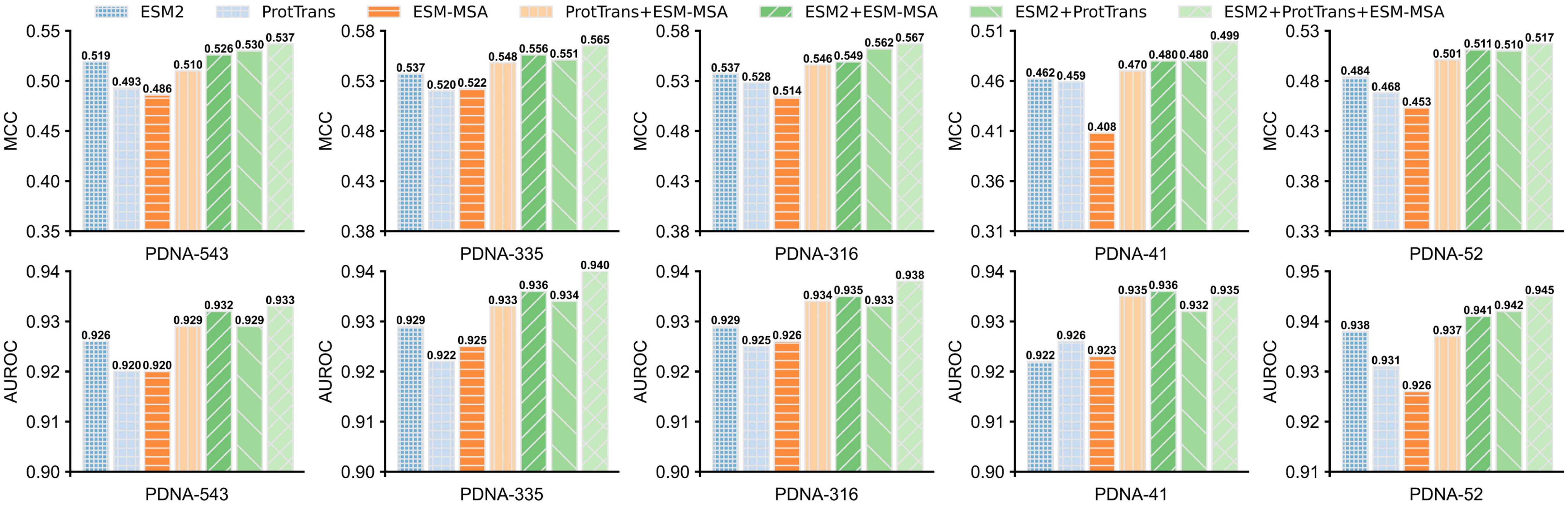
https://github.com/agemagician/ProtTrans

Protein Function Prediction (GO Prediction)

Performance comparison between ESM-1b and ESM2 on the test dataset of ATGO paper

Method	Fmax			AUPR		
	MF	BP	CC	MF	BP	CC
One-hot	0.371	0.321	0.560	0.321	0.237	0.572
ESM-1b	0.627	0.425	0.623	0.603	0.361	0.600
ESM2	**0.644**	**0.431**	**0.630**	**0.613**	**0.365**	**0.605**

Protein-DNA Binding Site Prediction



The MCC and AUROC values of seven feature embeddings on five benchmark datasets.

■ 经验总结

(1) ESM2和ProtTrans适用于蛋白质属性和功能预测领域（GO功能预测、配体绑定位点预测等）的绝大多数任务，二者结合起来使用通常能够互补，进一步提高预测精度。

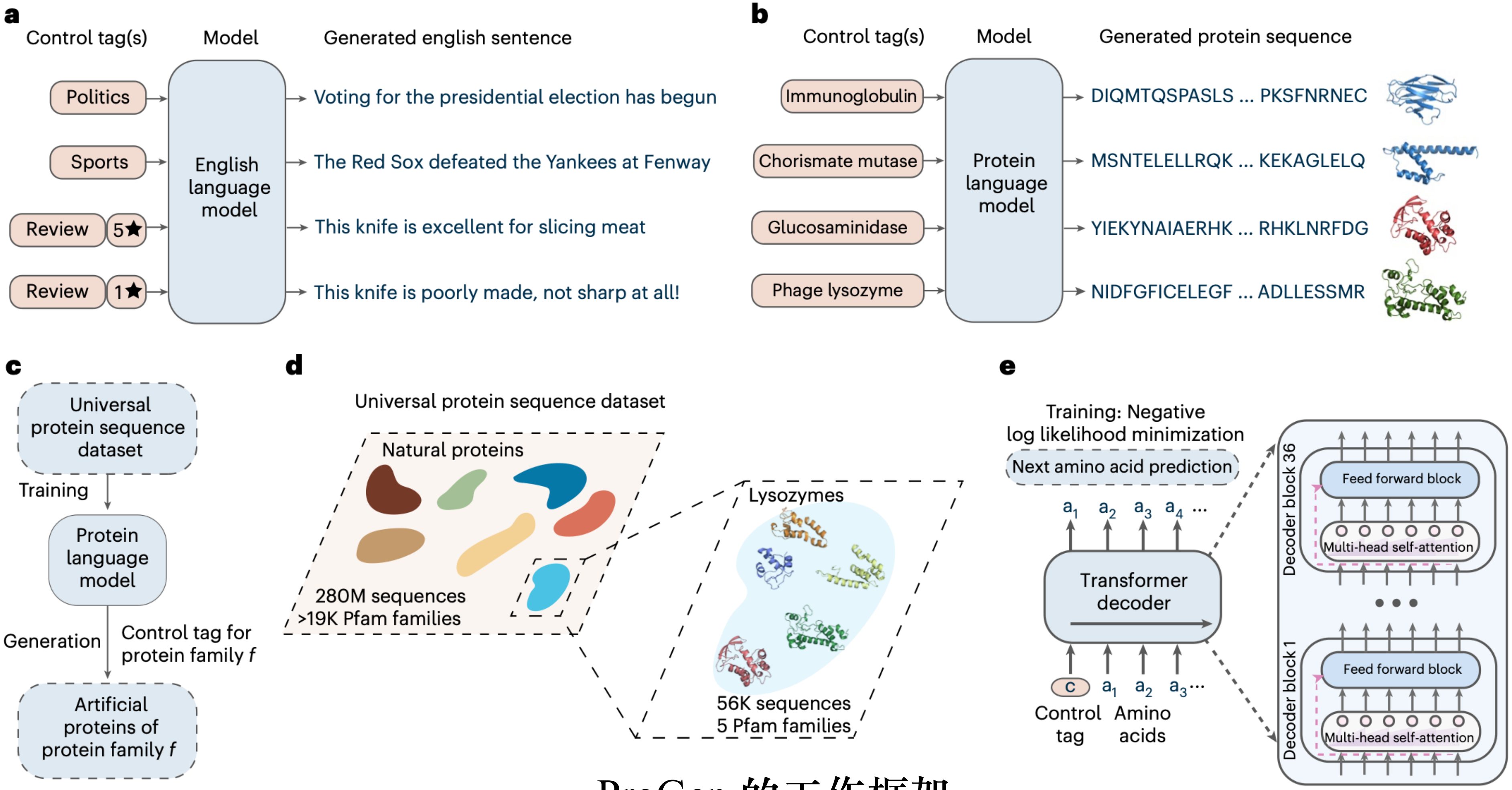
(2) ESM-MSA 适用于蛋白质结构预测领域的任务（单链接触图预测、复合物接触图预测等），效果通常好于ESM2和ProtTrans。

主要原因：

ESM-MSA是基于MSA数据集训练的，而MSA是通过hhblits工具搜索生成的。Hhblits和blast搜索MSA的方式不同，blast搜索MSA是基于序列的局部保守性（相似性），hhblits搜索MSA构建蛋白质家族序列的隐马尔可夫模型。

03 蛋白质序列生成模型

■ ProGen

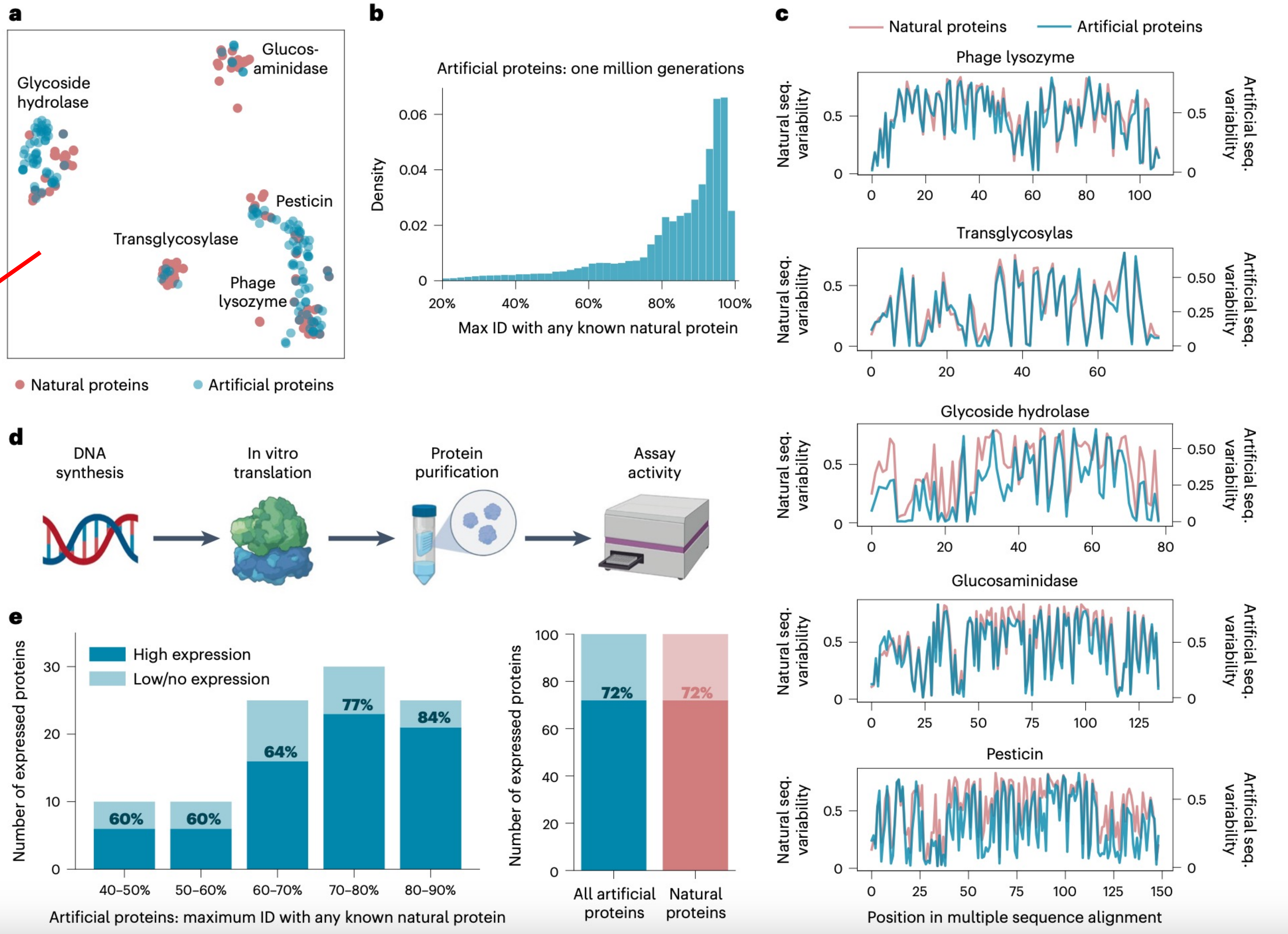

a
Control tag(s)
Model
Generated english sentence
Politics
Sports
Review
5★
Review
1★
English language model
Voting for the presidential election has begun
The Red Sox defeated the Yankees at Fenway
This knife is excellent for slicing meat
This knife is poorly made, not sharp at all!
b
Control tag(s)
Model
Generated protein sequence
Immunoglobulin
Chorismate mutase
Glucosaminidase
Phage lysozyme
Protein language model
DIQMTQSPASLS ... PKSFNRNEC
MSNTELELLRQK ... KEKAGLELQ
YIEKYNAIAERHK ... RHKLNRFDG
NIDFGFICELEGF ... ADLLESSMR
c
Universal protein sequence dataset
Training
Protein language model
Generation
Control tag for protein family f
Artificial proteins of protein family f
d
Universal protein sequence dataset
Natural proteins
280M sequences
>19K Pfam families
Lysozymes
56K sequences
5 Pfam families
e
Training: Negative log likelihood minimization
Next amino acid prediction
a1 a2 a3 a4 ...
Transformer decoder
C a1 a2 a3...
Control tag
Amino acids
Decoder block 36
Feed forward block
Multi-head self-attention
Decoder block 1
Feed forward block
Multi-head self-attention

ProGen 的工作框架

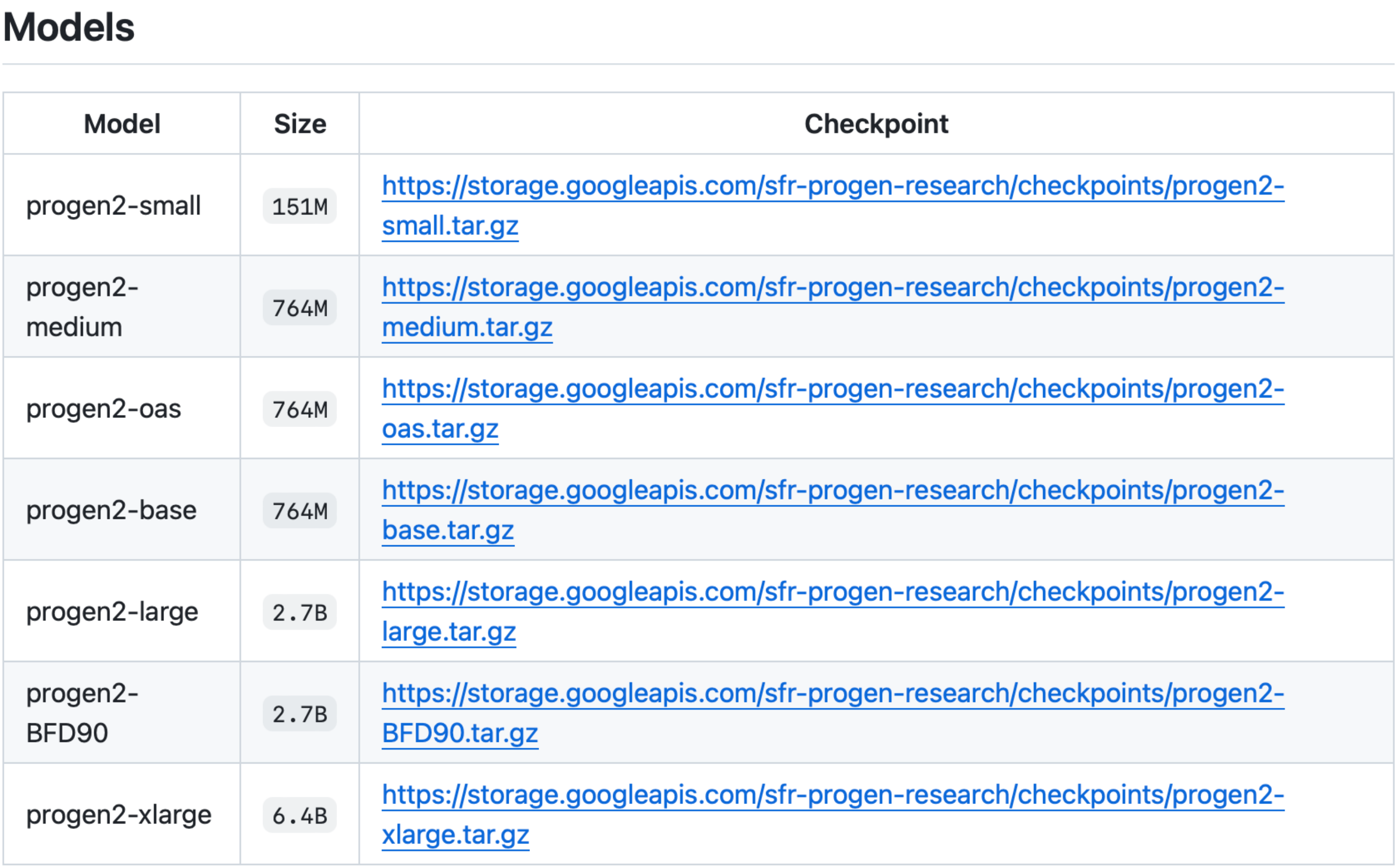
Madani A, Krause B, Greene E R, et al. Large language models generate functional protein sequences across diverse families[J]. Nature Biotechnology, 2023: 1-8. (谷歌学术引用量: 178)

5种溶菌酶家族

t-distributed stochastic neighbor embedding (*t*-SNE)



- 代码和模型下载地址：https://github.com/salesforce/progen/tree/main/progen2
- 主要应用：de novo protein design



Models

Model	Size	Checkpoint
progen2-small	151M	https://storage.googleapis.com/sfr-progen-research/checkpoints/progen2-small.tar.gz
progen2-medium	764M	https://storage.googleapis.com/sfr-progen-research/checkpoints/progen2-medium.tar.gz
progen2-oas	764M	https://storage.googleapis.com/sfr-progen-research/checkpoints/progen2-oas.tar.gz
progen2-base	764M	https://storage.googleapis.com/sfr-progen-research/checkpoints/progen2-base.tar.gz
progen2-large	2.7B	https://storage.googleapis.com/sfr-progen-research/checkpoints/progen2-large.tar.gz
progen2-BFD90	2.7B	https://storage.googleapis.com/sfr-progen-research/checkpoints/progen2-BFD90.tar.gz
progen2-xlarge	6.4B	https://storage.googleapis.com/sfr-progen-research/checkpoints/progen2-xlarge.tar.gz

- 其他蛋白质序列生成模型：ProtGPT2

Ferruz N, Schmidt S, Höcker B. ProtGPT2 is a deep unsupervised language model for protein design [J]. Nature Communications, 2022, 13(1): 4348. (谷歌学术引用量: 153)

04 核酸序列语言模型

■ Genomic Pre-trained Network (GPN)

➢ 代码和模型下载地址

https://github.com/songlab-cal/gpn (GPN, GPN-MSA)

➢ 下游任务

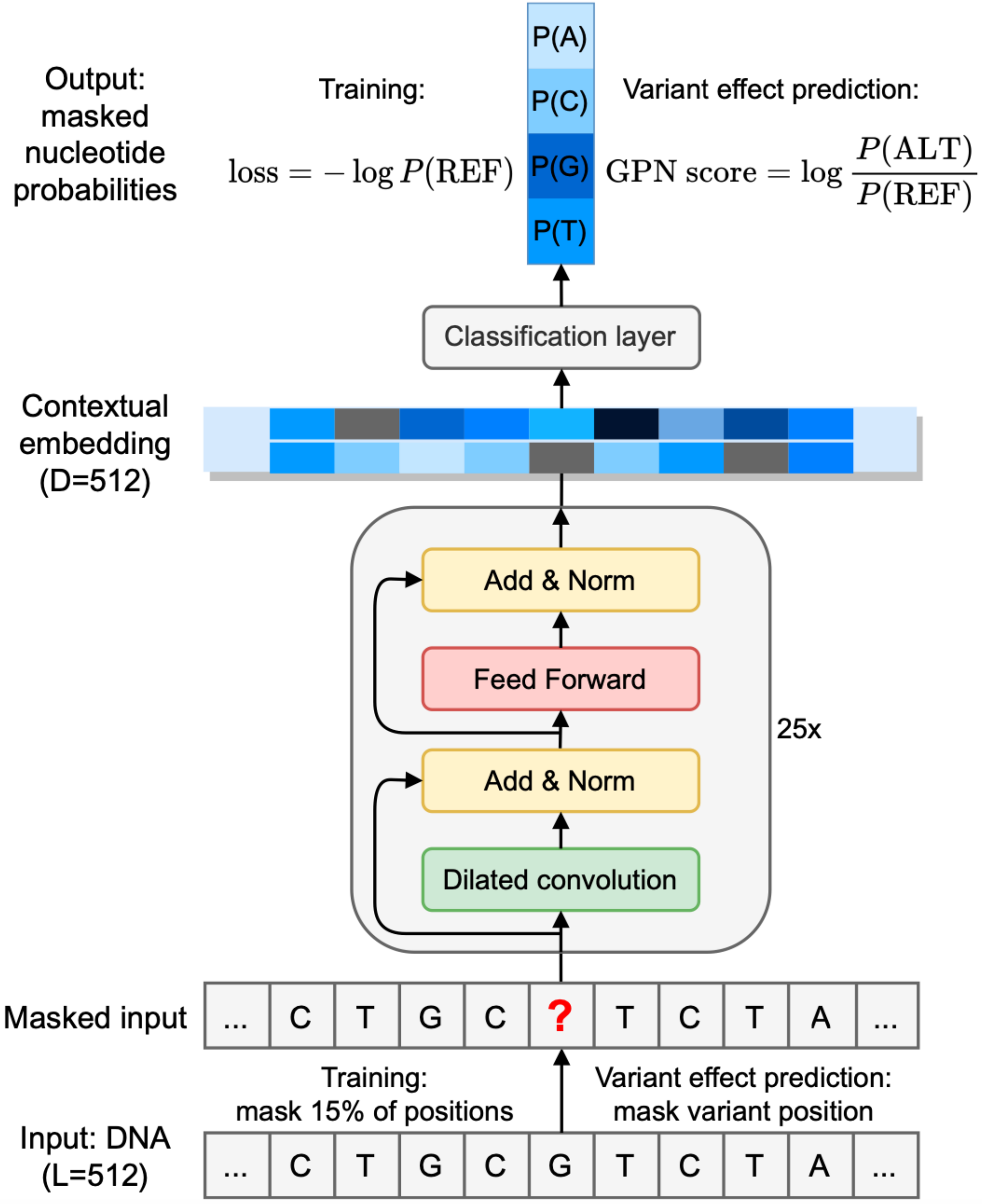
(1) Variant effect prediction

(2) RNA modification prediction

(3) Gene function prediction (????)

[1] Benegas G et al. DNA language models are powerful predictors of genome-wide variant effects [J]. Proceedings of the National Academy of Sciences, 2023, 120(44): e2311219120.

[2] Benegas G et al. GPN-MSA: an alignment-based DNA language model for genome-wide variant effect prediction [J]. bioRxiv, 2023.


Output: masked nucleotide probabilities
Training:
$\text{loss} = -\log P(\text{REF})$
P(A)
P(C)
P(G)
P(T)
Variant effect prediction:
$\text{GPN score} = \log \frac{P(\text{ALT})}{P(\text{REF})}$
Classification layer
Contextual embedding (D=512)
Add & Norm
Feed Forward
Add & Norm
Dilated convolution
25x
Masked input
... C T G C ? T C T A ...
Training: mask 15% of positions
Variant effect prediction: mask variant position
Input: DNA (L=512)
... C T G C G T C T A ...

GPN 的工作框架

02 Part two

AlphaFold3进展与应用

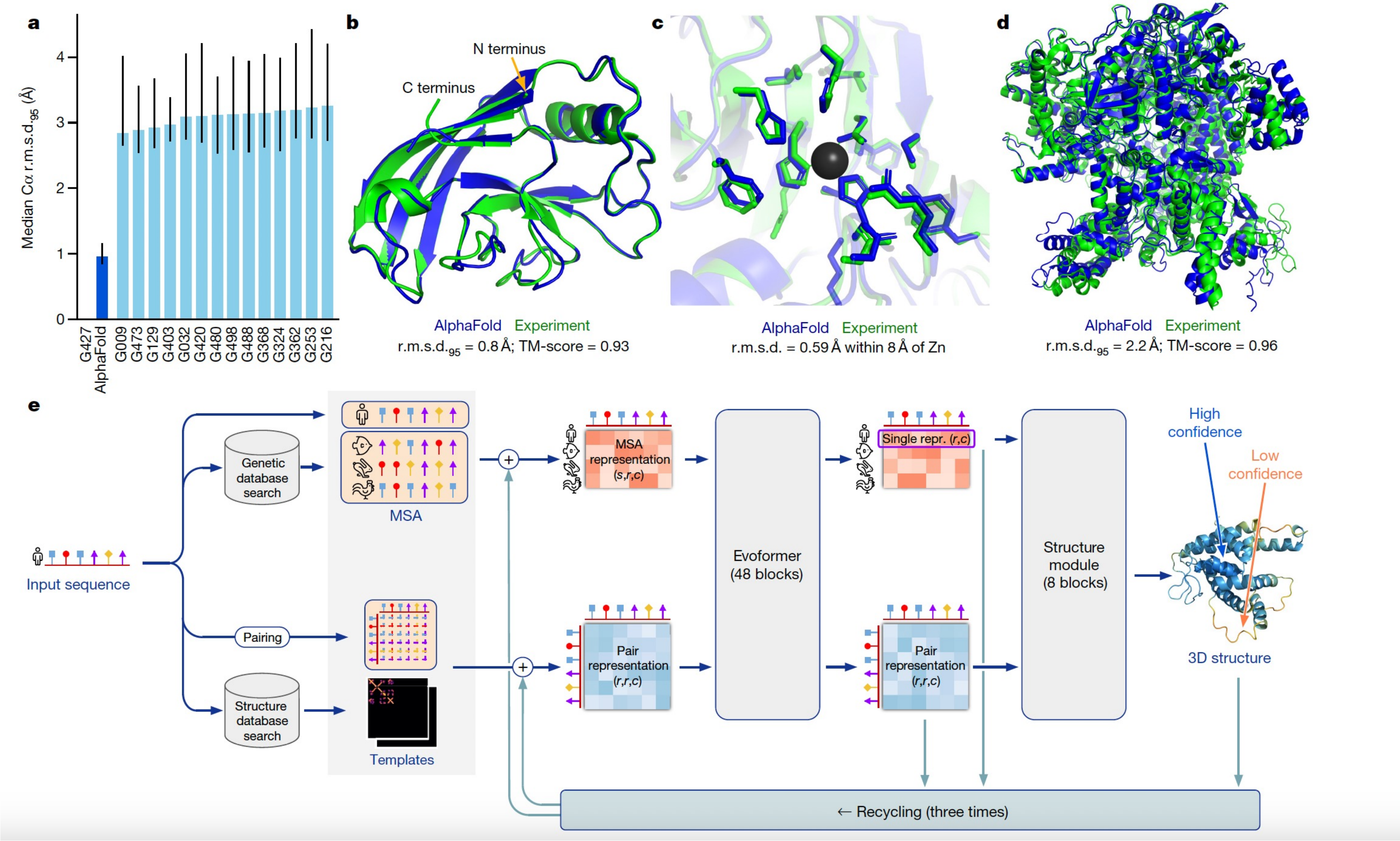
01 AlphaFold的发展历史

- CASP13 (2018): 第一代 AlphaFold，排名第一；相比于其他参赛组，优势并不明显。
- CASP14 (2020): 第二代 AlphaFold2，排名第一，平均中位数高于0.9；基本解决了单链结构预测问题。

```
>4WWC
GSHMNINKQSPIPIYYQIMEQLKTQIKNGELQPD
MPLPSEREYAEQFGISRMTVRQALSNLVNEGLL
YRLKGRGTFVSKPKMEQALQGLTSFTEDMKSR
GMTPGSRLIDYQLIDSTEELAAILGCGHPSSIHKI
TRVRLANDIPMAIESSHIPFELAGELNESHFQSSI
YDHIERYNSIPISRAKQELEPSAATTEEANILGIQ
KGAPVLLIKRTTYLQNGTAFEHAKSVYRGDRYT
FVHYMDRLS
```

?

02 AlphaFold2



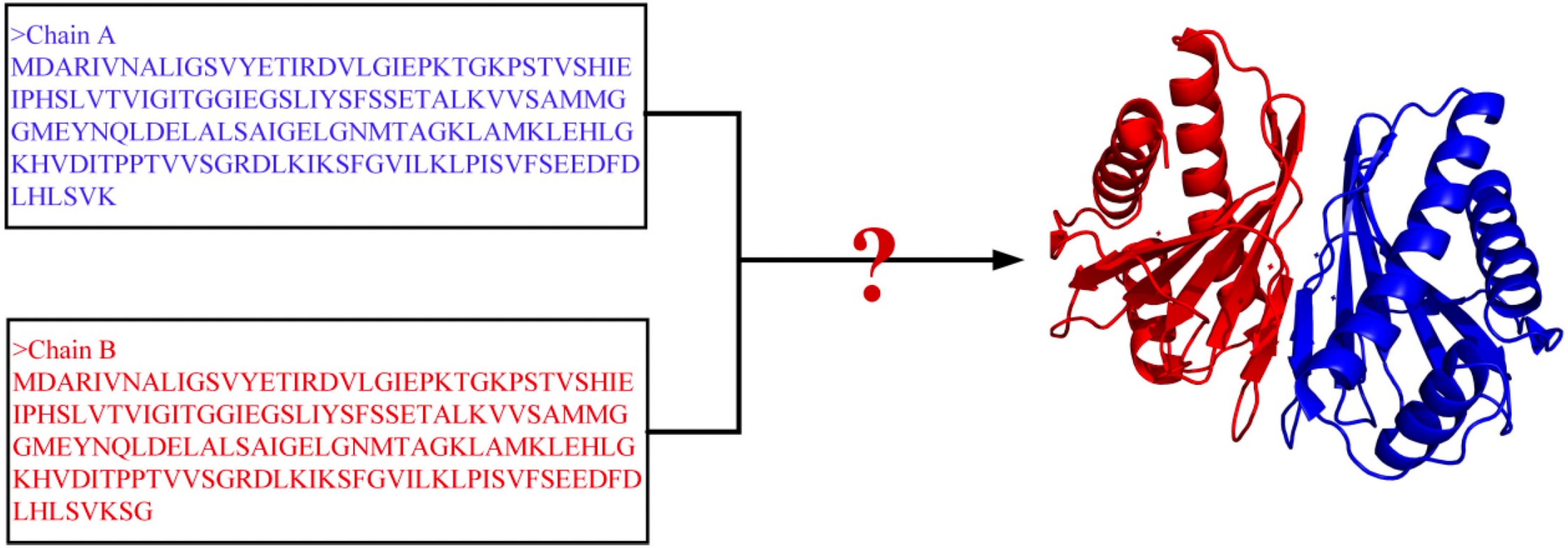
AlphaFold2 的工作框架

Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold [J]. Nature, 2021, 596(7873): 583-589. (谷歌学术引用量: 16619)

➢ AlphaFold2尚未解决的问题

（1）protein-protein complex structure prediction (AlphaFold-Multimer)

Evans R et al. Protein complex prediction with AlphaFold-Multimer [J]. bioRxiv, 2021: 2021.10. 04.463034.



>Chain A
MDARIVNALIGSVYETIRDVLGIEPKTGKPSTVSHIE
IPHSLVTVIGITGGIEGSLIYSFSSETALKVVSAMMG
GMEYNQLDELALSAIGELGNMTAGKLAMKLEHLG
KHVDITPPTVVSGRDLKIKSFGVILKLPISVFSEEDFD
LHLSVK

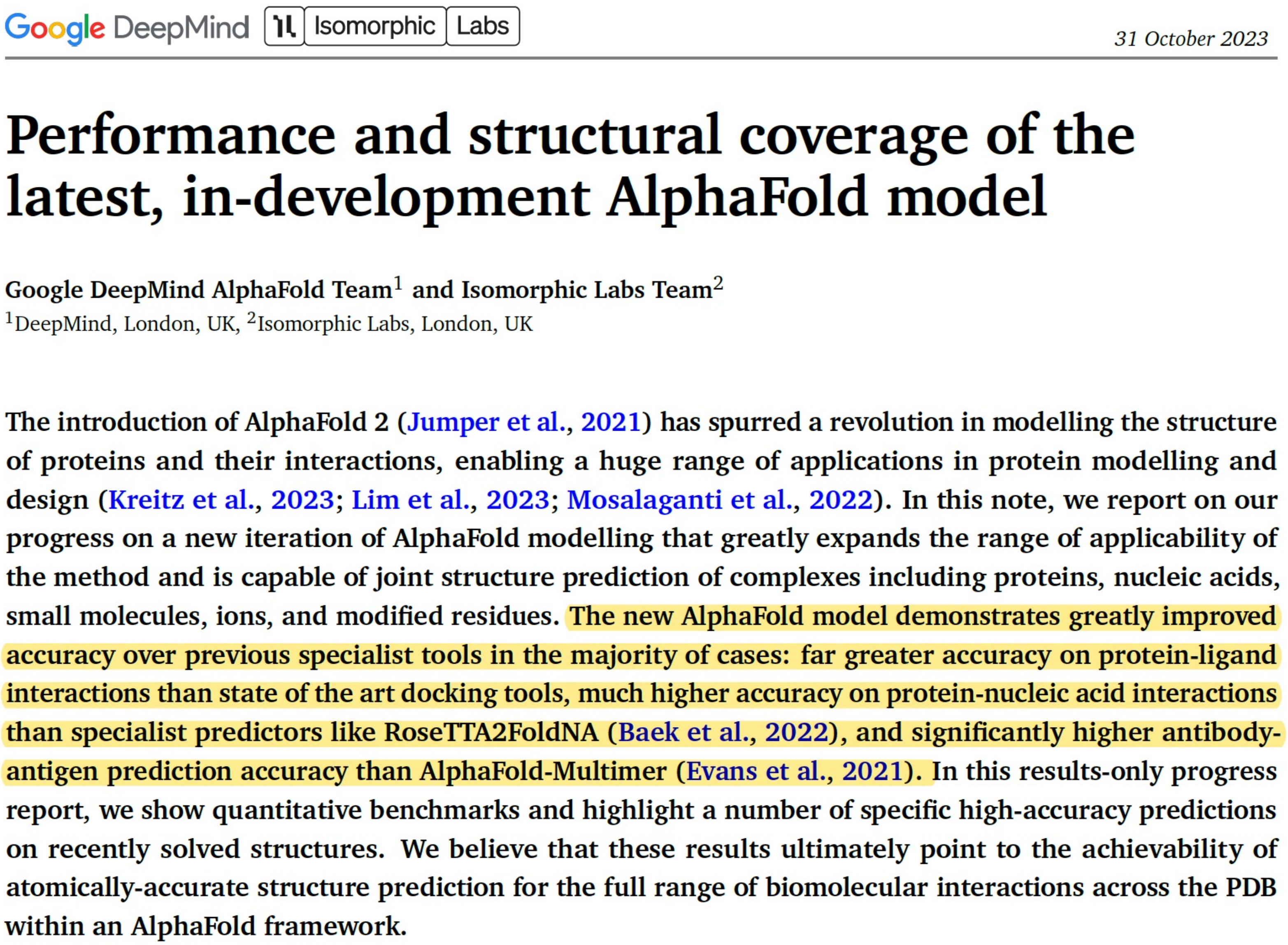
>Chain B
MDARIVNALIGSVYETIRDVLGIEPKTGKPSTVSHIE
IPHSLVTVIGITGGIEGSLIYSFSSETALKVVSAMMG
GMEYNQLDELALSAIGELGNMTAGKLAMKLEHLG
KHVDITPPTVVSGRDLKIKSFGVILKLPISVFSEEDFD
LHLSVKSG

（2）RNA structure prediction

（3）protein-ligand complex structure prediction

03 AlphaFold3

- 2023 年 10 月 31 日，DeepMind分享了最新一代AlphaFold 的进展（论文称之为AlphaFold-last）。



Google DeepMind | Isomorphic Labs

31 October 2023

Performance and structural coverage of the latest, in-development AlphaFold model

Google DeepMind AlphaFold Team[1] and Isomorphic Labs Team[2]

[1]DeepMind, London, UK, [2]Isomorphic Labs, London, UK

The introduction of AlphaFold 2 (Jumper et al., 2021) has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein modelling and design (Kreitz et al., 2023; Lim et al., 2023; Mosalaganti et al., 2022). In this note, we report on our progress on a new iteration of AlphaFold modelling that greatly expands the range of applicability of the method and is capable of joint structure prediction of complexes including proteins, nucleic acids, small molecules, ions, and modified residues. The new AlphaFold model demonstrates greatly improved accuracy over previous specialist tools in the majority of cases: far greater accuracy on protein-ligand interactions than state of the art docking tools, much higher accuracy on protein-nucleic acid interactions than specialist predictors like RoseTTA2FoldNA (Baek et al., 2022), and significantly higher antibody-antigen prediction accuracy than AlphaFold-Multimer (Evans et al., 2021). In this results-only progress report, we show quantitative benchmarks and highlight a number of specific high-accuracy predictions on recently solved structures. We believe that these results ultimately point to the achievability of atomically-accurate structure prediction for the full range of biomolecular interactions across the PDB within an AlphaFold framework.

原文链接：https://link.zhihu.com/?target=https%3A//storage.googleapis.com/deepmind-media/DeepMind.com/Blog/a-glimpse-of-the-next-generation-of-alphafold/alphafold_latest_oct2023.pdf

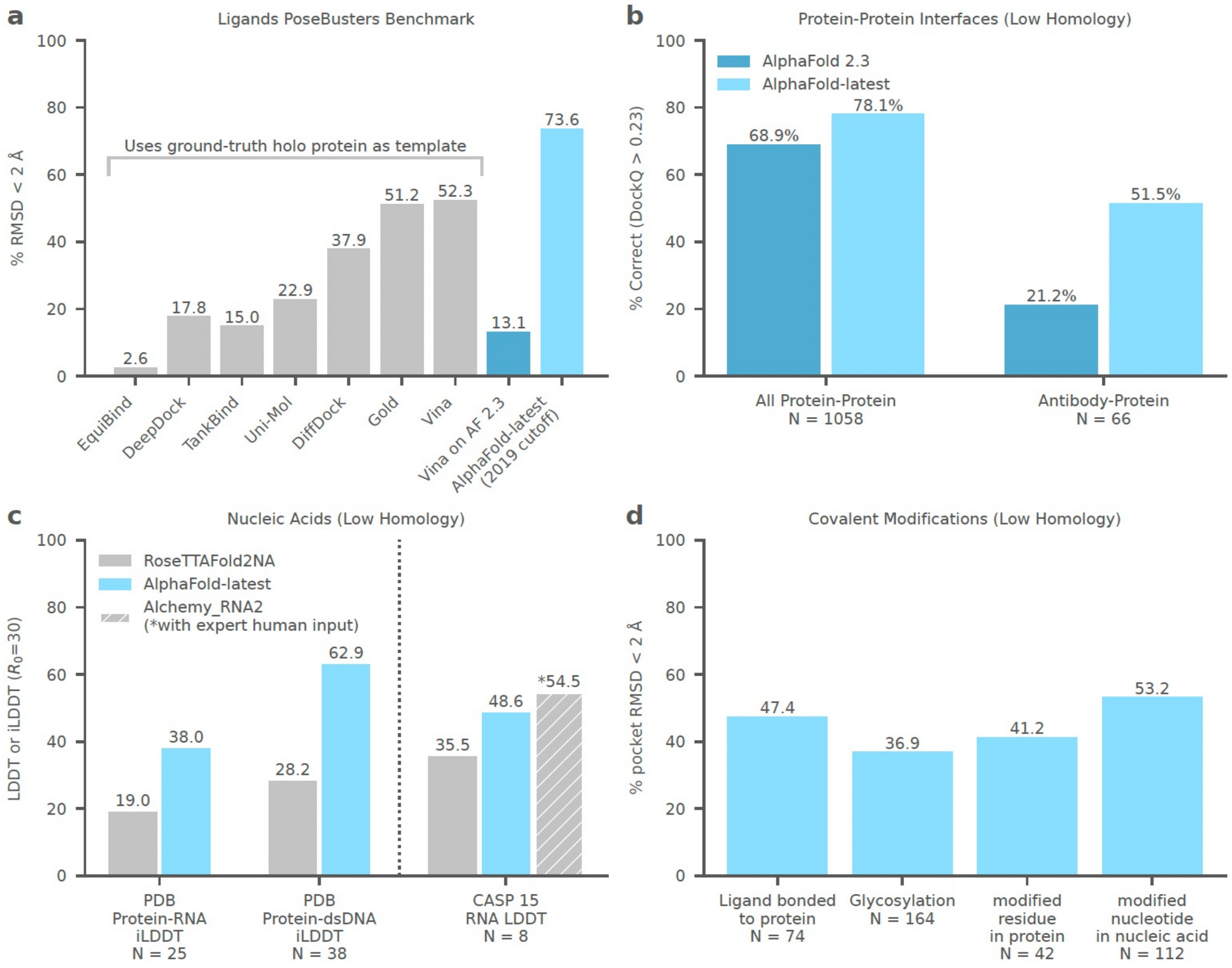


Figure 2 | Summary of AlphaFold-latest capabilities and performance. **a**, Ligand docking performance on PoseBusters benchmark set. N=428 targets. **b**, Protein-protein interaction accuracy **c**, Nucleic acid interaction and RNA accuracy. Nucleic acid LDDT is computed with an inclusion radius (R_0) of 30 Å. **d**, Accuracy on various covalent modifications.

Figure 2 demonstrates performance in four categories:

(a) AlphaFold-latest outperforms classical systems like AutoDock Vina (Eberhardt et al., 2021; Trott and Olson, 2009) on the PoseBusters benchmark (Buttenschoen et al., 2023) for ligand docking despite baselines using ground truth bound protein structures as inputs while AlphaFold-latest starts from the protein sequences and ligand identities only. See Figure 6 for a range of example ligand predictions.

(b) It improves upon AlphaFold 2.3 for protein-protein structure prediction, especially in certain categories such as antibody binding structures.

(c) On protein-nucleic acid interfaces AlphaFold-latest outperforms competing systems (Baek et al., 2022), while for RNA structure prediction it outperforms automated methods but is slightly below the top CASP15 entrant which uses manual expert intervention (AIchemy_RNA2) (Chen et al., 2022; Xiong et al., 2021).

(d) Finally, AlphaFold-latest is able to predict the structure of further entities like bonded ligands,

2. Model Inputs and Outputs

AlphaFold-latest takes as input a description of the biological assembly, with sequences for polymers and SMILES for ligands, and optionally the sequence location of covalently bonded ligands, and outputs a prediction for the 3D position of each heavy atom. Water and hydrogens are excluded. All experimental structures used for training the model were from PDB with release dates up to 2021-09-30. Templates were filtered to only those released prior to 2021-09-30.

Inputs are "tokenized" to get model inputs, with one token per standard polymer residue and one token per heavy atom for ligands and nonstandard polymer residues. The number of tokens is the primary driver of compute time and limits of prediction sizes on different hardware. We evaluate system performance on complexes up to 5,120 tokens for computational ease, but the system is capable of running larger complexes on accelerators with large amounts of memory.

Each output structure comes with per-atom, per-token-pair, and aggregated structure-level confidence measures. In addition, each entity within the structure and each interface between entities within the structure has an associated confidence measure.

- Performance on protein-ligand structure prediction

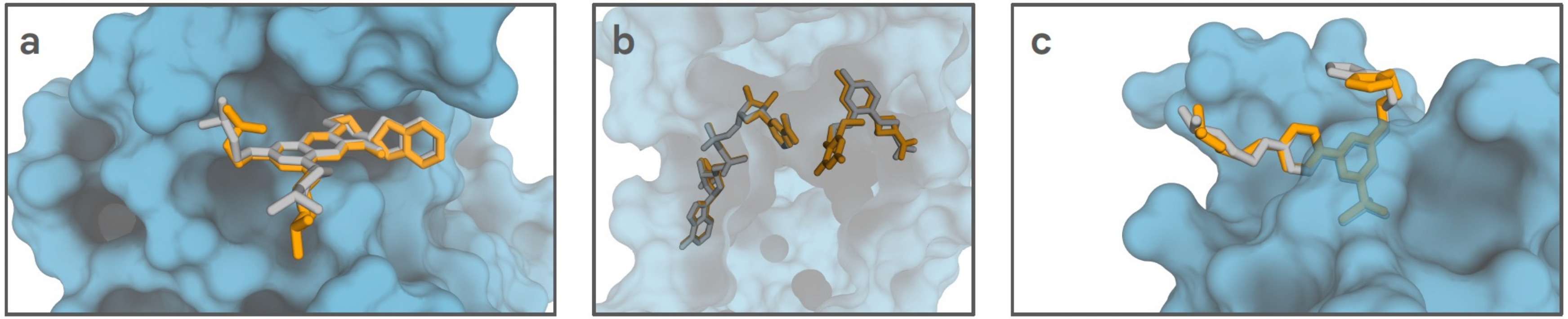


Figure 3 | Three AlphaFold-latest examples from the PoseBusters benchmark set where docking programs Vina and Gold fail to achieve accurate predictions. Surface representation of the predicted protein structure shown in blue, predicted ligand pose shown as sticks in orange, ground truth ligand pose shown as sticks in grey. **a**, PDB ID 7OCB: best docking RMSD = 4.6 Å, AlphaFold-latest RMSD = 0.96 Å. **b**, PDB ID 5SD5: best docking RMSD = 4.5 Å, AlphaFold-latest RMSD = 0.92 Å. **c**, PDB ID 7BLA: best docking RMSD = 6.3 Å, AlphaFold-latest RMSD = 2.0 Å.

RMSD: Root Mean Square Deviation

- Performance on protein-nucleic acid structure prediction

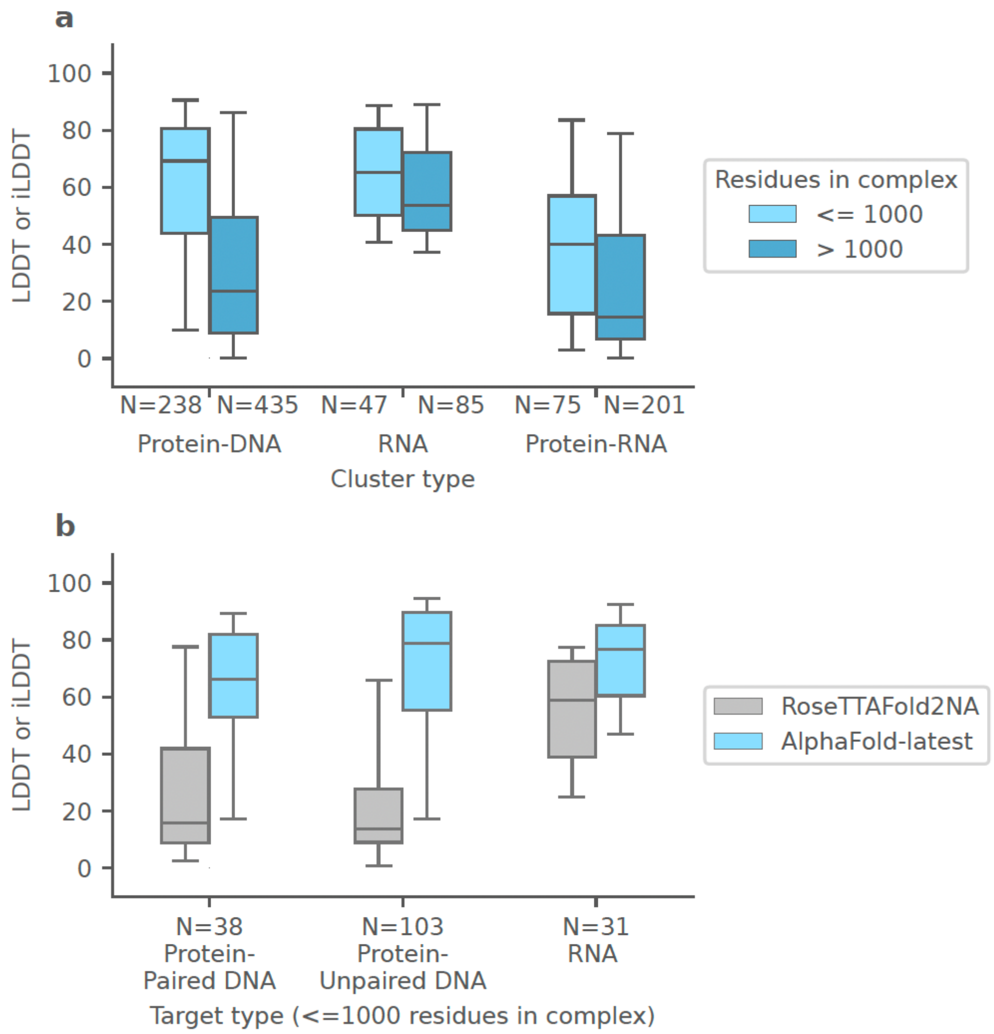


Figure 9 | Nucleic acid complex performance. **a**, Performance of AlphaFold-latest at predicting low homology protein-DNA interfaces, RNA chains, and protein-RNA interfaces from our PDB test set (averages per interface cluster, split by as many as or more than 1000 total complex residues - amino acids and nucleotides). N refers to the number of interface clusters. iLDDT used for Protein-DNA and Protein-RNA; LDDT for RNA chains. **b**, Performance of AlphaFold-latest vs RoseTTAFold2NA on low homology nucleic acid targets with up to 1000 residues from our PDB test set. N refers to the number of targets, used instead of interface clusters to separate targets with and without paired DNA. iLDDT used for Protein-DNA; LDDT for RNA chains. Box, center line, and whiskers boundaries are at (25%, 75%) intervals, median, and (5%, 95%) intervals.

03 Part three

未来研究方向

未来研究方向

结合蛋白质序列大语言模型（特征表示）和AlphaFold预测的结构，后续可以尝试：

（1）蛋白质功能预测

（2）药物设计（蛋白质-药物相互作用预测、药物筛选）

（3）蛋白质-配体亲和力预测

谢谢各位老师和同学观看

请批评指正！