# ULDNA: Integrating Unsupervised Multi-Source Language Models with LSTM-Attention Network for Protein-DNA Binding Site Prediction

Yi-Heng Zhu, Dong-Jun Yu*

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, People's Republic of China

*Correspondence should be addressed to njyudj@njust.edu.cn

## Abstract

Accurate identification of protein-DNA interactions is critical to understand the molecular mechanisms of proteins and design new drugs. We proposed a novel deep-learning method, ULDNA, to predict DNA-binding sites from protein sequences through a LSTM-attention architecture embedded with three unsupervised language models pretrained in multiple large-scale sequence databases. The method was systematically tested on 1287 proteins with DNA-binding site annotation from Protein Data Bank. Experimental results showed that ULDNA achieved a significant increase of the DNA-binding site prediction accuracy compared to the state-of-the-art approaches. Detailed data analyses showed that the major advantage of ULDNA lies in the utilization of three pre-trained transformer language models which can extract the complementary DNA-binding patterns buried in evolution diversity-based feature embeddings in residue-level. Meanwhile, the designed LSTM-attention network could further enhance the correlation between evolution diversity and protein-DNA interaction. These results demonstrated a new avenue for high-accuracy deep-learning DNA-binding site prediction that is applicable to large-scale protein-DNA binding annotation from sequence alone.

**Key words:** Protein-DNA interaction, deep learning, unsupervised language model, evolution diversity, LSTM-attention network.

## 1. Introduction

Protein-DNA interactions play critical roles in various biological processes, including gene expression and regulation, DNA replication, repair, and recombination [1, 2]. The accurate identification of protein-DNA binding residues not only contributes to understanding the molecular mechanisms of proteins, but also has important practical significance for drug design [3]. Direct determination of DNA-binding sites via biochemical experiments, such as fast ChIP5 [4], X-ray crystallography [5], and Cryo-EM [6], is typically time-consuming and laborious, and often incomplete. As a result, numerous sequenced proteins have no available DNA-binding annotation to date. As of June 2023, for example, the UniProt database [7] harbored ~246 million protein sequences, but only <0.1% of them were annotated with known DNA-binding site records using experimental evidence. To fill the gap between sequence and DNA-binding annotation, it is urgent to develop efficient computational methods for protein-DNA binding site prediction [8, 9].

Existing DNA-binding site prediction methods can be divided into two categories, i.e., template detection-based methods and machine learning-based methods [10]. In the early stage, template detection-based methods lead the trend of protein-DNA interaction prediction [11, 12]. Specifically, these methods identify DNA-binding sites through detecting the templates that have similar sequence or structure to the query. For examples, S-SITE [13] identifies sequence templates using PSI-BLAST alignment [14], while PreDNA [15] and DBD-Hunter [16] search templates through structure alignment. There exist other elegant predictors, including PreDs [17], DBD-Threader [18], DR_bind [19], and Morozov's method [20].

Template-based approaches have a common drawback: the accuracy of these methods is contingent upon the availability of templates with readily identifiable DNA-binding site annotation. To eliminate such dependence, machine learning-based methods have emerged to extract hand-crafted features from sequences and structures (e.g., position-specific scoring matrix [21] and solvent accessible surface area [22]), which can then be used by machine learning approaches (e.g., support vector machine [23] and random forest [24]) to implement DNA-binding site prediction, with typical examples including DNAPred [10], TargetDNA [25], MetaDBSite [26], and TargetS [27].

Despite the potential advantage, the prediction accuracy of many early machine learning-based methods was not satisfactory. One of the major reasons is due to the lack of informative feature representation methods, as most of the approaches are based on simple feature representations, such as amino acid coding, physiochemical properties, and evolution conservation, which cannot fully extract the complex pattern of protein-DNA interaction [28, 29]. To partly overcome this barrier, several methods, e.g., Guan's method [30], PredDBR [31], iProDNA-CapsNet [32], and GraphBind [33], utilize deep learning technology to predict DNA-binding sites. Compared to traditional machine learning approaches, one advantage of deep learning technologies is that they could extract more discriminative feature embeddings from sequences and structures through designing complex neural networks. Nevertheless, the performance of deep learning methods is often hampered by the limitation of experimental annotation data consisting of only ~4000 protein-DNA complexes from Protein Data Bank (PDB) [34]. The insufficient experimental data significantly limit the effectiveness of training the deep neural network models.

To alleviate the issue caused by the lack of annotated data, a promising approach is to utilize protein language models pre-trained through deep-learning networks on large-scale sequence databases without DNA-binding annotations. Due to the extensive sequence training and learning, important inter-residue correlation patterns, which are critical for protein-DNA interaction, can be extracted through the language models and utilized for feature embedding. Recently, several protein language models, such as TAPE [35], SeqVec [36], and Bepler's approach [37], have been emerged, often through supervised learners such as convolutional neural networks (CNNs) [38], in protein structure and function prediction tasks, with examples including the predictions of contact map [39], molecular function [40], mutation and stability [35], and GO transferals [41].

In this work, we proposed a new deep learning model, ULDNA, for high accuracy protein-DNA binding site prediction by the integration of the unsupervised protein language models from multiple information sources with the designed LSTM-attention network. Specifically, we utilize three recently proposed language models (i.e., ESM2 [42], ProtTrans [43], and ESM-MSA [44]), separately pre-trained on different large-scale sequence databases, to extract the complementary evolution diversity-based feature embeddings, which are highly associated with protein-DNA interaction. Then, a LSTM-attention network is designed to train DNA-binding site models from multi-source feature embeddings through enhancing the correlation between evolution diversity and DNA-binding pattern. ULDNA has been systematically tested on five protein-DNA binding site datasets, where the results demonstrated significant advantage on accurate DNA-binding site prediction over the current state-of-the-art of the field. The standalone package and an online server of ULDNA are made freely available through URL http://csbio.njust.edu.cn/bioinf/uldna/.

## 2. Materials and methods

### 2.1 Benchmark datasets

The proposed methods were evaluated by five protein-DNA binding site datasets, including PDNA-543 [25], PDNA-41 [25], PDNA-335 [27], PDNA-52 [27], and PDNA-316 [26], from previous works.

PDNA-543 and PDNA-41 separately consist of 543 and 41 DNA-binding protein chains, which were deposited in the PDB before and after October 10, 2014, respectively. Here, a sequence identity cut-off 30% has been used to filter out the redundant proteins within each dataset and between different datasets using the CD-HIT program [45].

PDNA-335 and PDNA-52 contain 335 and 52 DNA-binding chains, respectively, which are released in PDB before and after March 10, 2010. The sequence identity within each dataset and between different datasets is reduced to 40% through PISCES software [46].

PDNA-316 collects 316 DNA-binding chains before 2011, where the maximal pairwise sequence identity of proteins is culled to 30% using CD-HIT [45]. The detailed statistical summary of five datasets is presented in Table 1.

Table 1. Statistical summary of five protein-DNA binding site datasets.

| Dataset | No. of Sequences | No. of DNA-binding residues | No. of No-DNA-binding residues |
|---|---|---|---|
| PDNA-543 | 543 | 9,549 | 134,995 |
| PDNA-41 | 41 | 734 | 14,021 |
| PDNA-335 | 335 | 6,461 | 71,320 |
| PDNA-52 | 52 | 973 | 16,225 |
| PDNA-316 | 316 | 5,609 | 67,109 |

### 2.2 The framework of ULDNA

ULDNA is a deep learning-based method for protein-DNA binding site prediction, with input being a query amino acid sequence and output including confidence scores of belonging DNA-binding sites. As shown in Figure 1, ULDNA consists of two procedures, including feature embedding extraction using multi-source language models and DNA-binding site prediction using LSTM-attention network.

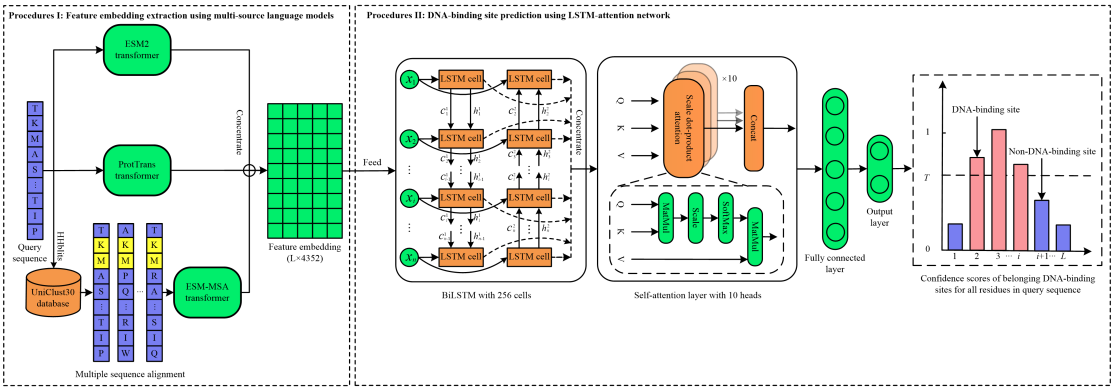
Procedures I: Feature embedding extraction using multi-source language models
ESM2 transformer
ProtTrans transformer
ESM-MSA transformer
Concentrate
T K M A S ⋮ T I P
Query sequence
HHblits
UniClust30 database
T K M A S ⋮ T I P
A K M P Q ⋮ R I W
T K M R A ⋮ S I Q
Multiple sequence alignment
Feature embedding (L×4352)
Feed
Procedures II: DNA-binding site prediction using LSTM-attention network
$x_1$
$x_2$
$x_i$
$x_n$
LSTM cell
$c_1^1$ $h_1^1$ $c_2^2$ $h_2^2$
$c_2^1$ $h_2^1$ $c_3^2$ $h_3^2$
$c_{i-1}^1$ $h_{i-1}^1$ $c_i^2$ $h_i^2$
$c_i^1$ $h_i^1$ $c_n^2$ $h_n^2$
$c_{n-1}^1$ $h_{n-1}^1$
Concentrate
BiLSTM with 256 cells
Q K V
Scale dot-product attention
×10
Concat
MatMul
Scale
SoftMax
MatMul
Self-attention layer with 10 heads
Fully connected layer
Output layer
DNA-binding site
Non-DNA-binding site
1
T
0
1 2 3 ··· i i+1··· L
Confidence scores of belonging DNA-binding sites for all residues in query sequence

Figure 1. The workflow of ULDNA

**Procedure I: Feature embedding extraction using multi-source language models.** The input sequence is fed to ESM2 [42] and ProtTrans [43] transformers to output two feature embedding matrices with the scales of $L \times 2560$ and $L \times 1024$, respectively; Meanwhile, we search the multiple sequence alignment (MSA) of the input sequence from UniClust30 database [47], which is further fed to ESM-MSA transformer [44] to generate another feature embedding matrix with the scale of $L \times 768$, where $L$ is the length of input sequence, 2560, 1024 and 768 are preset hyper-parameters in transformer models. ESM2, ProtTrans, and ESM-MSA are both unsupervised attention networks with 36, 24, and 12 layers, respectively, and separately trained on Uniref50 [48], UniClust30 & Uniref50, and BFD (Big Fantastic Database) [49] & Uniref50, respectively, where "&" means that two databases are both used to train a transformer. Each transformer has learnt abundant evolution knowledge from millions of sequences and could encode the input sequence (or MSA) as a feature embedding matrix with evolution diversity. Considering that the evolution knowledge from multiple database sources could be complementary, we concentrate the above-mentioned three feature embedding matrices from different transformer models as a combination embedding matrix with the scale of $L \times 4352$.

**Procedure II: DNA-binding site prediction using LSTM-attention network.** The concentrated feature embedding is fed to a designed LSTM-attention network to generate a score vector with $L$ dimensions, indicating the confidence scores of belonging DNA-binding sites for all residues in query sequence. In LSTM-attention network, a BiLSTM layer and a self-attention layer are combined to enhance the correlation between evolution diversity and DNA-binding in residue-level to improve DNA-binding prediction.

### 2.3 Unsupervised protein language models

The architecture of ESM2 transformer [42] is illustrated in Figure S1, with input and output being a query amino acid sequence and an evolution diversity-based feature embedding matrix, respectively. ESM2 includes 36 attention layers, each of which consists of 20 attention heads and a feed-forward network (FFN). In each attention head, the scale dot-product attention is performed to learn the evolution correlation between amino acids in the query sequence from an individual view. Then, the FFN fuses the evolution knowledge from all attention heads to capture the evolution diversity for the entire sequence. The ESM2 model with 3 billion parameters was trained on over 60 million proteins from UniRef50 database, as carefully described in Text S1.

ProtTrans transformer [43] shares the similar architecture to ESM2, with including 24 layers, each of which consists of 32 attention heads. The ProtTrans model with 3 billion parameters was trained on over 45 million proteins from BFD and UniRef50 databases.

ESM-MSA transformer [44] is designed to extract the co-evolution-based feature embedding matrix for a MSA, as shown in Figure S2. ESM-MSA consists of 12 attention blocks, each of which contains a row-attention layer and a column-attention layer which separately learn the co-evolution correlation between amino acids in sequence-level and position-level. The ESM-MSA model with 100 million parameters was trained on over 26 million MSAs from Unclust30 and UniRef50 databases, with details in Text S2.

### 2.4 LSTM-attention network

As shown in Figure 1, the designed LSTM-attention network consists of a BiLSTM layer, a self-attention layer, a fully connected layer, and an output layer.

The BiLSTM layer includes a forward LSTM and a backward LSTM, which have the same architecture consisting of 256 cells with reverse propagation directions. Each LSTM cell is mainly composed of two states (i.e., cell state $c$ and hidden state $h$) and three gates (i.e., forget gate $f$, input gate $i$, and output gate $o$). Cell state and hidden state are separately used to store and output the signals at the current time-step. Forget gate, input gate, and output gate are used to control the ratios of incorporating history signal, inputting current signal, and outputting updated signal, respectively. Specifically, at time-step $t$ ($t \leq L$, $L$ is the length of input sequence), the above-mentioned states and gates are computed as follows:

$$h_t = o_t \cdot \tanh\,(C_t) \tag{1}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t' \tag{2}$$

$$C_t' = \tanh\left(w_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{3}$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{4}$$

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{6}$$

where $C_{t-1}$ and $h_{t-1}$ are cell state and hidden state, respectively, at the time-step $t-1$, $x_t$ is the input vector at the time-step $t$ (i.e., the feature embedding vector with 4352 dimensions of the $t$-th residue in the query sequence for DNA-binding prediction), $w_*$ is the weight vector, $b_*$ is the bias, $[\,,]$ is concentration operation between two vectors, and $\sigma$ is the Sigmoid function. The output of BiLSTM layer is represented as a $L \times 512$ matrix through concentrating the hidden states in all LSTM cells at all time-steps.

The self-attention layer consists of 10 attention heads, each of which performs the scale dot-product attention as follows:

$$A_i = SoftMax(M_i^Q \cdot M_i^K / \sqrt{d_i})\ M_i^V \tag{7}$$

$$M_i^Q = H \cdot W_i^Q,\ M_i^K = H \cdot W_i^K,\ M_i^V = H \cdot W_i^V \tag{8}$$

where $H$ is the output matrix by the BiLSTM layer, $A_i$ is a $L \times 64$ attention matrix in the $i$-th attention head, $M_i^Q$, $M_i^K$, and $M_i^V$ are Query, Key, and Value matrices, respectively, $M_i^Q \cdot M_i^K$ is a $L \times L$ weight matrix measuring the position-correlation for each amino acid pair in the query sequence, and $d_i$ is the scale parameter.

The attention matrices in all of 10 heads are concentrated and then fed to a fully connected layer with 1024 neurons, followed by an output layer with 1 neuron:

$$A = A_1 A_2 \dots A_{10} \tag{9}$$

$$F = Relu(W_a \cdot A + b_a) \tag{10}$$

$$s = \sigma(W_s \cdot F + b_s) \tag{11}$$

where $s$ is a score vector with $L$ dimensions, indicating the confidence scores of belonging DNA-binding sites for all residues for the query sequence.

The cross-entropy loss [50] is used as the training loss:

$$Loss = \frac{1}{L} \cdot \sum_{i=1}^{L} ((y_i \cdot \log(s_i) + (1 - y_i) \cdot \log(1 - s_i)) \tag{12}$$

where $s_i$ is the confidence score of belonging DNA-binding site for the $i$-th residue in the query sequence; $y_i = 1$, if the $i$-th residue is a DNA-binding site in the experimental annotation; otherwise, $y_i = 0$. We minimize the loss function to optimize ULDNA pipeline using Adam optimization algorithm [51].

### 2.5 Evaluation indices

Four indices, i.e., Sensitivity (Sen), Specificity (Spe), Accuracy (Acc), and Mathew's Correlation Coefficient (MCC), are utilized to evaluate the proposed methods:

$$Sen = TP/(TP + FN) \tag{13}$$

$$Spe = TN/(TN + FP) \tag{14}$$

$$Acc = (TP + TN)/(TP + FP + TN + FN) \tag{15}$$

$$MCC = (TP \times TN - FP \times FN)/\sqrt{(TP + FP) \times (TN + FN) \times (TP + FN) \times (TN + FP)} \tag{16}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Because the above four indices are threshold-dependent, it is critical to select an appropriate threshold for fair comparisons between various predictors. In this work, we select the threshold which maximizes the value of MCC over ten-fold cross-validation. Moreover, a threshold-independent index, i.e., area under the receiver operating characteristic curve (AUROC), is used to evaluate the overall prediction performances of predictors.

## 3. Results and discussions

### 3.1 Comparison with existing DNA-binding site predictors

To demonstrate the strong performance of the proposed ULDNA, we compared it with 12 start-of-the-art DNA-binding site predictors, including BindN [52], ProteDNA [53], BindN+ [54], MetaDBSite [26], DP-Bind [55], DNABind [56], TargetDNA [25], iProDNA-CapsNet [32], DNAPred [10], Guan's method [30], COACH [13], and PredDBR [31], on PDNA-41 test dataset under independent validation, as shown in Table 1. It could be found that ULDNA achieves the highest MCC values among all of 13 competing methods. Compared to the second-best performer, i.e., PredDBR (a recently proposed deep learning model), ULDNA gains 43.9% average improvement of MCC values under three thresholds, respectively. More importantly, four evaluation metrics of ULDNA are both higher than those of PredDBR under $Sen \approx Spe$ and $Spe \approx 0.95$. Meanwhile, a similar trend but with more significant distinctions can be observed in comparison with other predictors. Taking DNAPred as an example, ULDNA shares 9.3%, 18.6%, 17.6%, 76.2%, and 9.0% improvements for Sen, Spe, Acc, MCC, and AUROC values, respectively, under $Sen \approx Spe$. It cannot escape from our notice that ProteDNA obtains the highest Spe value (0.998) but with the lowest Sen (0.048). The underlying reason is that ProteDNA predict too many false negatives.

Table 1. Performance comparisons between ULDNA and 12 competing predictors on PDNA-41 under independent validation.

| Method | Sen | Spe | Acc | MCC | AUROC |
|---|---|---|---|---|---|
| BindN [a] | 0.456 | 0.809 | 0.792 | 0.143 | - |
| ProteDNA [a] | 0.048 | **0.998** | 0.951 | 0.160 | - |
| BindN+ ($Spe \approx 0.95$) [a] | 0.241 | 0.951 | 0.916 | 0.178 | - |
| BindN+ ($Spe \approx 0.85$) [a] | 0.508 | 0.854 | 0.837 | 0.213 | - |
| MetaDBSite [a] | 0.342 | 0.934 | 0.904 | 0.221 | - |
| DP-Bind [a] | 0.617 | 0.824 | 0.814 | 0.241 | - |
| DNABind [a] | 0.702 | 0.803 | 0.798 | 0.264 | - |
| TargetDNA ($Sen \approx Spe$) [a] | 0.602 | 0.858 | 0.845 | 0.269 | - |
| TargetDNA ($Spe \approx 0.95$) [a] | 0.455 | 0.933 | 0.909 | 0.300 | - |
| iProDNA-CapsNet ($Sen \approx Spe$) [b] | 0.754 | 0.753 | 0.753 | 0.245 | - |
| iProDNA-CapsNet ($Spe \approx 0.95$) [b] | 0.422 | 0.949 | 0.924 | 0.315 | - |
| DNAPred ($Sen \approx Spe$) [c] | 0.761 | 0.767 | 0.761 | 0.260 | 0.858 |
| DNAPred ($Spe \approx 0.95$) [c] | 0.447 | 0.949 | 0.924 | 0.337 | 0.858 |
| Guan's method [d] | 0.476 | 0.964 | 0.949 | 0.357 | - |
| COACH [e] | 0.462 | 0.951 | 0.927 | 0.352 | - |
| PredDBR ($Sen \approx Spe$) [e] | 0.764 | 0.758 | 0.758 | 0.264 | - |
| PredDBR ($Spe \approx 0.95$) [e] | 0.431 | 0.958 | 0.931 | 0.351 | - |
| PredDBR (threshold = 0.5) [e] | 0.391 | 0.968 | 0.939 | 0.359 | - |
| ULDNA ($Sen \approx Spe$) | **0.824** | 0.899 | 0.895 | 0.458 | **0.935** |
| ULDNA ($Spe \approx 0.95$) | 0.556 | 0.970 | 0.950 | **0.499** | **0.935** |
| ULDNA (threshold = 0.5) | 0.271 | 0.994 | **0.958** | 0.417 | **0.935** |

[a] Results excerpted from TargetDNA [25]; [b] Results excerpted from iProDNA-CapsNet [32]; [c] Results excerpted from DNAPred [10]; [d] Results excerpted from Guan et al [30]; [e] Results excerpted from PredDBR [31]; "$Sen \approx Spe$" and "$Spe \approx 0.95$" mean the thresholds that make $Sen \approx Spe$ and "$Spe \approx 0.95$", respectively, on PDNA-543 training dataset over ten-fold cross-validation. Bold fonts highlight the best performer in each evaluation metric.

Table 2 summarizes the performance comparison among DNABR [29], MetaDBSite [26], TargetS [27], DNAPred [10], COACH [13], PredDBR [31], and ULDNA on PDNA-52 test dataset under independent validation, where ULDNA achieves the highest MCC value among all control methods. Specifically, the improvements of MCC values between ULDNA and other 6 predictors range from 14.6% to 179.5%.

We further compare our method with all the above mentioned methods as well as other 4 competing methods, including EC-RUS [57], DBS-PRED [58], DISIS [59] and BindN-rf [28], on three training datasets (i.e., PDNA-543, PDNA-335, and PDNA-316) over cross-validation, as listed in Tables S1, S2, and S3. Again, the proposed ULDNA outperforms all other methods.

Table 2. Performance comparisons between ULDNA and 6 competing

predictors on PDNA-52 under independent validation.

| Method | Sen | Spe | Acc | MCC | AUROC |
|---|---|---|---|---|---|
| DNABR [a] | 0.407 | 0.873 | 0.846 | 0.185 | - |
| MetaDBSite [a] | 0.580 | 0.764 | 0.752 | 0.192 | - |
| TargetS [a] | 0.413 | **0.965** | 0.933 | 0.377 | 0.836 |
| DNAPred [b] | 0.518 | 0.949 | 0.925 | 0.405 | 0.876 |
| COACH [c] | 0.599 | 0.935 | 0.916 | 0.420 | - |
| PredDBR [c] | 0.539 | 0.958 | **0.935** | 0.451 | - |
| ULDNA | **0.704** | 0.944 | 0.931 | **0.517** | **0.945** |

[a] Results excerpted from TargetDNA [25]; [b] Results excerpted from DNAPred [10]; [c] Results excerpted from PredDBR [31]; Bold fonts highlight the best performer in each evaluation metric.

### 3.2 Contribution analysis of different protein language models

To analyze the contributions of three protein language models (i.e., ESM2, ProtTrans, and ESM-MSA) in DNA-binding site prediction, we further benchmarked the designed LSTM-attention network with seven different feature embeddings, respectively, including three individual embeddings from ESM2, ProtTrans, and ESM-MSA, and four combination embeddings from ProtTrans + ESM-MSA (PE), ESM2 + ESM-MSA (EE), ESM2 + ProtTrans (EP), and ESM2 + ProtTrans + ESM-MSA (EPE), where "+" means the individual feature embeddings from different language models are concentrated as a combination embeddings. Figure 1 illustrates the performance of seven feature embeddings on three training datasets (i.e., PDNA-543, PDNA-335, and PDNA-316) under cross-validation and two test datasets (i.e., PDNA-41 and PDNA-52) under independent validation.

It could be found that EPE achieves the best performance among seven feature embeddings. From the view of MCC values, EPE separately gains 5.8%, 8.8%, 13.1%, 3.2%, 2.4%, and 2.0% average improvements on five datasets in comparison with ESM2, ProtTrans, ESM-MSA, PE, EE, and EP, respectively. With respect to AUROC values, EPE occupies the top-1 positions on four out of five datasets. Moreover, ESM2 shows the highest MCC and AUROC values among three individual embeddings; meanwhile, the largest increase is caused by adding ESM2 to PE on each dataset.

These data demonstrate the following two conclusions. First, three language models pretrained on different large-scale sequence databases are complementary to improve DNA-binding site prediction. Second, ESM2 made the most important contribution among three language models.

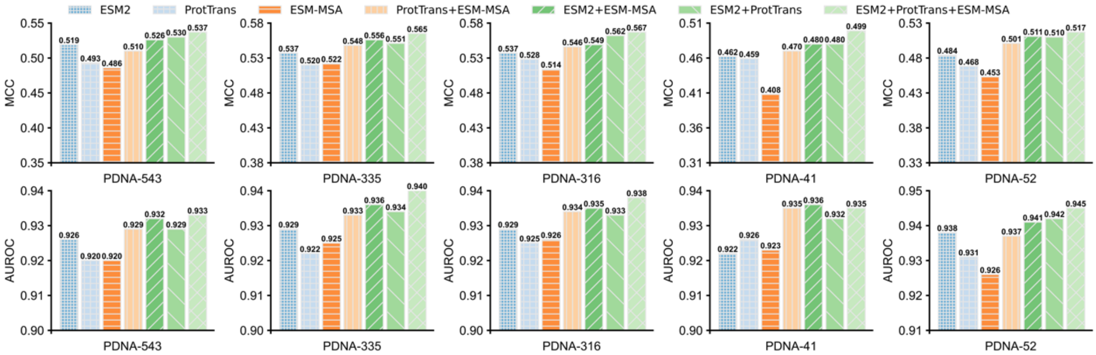

Figure 1. The MCC and AUROC values of seven feature embeddings on five benchmark datasets.

### 3.3 Ablation study

To analyze the contributions of algorithmic innovations in ULDNA to its improved performance, we design an ablation study, in which we start from a baseline model (M0) and incrementally add algorithmic components of ULDNA to build two advanced models (M1 and M2, with M2 = ULDNA). The pipelines of the three models are designed as follows (see Figure S1 for the architectures):

**M0**: Model is trained on BiLSTM consisting of 256 cells with a one-hot coding matrix [60] extracted from the input sequence, followed by a fully connected network, in which an output layer with SoftMax function [61] is used to generate the confidence scores of belonging DNA-binding sites for all residues in the input sequence. In the training stage, the cross-entropy loss [62] is used as the loss function, as described in Eq 9.

**M1**: We replace the one-hot coding matrix by a combination feature embedding matrix concentrated by three individual embeddings from ESM2, ProtTrans, and ESM-MSA. This combination embedding is further fed to the BiLSTM architecture used in M0 to output the confidence scores of belonging DNA-binding sites.

**M2 (M2=ULDNA)**: We add a self-attention layer consisting of 10 attention heads and a feed-forward network in the BiLSTM used in M1.

Figure 2 summarizes the performance of three ablation models on three training datasets under cross-validation and two test datasets under independent validation, where we run each model for 10 times and then used the average of all predictions as the final-result. Compared with M0, M1 achieves a significant gain with the average MCC and AUROC values increased by 148.4% and 23.4%, respectively, on five datasets, demonstrating that the protein language models are critical to improve DNA-binding site prediction of the ULDNA pipeline. After adding the self-attention layer in

M1, the corresponding MCC values are increased on average by 1.3% on five datasets. The AUROC values of M2 cannot be further improved and even be slightly degraded on PDNA-543 and PDNA-41 in comparison with M1, but the corresponding values are sustainably increased on other three datasets. These observations indicate that the additional self-attention layer is helpful for enhancing the overall performance of function prediction, although less significant than the protein language models.

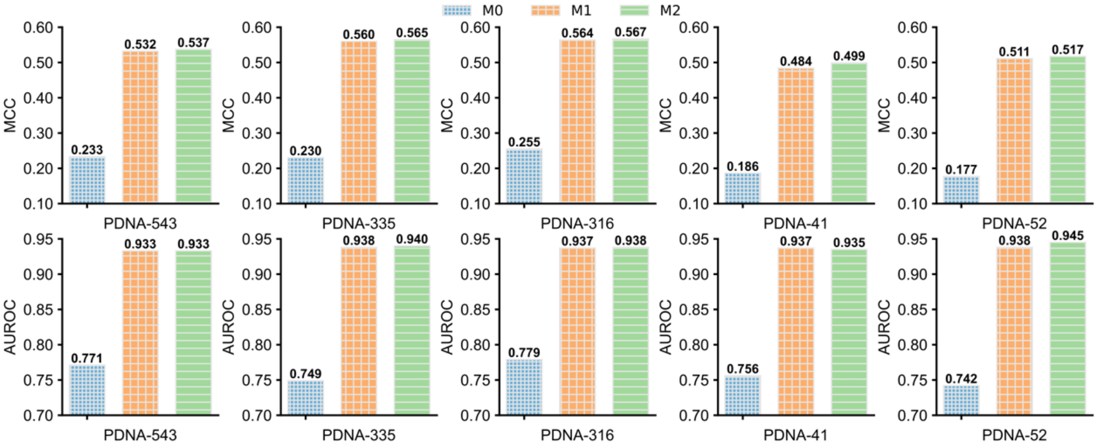

Figure 2. The MCC and AUROC values of three ablation models on five benchmark datasets.

### 3.4 Case study

To further examine the effects of different DNA-binding site prediction methods, we selected two proteins (2MXF_A and 3ZQL_A) from our test datasets as illustrations. For each protein, we used four in-house methods (denoted as LA-ESM2, LA- ProtTrans, LA-EMS-MSA-1b, and ULDNA) and a competing method (PredDBR [31]) to predict the corresponding DNA-binding sites. Four in-house methods use the same LSTM-attention network with different feature embeddings from ESM2, ProtTrans, ESM-MSA, and ESM2+ProtTrans+ESM-MSA, respectively, where "+" means the individual feature embeddings from different language models are concentrated as a combination embedding. Table 3 summarizes the modeling results of two proteins for five DNA-binding site prediction methods, where the corresponding visualization results are illustrated in Figure 3. In addition, the predicted and native DNA-binding sites of two proteins by five methods are listed in Table S4.

Several interesting observations can be made from the data. First, the protein language models are critical to improve DNA-binding site prediction. Specifically, each of four in-house methods with pre-trained protein language models shows the higher MCC values than the competing PredDBR without language models on two proteins.

Taking ULDNA as an example, it gains 52.7% and 23.5% improvements of MCC values on 2MXF and 3ZQL_A, respectively, in comparison with PredDBR.

Second, the combination of complementary protein language models can further increase the accuracy of ULDNA. In 2MXF_A, three in-house methods (i.e., LA-ESM2, LA-ProtTrans, and LA-ESM-MSA) with different language models hit 14 true positives in total, which is more than that by each individual method, indicating that three language models (i.e., ESM2, ProtTrans, and ESM-MSA) derive complementary information from different database sources. Meanwhile, the false positives predicted by one in-house method can be corrected by other two methods. For example, LA-ESM2 generates two false positives (10P and 11H), which are correctly predicted as non-DNA-binding sites by LA-ProtTrans and LA-ESM-MSA. As a result, by taking the combination of three language models, ULDNA gains the most-true positives without false positives among all methods. Sometimes, one in-house method can cover all true positives predicted by other methods. For example, for 3ZQL_A, all true positives of LA-ESM2 and LA-ProtTrans are covered by LA-ESM-MSA. Even in this case, the accuracy of final ULDNA is still improved by including the less accurate methods to reduce false positives.

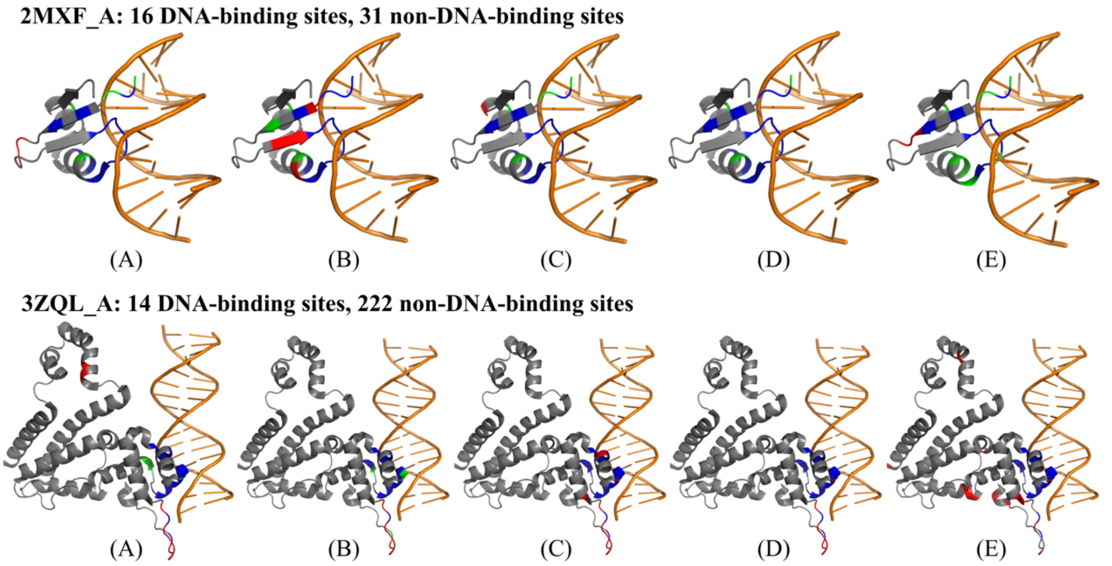
2MXF_A: 16 DNA-binding sites, 31 non-DNA-binding sites
(A)
(B)
(C)
(D)
(E)
3ZQL_A: 14 DNA-binding sites, 222 non-DNA-binding sites
(A)
(B)
(C)
(D)
(E)

Figure 3. Visualization of prediction results for two proteins using five DNA-binding site prediction models: (A) LA-ESM2, (B) LA-ProtTrans, (C) LA-ESM-MSA, (D) ULDNA, (E) PredDBR. The color scheme is used as follows: DNA in orange, true positives in blue, false positives in red, false negatives in green. The pictures are made with PyMOL [63].

Table 3. The modeling results of five DNA-binding site prediction methods on two representative examples

| Method | 2MXF_A | | | | | 3ZQL_A | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | TP | FP | TN | FN | MCC | TP | FP | TN | FN | MCC |
| LA-ESM2 | 12 | 2 | 29 | 4 | 0.710 | 12 | 9 | 213 | 2 | 0.678 |
| LA-ProtTrans | 12 | 4 | 27 | 4 | 0.621 | 11 | **8** | 214 | 3 | 0.651 |
| LA-ESM-MSA | 12 | 1 | 30 | 4 | 0.760 | **14** | 10 | 212 | **0** | 0.746 |
| ULDNA | **13** | **0** | **31** | **3** | **0.861** | **14** | **8** | **214** | **0** | **0.783** |
| PredDBR | 8 | 2 | 29 | 6 | 0.564 | **14** | 18 | 204 | **0** | 0.634 |

Bold fonts highlight the best performer in each evaluation metric.

## 4. Conclusions

We developed a new deep learning-based method, named ULDNA, to predict DNA-binding sites from the primary protein sequences. The algorithm was built on transformer embedding and LSTM-attention decoding. The large-scale tests on five protein-DNA binding site datasets demonstrated that ULDNA consistently outperforms other state-of-the-art approaches in the accuracy of DNA-binding site prediction. The improvement of ULDNA can be attributed to several advancements. First and most importantly, three transformers can effectively extract complementary evolution diversity-based feature embeddings for the input sequence from different database sources. Second, the designed LSTM-attention network enhances the correlation between evolution diversity and protein-DNA interaction pattern to improve prediction accuracy.

Despite the encouraging performance, there is still considerable room for further improvements. First, the serial feature concentration strategy, currently used in ULDNA, cannot perfectly deal with the redundant information among the feature embeddings from different transformers, where a more advanced feature fusion approaches may alleviate the negative impact caused by information redundancy in the future. Second, with the development of protein structure prediction models (e.g., AlphaFold2 [64] and ESMFold [42]), the predicted structures will have the huge potential to improve DNA-binding site prediction. Studies along these lines are under progress.

## Funding

This work is supported by the National Natural Science Foundation of China (62072243 and 61772273), the Natural Science Foundation of Jiangsu (BK20201304), and the Foundation of National Defense Key Laboratory of Science and Technology (JZX7Y202001SY000901).

## Reference

[1] G. D. Stormo, and Y. Zhao, Determining the specificity of protein–DNA interactions, *Nature Reviews Genetics,* 2010, 11 (11): 751-760.

[2] L. A. Gallagher, E. Velazquez, S. B. Peterson, J. C. Charity, M. C. Radey, M. J. Gebhardt, F. Hsu, L. M. Shull, K. J. Cutler, and K. Macareno, Genome-wide protein–DNA interaction site mapping in bacteria using a double-stranded DNA-specific cytosine deaminase, *Nature Microbiology,* 2022, 7 (6): 844-855.

[3] R. Esmaeeli, A. Bauzá, and A. Perez, Structural predictions of protein–DNA binding: MELD-DNA, *Nucleic Acids Research,* 2023, 51 (4): 1625-1636.

[4] Y. Mandel-Gutfreund, and H. Margalit, Quantitative parameters for amino acid-base interaction: implications for prediction of protein-DNA binding sites, *Nucleic acids research,* 1998, 26 (10): 2306-2312.

[5] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, CATH–a hierarchic classification of protein domain structures, *Structure,* 1997, 5 (8): 1093-1109.

[6] Y. Yu, S. Li, Z. Ser, H. Kuang, T. Than, D. Guan, X. Zhao, and D. J. Patel, Cryo-EM structure of DNA-bound Smc5/6 reveals DNA clamping enabled by multi-subunit conformational changes, *Proceedings of the National Academy of Sciences,* 2022, 119 (23): e2202799119.

[7] U. Consortium, UniProt: a worldwide hub of protein knowledge, *Nucleic acids research,* 2019, 47 (D1): D506-D515.

[8] Q. Yuan, S. Chen, J. Rao, S. Zheng, H. Zhao, and Y. Yang, AlphaFold2-aware protein–DNA binding site prediction using graph transformer, *Briefings in Bioinformatics,* 2022, 23 (2): bbab564.

[9] K. Qu, L. Wei, and Q. Zou, A review of DNA-binding proteins prediction methods, *Current Bioinformatics,* 2019, 14 (3): 246-254.

[10] Y.-H. Zhu, J. Hu, X.-N. Song, and D.-J. Yu, DNAPred: accurate identification of DNA-binding sites from protein sequence by ensembled hyperplane-distance-based support vector machines, *Journal of Chemical Information and Modeling,* 2019, 59 (6): 3057-3071.

[11] S. Jones, H. P. Shanahan, H. M. Berman, and J. M. Thornton, Using electrostatic potentials to predict DNA-binding sites on DNA-binding proteins, *Nucleic acids research,* 2003, 31 (24): 7189-7198.

[12] Y. Tsuchiya, K. Kinoshita, and H. Nakamura, Structure-based prediction of DNA-binding sites on proteins using the empirical preference of electrostatic potential and the shape of molecular surfaces, *PROTEINS: structure, Function, and Bioinformatics,* 2004, 55 (4): 885-894.

[13] J. Yang, A. Roy, and Y. Zhang, Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment, *Bioinformatics,* 2013, 29 (20): 2588-2595.

[14] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Research,* 1997, 25 (17): 3389-3402.

[15] T. Li, Q.-Z. Li, S. Liu, G.-L. Fan, Y.-C. Zuo, and Y. Peng, PreDNA: accurate

prediction of DNA-binding sites in proteins by integrating sequence and geometric structure information, *Bioinformatics,* 2013, 29 (6): 678-685.

[16] M. Gao, and J. Skolnick, DBD-Hunter: a knowledge-based method for the prediction of DNA–protein interactions, *Nucleic acids research,* 2008, 36 (12): 3978-3992.

[17] Y. Tsuchiya, K. Kinoshita, and H. Nakamura, PreDs: a server for predicting dsDNA-binding site on protein molecular surfaces, *Bioinformatics,* 2005, 21 (8): 1721-1723.

[18] M. Gao, and J. Skolnick, A threading-based method for the prediction of DNA-binding proteins with application to the human genome, *PLoS computational biology,* 2009, 5 (11): e1000567.

[19] Y. C. Chen, J. D. Wright, and C. Lim, DR_bind: a web server for predicting DNA-binding residues from the protein structure based on electrostatics, evolution and geometry, *Nucleic acids research,* 2012, 40 (W1): W249-W256.

[20] A. V. Morozov, J. J. Havranek, D. Baker, and E. D. Siggia, Protein–DNA binding specificity predictions with structural models, *Nucleic acids research,* 2005, 33 (18): 5781-5798.

[21] D. T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, *Journal of molecular biology,* 1999, 292 (2): 195-202.

[22] T. J. Richmond, Solvent accessible surface area and excluded volume in proteins: Analytical equations for overlapping spheres and implications for the hydrophobic effect, *Journal of molecular biology,* 1984, 178 (1): 63-89.

[23] W. S. Noble, What is a support vector machine?, *Nature biotechnology,* 2006, 24 (12): 1565-1567.

[24] G. Biau, and E. Scornet, A random forest guided tour, *Test,* 2016, 25 197-227.

[25] J. Hu, Y. Li, M. Zhang, X. Yang, H.-B. Shen, and D.-J. Yu, Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs, *IEEE/ACM transactions on computational biology and bioinformatics,* 2016, 14 (6): 1389-1398.

[26] J. Si, Z. Zhang, B. Lin, M. Schroeder, and B. Huang, MetaDBSite: a meta approach to improve protein DNA-binding sites prediction, *BMC systems biology,* 2011, 5 (1): 1-7.

[27] D.-J. Yu, J. Hu, J. Yang, H.-B. Shen, J. Tang, and J.-Y. Yang, Designing template-free predictor for targeting protein-ligand binding sites with classifier ensemble and spatial clustering, *IEEE/ACM transactions on computational biology and bioinformatics,* 2013, 10 (4): 994-1008.

[28] L. Wang, M. Q. Yang, and J. Y. Yang, Prediction of DNA-binding residues from protein sequence information using random forests, *Bmc Genomics,* 2009, 10 1-9.

[29] X. Ma, J. Guo, H.-D. Liu, J.-M. Xie, and X. Sun, Sequence-based prediction of DNA-binding residues in proteins with conservation and correlation information, *IEEE/ACM transactions on computational biology and bioinformatics,* 2012, 9 (06): 1766-1775.

[30] S. Guan, Q. Zou, H. Wu, and Y. Ding, Protein-dna binding residues prediction

using a deep learning model with hierarchical feature extraction, *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 2022,

[31] J. Hu, Y.-S. Bai, L.-L. Zheng, N.-X. Jia, D.-J. Yu, and G.-J. Zhang, Protein-dna binding residue prediction via bagging strategy and sequence-based cube-format feature, *IEEE/ACM transactions on computational biology and bioinformatics,* 2021, 19 (6): 3635-3645.

[32] B. P. Nguyen, Q. H. Nguyen, G.-N. Doan-Ngoc, T.-H. Nguyen-Vo, and S. Rahardja, iProDNA-CapsNet: identifying protein-DNA binding residues using capsule neural networks, *BMC bioinformatics,* 2019, 20 1-12.

[33] Y. Xia, C.-Q. Xia, X. Pan, and H.-B. Shen, GraphBind: protein structural context embedded rules learned by hierarchical graph neural networks for recognizing nucleic-acid-binding residues, *Nucleic acids research,* 2021, 49 (9): e51-e51.

[34] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, The protein data bank, *Nucleic Acids Research,* 2000, 28 (1): 235-242.

[35] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, P. Chen, J. Canny, P. Abbeel, and Y. Song, Evaluating protein transfer learning with TAPE, *Advances in neural information processing systems,* 2019, 32

[36] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost, Modeling aspects of the language of life through transfer-learning protein sequences, *BMC bioinformatics,* 2019, 20 (1): 1-17.

[37] T. Bepler, and B. Berger, Learning protein sequence embeddings using information from structure, *arXiv preprint arXiv:1902.08661,* 2019,

[38] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, and J. Cai, Recent advances in convolutional neural networks, *Pattern recognition,* 2018, 77 354-377.

[39] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, and J. Ma, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, *Proceedings of the National Academy of Sciences,* 2021, 118 (15): e2016239118.

[40] A. Villegas-Morcillo, S. Makrodimitris, R. C. van Ham, A. M. Gomez, V. Sanchez, and M. J. Reinders, Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function, *Bioinformatics,* 2021, 37 (2): 162-170.

[41] M. Littmann, M. Heinzinger, C. Dallago, K. Weissenow, and B. Rost, Protein embeddings and deep learning predict binding residues for various ligand classes, *Scientific Reports,* 2021, 11 (1): 23916.

[42] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, and Y. Shmueli, Evolutionary-scale prediction of atomic-level protein structure with a language model, *Science,* 2023, 379 (6637): 1123-1130.

[43] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, and M. Steinegger, Prottrans: Toward understanding the language of life through self-supervised learning, *IEEE transactions on pattern*

*analysis and machine intelligence,* 2021, 44 (10): 7112-7127.

[44] R. M. Rao, J. Liu, R. Verkuil, J. Meier, J. Canny, P. Abbeel, T. Sercu, and A. Rives, "MSA transformer." pp. 8844-8856.

[45] W. Li, and A. Godzik, Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, *Bioinformatics,* 2006, 22 (13): 1658-1659.

[46] G. Wang, and R. L. Dunbrack Jr, PISCES: a protein sequence culling server, *Bioinformatics,* 2003, 19 (12): 1589-1591.

[47] M. Mirdita, L. Von Den Driesch, C. Galiez, M. J. Martin, J. Söding, and M. Steinegger, Uniclust databases of clustered and deeply annotated protein sequences and alignments, *Nucleic acids research,* 2017, 45 (D1): D170-D176.

[48] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, UniRef: comprehensive and non-redundant UniProt reference clusters, *Bioinformatics,* 2007, 23 (10): 1282-1288.

[49] M. Steinegger, M. Mirdita, and J. Söding, Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold, *Nature methods,* 2019, 16 (7): 603-606.

[50] Z. Zhang, and M. R. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, *arXiv preprint arXiv:1805.07836,* 2018,

[51] D. P. Kingma, and J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980,* 2014,

[52] L. Wang, and S. J. Brown, BindN: a web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences, *Nucleic acids research,* 2006, 34 (suppl_2): W243-W248.

[53] W.-Y. Chu, Y.-F. Huang, C.-C. Huang, Y.-S. Cheng, C.-K. Huang, and Y.-J. Oyang, ProteDNA: a sequence-based predictor of sequence-specific DNA-binding residues in transcription factors, *Nucleic acids research,* 2009, 37 (suppl_2): W396-W401.

[54] L. Wang, C. Huang, M. Q. Yang, and J. Y. Yang, BindN+ for accurate prediction of DNA and RNA-binding residues from protein sequence features, *BMC Systems Biology,* 2010, 4 1-9.

[55] S. Hwang, Z. Gou, and I. B. Kuznetsov, DP-Bind: a web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins, *Bioinformatics,* 2007, 23 (5): 634-636.

[56] R. Liu, and J. Hu, DNABind: A hybrid algorithm for structure-based prediction of DNA-binding residues by combining machine learning-and template-based approaches, *PROTEINS: structure, Function, and Bioinformatics,* 2013, 81 (11): 1885-1899.

[57] Y. Ding, J. Tang, and F. Guo, Identification of protein–ligand binding sites by sequence information and ensemble classifier, *Journal of Chemical Information and Modeling,* 2017, 57 (12): 3149-3161.

[58] S. Ahmad, M. M. Gromiha, and A. Sarai, Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information, *Bioinformatics,* 2004, 20 (4): 477-486.

[59] Y. Ofran, V. Mysore, and B. Rost, Prediction of DNA-binding residues from sequence, *Bioinformatics,* 2007, 23 (13): i347-i353.

[60] P. Rodríguez, M. A. Bautista, J. Gonzalez, and S. Escalera, Beyond one-hot encoding: Lower dimensional target embedding, *Image and Vision Computing,* 2018, 75 21-31.

[61] S. Gold, and A. Rangarajan, Softmax to softassign: Neural network algorithms for combinatorial optimization, *Journal of Artificial Neural Networks,* 1996, 2 (4): 381-399.

[62] Z. Zhang, and M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, *Advances in neural information processing systems,* 2018, 31

[63] S. Yuan, H. S. Chan, and Z. Hu, Using PyMOL as a platform for computational drug design, *Wiley Interdisciplinary Reviews: Computational Molecular Science,* 2017, 7 (2): e1298.

[64] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, and A. Potapenko, Highly accurate protein structure prediction with AlphaFold, *Nature,* 2021, 596 (7873): 583-589.

# Supporting Texts

## Text S1. The procedures for ESM2 transformer

### A. Masking

For an input sequence, the masking strategy [1] 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 [2], 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 position information

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 the position of each token in the masked sequence to generate a position embedding matrix $H_P$:

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

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

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 a 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) and measures the evolution correlation for each amino acid pair in the sequence, $M_{i,j}^{Q}$, $M_{i,j}^{K}$, 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 $U_i$ :

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

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

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 $T_i$:

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

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

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

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:

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

where $H_{i+1}$ is the evolution diversity-based embedding matrix in $i$-th attention layer.

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 a negative log likelihood function between inputted one-hot and outputted probability matrices, to ensure that the prediction model correctly predicts the true amino acids in the masked position as much as possible:

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

where $x$ is a sequence in training protein set $X$, $x(M)$ is a set of masking position in $x$, $|x(M)|$ is the number of elements in $x(M)$, $c(l)$ is the type index of amino acid for the $l$-th token in $x$ before masking, and $-logP_{l,c(l)}$ is negative log likelihood of the true amino acid $x_l$ under condition of masking.

The ESM2 transformer is optimized by minimizing the loss function via Adam optimization algorithm [3]. Then, the output of last attention layer is represented as a $L \times D$ matrix, as the evolution diversity-based embedding for DNA-binding site prediction, where $D$ is the number of neurons of FFN. The current ESM2 model with 3 billion parameters was trained over 60 million proteins from UniRef50 database and can be freely download at https://github.com/facebookresearch/esm, where $n = 36$, $m = 20$, and $D = 2560$.

## Text S2. The procedures for ESM-MSA transformer

### A. 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.

### B. 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.

$$\boldsymbol{x}_{ij} = \left(x_{ij1}, x_{ij2}, \dots, x_{ijC_{max}}\right) \in R^{C_{max}}, x_{ijk} = \begin{cases} 1, & k = c_{ij} \\ 0, & k \neq c_{ij} \end{cases} \tag{S16}$$

$$\boldsymbol{y}_{ij} = \left(y_{ij1}, y_{ij2}, \dots, y_{ijM_{max}}\right) \in R^{M_{max}}, y_{ijk} = \begin{cases} 1, & k = i \\ 0, & k \neq i \end{cases} \tag{S17}$$

$$\boldsymbol{z}_{ij} = \left(z_{ij1}, z_{ij2}, \dots, z_{ijL_{max}}\right) \in R^{L_{max}}, z_{ijk} = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \tag{S18}$$

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. S16-S18, 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.

## C. Initial embedding

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

$$\boldsymbol{H}_{token} = \boldsymbol{X}\boldsymbol{W}_{token} = \begin{bmatrix} \boldsymbol{X}[1] \\ \boldsymbol{X}[2] \\ \dots \\ \boldsymbol{X}[M] \end{bmatrix} \boldsymbol{W}_{token} = \begin{bmatrix} \boldsymbol{X}[1]\boldsymbol{W}_{token} \\ \boldsymbol{X}[2]\boldsymbol{W}_{token} \\ \dots \\ \boldsymbol{X}[M]\boldsymbol{W}_{token} \end{bmatrix} \in R^{M\times L\times D} \tag{S19}$$

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

$$\boldsymbol{H}_{row} = \boldsymbol{X}\boldsymbol{W}_{row} = \begin{bmatrix} \boldsymbol{Y}[1] \\ \boldsymbol{Y}[2] \\ \dots \\ \boldsymbol{Y}[M] \end{bmatrix} \boldsymbol{W}_{row} = \begin{bmatrix} \boldsymbol{Y}[1]\boldsymbol{W}_{row} \\ \boldsymbol{Y}[2]\boldsymbol{W}_{row} \\ \dots \\ \boldsymbol{Y}[M]\boldsymbol{W}_{row} \end{bmatrix} \in R^{M\times L\times D} \tag{S20}$$

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

$$\boldsymbol{H}_{col} = \boldsymbol{Z}\boldsymbol{W}_{col} = \begin{bmatrix} \boldsymbol{Z}[1] \\ \boldsymbol{Z}[2] \\ \dots \\ \boldsymbol{Z}[M] \end{bmatrix} \boldsymbol{W}_{col} = \begin{bmatrix} \boldsymbol{Z}[1]\boldsymbol{W}_{col} \\ \boldsymbol{Z}[2]\boldsymbol{W}_{col} \\ \dots \\ \boldsymbol{Z}[M]\boldsymbol{W}_{col} \end{bmatrix} \in R^{M\times L\times D} \tag{S21}$$

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

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 $\boldsymbol{H}_{init}$:

$$\boldsymbol{H}_{init} = \boldsymbol{H}_{token} + \boldsymbol{H}_{row} + \boldsymbol{H}_{col}, \boldsymbol{H}_{init} \in R^{M\times L\times D} \tag{S22}$$

## D. Batch normalization and dropout

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

$$\boldsymbol{H}_1 = BN(\boldsymbol{H}_{init}) = \begin{bmatrix} BN(\boldsymbol{h}_{11}) & \cdots & BN(\boldsymbol{h}_{1L}) \\ \vdots & \ddots & \vdots \\ BN(\boldsymbol{h}_{M1}) & \cdots & BN(\boldsymbol{h}_{ML}) \end{bmatrix} \tag{S23}$$

$$BN(\boldsymbol{h}_{ij}) = \gamma \cdot \frac{\boldsymbol{h}_{ij} - u_{ij}}{\sqrt{\sigma_{ij}^2 + \epsilon}} + \beta, \boldsymbol{h}_{ij} \in R^D \tag{S24}$$

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 $\sigma_{ij}^2$ are mean and variance for $\boldsymbol{h}_{ij}$, respectively, and $\gamma$, $\beta$, and $\epsilon$ are normalized factors.

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

$$\boldsymbol{H}_1 \leftarrow dropout(\boldsymbol{H}_1, r) \tag{S25}$$

where $r$ is the rate of neurons which are randomly dropped in each training step, indicating that the corresponding weight vectors will be not optimized.

## E. Self-attention

The initial embedding matrix $\boldsymbol{H}_1$ is fed to the self-attention network with $N$ blocks, each of which consists of three sub-blocks. In this work, $N = 12$.

The first sub-block consists of a batch normalization layer, a row attention layer, a dropout layer, and a short connection, as follows.

$$\boldsymbol{H}_k^B = BN(\boldsymbol{H}_k) \tag{S26}$$

$$\boldsymbol{H}_k^R = RA(\boldsymbol{H}_k^B) \tag{S27}$$

$$\boldsymbol{H}_k^R \leftarrow dropout(\boldsymbol{H}_k^R, r) \tag{S28}$$

$$\boldsymbol{F}_k = SC(\boldsymbol{H}_k, \boldsymbol{H}_k^R) = \boldsymbol{H}_k + \boldsymbol{H}_k^R \tag{S29}$$

where $\boldsymbol{H}_k$ and $\boldsymbol{F}_k$ are the input and output matrices in the first sub-block of the $k$-th self-attention block, respectively, $BN(\cdot)$ is the batch normalization function (see Eqs. S23-S24), $SC(\cdot)$ is the short connection, and $RA(\cdot)$ is the row attention layer (see Eqs. S38-S45), $\boldsymbol{H}_k$, $\boldsymbol{H}_k^B$, $\boldsymbol{H}_k^R$, $\boldsymbol{F}_k \in R^{M\times L\times D}$.

The second sub-block consists of a batch normalization layer, a column attention layer, a dropout layer, and a short connection, as follows.

$$\boldsymbol{F}_k^B = BN(\boldsymbol{F}_k) \tag{S30}$$

$$\boldsymbol{F}_k^C = CA(\boldsymbol{F}_k^B) \tag{S31}$$

$$\boldsymbol{F}_k^C \leftarrow dropout(\boldsymbol{F}_k^C, r) \tag{S32}$$

$$\boldsymbol{U}_k = SC(\boldsymbol{F}_k, \boldsymbol{F}_k^C) = \boldsymbol{F}_k + \boldsymbol{F}_k^C \tag{S33}$$

where $\boldsymbol{F}_k$ and $\boldsymbol{U}_k$ are the input and output matrices in the second sub-block of the $k$-th self-attention block, respectively, $CA(\cdot)$ is the column attention layer (see Eqs. S46-S54), and $\boldsymbol{F}_k^B$, $\boldsymbol{F}_k^C$, $\boldsymbol{U}_k \in R^{M \times L \times D}$.

The last sub-block consists of a batch normalization layer, a feed-forward network, a dropout layer, and a short connection, as follows.

$$\boldsymbol{U}_k^B = BN(\boldsymbol{U}_k) \tag{S34}$$

$$\boldsymbol{U}_k^F = FFN(\boldsymbol{U}_k^B) \tag{S35}$$

$$\boldsymbol{U}_k^F \leftarrow dropout(\boldsymbol{U}_k^F, r) \tag{S36}$$

$$\boldsymbol{H}_{k+1} = SC(\boldsymbol{U}_k, \boldsymbol{U}_k^F) = \boldsymbol{U}_k + \boldsymbol{U}_k^F \tag{S37}$$

where $\boldsymbol{U}_k$ and $\boldsymbol{H}_{k+1}$ are the input and output matrices in the third sub-block of the $k$-th self-attention block, respectively, $FFN(.)$ is the feed-forward network (see Eqs. S55-S60), and $\boldsymbol{U}_k^B$, $\boldsymbol{U}_k^F$, $\boldsymbol{H}_{k+1} \in R^{M \times L \times D}$.

**(a) Row attention**

Each row attention layer consists of $m$ attention heads and a linear unit, where $m = 12$. In each attention head, the input matrix is multiplied by three weight matrices to generate the corresponding Query, Key, and Value matrices.

$$\boldsymbol{Q}_{kt}^R = \boldsymbol{H}_k^B \boldsymbol{W}_{kt}^{QR} = \begin{bmatrix} \boldsymbol{H}_k^B[1] \\ \boldsymbol{H}_k^B[2] \\ \cdots \\ \boldsymbol{H}_k^B[M] \end{bmatrix} \boldsymbol{W}_{kt}^{QR} = \begin{bmatrix} \boldsymbol{H}_k^B[1]\boldsymbol{W}_{kt}^{QR} \\ \boldsymbol{H}_k^B[2]\boldsymbol{W}_{kt}^{QR} \\ \cdots \\ \boldsymbol{H}_k^B[M]\boldsymbol{W}_{kt}^{QR} \end{bmatrix} \in R^{M \times L \times (\frac{D}{m})} \tag{S38}$$

$$\boldsymbol{K}_{kt}^R = \boldsymbol{H}_k^B \boldsymbol{W}_{kt}^{KR} = \begin{bmatrix} \boldsymbol{H}_k^B[1] \\ \boldsymbol{H}_k^B[2] \\ \cdots \\ \boldsymbol{H}_k^B[M] \end{bmatrix} \boldsymbol{W}_{kt}^{KR} = \begin{bmatrix} \boldsymbol{H}_k^B[1]\boldsymbol{W}_{kt}^{KR} \\ \boldsymbol{H}_k^B[2]\boldsymbol{W}_{kt}^{KR} \\ \cdots \\ \boldsymbol{H}_k^B[M]\boldsymbol{W}_{kt}^{KR} \end{bmatrix} \in R^{M \times L \times (\frac{D}{m})} \tag{S39}$$

$$\boldsymbol{V}_{kt}^R = \boldsymbol{H}_k^B \boldsymbol{W}_{kt}^{VR} = \begin{bmatrix} \boldsymbol{H}_k^B[1] \\ \boldsymbol{H}_k^B[2] \\ \cdots \\ \boldsymbol{H}_k^B[M] \end{bmatrix} \boldsymbol{W}_{kt}^{VR} = \begin{bmatrix} \boldsymbol{H}_k^B[1]\boldsymbol{W}_{kt}^{VR} \\ \boldsymbol{H}_k^B[2]\boldsymbol{W}_{kt}^{VR} \\ \cdots \\ \boldsymbol{H}_k^B[M]\boldsymbol{W}_{kt}^{VR} \end{bmatrix} \in R^{M \times L \times (\frac{D}{m})} \tag{S40}$$

$$\boldsymbol{H}_k^B[i] \in R^{L\times D}, \boldsymbol{W}_{kt}^{QR}, \boldsymbol{W}_{kt}^{KR}, \boldsymbol{W}_{kt}^{VR} \in R^{D\times(\frac{D}{m})}$$

where $\boldsymbol{H}_k^B$ is the input matrix of row attention layer in the $k$-th self-attention block (See Eq. S27), $\boldsymbol{Q}_{kt}^R$, $\boldsymbol{K}_{kt}^R$, and $\boldsymbol{V}_{kt}^R$ are Query, Key, and Value matrices in the *t*-th head of the row attention layer in the $k$-th block, respectively, $\boldsymbol{W}_{kt}^{QR}$, $\boldsymbol{W}_{kt}^{KR}$, and $\boldsymbol{W}_{kt}^{VR}$ are corresponding weight metrices.

Then, the dot-product between $\boldsymbol{Q}_{kt}^R$ and $\boldsymbol{K}_{kt}^R$ is performed and then normalized by SoftMax function to generate a row attention weight matrix:

$$\boldsymbol{W}_{kt}^{AR} = SoftMax\left(\frac{\sum_{i=1}^{M} \boldsymbol{Q}_{kt}^R[i]\cdot(\boldsymbol{K}_{kt}^R[i])^T\}}{\sqrt{MD/m}}\right) \in R^{L\times L},\ \boldsymbol{Q}_{kt}^R[i],\ \boldsymbol{K}_{kt}^R[i] \in R^{L\times(D/m)} \quad \text{(S41)}$$

$$\boldsymbol{W}_{kt}^{AR} \leftarrow dropout(\boldsymbol{W}_{kt}^{AR}, r) \quad \text{(S42)}$$

where $\boldsymbol{W}_{kt}^{AR}$ is the attention weight matrix in the *t*-th head of the row attention layer in the $k$-th block and measures the correlation for each pair of columns in the masked MSA.

Next, the row attention weight matrix $\boldsymbol{W}_{kt}^{AR}$ is multiplied by Value matrix $\boldsymbol{V}_{kt}^R$ to generate the corresponding row attention matrix:

$$\boldsymbol{A}_{kt}^R = \boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^R = \boldsymbol{W}_{kt}^{AR}\begin{bmatrix}\boldsymbol{V}_{kt}^R[1]\\ \boldsymbol{V}_{kt}^R[2]\\ \cdots\\ \boldsymbol{V}_{kt}^R[M]\end{bmatrix} = \begin{bmatrix}\boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^R[1]\\ \boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^R[2]\\ \cdots\\ \boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^R[M]\end{bmatrix} \in R^{M\times L\times(\frac{D}{m})}, \boldsymbol{V}_{kt}^R[i] \in R^{L\times(\frac{D}{m})} \quad \text{(S43)}$$

where $\boldsymbol{A}_{kt}^R$ is the attention matrix in the *t*-th head of the row attention layer in the $k$-th block.

Finally, the outputs of all attention heads are concatenated as a new matrix, which is further fed to a linear unit:

$$\boldsymbol{A}_k^R = \boldsymbol{A}_{k1}^R\boldsymbol{A}_{k2}^R \ldots \boldsymbol{A}_{km}^R \in R^{M\times L\times D} \quad \text{(S44)}$$

$$\boldsymbol{H}_k^R = \boldsymbol{A}_k^R\boldsymbol{W}_k^R + \boldsymbol{b}_k^R = \begin{bmatrix}\boldsymbol{A}_k^R[1]\\ \boldsymbol{A}_k^R[2]\\ \cdots\\ \boldsymbol{A}_k^R[M]\end{bmatrix}\boldsymbol{W}_k^R + \boldsymbol{b}_k^R = \begin{bmatrix}\boldsymbol{A}_k^R[1]\boldsymbol{W}_k^R\\ \boldsymbol{A}_k^R[2]\boldsymbol{W}_k^R\\ \cdots\\ \boldsymbol{A}_k^R[M]\boldsymbol{W}_k^R\end{bmatrix} + \boldsymbol{b}_k^R \in R^{M\times L\times D} \quad \text{(S45)}$$

$$\boldsymbol{W}_k^R \in R^{D\times D}, \boldsymbol{A}_k^R[i] \in R^{L\times D}$$

where $\boldsymbol{H}_k^R$ in the output matrix of row attention layer in the $k$-th attention block (See Eq. S27), and $\boldsymbol{W}_k^R$ and $\boldsymbol{b}_k^R$ are weight matrix and bias in the linear unit, respectively.

**(b) Column attention**

Each column attention layer consists of $m$ attention heads and a linear unit. In each attention head, the input matrix is multiplied by three weight matrices to generate the corresponding Query, Key, and Value matrices.

$$\boldsymbol{Q}_{kt}^{C} = \boldsymbol{F}_{k}^{B}\boldsymbol{W}_{kt}^{QC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1] \\ \boldsymbol{F}_{k}^{B}[2] \\ \cdots \\ \boldsymbol{F}_{k}^{B}[M] \end{bmatrix} \boldsymbol{W}_{kt}^{QC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1]\boldsymbol{W}_{kt}^{QC} \\ \boldsymbol{F}_{k}^{B}[2]\boldsymbol{W}_{kt}^{QC} \\ \cdots \\ \boldsymbol{F}_{k}^{B}[M]\boldsymbol{W}_{kt}^{QC} \end{bmatrix} \in R^{M\times L\times(\frac{D}{m})} \tag{S46}$$

$$\boldsymbol{K}_{kt}^{C} = \boldsymbol{F}_{k}^{B}\boldsymbol{W}_{kt}^{KC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1] \\ \boldsymbol{F}_{k}^{B}[2] \\ \cdots \\ \boldsymbol{F}_{k}^{B}[M] \end{bmatrix} \boldsymbol{W}_{kt}^{KC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1]\boldsymbol{W}_{kt}^{KC} \\ \boldsymbol{F}_{k}^{B}[2]\boldsymbol{W}_{kt}^{KC} \\ \cdots \\ \boldsymbol{F}_{k}^{B}[M]\boldsymbol{W}_{kt}^{KC} \end{bmatrix} \in R^{M\times L\times(\frac{D}{m})} \tag{S47}$$

$$\boldsymbol{V}_{kt}^{C} = \boldsymbol{F}_{k}^{B}\boldsymbol{W}_{kt}^{VC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1] \\ \boldsymbol{F}_{k}^{B}[2] \\ \cdots \\ \boldsymbol{F}_{k}^{B}[M] \end{bmatrix} \boldsymbol{W}_{kt}^{VC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1]\boldsymbol{W}_{kt}^{VC} \\ \boldsymbol{F}_{k}^{B}[2]\boldsymbol{W}_{kt}^{VC} \\ \cdots \\ \boldsymbol{F}_{k}^{B}[M]\boldsymbol{W}_{kt}^{VC} \end{bmatrix} \in R^{M\times L\times(\frac{D}{m})} \tag{S48}$$

$$\boldsymbol{F}_{k}^{B}[i] \in R^{L\times D}, \boldsymbol{W}_{kt}^{QC}, \boldsymbol{W}_{kt}^{KC}, \boldsymbol{W}_{kt}^{VC} \in R^{D\times(\frac{D}{m})}$$

where $\boldsymbol{F}_{k}^{B}$ is the input matrix of column attention layer in the $k$-th self-attention block (see Eq. S31), $\boldsymbol{Q}_{kt}^{C}$, $\boldsymbol{K}_{kt}^{C}$, and $\boldsymbol{V}_{kt}^{C}$ are Query, Key, and Value matrices in the $t$-th head of column attention layer in the $k$-th block, respectively, $\boldsymbol{W}_{kt}^{QC}$, $\boldsymbol{W}_{kt}^{KC}$, and $\boldsymbol{W}_{kt}^{VC}$ are corresponding weight metrices.

Then, the dot-product between $\boldsymbol{Q}_{kt}^{C}$ and $\boldsymbol{K}_{kt}^{C}$ is performed and then normalized by SoftMax function to generate an attention weight matrix:

$$\boldsymbol{W}_{kt}^{AC} = SoftMax\left(\frac{\boldsymbol{Q}_{kt}^{C}(\boldsymbol{K}_{kt}^{C})^{T}}{\sqrt{D/m}}\right) \in R^{M\times L\times M} \tag{S49}$$

$$\boldsymbol{W}_{kt}^{AC} \leftarrow dropout(\boldsymbol{W}_{kt}^{AC}, r) \tag{S50}$$

$$\boldsymbol{Q}_{kt}^{C}(\boldsymbol{K}_{kt}^{C})^{T} = \left[\boldsymbol{Q}_{kt}^{C}[:,1,:]\ \boldsymbol{Q}_{kt}^{C}[:,2,:]\ \ldots \boldsymbol{Q}_{kt}^{C}[:,L,:]\right] \cdot \left[\boldsymbol{K}_{kt}^{C}[:,1,:]\ \boldsymbol{K}_{kt}^{C}[:,2,:]\ \ldots \boldsymbol{K}_{kt}^{C}[:,L,:]\right]^{T} =$$

$$[\boldsymbol{Q}_{kt}^{C}[:,1,:] \cdot \boldsymbol{K}_{kt}^{C}[:,1,:]^{T}\ \boldsymbol{Q}_{kt}^{C}[:,2,:] \cdot \boldsymbol{K}_{kt}^{C}[:,2,:]^{T} \ldots \boldsymbol{Q}_{kt}^{C}[:,L,:] \cdot \boldsymbol{K}_{kt}^{C}[:,L,:]^{T}] \in R^{M\times L\times M} \tag{S51}$$

$$\boldsymbol{Q}_{kt}^{C}[:,j,:], \boldsymbol{K}_{kt}^{C}[:,j,:] \in R^{M\times\left(\frac{D}{m}\right)}, \boldsymbol{Q}_{kt}^{C}[:,j,:] \cdot \boldsymbol{K}_{kt}^{C}[:,j,:]^{T} \in R^{M\times M}$$

where $\boldsymbol{W}_{kt}^{AC}$ is the attention weight matrix in the $t$-th head of column attention layer in the $k$-th block, and $\boldsymbol{W}_{kt}^{AC}[:,j,:]$ measures the correlation for each pair of alignments at the $j$-th position.

Next, the column attention weight matrix $\boldsymbol{W}_{kt}^{AC}$ is multiplied by Value matrix $\boldsymbol{V}_{kt}^{C}$ to generate the corresponding column attention matrix:

$$\boldsymbol{A}_{kt}^{C} = \boldsymbol{W}_{kt}^{AC}\boldsymbol{V}_{kt}^{C} = \left[\boldsymbol{W}_{kt}^{AC}[:,1,:]\ \boldsymbol{W}_{kt}^{AC}[:,2,:]\ \ldots \boldsymbol{W}_{kt}^{AC}[:,L,:]\right] \cdot \left[\boldsymbol{V}_{kt}^{C}[:,1,:]\ \boldsymbol{V}_{kt}^{C}[:,2,:]\ \ldots \boldsymbol{V}_{kt}^{C}[:,L,:]\right] = \Big[\boldsymbol{W}_{kt}^{AC}[:,1,:] \cdot$$

$$\boldsymbol{V}_{kt}^{C}[:,1,:]\ \ \boldsymbol{W}_{kt}^{AC}[:,2,:] \cdot \boldsymbol{V}_{kt}^{C}[:,2,:]\ \ \ldots \boldsymbol{W}_{kt}^{AC}[:,L,:] \cdot \boldsymbol{V}_{kt}^{C}[:,L,:]\Big] \in R^{M\times L\times(\frac{D}{m})} \tag{S52}$$

$$\boldsymbol{W}_{kt}^{AC}[:,j,:] \in R^{M\times M}, \boldsymbol{V}_{kt}^{C}[:,j,:] \in R^{M\times(\frac{D}{m})}, \boldsymbol{W}_{kt}^{AC}[:,j,:]\cdot \boldsymbol{V}_{kt}^{C}[:,j,:] \in R^{M\times(\frac{D}{m})}$$

where $\boldsymbol{A}_{kt}^{C}$ is the attention matrix in the *t*-th head of column attention layer in the $k$-th block.

Finally, the outputs of all attention heads are concatenated as a new matrix, which is further fed to a linear unit:

$$\boldsymbol{A}_k^C = \boldsymbol{A}_{k1}^C \boldsymbol{A}_{k2}^C \dots \boldsymbol{A}_{km}^C \in R^{M\times L\times D} \tag{S53}$$

$$\boldsymbol{F}_k^C = \boldsymbol{A}_k^C \boldsymbol{W}_k^C + \boldsymbol{b}_k^C = \begin{bmatrix} \boldsymbol{A}_k^C[1] \\ \boldsymbol{A}_k^C[2] \\ \dots \\ \boldsymbol{A}_k^C[M] \end{bmatrix} \boldsymbol{W}_k^C = \begin{bmatrix} \boldsymbol{A}_1^C[1]\boldsymbol{W}_k^C \\ \boldsymbol{A}_2^C[2]\boldsymbol{W}_k^C \\ \dots \\ \boldsymbol{A}_k^C[M]\boldsymbol{W}_k^C \end{bmatrix} + \boldsymbol{b}_k^C \in R^{M\times L\times D} \tag{S54}$$

$$\boldsymbol{W}_k^C \in R^{D\times D}, \boldsymbol{A}_k^C[i] \in R^{L\times D}$$

where $\boldsymbol{F}_k^C$ in the output matrix of column attention layer in the $k$-th attention block, (See Eq. S31), and $\boldsymbol{W}_k^C$ and $\boldsymbol{b}_k^C$ are weight matrix and bias in the linear unit, respectively.

**(c) Feed-forward network**

$$\boldsymbol{T}_k^F = gelu(\boldsymbol{U}_k^B \boldsymbol{W}_k^1 + \boldsymbol{b}_k^1) \in R^{M\times L\times D_1} \tag{S55}$$

$$\boldsymbol{T}_k^F \leftarrow dropout(\boldsymbol{T}_k^F, r) \tag{S56}$$

$$\boldsymbol{U}_k^F = \boldsymbol{T}_k^F \boldsymbol{W}_k^2 + \boldsymbol{b}_k^2 \in R^{M\times L\times D} \tag{S57}$$

$$gelu(x) = x\emptyset(x) \tag{S58}$$

$$\boldsymbol{U}_k^B \boldsymbol{W}_k^1 = \begin{bmatrix} \boldsymbol{U}_k^B[1] \\ \boldsymbol{U}_k^B[2] \\ \dots \\ \boldsymbol{U}_k^B[M] \end{bmatrix} \boldsymbol{W}_k^1 = \begin{bmatrix} \boldsymbol{U}_k^B[1]\boldsymbol{W}_k^1 \\ \boldsymbol{U}_k^B[2]\boldsymbol{W}_k^1 \\ \dots \\ \boldsymbol{U}_k^B[M]\boldsymbol{W}_k^1 \end{bmatrix} \in R^{M\times L\times D_1} \tag{S59}$$

$$\boldsymbol{T}_k^F \boldsymbol{W}_k^2 = \begin{bmatrix} \boldsymbol{T}_k^F[1] \\ \boldsymbol{T}_k^F[2] \\ \dots \\ \boldsymbol{T}_k^F[M] \end{bmatrix} \boldsymbol{W}_k^2 = \begin{bmatrix} \boldsymbol{T}_k^F[1]\boldsymbol{W}_k^2 \\ \boldsymbol{T}_k^F[2]\boldsymbol{W}_k^2 \\ \dots \\ \boldsymbol{T}_k^F[M]\boldsymbol{W}_k^2 \end{bmatrix} \in R^{M\times L\times D} \tag{S60}$$

$$\boldsymbol{U}_k^B[i] \in R^{L\times D}, \boldsymbol{W}_k^1 \in R^{D\times D_1}, \boldsymbol{T}_k^F[i] \in R^{L\times D_1}, \boldsymbol{W}_k^2 \in R^{D_1\times D}, D_1\text{=3072}$$

where $\boldsymbol{U}_k^B$ and $\boldsymbol{U}_k^F$ are the input and output matrices of feed-forward network in the $k$-th self-attention block, respectively, (see Eq. S35), $\boldsymbol{W}_k^1$ and $\boldsymbol{W}_k^2$ are weight matrices, $\boldsymbol{b}_k^1$ and $\boldsymbol{b}_k^2$ are bias, and $\emptyset(x)$ is the integral of Gaussian Distribution for $x$.

## F. Output layer

The output of the last self-attention block is fed to a fully connected layer with SoftMax function to generate a probability matrix:

$$\boldsymbol{P} = SoftMax(\boldsymbol{H}_{N+1}\boldsymbol{W}^{O} + \boldsymbol{b}^{O}) \in R^{M\times L\times C_{max}} \quad \text{(S61)}$$

$$\boldsymbol{H}_{N+1}\boldsymbol{W}^{O} = \begin{bmatrix} \boldsymbol{H}_{N+1}[1]\boldsymbol{W}^{O} \\ \boldsymbol{H}_{N+1}[2]\boldsymbol{W}^{O} \\ \cdots \\ \boldsymbol{H}_{N+1}[M]\boldsymbol{W}^{O} \end{bmatrix}, \boldsymbol{H}_{N+1}[i] \in R^{L\times D}, \boldsymbol{W}^{O} \in R^{D\times C_{max}} \quad \text{(S62)}$$

where $\boldsymbol{H}_{N+1}$ is the outputted embedding matrix in the $N$-th self-attention block, $\boldsymbol{W}^{O}$ and $\boldsymbol{b}^{O}$ are weight matrix and bias, respectively, and the $\boldsymbol{P}(i,j,c)$ indicates the probability that the $j$-th position of the $i$-th sequence in the masked MSA is predicted as the $c$-th type of amino acid.

## G. Loss function

For an individual MSA, the loss function is designed as:

$$Loss_{msa} = \frac{1}{M} \cdot \sum_{i=1}^{M} \{\frac{1}{|mask(i)|} \cdot \sum_{j\in mask(i)} -log\boldsymbol{P}_{i,j,c(i,j)}\} \quad \text{(S63)}$$

where $M$ is the number of alignments, $mask(i)$ is a set of masking position in the $i$-th sequence, $|mask(i)|$ is the number of elements in $mask(i)$, $c(i,j)$ is the type index of amino acid for the $j$-th position in the $i$-th sequence before masking, and -$log\boldsymbol{P}_{i,j,c(i,j)}$ is negative log likelihood of the true amino acid at the $j$-th position in the $i$-th sequence under condition of masking.

# Supporting Figures

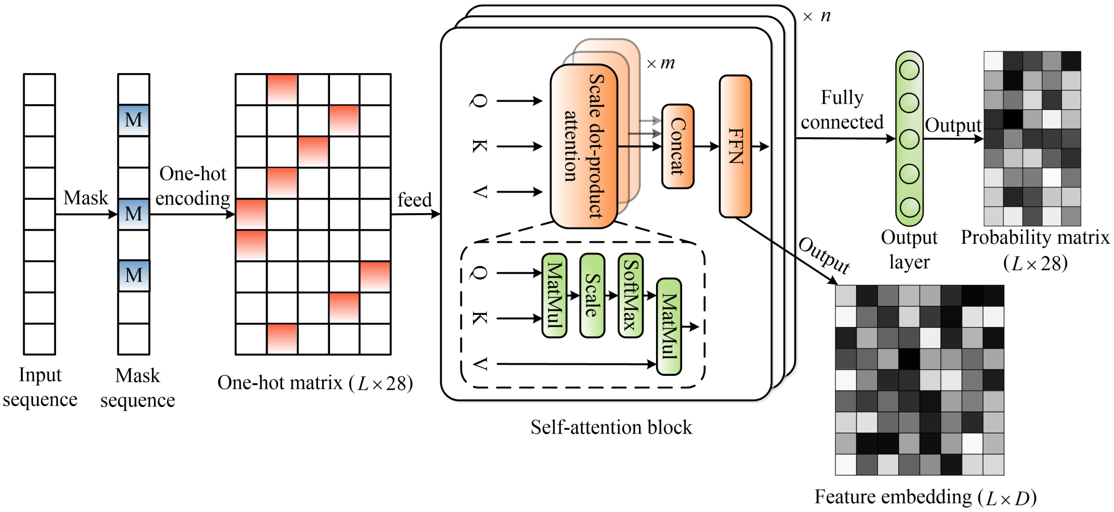

Figure S1. The workflow of ESM2 transformer.

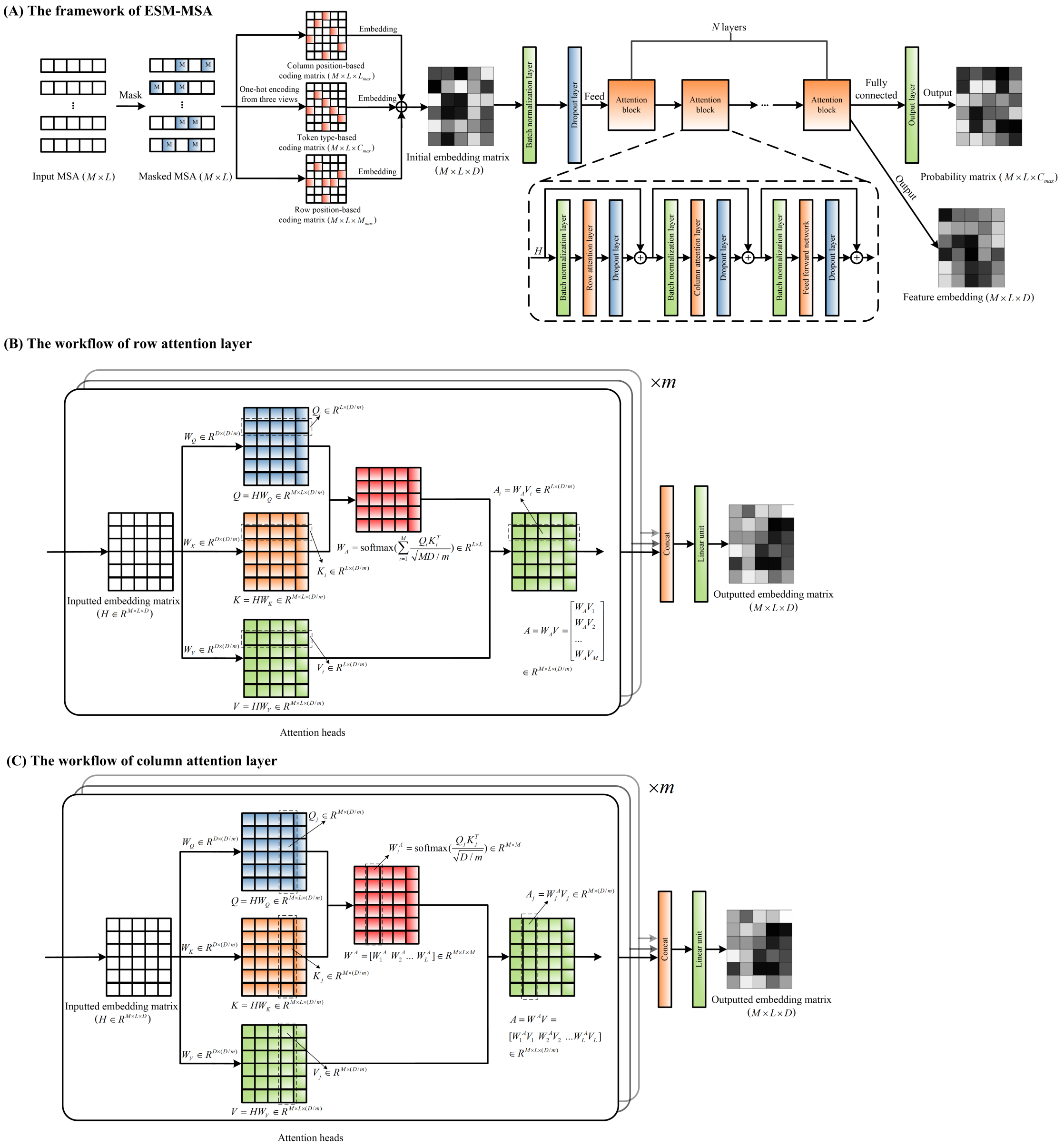
(A) The framework of ESM-MSA
Input MSA ($M \times L$)
Mask
M
Masked MSA ($M \times L$)
One-hot encoding from three views
Column position-based coding matrix ($M \times L \times L_{max}$)
Token type-based coding matrix ($M \times L \times C_{max}$)
Row position-based coding matrix ($M \times L \times M_{max}$)
Embedding
Embedding
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
(B) The workflow of row attention layer
$\times m$
$W_Q \in R^{D\times(D/m)}$
$Q_i \in R^{L\times(D/m)}$
$Q = HW_Q \in R^{M\times L\times(D/m)}$
$W_K \in R^{D\times(D/m)}$
$K_i \in R^{L\times(D/m)}$
$K = HW_K \in R^{M\times L\times(D/m)}$
$W_V \in R^{D\times(D/m)}$
$V_i \in R^{L\times(D/m)}$
$V = HW_V \in R^{M\times L\times(D/m)}$
$W_A = \text{softmax}(\sum_{i=1}^{M} \frac{Q_i K_i^T}{\sqrt{MD/m}}) \in R^{L\times L}$
$A_i = W_A V_i \in R^{L\times(D/m)}$
$A = W_A V = \begin{bmatrix} W_A V_1 \\ W_A V_2 \\ \dots \\ W_A V_M \end{bmatrix} \in R^{M\times L\times(D/m)}$
Inputted embedding matrix ($H \in R^{M\times L\times D}$)
Concat
Linear unit
Outputted embedding matrix ($M \times L \times D$)
Attention heads
(C) The workflow of column attention layer
$\times m$
$W_Q \in R^{D\times(D/m)}$
$Q_j \in R^{M\times(D/m)}$
$Q = HW_Q \in R^{M\times L\times(D/m)}$
$W_K \in R^{D\times(D/m)}$
$K_j \in R^{M\times(D/m)}$
$K = HW_K \in R^{M\times L\times(D/m)}$
$W_V \in R^{D\times(D/m)}$
$V_j \in R^{M\times(D/m)}$
$V = HW_V \in R^{M\times L\times(D/m)}$
$W_j^A = \text{softmax}(\frac{Q_j K_j^T}{\sqrt{D/m}}) \in R^{M\times M}$
$W^A = [W_1^A \; W_2^A \dots W_L^A] \in R^{M\times L\times M}$
$A_j = W_j^A V_j \in R^{M\times(D/m)}$
$A = W^A V = [W_1^A V_1 \; W_2^A V_2 \dots W_L^A V_L] \in R^{M\times L\times(D/m)}$
Inputted embedding matrix ($H \in R^{M\times L\times D}$)
Concat
Linear unit
Outputted embedding matrix ($M \times L \times D$)
Attention heads

Figure S2. The workflow of ESM-MSA transformer.

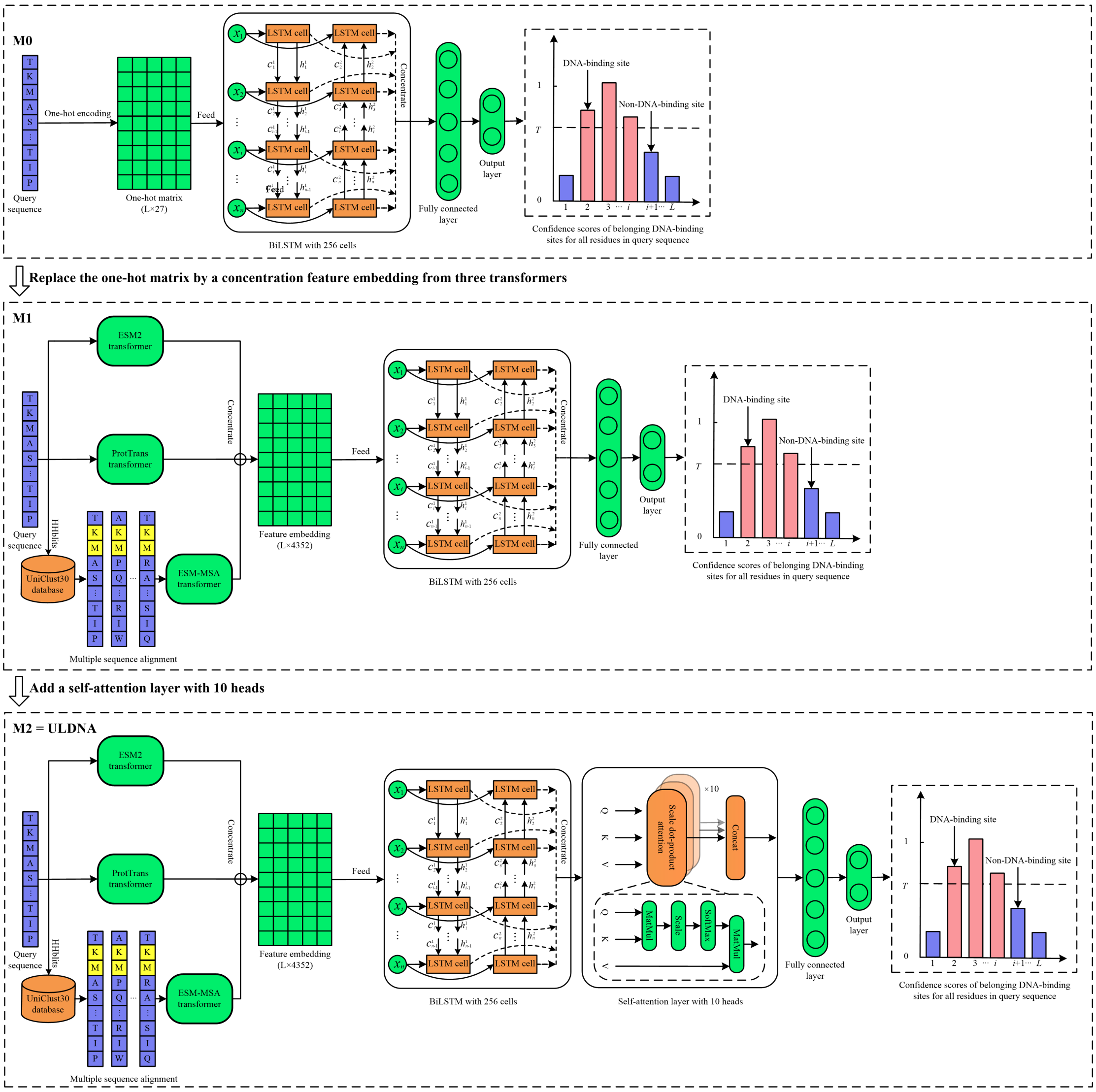
M0
Query sequence
One-hot encoding
One-hot matrix (L×27)
Feed
LSTM cell
Concentrate
BiLSTM with 256 cells
Fully connected layer
Output layer
DNA-binding site
Non-DNA-binding site
1
T
0
Confidence scores of belonging DNA-binding sites for all residues in query sequence
Replace the one-hot matrix by a concentration feature embedding from three transformers
M1
ESM2 transformer
ProtTrans transformer
ESM-MSA transformer
HHblits
UniClust30 database
Multiple sequence alignment
Feature embedding (L×4352)
Add a self-attention layer with 10 heads
M2 = ULDNA
Scale dot-product attention
×10
Concat
MatMul
Scale
SoftMax
Q
K
V
Self-attention layer with 10 heads

Figure S3. The architectures of three ablation models.

# Supporting Tables

Table S1. The performance of five DNA-binding site predictors on PDNA-543 over ten-fold cross-validation.

| Method | Sen | Spe | Acc | MCC | AUROC |
|---|---|---|---|---|---|
| TargetDNA ($Sen \approx Spe$) [a] | 0.770 | 0.771 | 0.770 | 0.304 | 0.845 |
| DNAPred ($Sen \approx Spe$) [b] | 0.771 | 0.785 | 0.784 | 0.318 | 0.861 |
| PredDBR ($Sen \approx Spe$) [c] | 0.776 | 0.774 | 0.774 | 0.358 | - |
| ULDNA ($Sen \approx Spe$) | **0.864** | **0.861** | **0.861** | **0.462** | **0.933** |
| TargetDNA ($Spe \approx 0.95$) [a] | 0.406 | 0.950 | 0.914 | 0.339 | 0.845 |
| DNAPred ($Spe \approx 0.95$) [b] | 0.449 | 0.950 | 0.917 | 0.373 | 0.861 |
| PredDBR ($Spe \approx 0.95$) [c] | 0.454 | **0.955** | 0.914 | 0.415 | - |
| Guan's method ($Spe \approx 0.95$) [d] | 0.452 | 0.954 | 0.928 | 0.352 | - |
| ULDNA ($Spe \approx 0.95$) | **0.668** | 0.950 | **0.931** | **0.534** | **0.933** |

[a] Results excerpted from TargetDNA [4]; [b] Results excerpted from DNAPred [5]; [c] Results excerpted from PredDBR [6]; [d] Results excerpted from Guan et al [7]. "$Sen \approx Spe$" means the threshold that makes $Sen \approx Spe$; "$Spe \approx 0.95$" means the threshold that makes $Spe \approx 0.95$. '-' means the value is not given. Bold fonts highlight the best performer in each evaluation metric.

Table S2. The performance of five DNA-binding site predictors on PDNA-335 over five-fold cross-validation.

| Method | Sen | Spe | Acc | MCC | AUROC |
|---|---|---|---|---|---|
| EC-RUS [a] | 0.487 | 0.951 | **0.926** | 0.378 | 0.852 |
| TargetS [b] | 0.417 | 0.945 | 0.899 | 0.362 | 0.824 |
| DNAPred [c] | 0.543 | 0.917 | 0.886 | 0.390 | 0.856 |
| PredDBR [d] | 0.426 | **0.953** | 0.910 | 0.390 | - |
| ULDNA | **0.676** | 0.948 | 0.925 | **0.565** | **0.940** |

[a] Results excerpted from EC-RUS [8]; [b] Results excerpted from TargetS [9]; [c] Results excerpted from DNAPred [5]; [d] Results excerpted from PredDBR [6]. '-' means the value is not given. Bold fonts highlight the best performer in each evaluation metric.

Table S3. The performance of eleven DNA-binding site predictors on PDNA-316 over ten-fold cross-validation.

| Method | Sen | Spe | Acc | MCC |
|---|---|---|---|---|
| DBS-PRED [a] | 0.530 | 0.760 | 0.750 | 0.170 |
| BindN [a] | 0.540 | 0.800 | 0.780 | 0.210 |
| DNABindR [a] | 0.660 | 0.740 | 0.730 | 0.230 |
| DISIS [a] | 0.190 | **0.980** | 0.920 | 0.250 |
| DP-Bind [a] | 0.690 | 0.790 | 0.780 | 0.290 |
| BindN-rf [a] | 0.670 | 0.830 | 0.820 | 0.320 |
| MetaDBSite [a] | 0.770 | 0.770 | 0.770 | 0.320 |
| TargetDNA ($Sen \approx Spe$) [a] | 0.780 | 0.780 | 0.780 | 0.339 |
| TargetDNA ($Spe \approx 0.95$) [a] | 0.430 | 0.950 | 0.910 | 0.375 |
| DNAPred ($Sen \approx Spe$) [b] | 0.800 | 0.799 | 0.799 | 0.370 |
| DNAPred ($Spe \approx 0.95$) [b] | 0.521 | 0.951 | 0.918 | 0.452 |
| PredDBR ($Sen \approx Spe$) [c] | 0.815 | 0.807 | 0.808 | 0.398 |
| PredDBR ($Spe \approx 0.95$) [c] | 0.561 | 0.953 | 0.921 | 0.497 |
| PredDBR (threshold = 0.5) [c] | 0.531 | 0.958 | 0.923 | 0.489 |
| ULDNA ($Sen \approx Spe$) | **0.871** | 0.867 | 0.867 | 0.502 |
| ULDNA ($Spe \approx 0.95$) | 0.676 | 0.950 | 0.929 | 0.561 |
| ULDNA (threshold = 0.5) | 0.449 | 0.983 | **0.942** | 0.526 |

[a] Results excerpted from TargetDNA [4]; [b] Results excerpted from DNAPred [5]; [c] Results excerpted from PredDBR [6]; "$Sen \approx Spe$" means the threshold that makes $Sen \approx Spe$; "$Spe \approx 0.95$" means the threshold that makes $Spe \approx 0.95$. Bold fonts highlight the best performer in each evaluation metric.

Table S4. The predicted and native DNA-binding sites of two representative proteins for five DNA-binding prediction methods

| Protein | Method | Predicted DNA-binding sites |
|---|---|---|
| 2MXF_A | LA-ESM2 | **2R 5K 7Y** 10P 11H **18T 19K 20G 21G 22N 23H 24K 27K 30K** |
| | LA-ProtTrans | **1A 2R 3K** 4V **5K** 16I 17E **18T 19K 20G 21G 22N 23H 24K** 25T **27K** |
| | LA-ESM-MSA | **2R 5K 7Y 18T 19K 20G 21G 22N 23H 24K 27K 30K** 41W |
| | ULDNA | **2R 3K 5K 7Y 18T 19K 20G 21G 22N 23H 24K 27K 30K** |
| | PredDBR | **2R 5K 7Y** 8K 9N **18T 19K 20G 21G 23H** |
| | Native DNA-binding sites | 1A 2R 3K 5K 7Y 18T 19K 20G 21G 22N 23H 24K 26L 27K 30K 39E |
| 3ZQL_A | LA-ESM2 | **12R** 13R 14S 15A 16R 17S 18H **19R** 20T **43S 44M 45R 54G 55T 56M 57S 59Y 60Y 61Y** 180R 183 M |
| | LA-ProtTrans | 13R 14S 15A 16R 17S 18H **19R** 20T 21L **43S 44M 45R 55T 56M 57S 59Y 60Y 61Y 65K** |
| | LA-ESM-MSA | **12R** 13R 14S 15A 16R 17S 18H **19R** 20T 23R **43S 44M 45R** 46R 53A **54G 55T 56M 57S 59Y 60Y 61Y 64T 65K** |
| | ULDNA | **12R** 13R 14S 15A 16R 17S 18H **19R** 20T **43S 44M 45R** 53A **54G 55T 56M 57S 59Y 60Y 61Y 64T 65K** |
| | PredDBR | 1V 4W 6H 7P **12R** 15A 18H **19R** 22S 23R **43S 44M 45R** 53A **54G 55T 56M 57S 59Y 60Y 61Y 64T 65K** 115W 117N 119H 124P 125N 126S 182W 187G 236D |
| | Native DNA-binding sites | 12R 19R 43S 44M 45R 54G 55T 56M 57S 59Y 60Y 61Y 64T 65K |

Bold font means a DNA-binding site can be correctly predicted by a DNA-binding prediction method.

## Reference

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[2] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[3] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[4] J. Hu, Y. Li, M. Zhang, X. Yang, H.-B. Shen, and D.-J. Yu, "Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 14, no. 6, pp. 1389-1398, 2016.

[5] Y.-H. Zhu, J. Hu, X.-N. Song, and D.-J. Yu, "DNAPred: accurate identification of DNA-binding sites from protein sequence by ensembled hyperplane-distance-based support vector machines," *Journal of chemical information and modeling,* vol. 59, no. 6, pp. 3057-3071, 2019.

[6] J. Hu, Y.-S. Bai, L.-L. Zheng, N.-X. Jia, D.-J. Yu, and G.-J. Zhang, "Protein-dna binding residue prediction via bagging strategy and sequence-based cube-format feature," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 19, no. 6, pp. 3635-3645, 2021.

[7] S. Guan, Q. Zou, H. Wu, and Y. Ding, "Protein-dna binding residues prediction using a deep learning model with hierarchical feature extraction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 2022.

[8] Y. Ding, J. Tang, and F. Guo, "Identification of protein–ligand binding sites by sequence information and ensemble classifier," *Journal of Chemical Information and Modeling,* vol. 57, no. 12, pp. 3149-3161, 2017.

[9] D.-J. Yu, J. Hu, J. Yang, H.-B. Shen, J. Tang, and J.-Y. Yang, "Designing template-free predictor for targeting protein-ligand binding sites with classifier ensemble and spatial clustering," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 10, no. 4, pp. 994-1008, 2013.