Supplemental File for "Integrating Unsupervised Language Model with Multi-View Multiple Sequence Alignments for High-Accuracy Inter-Chain Contact Prediction"

Zi Liu [1,2,#], Yi-Heng Zhu [3,#], Long-Chen Shen [1], Xuan Xiao [2], Wang-Ren Qiu [2,*], Dong-Jun Yu [1,*]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing, China, 210094

[2]Computer Department, Jingdezhen Ceramic University, Jingdezhen, China, 333403

[3] College of Artificial Intelligence, Nanjing Agricultural University, Nanjing, China, 210095

*All correspondence should be addressed to E-mail: njyudj@njust.edu.cn

#These authors contributed equally

Table of Content

Supporting Texts

Text S1. The details of ESM-MSA transformer

A. Masking

For an input MSA, the masking strategy is performed. Specifically, for each 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.

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

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

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

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.

C. Initial embedding

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

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

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

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

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

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

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

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

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

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 $\boldsymbol{H}_1$:

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

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

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:

$$\boldsymbol{H}_1 \leftarrow dropout(\boldsymbol{H}_1, r) \tag{10}$$

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.

$$\boldsymbol{H}_k^B = BN(\boldsymbol{H}_k) \tag{11}$$

$$\boldsymbol{H}_k^R = RA(\boldsymbol{H}_k^B) \tag{12}$$

$$\boldsymbol{H}_k^R \leftarrow dropout(\boldsymbol{H}_k^R, r) \tag{13}$$

$$\boldsymbol{F}_k = SC(\boldsymbol{H}_k, \boldsymbol{H}_k^R) = \boldsymbol{H}_k + \boldsymbol{H}_k^R \tag{14}$$

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. 8-9), $SC(\cdot)$ is the short connection, and $RA(\cdot)$ is the row attention layer (see Eqs. 23-30), $\boldsymbol{H}_k$, $\boldsymbol{H}_k^B$, $\boldsymbol{H}_k^R$, $\boldsymbol{F}_k \in R^{M \times L \times D}$.

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

$$\boldsymbol{F}_k^B = BN(\boldsymbol{F}_k) \tag{15}$$

$$\boldsymbol{F}_k^C = CA(\boldsymbol{F}_k^B) \tag{16}$$

$$\boldsymbol{F}_k^C \leftarrow dropout(\boldsymbol{F}_k^C, r) \tag{17}$$

$$\boldsymbol{U}_k = SC(\boldsymbol{F}_k, \boldsymbol{F}_k^C) = \boldsymbol{F}_k + \boldsymbol{F}_k^C \tag{18}$$

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. 31-39), 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.

$$\boldsymbol{U}_k^B = BN(\boldsymbol{U}_k) \tag{19}$$

$$\boldsymbol{U}_k^F = FFN(\boldsymbol{U}_k^B) \tag{20}$$

$$\boldsymbol{U}_k^F \leftarrow dropout(\boldsymbol{U}_k^F, r) \tag{21}$$

$$\boldsymbol{H}_{k+1} = SC(\boldsymbol{U}_k, \boldsymbol{U}_k^F) = \boldsymbol{U}_k + \boldsymbol{U}_k^F \tag{22}$$

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. 40-45), 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.

$$\boldsymbol{Q}_{kt}^R = \boldsymbol{H}_k^B \boldsymbol{W}_{kt}^{QR} = \begin{bmatrix} \boldsymbol{H}_k^B[1] \\ \boldsymbol{H}_k^B[2] \\ \cdots \\ \boldsymbol{H}_k^B[M] \end{bmatrix} \boldsymbol{W}_{kt}^{QR} = \begin{bmatrix} \boldsymbol{H}_k^B[1]\boldsymbol{W}_{kt}^{QR} \\ \boldsymbol{H}_k^B[2]\boldsymbol{W}_{kt}^{QR} \\ \cdots \\ \boldsymbol{H}_k^B[M]\boldsymbol{W}_{kt}^{QR} \end{bmatrix} \in R^{M\times L\times(\frac{D}{m})} \quad (23)$$

$$\boldsymbol{K}_{kt}^R = \boldsymbol{H}_k^B \boldsymbol{W}_{kt}^{KR} = \begin{bmatrix} \boldsymbol{H}_k^B[1] \\ \boldsymbol{H}_k^B[2] \\ \cdots \\ \boldsymbol{H}_k^B[M] \end{bmatrix} \boldsymbol{W}_{kt}^{KR} = \begin{bmatrix} \boldsymbol{H}_k^B[1]\boldsymbol{W}_{kt}^{KR} \\ \boldsymbol{H}_k^B[2]\boldsymbol{W}_{kt}^{KR} \\ \cdots \\ \boldsymbol{H}_k^B[M]\boldsymbol{W}_{kt}^{KR} \end{bmatrix} \in R^{M\times L\times(\frac{D}{m})} \quad (24)$$

$$\boldsymbol{V}_{kt}^R = \boldsymbol{H}_k^B \boldsymbol{W}_{kt}^{VR} = \begin{bmatrix} \boldsymbol{H}_k^B[1] \\ \boldsymbol{H}_k^B[2] \\ \cdots \\ \boldsymbol{H}_k^B[M] \end{bmatrix} \boldsymbol{W}_{kt}^{VR} = \begin{bmatrix} \boldsymbol{H}_k^B[1]\boldsymbol{W}_{kt}^{VR} \\ \boldsymbol{H}_k^B[2]\boldsymbol{W}_{kt}^{VR} \\ \cdots \\ \boldsymbol{H}_k^B[M]\boldsymbol{W}_{kt}^{VR} \end{bmatrix} \in R^{M\times L\times(\frac{D}{m})} \quad (25)$$

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

where $\boldsymbol{H}_k^B$ is the input matrix of row attention layer in the k-th self-attention block (See Eq. 12), $\boldsymbol{Q}_{kt}^R$, $\boldsymbol{K}_{kt}^R$, 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:

$$\boldsymbol{W}_{kt}^{AR} = SoftMax(\frac{\sum_{i=1}^{M} \boldsymbol{Q}_{kt}^R[i]\cdot(\boldsymbol{K}_{kt}^R[i])^T\}}{\sqrt{MD/m}}) \in R^{L\times L},\ \boldsymbol{Q}_{kt}^R[i],\ \boldsymbol{K}_{kt}^R[i] \in R^{L\times(D/m)} \quad (26)$$

$$\boldsymbol{W}_{kt}^{AR} \leftarrow dropout(\boldsymbol{W}_{kt}^{AR}, r) \quad (27)$$

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:

$$\boldsymbol{A}_{kt}^{R} = \boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^{R} = \boldsymbol{W}_{kt}^{AR}\begin{bmatrix} \boldsymbol{V}_{kt}^{R}[1] \\ \boldsymbol{V}_{kt}^{R}[2] \\ \dots \\ \boldsymbol{V}_{kt}^{R}[M] \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^{R}[1] \\ \boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^{R}[2] \\ \dots \\ \boldsymbol{W}_{kt}^{AR}\boldsymbol{V}_{kt}^{R}[M] \end{bmatrix} \in R^{M\times L\times\left(\frac{D}{m}\right)}, \boldsymbol{V}_{kt}^{R}[i] \in R^{L\times\left(\frac{D}{m}\right)} \quad (28)$$

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:

$$\boldsymbol{A}_{k}^{R} = \boldsymbol{A}_{k1}^{R}\boldsymbol{A}_{k2}^{R} \dots \boldsymbol{A}_{km}^{R} \in R^{M\times L\times D} \quad (29)$$

$$\boldsymbol{H}_{k}^{R} = \boldsymbol{A}_{k}^{R}\boldsymbol{W}_{k}^{R} + \boldsymbol{b}_{k}^{R} = \begin{bmatrix} \boldsymbol{A}_{k}^{R}[1] \\ \boldsymbol{A}_{k}^{R}[2] \\ \dots \\ \boldsymbol{A}_{k}^{R}[M] \end{bmatrix}\boldsymbol{W}_{k}^{R} + \boldsymbol{b}_{k}^{R} = \begin{bmatrix} \boldsymbol{A}_{k}^{R}[1]\boldsymbol{W}_{k}^{R} \\ \boldsymbol{A}_{k}^{R}[2]\boldsymbol{W}_{k}^{R} \\ \dots \\ \boldsymbol{A}_{k}^{R}[M]\boldsymbol{W}_{k}^{R} \end{bmatrix} + \boldsymbol{b}_{k}^{R} \in R^{M\times L\times D} \quad (30)$$

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

where $\boldsymbol{H}_{k}^{R}$ in the output matrix of row attention layer in the k-th attention block (See Eq. 12), 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.

$$\boldsymbol{Q}_{kt}^{C} = \boldsymbol{F}_{k}^{B}\boldsymbol{W}_{kt}^{QC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1] \\ \boldsymbol{F}_{k}^{B}[2] \\ \dots \\ \boldsymbol{F}_{k}^{B}[M] \end{bmatrix}\boldsymbol{W}_{kt}^{QC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1]\boldsymbol{W}_{kt}^{QC} \\ \boldsymbol{F}_{k}^{B}[2]\boldsymbol{W}_{kt}^{QC} \\ \dots \\ \boldsymbol{F}_{k}^{B}[M]\boldsymbol{W}_{kt}^{QC} \end{bmatrix} \in R^{M\times L\times\left(\frac{D}{m}\right)} \quad (31)$$

$$\boldsymbol{K}_{kt}^{C} = \boldsymbol{F}_{k}^{B}\boldsymbol{W}_{kt}^{KC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1] \\ \boldsymbol{F}_{k}^{B}[2] \\ \dots \\ \boldsymbol{F}_{k}^{B}[M] \end{bmatrix}\boldsymbol{W}_{kt}^{KC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1]\boldsymbol{W}_{kt}^{KC} \\ \boldsymbol{F}_{k}^{B}[2]\boldsymbol{W}_{kt}^{KC} \\ \dots \\ \boldsymbol{F}_{k}^{B}[M]\boldsymbol{W}_{kt}^{KC} \end{bmatrix} \in R^{M\times L\times\left(\frac{D}{m}\right)} \quad (32)$$

$$\boldsymbol{V}_{kt}^{C} = \boldsymbol{F}_{k}^{B}\boldsymbol{W}_{kt}^{VC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1] \\ \boldsymbol{F}_{k}^{B}[2] \\ \dots \\ \boldsymbol{F}_{k}^{B}[M] \end{bmatrix}\boldsymbol{W}_{kt}^{VC} = \begin{bmatrix} \boldsymbol{F}_{k}^{B}[1]\boldsymbol{W}_{kt}^{VC} \\ \boldsymbol{F}_{k}^{B}[2]\boldsymbol{W}_{kt}^{VC} \\ \dots \\ \boldsymbol{F}_{k}^{B}[M]\boldsymbol{W}_{kt}^{VC} \end{bmatrix} \in R^{M\times L\times\left(\frac{D}{m}\right)} \quad (33)$$

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

where $\boldsymbol{F}_{k}^{B}$ is the input matrix of column attention layer in the k-th self-attention block (see Eq. 16), $\boldsymbol{Q}_{kt}^{C}$, $\boldsymbol{K}_{kt}^{C}$, 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:

$$\boldsymbol{W}_{kt}^{AC} = SoftMax\left(\frac{\boldsymbol{Q}_{kt}^{C}(\boldsymbol{K}_{kt}^{C})^{T}}{\sqrt{D/m}}\right) \in R^{M\times L\times M} \tag{34}$$

$$\boldsymbol{W}_{kt}^{AC} \leftarrow dropout(\boldsymbol{W}_{kt}^{AC}, r) \tag{35}$$

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

$$[\boldsymbol{Q}_{kt}^{C}[:,1,:] \cdot \boldsymbol{K}_{kt}^{C}[:,1,:]^{T}\ \boldsymbol{Q}_{kt}^{C}[:,2,:] \cdot \boldsymbol{K}_{kt}^{C}[:,2,:]^{T} \dots \boldsymbol{Q}_{kt}^{C}[:,L,:] \cdot \boldsymbol{K}_{kt}^{C}[:,L,:]^{T}] \in R^{M\times L\times M} \tag{36}$$

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

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:

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

$$\left.\boldsymbol{V}_{kt}^{C}[:,1,:]\ \ \boldsymbol{W}_{kt}^{AC}[:,2,:] \cdot \boldsymbol{V}_{kt}^{C}[:,2,:]\ \ \dots \boldsymbol{W}_{kt}^{AC}[:,L,:] \cdot \boldsymbol{V}_{kt}^{C}[:,L,:]\right] \in R^{M\times L\times\left(\frac{D}{m}\right)} \tag{37}$$

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

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:

$$\boldsymbol{A}_{k}^{C} = \boldsymbol{A}_{k1}^{C}\boldsymbol{A}_{k2}^{C} \dots \boldsymbol{A}_{km}^{C} \in R^{M\times L\times D} \tag{38}$$

$$\boldsymbol{F}_{k}^{C} = \boldsymbol{A}_{k}^{C}\boldsymbol{W}_{k}^{C} + \boldsymbol{b}_{k}^{C} = \begin{bmatrix} \boldsymbol{A}_{k}^{C}[1] \\ \boldsymbol{A}_{k}^{C}[2] \\ \dots \\ \boldsymbol{A}_{k}^{C}[M] \end{bmatrix} \boldsymbol{W}_{k}^{C} = \begin{bmatrix} \boldsymbol{A}_{1}^{C}[1]\boldsymbol{W}_{k}^{C} \\ \boldsymbol{A}_{2}^{C}[2]\boldsymbol{W}_{k}^{C} \\ \dots \\ \boldsymbol{A}_{k}^{C}[M]\boldsymbol{W}_{k}^{C} \end{bmatrix} + \boldsymbol{b}_{k}^{C} \in R^{M\times L\times D} \tag{39}$$

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

where $\boldsymbol{F}_{k}^{C}$ in the output matrix of column attention layer in the k-th attention block, (See Eq. 16), and $\boldsymbol{W}_{k}^{C}$ and $\boldsymbol{b}_{k}^{C}$ are weight matrix and bias in the linear unit, respectively.

(C) Feed-forward network

$$\boldsymbol{T}_k^F = gelu(\boldsymbol{U}_k^B \boldsymbol{W}_k^1 + \boldsymbol{b}_k^1) \in R^{M \times L \times D_1} \quad (40)$$

$$\boldsymbol{T}_k^F \leftarrow dropout(\boldsymbol{T}_k^F, r) \quad (41)$$

$$\boldsymbol{U}_k^F = \boldsymbol{T}_k^F \boldsymbol{W}_k^2 + \boldsymbol{b}_k^2 \in R^{M \times L \times D} \quad (42)$$

$$gelu(x) = x\emptyset(x) \quad (43)$$

$$\boldsymbol{U}_k^B \boldsymbol{W}_k^1 = \begin{bmatrix} \boldsymbol{U}_k^B[1] \\ \boldsymbol{U}_k^B[2] \\ \cdots \\ \boldsymbol{U}_k^B[M] \end{bmatrix} \boldsymbol{W}_k^1 = \begin{bmatrix} \boldsymbol{U}_k^B[1]\boldsymbol{W}_k^1 \\ \boldsymbol{U}_k^B[2]\boldsymbol{W}_k^1 \\ \cdots \\ \boldsymbol{U}_k^B[M]\boldsymbol{W}_k^1 \end{bmatrix} \in R^{M \times L \times D_1} \quad (44)$$

$$\boldsymbol{T}_k^F \boldsymbol{W}_k^2 = \begin{bmatrix} \boldsymbol{T}_k^F[1] \\ \boldsymbol{T}_k^F[2] \\ \cdots \\ \boldsymbol{T}_k^F[M] \end{bmatrix} \boldsymbol{W}_k^2 = \begin{bmatrix} \boldsymbol{T}_k^F[1]\boldsymbol{W}_k^2 \\ \boldsymbol{T}_k^F[2]\boldsymbol{W}_k^2 \\ \cdots \\ \boldsymbol{T}_k^F[M]\boldsymbol{W}_k^2 \end{bmatrix} \in R^{M \times L \times D} \quad (45)$$

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

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. 20), $\boldsymbol{W}_k^1$ 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.

G. 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:

$$\boldsymbol{P} = SoftMax(\boldsymbol{H}_{N+1}\boldsymbol{W}^O + \boldsymbol{b}^O) \in R^{M \times L \times C_{max}} \quad (46)$$

$$\boldsymbol{H}_{N+1}\boldsymbol{W}^O = \begin{bmatrix} \boldsymbol{H}_{N+1}[1]\boldsymbol{W}^O \\ \boldsymbol{H}_{N+1}[2]\boldsymbol{W}^O \\ \cdots \\ \boldsymbol{H}_{N+1}[M]\boldsymbol{W}^O \end{bmatrix}, \boldsymbol{H}_{N+1}[i] \in R^{L \times D}, \boldsymbol{W}^O \in R^{D \times C_{max}} \quad (47)$$

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.

F. Loss function

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

$$Loss_{msa} = \frac{1}{M} \cdot \sum_{i=1}^{M} \{ \frac{1}{|mask(i)|} \cdot \sum_{j \in mask(i)} -log\boldsymbol{P}_{i,j,c(i,j)} \} \quad (48)$$

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 Tables

Table S1. Average precision in top *N* predicted contacts on TS630 dataset.

Methods	Top 1	Top 5	Top 10	Top 20	Top 50	Top 100
ICCPred	**0.267**	**0.245**	**0.240**	**0.231**	**0.217**	**0.198**
GLINTER	0.153	0.154	0.160	0.154	0.146	0.139
HDIContact	0.041	0.038	0.035	0.040	0.081	0.103

Table S2. Average precision in top *L*/*K* predicted contacts, AUPR, and AUROC on TS630 dataset. ALL represents the number of native contacts on the target.

Methods	AUROC	AUPR	Top *L*/30	Top *L*/20	Top *L*/10	Top *L*/5	Top *L*/2	Top ALL
ICCPred	**0.725**	**0.115**	**0.238**	**0.233**	**0.225**	**0.210**	**0.181**	**0.133**
GLINTER	0.411	0.111	0.156	0.152	0.149	0.143	0.137	0.116
HDIContact	0.659	0.070	0.035	0.035	0.055	0.091	0.107	0.091

Table S3. Average contact precision of different features on the TS630 dataset.

Feature	Top										
	1	5	10	20	50	100	*L*/30	*L*/20	*L*/10	*L*/5	*L*/2
ESM2	0.046	0.039	0.036	0.038	0.039	0.038	0.039	0.038	0.038	0.038	0.037
CPX	0.251	0.233	0.218	0.209	0.199	0.181	0.218	0.214	0.205	0.193	0.167
GDS	0.190	0.177	0.170	0.167	0.157	0.145	0.171	0.168	0.161	0.152	0.133
PPIS	0.157	0.160	0.157	0.151	0.142	0.131	0.159	0.154	0.148	0.138	0.122
PIS	0.237	0.223	0.214	0.201	0.186	0.171	0.214	0.207	0.194	0.181	0.158
PP	0.208	0.203	0.200	0.194	0.179	0.166	0.202	0.198	0.187	0.175	0.154
GP	0.235	0.227	0.223	0.218	0.201	0.186	0.219	0.218	0.208	0.195	0.172
GI	0.195	0.190	0.188	0.181	0.172	0.160	0.186	0.184	0.178	0.167	0.149
GSP	**0.267**	**0.245**	**0.240**	**0.231**	**0.217**	**0.198**	**0.238**	**0.233**	**0.225**	**0.210**	**0.181**

Supporting Figures

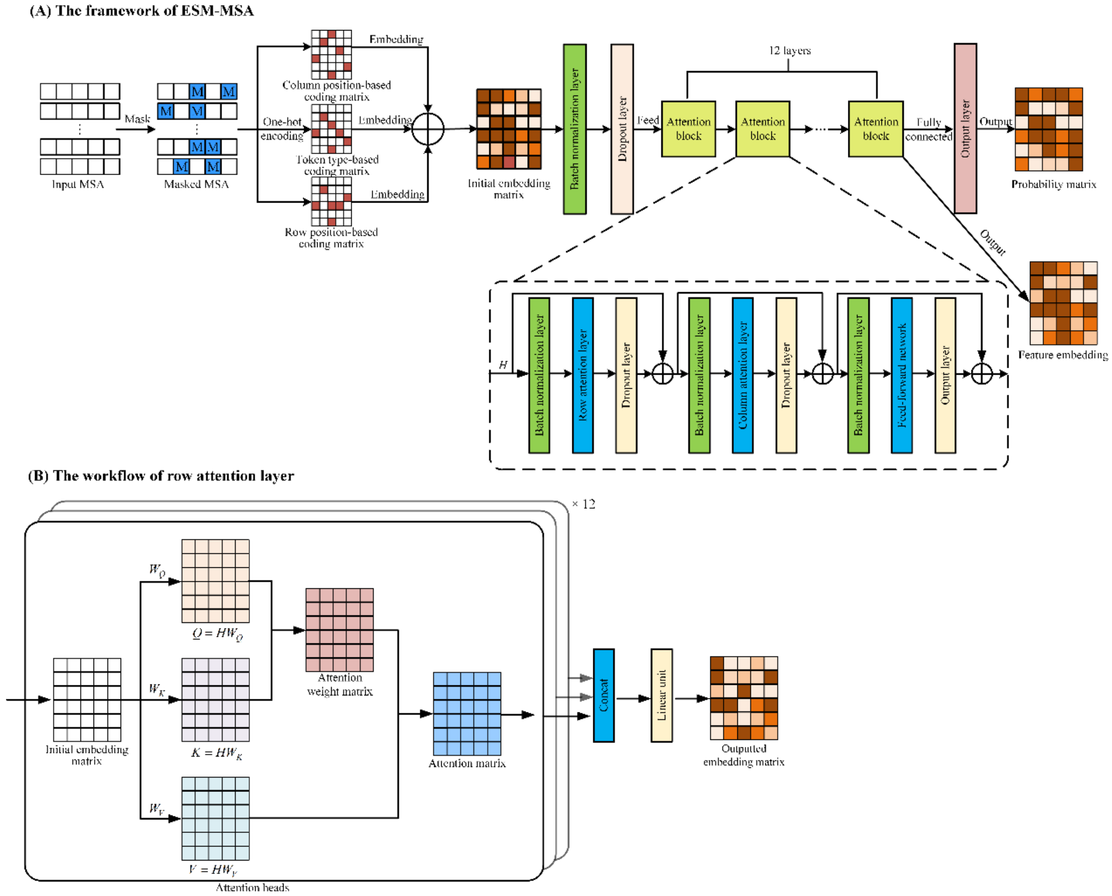

Figure S1. The workflow of ESM-MSA

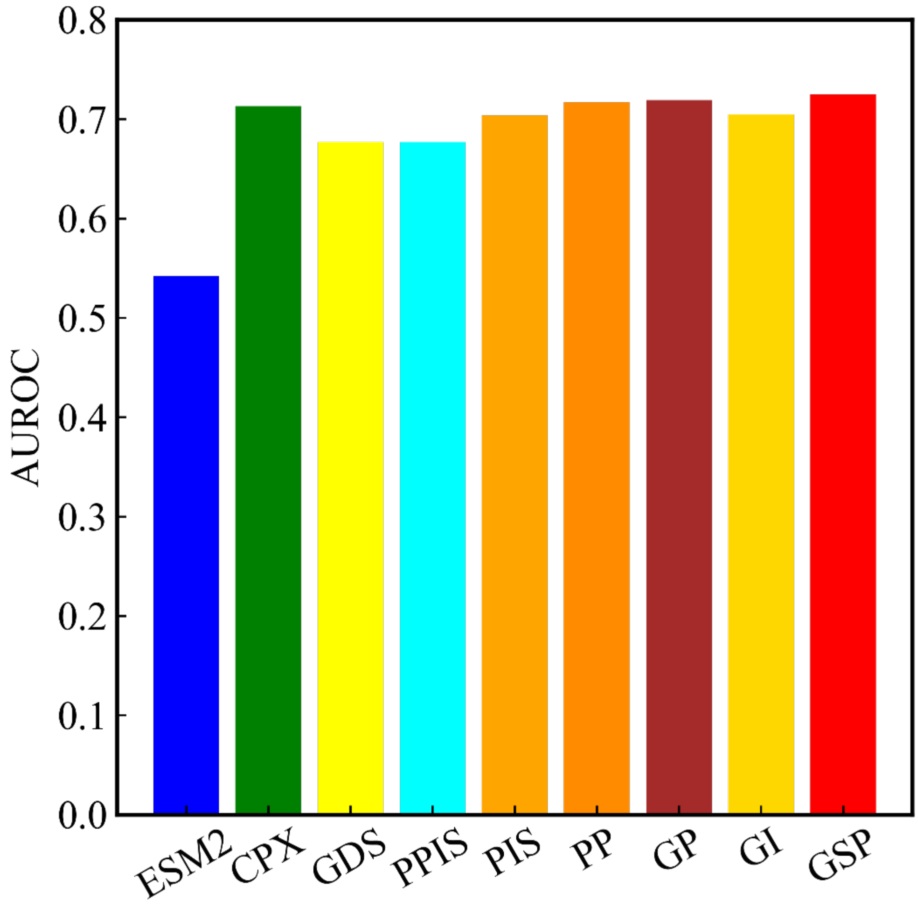

Figure S2. Performance comparison between nine feature embeddings regarding AUROC on the TS630 dataset.