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

A. Masking

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

B. One-hot encoding

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

C. Embedding with positions

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

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

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

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

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

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

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

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

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

D. Self-attention

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

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

$$M_{i,j}^{Q} = H_i W_{i,j}^{Q},\ M_{i,j}^{K} = H_i W_{i,j}^{K},\ M_{i,j}^{V} = H_i W_{i,j}^{V} \quad \text{(S6)}$$

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

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

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

$$A_i = A_{i,1} A_{i,2} \ldots A_{i,m} \quad \text{(S8)}$$

$$U_i = A_i W_i^1 + b_i^1,\ W_i^1 \in R^{D\times D},\ A_i,\ b_i^1, U_i \in R^{L\times D} \quad \text{(S9)}$$

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:

$$F_i = H_i + U_i \quad \text{(S10)}$$

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

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

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:

$$H_{i+1} = F_i + T_i,\ H_{i+1} \in R^{L\times D} \quad \text{(S13)}$$

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

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

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

F. Loss function

The loss function is designed as:

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

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.