Journal of Cheminformatics

METHODOLOGY Open Access

SESNet: sequence-structure feature-integrated deep learning method for data-efficient protein engineering

Mingchen Li[1,4†], Liqi Kang[1,3†], Yi Xiong[5], Yu Guang Wang[1,2], Guisheng Fan[4], Pan Tan[1,2*] and Liang Hong[1,2,3*]

Abstract

Deep learning has been widely used for protein engineering. However, it is limited by the lack of sufficient experimental data to train an accurate model for predicting the functional fitness of high-order mutants. Here, we develop SESNet, a supervised deep-learning model to predict the fitness for protein mutants by leveraging both sequence and structure information, and exploiting attention mechanism. Our model integrates local evolutionary context from homologous sequences, the global evolutionary context encoding rich semantic from the universal protein sequence space and the structure information accounting for the microenvironment around each residue in a protein. We show that SESNet outperforms state-of-the-art models for predicting the sequence-function relationship on 26 deep mutational scanning datasets. More importantly, we propose a data augmentation strategy by leveraging the data from unsupervised models to pre-train our model. After that, our model can achieve strikingly high accuracy in prediction of the fitness of protein mutants, especially for the higher order variants (> 4 mutation sites), when finetuned by using only a small number of experimental mutation data (< 50). The strategy proposed is of great practical value as the required experimental effort, i.e., producing a few tens of experimental mutation data on a given protein, is generally affordable by an ordinary biochemical group and can be applied on almost any protein.

Introduction

Proteins are workhorses of the life activities. Their various functions such as catalysis, binding, and transportation undertake most of the metabolic activities in cells. In addition, they are the key components of the cytoskeleton, supporting the stable and diverse form of organisms. Nature provides numerous proteins with great potential value for practical applications. However, the natural proteins often do not have the optimal function to meet the demand of bioengineering. Directed evolution is a widely used experimental method to optimize proteins' functionality, namely fitness, by employing a greedy local search to optimize protein fitness [1, 2]. During this process, gain-of-function mutants are achieved and optimized via mutating several Amino Acids (AA) in the protein, which were selected and accumulated through the iterative processes of mutation by testing hundreds to thousands of variants in each generation. Despite the

[†] Mingchen Li and Liqi Kang have contributed equally to this work

*Correspondence:
Pan Tan
tpan1039@alumni.sjtu.edu.cn
Liang Hong
hongl3liang@sjtu.edu.cn
[1] Shanghai National Center for Applied Mathematics (SJTU Center), & Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai 200240, China
[2] Shanghai Artificial Intelligence Laboratory, Shanghai 200240, China
[3] School of Physics and Astronomy & School of Pharmacy, Shanghai Jiao Tong University, Shanghai 200240, China
[4] School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200240, China
[5] School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China

great success directed evolution has achieved, the phase space of the protein fitness landscape can be screened by this method is rather limited. Furthermore, to acquire a mutant of excellent fitness, especially a high-order mutant with multiple AA being mutated, the directed evolution often needs to develop an effective high-throughput screening or conduct a large number of experimental tests, which is experimentally and economically challenging [3].

Since experimental screening for directed evolution is largely costing, particularly for high-order mutations, prediction of the fitness of protein variants in silico are highly desirable. Recently, deep learning methods have been applied for predicting the fitness landscape of the protein variants [2]. By building models trained to learn the sequence-function relationship, deep learning can predict the fitness of each mutant in the whole sequence space and give a list of the most favorable candidate mutants for experimental tests. Generally, these deep learning models can be classified into protein language models [4–11], learning the representations from the global unlabeled sequences [6, 7, 12] and multiple sequence alignment (MSA) based model, capturing the feature of evolutional information within the family of the protein targeted [13–16]. And more recent works have proposed to combine these two strategies: learning on evolutionary information together with global natural sequences as the representation [17, 18], and trained the model on the labelled experimental data of screened variants to predict the fitness of all possible sequences. Nevertheless, all these models are focused on protein sequence, i.e., using protein sequence as the input of the model. Apart from sequence information, protein structure can provide additional information on function. Due to the experimental challenge of determining the protein structure, the number of reported protein structures is orders of magnitude smaller than that of known protein sequences, which hinders the development of geometric deep learning model to leverage protein structural feature. Thanks to the dramatic breakthrough in deep learning-based technique for predicting protein structure [19, 20], especially AlphaFold 2, it is now possible to efficiently predict protein structures from sequences at a large scale [21]. Recently, some researches directly take the protein structure feature as input to train the geometric deep learning model, which has been proved to achieve better or similar performance in prediction of protein function compared to language models [22–24]. However, the fused deep-learning method which can make the use of both sequence and structural information of the protein to map the sequence-function is yet much to be explored [25].

Recently, both supervised and unsupervised models have been developed for protein engineering, i.e., prediction of the fitness of protein mutants [24, 26]. Generally speaking, the supervised model can often achieve better performance as compared to the unsupervised model [26], but the former requires a great amount (at least hundreds to thousands) of experimental mutation data of the protein studied for training, which is experimentally challenging [18]. In contrast, the unsupervised model does not need any of such experimental data, but its performance is relatively worse, especially for the high-order mutant, which is often the final product of a direct-evolution project. It is thus highly desirable to develop a deep-learning algorithm, which can efficiently and accurately predict the fitness of protein variants, especially the high-order mutant, without the need of a large size of experimental mutation data of the protein concerned. In the present work, we built a supervised deep learning model (SESNet), which can effectively fuse the protein sequence and structure information together to predict the fitness of variant sequences (Fig. 1A). We demonstrated that SESNet outperforms several state-of-the-art models on 26 metagenesis datasets. Moreover, to reduce the dependence of the model on the quantity of experimental mutation data, we proposed a data-augmentation strategy (Fig. 1B), where the model was firstly pre-trained using a large quantity of the low-quality results derived from the unsupervised model and then finetuned by a small amount of the high-quality experimental results. We showed that the proposed model can achieve very high accuracy in predicting the fitness of high-order variants of a protein, even for those with more than four mutation sites, when the experimental dataset used for finetuning is as small as 40. Moreover, our model can predict the key AA sites, which are crucial for the protein fitness, and thus the protein engineer can focus on these key sites for mutagenesis. This can greatly reduce the experiment cost of trial and error.

Results

Deep learning-based architecture of SESNet for predicting protein fitness

To exploit the diverse information from protein sequence, coevolution and structure, we fuse three encoder modules into our model. As shown in Fig. 1A: the first one (local encoder) got from MSA accounts for residue interdependence in a specific protein learned from homologous evolution-related sequences [15, 16]; the second one (global encoder) coming from protein language model, captures the sequence feature in global protein sequence universe [6, 12]; and the third one (structure module) captures surrounding structural features around each residue learned from 3D

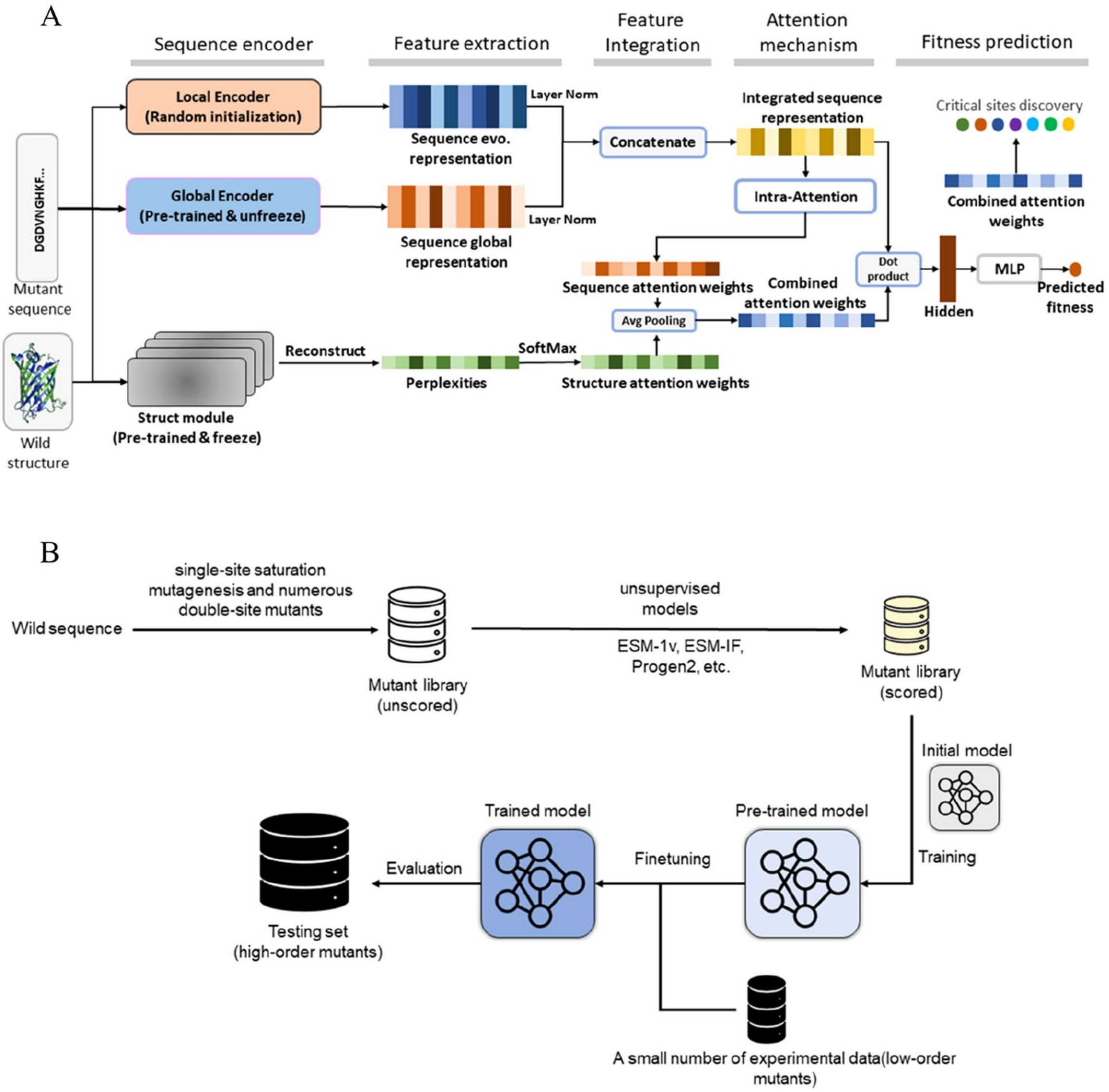
A
Sequence encoder
Feature extraction
Feature Integration
Attention mechanism
Fitness prediction
Local Encoder (Random initialization)
Global Encoder (Pre-trained & unfreeze)
DGDVNGHKF...
Mutant sequence
Wild structure
Layer Norm
Sequence evo. representation
Sequence global representation
Layer Norm
Concatenate
Integrated sequence representation
Intra-Attention
Critical sites discovery
Combined attention weights
Dot product
Hidden
MLP
Predicted fitness
Sequence attention weights
Combined attention weights
Avg Pooling
Struct module (Pre-trained & freeze)
Reconstruct
Perplexities
SoftMax
Structure attention weights
B
Wild sequence
single-site saturation mutagenesis and numerous double-site mutants
Mutant library (unscored)
unsupervised models
ESM-1v, ESM-IF, Progen2, etc.
Mutant library (scored)
Initial model
Training
Pre-trained model
Finetuning
Trained model
Evaluation
Testing set (high-order mutants)
A small number of experimental data(low-order mutants)

Fig. 1 Architecture of model and the schematic of data-augmentation strategy. Architecture of SESNet): The local encoder accounts for the inter-residue dependence in a protein learned from MSA of homologous sequences using a Markov random field [27]. The global encoder captures the sequence feature in global protein sequence universe using protein language model [6]. The structure module accounts for the microscopically environmental feature of a residue learned from 3D geometric structure of the protein [23, 28]. Schematic of data-augmentation strategy. **B**: We first build a mutant library containing all of the single-site mutants and numerous double-site mutants. Then, all of these mutated sequences are scored by the unsupervised model. After that, these mutants are used to pre-train the initial model (SESNet), which will be further finetuned on a small number of low-order experimental mutational data

geometric structure of the protein [23, 24]. While the protein language model is regarded as global encoder is because that it captured the rich semantic from the universal protein sequence space, such as the databases UniProt or Pfam, containing more than 100 million sequences of proteins of vastly different sequences and functions. As a result, the homologous sequences of the target protein are only a tiny portion of the universal protein sequence space. Such definition ("local" vs "global") has also been used in the Ref [17]. To integrate the information of different modules, we first concatenate representations of local and global encoders and get an integrated sequence representation. This integrated sequence representation is then sent to an attention layer and becomes the sequence attention weights, which will be further averaged with the structure attention weights derived from structure module, leading to the combined attention weights. Finally, the product of combined attention weights and the integrated sequence representation is then fed into a fully connected layer to generate the predicted fitness. The combined attention weights can also be used to predict the key AA sites, critical for the protein fitness, details of which is discussed in the section of Method.

SESNet outperforms state-of-the-art methods for predicting fitness of variants on deep mutation scan (DMS) datasets

We compared our supervised model against the existing state-of-the-art supervised models, ECNet [17], ESM-1b [6]; and unsupervised models, ESM-1v [9], ESM-IF1 [23] and MSA transformer [15]. As can be seen in Fig. 2A, in 19 out of 20 datasets, the supervised models generally outperform the unsupervised ones as expected, and our model (SESNet) achieves the best performance among all the models. Moreover, we further explored the ability of our model to predict the fitness of higher-order variants by training it using the experimental results of the low-order variants on 6 datasets of DMS. As shown in Fig. 2B and C, our model outperforms all the other models. Data in Fig. 2 is presented in Additional file 1: Tables S1–S3. These datasets cover various proteins and different types of functionalities, including catalytic rate, stability, and binding affinity to peptide, DNA, RNA and antibody, as well as fluorescence intensity (Additional file 1: Table S4). While most of the datasets contain only single-site mutants, five of them involve both single-site and double-site mutants, and the dataset of GFP contains data up to 15-site mutants.

All three components contribute positively to the performance of SESNet

As described in the above architecture (Fig. 1A), our model integrates three different encoders or modules together. To investigate how much contribution each of the three parts makes, we performed ablation studies in 20 datasets of single-site mutants. Briefly, we removed each of the three components and compared the performance to that of the original model. As shown in Additional file 1: Table S5, the average spearman correlation of the original model is 0.672, much higher than that without local encoder (0.639), that without global encoder (0.247) and that without structure module (0.630). The ablation study reveals that all three components contribute to the improvement of model performance, and the contribution from the global encoder, which captures the sequence feature in global protein sequence universe, is the most significant.

The combined attention weights guide the finding of the key AA site

The combined attention weights can be used to measure the importance of each AA site on protein fitness when mutated. To the first approximation, higher the attention score is, more important the AA site is. To test this approximation, we trained our model on the experimental data of 1084 single-site mutants in the dataset of GFP [29], a green fluorescent protein from *Aequorea victoria*. The ground truth of the key sites of GFP are defined here as the experimentally discovered top 20 sites, which exhibit the largest change of protein fitness when mutated, or the AAs forming and stabilizing the chromophore, which are known to significantly affect the fluorescent function of the protein [30], but lack the fitness results in the experimental dataset. Indeed, one can observe that, at least 5 out of 20 top attention-score AA sites predicted by our model are the key sites as two of them (G65 and T201) are located at the chromophore, and the other three (P73, R71 and G230) were among the top 20 residues discovered in experiment to render the highest change of fitness when mutated (Fig. 3A and Additional file 1: Figure S1A). Interestingly, when we removed the structure module from the model, only three residues in the predicted top-20 attention-score AA is the key site (Fig. 3B and Additional file 1: Figure S1B).

To further verify this discovery, we also performed these tests on the dataset of RRM, the RNA recognition motif of the *Saccharomyces cerevisiae* poly(A)-binding protein [31]. The key sites of RRM are defined as the experimentally discovered top 20 sites, which render the largest change of fitness of the protein when mutated, or the binding sites, which are within 5 Å of the RNA molecules as revealed in the structure of PDB 6R5K. Figure 3C and Additional file 1: Figure S2A show that 11 out of 20 top attention-score AA sites predicted by our model are the key AAs. Six of them (F4, L8, I12, I27 S29 and K31) are among the top 20 residues and seven of them (N7, L28, S29, K31, A33, T34 and K39) are binding sites. 3 key residues can be found in the predicted top-seven attention-score AAs, when we removed the structure module. (Fig. 3D and Additional file 1: Figure S2B). S29 is the binding site. A57 and I71 are among the experimentally-discovered top 20 sites.

The results in Fig. 3 demonstrate that the structural module which learns the microscopically structural information around each residue makes important contribution to identify the key AAs, which are crucial for the protein fitness. Although the ablation study (Additional file 1: Table S5) reveals that the addition of the structural module improves the average spearman correlation over 20 datasets only by 4 percent, Fig. 3 demonstrates an important role of the structural module, which

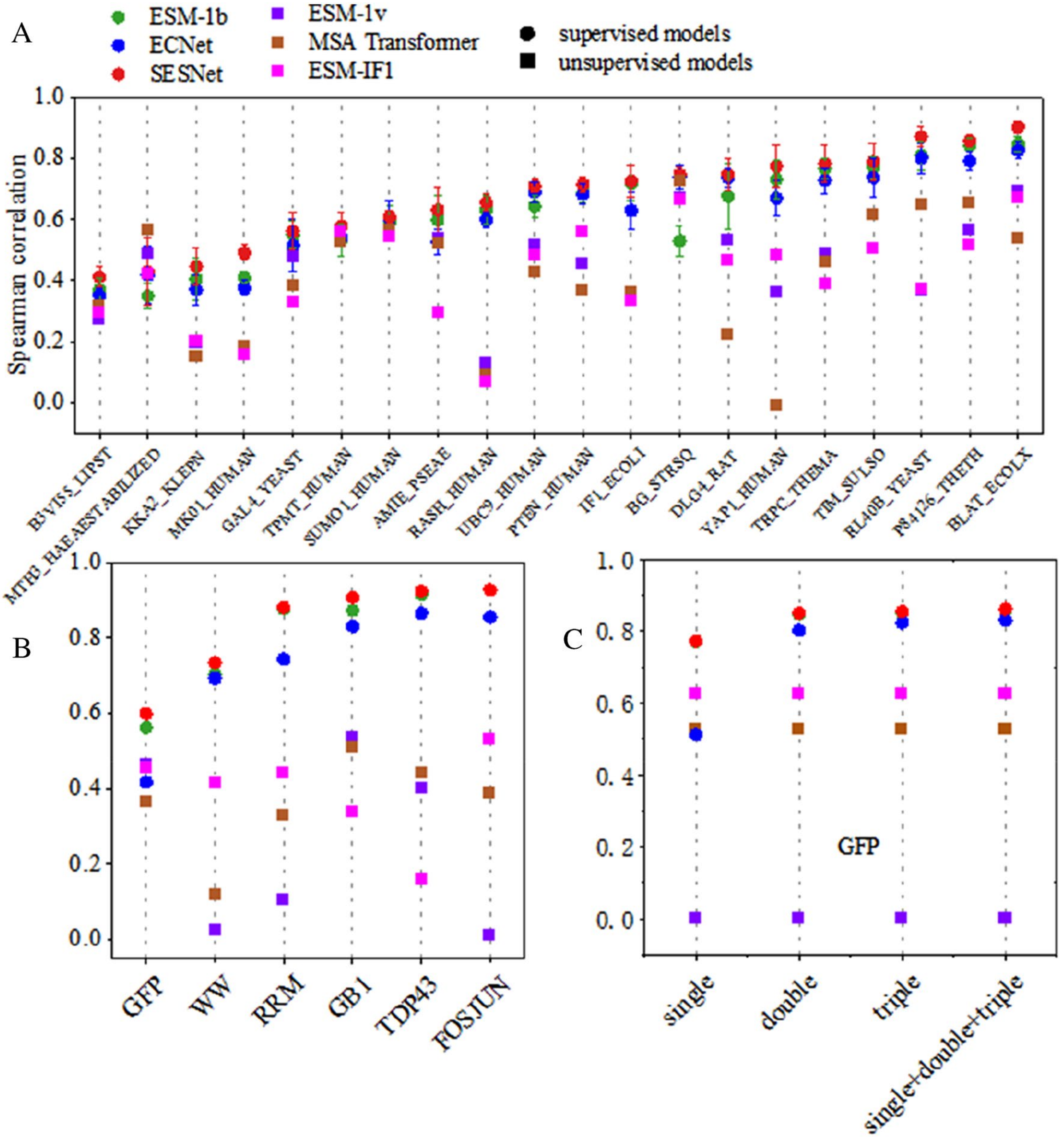

Fig. 2 Spearman correlation of predicted fitness. **A**: Comparison of our model to other models on the predicted fitness of the single-site mutants on 20 datasets. We randomly split a given dataset into five folds by randomized shuffling and splitting. All the supervised models are trained and evaluated for five times on different folds splitting. In the *i*-th iteration, the fold-*i* is used as the test set while the remaining four folds are used for training and validation. Later, we perform a simple random strategy to split the remaining four folds of dataset into training and validation as a ratio of 7:1. The error bars of each model are the standard deviations of the five-time testing results. **B**: Comparison of predicted fitness of double-site mutants of our model with other unsupervised models (ESM-1v, ESM-IF1 and MSA transformer), or supervised models (ECNet and ESM-1b). Here, we performed five-fold cross-validation on the data of single-site mutants and used double-site mutants as external test set. Briefly, we randomly split the data of single-site mutants into five folds, and then picked one fold as validation set and the remaining four folds as training set. This process was repeated five times and each fold of data was employed once as the validation set. The model that performed best in the validation set was tested on the double-site mutants. B: Comparison of our model to other models on fitness prediction of quadruple-site mutants of GFP. Here, our model and other supervised model were trained using the single, double, triple-site mutants and all the three together. Where the quadruple-site mutants are the external test set. We performed five-fold cross-validation on the train set and tests the models on quadruple-site mutants

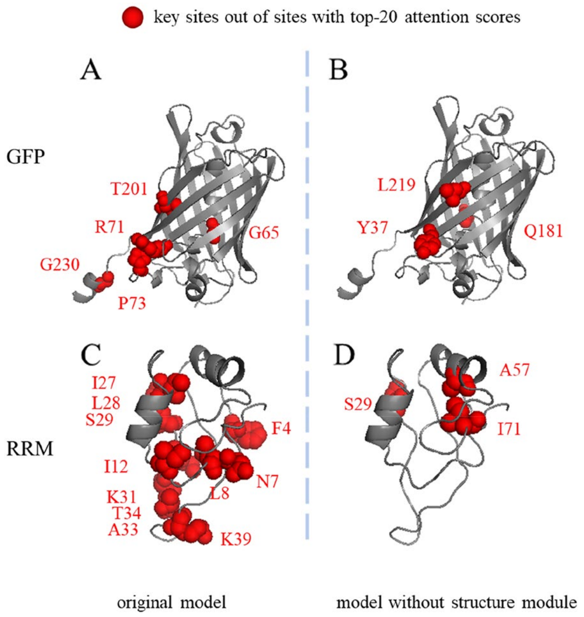

Fig. 3 The key sites out of the sites with the top 20 largest attention scores on the wildtype sequence. **A** and **B**: The key sites of GFP have been marked as red spheres. **A**: 5 key sites were recovered by our model. G65 and T201 are the active residues helping to form and stabilize the chromophore in GFP as described by Ref [30]. P73, G230 and R71 are among the experimentally-discovered top 20 sites, which render the highest change of fitness when mutated. **B**: 3 key sites were identified by the model when removing the structure module. Y37 and L219 are among the experimentally-discovered top 20 AA sites. Q181 is the active residue. **C** and **D**: The key sites of RRM have been marked as red spheres. **C**: 11 key sites were recovered by the original model. N7, L28, S29, K31, A33, T34 and K39 are the binding sites which are within 5 Å of the RNA molecules. F4, L8, I12, I27, S29 and K31 are among the experimentally-discovered top 20 sites, which render the highest change of fitness when mutated. **D**: There are 3 key sites identified by the model when removing the structure module. S29 is the binding site. A57 and I71 are among the experimentally-discovered top 20 sites

can guide the protein engineer to identify the important AA sites in a protein for mutagenesis.

As can be seen from the above comparison, our SESNet with structural module considered has much better performance in identifying the key amino acids as compared to the model without the structural module. We suspect this might result from the fact that the key AA site affecting the function of the protein the most has important structural roles, which are better captured when the structure module is implemented. However, one can also see that the performance of identifying the key AA site of our model on RRM is much better than on GFP. The former is testing the binding affinity between a protein and RNA molecule while the latter is examining the fluorescence intensity of a protein. Fluorescence intensity of GFP is a very fragile property, strongly depending on the local physicochemical environment to form the central chromophore and radiating of it. So much precise structural, spatial and chemical information of the amino acids surrounding the chromophore, including the orientation of the side groups and the true charge of them, could be essential for optimizing the fluorescence intensity. The current structure module in our SESNet is not sufficient to fully capture such information and needs improvement. But this is beyond the scope of the current work and will be done in the future.

Data-augmentation strategy boosts the performance of the fitness prediction when finetuned by a small size of labelled experimental data

Supervised model is normally performing better than the unsupervised models (see Fig. 2) [26]. But the accuracy of the supervised model is highly affected by the amount of input experimental results used for training. However, it is experimentally challenging and costly to generate sufficient data (many hundreds or even thousands) for such purpose on every protein studied. To address this challenge, we propose a simple strategy of data augmentation by using the result generated by one unsupervised model to pre-train our model on a given protein, and then finetuning it using a limited number of experimental results on the same protein. We call it a pre-trained model. We note that data-augmentation strategy has been applied in various earlier work and has achieved good success in protein design [23, 32, 33]. In particular, to improve the accuracy of inverse folding, ref [23] used 16,153 experimentally determined 3-D structures of proteins and 12 million structures predicted by the AlphaFold 2 [19] to train the model ESM-IF1 [23]. In the present work, the data augmentation strategy is used for a different purpose that it can reduce the dependence of the supervised model on the size of the experimental data when predicting the fitness of protein mutants. We took GFP as an example to illustrate our data-augmentation strategy as GFP has a large number of experimental data for testing, particularly the experimental data for high-order mutants (up to 15-site mutant). We used the fitness results of low-order mutants predicted by the unsupervised model, ESM-IF1, to pre-train our model. The pre-training dataset contains the fitness of all single-site mutants and 30,000 double-site mutants randomly selected out of tens of million double-site variants. Then, we finetuned the pre-trained model by a certain number of experimental results of single-site mutants. The resulting model was used to predict the fitness of high-order mutants. As can be seen in Fig. 4A–D, when comparing with the original model without pre-training (blue bars), the performance of the pre-trained model is significantly improved (red bars). Such improvement is particularly large when only a small number of experimental data (40) is fed for

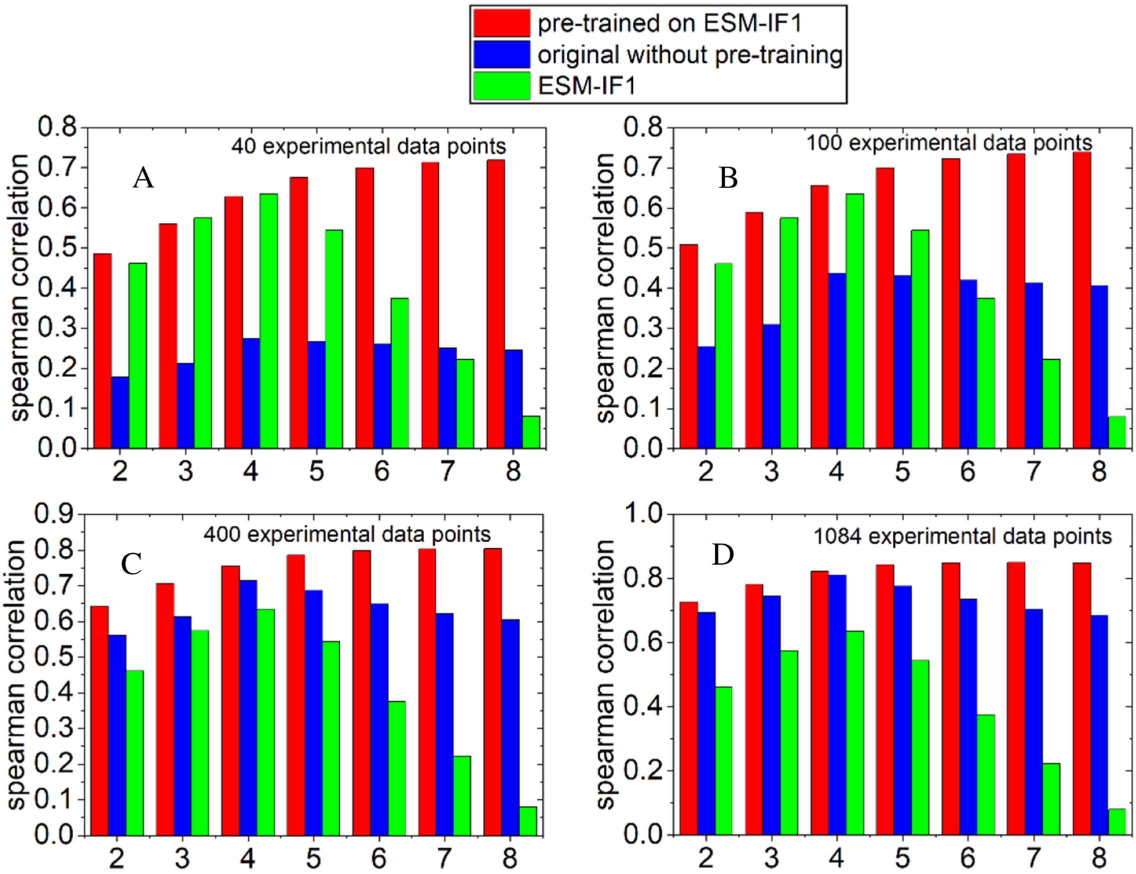

Fig. 4 Results of models trained on different number of experimental variants. **A**–**D**: The spearman correlation of fitness prediction on multiple sites (2–8 sites) mutants by finetuning using 40, 100, 400, 1084 single-site experimental mutation results from dataset of GFP. Where the red and blue bars represent the results of the pre-trained model and the original model without pretraining, respectively. And the green bars correspond to the results of the unsupervised model ESM-IF1 as a control

training, and it will be gradually reduced when feeding more experimental data, eventually disappearing when more than 1000 experimental data were used for training. Here, we would like to particularly highlight the case when the finetuning experimental dataset contains only 40 data points. As can be seen in Fig. 4A, the pretrained model can achieve high spearman correlation of 0.5–0.7 for multisite-mutants, even for high-order mutants with 5–8 mutation sites. This is remarkably important for most protein engineers, as such experimental workload (40 data points) is generally affordable in an ordinary biochemical research group. However, without pre-training, the performance of the supervised model is rather low (~ 0.2). This comparison demonstrates the advantage of the data augmentation strategy proposed in the present work.

Moreover, we also compared the performance of the pretrained model with respect to the unsupervised model (green bars), which were used for generating the low-quality pretraining datasets. As can be seen, when only 40 experimental data were used for training, the pretrained model has similar performance as compared to the unsupervised model for low-order mutants (< 4 mutation sites), but clearly outperforms the latter for high-order mutants (> 4 mutation sites). When feeding more experimental data, especially a couple of hundreds, the pretrained model will outperform the unsupervised model regardless of how many sites of the protein were mutated.

The unsupervised model used for analysis in Fig. 4 is ESM-1F1, which captures the surrounding structural information of a residue. To demonstrate the general superiority of data-augmentation strategy proposed here, we also tested the results using other unsupervised model to generate the augmented datasets for GFP. As can be seen in Additional file 1: Figure S3, we used ProGen2 [8], an unsupervised model to learn the global sequence information, for data augmentation, and still derived the similar conclusion as in Fig. 4. That is, the pretrained model outperforms the original model without pretraining especially when a small experimental dataset is used for training, and it also beats the unsupervised model particularly for the high-order mutants.

To further validate the generality of the data augmentation strategy proposed here, we did the analysis on the dataset of other proteins: toxin-antitoxin complex (F7YBW8) [34]containing data up to 4 sites mutants, and Adeno-associated virus capsids (CAPSD_AAV2S) [35], a deep mutational dataset including data up to 23-site mutants. We used the unsupervised model ProGen2 [8] to generate the low-quality data of F7YBW8 for pretraining, since we found ProGen2 performs better than ESM-IF1 on this dataset. As shown in Fig. 5A, the pre-trained model outperforms both the original model without pretraining and the unsupervised model in the fitness prediction of all multi-site mutants (2–4 sites) after finetuned by using only 37 experimental data points. In addition, in the dataset of CAPSD_AAV2S (Fig. 5B), the pre-trained model also achieves the best performance in all of the high-order mutants ranging from 2 to 23 sites, when finetuned by only 20 experimental data points. These results further support the practical use of our data augmentation strategy, as the required experimental effort is largely affordable on most proteins.

In addition, we also compared the performance of the pretrained model with respect to the original model without pretraining on the prediction of single-site mutants. As shown in Additional file 1: Figure S4, our pre-trained model generally outperforms the original one in majority of datasets: 18 out of 20 datasets when finetuning on 20 experimental data points, and in 19 out of 20 datasets when finetuning on 40 or 100 experimental data points. These results further support the value of the data augmentation strategy proposed in the present work.

Learned models provide insight into protein fitness

SESNet projects a protein sequence into a high dimensional latent space and represents each mutant as a vector by the last hidden layer. Thus, we can visualize the relationships between sequences in these latent spaces to reveal how the networks learn and comprehend protein fitness. Specifically, we trained SESNet on the experimental data of single-site mutants from the datasets of GFP and RRM, then we used the trained model and untrained model to encode each variant and extracted the output of the last hidden layer as a representation of the variant sequence. Additional file 1: Figure S5 shows a two-dimensional projection of the high dimensional latent space using t-SNE [36]. We found that the representations of positive and negative variants, i.e., the experimental fitness values being larger or smaller than that of wildtype, generated by the trained SESNet are clearly clustered into distinct groups (Additional file 1: Figure S5A, B). In contrast, the representations from untrained model cannot provide a distinguishable boundary between positive and negative variants (Additional file 1: Figure S5C, D). Therefore, SESNet can learn to distinguish functional fitness of mutants into a latent representation space with supervised training.

Furthermore, to explore why the data-augmentation strategy works, we performed a case study on GFP dataset. Here, we compared the latent-space representation from the last hidden layer generated by our model with and without pre-training using the augmented data from the unsupervised model. As seen in Additional file 1: Figure S6A, after pretraining even without finetuning by the experimental data, SESNet can already roughly

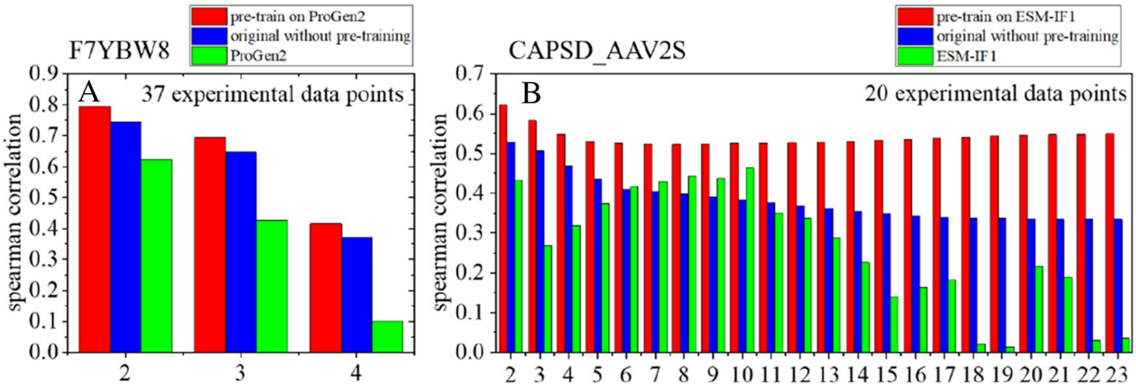

Fig. 5 Results of models trained on different datasets. **A**–**B**: The spearman correlation of fitness prediction on high-order mutants by finetuning on 37 experimental single-site mutation results from datasets of F7YBW8 and on 20 experimental single-site mutation results of CAPSD_AAV2S, respectively. Where the red and blue bars represent the results of the pre-trained model and the original model without pretraining. And the green bars correspond to the results of the unsupervised model, which is ProGen2 for F7YBW8 and ESM-IF1 for CAPSD_AAV2S, respectively

distinguish the negative and positive mutants. One thus can deduce that the pre-training can furnish a good parameter initialization for SESNet. After further fine-tuning the pre-trained SESNet by only 40 experimental data points of single-site mutants, a rather clear boundary between negative and positive high-order mutants is further outlined (Additional file 1: Figure S6B). In contrast, when we skipped the pretraining process, i.e., directly training the model on 40 experimental data points, the separation between the positive and negative high-order mutants is rather ambiguous (Additional file 1: Figure S6C). This comparison demonstrates the superiority of our data-augmentation strategy in distinguishing mutants of distinct fitness values, when the number of available experimental data is limited.

Discussion

In this study, we present a supervised deep learning model, which leverages the information of both sequence and structure of protein to predict the fitness of variants. And this model is found to outperform the existing state-of-the-art ones for protein engineering. Moreover, we proposed a data augmentation strategy, which pretrains our model using the results predicted by other unsupervised model, and then finetunes the model with only a small number of experimental results. We demonstrated that such data augmentation will significantly improve the accuracy of the model when the experimental results are very limited (~40), and also for high-order mutants with > 4 mutation sites. We noted that our work, especially the data-augmentation strategy proposed here, will be of great practical importance as the experimental effort it requires is generally affordable by an ordinary biochemical research group and can be applied on most protein.

Method

Details of model architecture

Local encoder

Residue interdependencies are crucial to evaluate if a mutation is acceptable. Several models, including ESM-MSA-1b [37], DeepSequence [14], EVE [38] and the Potts model [27], such as EVmutation [16] and ECNet [39], utilize multiple sequence alignment (MSA) to dig the constraints of evolutionary process in the residues level. In the present work, we use Potts model to establish the local encoder. This method first searches for the homologous sequences and builds MSA of the given protein with HHsuite [40]. After that, a statistical model is used to identify the evolutionary couplings by learning a generative model of the MSA of homologous sequences using a Markov random field. In the model, the probability of each sequence depends on an energy function, which is defined as the sum of single-site constraints e_i and all pairwise coupling constraints e_{ij}:

$$E(x) = \sum_{i} \boldsymbol{e}_i(x_i) + \sum_{i \neq j} \boldsymbol{e}_{ij}\left(x_i, x_j\right) \tag{1}$$

where i and j are position indices along the sequence. The i-th amino acid x_i is encoded by a vector, in which elements are set to the single-site term $\boldsymbol{e}_i(x_i)$ and pairwise coupling terms $\boldsymbol{e}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $j = 1,\ldots,n$, n is the number of residues in the sequence. These coupling parameters $\boldsymbol{e}_i$ and $\boldsymbol{e}_{ij}$ can be estimated using regularized maximum pseudolikelihood algorithm [41, 42]. As the result, the i-th amino acid x_i in the sequence is represented by a $(L + 1)$-long vector:

$$\mathbf{v}_i = [\mathbf{e}_i(x_i), \mathbf{e}_{i1}(x_i, x_1), \mathbf{e}_{i2}(x_i, x_2), \ldots \mathbf{e}_{iL}(x_i, x_L)] \tag{2}$$

where, the first $\mathbf{e}_i(x_i)$ is single-site constraint and the following pairwise coupling terms $\mathbf{e}_{i1\ldots}\mathbf{e}_{iL}$ were got by mapping the values to the elements of matrix $\mathbf{e}_{ij}(x_i, x_j)$ based on the residue types and positions of i-th (j-th) amino acid. Therefore, the full representation of a protein sequence was obtained by stacking local evolutionary representations for every amino acid, resulting in an $L \times (L+1)$ matrix. Since the length of the local evolutionary representation of each amino acid is close to the length of the sequence, the $(L + 1)$-long vector would be transformed into a new vector with fixed length d_l (in our local encoder, d_l=128) through a fully connected layer to avoid the overfitting issue. Sequence of protein would also pass a Bi-LSTM layer and be transformed into an $L \times d_l$ matrix for random initialization. By concatenating two matrices above, we obtain the output of local encoder $\boldsymbol{e}' =< \boldsymbol{e}'_1, \boldsymbol{e}'_2, \ldots \boldsymbol{e}'_L >$, whose size is $L \times 2d_l$.

Global encoder

Recently, the large scale pre-trained models have been successfully applied in diverse tasks for inferring protein structure or function based on sequence information. Such as prediction of secondary structure, contact prediction and prediction of mutational effects. Thus, we take a pre-trained protein language model as the global encoder which is responsible to extract biochemical properties and evolution information of the protein sequences. There are some effective language models such as UniRep [12], TAPE [43], ESM-1v [44], ESM-1b [37], ProteinBERT [11] etc. We test these language models on our validation datasets, and results show that ESM-1b performs better than others. Therefore, we chose to use ESM-1b as the global encoder. The model is a bert-based [45] context-aware language model for protein, trained on the protein sequence dataset of UniRef 50 (86 billion amino acids across 250 million protein sequences). Due to its ability to represent the biological properties and evolutionary diversity of proteins, we utilize this model as our global encoder to encode the evolutionary protein sequence. Formally, given a protein sequence $\boldsymbol{x} =< \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L > \in \boldsymbol{L}^{\boldsymbol{N}}$ as input, where $\boldsymbol{x}_i$ is the one-hot representation of $\boldsymbol{i}_{th}$ amino acids in the evolutionary sequence, $\boldsymbol{L}$ is the length of the sequence, and $\boldsymbol{N}$ is the size of amino acids alphabet. The global encoder first encodes each amino acid and its context to $\boldsymbol{g} =< \boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_L >$, where $\boldsymbol{g}_i \in \boldsymbol{R}^{\boldsymbol{n}}$, (in ESM-1b, $n = 1420$). Then $\boldsymbol{g}_i$ is projected to $\boldsymbol{g}_i^{'}$ of a hidden space $\boldsymbol{R}^{\boldsymbol{h}}$ with a lower dimension (in our default model configuration, $h = 256$), $\boldsymbol{g}_i^{'} = \boldsymbol{W}_{\boldsymbol{G}} \boldsymbol{g}_i + \boldsymbol{b}$, where $\boldsymbol{W}_{\boldsymbol{G}} \in \boldsymbol{R}^{\boldsymbol{n} \times \boldsymbol{h}}$ is a learnable affine transform parameter matrix and $\boldsymbol{b} \in \boldsymbol{R}^{\boldsymbol{h}}$ is the bias. The output of global encoder is $\boldsymbol{g}^{'} =< \boldsymbol{g}_1^{'}, \boldsymbol{g}_2^{'}, \ldots \boldsymbol{g}_L^{'} > \in \boldsymbol{R}^{\boldsymbol{L} \times \boldsymbol{h}}$. We integrate the ESM-1b architecture into our model i.e.; we update the parameters of ESM-1b dynamically during the training process.

Structure module

Structure module utilizes the microenvironmental information to guide the fitness prediction. In this part, we use the ESM-IF1 model [23] to generate the scores of mutant sequences, which evaluate their ability to be folded to the wildtype structure of the given protein. Higher scores mean these mutations are more favorable than others. Specifically, all possible single mutants at each position of a sequence would obtain the corresponding scores. The prediction sequence distribution is an $(L \times 20)$ matrix. Then we calculated the cross-entropy at each position of the sequence between the matrix above and one-hot encoding matrix of mutant sequence. After passing the results through a SoftMax function, we obtained an $(L \times 1)$ output vector, which is the reconstruction perplexities $\boldsymbol{p}^{'} =< p_1^{'}, p_2^{'}, \ldots p_L^{'} >$ align the evolutionary sequence. In the present work, we do not directly encode distance map or the 3D coordinate of mutated protein. Since before that encoding process, we need to fold every specific mutant from their sequences, which will lead to unaffordable computational cost and is unpractical for the task of fitness prediction.

Intra-Attention

The outputs of local encoder and global encoder are embedding vectors, aligning all positions of input sequence. We utilize intra-attention mechanism to compress the whole embeddings to a context vector. The inputs of attention layer are: (1) the global representations $\boldsymbol{g}^{'} =< \boldsymbol{g}_1^{'}, \boldsymbol{g}_2^{'}, \ldots \boldsymbol{g}_L^{'} >$ (2) the local representations $\boldsymbol{e}^{'} =< \boldsymbol{e}_1^{'}, \boldsymbol{e}_2^{'}, \ldots \boldsymbol{e}_L^{'} >$ (3) the reconstruction perplexities $\boldsymbol{p}^{'} =< p_1, p_2, \ldots p_L >$. Firstly, the local representations and global representations are normalized by layer normalization [46] over the length dimension respectively for stable training. That is, $\boldsymbol{g}^{'} = \boldsymbol{LayerNorm}(\boldsymbol{g}^{'})$ and $\boldsymbol{e}^{'} = \boldsymbol{LayerNorm}\left(\boldsymbol{e}^{'}\right)$. Secondly, the normalized global representations and local representations are concatenated to joint-representations $\boldsymbol{r} =< \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots \boldsymbol{r}_L >$, where $\boldsymbol{r}_i = \left[\boldsymbol{g}_i^{'}; \boldsymbol{r}_i^{'}\right] \in \boldsymbol{R}^{2\boldsymbol{h}}$. Then we use an dot attention layer to compute the sequence attention weights $a =< a_1, a_2, \ldots, a_L > \in R^L$, where $a_i \in R$ is the attention weight on the i_{th} position, $a_i = \frac{\exp(r_i \bullet W_a r_i)}{\sum_{k=1}^{n} \exp(r_k \bullet W_a r_k)}$, $W_a \in R^{h \times 1}$ is the learnable parameter. Besides the sequence attention weights, there is structure attention weights called structure attention $s =< s_1, s_2, \ldots, s_L > \in R^L$, which are calculated by reconstruction perplexities, $s_i = \frac{\exp(p_i^{'})}{\sum_{k=1}^{n} \exp\left(p_k^{'}\right)}$. We use the average of sequence attention and structure attention as the final combined attention weights, that is $w =< w_1, w_2, \ldots, w_L >$, where $w_i = \frac{a_i + s_i}{2}$. According to the combined attention weights, we get the context vector $\boldsymbol{c} = \sum_{i=1}^{L} w_i \boldsymbol{r}_i$ as the embedding vector of the entire sequence.

Output layer

The input of output layer is the context vector $\boldsymbol{c}$ from the output of attention aggregator, and an evolutionary score d from the unsupervised model [23]. While the evolutionary score may not be trusted in many cases, we use a dynamic weight to take the score into account. The context vector $\boldsymbol{c}$ was firstly transformed to a hidden vector $\boldsymbol{h}$, where $\boldsymbol{h} = ReLU(W_h c + b)$, W_h and b are learnable parameters, and **ReLU** [47] is the activation function. Then, the hidden vector $\boldsymbol{h}$ is used to calculate the weight $p \in (0, 1)$ on d: $p = Sigmoid(W_p[\boldsymbol{h}; d])$. The scale of $\boldsymbol{p}$ quantifies how much should the model trust the score from the zero-shot model. At last, we use a linear layer to compute a fitness score $y_q \in R$ according to the hidden vector $\boldsymbol{h}$ directly, where $y_q = W_q h + b$. The output of our model, i.e., the prediction fitness $y \in R$ is computed as:

$$y = (1 - p) \times y_p + p \times y_q \tag{3}$$

We utilize the mean square error (MSE) as the loss function to update model parameters during back-propagation:

$$loss = \frac{1}{N} \sum_{i=1}^{N} \left(t_i - y_i\right)^2 \tag{4}$$

where N is the number of samples in a mini-batch, t_i is the target fitness and y_i is the output fitness.

Dataset and experimental settings

Benchmark dataset collection

We first collected 20 multiple deep mutational scanning datasets from Ref [14]. Most of them only contain the fitness data of single-site mutants, while one of them (RRM) [31] also provides data of high-order mutants. The fitness data measured in these datasets include enzyme function, growth rate, peptide binding, viral replication and protein stability. We also collected the mutant data of the WW domain of human Yap1, GB1 domain of protein G in *Streptococcus sp. group G* and FOS-JUN heterodimer from Ref [48], and the prion-like domain of TDP-43 from Ref [49] to evaluate the ability of our model to predict the effect of double-sites mutant by learning from the data of single-site mutant. Besides, the ability to predict the fitness of higher order mutants (larger than 2) is tested in the dataset from Ref [29]. This study analyzed the local fitness landscape of the green fluorescent protein from *Aequorea victoria* (avGFP) by measuring the native function (fluorescence) of tens of thousands of derivative genotypes of avGFP. The detailed information on these datasets are provided in Additional file 1: Table S4.

Prediction of single-site mutation effects

We compared our model to ECNet, ESM-1b, ESM-1v and MSA transformer model on the DMS datasets. Since there is no public benchmark test set for mutant prediction task, we have to split it by ourselves. Obtaining a fair external train-test splitting for model comparison on the single-site mutant dataset can be difficult. Because the dataset splitting may affect the results of model comparison. To ensure fairness of predicting the fitness on single-site mutants, we randomly split a given dataset into five folds by randomized shuffling and splitting. Our model, SESNet, and other supervised models (ECNet and ESM-1b) are trained and evaluated for five times on different folds splitting. In the *i*-th iteration, the fold-*i* is used as the test set while the remaining four folds are used for training and validation. Splitting the remaining dataset into a training and validation set involves a trade-off between fairness and computational cost. Internal five-fold cross-validation can provide a fair comparison, but it will also increase huge computational cost. Therefore, we only perform a simple random strategy to split the remaining four folds of dataset into training and validation in a 7:1 ratio. The training set is utilized to train the models for *N* epochs at most. And the validation set is used to avoid overfitting through the early-stopping mechanism. The error bars of each model in Fig. 2a are the standard deviations of the five-time testing results (spearman correlation).

Prediction of high-order mutation effects

We evaluated the performance for predicting the fitness of high-order mutants by the model trained on low-order mutants. Here, we used the data of high-order mutants as an external test set and did five-cross validation of the low-order mutant data. Briefly, we randomly split the data of low-order mutants into five folds, and then picked one fold as validation set and the remaining four folds as training set. This process was repeated five times and each fold of data was employed once as the validation set. The model that performed best in the validation set was tested on the high-order mutants.

Data-augmentation strategy

The data augmentation was conducted by pre-training our model on the results predicted by the unsupervised model. To be specific, we first built a mutant library, which contains all the single-site mutants and 30,000 double-site mutants randomly selected from tens of millions of saturated double-site mutants. Then, we used ESM-IF1 (or ProGen2) to score all these sequences. Those sequence-score data were used to pre-train our model. While we used 90% of the data as training test, 10% as validation set. At first, our model was randomly initialized except the global encoder (ESM-1b module). Then the normalized mutant library was used to train the model for 10 epochs using the Adam optimizer whose learning rate is 5e-4. All of the parameters including global encoder were trainable during this process, and the hidden size, batch size, dropout and warmup steps were consistent with the hyper-parameter configuration for the multiple-sites dataset shown in Additional file 1: Table S7. In the fine-tuning stage, the pretrained model was finetuned on a small subset of the experimental dataset, also allowing all model parameters to be trainable. The learning rate was set to 5e-4. In the evaluation stage, we used the finetuned model to predict the fitness for high-order mutants and compute the spearman correlation between the experimental fitness (ground truth) and predicted fitness.

Training details

SESNet was trained using the Adam optimizer with weight decay. Hyperparameters of the model were tuned with a local grid search on two representative datasets, GFP for multi-sites dataset and RRM for single-site dataset. The searched optimal hyperparameters configuration are applied in other datasets. We tested the hidden size of [128, 256, 512], learning rate of [1e-3, 5e-4, 1e-4, 5e-5, 1e-5], and dropout of [0.1, 0.2, 0.4]. Additional file 1: Table S7 in SI shows the details of the hyperparameter configurations. All experiments are conducted on a GPU server with 10 RTX 3090 GPUs (24 GB VRAM) and 2 Intel Gold 6226R CPUs with 2 TB RAM.

Model contrast

The source code of ECNet model for contrast is downloaded from the GitHub website (https://github.com/luoyunan/ECNet) provided by Ref [17]. The ESM-1b model is also reproduced in our local computers with architecture that is described in their publication [6]. The code of ESM-IF1, ESM-1v and MSA transformer (ESM-MSA-1b) are got from the GitHub website of Facebook research (https://github.com/facebookresearch/esm). For each assay, all experiments of three different models are performed in the same dataset.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-023-00688-x.

Additional file 1: Table S1. Spearman correlation in Figure 2A. Comparison to other supervised and unsupervised models for fitness prediction on the single-site mutants of 20 datasets. The one marked in bold denotes the best performance. **Table S2.** Spearman correlation in Figure 2B. Fitness prediction of double-site mutants by unsupervised models (ESM-IF1, ESM-1v and MSA transformer), or supervised models (ECNet and ESM-1b) and SESNet trained on the data of single-site mutants. The one marked in bold denotes the best performance. **Table S3.** Spearman correlation in Figure 2C. Prediction of quadruple variants of avGFP using models trained on single, double, triple-site mutants and all the above three. **Table S4.** Detailed information on the proteins listed in the dataset of Tables 1-3. The protein fitness classification and the number of sites being mutated of each protein. **Table S5.** Ablation study results. Ablation study was performed in the testing when we removed each of the three modules in the integrated model. The average spearman correlation of all datasets shows that model including all the three components are the most accurate, and all three parts contribute positively to the performance of the integrated model, with the global encoder contributing the most. **Table S6.** Ablation study of the pre-trained model tested on GFP datasets. The spearman correlation was predicted by our models which is pre-trained on single-site and numerous double-sites variants generated by the unsupervised model ESM-IF1. **Table S7.** Hyperparameter configurations for different dataset. **Figure S1.** Attention score of sites on the wildtype sequence. Attention scores of sites generated by SESNet (A) and the model without the structure module (B) trained on the 1084 single-site mutants of the dataset of GFP. We picked up the top 20 attention-score AA sites predicted by SESNet with and without structure module, respectively. When the structural module is present, there are five sites (marked by the blue ellipse in the subgraph A) identified by our model accords with the key AA sites discovered by experiments (mentioned in main text). However, this number is reduced to three when we remove the structural module from the model (marked by the blue ellipse in the subgraph B). **Figure S2.** Attention score of sites on the wildtype sequence. Attention scores of sites generated by SESNet (A) and the model when removing the structure encoder (B) trained on the 1064 single-site mutants of the dataset of RRM. We picked up the top-20 attention-score AA sites predicted by SESNet model with and without the structure module, respectively. When the structural module is present, there are 11 sites (marked with blue ellipse in the subgraph A) predicted by the model accord with the key AA sites discovered by experiments (mentioned in main text). In contrast, when removing the structure module, 3 of the predicted top-20 AA sites accord with the experimentally discovered (see the subgraph B). **Figure S3.** Results of models pre-trained on the dataset generated by the unsupervised model ProGen2 (ref), and then fine-tuned on different number of experimental data points. A-D: The spearman correlation of fitness prediction on multiple sites (2-8 sites) mutants by finetuning on 40, 100, 400, 1084 experimental single-site variants from dataset of GFP. Here, the red and blue bars represent the results of the model with and without pre-training, respectively. And the green bars correspond to the results of the unsupervised model ProGen2 as a control. **Figure S4.** Results of models trained on different number of single-site experimental variants. A-C: The spearman correlation of fitness prediction on single-site mutants by finetuning on 20, 40, 100 single-site variants from different datasets. Where the blue and red symbols represent the results of the pre-trained model and the original model without pretraining, respectively. **Figure S5.** Variant sequence representations of trained and untrained SESNet by the experimental data. Each point represents a variant, where the positive and negative variants are colored as red and blue, respectively. The models were trained on single-site mutants from the dataset of GFP and RRM. Here a red point represents a mutant whose experimental fitness value is higher than that of the wild type, while the blue point gives mutant whose experimental fitness value is lower than the wild type. As can be seen in A and B, after training by part of the experimental data set, the positive and negative mutants can be separated into different spaces. In contrast, those representations from untrained model with random parameter initialization (C and D) do not reflect any clear separation between the positive and negative mutants as expected. This comparison shows that our model can learn to distinguish functional fitness of mutants into a latent representation space with supervised training. **Figure S6.** Representations of variants in different training ways. A: The representations from the pre-trained models without fine-tuning by any experimental data. B: the representations from the pre-trained models, which is further finetuned on 40 single-site experimental mutants. C: the representations from the model directly trained on 40 single-site experimental mutants without pre-training.

Author contributions

LH and PT designed this project, PT and ML proposed this model, ML, LQ, YX, YGW and GF implemented the method, performed the calculations. All the authors read and approved the final manuscript.

Funding

This work was financially supported by the Natural Science Foundation of China (Grant No. 12104295, 11974239, 31630002, 61872094, 61832019), the Innovation Program of Shanghai Municipal Education Commission, and Shanghai Jiao Tong university Multidisciplinary research fund of medicine and engineering YG 2016QN13. The main work was supported by Shanghai Artificial Intelligence Laboratory, and the computing hardware resource was also mainly supported by them. While some testing were supported by the Center for High Performance Computing at Shanghai Jiao Tong University.

Availability of data and materials

Source code for SESNet and all the datasets used in the present work can be found in the supplemental materials. Where the original sources of datasets have been declaimed and cited in the main text.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 3 November 2022 Accepted: 23 January 2023
Published online: 03 February 2023

References

1. Arnold FH (1998) Design by directed evolution. Acc Chem Res 31(3):125–131
2. Wu Z et al (2019) Machine learning-assisted directed protein evolution with combinatorial libraries. Proc Natl Acad Sci 116(18):8852–8858
3. Cui Y et al (2021) Computational redesign of a PETase for plastic biodegradation under ambient condition by the GRAPE strategy. ACS Catal 11(3):1340–1350
4. Hie B et al (2021) Learning the language of viral evolution and escape. Science 371(6526):284–288
5. Hie BL, Yang KK, Kim PS (2022) Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins. Cell Syst. https://doi.org/10.1016/j.cels.2022.01.003
6. Rives A et al (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci 118(15):e2016239118
7. Rao R et al (2019) Evaluating protein transfer learning with TAPE. Adv Neural Inf Process Syst 32:9689
8. Nijkamp E et al (2022) ProGen2: exploring the boundaries of protein language models. arXiv Preprint. https://doi.org/10.48550/arXiv.2206.13517
9. Meier J et al (2021) Language models enable zero-shot prediction of the effects of mutations on protein function. bioRxiv. https://doi.org/10.1101/2021.07.09.450648
10. Elnaggar A et al (2020) ProtTrans: towards cracking the language of Life's code through self-supervised deep learning and high performance computing. arXiv Preprint. https://doi.org/10.48550/arXiv.2007.06225
11. Brandes N et al (2022) ProteinBERT: a universal deep-learning model of protein sequence and function. Bioinformatics 38(8):2102–2110
12. Alley EC et al (2019) Unified rational protein engineering with sequence-based deep representation learning. Nat Methods 16(12):1315–1322
13. Russ WP et al (2020) An evolution-based model for designing chorismate mutase enzymes. Science 369(6502):440–445
14. Riesselman AJ, Ingraham JB, Marks DS (2018) Deep generative models of genetic variation capture the effects of mutations. Nat Methods 15(10):816–822
15. Rao RM et al (2021) MSA transformer, in proceedings of the 38th international conference on machine learning. In: Marina M, Tong Z (eds). PMLR: proceedings of machine learning research. p. 8844-8856.
16. Hopf TA et al (2017) Mutation effects predicted from sequence co-variation. Nat Biotechnol 35(2):128–135
17. Luo Y et al (2021) ECNet is an evolutionary context-integrated deep learning framework for protein engineering. Nat Commun 12(1):5743
18. Biswas S et al (2021) Low-N protein engineering with data-efficient deep learning. Nat Methods 18(4):389–396
19. Jumper J et al (2021) Highly accurate protein structure prediction with AlphaFold. Nature 596(7873):583–589
20. Baek M et al (2021) Accurate prediction of protein structures and interactions using a three-track neural network. Science 373(6557):871
21. Varadi M et al (2022) AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res 50(D1):D439–D444
22. Zhang Z et al (2022) Protein representation learning by geometric structure pretraining. arXiv Preprint. https://doi.org/10.48550/arXiv.2203.06125
23. Hsu C et al (2022) Learning inverse folding from millions of predicted structures. bioRxiv. https://doi.org/10.1101/2022.04.10.487779
24. Lu H et al (2022) Machine learning-aided engineering of hydrolases for PET depolymerization. Nature 604(7907):662–667
25. Wang Z et al (2022) LM-GVP: an extensible sequence and structure informed deep learning framework for protein property prediction. Sci Rep 12(1):6832
26. Gelman S et al (2021) Neural networks to learn protein sequence–function relationships from deep mutational scanning data. Proc Natl Acad Sci 118(48):e2104878118
27. Ekeberg M et al (2013) Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. Phys Rev E 87(1):012707
28. Shroff R et al (2020) Discovery of novel gain-of-function mutations guided by structure-based deep learning. ACS Synth Biol 9(11):2927–2935
29. Sarkisyan KS et al (2016) Local fitness landscape of the green fluorescent protein. Nature 533(7603):397–401
30. Zimmer M (2002) Green fluorescent protein (GFP): applications, structure, and related photophysical behavior. Chem Rev 102(3):759–782
31. Melamed D et al (2013) Deep mutational scanning of an RRM domain of the Saccharomyces cerevisiae poly (A)-binding protein. RNA 19(11):1537–1551
32. Minot M, Reddy ST (2022) Nucleotide augmentation for machine learning-guided protein engineering. bioRxiv. https://doi.org/10.1101/2022.03.08.483422
33. Hsu C et al (2022) Learning protein fitness models from evolutionary and assay-labeled data. Nat Biotechnol 40(7):1114–1122
34. Aakre CD et al (2015) Evolving new protein-protein interaction specificity through promiscuous intermediates. Cell 163(3):594–606
35. Sinai S et al (2021) Generative AAV capsid diversification by latent interpolation. bioRxiv. https://doi.org/10.1101/2021.04.16.440236
36. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(11):2579
37. Rao R et al (2021) Msa transformer. In international conference on machine learning. PMLR.
38. Frazer J et al (2021) Disease variant prediction with deep generative models of evolutionary data. Nature 599(7883):91–95
39. Luo Y et al (2021) ECNet is an evolutionary context-integrated deep learning framework for protein engineering. Nat Commun 12(1):1–14
40. Steinegger M et al (2019) HH-suite3 for fast remote homology detection and deep protein annotation. BMC Bioinformatics 20(1):1–15
41. Hopf TA et al (2014) Sequence co-evolution gives 3D contacts and structures of protein complexes. elife 3:e03430
42. Seemayer S, Gruber M, Söding J (2014) CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. Bioinformatics 30(21):3128–3130
43. Rao R et al (2019) Evaluating protein transfer learning with TAPE. Advances in neural information processing systems. 32.
44. Meier J et al (2021) Language models enable zero-shot prediction of the effects of mutations on protein function. Adv Neural Inf Process Syst 34:29287–29303
45. Devlin J et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv Preprint. https://doi.org/10.48550/arXiv.1810.04805
46. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv Preprint. https://doi.org/10.48550/arXiv.1607.06450
47. Fukushima K (1975) Cognitron: a self-organizing multilayered neural network. Biol Cybern 20(3):121–136
48. Rollins NJ et al (2019) Inferring protein 3D structure from deep mutation scans. Nat Genet 51(7):1170–1176
49. Bolognesi B et al (2019) The mutational landscape of a prion-like domain. Nat Commun 10(1):1–12

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.