Self-Supervised Representation Learning of Protein Tertiary Structures (PtsRep) and Its Implications for Protein Engineering

Junwen Luo[1]†, Yi Cai[1]†, Jialin Wu[1], Hongmin Cai[2], Xiaofeng Yang[1]* and Zhanglin Lin[1]*

[1]School of Biology and Biological Engineering, [2]School of Computer Science and Engineering, South China University of Technology, University Park, Guangzhou, Guangdong 510006, China.

* To whom correspondence should be addressed:

School of Biology and Biological Engineering, South China University of Technology, 382 East Outer Loop Road, University Park, Guangzhou, Guangdong 510006, China; Tel: +86 (20) 3938-0680; Fax: +86 (20) 3938-0601; Email: zhanglinlin@scut.edu.cn (Z.L.); biyangxf@scut.edu.cn (X.Y.).

† These authors contributed equally to this work.

Abstract

In recent years, deep learning has been increasingly used to decipher the relationships among protein sequence, structure, and function. Thus far these applications of deep learning have been mostly based on primary sequence information, while the vast amount of tertiary structure information remains untapped. In this study, we devised a self-supervised representation learning framework (PtsRep) to extract the fundamental features of unlabeled protein tertiary structures deposited in the PDB, a total of 35,568 structures. The learned embeddings were challenged with two commonly recognized protein engineering tasks: the prediction of protein stability and prediction of the fluorescence brightness of green fluorescent protein (GFP) variants, with training datasets of 16,431 and 26,198 proteins or variants, respectively. On both tasks, PtsRep outperformed the two benchmark methods UniRep and TAPE-BERT, which were pre-trained on two much larger sets of data of 24 and 32 million protein sequences, respectively. Protein clustering analyses demonstrated that PtsRep can capture the structural signatures of proteins. Further testing of the GFP dataset revealed two important implications for protein engineering: (1) a reduced and experimentally manageable training dataset (20%, or 5,239 variants) yielded a satisfactory prediction performance for PtsRep, achieving a recall rate of 70% for the top 26 brightest variants with 795 variants in the testing dataset retrieved; (2) counter-intuitively, when only the bright variants were used for training, the performances of PtsRep and the benchmarks not only did not worsen but they actually slightly improved. This study provides a new avenue for learning and exploring general protein structural representations for protein engineering.

Introduction

Self-supervised learning is a powerful method for learning general representations from unlabeled samples[1-5]. In natural language processing (NLP), this takes the forms of word2vec (continuous skip-gram model, and continuous bag-of-words model)[1], next-token prediction[2], and masked-token prediction[3]. Recently NLP-based techniques have been applied to representation learning of protein sequences[6-10]. Two representative examples are UniRep[10], which is based on next-token prediction and is trained on 24 million protein sequences, and a BERT[11] model (hereinafter referred to as TAPE-BERT), which is based on masked-token prediction and is trained on 32 million protein sequences. Through transfer learning, both methods showed good performance for the prediction of protein stability landscape, and green fluorescence protein (GFP) activity landscape. For these tasks, two datasets containing about 69,000 protein sequences[12] and 50,000 GFP variants[13], respectively, were used. Gaining the ability to predict protein engineering outcomes is a major goal for biotechnology, which would significantly expand the uses of proteins and enzymes for pharmaceutical, industrial, and agricultural purposes. However, predictions based on traditional rational design approaches have not yet reached the level of accuracy required for routine practice[12,14-16], and the directed protein evolution approach often demands iterative processes and high throughput screening assays[17,18]. Artificial intelligence approaches such as machine learning[16], UniRep[10], and TAPE-BERT[11] have great potentials to accelerate the rational design of proteins.

The methods for protein representation learning developed thus far have mostly relied on protein sequence information, whereas the vast amount of protein tertiary structure information available in the protein data bank (PDB)[19] remains unused, and the recent release of the breakthrough algorithm AlphaFold 2[20] will likely make protein tertiary structures more readily available. How to utilize protein structural information in deep learning remains a fundamental open question whose answers will strongly benefit protein design and engineering.

In this work, we present a self-supervised learning model designed to extract embedded representations of protein tertiary structures (PtsRep) (Fig. 1), using KNR (*K* nearest residues) as the input format for the protein structures (Fig. 1A). Considering that in proteins the properties of a residue are affected by the surrounding residues[21], we adopted the bidirectional language model[22], an advanced next-token algorithm, to predict multiple neighboring residues on both forward and backward directions of a given protein. In order to increase the stringency of the prediction, the residues immediately adjacent to any given residue were omitted from the prediction. The learned representations summarized the protein tertiary structures into fixed-length vectors of 768 dimensions. These vectors were then used for two tasks: (1) prediction of protein stability, and (2) prediction of GFP fluorescence brightness (Fig. 1C). The prediction performances of PtsRep were compared to those of the benchmarks UniRep and TAPE-BERT. In addition, the advantages of PtsRep over the benchmarks was evaluated in terms of "testing budget" required to identify a target variant, which has critical implications in the application of these machine learning methods to answer practical protein engineering questions.

Methods

1. Representation learning network

The learning network architecture is shown in Fig. 1 and Supplementary Fig 1. The bidirectional language model [2,22] was used to predict the two contiguous residues beside any given residues in both directions, with a $\boldsymbol{d}$ number of adjacent residues omitted. Specifically, we defined $R(\boldsymbol{t}, \boldsymbol{d})$, the sum of log likelihood in both the forward and backward directions, as follows:

$$\boldsymbol{R(t,d) = log\ p(r_t \mid r_1, r_2, \ldots, r_{t-d}; \theta_r, \theta_{LSTM}^{\rightarrow}, \theta_s) + log\ p(r_t \mid r_{t+d}, r_{t+d+1}, \ldots, r_l; \theta_r, \theta_{LSTM}^{\leftarrow}, \theta_s)}$$

Where, $\boldsymbol{r}$ refers to a residue (token) in an input protein tertiary structure, $\boldsymbol{l}$ as the length of a protein, $\boldsymbol{t}$ as the position of the residue in the protein sequence. $\boldsymbol{\theta_x}$ as the K nearest residue representations of a protein, $\boldsymbol{\theta_{LSTM}}$ as the LSTM layer, $\boldsymbol{\theta_s}$ as the Softmax layer, and $\boldsymbol{p}$ as the probability[23] of the predicted residue corresponding to the actual residue.

We then calculated $R(\boldsymbol{t}, \boldsymbol{d})$ for every $\boldsymbol{t} \in \{\boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{l}\}$ and $\boldsymbol{d} \in \{\boldsymbol{d}, \boldsymbol{d}+\boldsymbol{1}\}$, where the instances $\boldsymbol{t}-\boldsymbol{d} < 0$ and $\boldsymbol{t}+\boldsymbol{d} > \boldsymbol{l}$ were not calculated. The final loss was the average of the calculated $\boldsymbol{Loss}$ for all positions of $\boldsymbol{t}$:

$$\boldsymbol{Loss(l,d) = \frac{1}{4l-4d-2} \sum_{t=1}^{l} \sum_{w=d}^{d+1} -R(t,w)}$$

Considering the large amount of protein structural data used in this study, we adopted a double-layer bi-LSTM, and chose 768 as the dimension of embeddings, as in TAPE-BERT[11]. Convergence was defined as no improvement in the validation loss for 5 epochs. The best model was trained for 1.8 million weight updates corresponding to ~50 epochs. Because the length of each protein sequence was different, we used a batch size of 1. All models were trained with the Adam

optimizer[24]. Where indicated, Swish activation[25] and layer normalization[26] were applied, and to reduce the risk of overfitting, a dropout rate of 10% was applied to the output layer (supplementary Fig. 1).

2. Benchmark representations

We used UniRep, which is an mLSTM model trained on about 24 million protein sequences, as a benchmark representation[10]. We downloaded the trained weights and obtained the representations using the code in https://github.com/churchlab/UniRep. We chose the 1900-dimensional UniRep representation, which was reported to perform best among the different dimensions[10]. We also used TAPE-BERT, which is trained on about 32 million protein sequences, as an additional benchmark representation[3]. We downloaded the trained weights and obtained the representations using the code in https://github.com/songlab-cal/tape. Furthermore, we used one-hot encoding and KNR as baseline representations.

3. Downstream task evaluation

To better compare the performances with the benchmarks, we used the same downstream architecture with attention mechanism and multilayer perceptron, as described in TAPE-BERT[11]. The embeddings of protein tertiary structures extracted from the pretrained network were used without end-to-end optimization. For these tasks, we reported the Pearson's γ (product-moment correlation coefficient), Spearman's ρ (rank correlation coefficient), accuracy (ACC) and mean-

square error (MSE) from different representations. Five-fold random cross validation was used, and the 10 models with the highest Spearman's ρ on the validation dataset of each fold were selected as the test mode. Convergence was defined as an improvement lower than 0.002 for the Spearman's ρ after 20 epochs during the validation.

4. Datasets

1) Self-supervised learning dataset.

We used ProteinNet[27] as the training and validation sets for the self-supervised learning model. We first collected the 90% thinning version of ProteinNet12[27], which included 49,600 protein chains with tertiary structures. Then we excluded the protein chains (1) with missing Cα coordinates, or (2) with sequences longer than 700 aa or shorter than 30 aa. The resulting dataset had 35,568 protein chains. We used 95% of the data as the training dataset, and 5% as the validation dataset (Supplementary Table 1).

2) Stability landscape prediction dataset

We used the dataset from Rocklin *et al*[12]. This set includes the sequences and the chymotrypsin stability scores of 69,034 proteins (17,773 *de novo* designed proteins, 10,674 variants of these designed proteins, 1,193 natural proteins, 2,423 variants of these natural proteins, 24,900 scrambled versions of the proteins, and 12,071 control inactivated sequences where a buried aspartate residue was inserted). Among these proteins or variants, we used a total of 16,431 proteins or variants with structural information (16,159 *de novo* designed mini proteins and 272

natural proteins) for training and validation, and 12,851 point variants of 14 de novo designed proteins and 3 natural proteins with structural information for testing[12]. For training and validation, we used an 80%/20% splitting strategy (Supplementary Table 1).

3) Fluorescence landscape prediction

We used the dataset from Sarkisyan *et al*[13], which contains more than 50,000 variants of green fluorescent protein from *Aequorea victoria*. The number of amino acid substitutions in the variants ranged from 1 to 15[14]. Following a previous study, we used the variants with 1~3 substitutions (a total of 26,198 variants) for training and validation, and the variants with 4 or more substitutions (a total of 25,517 variants) for testing[20]. For training and validation, we used an 80%/20% splitting strategy (Supplementary Table 1).

Results

1. Network architecture for protein tertiary structure representation learning (PtsRep)

PtsRep comprises of three modules as shown in Fig. 1. In the input module (Fig. 1A), to feed the protein tertiary structural information, we adopted an algorithm[28] that enables each residue to be represented by a series of properties (i.e., bulkiness[29], hydrophobicity[30], flexibility[31], relative spatial distance, relative sequential residue distance, and spatial position based on spherical coordinate system) of its *K* nearest residues (KNR) in the Euclidean space[28]. The parameter *K* was set at 15 after initial optimization based on the performance criteria for the downstream tasks (supplementary Table 2). A dataset of 35,568 protein chains was selected from PDB, 95% of which was used for training and 5% for validation. In the training module (Fig. 1B), a bidirectional language model[2] was adopted to predict the four nearest noncontiguous residues on both the forward and backward directions of a given amino acid $\boldsymbol{t}$, with the immediate adjacent residue omitted (*i.e.*, positions $\boldsymbol{t-3}$, $\boldsymbol{t-2}$, $\boldsymbol{t+2}$, $\boldsymbol{t+3}$). The omission of these two residues was again chosen after optimization based on the performance criteria for the downstream tasks (see supplementary Table 3, and Methods). The network was iterated to maximize the prediction accuracy. This yielded a top model, which summarized the protein tertiary structures into $\boldsymbol{l} \times 768$-dimensional vectors for any given protein sequence of length $\boldsymbol{l}$. Finally, this trained or pre-trained (with respect to the downstream tasks) PtsRep model was applied to two protein landscape prediction tasks, protein stability and green fluorescence protein brightness, through a downstream network (se Methods)[11], as shown in Fig. 1C.

171 **2. PtsRep significantly outperformed benchmarks on protein stability prediction**

172 Protein stability is a key objective for protein engineering. In this study, we used the dataset
173 from Rocklin *et al*[12], which consists of the following four protein topologies:
174 $\alpha\alpha\alpha, \alpha\beta\beta\alpha, \beta\alpha\beta\beta, \beta\beta\alpha\beta\beta$, where α denotes a α-helix and β denotes a β-sheet. It should be noted
175 that the training/validation dataset for the benchmarks UniRep and TAPE-BERT contains 56,180
176 proteins, among which only 16,431 have structural information suitable for PtsRep processing
177 (Supplementary Table 1). Thus, the PtsRep model was trained on only 29.3% of the data used for
178 UniRep and TAPE-BERT. However, all the proteins in the testing dataset for the benchmarks had
179 solved structures, thus the PtsRep model was tested on the same testing dataset used in previous
180 studies for UniRep and TAPE-BERT.

181 As shown in Table 1, we compared Spearman's ρ, ACC and MSE. We found that PtsRep
182 significantly outperformed the best benchmark TAPE-BERT (Spearman's ρ = 0.79 *versus* 0.73),
183 and much more so the baseline KNR (Spearman's ρ = 0.79 *versus* 0.37). PtsRep also generally
184 showed better ACC and MSE, compared with both the benchmarks. Interestingly, the benchmark
185 UniRep performed differently for different protein topologies (e.g., $\alpha\beta\beta\alpha$, $\alpha\alpha\alpha$), whereas PtsRep
186 performed more evenly in this regard, and resulted in a low standard deviation, similar to that of
187 TAPE-BERT (0.11 *versus* 0.08), but significantly better than that of UniRep (0.11 *versus* 0.25)
188 (Supplementary Table 4).

189 **3. PtsRep improved the prediction of the fluorescence brightness of GFP variants**

190 Since the protein stability landscape dataset consists of mainly artificially designed small
191 proteins and only an insufficient number of natural proteins, in further endeavors we focused on
192 the GFP variants dataset of Sarkisyan *et al*[13]. As described in a previous study[11], we allocated the
193 variants with 1-3 substitutions (for a total of 26,198) as the dataset for training and validation, and
194 the variants with four or more substitutions (for a total of 25,517) as the dataset for testing. As
195 shown in Table 1, PtsRep outperformed the best benchmark (*i.e.*, TAPE-BERT) in this task
196 (Spearman's $\rho = 0.70$ *versus* 0.68).

197 Since these statistical metrics are often difficult to correlate with the actual practice of protein
198 engineering, we explored an alternative and more direct measure, the "testing budgets" for target
199 variants[10,11]. We selected the top 0.1% of the GFP variants in the testing dataset in terms of
200 brightness, for a total of 26 variants (with 4 to 15 mutations), and tested the ability of each method
201 to prioritize this set of brightest variants under different testing budgets. As shown in Fig. 2A,
202 PtsRep performed better than the benchmarks UniRep and TAPE-BERT, and significantly better
203 than the baselines. For example, PtsRep achieved a recall rate of 70% for this top 0.1% brightest
204 variants with about 380 variants retrieved, compared with 559 variants for TAPE-BERT, 737
205 variants for UniRep, and 1,609 variants for the baseline KNR, or 1.5-fold, 1.9-fold, and 4.2-fold
206 the experimental burden required for PtsRep, respectively.

207 Subsequently, we tested how the brightest GFP variant fared in the ranking obtained with
208 each method[10,11]. As shown in Fig. 2B, we found that the brightest protein ranked 28th when using

PtsRep, whereas it ranked 71st when using TAPE-BERT, respectively, and 497th for the baseline KNR, corresponding to 2.5-fold and 17.8-fold the experimental cost for PtsRep, respectively.

4. Refinement of PtsRep for the prediction of the fluorescence brightness of GFP variants

We then asked two additional questions: (1) what if only a fraction of the training dataset was used? and (2) what if only the bright mutants were used in the training dataset?

As shown in Supplementary Fig. 2, the use of a subset representing 20% of the whole training and validation dataset (randomly selected, for a total of 5,239 variants) resulted in a fairly satisfactory performance. In particular, PtsRep achieved a recall rate of 70% for the top 26 brightest variants with about 795 variants retrieved, *i.e.*, only 2-fold the experimental burden required when using the whole training dataset. The brightest GFP variant ranked 87th, *i.e.*, 3-fold the experimental burden required when using the whole training dataset. For comparison, TAPE-BERT needed to retrieve 1,347 variants to reach the same recall rate, and the brightest GFP variant ranked 119th.

Very surprisingly, however, when the dark variants were removed from the training dataset (resulting in a total of 21,316 variants), the prediction performances of all methods did not suffer, but rather slightly improved. For example, to achieve a recall rate of 70% for the top 26 brightest variants, PtsRep and TAPE-BERT now needed to retrieve a reduced number of 314 and 432 variants, respectively. Furthermore, the brightest GFP variant ranked 7th and 6th, respectively (Supplementary Fig. 3). When 20% of these bright variants (randomly selected, for a total of 4,263

228 variants) was used, to obtain the same recall rate, PtsRep and TAPE-BERT needed to retrieve 761
229 and 926 variants, respectively, and the brightest GFP variant ranked 84th and 88th, respectively.

230 Lastly, ensemble methods with weighted contributions of multiple models are known to
231 generally improve the performance of machine learning models[32]. In our case, a combination of
232 the three different representation methods PtsRep, TAPE-BERT, and UniRep improved the
233 prediction performance as expected. As shown in supplementary Table 5, for example, to reach a
234 recall rate of 70% for the top 0.1% brightest variants, the PtsRep-TAPE-BERT-UniRep model
235 needed to retrieve 329 variants compared with 380 variants for the PtsRep model alone, and the
236 brightest GFP variant ranked 21st with the combined model, compared with 28th for the PtsRep
237 model alone.

238 **5. PtsRep performed well for protein fold classification**

239 To examine what PtsRep learned from the PDB[19] dataset, we used t-distributed stochastic
240 neighbor embedding (t-SNE)[33] to test its unsupervised clustering ability on a dataset of 15,444
241 semantically related unlabeled proteins, which have been classified in the Structural Classification
242 of Proteins (SCOP)[34] database. We found that PtsRep well separated the five different types of
243 protein structures (Fig. 3, right panel). Using the Davies-Bouldin Index (DBI)[35], a general index
244 for evaluating clustering algorithms, we found that the DBI for PtsRep was as low as 1.3. In
245 contrast, for the baseline representation KNR, which utilized the same amount of structural
246 information but was not trained with PtsRep, the DBI was 11.6 (Fig. 3, left panel). Therefore, the
247 trained PtsRep model improved the clustering by 8.9-fold in this SCOP classification task.

248 We also used t-SNE to test the ability of PtsRep to cluster the GFP variants. As shown in
249 supplementary Fig. 4, the variants with bright fluorescence were more clustered when using
250 PtsRep resulting in a DBI index of 2.5, compared with the more scattered distribution obtained
251 when using KNR, which scored a DBI of 4.6.
252
253

Discussion

In this work, we presented PtsRep, a self-supervised learning method that was designed to learn general protein structure representations from unlabeled protein tertiary structures deposited in the PDB archive. The learned embeddings were applied to the predictions of protein stability and GFP fluorescence using two publicly available datasets. PtsRep showed an outstanding performance on both tasks compared with the protein sequence-based benchmark methods UniRep and TAPE-BERT. Compared to the best benchmark TAPE-BERT, which is trained on 32 million protein sequences (with 96 million parameters), our PtsRep model was trained on only 35,568 protein structures (with 2.5 million parameters), or 0.11% of the data used in TAPE-BERT in terms of protein entries.

While what deep learning learns from data is often considered a black box, we attempted a partial dissection of what PtsRep has learned from the PDB dataset. Given the fact that PtsRep was able to perform much better than KNR, in terms of protein fold classification, as shown in Fig. 3, we suggest that PtsRep has at least understood which protein structures are stable, since unstable and thus inactive protein structures are not likely present in the PDB. We also know from the practice of directed evolution of enzymes that around 30% of randomly mutated protein variants are inactive[36]. This explains, in part, the performance improvement that we saw for PtsRep over UniRep and TAPE-BERT as shown in Fig. 2.

This same reason could be a major contributor to the convergence of the performances of the benchmarks towards that of PtsRep when only bright GFP variants were used for training (Fig. 3),

274 as if in this manner the benchmarks indirectly gained the ability to distinguish stable bright GFP
275 variants as well as PtsRep. Nonetheless, it is worth noting that even in such instance PtsRep was
276 still superior to Unirep and TAPE-BERT, since it was trained on only a very small fraction of the
277 data used for the benchmarks.

278 However, how PtsRep is able to cluster GFP variants better then KNR remains to be
279 understood (supplementary Fig. 2). Since a similar observation was made for TAPE-BERT[11],
280 which is trained on protein sequences alone, it is likely that the better clustering of GFP seen for
281 PtsRep was not only structure-driven, but also partially sequence- and algorithm-driven. Along
282 this line, it is interesting to note that the performance of PtsRep was still much better than the
283 baseline model KNR, and was in fact close to those of UniRep and TAPE-BERT even when the
284 parameter *K* was set at 1 (supplementary Table 2). This can likely be attributed to two aspects of
285 the learning algorithm: 1) the fact that the algorithm predicts not just one "next token," but two
286 continuous residues in both directions, 2) the fact that the algorithm omits the adjacent residues
287 that might be easy to predict given the KNR format, both of which might force the model to better
288 internally represent proteins (supplementary Table 3). Predicting more than two continuous
289 residues, however, failed to further improve the performance of PtsRep for downstream tasks (data
290 not shown).

291 As far as protein engineering is concerned, we have learned two very important lessons from
292 the deep learning of GFP fluorescence landscape: 1) only a few thousand variants are sufficient to
293 train a predictive model, which is experimentally manageable; and 2) the active (*i.e.*, bright)

294 variants provide a better guidance. If the latter holds true for other proteins, this could result in a
295 tremendous saving in terms of experimental cost, as it would not be necessary to characterize
296 inactive variants. Lastly, it is also possible to combine PtsRep (alone or together with TAPE-BERT
297 and UniRep) with traditional methods such as simulated annealing[37] and Bayesian optimization[38],
298 and extend the use of these deep learning strategies for disease diagnosis[10,39,40], and drug design[41,42].
299

Acknowledgments

This work was supported by the National Key R&D Program of China (2018YFA0901000), and the Guangzhou Science and Technology Program key projects (201904020016). We thank Xing Zhang for technical assistance.

References

1 Mikolov T, Chen K, Corrado G *et al.* Efficient estimation of word representations in vector space. *arXiv*, 1301.3781 (2013).

2 Peters M E, Neumann M, Iyyer M *et al.* Deep contextualized word representations. *arXiv*, 1802.05365 (2018).

3 Devlin J, Chang M-W, Lee K *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 1810.04805 (2019).

4 Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image rotations. *arXiv*, 1803.07728 (2018).

5 Goyal P, Mahajan D, Gupta A *et al.* Scaling and Benchmarking Self-Supervised Visual Representation Learning. *International Conference on Computer Vision*, 6400-6409 (2019).

6 Heinzinger M, Elnaggar A, Wang Y *et al.* Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* **20**, 723 (2019).

7 Yang K K, Wu Z, Bedbrook C N *et al.* Learned protein embeddings for machine learning. *Bioinformatics* **34**, 2642-2648 (2018).

8 Rives A, Meier J, Sercu T *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 622803 (2020).

9 Bepler T, Berger B. Learning protein sequence embeddings using information from structure. *arXiv*, 1902.08661 (2019).

10 Alley E C, Khimulya G, Biswas S *et al.* Unified rational protein engineering with sequence-

325 based deep representation learning. *Nature Methods* **16**, 1315-1322 (2019).

326 11 Rao R S, Bhattacharya N, Thomas N *et al.* Evaluating protein transfer Learning with TAPE.

327 *Advances in Neural Information Processing Systems* **32**, 9689-9701 (2019).

328 12 Rocklin G J, Chidyausiku T M, Goreshnik I *et al.* Global analysis of protein folding using

329 massively parallel design, synthesis, and testing. *Science* **357**, 168-174 (2017).

330 13 Sarkisyan K S, Bolotin D A, Meer M V *et al.* Local fitness landscape of the green

331 fluorescent protein. *Nature* **533**, 397-401 (2016).

332 14 Huang P S, Boyken S E, Baker D. The coming of age of de novo protein design. *Nature*

333 **537**, 320-327 (2016).

334 15 Coluzza, Ivan. Computational protein design: a review. *J Phys Condens Matter* **29**, 143001

335 (2017).

336 16 Wu Z, Kan S B J, Lewis R D *et al.* Machine learning-assisted directed protein evolution

337 with combinatorial libraries. *Proc Natl Acad Sci U S A* **116**, 8852-8858 (2019).

338 17 Goldsmith M, Tawfik D S. Enzyme engineering: reaching the maximal catalytic efficiency

339 peak. *Curr Opin Struct Biol* **47**, 140-150 (2017).

340 18 Zeymer C, Hilvert D. Directed Evolution of Protein Catalysts. *Annu Rev Biochem* **87**, 131-

341 157 (2018).

342 19 Berman H M, Kleywegt G J, Nakamura H *et al.* The protein data bank archive as an open

343 data resource. *Journal of Computer-aided Molecular Design* **28**, 1009-1014 (2014).

344 20 Jumper J, Evans R, Pritzel A *et al.* High accuracy protein structure prediction using deep

345 learning. *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction*
346 *(Abstract Book)*, 22-24 (2020).
347 21 Ng P C, Henikoff S. SIFT: Predicting amino acid changes that affect protein function.
348 *Nucleic Acids Res* **31**, 3812-3814 (2003).
349 22 Peters M E, Ammar W, Bhagavatula C *et al.* Semi-supervised sequence tagging with
350 bidirectional language models. *Proceedings of the 55th Annual Meeting of the Association*
351 *for Computational Linguistics* 1756-1765 (2017).
352 23 Bengio Y, Ducharme R, Vincent P *et al.* A neural probabilistic language model. *Journal of*
353 *Machine Learning Research* **3**, 1137-1155 (2003).
354 24 Kingma D P, Ba J L. Adam: A method for stochastic optimization. *arXiv*, 1412.6980 (2015).
355 25 Ramachandran P, Zoph B, Le Q V. Searching for activation functions. *arXiv*, 1710.05941
356 (2017).
357 26 Ba J L, Kiros J R, Hinton G E. Layer normalization. *arXiv*, 1607.06450 (2016).
358 27 Mohammed A. ProteinNet: a standardized data set for machine learning of protein structure.
359 *BMC Bioinformatics* **20**, 311 (2019).
360 28 Wang J, Cao H, Zhang J Z H *et al.* Computational Protein Design with Deep Learning
361 Neural Networks. *Scientific Reports* **8**, 6349 (2018).
362 29 Zimmerman J M, Eliezer N, Simha R. The characterization of amino acid sequences in
363 proteins by statistical methods. *Journal of Theoretical Biology* **21**, 170-201 (1968).
364 30 Kyte J, Doolittle R F. A simple method for displaying the hydropathic character of a protein.

365 *Journal of Molecular Biology* **157**, 105-132 (1982).

366 31 Huang F, Nau W M. A conformational flexibility scale for amino acids in peptides.
367 *Angewandte Chemie International Edition* **42**, 2269-2272 (2003).

368 32 Dietterich T G. Ensemble methods in machine learning. *International Workshop on*
369 *Multiple Classifier Systems* **Springer, Berlin, Heidelberg**, 1-15 (2000).

370 33 van der Maaten L, Hinton G. Visualizing Data using t-SNE. *Journal of Machine Learning*
371 *Research* **9**, 2579-2605 (2008).

372 34 Fox N K, Brenner S E, Chandonia J M. SCOPe: Structural Classification of Proteins—
373 extended, integrating SCOP and ASTRAL data and classification of new structures.
374 *Nucleic Acids Research* **42**, D304-309 (2014).

375 35 Davies D L, Bouldin D W. A cluster separation measure. *IEEE Transactions on Pattern*
376 *Analysis and Machine Intelligence* **1**, 224-227 (1979).

377 36 Romero P A, Arnold F H. Exploring protein fitness landscapes by directed evolution.
378 *Nature Reviews Molecular Cell Biology* **10**, 866-876 (2009).

379 37 Biswas S, Kuznetsov G, Ogden P J *et al.* Toward machine-guided design of proteins.
380 *bioRxiv* (2018).

381 38 Yang K K, Chen Y X, Lee A *et al.* Batched stochastic Bayesian optimization via
382 combinatorial constraints design. *International Conference on Artificial Intelligence and*
383 *Statistics* **89** (2019).

384 39 Fang J. A critical review of five machine learning-based algorithms for predicting protein

385 stability changes upon mutation. *Briefings in Bioinformatics* **21**, 1285-1292 (2020).

386 40 Li M, Kales S C, Ma K *et al.* Balancing protein stability and activity in cancer: a new
387 approach for identifying driver mutations affecting CBL ubiquitin ligase activation. *Cancer*
388 *Research* **76**, 561-571 (2016).

389 41 Mysinger M M, Carchia M, Irwin J J *et al.* Directory of useful decoys, enhanced (DUD-
390 E): better ligands and decoys for better benchmarking. *J Med Chem* **55**, 6582-6594 (2012).

391 42 Zheng S, Li Y, Chen S *et al.* Predicting drug–protein interaction using quasi-visual question
392 answering system. *Nature Machine Intelligence* **2**, 134-140 (2020).

393

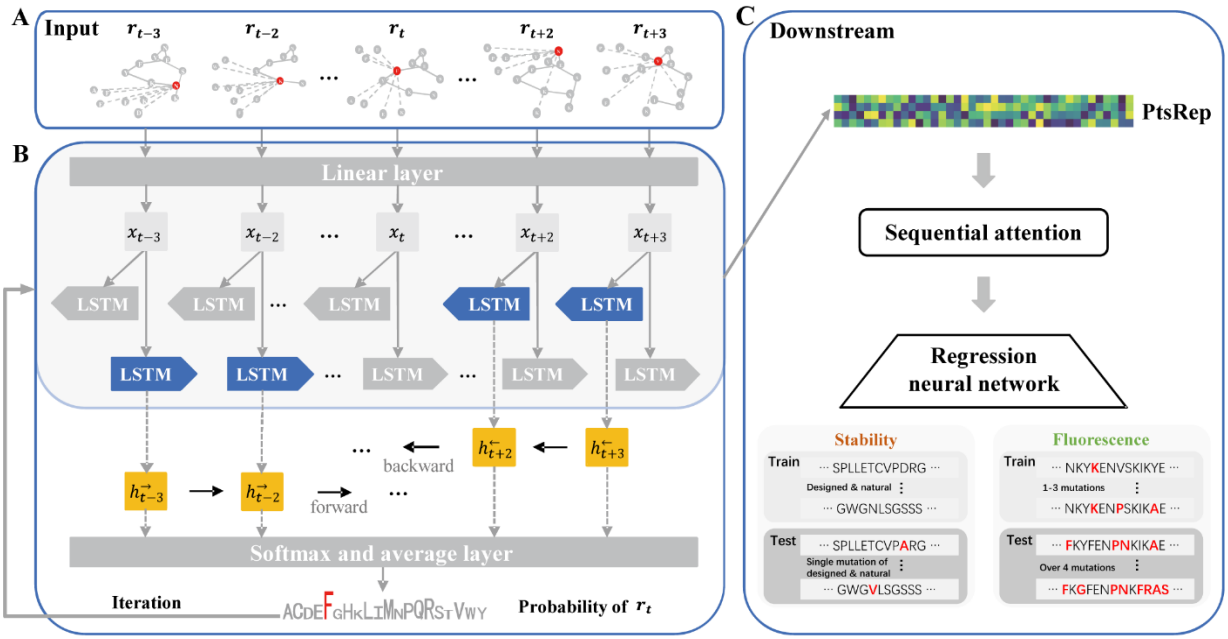

Figure 1. Workflow used by PtsRep for learning and applying protein tertiary structure representation. (A) In the input module, protein structures from PDB were encoded with the KNR algorithm, with each amino acid represented by the nearest 15 amino acids and their features. (B) In the learning (pre-training) module, PtsRep performed contextual noncontiguous residue prediction using a cross-entropy loss function, and internally represent proteins. (C) A regression neural network with attention mechanism was used to transfer the embedded representations to downstream prediction tasks.

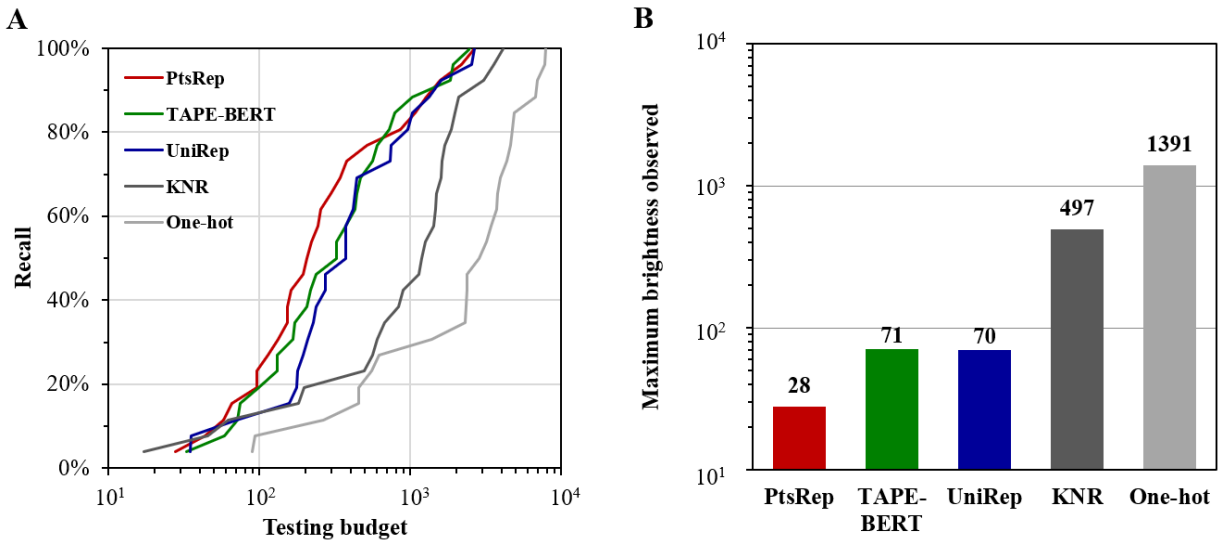

Figure 2. (A) Recall rates of the top 0.1% brightest GFP variants retrieved *versus* the required testing budget for each representation method. (B) The number of trials required to identify the brightest GFP variant based on the ranking obtained with each representation method.

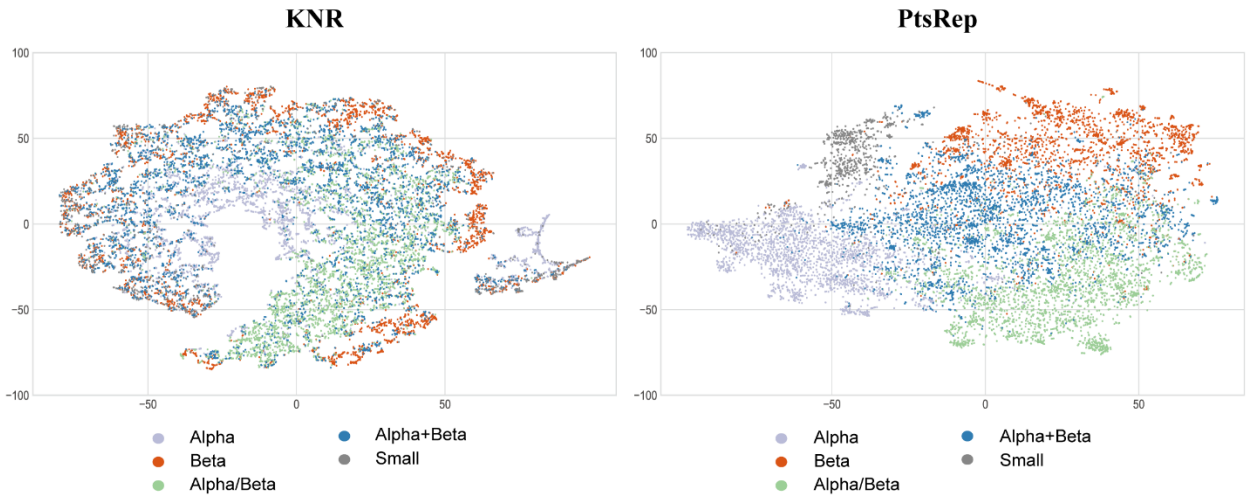

Figure 3. t-SNE representations obtained with KNR (left panel) and PtsRep (right panel) for 15,444 proteins classified by the Structural Classification of Proteins (SCOP)[34]. The t-SNE projections from the embedded space onto a low dimensional representation are shown. The sequences from SCOP are colored according to their ground-truth structural class (alpha, beta, alpha/beta, alpha+beta, and small proteins).

Table 1. Embedding performances for downstream tasks of PtsRep, the outlined benchmarks and the baselines.

Method	Embedding Parameters	Pretrained dataset	Stability			Fluorescence		
			ρ	ACC	MSE	ρ	ACC	MSE
UniRep[10]	18M	24M	0.73∗	0.69∗	0.15	0.67∗	0.96	**0.20**∗
TAPE-BERT[11]	92M	32M	0.73∗	0.70∗	0.12	0.68∗	0.96	0.22∗
Bepler[9]	19M	21.8M	0.64∗	0.67∗	-	0.33∗	-	2.17∗
one-hot	0	32M	0.19∗	0.58∗	0.70	0.14∗	0.86	2.69∗
KNR	0	0.04M	0.37	0.71	0.38	0.67	0.92	0.66
PtsRep	2.5M	0.04M	**0.79**	**0.73**	**0.09**	**0.70**	**0.97**	0.57

* taken from ref 11.