DeepAcr: Predicting Anti-CRISPR with Deep Learning

Yunxiang Li†1**, Yumeng Wei**†1**, Qingxiong Tan**1**, Licheng Zong**1**, Yixuan Wang**1,2**,**

Jiayang Chen1**, Yu Li**1 *

Abstract

As an important group of proteins discovered in phages, anti-CRISPR inhibits the activity of the immune system of bacteria (*i.e.*, CRISPR-Cas), showing great potential for gene editing and phage therapy. However, the prediction and discovery of anti-CRISPR are challenging for its high variability and fast evolution. Existing biological studies often depend on known CRISPR and anti-CRISPR pairs, which may not be practical considering the huge number of pairs in reality. Computational methods usually struggle with prediction performance. To tackle these issues, we propose a novel **deep** learning method for **a**nti-**CR**ISPR analysis (**DeepAcr**), which achieves impressive performance. On both the cross-fold and cross-dataset validation, our method outperforms the previous state-of-the-art methods significantly. Impressively, DeepAcr improves the prediction performance by at least 40% regarding the F1 score for the cross-dataset test. Moreover, DeepAcr is the first computational method to predict the detailed anti-CRISPR classes, which may help illustrate the anti-CRISPR mechanism. Taking advantage of a Transformer protein language model pre-trained on 250 million protein sequences, DeepAcr overcomes the data scarcity problem. Extensive experiments and analysis suggest that Transformer model feature, evolutionary feature, and local structure feature complement each other, which indicates the critical properties of anti-CRISPR proteins. Combined with AlphaFold prediction, further motif analysis and docking experiments demonstrate that DeepAcr captures the evolutionarily conserved pattern and the interaction between anti-CRISPR and the target implicitly. With the impressive prediction capability, DeepAcr can serve as a valuable tool for anti-CRISPR study and new anti-CRISPR discovery.

1 Introduction

Anti-CRISPR is an important group of proteins discovered in phages for fighting against the immune system of certain bacteria. To resist the invasion of phages, bacteria have evolved different types of defense mechanisms, including the important and adaptive immune system CRISPR-Cas. Correspondingly, phages evolved inhibitor proteins Anti-CRISPRs (Acrs) to fight with the CRISPR-Cas system. Because of the strong ability to adaptively detect, destroy, and modify DNA sequences, CRISPR-Cas has become a popular gene-editing tool. Since for each CRISPR-Cas system, there could be a dedicated Acr available [Stanley and Maxwell, 2018], performing accurate Acr predictions to find new Acrs is becoming increasingly important for many real-world applications, such as reducing off-target accidents in gene editing, measurement of gene drive, and phage therapy [Stanley and Maxwell, 2018, Marino et al., 2020, Pawluk et al., 2018].

*Correspondence to: liyu@cse.cuhk.edu.hk. †Equal contribution. [1]The Chinese University of Hong Kong, [2]Harbin Institute of Technology. Code available at: https://github.com/banma12956/DeepAcr.

In recent years, many biological and bioinformatic approaches have been built to predict and discover new Acrs. Based on a plasmid-based functional assay, the activity of prophage, which integrated phage genome and bacterial genome, was observed and evaluated, leading to the discovery of the first Acr that enabled phage to replicate successfully under CRISPR-Cas attack [Bondy-Denomy et al., 2013]. By utilizing the BLAST search strategy on anti-CRISPR-associated (Aca) gene, which is an important characteristic of certain Acr genes, Pawluk et al. [2016] developed a bioinformatic approach to find additional Acr proteins in phages and related diverse mobile genetic data in bacteria. Motivated by the idea that "self-targeting" prophage contain both a DNA target and CRISPR spacer as the CRISPR-Cas system is inactivated, Rauch et al. [2017] conducted a study to search bacterial genomes for the co-existence of spacer and its target, discovering new Acrs in phages with this self-targeting phenomenon. The design of a phage-oriented approach that allowed phages to challenge bacterial strains and ensured that screen phages can overcome the CRISPR-Cas system also led to the discovery of new Acrs [Hynes et al., 2017]. However, these methods depend on expensive and time consuming experimentally-generated data. Furthermore, they rely heavily on the homologs of Acrs and their functional characteristics, which is not practical because of the rapid emergence of a large number of new types of proteins. Finally, these methods require human participation, which limits their real-world applications dealing with large-scale genetic and protein sequences.

To predict Acrs and accelerate biological experiments, several machine learning (ML) methods were introduced based on directly learning useful knowledge from the Acr protein data. An ensemble learning-based Acrs prediction tool, PaCRISPR [Wang et al., 2020], was developed to apply the support vector machine (SVM) model on evolutionary features, which were extracted from Position-Specific Scoring Matrix (PSSM), including PSSM-composition [Zou et al., 2013], DPC-PSSM [Liu et al., 2010], PSSM-AC [Dong et al., 2009], and RPSSM [Ding et al., 2014]. Based on XGBoost ranking, a new Acrs prediction model named AcRanker [Eitzinger et al., 2020] was built to deal with the mixture of different sequence features, including grouped dimer, amino acid composition, and trimer frequency counts [Eitzinger et al., 2020]. A server called AcrFinder was built to pre-screen genomic data for Acr candidates by combining three well-accepted ideas, namely guilt-by-association (GBA), homology search, and CRISPR-Cas self-targeting spacers [Yi et al., 2020]. A machine learning approach consisting of a random forest with 1000 trees was built to predict comprehensive Acrs, which showed strong forecasting capability for the unseen test set [Gussow et al., 2020]. Also built upon the random forest algorithm, AcrDetector utilized merely six features to identify Acrs from the whole genome scale but still maintained the prediction precision [Dong et al., 2020]. An integrative database named AcrHub was developed by incorporating three state-of-the-art predictors, including homology network analysis, phylogenetic analysis, and similarity analysis [Wang et al., 2021]. This platform can provide investigating, mapping, and predicting services for Acr proteins. Compared with the traditional bioinformatic approaches, these models are relatively flexible since they can learn useful genetic information from the protein data itself, which provides good chances to analyze the large-scale protein sequences. However, these methods are usually built upon traditional machine learning techniques, which utilize simple linear models or shallow nonlinear models to analyze data, limiting their modeling capacity. This limits their prediction performance and makes it difficult to promote the development of Acr-based precise treatment.

Considering the scarce database, Acrs' quick evolution, and under-explored Arc features, here, we propose a novel deep learning approach (DeepAcr) for the effective and accurate prediction and classification of Acr proteins to facilitate new Acr discovery from large-scale protein databases. The biggest challenge of developing deep learning methods to predict Acrs is the lack of data. For example, in our dataset, we only have 1094 non-redundant Acr sequences, which is usually insufficient to train an effective deep learning model. To deal with the data scarcity issue, we introduce the pre-trained large-scale protein language model, Transformer protein language learning model, to provide more informative representations. To be specific, instead of merely using a few sequences to train our model, we introduce a Transformer learning module trained with UR50/S dataset [Rives et al., 2021], which can extract internal statistical properties to effectively promote the prediction performance for structure, function, and other tasks [Rives et al., 2021, Suzek et al., 2015]. This learning module explored 250 million protein sequences, significantly broadening our database and helping achieve better prediction performance. Meanwhile, this module is computationally efficient, which can meet the time requirement even used for tremendous protein sequences. Protein language models have been applied in many downstream tasks and lead to improvement, including protein-protein interaction [Sledzieski et al., 2021], residue-residue contact prediction [Rao et al., 2021]. Furthermore, to fully utilize valuable information of protein data to promote the performance of the proposed pipeline, we combine the protein language model feature with various other features, including Acr sequence information, Acr protein structure information, relative solvent accessibility, and four evolutionary features from PSSM (PSSM-composition, DPC-PSSM, PSSM-AC and RPSSM), which are proved helpful for Acrs prediction [Wang et al., 2020, Eitzinger et al., 2020]. Compared with the current classification scheme in the anti-CRISPR system, which broadly divides proteins into two categories according to whether a protein is Acr or not [Koonin et al., 2017], we can provide more detailed and informative hierarchical classification results. At the first level, similar to the current anti-CRISPR system, we predict whether a protein is an Acr. Then, at the second level, if it is an Acr protein, we further provide which class of Acrs the protein belongs to, which can bridge the large-scale protein database and Acrs to accelerate the discovery and verification of Acrs. Down to the details of prediction and classification procedure, we develop the model based on convolutional neural networks (CNNs) and deep neural networks (DNNs).

Our contributions are as follows.

- Based on the datasets from anti-CRISPRdb [Dong et al., 2018] and PaCRISPR [Wang et al., 2020], we propose the first deep learning method, DeepAcr, for anti-CRISPR prediction, which outperforms the previous state-of-the-art methods by at least 40% regarding the F1 score on cross-dataset validation.
- We develop the first method that can predict the detailed anti-CRISPR classes, which may help illustrate the anti-CRISPR mechanism.
- We are the first one in this field to resolve the data scarcity issue by transferring the knowledge from a large-scale protein language model trained on 250 million protein sequences.
- We perform extensive experiments and analysis to investigate the relation among the Transformer model feature, evolutionary feature, and local structure feature, which suggests that they complement each other, indicating the critical properties of anti-CRISPR proteins.
- Combining DeepAcr with AlphaFold and motif detection methods, we propose a computational pipeline to understand the model prediction basis better and validate our prediction computationally.

2 Results

Overview of DeepAcr. DeepAcr contains three basic parts, as illustrated in Fig. 1. First, a large collection of protein sequences are compiled into the pipeline to extract various protein features, including secondary structure, relative solvent accessibility, four evolutionary features, and Transformer features. Furthermore, by introducing the Transformer learning module to learn the Transformer features, we tackle the scarcity issue of Acrs database with a large-scale pre-training database. Second, the learned features are injected into DeepAcr, which jointly models these features using two deep learning modules, namely CNN and DNN, in an end-to-end trainable deep architecture. To this end, the Acr prediction results provide the confidence score of Acr for each protein, estimating the likelihood that the protein is an Acr protein. Finally, based on the predicted likelihood, we can perform some downstream tasks. For example, positive Acr candidates are further inputted into DeepAcr to perform the second-level prediction, predicting which sub-category the Acr belongs to. This sub-category prediction helps narrow down the candidate list, assisting biologists in carrying out biological experiments validation more effectively and discovering novel Acrs. We also conduct motif analysis to illustrate the implicit features learned by DeepAcr. Additionally, we further show the interaction details between the Acr protein and the target by predicting structure with AlphaFold and conducting protein-protein docking.

Extracting evolutionary features and structure features for comprehensive protein representation. Protein features necessary for Acr prediction are not well-defined. Thus, we focused on evolutionary and sequence features that have proven useful in earlier studies [Wang et al., 2020, 2021]. We also incorporated secondary structure, which is applied in protein function prediction [Li et al., 2018, Zou et al., 2019] and known to be related to protein-protein interaction [Jedhe and Arora, 2021]. First, we use one-hot encoding to encode protein sequence, which contains original amino acid information, reflecting vital sequential patterns of Acr proteins. Second, we extract PSSM features to introduce evolutionary information of proteins. The PSSM score is a $L \times 20$ matrix summarizing the homologs similar to a given protein sequence in the specified database. Each score

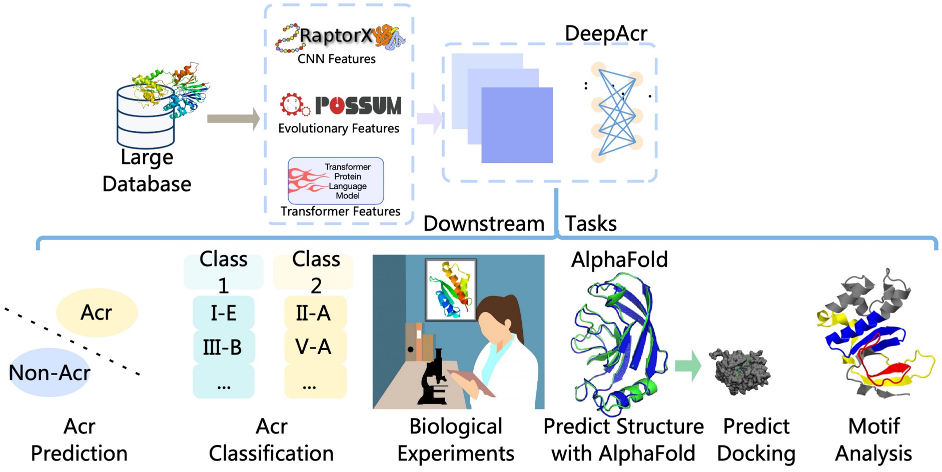


Figure 1: **The DeepAcr pipeline.** The process contains three steps. Firstly, we compile the large protein sequence database to obtain secondary structure, relative solvent accessibility, evolutionary features, and Transformer features. Secondly, based on the features, the DeepAcr model predicts the Acr proteins. Finally, with the predicted likelihood of the protein being an Acr protein, we can perform multiple downstream tasks, including predicting classes of Acrs, doing biological validation experiments, predicting the Acr structure with AlphaFold, investigating the interaction between Acr and the target with protein-protein docking, and motif analysis.

in the matrix reflects the conservation of the corresponding amino acid at this position, where large scores imply highly conservative. From the PSSM matrix, we calculate four evolutionary features, PSSM-composition, DPC-PSSM, PSSM-AC, and RPSSM, which are fixed-size matrices. They deal with PSSM varying lengths, encode relation between two elements, and local sequence order effect [Wang et al., 2020]. Third, we extract secondary structure information with traditional three classes [Pauling et al., 1951] and eight classes extended by Kabsch and Sander [1983], and convert them with one-hot encoding. We also consider solvent accessibility which presents local exposure of a protein. We label them with three states and apply one-hot encoding. The last type of feature that we consider is Transformer feature, which helps consider the biological structure and function of protein data. During the process of learning such features, we can simultaneously handle the scarcity issue of Acrs database. We discuss it as a separate section in the following paragraph.

Introducing Transformer learning to tackle data scarcity. The proposed deep learning model has a huge number of trainable parameters, which enables the model to have a very strong modeling capacity to analyze the protein data. However, it may suffer from the overfitting problem if there is a data scarcity issue, especially for the Acr prediction, where we only have around 1000 sequences. To resolve the issue, we introduce the Transformer learning algorithm to learn more informative representations [Rives et al., 2021], significantly broadening the database. With the explosive growth of protein sequence data, the unsupervised method was introduced to learn this representation, which can capture statistical regularities [Hinton et al., 2006]. The design of the Transformer learning module is based on the protein sequential structure, which determines the function and adaptability of proteins during evolution. This module learns the sequential information of the protein data by predicting the contents of the masked parts, therefore guiding it to learn useful structure information from the sequential data to provide effective representations. Specifically, the objective function is designed as follows to minimize the masked language modeling (MLM) loss:

$$\mathcal{L}_{MLM} = \mathbb{E}_{x \sim X} \mathbb{E}_M \sum_{i \in M} \log p(x_i | x_{/M}), \tag{1}$$

where x represents a protein sequence; M is the set of mask indices; x_i denotes a protein sequence with mask token at index i. Furthermore, because when we train the language model, we do not need the human-labeled data, we can train the model with as many protein sequences as possible. We adopt a protein language model trained over 250 million protein sequences. With such a huge amount of unsupervised training data, the model can implicitly learn the data distribution, evolutionary

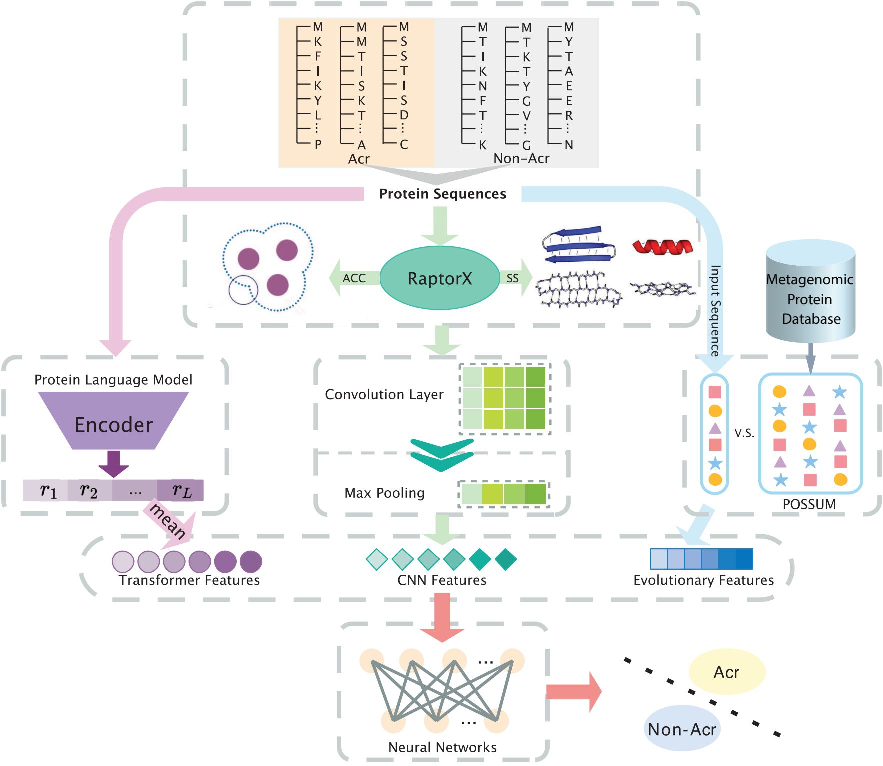


Figure 2: **The architecture of the proposed DeepAcr model.** We first obtain the relative solvent accessibility, secondary structure, evolutionary features, and Transformer features with RaptorX, POSSUM, and a pretrained protein language model. Then, we concatenate the features and input them to CNNs and DNNs to obtain the predicted likelihood of the protein being an Acr protein.

information, and protein sequential structure, leading to effective protein representation. In addition, compared with the time and memory-consuming PSI-BLAST, POSSUM [Wang et al., 2017] and RaptorX [Källberg et al., 2012], the introduced ESM-1b module [Yu et al., 2021, Chen et al., 2021] is able to generate Transformer features much quicker and simpler.

Architecture for complex feature learning and Acr prediction and classification. Since original information from proteins is complex, we leverage CNNs and DNNs to learn useful features, designed in the end-to-end trainable way as illustrated in Fig. 2. The one-hot matrices of sequence, structure feature, and solvent accessibility are concatenated together as the inputs of the CNN module. The convolutional layers learn deep features by extracting important local information from the one-hot matrices. After this, the max-pooling layer is attached along the sequential direction to calculate the largest value in each feature map. The features learned by the CNN module are further combined with the evolutionary features and Transformer features from ESM-1b, which are then jointly modeled by the DNN module. The Acr prediction is a binary classification task, which makes an estimation about whether a certain protein sequence is an Acr. Therefore, the final outputs are produced by fully connected layers with a two-unit output.

Considering biologists may also be interested in Acr types, if a protein is predicted to be an Acr, we will further estimate which kind of Acr this protein belongs to and provide more detailed information for the downstream biological experimental verification. Due to the limited number of Acr samples, we separate this task from Acr prediction to avoid the imbalanced data issue and defined it as the Acr classification task, predicting the specific category of the Acr. This classification task utilizes a similar structure with the Acr prediction task with slight modification, namely changing the dimension of the final output from two to five classes. In addition, we remove the non-anti-CRISPRs samples from the whole samples and only analyze samples predicted to be Acr proteins. We use the cross-entropy as the loss function and Adam as the optimizer to train the model. Moreover, to increase the flexibility of the model for real-world applications, we make the model accept the input with variant features, which allows users to select the input features according to their needs.

Anti-CRISPR prediction performance. In this section, we discuss the performance of our model on predicting whether a certain protein sequence is an Acr or not. To sufficiently compare the prediction performance of our proposed method and existing models, we perform 5-fold cross-validation test and an additional cross-dataset test to evaluate their generalization ability. To exclude the influence of random seeds, we randomly generate seeds ten times for the initialization of model parameters and the split of samples of training and test groups. The final result of each model is their average of ten times prediction results. Five evaluation metrics, namely accuracy (ACC), precision, recall, F1-score, and Matthews correlation coefficient (MCC), are utilized to evaluate the prediction results. The mathematical expressions for calculating these metrics are provided in Appendix C. Since other methods require additional information other than the sequence alone, such as gene location on chromosome [Dong et al., 2020], we compare DeepAcr with three recently proposed methods, namely PaCRISPR [Wang et al., 2020], AcRanker [Eitzinger et al., 2020], and the one from Gussow et al. [2020]. All of them have shown strong Acr predicting capacities and efficiency.

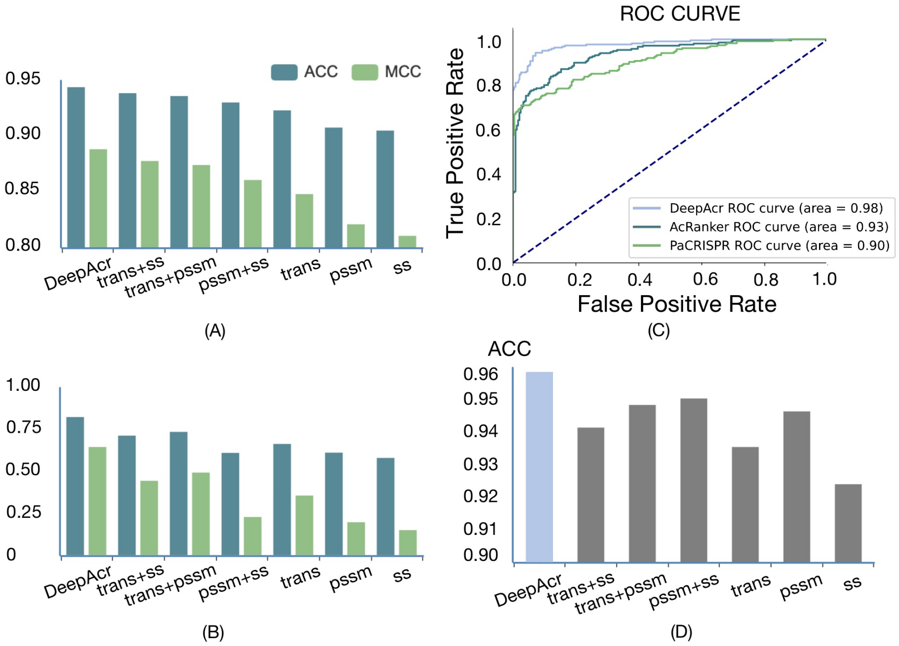
Table 1: **Five-fold cross-validation test results of anti-CRISPRs prediction.** DeepAcr achieves the best results, outperforming the other methods significantly. DeepAcr model with only Transformer features also has relatively good performance, better than PaCRISPR and AcRanker. We also show the prediction performance from Gussow et al. [2020], which utilizes a random forest model. Results in this table are averaged over 10 different random seeds in our experiments, and variances are smaller than 0.001.

Metrics	Accuracy	Precision	Recall	F1 score	MCC
AcRanker	0.8752	0.8591	0.8953	0.8666	0.7509
PaCRISPR	0.8087	0.7344	**0.9738**	0.7588	0.6552
Gussow et al. [2020]	/	0.78	0.57	0.6587	/
DeepAcr (Transformer only)	0.9326	0.9351	0.9248	0.9299	0.8652
DeepAcr	**0.9442**	**0.9471**	0.9409	**0.9418**	**0.8883**

Table 2: **Cross-dataset test results of anti-CRISPRs prediction.** DeepAcr accomplishes great improvement compared with the other state-of-the-art computational methods, especially on precision, F1 score, and MCC. The Transformer features contribute greatly to the DeepAcr generalization capability, but the other features are also very helpful. Results in this table are averaged over 10 different random seeds in our experiments, and variances are smaller than 0.001.

Metrics	Accuracy	Precision	Recall	F1 score	MCC
AcRanker	0.6319	0.2524	0.7681	0.3799	0.2681
PaCRISPR	0.5617	0.0286	0.7500	0.0550	0.0802
DeepAcr (Transformer only)	0.6957	**0.8764**	0.3714	0.5217	0.4176
DeepAcr	**0.7679**	0.7571	**0.8368**	**0.7950**	**0.5527**

The results of 5-fold cross-validation test and the cross-dataset test are demonstrated in Table 1 and Table 2 respectively, which clearly indicates that our proposed method significantly outperforms PaCRISPR, AcRanker, and the random forest model [Gussow et al., 2020]. Some important discoveries can be observed from these results. First, DeepAcr substantially outperforms other methods in both tests, especially in the cross-dataset test. The p-values for DeepAcr performing better than PaCRISPR and AcRanker are both < 0.0001. In the 5-fold cross-validation test, the F1 score is promoted by around 10%. The ROC curves also suggest that DeepAcr is more robust than the other methods (Fig. 3(C)). Significantly, DeepAcr achieves at least 40% improvement regarding F1 score in the cross-dataset test, where training and testing data have different distributions. The results demonstrate that DeepAcr is more suitable for the real Acr prediction task. We assume that DeepAcr method can automatically learn more useful knowledge from the protein data, including the important structure information, which enables DeepAcr to outperform other models. Furthermore, the Transformer model that we introduce into our model contains common properties of protein



(A)

(C)

(B)

(D)

Figure 3: **Feature influence on the prediction results.** "ss" represents sequence encoding, 3-class secondary structure, 8-class secondary structure, and solvent accessibility; "trans" indicates Transformer features; "pssm" denotes 4 evolutionary features extracted from PSSM. Results in the figure are averaged over 10 different random seeds in our experiments. (A) Contribution evaluation of different features on 5-fold cross-validation test. (B) Contribution evaluation of different features on cross-dataset test. (C) Performance comparison on Acr prediction regarding ROC curves of DeepAcr, AcRanker and PaCRISPR. (D) Contribution evaluation of different features for the detailed classification task.

sequences from an extensive database, which effectively improve the generalization ability for unseen sequences.

Cross-dataset test evaluates the model's generalization capacity when dealing with testing data with low sequence similarity from training data. From the results of the cross-dataset test, we can observe that both AcRanker and PaCRISPR achieve small Precision and F1-score scores, which may be because they tend to predict new sequences as positive and cause wrong predictions. However, our model can effectively avoid the wrong predictions because the rich knowledge about the extensive database contained in the Transformer learner can help DeepAcr make better predictions based on global considerations.

Evaluation of feature influence on prediction results. We perform ablation studies on different features with the 5-fold cross-validation and cross-dataset test to evaluate their influence on prediction results, as shown in Fig. 3(A) and (B). We can draw some conclusions from this study. Firstly, the performances of the combination of features are always better than the performances of features alone, which means the three kinds of features complement each other. They all reflect the different aspects of the Acr properties. Secondly, Transformer features play critical roles in both cross-validation and cross-dataset tests. Especially, Transformer features always lead to the best results among all the features and meanwhile can improve the results more significantly than other features. This indicates that the Transformer learning module can effectively learn valuable knowledge from the large database to promote the Acr prediction. Thirdly, due to the task difficulty, the structure information prediction would not be 100% accurate, which may influence the final prediction performance and thus should be taken into consideration when using it for prediction. Relatively, evolutionary features and Transformer features are more reliable, which can always promote the downstream tasks, such as the Acr prediction. Lastly, using the three features, DeepAcr (as given in Fig. 3 (A), (B)) is usually better than the shallow learning models (as provided in Table 1 and Table 2), which suggests the usefulness of the three features and the effectiveness of deep learning methods for this problem.

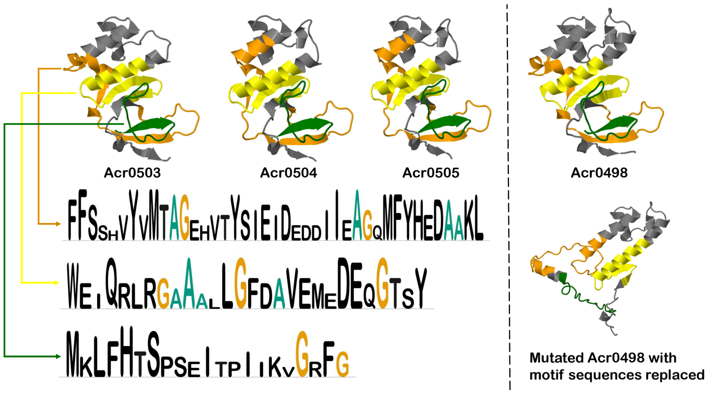

Acr0503
Acr0504
Acr0505
Acr0498
FFSSHVYVMTAGEHVTYSIEIDEDDIIEAGQMFYHEDAAKL
WEIQRLRGAAALGFDAVEMEDEQGTSY
MKLFHTSPSEITPIIKVGRFG
Mutated Acr0498 with motif sequences replaced

Figure 4: **Motif analysis results.** The structures of four samples Acr0503, Acr0504, Acr0505, and Acr0490 from the same type AcrFI11 are presented. Three found motifs are listed, and the corresponding structures are colored orange, yellow and green. The lengths of three motif sequences are 41, 28 and 21. The structure of mutated Acr0490, whose motif sequence is replaced by random sequences, is also shown.

Classification performance on Acr classes. To further estimate the capacity of our model in predicting the specific class that an Acr protein belongs to, we utilize the same features in the classification task as described in Section **Input data.**. The classification results on Acr categories and corresponding ablation studies are demonstrated in Table 3 and Fig. 3(D) respectively. We can observe from the histograms in Fig. 3(D) that the prediction results obtained using each type of feature are similar, which indicates that the Acr classification task is relatively straightforward when using these inputs. The evolutionary features obtained from PSSM can achieve slightly better results than other features in both single feature and combination with others. The graph may reveal some correlations between Acr classes and their features. To the best of our knowledge, this is the first attempt to predict the specific types of Acrs. To compare DeepAcr with the baseline, we adopt the one-vs-rest strategy to AcRanker and PaCRISPR to convert the binary prediction methods to multi-class prediction methods. Because Acrs belonging to the same class are more likely to form motif sequences, Hidden Markov Model (HMM) may also be able to capture these motif features. Thus, we apply HMM on protein sequences ans use it a baseline for Acr classification comparison. DeepAcr outperforms these methods across all the metrics, especially on macro-average metrics (improved by around 20% regarding F1 score), suggesting that DeepAcr is an unbiased predictor, performing well on the rare Acr classes. Such accurate, detailed predictions from our model can facilitate biological experiments and have the potential of inspiring more insightful studies on the working mechanisms of Acr proteins. In addition to separated prediction and classification problems, we also add non-Acr samples as the sixth class, and do classification on all data, which means that we combine prediction and classification together. The performance in Table 4 shows that separate prediction and classification have better results.

DeepAcr learns Acr motifs implicitly. To explain the hidden rule of model prediction, we conduct motif analysis to study the Acr sequence and structure patterns. MEME [Bailey et al., 1994] can find motif sequences in the protein data. Applying MEME on the Acr dataset, we show that Acrs belonging to the same class are more likely to form motif sequences. To further investigate the structure pattern of these motif sequences, we utilize an accurate protein structure prediction method, AlphaFold. Figure 4 presents some motif sequence and structure results. Such structures reveal that these motif sequences correspond to highly similar protein secondary structures, which may serve as the hidden rule for DeepAcr. Also, replacing motif sequences will change these importance protein secondary structures, which may make positive samples lose Acr-related features. To verify this conjecture, we conduct motif mutation by replacing the motif sequences with random sequences for

Table 3: **Detailed class prediction performance comparison.** We adopt the one-vs-rest strategy for AcRanker and PaCRISPR, converting binary classification methods to 5-class classification methods, and compare their performance on the class prediction problem with DeepAcr ("mi":micro-average, "ma": macro-average). DeepAcr outperforms the other methods across all the evaluation criteria significantly and consistently, especially on macro-average, suggesting that DeepAcr is an unbiased predictor for small classes. Results in this table are averaged over 10 different random seeds in our experiments.

	Accuracy (mi)	Accuracy (ma)	Precision (ma)	Recall (ma)	F1 score (ma)
AcRanker	0.8903	0.6318	0.8532	0.6318	0.6830
PaCRISPR	0.8903	0.5552	0.7071	0.5552	0.5911
HMM	0.8600	0.4994	0.7755	0.4994	0.5607
DeepAcr	**0.9480**	**0.8583**	**0.8676**	**0.8583**	**0.8531**

Table 4: **Each class prediction performance in DeepAcr.** We compare classification performance of each class in DeepAcr

Classes	Accuracy	Precision	Recall	F1 score
II-A	0.9801	0.9875	0.9802	0.9836
I-F	0.9199	0.8414	0.9199	0.8768
I-D	0.9889	0.9567	0.9889	0.9710
II-C	0.7915	0.7892	0.7916	0.7646
others	0.6110	0.7630	0.6110	0.6696

Table 5: **Use non-Acr as the sixth class in classification.** We add non-Acr samples as the sixth class and classify all data together. In this case we can solve prediction and classification problem at the same time. Whereas the performance is not as good as separate prediction and classification.

Classes	Accuracy	Precision	Recall	F1 score
II-A	0.9690	0.9836	0.9690	0.9761
I-F	0.8770	0.7848	0.8770	0.8148
I-D	0.9243	0.8938	0.9243	0.8999
II-C	0.7279	0.8475	0.7279	0.7622
others	0.3386	0.5489	0.3386	0.3932
non-Acr	0.9554	0.9483	0.9554	0.9516
Macro	0.7987	0.8344	0.7987	0.7996
Weighted	0.9372	0.9399	0.9372	0.9356

randomly selected Acrs, including Acr0498 (AcrFI11), Acr0562 (AcrIIA7), Acr0559 (AcrIIA8), and Acr0560 (AcrIIA9). The native Acr sequences and mutated sequences are inputted to DeepAcr for Acr prediction. The results show that the native Acrs are all predicted as Anti-CRISPR successfully, and all the mutated sequences are predicted as non-anti-CRISPR. Table 6 and 7 show the prediction results. The above results suggest that DeepAcr learns the important motifs of the Acr sequences implicitly, which serves as the foundation of its prediction. Considering rigorousness, we also mutate the non-motif sequences and follow the same steps mentioned above to study their effects. The results from Table 8 show that three mutated sequences are still predicted as anti-CRISPR but only one not, suggesting that our model indeed captures the important motif information in most Acr sequences.

Docking methods validate our predictions. Biological experimental validation is time-consuming and expensive, and we wish to facilitate it with protein-protein docking. For a new candidate Acr protein, we predict protein structure with AlphaFold and investigate the interaction between the protein and its target using protein-protein docking tools, which could also provide information about the Acr mechanism. Various docking tools are available including ClusPro [Desta et al., 2020, Vajda et al., 2017, Kozakov et al., 2017, 2013], HDOCK [Yan et al., 2020, 2017b,a, Huang and Zou, 2014, 2008], LzerD [Christoffer et al., 2021], Schrödinger [Zhu et al., 2014]. We study the interaction between Acr0275 from Anti-CRISPRdb and its receptor by HDOCK. We mutated the motif sequence from the Acr with random sequences and conducted docking for both the native protein and the mutated one. The docking results are shown in Fig. 5. The docking energy scores increased from -364.51 (predicted as Acr by DeepAcr) to -254.79 (predicted as non-Acr by DeepAcr), showing the

Table 6: DeepAcr prediction confidence scores of native Acr sequences.

	Acr0498	Acr0562	Acr0559	Acr0560
Acr	1	0.9881	1	1
Non-Acr	0	0.0119	0	0

Table 7: DeepAcr prediction confidence scores of mutated Acr sequences with motif sequences replaced by random sequences.

	Acr0498	Acr0562	Acr0559	Acr0560
Acr	0	0.0015	0.0001	0.1105
Non-Acr	1	0.9985	0.9999	0.8895

Table 8: DeepAcr prediction confidence scores of mutated Acr sequences with non-motif sequences replaced by random sequences.

	Acr0498	Acr0562	Acr0559	Acr0560
Acr	0.0922	1	1	1
Non-Acr	0.9078	0	0	0

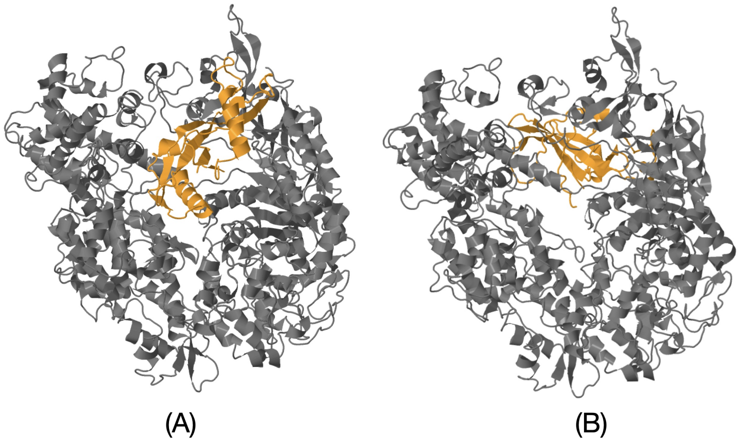


Figure 5: **Acr0275 and receptor docking results with HDOCK.** We check the docking results of both the native Acr0275 structure and the mutated one, with the motif sequence replaced and its structure predicted by AlphaFold. The structure in orange is the Acr sample, and the structure in grey is the receptor. (A) Docking result of the native Acr0275 and its receptor. (B) Docking result of the mutated Acr0275 with motif sequences replaced and its receptor.

mutated proteins is less stable interacting with the original receptor. Understandably, the current docking method may not be 100% accurate, but it could facilitate the discovery and study of Acr proteins, together with DeepAcr and AlphaFold. Further improvement on the tools could lead to better toolkits for Acr investigation.

3 Discussion and conclusion

Performing accurate anti-CRISPR predictions can help us reduce off-target accidents in gene editing and develop phage therapy. Here, we propose the first deep learning method, DeepAcr, for anti-CRISPR prediction. When receiving a protein sequence as input, DeepAcr will accurately estimate whether this sequence is an Acr, and further predict its specific type if it is an Acr, assisting biological identification of new Acrs efficiently. Without any prior knowledge, using biological experiments to verify whether a protein sequence is an Acr and its specific category is often very time-consuming and expensive, which is nearly impossible with the rapid emergence of a large number of protein sequences. DeepAcr overcomes the limitation of the experimental methods that rely heavily on the reaction between CRISPR-Cas and Acr to find new Acrs, thus can find Acrs from the large-scale protein database. Furthermore, the proposed DeepAcr can achieve accurate prediction and classification tasks efficiently, which is very important for practical applications in big proteins databases. By predicting Acr accurately, DeepAcr provides useful prior knowledge for biological researchers to identify Acr more efficiently.

DeepAcr achieves high accuracy on the Acr prediction tasks. One important reason is that we introduce the Transformer learning algorithm to deal with the problem of data scarcity. Previous methods only study small sizes of Acr datasets, which makes it difficult to learn general and informative patterns. As a result, these models cannot perform accurate Acr predictions and provide limited prior knowledge on protein sequences, which still need a large number of trial biological experiments for verification, providing an insufficient driving force for practical usages. By introducing the Transformer module, we can successfully generate effective representations by considering the structural information of protein sequences themselves and using the knowledge on the common properties of protein sequences learned from extensive databases in the pre-training process. These informative representations promote the training of the model effectively and improve the final prediction results, which will greatly decrease the time and cost of biological verification. Such an idea and features can also be applied to similar computational problems with limited annotated data.

The proposed DeepAcr method provides the first solution for predicting Acr classes, which have not been studied before. Unlike other methods, which rely on the CRISPR-Cas system to perform predictions, the proposed DeepAcr can directly predict the specific types of Acr without such limitation. Experimental results demonstrate that the four evolutionary features result in better prediction performance than other features, indicating that the specific Acr types are more related to the protein evolutionary information. In addition, experiments indicate that using more features extracted in our framework can effectively promote prediction accuracy, which proves that the feature extractions can promote the Acr protein analysis tasks.

The motif analysis shows the importance of motif sequences and their corresponding structures to Acrs prediction. Acrs belonging to the same category are more likely to have similar motif sequences, and these motif sequences correspond to highly similar protein secondary structures, which are the unique feature of this type of Acrs. Such sequence patterns and structure patterns can be learned by DeepAcr implicitly and serve as the prediction factors. Experiment results show that Acrs with motif sequences replaced will be predicted as non-Acrs by DeepAcr, which indicates the prediction basis of DeepAcr.

The success of AlphaFold and protein-protein docking analysis enhance our analysis pipeline. We can simulate the interaction between Acr and CRISPR-Cas proteins by utilizing docking tools before biological experiments. These tools can provide useful information, including docking position and docking energy, which can assist in designing and implementing biological experiments.

Despite the great improvement of DeepAcr over the previous methods, our method could be improved further with an even larger dataset. Also, information representation of the protein structure from AlphaFold could boost our model, although currently, it is still time-consuming to run AlphaFold, and efficient protein 3D structure representation is still under exploration. Finally, with the assistance of AlphaFold and docking tools, we can illustrate the Acr mechanism to some extent. However, they are not within the DeepAcr model. In the future, it will be desirable to design a comprehensive deep learning model, which can perform Acr prediction and illustrate its mechanisms simultaneously.

4 Methods

Features. Here we provide more details about our features. First, for protein sequence, we utilize the one-hot encoding technique to represent the protein sequence as $L \times 20$ matrix, where L indicates the length of the sequence and 20 is the number of native amino acid types. As illustrated in Appendix A Fig. 6, for each protein sequence, one vector only has a single '1' to represent the certain type of amino acid while setting the rest positions as '0's. Second, we utilize the POSSUM toolkit [Wang et al., 2017] to obtain the PSSM evolutionary features, which first calculates the PSSM matrix using the PSI-BLAST program via multiple iterations and certain E-value [Wang et al., 2020], then extracts the four features, namely PSSM-composition, DPC-PSSM, PSSM-AC, and RPSSM. In our implementation, we use the default setting, including the UniRef50 database, and the iteration number and E-value are set as 3 and 0.001, respectively. Third, we extract two types of secondary structure features to consider the local folding information of protein fully. We first consider the traditional secondary structure that can be divided into three classes, namely two regular types alpha-helix

Table 9: Statistical summary of the Acr classes in the non-redundant Anti-CRISPRdb dataset. We keep the largest four classes and group the rest eight smaller classes as class five.

Acr type	II-A	I-F	I-D	II-C	I-E	V-A	I-C	VI-A	VI-B	III-1	III-B	I-B
Collection	828	134	46	30	19	15	8	7	7	1	1	1

Table 10: Statistics summary of the dataset used in our experiments. We combine Anti-CRISPRdb dataset and PaCRISPR dataset, and remove the sequence redundancy.

	Cross-dataset training	Cross-dataset testing	5-fold validation (sum)
Positive	884	210 (From type I-F, II-C, and I-D in anti-CRISPRdb)	1094
Negative	902	260	1162

(H), beta-strand (E), and one irregular type coil region (C) [Pauling et al., 1951]. Then, we further consider the extended secondary structure with eight classes, namely 3_{10} helix (G), alpha-helix (H), pi-helix (I), beta-strand (E), beta-bridge (B), beta-turn (T), high curvature loop (S), and irregular (L) [Kabsch and Sander, 1983]. These two types of secondary structure features are transformed into matrices with shapes of $L \times 3$ and $L \times 8$ respectively using one-hot encoding techniques, which are then combined to consider more secondary structure information, as illustrated in Appendix B Fig. 7. In our implementation, we adpoted the RaptorX tool [Källberg et al., 2012] to predict secondary structure. Then, three states of solvent accessibility are derived from two thresholds, namely buried (0-10%), medium (11%-40%), and exposed (41%-100%). These features are then encoded into a $L \times 3$ matrix and appended to the matrixes of the secondary structure mentioned in the previous part, as provided in Appendix B Fig. 7. Similarly, the RaptorX tool in Källberg et al. [2012] is utilized to calculate the solvent accessibility information.

Finally, we utilize the ESM-1b Transformer to calculate Transformer features, which trained a 33-layer Transformer model on UR50/S dataset with 250 million sequences by comparing the results of Transformer models with different sizes and training datasets [Rives et al., 2021, Suzek et al., 2007]. This Transformer module consists of 33 encoder blocks, each of which contains a multi-headed self-attention unit and a feed-forward network unit. In our implementation, the outputs of the last encoder block are used. To deal with the issue of unequal protein sequence lengths, we calculate the mean values of the hidden states of all tokens and recorded them as the Transformer features.

Input data. We collect the anti-CRISPRs samples and non-anti-CRISPRs samples from Anti-CRISPRdb [Dong et al., 2018] and PaCRISPR [Wang et al., 2020], respectively. We firstly describe the process of collecting non-redundant positive samples from Anti-CRISPRdb [Dong et al., 2018]. More than 3000 experimentally characterized Acrs are initially collected [Dong et al., 2018] and CD-HIT is then applied to these samples to remove the redundant sequences by using a 95% identity threshold. After this selection process, we obtain 1094 non-redundant protein sequences of the Acrs. Then, we introduce the procedures of constructing the negative training dataset and the cross-dataset test dataset from PaCRISPR [Wang et al., 2020]. The non-anti-CRISPRs samples in this dataset [Wang et al., 2020] are all from Acr-containing phage or bacterial mobile genetic elements. The sequence sizes of these samples range from 50 to 350, with the inter-similarity of less than 40%. By using the criteria for selecting the negative sample proteins as introduced in Wang et al. [2020], we obtain 902 negative samples from the negative training dataset and 260 negative samples from the negative testing dataset for cross-dataset test. The 1094 non-redundant positive samples from Anti-CRISPRdb, and the 902 negative samples from the negative training dataset, and 260 negative samples from the cross-dataset dataset of Wang et al. [2020], are combined as the whole experiments dataset, where the sequence similarity between positive samples and negative samples is also less than 40%. In the cross-dataset test, the Acrs of types I-F, II-C, and I-D are selected as positive test samples, and the resting Acrs are used as positive training samples. Since the sequence similarity between the training samples and the test samples is less than 40%, this cross-dataset test can effectively reflect the generalization ability of the proposed model.

To collect the dataset for the task of the Acr type prediction task, we utilize the non-redundant samples in Anti-CRISPRdb, which contains 12 types of Acrs. The names of these 12 types of Acrs and the corresponding number of collections for each kind are in Table 9. In our design, we re-group these 12 types into 5 types. Specifically, the samples in the first four types, namely II-A, I-F, I-D, and II-C, are maintained, while the rest 8 types with very few samples are grouped into the 'fifth type'. Therefore, as described in the Results section, the Acrs classification problem is a 5-class classification task.

Evaluation test setup. To evaluate the performance of our deep learning model, we adopt 5-fold cross-validation tests on the dataset, consisting of 1094 positive samples and 1162 negative samples, as described in the previous section. Furthermore, another cross-dataset test is conducted to evaluate the generalization ability of the proposed model for the Acr prediction. In the cross-dataset test, the Acrs of types I-F, II-C, and I-D are selected as positive test samples, while the rest of the Acrs are used as positive training samples. Since the sequence similarity between the training samples and the test samples is less than 40%, this cross-dataset test can effectively reflect the generalization ability of the proposed model. For negative samples, we use the separation method provided by PaCRISPR to separate samples, which means 902 negative samples from the negative training dataset of PaCRISPR are used as negative training samples, while 260 negative samples from the cross-dataset dataset of PaCRISPR are used as negative test samples. The specific arrangements are illustrated in Table 10.

Implementation details of DeepAcr. The proposed DeepAcr, which mainly consists of CNN and DNN, is implemented with Python 3.7 and PyTorch 1.8 [Paszke et al., 2019], and is trained on NVIDIA GeForce RTX 3090. The one-hot encoded inputs, such as sequence, secondary structure, and relative accessibility, are concatenated together, then further processed by a 2D CNN to learn more high-level and informative features. For convenience, the kernel width of the CNN module is set the same as that of the concatenated feature. Max-pooling is connected after the CNN layer. The four evolutionary features and Transformer features were firstly injected into a 2-layer DNN, then concatenated with the high-level features learned from the one-hot encoded features via CNN. The concatenated features are finally inputted into another DNN layer, which has two-dimensional outputs for the Acr prediction task while has five-dimensional outputs for the Acr classification task. During the training process, the batch size is set as 16, the number of epochs is 3000, and the learning rate is set as 0.001. We utilize the Adam [Kingma and Ba, 2014] provided by PyTorch as the optimizer to train the model. 3000 epochs complete in about 179.22 seconds and inference time is 0.013 seconds. Total 2256 sequences require 202.62 seconds to compute Transformer features. ESM-1b was trained for 56 epochs, 8.5 hours on 64 GPUs for each epoch. To deal with the imbalance issue in the Acr classification tasks, as illustrated in Table 9, we select each sequence with different weights to ensure each class has the same probability of being sampled.

References

T. L. Bailey, C. Elkan, et al. Fitting a mixture model by expectation maximization to discover motifs in bipolymers. 1994.

J. Bondy-Denomy, A. Pawluk, K. L. Maxwell, and A. R. Davidson. Bacteriophage genes that inactivate the crispr/cas bacterial immune system. *Nature*, 493(7432):429–432, 2013.

S. Chen, Q. Tan, J. Li, and Y. Li. Uspnet: unbiased organism-agnostic signal peptide predictor with deep protein language model. *bioRxiv*, 2021.

C. Christoffer, S. Chen, V. Bharadwaj, T. Aderinwale, V. Kumar, M. Hormati, and D. Kihara. Lzerd webserver for pairwise and multiple protein–protein docking. *Nucleic Acids Research*, 2021.

I. T. Desta, K. A. Porter, B. Xia, D. Kozakov, and S. Vajda. Performance and its limits in rigid body protein-protein docking. *Structure*, 28(9):1071–1081, 2020.

S. Ding, Y. Li, Z. Shi, and S. Yan. A protein structural classes prediction method based on predicted secondary structure and psi-blast profile. *Biochimie*, 97:60–65, 2014.

C. Dong, G.-F. Hao, H.-L. Hua, S. Liu, A. A. Labena, G. Chai, J. Huang, N. Rao, and F.-B. Guo. Anti-crisprdb: a comprehensive online resource for anti-crispr proteins. *Nucleic acids research*, 46(D1):D393–D398, 2018.

C. Dong, D.-K. Pu, C. Ma, X. Wang, Q.-F. Wen, Z. Zeng, and F.-B. Guo. Precise detection of acrs in prokaryotes using only six features. *bioRxiv*, 2020.

Q. Dong, S. Zhou, and J. Guan. A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, 25(20):2655–2662, 2009.

S. Eitzinger, A. Asif, K. E. Watters, A. T. Iavarone, G. J. Knott, J. A. Doudna, and F. u. A. A. Minhas. Machine learning predicts new anti-crispr proteins. *Nucleic acids research*, 48(9):4698–4708, 2020.

A. B. Gussow, A. E. Park, A. L. Borges, S. A. Shmakov, K. S. Makarova, Y. I. Wolf, J. Bondy-Denomy, and E. V. Koonin. Machine-learning approach expands the repertoire of anti-crispr protein families. *Nature communications*, 11(1):1–12, 2020.

G. Hinton, S. Osindero, M. Welling, and Y.-W. Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, 30(4):725–731, 2006.

S.-Y. Huang and X. Zou. An iterative knowledge-based scoring function for protein–protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2):557–579, 2008.

S.-Y. Huang and X. Zou. A knowledge-based scoring function for protein-rna interactions derived from a statistical mechanics-based iterative method. *Nucleic acids research*, 42(7):e55–e55, 2014.

A. P. Hynes, G. M. Rousseau, M.-L. Lemay, P. Horvath, D. A. Romero, C. Fremaux, and S. Moineau. An anti-crispr from a virulent streptococcal phage inhibits streptococcus pyogenes cas9. *Nature microbiology*, 2(10):1374–1380, 2017.

G. S. Jedhe and P. S. Arora. Chapter one - hydrogen bond surrogate helices as minimal mimics of protein -helices. In E. J. Petersson, editor, *Synthetic and Enzymatic Modifications of the Peptide Backbone*, volume 656 of *Methods in Enzymology*, pages 1–25. Academic Press, 2021. doi: https://doi.org/10.1016/bs.mie.2021.04.007. URL `https://www.sciencedirect.com/science/article/pii/S0076687921001427`.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12): 2577–2637, 1983.

M. Källberg, H. Wang, S. Wang, J. Peng, Z. Wang, H. Lu, and J. Xu. Template-based protein structure modeling using the raptorx web server. *Nature protocols*, 7(8):1511–1522, 2012.

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

E. V. Koonin, K. S. Makarova, and F. Zhang. Diversity, classification and evolution of crispr-cas systems. *Current opinion in microbiology*, 37:67–78, 2017.

D. Kozakov, D. Beglov, T. Bohnuud, S. E. Mottarella, B. Xia, D. R. Hall, and S. Vajda. How good is automated protein docking? *Proteins: Structure, Function, and Bioinformatics*, 81(12):2159–2166, 2013.

D. Kozakov, D. R. Hall, B. Xia, K. A. Porter, D. Padhorny, C. Yueh, D. Beglov, and S. Vajda. The cluspro web server for protein–protein docking. *Nature protocols*, 12(2):255–278, 2017.

Y. Li, S. Wang, R. Umarov, B. Xie, M. Fan, L. Li, and X. Gao. Deepre: sequence-based enzyme ec number prediction by deep learning. *Bioinformatics*, 34(5):760–769, 2018.

T. Liu, X. Zheng, and J. Wang. Prediction of protein structural class for low-similarity sequences using support vector machine and psi-blast profile. *Biochimie*, 92(10):1330–1334, 2010.

N. D. Marino, R. Pinilla-Redondo, B. Csörgő, and J. Bondy-Denomy. Anti-crispr protein applications: natural brakes for crispr-cas technologies. *Nature methods*, 17(5):471–479, 2020.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

L. Pauling, R. B. Corey, and H. R. Branson. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4): 205–211, 1951.

A. Pawluk, R. H. Staals, C. Taylor, B. N. Watson, S. Saha, P. C. Fineran, K. L. Maxwell, and A. R. Davidson. Inactivation of crispr-cas systems by anti-crispr proteins in diverse bacterial species. *Nature microbiology*, 1(8):1–6, 2016.

A. Pawluk, A. R. Davidson, and K. L. Maxwell. Anti-crispr: discovery, mechanism and function. *Nature Reviews Microbiology*, 16(1):12–17, 2018.

R. Rao, J. Meier, T. Sercu, S. Ovchinnikov, and A. Rives. Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2021.

B. J. Rauch, M. R. Silvis, J. F. Hultquist, C. S. Waters, M. J. McGregor, N. J. Krogan, and J. Bondy-Denomy. Inhibition of crispr-cas9 with bacteriophage proteins. *Cell*, 168(1-2):150–158, 2017.

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

S. Sledzieski, R. Singh, L. Cowen, and B. Berger. Sequence-based prediction of protein-protein interactions: a structure-aware interpretable deep learning model. *bioRxiv*, 2021.

S. Y. Stanley and K. L. Maxwell. Phage-encoded anti-crispr defenses. *Annual review of genetics*, 52: 445–464, 2018.

B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

S. Vajda, C. Yueh, D. Beglov, T. Bohnuud, S. E. Mottarella, B. Xia, D. R. Hall, and D. Kozakov. New additions to the c lus p ro server motivated by capri. *Proteins: Structure, Function, and Bioinformatics*, 85(3):435–444, 2017.

J. Wang, B. Yang, J. Revote, A. Leier, T. T. Marquez-Lago, G. Webb, J. Song, K.-C. Chou, and T. Lithgow. Possum: a bioinformatics toolkit for generating numerical sequence feature descriptors based on pssm profiles. *Bioinformatics*, 33(17):2756–2758, 2017.

J. Wang, W. Dai, J. Li, R. Xie, R. A. Dunstan, C. Stubenrauch, Y. Zhang, and T. Lithgow. Pacrispr: a server for predicting and visualizing anti-crispr proteins. *Nucleic acids research*, 48(W1): W348–W357, 2020.

J. Wang, W. Dai, J. Li, Q. Li, R. Xie, Y. Zhang, C. Stubenrauch, and T. Lithgow. Acrhub: an integrative hub for investigating, predicting and mapping anti-crispr proteins. *Nucleic Acids Research*, 49(D1):D630–D638, 2021.

Y. Yan, Z. Wen, X. Wang, and S.-Y. Huang. Addressing recent docking challenges: A hybrid strategy to integrate template-based and free protein-protein docking. *Proteins: Structure, Function, and Bioinformatics*, 85(3):497–512, 2017a.

Y. Yan, D. Zhang, P. Zhou, B. Li, and S.-Y. Huang. Hdock: a web server for protein–protein and protein–dna/rna docking based on a hybrid strategy. *Nucleic acids research*, 45(W1):W365–W373, 2017b.

Y. Yan, H. Tao, J. He, and S.-Y. Huang. The hdock server for integrated protein–protein docking. *Nature protocols*, 15(5):1829–1852, 2020.

H. Yi, L. Huang, B. Yang, J. Gomez, H. Zhang, and Y. Yin. Acrfinder: genome mining anti-crispr operons in prokaryotes and their viruses. *Nucleic acids research*, 48(W1):W358–W365, 2020.

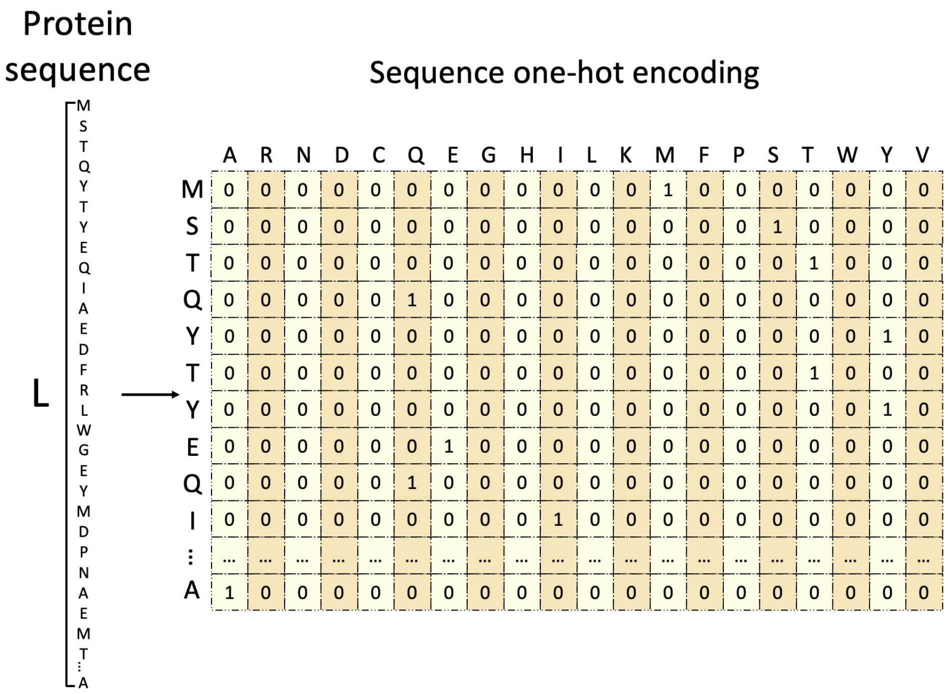
Q. Yu, Z. Dong, X. Fan, L. Zong, and Y. Li. Hmd-amp: Protein language-powered hierarchical multi-label deep forest for annotating antimicrobial peptides. *arXiv preprint arXiv:2111.06023*, 2021.

K. Zhu, T. Day, D. Warshaviak, C. Murrett, R. Friesner, and D. Pearlman. Antibody structure determination using a combination of homology modeling, energy-based refinement, and loop prediction. *Proteins: Structure, Function, and Bioinformatics*, 82(8):1646–1655, 2014.

L. Zou, C. Nan, and F. Hu. Accurate prediction of bacterial type iv secreted effectors using amino acid composition and pssm profiles. *Bioinformatics*, 29(24):3135–3142, 2013.

Z. Zou, S. Tian, X. Gao, and Y. Li. mldeepre: Multi-functional enzyme function prediction with hierarchical multi-label deep learning. *Frontiers in Genetics*, 9:714, 2019.

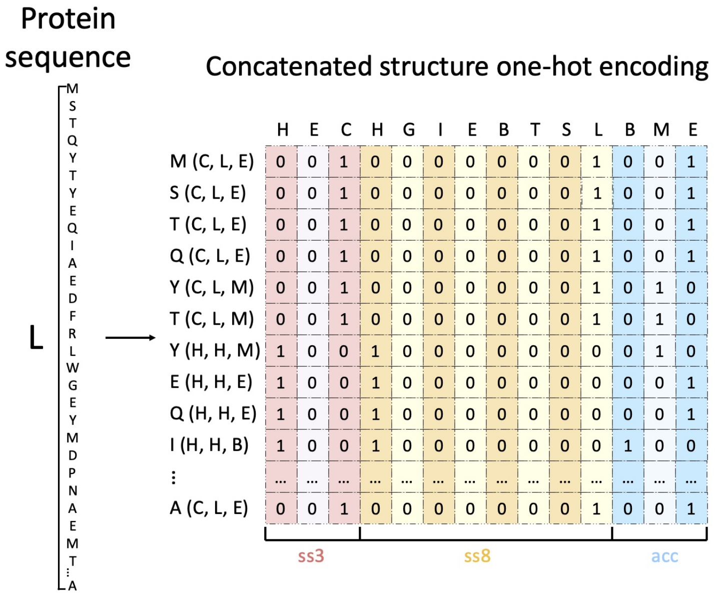
A One-hot illustration


Protein sequence
Sequence one-hot encoding
L
M S T Q Y T Y E Q I A E D F R L W G E Y M D P N A E M T ⋮ A

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
M	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Q	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
E	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
⋮	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6: Protein sequence one-hot matrix

B Secondary structure and solvent accessibility one-hot illustration


Protein sequence
Concatenated structure one-hot encoding
L
M S T Q Y T Y E Q I A E D F R L W G E Y M D P N A E M T ⋮ A
ss3
ss8
acc

	H	E	C	H	G	I	E	B	T	S	L	B	M	E
M (C, L, E)	0	0	1	0	0	0	0	0	0	0	1	0	0	1
S (C, L, E)	0	0	1	0	0	0	0	0	0	0	1	0	0	1
T (C, L, E)	0	0	1	0	0	0	0	0	0	0	1	0	0	1
Q (C, L, E)	0	0	1	0	0	0	0	0	0	0	1	0	0	1
Y (C, L, M)	0	0	1	0	0	0	0	0	0	0	1	0	1	0
T (C, L, M)	0	0	1	0	0	0	0	0	0	0	1	0	1	0
Y (H, H, M)	1	0	0	1	0	0	0	0	0	0	0	0	1	0
E (H, H, E)	1	0	0	1	0	0	0	0	0	0	0	0	0	1
Q (H, H, E)	1	0	0	1	0	0	0	0	0	0	0	0	0	1
I (H, H, B)	1	0	0	1	0	0	0	0	0	0	0	1	0	0
⋮	…	…	…	…	…	…	…	…	…	…	…	…	…	…
A (C, L, E)	0	0	1	0	0	0	0	0	0	0	1	0	0	1

Figure 7: Structure information one-hot matrix

C Measurements

Let TP, TN, FP and FN denote true positives, true negatives, false positives, and false negatives, respectively. We measure the model performance by Accuracy (ACC), Precision, Recall, F1-value and Matthews correlation coefficient (MCC). The equations are as follows:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - value = 2 \times \frac{TP}{2TP + FP + FN}$$

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