HMD-AMP: Protein Language-Powered Hierarchical Multi-label Deep Forest for Annotating Antimicrobial Peptides

Qinze Yu[1,2,3], Zhihang Dong[1], Xingyu Fan[2,3], Licheng Zong[1], and Yu Li [*1,2]

[1]Department of Computer Science and Engineering, CUHK, Hong Kong SAR, China
[2]The CUHK Shenzhen Research Institute, Hi-Tech Park, Nanshan, Shenzhen, 518057, China
[3]University of Electronic Science and Technology of China, Chengdu, Sichuan, China

Abstract

Identifying the targets of an antimicrobial peptide is a fundamental step in studying the innate immune response and combating antibiotic resistance, and more broadly, precision medicine and public health. There have been extensive studies on the statistical and computational approaches to identify (i) whether a peptide is an antimicrobial peptide (AMP) or a non-AMP and (ii) which targets are these sequences effective to (Gram-positive, Gram-negative, etc.). Despite the existing deep learning methods on this problem, most of them are unable to handle the small AMP classes (anti-insect, anti-parasite, etc.). And more importantly, some AMPs can have multiple targets, which the previous methods fail to consider. In this study, we build a diverse and comprehensive multi-label protein sequence database by collecting and cleaning amino acids from various AMP databases. To generate efficient representations and features for the small classes dataset, we take advantage of a protein language model trained on 250 million protein sequences. Based on that, we develop an end-to-end hierarchical multi-label deep forest framework, HMD-AMP, to annotate AMP comprehensively. After identifying an AMP, it will further predict what targets the AMP can effectively kill from eleven available classes. Extensive experiments suggest that our framework outperforms state-of-the-art models in both the binary classification task and the multi-label classification task, especially on the minor classes. Compared with the previous deep learning methods, our method improves the performance on macro-AUROC by 11%. The model is robust against reduced features and small perturbations and produces promising results. We believe HMD-AMP will both contribute to the future wet-lab investigations of the innate structural properties of different antimicrobial peptides and build promising empirical underpinnings for precise medicine with antibiotics.

Keywords: antimicrobial peptides, deep forest, protein language model, multi-label classification

*Corresponding Author. Email: liyu@cse.cuhk.edu.hk

1 Introduction

Antimicrobial peptides (AMPs) are potent, broad-spectrum antibiotics, which can help us combat diseases such as bacterial infections. For example, they have been found and synthesized to combat *pseudomonas aeruginosa* [1], to heal wounds [2], and to even potentially work against coronavirus [3]. Meanwhile, natural antimicrobial peptides have become an exciting area of research over the recent decades due to rather an inevitable risk associated with antibiotics: some antimicrobials can bring detrimental effects on the body's normal microbial content by indiscriminately attacking both the pathological occurring and beneficial ones, which damage the essential functions of our lungs, intestines and other organs. These side-effects root from the broad-spectrum property of some antibiotics, which could cause serious repercussions on immunity, nutrition and worse still, leading to a relative overgrowth of certain bacteria and fungi [4–7]. The latter repercussion could further lead to secondary infection such as *clostridioides difficile*, which stems from the overgrowth of microorganisms that are antibiotic-resistant [8]. Consequently, to alleviate the harm caused by antibiotics, inferences on the target of AMPs and how different peptide sequences kill targets such as viral pathogens and gram-positive bacteria are vital for major progress in antimicrobial peptide research and a full utilization of AMP functions.

Scientific contributions to antimicrobial peptide research include a wide range of wet-lab studies and computational biology studies. Examples of the former include finding out novel AMPs such as SAAP-148 that combats drug-resistant bacteria and biofilm [9] and LL-37 that works against *staphylococcus aureus biofilm* [10], extracting antimicrobial from tropical fruits [2] and studying lipid and metal nanoparticles for antimicrobial peptide delivery [11]. While wet-lab research is crucial for knowledge discoveries in this domain, their limited generality considering the required time investment for each analysis makes it difficult to evaluate AMPs at scale. With the advance of statistical and computational methodologies, we have observed exciting recent progress on this challenge. These contributions can be divided into three categories. First and foremost, there are improving data availability, such as DBAASP [12], LAMP [13], CAMP [14], APD-3 [15], DAMPD [16] and DRAMP 2.0 [10]. The abundance of data and computational resources enables the scientific community to train large-scale models. Second, there are computational design and syntheses efforts using methods like semi-supervised learning [17]. Finally, we have observed an accelerated growth of efforts on computational and statistical approaches analyzing AMPs, including statistical inferences like AntiBP2 [18], propensity score-based binary response models [19] and novel multi-level pseudo-amino acid composition in iAMP-2L [20] using fuzzy k-nearest neighbors. In recent years, AMP classification becomes a question of interest in the machine learning community, too. The discussion over the possibility of identifying novel antibacterial peptides using chemoinformatics and machine learning can be dated as early as 2009 [21]. As the machine learning toolkit expands and computational resources become more affordable, the methods applied in this question also become more diverse. For example, one study uses random forest in their AmPEP framework to predict antimicrobial peptides using distribution patterns of amino acids [22]; other studies use a deep regression model to perform antimicrobial peptide design [23]. There are even studies using deep generative networks and molecular dynamics simulations to speed up antimicrobial peptide discoveries [24]. Perhaps one of the most notable development that sparked our interests leveraged the power of convolutional neural networks (CNN) and long-short-term-memory networks (LSTM) [25] for the task of antimicrobial peptides classification. The paper provides a novel use of deep learning methodologies on large-scale AMP databases to identify the target of antimicrobial peptides.

At the same time, there are several major challenges with computational approaches. Most antimicrobial databases include only sequences that are antimicrobials (positive labels), meaning that generating negative samples is challenging. Although there were efforts synthesizing some negative samples in previous studies, the "negative" cases are constructed in a way that is too easy to classify, which does not reflect the true intrinsic structures of non-AMPs. In addition, most antimicrobial databases include only gram-positive and gram-negative cases. In situations where multi-label data are provided, the class distribution tends to go extremely imbalanced. With respect to available models, traditional methodologies tend to overfit and fail to deliver similar performance on other homogeneous tasks. As a result, model performances of existing frameworks tend to decrease noticeably as the complexity of class distribution increases.

In this study, we address these problems by proposing a novel, end-to-end deep learning framework, HMD-AMP. We curated a challenging dataset that more closely aligns with the structural diversity of AMPs and non-AMPs. Our architecture comprises the following major components: an embedding layer of protein sequences, a protein language encoder, a feature transformer and a hierarchical deep forest framework making binary classifications (AMP or non-AMP) and multi-label classifications. Our framework is then compared with other state-of-the-art models in a binary task (Task 1: classification of AMP/non-AMP) and a multi-label task (Task 2: classification of effectiveness among 11 possible antimicrobial targets). Our framework outperforms all SOTA models in both tasks. We then evaluate the performance with an ablation study and a reduced feature test, and our findings are robust

against data and feature perturbations.

The rest of this paper is organized as follows. We start by a more thorough overview of our end-to-end framework and an evaluation of the associated 'model problem' and the 'data problem' with previous approaches in greater details in Section 2. We then compare our method side-by-side with many state-of-the-art approaches in section 3, followed by an introduction of our dataset and our experiment results, which include an ablation study, and a sensitivity analysis with reduced model sizes for evaluating the model robustness. Finally, we elaborate the observed limitations and potential future work based on our findings in Section 4 with a short discussion of potential applications.

2 Methods

2.1 Overview of HMD-AMP

HMD-AMP is a supervised machine learning framework consisting of one feature extraction model as well as two prediction models. Given a protein sequence as input, HMD-AMP first extracts its features and then uses the generated features as the inputs to the prediction models. Our prediction models are designed with a two-level prediction strategy including a prediction model that predicts whether a given protein sequence is an AMP, and a second prediction model annotating the AMP's antimicrobial activities [26, 27]. Specifically, HMD-AMP performs feature extraction and function prediction respectively, and it deploys a hierarchical structure of our AMP dataset labeling space. Accordingly, given any sequence analyzed by the HMD-AMP framework, the first model extracts the structural feature. The extracted features are then used as inputs to the prediction model to predict whether the sequence is an AMP or not. Furthermore, if the sequence is predicted as an AMP, the next prediction model predicts the sequence into 11 biological functions (see Figure 1), and biological functions of an AMP are virtually its target groups. This hierarchical framework aims to use the detailed structural information of AMPs to improve the accuracy of prediction model results and to help alleviating the data imbalance problem.

For the feature extraction part of HMD-AMP, the model is an ESM-1b Transformer [28], which takes the raw sequences as inputs. Then, the outputs of the feature extraction model are used as the inputs for the following deep forest [29] model. The structure of the hierarchical classification model is illustrated in Figure 1 (top panel). Section 2.2 elaborates each component of our proposed model structure.

2.2 Deep learning model

At the feature extraction level, the model is an ESM-1b transformer [28]: a transformer-based self-supervised protein language model trained on the UniRef UR50/50 database [30]. ESM-1b processes inputs as character amino acids sequences, using positional embeddings instead of making assumptions on the ordering of the input. From ESM-1b, we obtain residue-level sequence embeddings. In order to get the protein-level embeddings as the inputs to the function prediction model, we average across all residue positions of residue-level sequence embeddings, hence getting a 1280-dimension feature vector for each sequence.

Transformer's [31] self-attention mechanism and its ability to model long-range dependencies, which reflect structural properties of protein sequences, enable themselves to predict amino acid residual contact because the attention maps generated within the Transformer naturally correspond to the information between the various residues in the sequence.

Then, at each function prediction level, the model is a deep forest [29] model, which demonstrates a cascade forest structure. Each level of the cascade receives the feature information processes by its previous level and feeds its outputs to the next level as inputs. Each cascade level is an ensemble of decision tree forests, and different types of forests are included to make the model diverse. In our model design, at each level, we deploy two completely-random tree forests and two random forests: for these forests, the inputs are embeddings obtained from the feature extraction model. Each random forest contains 1000 trees, it randomly selects $\sqrt{d}$ features as candidate features (d being the number of input features) and the one with the best *gini* value is chosen for the split. For our 11 labels in the multi-label classification, we compute their *gini* values as follows:

$$\mathbf{Gini}(p) = \sum_{k=1}^{11} 2p_k(1 - p_k), \tag{1}$$

where p_k is the probability that the sample has the label k. Every complete-random tree forest also contains 1000 completely random trees, which are designed to detect important motifs across the inputs.

To enhance the model's ability to handle feature relationships, a multi-grained scanning procedure is designed as shown in Figure 1 (bottom panel). Specifically, sliding windows scan the given input features. Our inputs are 1280-dimension raw feature vectors, and a window size of 100 is used. For sequence data, a 100-dimension feature vector will be generated by sliding the window for one feature. As a result, a total of 1181 feature vectors are produced for each iteration. Feature vectors extracted from positive/negative training examples are regarded as corresponding instances, and these instances are used to train the forest and then to generate the estimated class distributions, which would be converted to class vectors. Finally, the class vectors are concatenated as transformed features. Take the binary classification task as an example: we have 2 classes, and 1181 2-dimension class vectors are produced by each forest. As a result, the 9448-dimension transformed feature vector is taken as the counterpart of the original 1280-dimension raw feature vector.

In general, each level of the cascade receives feature information processed by its preceding level and feed its processed results to the next level as inputs until there is no significant performance gain, when the training process terminates. This process makes the deep forest appropriately determines the complexity of its model by termination. Also, deep forest does not rely on backpropagation, so it is suitable for training data with either imbalance labels or small sample sizes, hence preventing the model from overfitting. Further, two mechanisms are added to help the deep forest performs well in predicting specific antimicrobial activities. The first mechanism is a measure-aware feature reuse [32]. That is, if the confidence of the current layer is lower than the threshold determined during training, the better representation of the previous layer is partially reused. The confidence to each label is an estimation of the respective label distributions. To accommodate labels with smaller representation, we choose macro-AUC as the confidence-computing metric (Figure 1 top panel). Macro-AUC is a label-based

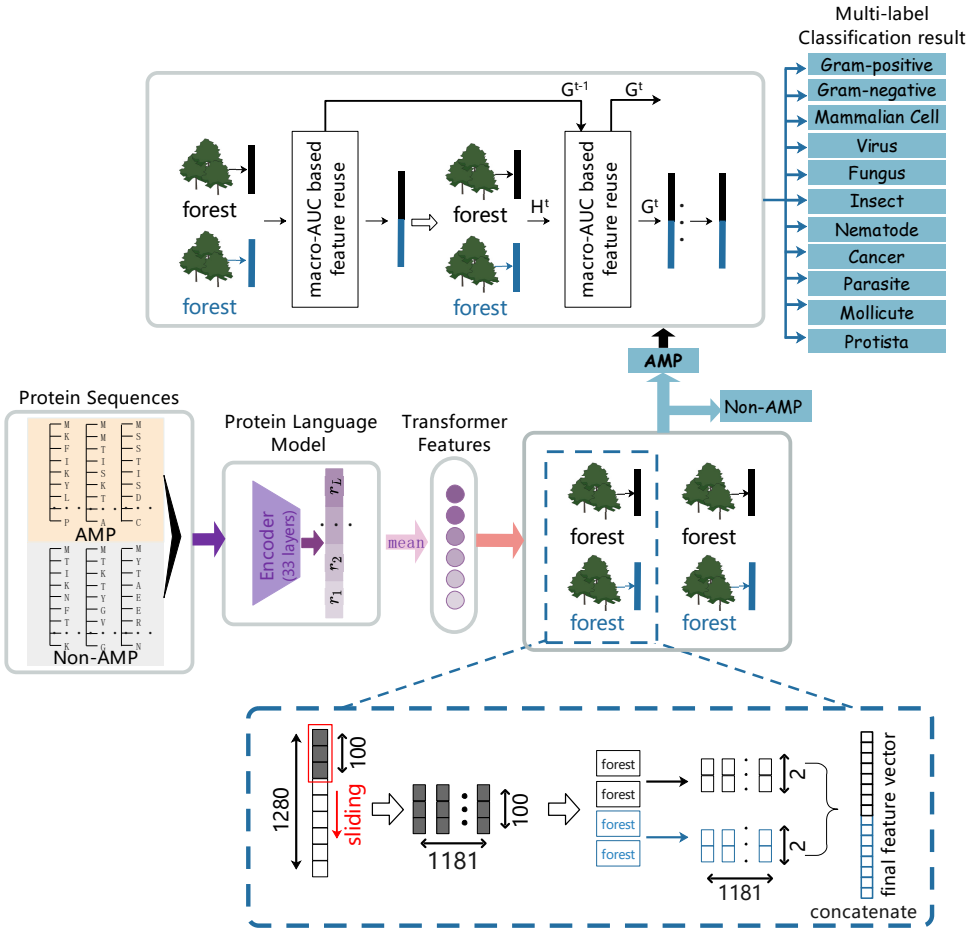

Figure 1: Overview of HMD-AMP. Top panel: HMD-AMP is consisting of one feature extraction model as well as two prediction models. The feature extraction model processes protein sequences into feature vectors. The first prediction model's inputs are feature vectors, and the model predicts a protein is an AMP or not. If the protein is an AMP, the second prediction model predicts its 11 biological functions (multi-label classification). Bottom panel: the way sliding windows scan the features.

measure [33] that is defined as follows:

$$\mathbf{AUC}_{macro}(F) = \frac{1}{l}\sum_{j=1}^{l}\frac{\left|\mathcal{S}_{macro}^{j}\right|}{\left|Y_{\cdot j}^{+}\right|\left|Y_{\cdot j}^{-}\right|}, \tag{2}$$

which is per-class raw average of AUC, where Y is the true label; $Y_{\cdot j}$ is the j-th column of the label matrix, and '+' ('-') is the relevant (irrelevant) note. $\mathcal{S}_{macro}$ is the set of correctly ordered instance pairs on each label:

$$\mathcal{S}_{macro}^{j} = \left\{(a,b) \in Y_{\cdot j}^{+} \times Y_{\cdot j}^{-} \mid f_j\left(x_a\right) \geq f_j\left(x_b\right)\right\}, \tag{3}$$

where the f_{ij} means the confidence score of i-th instance on j-th label,

Therefore, the confidence computing method of each label is shown as Equation 4. Here, *m* refers to the number of sequences and p_{ij} means Pr[$\hat{y}_{ij}$=1].

$$\mathbf{confidence}_{(j)} = \sum_{i=0}^{m}\prod_{k=1}^{i} p_{kj} \prod_{k=i+1}^{m}(1-p_{kj}). \tag{4}$$

The second mechanism is the measure-aware layer growth, which focuses on the learning of representation, and it efficiently enhances the representation through various measures while reducing overfitting and controlling model complexity. For the cascade forest, we artificially set the maximal depth of the layers as 20 in the initialization step. If the model has grown to the maximal number, the training process terminates. In the initialization step, we also initialize the performance vector, which records the performance value on training data in each layer. Here, we still choose macro-AUC as the measure to indicate performance. During each layer t, the forest is fitted according to the training data so that we get this layer's classifier h_t. With the classifier, we predict the representation H_t (Equation 5), where X is the training data, and G_{t-1} is the representation of the last layer. Then we obtain the new representation of the current layer by measure-aware feature reuse. Due to the layer growth being measure-aware, the model needs to compute the macro-AUC after fitting each layer. When the measure is not getting better in recent three layers, an early stopping mechanism forces the deep forest to stop growing even if it yet reaches the maximum depth. At the same time, the best performance layer index is recorded. Therefore, with such well-designed mechanisms, multi-label deep forest [32] is very appropriate for solving multi-label problems.

$$\mathrm{H}_t = h_t\left(\left[\mathrm{X}, \mathrm{G}_{t-1}\right]\right). \tag{5}$$

3 Results

In this section, we analyze our model in comparison with several published, state-of-the-art methods and train them with our dataset. In particular, we start by describing the different settings of the experiments and evaluating our proposed model compared to these SOTA models.

3.1 Alternative methods

We first introduce the benchmark result produced by Veltri et al. [25] combining CNN and LSTM for antimicrobial recognition. In addition, the development of meta learning makes Model-Agnostic Meta-Learning (MAML) [34] a suitable candidate. Instead of learning a model that can be used directly for prediction, such meta-learning methods learn how to learn a model faster and better instead. Meanwhile, we include the *Probabilistic Model-Agnostic Meta-Learning* [35], an extension of the original MAML, as another candidate. We encourage interested readers to refer to their initial manuscripts for details beyond our brief summary. MAML can be interpreted as approximate inference for the posterior [36]. It uses the *maximum a posteriori* (MAP) value. The algorithm evaluates the variational lower-bound for the logarithm of the approximate likelihood, which can be written as

$$\log p\left(\mathbf{y}_i^{test} \mid \mathbf{x}_i^{test}, \mathbf{x}_i^{\mathrm{tr}}, \mathbf{y}_i^{tr}\right) \geq E_{\theta\sim q_\psi}\left[\log p\left(\mathbf{y}_i^{test} \mid \mathbf{x}_i^{test}, \phi_i^{\star}\right) + \log p(\theta)\right] + \mathcal{H}\left(q_\psi\left(\theta \mid \mathbf{x}_i^{test}, \mathbf{y}_i^{test}\right)\right). \tag{6}$$

In this bound, it essentially performs approximate inference via MAP on ϕ_i to obtain $p\left(\phi_i \mid \mathbf{x}_i^{\mathrm{tr}}, \mathbf{y}_i^{\mathrm{tr}}, \theta\right)$, and uses the variational distribution for θ only. Then, the inference network is given by

$$q_\psi\left(\theta \mid \mathbf{x}_i^{test}, \mathbf{y}_i^{test}\right) = \mathcal{N}\left(\mu_\theta + \gamma_q \nabla \log p\left(\mathbf{y}_i^{test} \mid \mathbf{x}_i^{test}, \mu_\theta\right); \mathbf{v}_q\right). \tag{7}$$

Table 1: Target groups and the number of peptides in each category in our positive dataset.

Activity	No of peptides
Gram-positive	11486
Gram-negative	11958
Mammalian Cell	7403
Virus	3779
Fungus	5514
Insect	181
Cancer	2271
Parasite	405
Mollicute	31
Nematode	34
Protista	41

Table 2: Peptides' label amount in our positive dataset.

No of labels	No of peptides
1	6149
2[a]	4246
3	4861
4	2480
5	711
6	65
7	2

[a] Here 2 means a peptide has 2 labels (biological functions)

The training is performed by backpropagating gradients, and this process includes a term for the likelihood $\log p\left(\mathbf{y}_i^{test} \mid \mathbf{x}_i^{test}, \mathbf{x}^{\mathrm{tr}}, \mathbf{y}^{tr}, \phi_i^{\star}\right)$ and the KL-divergence between the sample $\theta \sim q_\psi$ and the prior $p(\theta)$. We try to implement both MAML and PMAML against CNN-LSTM [25] benchmark. Indeed, the parameters randomly initialized by PMAML are hard to train, and such PMAML model only performs well if we use parameters obtained from the trained MAML model for the training of PMAML.

Another approach to highlight is the AMAP [37], which is a hierarchical multi-label prediction model that annotates the biological functions of AMP sequences, using extreme Gradient Boosting (XGBoost) [38]. XGBoost is based on boosted trees, and it learns by minimizing the objective function:

$$L(\Phi) = \sum_i l\left(\hat{\mathrm{y}}_\mathrm{i}, \mathrm{y}_i\right) + \sum_k \Omega\left(f_k\right), \tag{8}$$

where $\Omega\left(f_k\right) = \gamma T + \frac{\lambda}{2}\left\|u_k\right\|$. Here, $l\left(\hat{\mathrm{y}}_\mathrm{i}, \mathrm{y}_i\right)$ is the loss function of predicted model output $\hat{\mathrm{y}}_\mathrm{i}$ and actual output y_i for all examples. $\Omega\left(f_k\right)$ is a regularization function that is based on the number of trees T and the norm of the vector of scores u at the k-leaf of the trees. The regularization parameters γ and λ control the relative contribution of the two regularization factors in contrast to the minimization of the loss function. We include AMAP in our comparisons in Section 3.4.

3.2 Datasets

We compile a comprehensive multi-label AMP database with a high degree of confidence. Specifically, we collect and clean amino acids sequences from three published AMP databases: Database of Antimicrobial Activity and Structure of Peptides (DBAASP) [12], an update to LAMP database linking antimicrobial peptide (LAMP2) [39], and data repository of antimicrobial peptides (DRAMP) [40]. Then, we remove the identical and duplicate sequences from our database. The resulting database is composed of 18514 high-quality sequences, coupled with labels of 11 antimicrobial activities classes (Table 1 and Table 2). In the design of our machine learning models, these sequences are taken as positive examples.

We extract 12659 peptides with the highest BLAST [41] similarity scores against the AMPs in our multi-label AMP database from Uniprot [42] and these peptides show no antimicrobial activity. Specifically, to avoid sequence and composition biases that affect our machine learning, we filter peptides using an approach similar to previous work [25] and remove all peptides with more than 40% sequence identity to each other using CDHIT [43]. It leaves a total of 8534 peptides and we use these as negative peptides to train and evaluate our model. We use 5-fold stratified cross-validation to evaluate the performance of our model, the dataset would randomly be divided into five folds, and at each time, four of them are chosen for the model training and the remaining one fold is used to test the trained model. As a result, average results are generated from repeating the above procedure five times.

3.3 Implementation details

We use the Tensorflow toolkit to write our code and train HMD-AMP with 2 NVIDIA GeForce RTX 3090 GPUs. We train the first deep forest model on the whole dataset. Because our negative set bears a strong resemblance to the

Table 3: The AMP/non-AMP classification results between different methods

	Accuracy	Precision[a]	Recall	F1-score
AMAP	0.882	0.882	0.851	0.863
DL	0.956	0.951	0.944	0.947
MAML+DL	0.941	0.935	0.939	0.937
PMAML+DL	**0.961**	**0.957**	0.951	0.954
HMD-AMP	0.956	0.955	**0.953**	**0.954**

[a] The precision, recall, and F1-score are the macro averages over the AMP and Non-AMP

positive set, deep forest is forced to learn to be a more powerful model. Then, we perform the next level multi-label deep forest model with our positive set, the dataset contains more than 18000 AMP sequences, and each of them has 11 labels, indicating which target groups the sequence resists. When training the models, we first input the protein sequence directly into ESM-1b model, and for each sequence, we can obtain a 1280-dimension embedding vector, which is the result of averaging across all residue positions of residue-level sequence embeddings. Such an embedding vector is then fed into a deep forest model, with two random forests and two complete-random tree forests with 1000 completely random trees for the binary classification. If one sequence is predicted to be an AMP, the multi-label deep forest annotates its target groups, this model has the same tree structure with the binary classification model.

3.4 Performance comparison

Here, we evaluate the models' performance we proposed above. We refer to Veltri et al.'s work [25] as DL, and DL model trained by MAML [34] and PMAML [35] algorithms are called MAML+DL and PMAML+DL respectively. We use a 5-fold stratified cross-validation to evaluate the performance of HMD-AMP. In this experiment, we randomly divide our dataset into five folds. Each time, we choose four folds from the dataset for the model training and test the trained model on the remaining one. To avoid data bias, average results are generated from repeating the above procedure five times.

3.4.1 Binary classification performance comparison

As shown in Table 3, for the AMP/non-AMP classification, HMD-AMP shows higher recall (0.953) and F1-score (0.954) with promising accuracy (0.956) and precision (0.955) than existing methods. One observation is that deep network methods generally have better performances on binary (AMP/non-AMP) classification than AMAP. One of the reasons is when there is enough data for training, deep models can often fit better functions for prediction, whereas AMAP is considered a traditional machine learning model.

3.4.2 Multi-label classification performance comparison

HMD-AMP beats the state-of-the-art results (as shown in Table 4 and Figure 2) in this more challenging task. HMD-AMP significantly outperforms across all measures including accuracy, precision, recall, and macro-AUC. Despite all methods having relatively high accuracy (still far behind the HMD-AMP) on the AMP biological functions classification, the simple DL method cannot correctly predict labels with only a small amount of data (Figure 3), and our HMD-AMP outperforms DL by more than 30% on both precision and recall scores. For example, in our dataset, only 41 peptides have resistance to Protista, DL has little effect on the classification of this

Table 4: The AMP biological functions (multi-label) classification results between different methods

	Accuracy	Precision[a]	Recall	macro-AUC
AMAP	0.913	0.704	0.695	0.859
DL	0.904	0.536	0.451	0.844
MAML+DL	0.909	0.794	0.671	0.861
PMAML+DL	0.927	0.846	0.748	0.897
HMD-AMP	**0.958**	**0.887**	**0.812**	**0.915**

[a] The precision and recall are the macro averages over the 11 biological functions classes

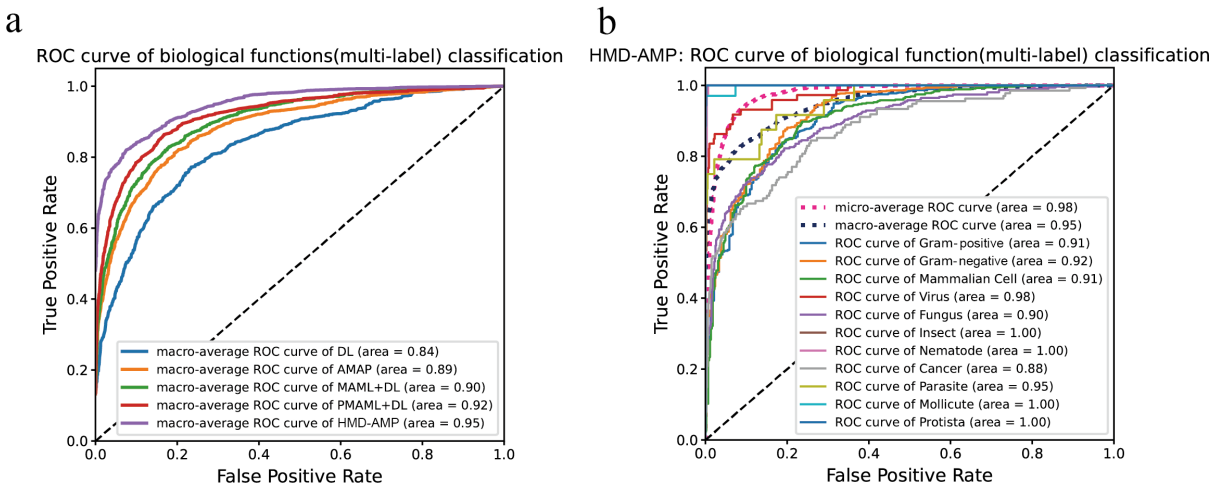

Figure 2: **a**: macro-average ROC curves comparison of 5 models. **b**: ROC curves of our model, including macro/micro-average and 11 biological function labels.

label, leading to many false negatives. Actually, both DL and MAML+DL have a very inconsistent performance across different classes, especially in terms of recall. In contrast, HMD-AMP is quite stable across different classes regardless of precision or recall (Figure 3). For AMAP, its two types of sequence-based feature representation help the method performs well on multi-label classification. Nevertheless, our method outperforms AMAP by about 10% on both precision and recall score. Although PMAML+DL has relatively good performance on 4 evaluation metrics (still about 5% behind our method on both precision and recall), it is largely based on parameters derived from the trained MAML+DL. If randomly initialized parameters are used, it would be difficult for PMAML+DL to obtain good performance.

3.5 Ablation study

With the aforementioned experiments validating the strengths of our model in multi-label classification, we develop a new experiment to see if our method works with very little data. We select 50, 100, 200 data points from our positive dataset to train HMD-AMP, MAML+DL, and PMAML+DL, and see whether our model still delivers consistent performance. We make sure that the selected data contain positive and negative data for all 11 labels. Again, every model is validated by using a 5-fold cross-validation test reporting the performance averaged over five trials, where each trial leaves out a different 20% of the selected data as a test set to validate the performance of the model trained on the other 80% of the selected data. It is worth mentioning that our method works well even with little data: we show the result in Table 5. When the number of data points is 200, our method outperforms the other two models on accuracy (0.928), precision (0.841), and macro-AUC (0.916). It is not surprising that PMAML has a slightly better recall score (0.821), because PMAML was designed to solve few-shot problems and it can be trained to converge at great speed. When the number of data points is 100, our method outperforms the other two models on accuracy (0.925) and macro-AUC (0.860), and PMAML gets the best score on precision (0.810) and recall (0.780), and when the number of data points is 50, PMAML shows slightly higher performance on all 4 measures. Although our approach is slightly behind PMAML, it still shows favorable performance and is superior to MAML across all metrics. The result indicates that our method is also suitable for the small sample problem.

To evaluate the effectiveness of the feature extraction model and the prediction model respectively, we conduct the model replacement test. To investigate the efficiency of the feature extraction model, we apply deep forest and take the raw representation of the sequence, i.e., one-hot encoding, as inputs. In this experiment, we name it Deep Forest. Further, to analyze our prediction model's importance, we change the prediction model into the random forest [44] and retain our feature extraction model. The random forest has 1000 trees. Likewise, we name this method as Random Forest. Table 6 and Figure 4a show the experimental results of HMD-AMP, Random Forest, and Deep Forest, all three methods have high accuracy, but Random Forest has low precision and recall scores. Through inspection, we find that Random Forest could hardly recognize some labels with unbalanced data (such as Protista and Mollicute): its correct predictions of few true positive cases lead to a large number of false negatives. Even a small number of false positives could result in low precision. Besides, we find that Deep Forest performs well on unbalanced labels, which is due to the large size of trees in forests (All trees in the deep forest are averaged

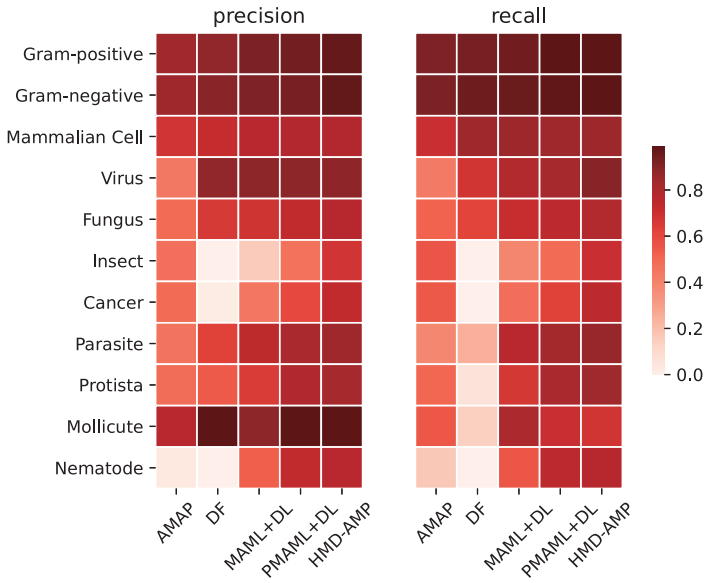

Figure 3: Detailed prediction performance comparison on each biological function label.

to generate an estimate of the distribution of classes) and the measure-aware layer growth mechanism. However, HMD-AMP has been consistently ranked as the best performance across all the measures, which indicates that our feature extraction model provides effective features. These features help deep forest using more suitable motifs to make the prediction.

3.6 Reduced feature model analysis

To verify the similarity of input features with exactly same labels, we perform visual processing on the feature vectors. We apply t-SNE [45], which is a non-linear cluster recognition algorithm, for data dimensionality reduction. T-SNE finds similarity patterns in data points with multiple features, and we can see data points with high similarity converge into a cluster by projecting dimensionality reduction results onto a two-dimensional plane. We use t-SNE to reduce a 1280-dimension feature vector to a 2-dimension vector. For each label combination, we assign different numbers and use the numbers to distinguish the AMP of different labels. Some AMP sequences from our dataset are selected, and the result is shown in Figure 4b. From the result, we clearly see differentiated clusters, which indicates the feature extraction model nicely mines the structural and functional features inside the protein sequences.

However, 1280-dimension vector features cause time-burden and memory-consumption, we try to reduce features' amount and choose valid features. we adopt Local Interpretable Model-agnostic Explanations (LIME)

Table 5: sensitivity analysis results

		Accuracy[c]	Precision[b]	Recall	macro-AUC
50[a]	MAML+DL	0.836	0.725	0.686	0.759
	PMAML+DL	**0.861**	**0.793**	**0.782**	**0.824**
	HMD-AMP	0.849	0.767	0.741	0.811
100	MAML+DL	0.910	0.744	0.701	0.807
	PMAML+DL	0.911	**0.810**	**0.780**	0.831
	HMD-AMP	**0.925**	0.804	0.769	**0.860**
200	MAML+DL	0.903	0.787	0.739	0.853
	PMAML+DL	0.907	0.837	**0.821**	0.874
	HMD-AMP	**0.928**	**0.841**	0.802	**0.916**

[a] 50, 100, and 200 means the number of data points used for training and testing.
[b] The precision and recall are the macro averages over the 11 biological functions classes.
[c] Every model is validated by using a 5-fold cross-validation test, and the performance averaged over five trials.

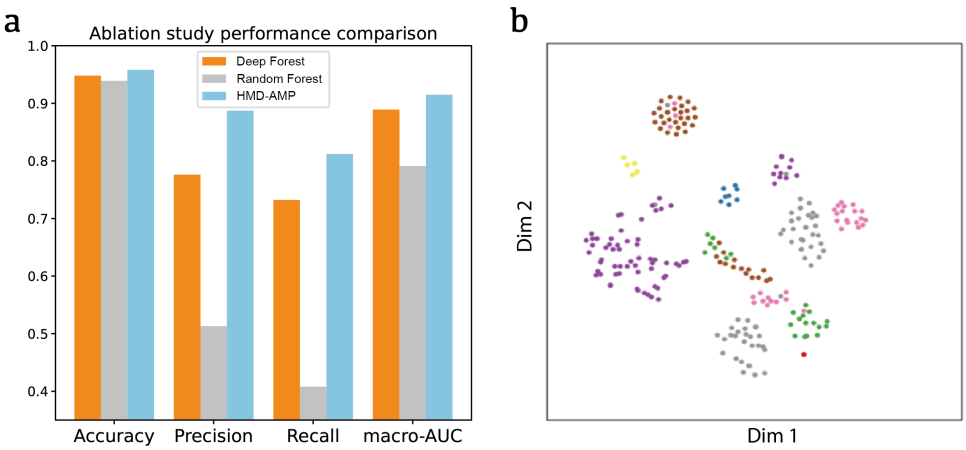

Figure 4: **a**: Ablation study of the feature extraction model and prediction model. **b**: A 2D t-SNE projection. Different colors represent AMP's different label-combination, and data points in the same cluster share a high similarity.

[46]. LIME is an explanation technique that explains the prediction of our classifier in an interpretable and faithful manner by learning an interpretable model locally around the prediction. LIME gets each feature's global weight (Figure B.1) by averaging the value of local prediction, and we selects 48 features with the highest global weight (Figure B.2) as new features to train our prediction model. Despite using only 48 features, HMD-AMP's performance still outperforms AMAP, DL, and MAML+DL on both the AMP/non-AMP classification and the AMP biological functions classification. Reduced features (48 features) model performance has a slight drop of about 0.7% on accuracy, 0.8% on precision and recall (Table 7) compared with HMD-AMP using 1280 features on the AMP/non-AMP classification, but it greatly improves the training speed.

3.7 Model application

We apply our model to 20 newly synthesized antimicrobial peptides candidate sequences (Table A.1) [24]. In fact, only two of these sequences (YI12 and FK13) were resistant to gram-positive and gram-negative, while the other 18 were not antimicrobial peptides. We use HMD-AMP to predict 20 peptides, and find that YI12 and FK13 are predicted to have resistance to gram-positive and gram-negative. Besides, 8 of the non-AMP peptides are identified by HMD-AMP, in contrast to the method [24] recognizing all 20 peptides as AMP. Our HMD-AMP consistently outperforms methods like AmpGram [47], AMPA [48], AMPScanner [25], and CAMP [49], which predict YI12 as non-AMP, and AMAP [37], which predicts more non-AMP as AMP than our method.

Another interesting observation is that FK13 has the highest gram-positive (0.893, rank:1) and gram-negative (0.930, rank:1) probability among the 12 peptides that HMD-AMP predicted as AMPs, YI12 also has promising probability on gram-positive (0.799, rank:5), gram-negative (0.829, rank:4). The result shows that our model not only predicts the existing AMP, but also carries out functional annotations for AMPs that are newly synthetic or have yet existed.

Table 6: ablation test results

	Accuracy	Precision[a]	Recall	macro-AUC
Deep Forest	0.948	0.776	0.732	0.889
Random Forest	0.939	0.513	0.408	0.791
HMD-AMP	**0.958**	**0.887**	**0.812**	**0.915**

[a] The precision and recall are the macro averages over the 11 biological functions classes.

Table 7: Reduced feature model performance

	senario of classification	accuracy[c]	precision[d]	recall	F1-score
HMD-AMP	binary[a]	0.951	0.947	0.945	0.946
	multi-label[b]	0.915	0.806	0.744	0.774

[a] The AMP/non-AMP classification
[b] The AMP biological functions classification
[b] The model is validated by using a 5-fold cross-validation test, and the performance averaged over five trials
[d] The precision and recall are the macro averages of the AMP and Non-AMP/the 11 biological functions classes.

4 Conclusion

We develop a hierarchical method, HMD-AMP, to facilitate the detection of antimicrobial peptides, providing detailed AMPs' biological functions annotations. Comprehensive experiments including cross-fold validation, sensitivity test, ablation study, and new peptides test, demonstrate the effectiveness and robustness of our proposed method, where our model consistently and significantly outperforms all counterparts. Most known AMP prediction methods only classify sequences into AMPs/non-AMPs, and methods that predict AMPs' biological functions are mostly based on statistics, sequence comparison, and traditional machine learning models. However, former methods' performances are worse than our model, they don't recognize labels with small amounts of data well, leading to very low recall.

Moreover, since the deep learning model trained on meta-learning algorithms performs well on little data, and it is developed to help the model learn faster in different scenarios, extending the model to do multi-tasks is a promising research direction. In the future, we will try to combine our method with the meta-learning algorithm and make it able to classify protein sequences into more groups, not limited to AMP. Also, the 20 peptides classification result in section 'Model application' enlightens us to further develop HMD-AMP as a generative model, which can sample large amounts of peptides and generate new AMPs.

We believe that HMD-AMP can serve as a powerful tool to promote the application of antimicrobial peptides and alleviate the global threat of antibiotic resistant genes. In the future, we will incorporate other dimensions of information, such as 3D structural information, into our framework to further improve our method's performance and extend the application scenarios.

References

[1] Mwangi, J. *et al.* The antimicrobial peptide zy4 combats multidrug-resistant pseudomonas aeruginosa and acinetobacter baumannii infection. *Proceedings of the National Academy of Sciences* **116**, 26516–26522 (2019).

[2] Thapa, R. K., Diep, D. B. & Tønnesen, H. H. Topical antimicrobial peptide formulations for wound healing: Current developments and future prospects. *Acta biomaterialia* **103**, 52–67 (2020).

[3] Elnagdy, S. & AlKhazindar, M. The potential of antimicrobial peptides as an antiviral therapy against covid-19. *ACS Pharmacology & Translational Science* **3**, 780–782 (2020).

[4] Rafii, F., Sutherland, J. B. & Cerniglia, C. E. Effects of treatment with antimicrobial agents on the human colonic microflora. *Therapeutics and clinical risk management* **4**, 1343 (2008).

[5] Price, L. B. *et al.* Staphylococcus aureus cc398: host adaptation and emergence of methicillin resistance in livestock. *MBio* **3**, e00305–11 (2012).

[6] Solomon, S. L. & Oliver, K. B. Antibiotic resistance threats in the united states: stepping back from the brink. *American family physician* **89**, 938–941 (2014).

[7] Organization, W. H. *et al. Antimicrobial resistance: global report on surveillance* (World Health Organization, 2014).

[8] Saha, S. *et al.* Increasing antibiotic resistance in clostridioides difficile: a systematic review and meta-analysis. *Anaerobe* **58**, 35–46 (2019).

[9] de Breij, A. *et al.* The antimicrobial peptide saap-148 combats drug-resistant bacteria and biofilms. *Science translational medicine* **10** (2018).

[10] Kang, J., Dietz, M. J. & Li, B. Antimicrobial peptide ll-37 is bactericidal against staphylococcus aureus biofilms. *PLoS One* **14**, e0216676 (2019).

[11] Makowski, M., Silva, Í. C., Pais do Amaral, C., Gonçalves, S. & Santos, N. C. Advances in lipid and metal nanoparticles for antimicrobial peptide delivery. *Pharmaceutics* **11**, 588 (2019).

[12] Pirtskhalava, M. *et al.* Dbaasp v. 2: an enhanced database of structure and antimicrobial/cytotoxic activity of natural and synthetic peptides. *Nucleic acids research* **44**, D1104–D1112 (2016).

[13] Zhao, X., Wu, H., Lu, H., Li, G. & Huang, Q. Lamp: a database linking antimicrobial peptides. *PloS one* **8**, e66557 (2013).

[14] Thomas, S., Karnik, S., Barai, R. S., Jayaraman, V. K. & Idicula-Thomas, S. Camp: a useful resource for research on antimicrobial peptides. *Nucleic acids research* **38**, D774–D780 (2010).

[15] Wang, G., Li, X. & Wang, Z. Apd3: the antimicrobial peptide database as a tool for research and education. *Nucleic acids research* **44**, D1087–D1093 (2016).

[16] Seshadri Sundararajan, V. *et al.* Dampd: a manually curated antimicrobial peptide database. *Nucleic acids research* **40**, D1108–D1112 (2012).

[17] Das, P. *et al.* Pepcvae: Semi-supervised targeted design of antimicrobial peptide sequences. *arXiv preprint arXiv:1810.07743* (2018).

[18] Lata, S., Mishra, N. K. & Raghava, G. P. Antibp2: improved version of antibacterial peptide prediction. *BMC bioinformatics* **11**, 1–7 (2010).

[19] Randou, E. G., Veltri, D. & Shehu, A. Binary response models for recognition of antimicrobial peptides. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, 76–85 (2013).

[20] Xiao, X., Wang, P., Lin, W.-Z., Jia, J.-H. & Chou, K.-C. iamp-2l: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Analytical biochemistry* **436**, 168–177 (2013).

[21] Fjell, C. D. *et al.* Identification of novel antibacterial peptides by chemoinformatics and machine learning. *Journal of medicinal chemistry* **52**, 2006–2015 (2009).

[22] Bhadra, P., Yan, J., Li, J., Fong, S. & Siu, S. W. Ampep: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest. *Scientific reports* **8**, 1–10 (2018).

[23] Witten, J. & Witten, Z. Deep learning regression model for antimicrobial peptide design. *BioRxiv* 692681 (2019).

[24] Das, P. *et al.* Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nature Biomedical Engineering* **5**, 613–623 (2021).

[25] Veltri, D., Kamath, U. & Shehu, A. Deep learning improves antimicrobial peptide recognition. *Bioinformatics* **34**, 2740–2747 (2018).

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

[27] Li, Y. *et al.* Hmd-arg: hierarchical multi-task deep learning for annotating antibiotic resistance genes. *Microbiome* **9**, 1–12 (2021).

[28] Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv* (2019). URL https://www.biorxiv.org/content/10.1101/622803v4.

[29] Zhou, Z.-H. & Feng, J. Deep forest. *arXiv preprint arXiv:1702.08835* (2017).

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

[31] Vaswani, A. *et al.* Attention is all you need. In *Advances in neural information processing systems*, 5998–6008 (2017).

[32] Yang, L., Wu, X.-Z., Jiang, Y. & Zhou, Z.-H. Multi-label learning with deep forest. *arXiv preprint arXiv:1911.06557* (2019).

[33] Wu, X.-Z. & Zhou, Z.-H. A unified view of multi-label performance measures. In *International Conference on Machine Learning*, 3780–3788 (PMLR, 2017).

[34] Finn, C., Abbeel, P. & Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 1126–1135 (PMLR, 2017).

[35] Finn, C., Xu, K. & Levine, S. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817* (2018).

[36] Grant, E., Finn, C., Levine, S., Darrell, T. & Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930* (2018).

[37] Gull, S., Shamim, N. & Minhas, F. Amap: Hierarchical multi-label prediction of biologically active and antimicrobial peptides. *Computers in biology and medicine* **107**, 172–181 (2019).

[38] Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794 (2016).

[39] Ye, G. *et al.* Lamp2: a major update of the database linking antimicrobial peptides. *Database* **2020** (2020).

[40] Shi, G. *et al.* Dramp 3.0: an enhanced comprehensive data repository of antimicrobial peptides. *Nucleic Acids Research* (2021).

[41] Madden, T. The blast sequence analysis tool. *The NCBI handbook* **2**, 425–436 (2013).

[42] Consortium, U. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research* **47**, D506–D515 (2019).

[43] Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152 (2012).

[44] Breiman, L. Random forests. *Machine learning* **45**, 5–32 (2001).

[45] Van der Maaten, L. & Hinton, G. Visualizing data using t-sne. *Journal of machine learning research* **9** (2008).

[46] Ribeiro, M. T., Singh, S. & Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144 (2016).

[47] Burdukiewicz, M. *et al.* Proteomic screening for prediction and design of antimicrobial peptides with ampgram. *International journal of molecular sciences* **21**, 4310 (2020).

[48] Torrent, M. *et al.* Ampa: an automated web server for prediction of protein antimicrobial regions. *Bioinformatics* **28**, 130–131 (2012).

[49] Waghu, F. H., Barai, R. S., Gurung, P. & Idicula-Thomas, S. Campr3: a database on sequences, structures and signatures of antimicrobial peptides. *Nucleic acids research* **44**, D1094–D1097 (2016).

[50] DeLano, W. L. *et al.* Pymol: An open-source molecular graphics tool. *CCP4 Newsletter on protein crystallography* **40**, 82–92 (2002).

Appendix

A Structure investigation of the novel AMPs

We draw the 3D structure graphs of YI12 and FK13, and find that 'YLR' subsequence in YI12 and 'WLK' subsequence in FK13 (stick model in Figure A.1) can be aligned, and the RMSD (Root-mean-square deviation of atomic positions) score is 0.111. We suspect that these three loci make two sequences resistant to gram-positive and gram-negative. However, many existing loci prediction tools failed to predict YI12 as an AMP, and we are unable to validate the above conjecture with the existing computational tools. In the future, we will try to develop more accurate generative models and loci prediction models based on our proposed method, discovering new AMPs and identifying their functional loci.

Table A.1: Sequences of the 20 peptides.

Description	Sequence
YI12	**YLRLIRYMAKMI**
FK13	**FPLTWLKWWKWKK**
-	HILRMRIRQMMT
-	ILLHAILGVRKKL
-	YRAAMLRRQYMMT
-	HIRLMRIRQMMT
-	HIRAMRIRAQMMT
-	KTLAQLSAGVKRWH
-	HILRMRIRQGMMT
-	HRAIMLRIRQMMT
-	EYLIEVRESAKMTQ
-	GLITMLKVGLAKVQ
-	YQLLRIMRINIA
-	VRWIEYWREKWRT
-	LIQVAPLGRLLKRR
-	YQLRLIMKYAI
-	HRALMRIRQCMT
-	GWLPTEKWRKLC
-	YQLRLMRIMSRI
-	LRPAFKVSK

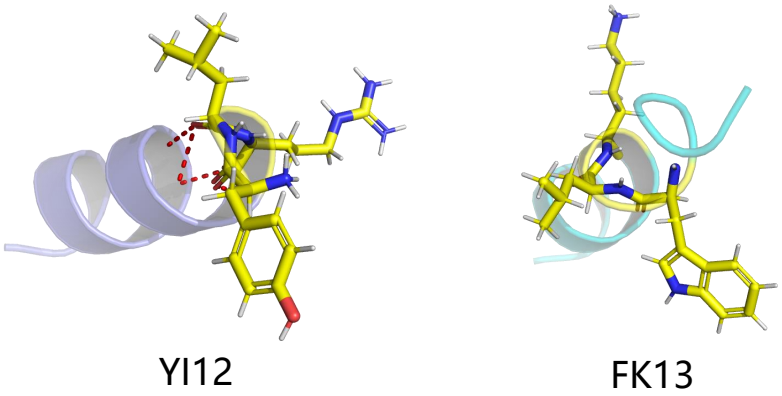

Figure A.1: 3D structure graphs of YI12 and FK13 generated by PyMOL [50]

B Feature analysis

We adopt Local Interpretable Model-agnostic Explanations (LIME) [46]. And LIME is an explanation technique that explains the prediction of our classifier in an interpretable and faithful manner by learning an interpretable model locally around the prediction. We obtain 1280 features' effect on the prediction (global weights) by using the LIME framework (Figure B.1). And those global weights of features are predicted based on the average value of local prediction. In fact, the main function of LIME framework is to find features that have the most positive impact on the model. And these selected features could help the model better fit the given data. We select 48 features (Figure B.2) with the highest weight, and test whether the model trained with these features performs well or not.

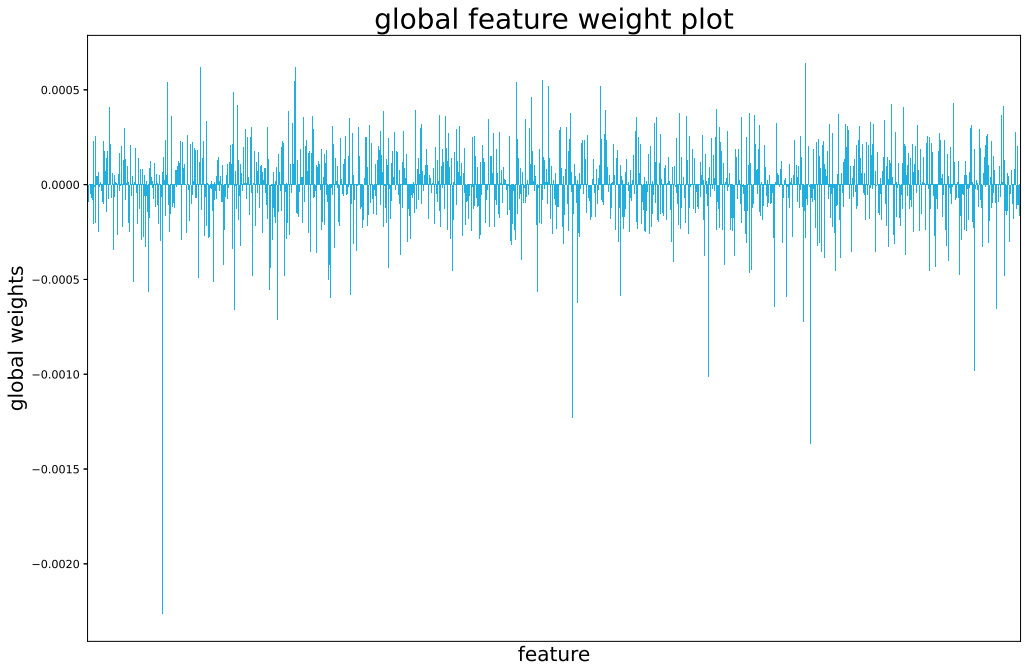

Figure B.1: Global weights of 1280 features. We get 1280 features' global weights in the classification task by applying LIME. Features with weights greater than 0 have positive effects on prediction, while features with weights less than 0 have negative effects.

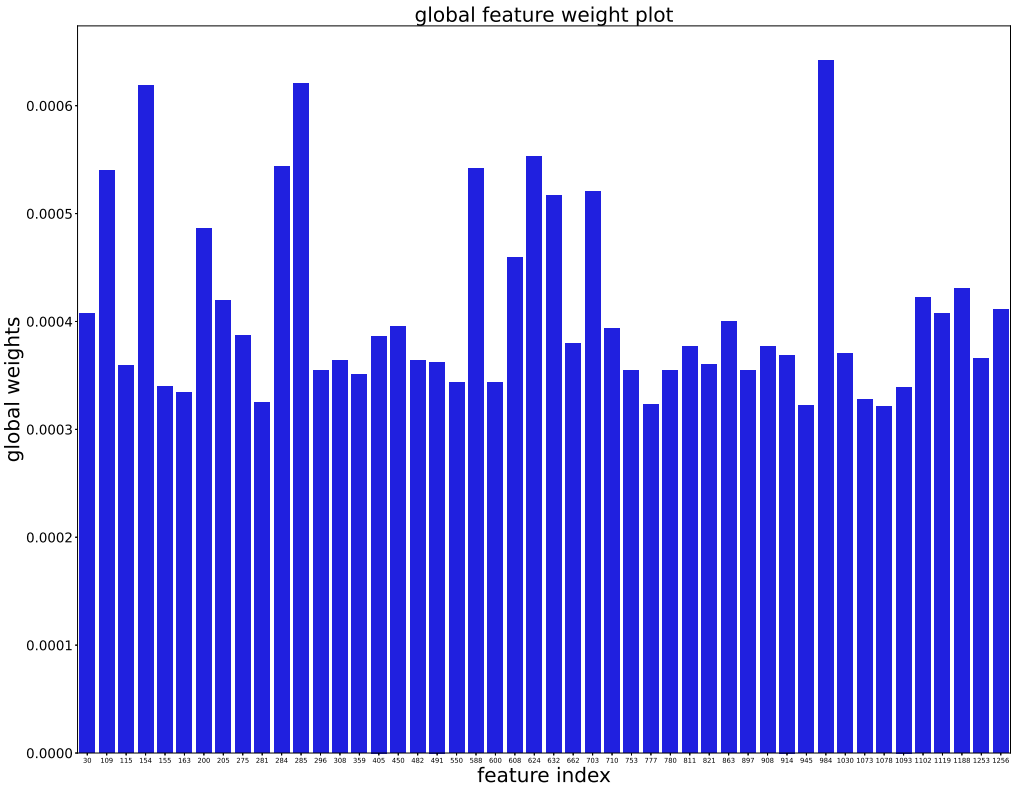

Figure B.2: We select 48 features with the highest weight among 1280 features. The X-axis shows indexes of 48 features. These features are used to train our function prediction model, in order to achieve faster training speed and less computational resource consumption.