OXFORD

Briefings in Functional Genomics, 2024, **23**, 879–890

https://doi.org/10.1093/bfgp/elae046

Advance access publication date 22 November 2024

Protocol Article

SAMP: Identifying antimicrobial peptides by an ensemble learning model based on proportionalized split amino acid composition

Junxi Feng[1,†], Mengtao Sun[2,†], Cong Liu[3], Weiwei Zhang[4], Changmou Xu[5], Jieqiong Wang[6], Guangshun Wang[4], Shibiao Wan[2,*]

[1]Department of Biostatistics, School of Public Health, Harvard University, Boston, MA 02115, United States
[2]Department of Genetics, Cell Biology and Anatomy, College of Medicine, University of Nebraska Medical Center, Omaha, NE 68198, United States
[3]Department of Mathematics, Data Science, University of Waterloo, Waterloo, ON N2L3G1, Canada
[4]Department of Pathology, Microbiology, and Immunology, College of Medicine, University of Nebraska Medical Center, Omaha, NE 68198, United States
[5]Department of Food Science and Human Nutrition, College of Agricultural, Consumer and Environmental Sciences, University of Illinois Urbana-Champaign, Urbana, IL 61801, United States
[6]Department of Neurological Sciences, College of Medicine, University of Nebraska Medical Center, Omaha, NE 68198, United States

*Corresponding author. Department of Genetics, Cell Biology and Anatomy, College of Medicine, University of Nebraska Medical Center, Omaha, NE 68198, United States. E-mail: swan@unmc.edu

†Junxi Feng and Mengtao Sun contributed equally to this work, and they should be regarded as co-first author.

Abstract

It is projected that 10 million deaths could be attributed to drug-resistant bacteria infections in 2050. To address this concern, identifying new-generation antibiotics is an effective way. Antimicrobial peptides (AMPs), a class of innate immune effectors, have received significant attention for their capacity to eliminate drug-resistant pathogens, including viruses, bacteria, and fungi. Recent years have witnessed widespread applications of computational methods especially machine learning (ML) and deep learning (DL) for discovering AMPs. However, existing methods only use features including compositional, physiochemical, and structural properties of peptides, which cannot fully capture sequence information from AMPs. Here, we present SAMP, an ensemble random projection (RP) based computational model that leverages a new type of feature called proportionalized split amino acid composition (PSAAC) in addition to conventional sequence-based features for AMP prediction. With this new feature set, SAMP captures the residue patterns like sorting signals at both the N-terminal and the C-terminal, while also retaining the sequence order information from the middle peptide fragments. Benchmarking tests on different balanced and imbalanced datasets demonstrate that SAMP consistently outperforms existing state-of-the-art methods, such as iAMPpred and AMPScanner V2, in terms of accuracy, Matthews correlation coefficient (MCC), G-measure, and F1-score. In addition, by leveraging an ensemble RP architecture, SAMP is scalable to processing large-scale AMP identification with further performance improvement, compared to those models without RP. To facilitate the use of SAMP, we have developed a Python package that is freely available at https://github.com/wan-mlab/SAMP.

Keywords: Antimicrobial peptides; Proportionalized split amino acid composition; Random projection; Ensemble learning; SAMP

Introduction

Antibiotics are a remarkable medication that has saved thousands of lives by defeating various infectious diseases [1–6]. However, the long-term and rapid increase of antibiotic use for disease treatment in large populations has resulted in drug resistance in pathogens [6–10]. According to the World Health Organization (WHO), approximately 700,000 patients worldwide die from drug-resistant bacterial infections every year, and the total number of deaths is predicted to increase to 10 million by 2050 [11]. Therefore, expanding a large range of new antimicrobial agents to fight against pathogens is essential to relieve the huge burden of global health [12].

Antimicrobial peptides (AMPs) are amino-acid-based oligomers, naturally widespread in all forms of life, such as bacteria, animal, and plant [13–15]. They served as the first line of defense against pathogens by interrupting pathogen-associated molecular processes [16–20]. AMPs, with their effectiveness against multi-resistant bacteria, fungi, parasites, and viruses, are promising for developing new antibiotics. [21–25]. However, the discovery of natural AMPs often depends on traditional wet-lab experiments that are time-consuming and labor-intensive. To streamline this process, developing in-silico predictive models to identify potential AMP candidates can facilitate a more efficient and convenient selection process before proceeding to synthesis and wet lab testing. In the past decade, numerous computational models based on various algorithms, such as support vector machine (SVM) [26], random forest (RF) [27], and logistic regression (LR) [28], have been introduced to identify peptides [29]. Most recently, Huang *et al.* [30] constructed a sequential model ensemble pipeline (SMEP) consisting of multiple steps, including empirical selection, classification, ranking, regression, and wet-lab validation. Algorithms, like boosting method (XGBoost) [31], RF as well as deep learning (DL) such as the convolutional neural network (CNN) [32] and the long short-term memory (LSTM) [33], were applied in different modules. With SMEP, a series of potent AMPs from the entire search space of peptide

Received: May 21, 2024. **Revised:** August 23, 2024. **Accepted:** November 1, 2024

libraries were identified accurately within a short period of time. In another study [34], multiple natural language processing neural network models, including LSTM layer, attention layer, and Encoder Representations from Transformers (BERT) [35], were combined to form a unified pipeline which has been used to mine functional peptides from metagenome data for in-depth investigations. The algorithms applied in the prediction models can be divided into two main categories. The first category is based on DL architectures, like AMPScanner V2 [36] and Deep-AmPEP30 [37]. AMPScanner V2 applied deep neural networks (DNN) [38] with convolutional, maximal pooling and LSTM layers for AMPs prediction. Deep-AmPEP30 is based on a CNN with two convolutional layers, two maximum pooling layers, and one fully connected hidden layer. As for the second category of models, conventional machine learning (ML) algorithms are generally exploited, such as iAMPpred [39], which uses SVM to classify positive or negative peptides. Previous studies [40] indicated that DL models did not always outperform conventional ML models due to the modeling complexities and/or modeling overfitting during the process of DL model construction based on training limited AMPs. Therefore, DL models are not necessarily the most suitable approach for AMPs identification [40]. Nonetheless, no matter the ML or DL-based methods, existing computational methods rely primarily on features derived from the composition, physicochemical, and structural features of the peptide sequence. These features may not be sufficient to fully express the rich information contained in AMPs and there is still considerable room for enhancing the accuracy of AMP prediction.

To address the aforementioned concerns, we propose herein an ensemble random projection (RP) [41] based computational model named SAMP, for which we develop a new type of feature called proportionalized split amino acid composition (PSAAC) [42] in addition to conventional sequence-based features to improve the prediction performance of AMP identification. Previous studies [43, 44] have evidenced that the composition of these regions can vary significantly across different types of proteins. Analyzing the composition of these regions independently provides more detailed information than analyzing the composition of the entire sequence. The primary advantage of this approach is that it allows for greater emphasis on proteins that have specific signals or features concentrated at either the N-terminal or C-terminal. Meanwhile, we demonstrate that SAMP outperforms existing state-of-the-art methods in terms of accuracy, Matthews correlation coefficient (MCC), the geometric mean of recall and precision (G-measure), and F1-score, including iAMPpred and AMPScanner V2, when benchmarking on both balanced and imbalanced datasets from different natural peptide groups, including humans, bacteria, amphibians, and plant. Furthermore, we integrate an ensemble RP architecture into SAMP to strengthen its ability to handle large-scale AMP screening while achieving enhanced performance compared to those without RP. In addition, we have developed a software package, SAMP, that is available for users to install and use on GitHub (https://github.com/wan-mlab/SAMP). Users can access the complete code for SAMP on GitHub along with a step-by-step guide document that explains how to use SAMP with an example dataset.

Materials and methods

Datasets

The positive data set for natural AMPs was accumulated in the antimicrobial peptide database in the past 20 years [45, 46] and the negative data set was extracted from UniProt [47]. To benchmark the performance of SAMP and other state-of-the-art approaches, we selected two sets of training data reported in the literature. As many existing approaches only provide web servers which have already been trained in different training data, to make a fair comparison, we will compare SAMP with those approaches based on the same training dataset based on which the corresponding web servers were trained. Specifically, the first set consists of 984 positive and 984 negative AMPs collected from the reference of iAMPpred [39]. This set is used to train our model and compare our proposed model SAMP with iAMPpred (Fig. 1A). The second set consists of 2021 positive and 2021 negative AMPs, exceeding 4000 sequences in total, collected from the reference of AMPScanner V2 [36], as shown in Fig. 1B. This set is used to train our model and compare SAMP with AMPScanner V2.

In addition, independent testing data were collected from the dbAMP database [48], containing AMP and non-AMP sequences (Fig. 1C). Specifically, we chose the AMP and non-AMP datasets across four different sources: plants, bacteria, amphibians, and humans, which were originally collected in the APD [45, 46] database. Given the varying peptide sequence length distributions of our AMP datasets (Fig. 1A–C), we filtered out sequences shorter than 10 amino acids and longer than 500 amino acids. The sequences containing non-standard amino acids were also removed. In the dbAMP benchmark dataset (Fig. 1D–E), there are 1089 AMPs and 9732 non-AMPs. Specifically, for the AMPs (Fig. 1D) of the dbAMP dataset, around half are amphibian, one-third belong to plant, and one-fifth are bacteria. On the contrary, in the non-AMP cases (Fig. 1D), amphibian sequences account for only 10%, and half of them are bacteria. Interestingly, human sequences constitute less than 10% in both AMPs and non-AMPs (Fig. 1D–E). While for the amino acid sequence length distribution (Fig. 1F–I), most AMPs for all sources are with shorter amino acid sequences compared to non-AMPs, suggesting significantly different sequence distributions between AMPs and non-AMPs. However, it is unlikely to use the length of peptide sequences to determine whether a peptide is an AMP or non-AMP, given that a significant portion of AMPs are also overlapped with non-AMPs, especially for bacteria, humans, and plants (Fig. 1F–I).

Feature extraction

Conventional features

We embedded the string of peptide sequences into categories of numeric feature vectors similar to those proposed by Meher *et al.* [39], which include amino acid sequence compositional features and physio-chemical (PHYC) features. The compositional features include amino acid composition (AAC), pseudo amino acid composition (PAAC), and normalized amino acid composition (NAAC). The PHYC features consider the hydrophobicity, net-charge, and iso-electric point of peptide sequences, and were calculated using the 'Peptide' package [49] in R.

Proportionalized split amino acid composition (PSAAC)

To maximally extract peptide sequence information, we propose a new compositional feature called proportionalized split amino acid composition (PSAAC). This concept refines the split amino acid composition (SAAC) approach, which differentiates between the AAC at the N and the C-terminal of protein sequences [43, 44]. PSAAC adapts this concept specifically for peptide sequences, dividing them into distinct segments according to proportions defined by the users. Given a peptide sequence $\mathcal{P}$ of length L, we split it into 3 segments using proportions (or percentage) p_1, p_2 and p_3, where p_1, p_2 and p_3 represent the proportion of amino acid segments for the N-terminal region, the middle region and the

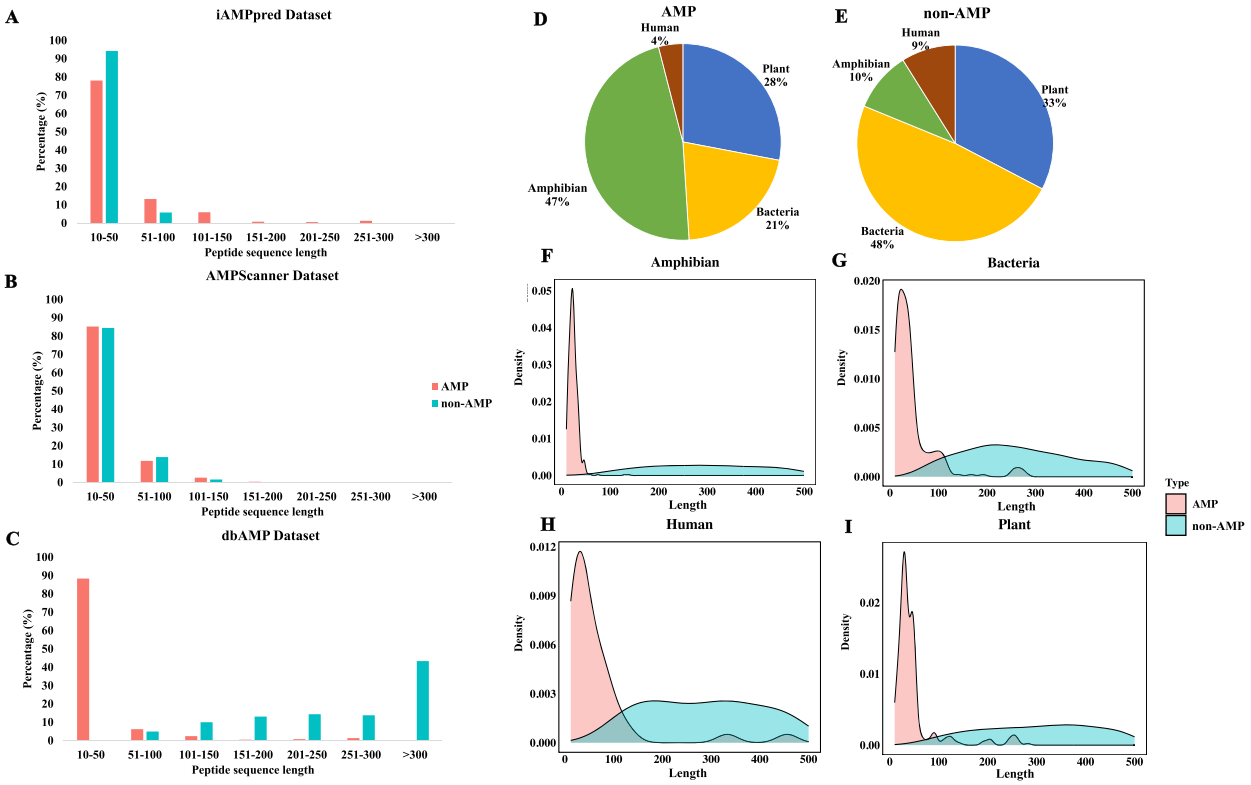

Figure 1. **Peptide sequence distribution of AMPs and non-AMPs in benchmarking datasets**. (**A**–**C**) The peptide sequence distribution of AMPs and non-AMPs collected from the iAMPpred dataset (**A**), the AMPScanner V2 dataset (**B**) and the dbAMP dataset (**C**), respectively. (**D**–**E**) Species breakdowns of AMPs (**D**) and non-AMPs (**E**) in the dbAMP dataset. (**F**-**I**) Source-specific peptide sequence distribution of AMPs and non-AMPs in the dbAMP dataset, including amphibian (**F**), bacteria (**G**), human (**H**) and plant (**I**).

C-terminal region, respectively, and $p_1 + p_2 + p_3 = 1$. The lengths of these segments, L_1, L_2, and L_3, are:

$$L_1 = \lceil L \times p_1 \rceil \quad (1)$$

$$L_2 = \lceil L \times p_2 \rceil \quad (2)$$

$$L_3 = L - L_1 - L_2 \quad (3)$$

The segments are:

$$l_1 = \mathcal{P}\lceil 1 : L_1 \rceil \quad (4)$$

$$l_2 = \mathcal{P}\lceil L_1 : (L_1 + L_2) \rceil \quad (5)$$

$$l_3 = \mathcal{P}\lceil (L_1 + L_2 + 1) : L \rceil \quad (6)$$

Now, let $\mathfrak{A}$ be the set of 20 standard amino acids. The AAC in segment l_i for $X \in \mathfrak{A}$is given by:

$$AAC_{i,X} = \frac{Count\ of\ X\ in\ l_i}{L} \quad (i = 1, 2, 3) \quad (7)$$

Note that the count of the X in each segment was divided by the whole length of the peptide sequence. Then, the PSAAC is:

$$PSAAC_{i,X} = [AAC_{1,A}, AAC_{1,C}, \cdots, AAC_{1,Y}, AAC_{2,A}, \cdots, AAC_{3,Y}] \quad (8)$$

Previous studies [50, 51] have reported that some sorting signals exist in the short segments of amino acid sequences around the N-terminal, representing special information on amino acid composition. In other words, different regions of a protein sequence can provide extra information. For example, some specific regions may form structural domains that determine the function of proteins, such as binding sites for other molecules, active sites for enzymes, or domains for protein–protein interaction [51]. The PSAAC feature captures the residue patterns around both the N-terminal region and the C-terminal region while also retaining the sequence order information from the middle region. Based on peptide sequences proposed by Daniel *et al.* [36], the amino acid compositions for each segment (e.g. the N-terminal region, the C-terminal region, and the middle region) were calculated, respectively. As shown in Fig. 2, Glycine and Leucine are the most abundant amino acids in AMPs and non-AMPs datasets, respectively (Fig. 2A and B). There are obvious differences in the composition of each amino acid at the N-terminal, the C-terminal, and middle region for both datasets (Fig. 2C and D). In the non-AMPs dataset, Leucine is the most abundant in all three segments. For the AMPs dataset, Glycine is the most abundant at both the N-terminal and the middle region, while Lysine is the most abundant at the C-terminal. Then, all the features are scaled by subtracting the mean from each column and dividing it by the standard deviation. For the data collected from AMPScanner V2 and dbAMP, the amino acid distribution at the N-terminal, the C-terminal and middle region is also investigated and shown in Supplementary Figs S1–S2. In addition, we also calculated the delta differences of

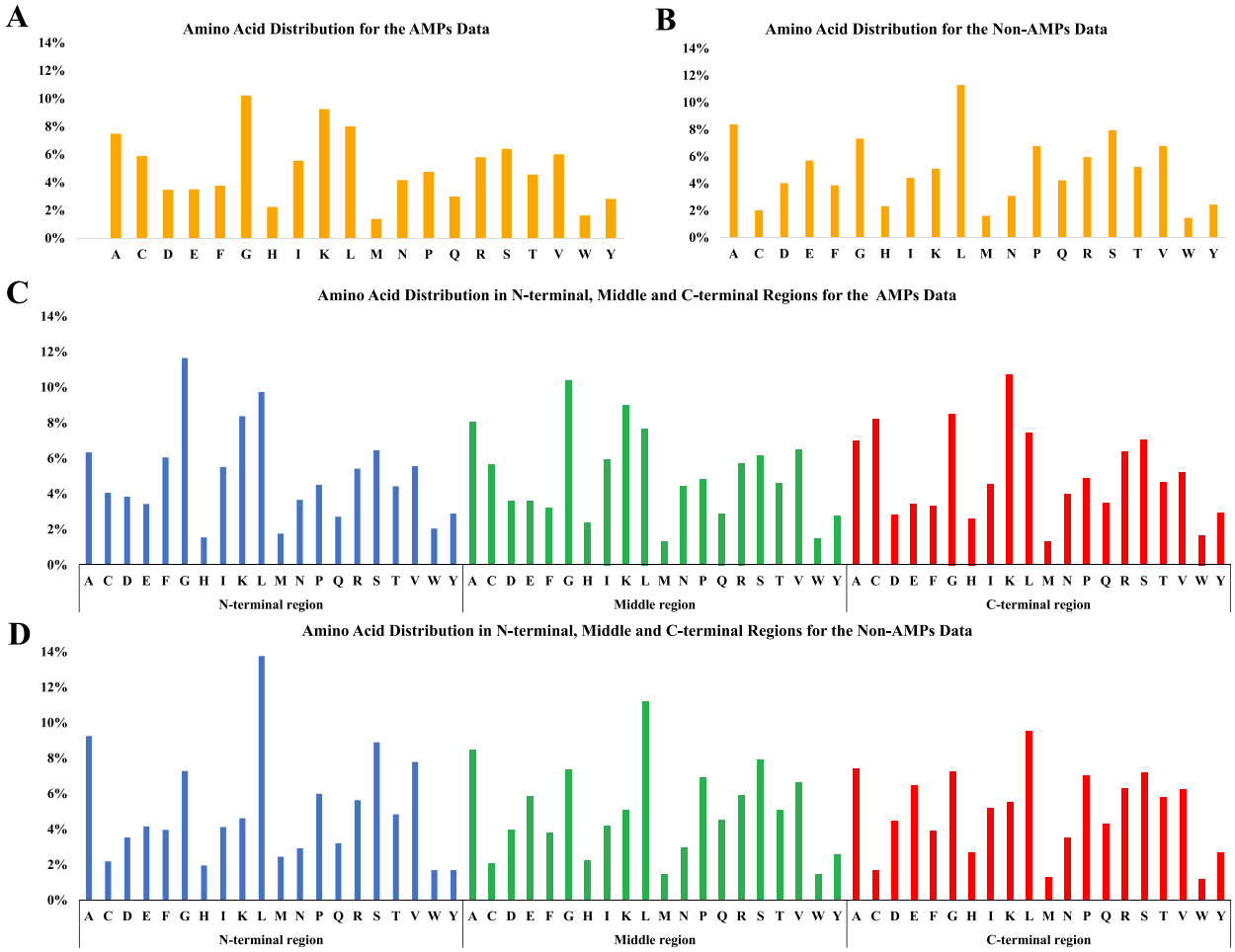

Figure 2. **Amino acid distribution in AMPs and non-AMPs datasets based on the dataset collected from iAMPpred**. Amino acid distribution of all sequences in (**A**) AMPs and (**B**) non-AMPs dataset. Distribution of amino acid sequences in the N-terminal region, the middle region and the C-terminal region of (**C**) AMPs dataset, and (**D**) non-AMPs dataset.

amino acid distributions between AMPs and non-AMPs for the total amino acid sequences and split regions (i.e. the C-terminal region, the middle region, and the N-terminal region) in three datasets (namely datasets from iAMPpred, AMPScanner V2, and dbAMP). As shown in Supplementary Fig. S3, Lysine exhibited the most significant difference in proportion of amino acids between AMPs and the non-AMPs. On the contrary, the largest differences of amino acid proportions between AMPs and non-AMPs for the N-terminal region, the middle region, and the C-terminal region were observed in Glycine, Lysine, and Cysteine, respectively. Similarly, for the dataset collected from AMPScanner V2, we demonstrated the delta differences of amino acid distributions between AMPs and non-AMPs in Supplementary Fig. S4. Specifically, the most remarkable differences among the 20 standard amino acids between AMPs and non-AMPs in the four cases (i.e. the overall amino acids, the C-terminal region, the middle region, and the N-terminal region) were Glutamate, Glycine, Glycine and Cysteine, respectively. In Supplementary Fig. S5, for the dataset from dbAMP, Cysteine was observed as the most significant difference amino acid between AMPs and non-AMPs for three cases (i.e. the overall amino acid sequences, the N-terminal region, and the C-terminal region), whereas Glycine was the most difference for the middle region. Of note, Glycine and Cysteine are also more abundant in natural AMPs than in globular proteins when all the natural sequences are considered [46]. For each peptide sequence, the PSAAC feature will be generated with 60 dimensions.

Random projection

Random projection (RP) is a dimension reduction technique proposed based on the Johnson-Lindenstrauss lemma [52]. For our experimental analysis, we used the Gaussian random matrix as our RP matrix, which is generated from the following distribution $N\left(0, \frac{1}{m_{components}}\right)$ where $m_{components}$ represents the number of dimensions to which the data is to be reduced. In our experiments, the optimal number of components to be kept was determined by the model training step using a grid-search approach. We also enabled the option of using a sparse matrix as the RP matrix in our package.

For dimension reduction, from original R dimension to the reduced r dimension, a very sparse random matric $\mathbf{Q} \in \mathbb{R}^{r\times R}$ is designed to reduce the computational complexity [53]. Specifically, elements of $\mathbf{Q}$ (i.e. $q_{i,j}$) are defined as:

$$q_{i,j} = \sqrt{t}\begin{cases} 1 \text{ with probability } \frac{1}{2t} \\ 0 \text{ with probability } 1-\frac{1}{t}, \text{ where } i=\{1,\cdots,r\},\ j=\{1,\cdots,R\} \\ -1 \text{ with probability } \frac{1}{2t} \end{cases} \tag{9}$$

As suggested by [53], we select $t = \sqrt{R}$.

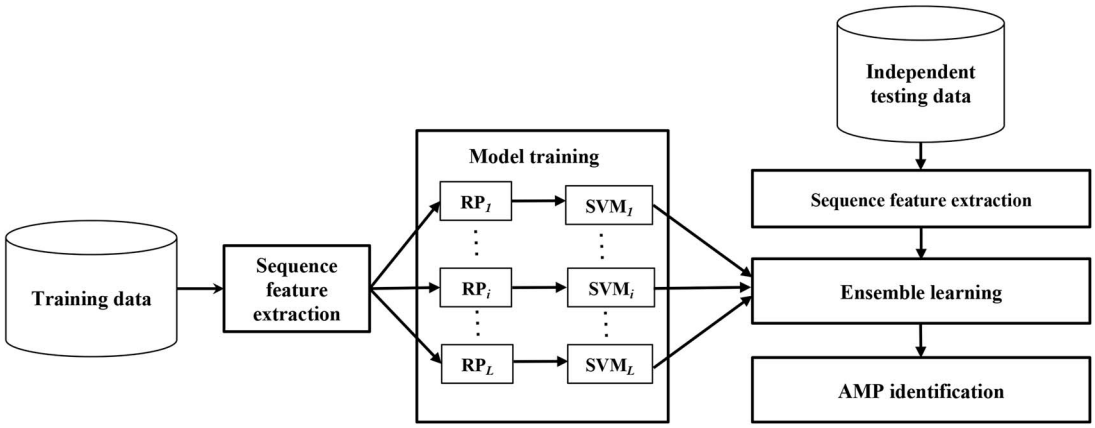

Figure 3. **Schematic representation of SAMP workflow**. Benchmarking data consisting of AMPs and non-AMPs were used for training. Features, including our proposed PSAAC, as well as conventional sequence features, were constructed. RP was applied multiple times to reduce the feature dimension for robustness. For each RP, the feature matrix was transformed in a low-dimensional space and was then fed into a classification model. The decision scores generated by the RBF-SVM model were integrated by an ensemble learning scheme, based on which predictions for independent test data were made to identify AMPs.

Ensemble learning

We use an ensemble learning model in SAMP (Fig. 3) where given the training and testing feature matrices $\mathbf{M}_{train}$ and $\mathbf{M}_{test}$ that have been scaled, and whose scaling process will be detailed in the feature scaling session, we first applied RP on the matrices respectively to get the new feature matrices $\mathbf{M}^*_{train}$ and $\mathbf{M}^*_{test}$ in a lower dimension. We then used the SVM as our base model to train and test on $\mathbf{M}^*_{train}$ and $\mathbf{M}^*_{test}$ respectively. We repeated the above steps 10 times to stabilize the result of RP, where randomness is often introduced when generating the RP matrix. Finally, the decision function scores in each iteration are recorded and averaged to get the final scores.

We then compared and selected the appropriate classifiers for AMP sequences classification, including RF, LR, SVM, multilayer perceptron (MLP), and XGBoost. Specifically, SVM allows for the use of different kernel functions to make predictions on both linear and non-linear data. Here, we use Radial Basis Function (RBF) kernel. RF works by building multiple decision trees and merging their outputs to make predictions. LR models the relationship between variables based on logistic/sigmoid function. MLP consists of at least three layers of notes, including an input layer, one/more hidden layers, and an output layer. XGBoost is designed for scalable gradient boosting, combining multiple decision trees for prediction. In each iteration of five classifier models, we trained them by performing a grid search with repeated 10-fold cross-validation to search for the best hyperparameters. Then, the model with the best hyperparameters was used to generate decision function scores for the independent testing datasets. The classifier demonstrating the highest accuracy will be selected to form the foundational architecture of SAMP.

Overview of SAMP

SAMP is an ensemble-based model that accurately classifies AMP by averaging the prediction scores from a set of base SVM models. Importantly, SAMP introduces the PSAAC feature, in addition to the widely used numeric features for AMP prediction task proposed in iAMPpred [39].

SAMP first encoded the peptide sequence into numeric features, such as AAC, PHYC, and PSAAC. The features were then scaled and projected to a pre-defined lower dimension using a RP technique. Base SVM models were built to generate the prediction scores for each run, which were eventually integrated by an ensemble learning scheme. SAMP was then evaluated on independent test data from four species (including amphibian, bacteria, human, and plant) and compared to other state-of-the-art methods, including iAMPpred and AMPScanner V2. To make fair comparisons, the same training data and independent test data were used to compare SAMP and other state-of-the-art methods.

Overall, the PSAAC enables SAMP to capture the peptide sequence information from both the middle region and the N/C-terminal regions, which significantly boosts the model performance in comparison to state-of-the-art methods.

Benchmarking with the state-of-the-art methods

We compared the performance of our model with two state-of-the-art methods, iAMPpred and AMPScanner V2. The benchmark test was performed by using the AMP and non-AMP data collected from the dbAMP database. The training data reported in the papers [36, 39] for iAMPpred and AMPScanner V2 were obtained to train SAMP separately. To demonstrate the importance of PSAAC and the robustness of our ensemble-based SVM model design, we conducted two types of further analyses. First, we trained models both with and without the PSAAC features, evaluating the results to ascertain the importance of PSAAC. Following this, we employed both the ensemble-based SVM model design and basic SVM model with one-time RP for training. For performance evaluation, we considered four major metrics: accuracy, MCC, G-measure, and F1-score. Here, MCC is a measure that produces a high score only if the prediction obtained good performance in all four aspects, true and false positives and negatives, of the confusion matrix, making it a reliable rate particularly for imbalanced datasets, as it is not biased toward the majority class [54]. The closer the value of MCC is to 1, the better the prediction effect of the classifier is. G-measure is the geometric mean of precision and recall, and it effectively balances the extreme ratio of positive to negative instances. The value ranges from 0 to 1, then a value closer to 1, indicating the classifier is performing well in both predicting the positive cases and maintaining accuracy. F1-score is the harmonic mean of precision and recall, and it differs from G-measure in that, F1-score is more sensitive to extreme values; if there is low precision or recall, the F1-score decreases significantly; however, g-measure will be more tolerant. Similarly, a closer value to 1 means the better prediction ability of the classifier.

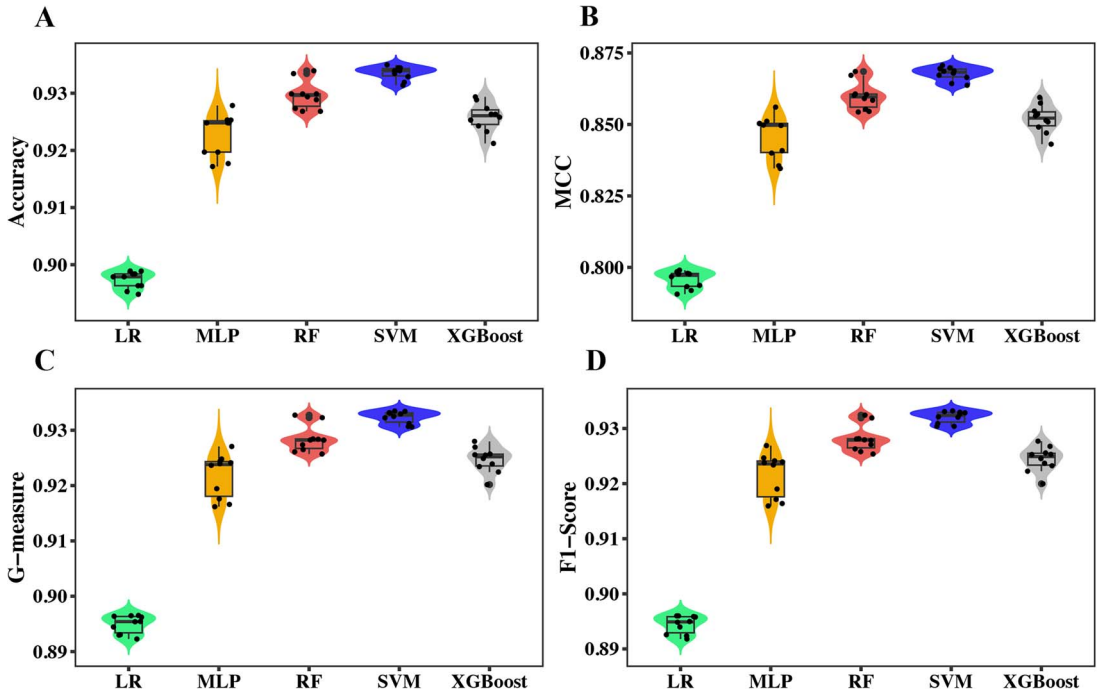

Figure 4. **Comparing different classifiers for SAMP**. All classifiers were trained on the same dataset collected from iAMPpred to perform 10 times of 10-fold cross-validation. Performance measures based on (**A**) Accuracy, (**B**) MCC, (**C**) G-measure, and (**D**) F1-score were reported. Classifiers include logistic regression (LR), multi-layer perceptron (MLP), random forest (RF), SVM (support vector machine), and XGBoost.

Results

Model performance and classifier selection

To enhance the prediction capability of SAMP, we initially selected SVM, RF, LR, MLP and XGBoost, using the same training and independent test dataset to train and test, then evaluated their performance. We performed 10-fold cross-validation for 10 times, each time we got an assessment value, as shown in Fig. 4, SVM had better performance than LR, MLP, RF and XGBoost, based on accuracy, MCC, G-measure and F1-score. Then, five trained classifiers were applied to predict labels for independent test data, as shown in Fig. 5, SVM exhibited the highest accuracy, MCC, G-measure and F1-score among all four test datasets. In summary, SVM presents a better performance than RF, MLP, XGBoost and LR, which was determined to serve as the basement of SAMP for further analysis. It was evident that the training performance in Fig. 4 was better than evaluation performance in Fig. 5. This discrepancy could be attributed to class imbalance. While the training set had a balanced class ratio (1:1), with 984 AMP sequences and 984 non-AMP sequences, the test set was highly skewed. For instance, the amphibian dataset contained 517 AMP and 932 non-AMP sequences (1:2), the bacteria dataset contained 226 AMP and 4721 non-AMP sequences (1:21), the human dataset contained 39 AMP and 894 non-AMP sequences (1:23), and the plant dataset contained 307 AMP and 3185 non-AMP sequences (1:10). Upon analyzing SAMP's prediction results, we found that the performance for AMP prediction was strong, with accuracy rates of 0.97, 0.91, 0.85, and 0.93 across species, respectively. However, for non-AMP prediction, the accuracy was significantly lower, at 0.67, 0.66, 0.64, and 0.66, respectively. This indicated that while SAMP performs well in predicting AMPs, it struggled with non-AMP predictions due to class imbalance.

We also measured the performance of SAMP across different dimensions of RP and all the possible proportions of PSAAC (Table 1). Grid-search with repeated 10-fold cross-validation was applied to assess the model performance on training datasets. The number of dimensions used in RP was 50, 100, and 150. Importantly, the novel feature of PSAAC enables a customized proportion of information to be obtained from a peptide sequence. To this end, we also evaluated the effect of different proportions of PSAAC on model performance. A given peptide sequence was first split into three parts according to the proportions specified. Next, the amino acid composition within each split was calculated, resulting in a total of 60 new features (see Method). The proportions evaluated include 2:2:6, 6:2:2, 2:6:2, and 3:4:3, where, for example, 2:2:6 represents cutting the peptide sequence from the N-terminal for 20% of the total sequence length, another 20% in the middle, and the remaining 60% for the C-terminal.

As shown in Table 1, it presented a comprehensive overview of the SAMP performance under varying ratios of PSAAC with different dimensions. It emphasized how different splitting schemes influenced the performance of SAMP, such as accuracy, MCC, Sn, Sp, and AUC. The accuracy presented minimum variation, ranging from 93.04 to 93.65, which indicated a consistently good performance across different configurations. The MCC, Sn, Sp, and AUC values varied slightly more but still could demonstrate the robust performance of SAMP, with MCC ranging from 86.06 to 87.36, Sn from 90.55 to 92.07, Sp from 94.82 to 95.83, and AUC ranging from 97.58 to 97.93. Among all the configurations, the 6:2:2 PSAAC ratio reached the highest Sp, while the 2:6:2 ratio got the best accuracy, and the 3:4:3 ratio outperformed others in terms of accuracy, MCC, Sn, and AUC. Analyzing performance based on dimensions, obviously, the dimension of 50 led in accuracy and MCC, the dimension of 100 exceeding in Sp, and the dimension of 150 topped in accuracy, Sn, and AUC. We used the prediction accuracy to determine the best ratio or proportion of the PSAAC. In this study, we hypothesize that the N and C-terminal contain key residue information regarding peptide stability, binding, and interaction. Different proportions were tried and were selected

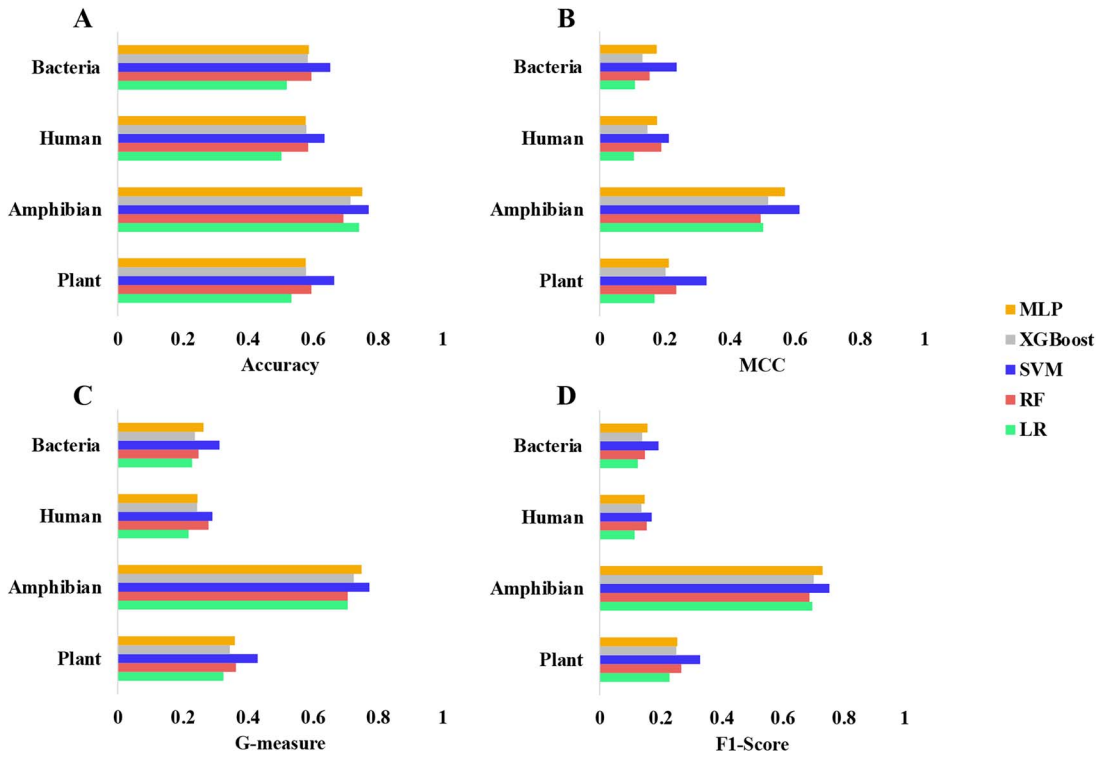

Figure 5. **Comparison of five ML models based on independent tests across multiple sources**. (**A**) Accuracy, (**B**) MCC, (**C**) G-measure, and (**D**) F1-score were compared across all sources, including bacteria, humans, amphibians, and plants. All models were trained on the dataset collected from [39] and tested on independent test datasets collected from [48].

Table 1. Comparing different splitting schemes and reduced dimensions for SAMP

PSAAC split	Dimensions	Evaluation metrics				
		Accuracy	MCC	Sn	Sp	AUC
2:2:6	50	93.04	86.04	91.06	94.92	97.58
	100	**93.29**	86.24	91.16	**95.02**	**97.79**
	150	93.09	**86.43**	**91.57**	94.82	97.77
6:2:2	50	93.24	86.28	90.65	95.53	97.63
	100	93.24	**86.70**	**90.75**	**95.83**	97.68
	150	**93.29**	86.59	**90.75**	95.73	**97.72**
2:6:2	50	93.09	86.29	90.55	**95.63**	97.63
	100	93.60	87.24	**91.97**	95.22	**97.87**
	150	**93.65**	**87.35**	**91.97**	95.33	97.82
3:4:3	50	**93.65**	**87.36**	91.77	**95.53**	97.83
	100	93.39	86.85	91.57	95.22	97.89
	150	**93.65**	87.34	**92.07**	95.22	**97.93**

The splitting scheme means different ratios of the sequence lengths of the N-terminal region, the middle region, and the C-terminal region. For example, 2:2:6 means splitting a peptide into three regions as the N-terminal region accounting for 20% of the total sequences, the middle region 20%, and the C-terminal region 60%. Here we tried four different splitting schemes including 2:2:6, 6:2:2, 2:6:2, and 3:4:3. For reduced dimensions of features, we tried three different cases, 50, 100, and 150. MCC, Matthews correlation coefficient; Sn, sensitivity; Sp, specificity; AUC, area under the receiver operating characteristic curve. Numbers in bold represent the best performance for each splitting scheme.

the best combination for AMP prediction. Based on the accuracy performance, 2:6:2 (with the dimension of 150) corresponded to the highest accuracy, 93.65%. We also noticed that 3:4:3 also gave the equivalently best accuracy, so it should also be regarded as the best proportion. In addition, we have evaluated whether the fixed length (e.g. the first 20 amino acids as the N-terminal region and the last 20 amino acids as the C-terminal region, with the remaining portion considered the middle region) based PSAAC construction method can get more stable result compared with the ratio-based (e.g. the first 20% of amino acids as the N-terminal region, the last 20% as the C-terminal region, and the middle 60% as the middle region) PSAAC construction method. Sequences longer than 100 amino acids from AMPScanner V2 and dbAMP were selected for model training and evaluation, respectively. However, sequences from amphibian species were excluded due to the lack of AMP sequences meeting the length requirement. As shown in Supplementary Fig. S6, SAMP has better prediction performance.

Benchmarking with the state-of-the-art methods

To further evaluate the predictive performance of SAMP, we first retrained SAMP with the same training data from the iAMPpred and AMPScanner V2, respectively. We tested their performance by using datasets collected from the dbAMP database. In particular,

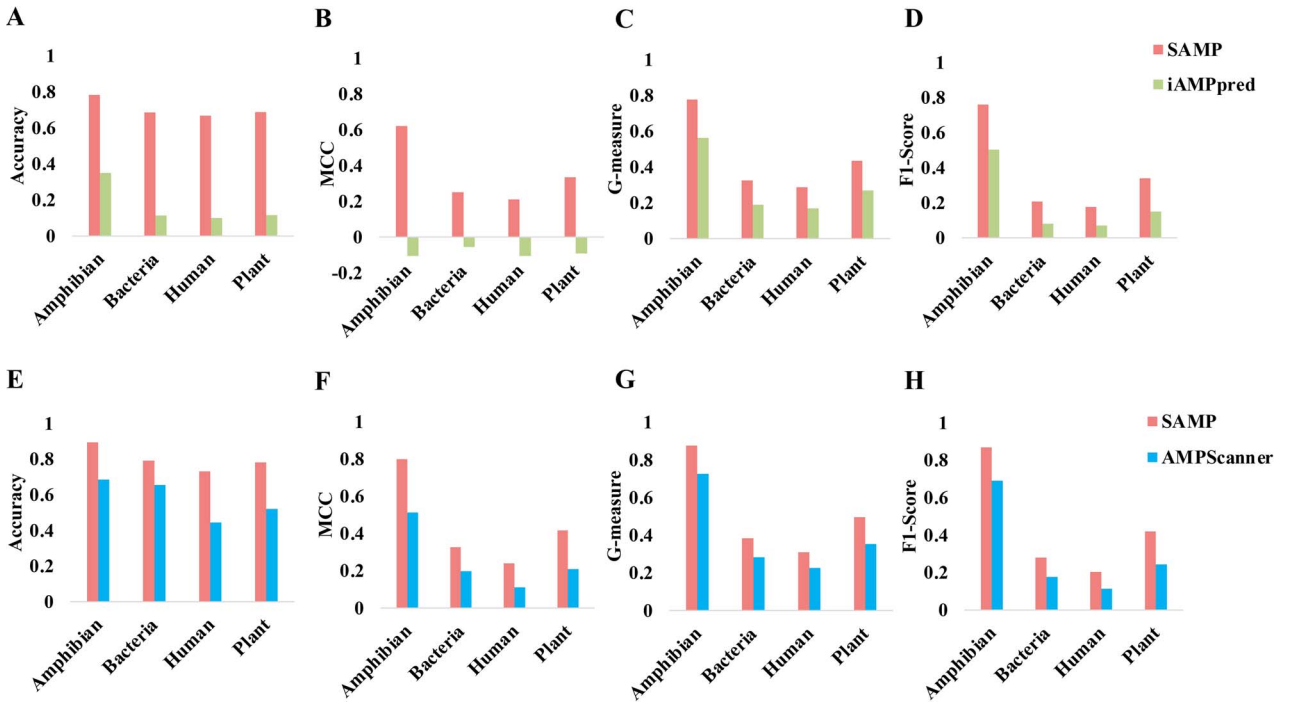

Figure 6. **Comparing SAMP with state-of-the-art methods on different source datasets**. Comparing SAMP and iAMPpred across different sources in terms of (**A**) Accuracy, (**B**) MCC, (**C**) G-measure and (**D**) F1-score. SAMP was trained on the same training dataset collected from iAMPpred and tested on independent test dataset collected from dbAMP. Comparing SAMP and AMPScanner V2 across different sources in terms of (**E**) Accuracy, (**F**) MCC, (**G**) G-measure and (**H**) F1-score. SAMP was trained on the same training dataset collected from AMPScanner V2 and tested on an independent test dataset collected from dbAMP.

we chose the AMPs and non-AMPs from plants, bacteria, amphibians, and humans. We considered accuracy, MCC, G-measure, and F-1 score as our major evaluation metrics.

First, SAMP was trained on 984 AMPs and 984 non-AMPs obtained from the iAMPpred paper. The trained SAMP was tested on the independent dataset from dbAMP. To assess the performance of iAMPpred, we uploaded the independent testing dataset to their web portal (http://cabgrid.res.in:8080/amppred/). Similarly, we trained SAMP using the exact same training dataset from AMPScanner V2 and uploaded the testing data to the web portal provided on https://www.dveltri.com/ascan/v2/ascan.html. For the performance differences among four sources, as shown in Fig. 6, regardless of whether the model was trained using data collected from iAMPpred or AMPScanner V2, the prediction performance was similar among four sources. Amphibian outperformed the other three sources in terms of accuracy, followed by plants, then bacteria, with humans showing the poorest performance. A similar trend was observed for MCC, G-measure, and F1-score. We conjecture that this might be due to the skewed class ratios within each dataset, where the amphibian dataset contains 517 AMP and 932 non-AMP sequences (1:2), the bacteria dataset contains 226 AMP and 4721 non-AMP sequences (1:21), the human dataset contains 39 AMP and 894 non-AMP sequences (1:23), and the plant dataset contains 307 AMP and 3185 non-AMP sequences (1:10). Additionally, SAMP demonstrates better performance compared to both iAMPpred and AMPScanner V2 across all four metrics. When specifically comparing SAMP with iAMPpred (Fig. 6A–D), the most obvious advantage of SAMP is observed in MCC for predicting amphibian labels, where SAMP is 73% more accurate than iAMPpred. On the other hand, the smallest difference is noticed in the F1-score for predicting human labels, with SAMP being 11% more effective than iAMPpred. Notably, all MCC values for iAMPpred are negative, indicating this tool may predict adverse results. Comparing SAMP with AMPScanner V2 (Fig. 6E–H) reveals similar trends. Probably due to a smaller dataset in the APD, the largest disparity is seen in accuracy for human AMP predictions, where SAMP shows a 29% improvement over AMPScanner V2, whereas the smallest difference is in the G-measure for human predictions, with a small improvement of 8% by SAMP over AMPScanner V2. Moreover, Fig. 6 highlights obvious performance differences between four sources and it likely should be attributed to the selection of independent testing data; additionally, factors such as data imbalance and the quality of the training data could also play significant roles.

Furthermore, we evaluated the impact of PSAAC and the ensemble-based SVM model architecture on the predictive performance (Fig. 7). After training with data from iAMPpred, SAMP consistently outperformed both the SAMP without the PSAAC feature and the vanilla SVM model without ensemble learning. This improvement was consistent in all performance metrics. Specifically, SAMP demonstrated at least an 11% increase in accuracy, 9% in MCC, 5% in G-measure, and 7% in F1-score compared to that without the PSAAC features, and SAMP significantly outperformed that without ensemble learning. Similar outcomes were observed when trained with AMPScanner V2 data, with SAMP outperforming the aforementioned situations across all measures.

Feature scaling

A crucial step in ML modeling is feature scaling. Intuitively, if features are measured in different scales, the decision boundary calculation of SVM would be dominated by the features with the largest scales. In our study, we always scaled the features after the feature generation stage using the *scale* function in R. In particular, the peptide sequence features are calculated in different scales. For example, the amino acid composition is measured as some

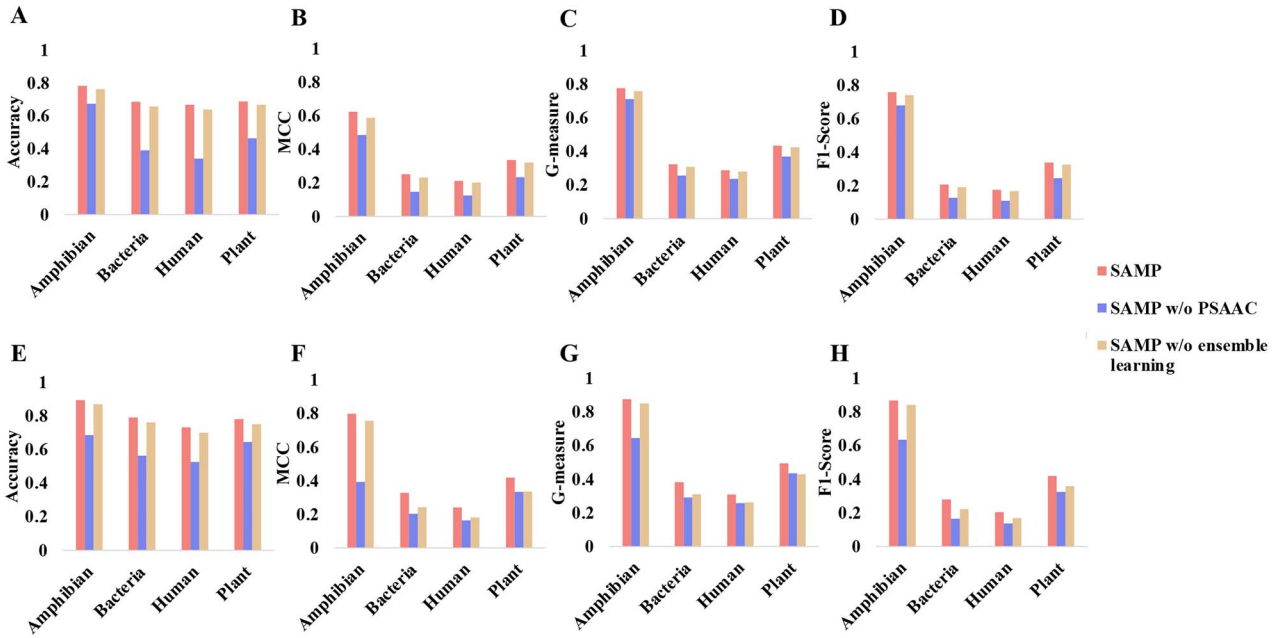

Figure 7. **PSAAC and ensemble learning contribute to improving prediction performance of SAMP for identifying AMPs**. Comparing SAMP and SAMP without the PSAAC feature across different sources in terms of (**A**) Accuracy, (**B**) MCC, (**C**) G-measure, and (**D**) F1-score. All models were trained on the same training dataset collected from iAMPpred and tested on an independent test dataset collected from dbAMP. Comparing SAMP and SAMP without ensemble learning across different sources in terms of (**E**) Accuracy, (**F**) MCC, (**G**) G-measure, and (**H**) F1-score. All models were trained on the same training dataset collected from AMPScanner V2 and tested on independent test dataset collected from dbAMP.

Table 2. Scaling the features is crucial for SAMP for identifying AMPs

Dataset	Metrics	SAMP (scaled)	SAMP (no scale)
plant AMPs	Accuracy	0.668	0.102
	AUC	0.744	0.112
	MCC	0.332	−0.184
bacteria AMPs	Accuracy	0.647	0.071
	AUC	0.703	0.088
	MCC	0.234	−0.165
amphibian AMPs	Accuracy	0.779	0.336
	AUC	0.844	0.039
	MCC	0.624	−0.169
human AMPs	Accuracy	0.637	0.058
	AUC	0.712	0.137
	MCC	0.204	−0.2

The scaling is performed by subtracting the mean of each feature and dividing by the feature's standard deviation. Scaling is a crucial step for SAMP. MCC, Matthews correlation coefficient; AUC, area under the receiver operating characteristic curve.

values between 0 and 1, but certain physio-chemical properties, such as hydrophobicity, can have various ranges of value. We generated two sets of features from the peptide sequences used to train iAMPpred, in which one set of features was scaled and the other was not. Two separate SAMP models were trained and evaluated on the independent test datasets. Our results indicate that scaling is indeed extremely important for SAMP, consistently boosting the model performance by at least 50% across datasets (Table 2).

Discussion

AMPs have gained greater attention as an alternative to chemical antibiotics or food preservatives [55]. Computational methods are developed as a supplement for wet lab experiments to design and identify AMPs, which reduces the cost and resources required. In this study, we present a novel ensemble-based model that achieves better AMP prediction performance than existing state-of-the-art methods. To the best of our knowledge, SAMP is the first method that adopts PSAAC as one of the numeric features for AMP prediction tasks. Amino acid compositional splitting sheds new light on amino acid compositions of natural AMPs, which was initially discovered in 2009 [56]. As for the biological rationale for selecting PSAAC ratios, specifically how to determine these ratios, lies in the inherent variability in the sequence lengths of the N-terminal, and C-terminal regions of peptides. These regions can differ significantly in length depending on the type of peptide or protein. For instance, the C-terminal region of the mouse Cplx1 molecule comprises residues 71–134 [57], almost half of the entire molecule. Similarly, the C-terminal of the assembly domain consists of residues 140–149 [58], acting as a morphogenetic switch. Additionally, the C-terminal domain (CTD) of human RNA polymerase II can have up to 52 repeats of the sequence Tyr-Ser-Pro-Thr-Ser-Pro-Ser [59]. Thus, the size of a C-terminal can vary widely among different proteins. Therefore, the length of C-terminal region, N-terminal region and middle region is dynamic and for a specific dataset, users could experiment with different ratios based on the prediction performance. However, based on prior experience, the proportion 2:2:6 has shown promising results, with relatively stable outcomes observed so far. In natural AMPs, Alanine, Glycine, Leucine, and Lysine are frequently occurring amino acids, while Histidine, Methionine, and Tryptophan are least abundant amino acids [46]. Our sequence splitting here reveals that Leucine is preferentially dominant at the N-terminal of AMPs, while Alanine is mainly located in the middle region. Glycine can appear frequently both at the N-terminal region and the middle region. In contrast, Lysine is primarily abundant in the middle and C-terminal of natural AMPs. Interestingly, after sequence splitting, the least abundant Methionine and Tryptophan appear mainly in the middle and the C-terminal regions,

whereas Histidine occupies the N-terminal. Also of note is that acidic glutamic acid is located at the N-terminal, and acidic aspartic acid prefers the C-terminal region. However, overall acidic amino acids are sparse in natural AMPs [46].

By combining this novel sequence-splitting feature with an ensemble-based SVM model architecture, SAMP is able to maximally extract peptide sequence information and outperform methods that apply either DL or traditional ML techniques. Additionally, we developed SAMP based on RP, a powerful dimension-reduction algorithm based on the Johnson–Lindenstrauss lemma [52], which can preserve the distances between data points while reducing the dimension [60]. As the number of data points continues to grow, the accuracy of prediction may be influenced due to the low efficiency of computational efficiency. Therefore, RP-based models should have better performance compared to those without it. This has been evidenced in a large-scale single-cell RNA-sequencing (scRNA-seq) data processed algorithm, which showed a higher classification efficiency under the contribution of ensemble RP layer [61]. In our proposed method SAMP, the ensemble learning is to assemble scores from multiple applications of RP. As shown in Fig. 3, we extracted features from the training dataset and subsequently applied the RP method for dimension reduction. Due to the inherent randomness in the RP method, we performed 10 times of RP. For each RP, the resulting dimension-reduced features were used to train an individual SVM model, resulting in the creation of 10 distinct SVM models. When evaluating the independent testing data, we extracted features and fed them into the 10 trained SVM models, generating 10 prediction scores for each input sequence. These prediction scores were then combined using an ensemble learning approach. Specifically, we averaged the 10 prediction scores, and based on these averaged scores, determining whether the input sequence was an AMP or not. As expected, our model with the ensemble RP layer also has a better performance, as shown in Fig. 7.

In addition, regarding whether a single model is an optimal approach to predict AMPs across different species (Fig. 6), it is generally more effective to train the model using data from a representative source and then apply the trained model to predict data from the same source. However, from the perspective of ML, the architecture of the ML model for data from different sources is the same. In this case, SAMP can be regarded as a single model for predicting AMPs across sources. Finally, as for whether there is a hidden hierarchy to the AMPs conditioned on source, we agree that information learned from training data across different sources might variably contribute to AMP identification with respect to different sources. However, whether those differences could constitute source-specific prediction results with statistical significance remains to be seen. As can be seen from our results (e.g. Fig. 6), there might be some differences in the accuracies among different sources, but these differences might not be statistically significant. In addition, a hidden hierarchy seems more like a hidden layer in a DL architecture. In our study, the method we proposed is based on ensemble SVM, which has no hidden hierarchy architecture.

Our prediction also implies that data size influences prediction performance since the human AMPs, with the least data (<150 AMPs in the current APD), behave poorest compared to AMPs from bacteria, plant, and amphibian with more known positive data. In addition, for the amino acid sequence length distribution among all four sources, length of the AMPs is always shorter than the non-AMPs which indicates there might be a significant difference of length distribution between AMPs and non-AMPs sequences. To evaluate the significance of the sequence length feature, we have calculated the length of each sequence collected from iAMPpred and dbAMP and included it as a feature, along with other features, to train and test the model respectively. As shown in Supplementary Fig. S7, the performance of the model incorporating the length feature, labeled as "SAMP+", was compared with the prediction performance of SAMP. SAMP+ exhibited worse results in terms of accuracy, MCC, G-measure, and F1-score across all four peptide sources. Therefore, incorporating the length feature into the model training resulted in lower prediction performance compared to the model without this feature. We should point out that this negative data set has not been validated experimentally.

We also assessed the performance of SAMP with specific tools, like iAMPpred and AMPScanner V2, which are also designed for AMP prediction based on SVM and DNN, respectively. SAMP proved to have slightly better performance than AMPScanner V2 and obviously higher accuracy than iAMPpred. Possible explanation for this discrepancy should be the omission of PSAAC and ensemble RP layer. In addition, we collected the data from one of the most recent methods, named E-CLEAP [62], which was just published in 2024. This method uses two independent features, AAC and PseACC feature, corresponding to two different models to predict the AMP. We have compared the E-CLEAP models and SAMP based on the same training dataset and the same independent test dataset. The training dataset was collected from E-CLEAP reference and the independent test datasets were collected from AMPScanner V2. As shown in Supplementary Fig. S8, SAMP outperformed both E-CLEAP models in terms of accuracy, MCC, G-measure and F1-score, demonstrating the superiority of SAMP over the latest state-of-the-art approaches for AMP identification. Overall, this newly designed tool, SAMP, is expected to compensate for the existing tools for AMP prediction.

For future research directions, we will consider different ensemble methods by including more diverse model categories to improve the prediction accuracy. Another potential research direction for SAMP is to predict the potential biological or clinical significance of AMPs. With the advance of DL, it would be appealing to investigate the performance of DL-based models combined with PSAAC features, or whether the deep neural networks are able to capture the PSAAC features within their embedding space.

Key Points

- We propose a novel method named SAMP that develops a new type of features called proportionalized split amino acid composition (PSAAC) to significantly boost the performance of identifying antimicrobial peptides.
- PSAAC can identify residue patterns at both the N-terminal and the C-terminal as well as to retain sequence order information from the middle region of peptide fragments.
- SAMP leverages an ensemble learning framework based on random projection to integrate various classifiers into a cohesive framework, effectively improving performance accuracy.
- SAMP outperforms state-of-the-art methods for AMP identification in terms of accuracy, G-measure, MCC, and F1-score.
- SAMP is a versatile tool capable of identifying AMPs from a variety of organisms including human, plant, bacteria and amphibian.

Supplementary data

Supplementary data are available at *Briefings in Functional Genomics* online.

Author contributions

SW conceived and designed the study. JF and MS developed the algorithm, performed the experiments and analyzed the data. JF implemented the SAMP package. All authors participated in writing the paper. The manuscript was approved by all authors.

Conflict of interest

The authors have declared that no competing interests exist.

Funding

Research reported in this publication was supported by the Office Of The Director, National Institutes Of Health of the National Institutes of Health under Award Number R03OD038391, and by the National Cancer Institute of the National Institutes of Health under Award Number P30CA036727. This work was supported by the American Cancer Society under award number IRG-22-146-07-IRG, and by the Buffett Cancer Center, which is supported by the National Cancer Institute under award number CA036727. This work was supported by the Buffet Cancer Center, which is supported by the National Cancer Institute under award number CA036727, in collaboration with the UNMC/Children's Hospital & Medical Center Child Health Research Institute Pediatric Cancer Research Group. This study was supported, in part, by the National Institute on Alcohol Abuse and Alcoholism (P50AA030407-5126, Pilot Core grant). This study was also supported by the Nebraska EPSCoR FIRST Award (OIA-2044049). This work was also partially supported by the National Institute of General Medical Sciences under Award Numbers P20GM103427 and P20GM130447. This study was in part financially supported by the Child Health Research Institute at UNMC/Children's Nebraska. This work was also partially supported by the University of Nebraska Collaboration Initiative Grant from the Nebraska Research Initiative (NRI). The content is solely the responsibility of the authors and does not necessarily represent the official views from the funding organizations.

Data availability

All the data used in this manuscript are publicly available in the corresponding references.

References

1. Fernandes P. Antibacterial discovery and development—the failure of success? *Nat Biotechnol* 2006;**24**:1497–503.
2. Adedeji WA. The TREASURE called antibiotics. *Ann Ib Postgrad Med* 2016;**14**:56–7.
3. Thomas L. *The Youngest Science: Notes of a Medicine-Watcher*. Penguin Publishing Group, Pennsylvania, USA, 1995.
4. Aminov RI. A brief history of the antibiotic era: lessons learned and challenges for the future. *Front Microbiol* 2010;**1**:134.
5. Hutchings MI, Truman AW, Wilkinson B. Antibiotics: past, present and future. *Curr Opin Microbiol* 2019;**51**:72–80. https://doi.org/10.1016/j.mib.2019.10.008.
6. Prestinaci F, Pezzotti P, Pantosti A. Antimicrobial resistance: a global multifaceted phenomenon. *Pathog Glob Health* 2015;**109**:309–18. https://doi.org/10.1179/2047773215Y.0000000030.
7. De Oliveira DM, Forde BM, Kidd TJ. *et al.* Antimicrobial resistance in ESKAPE pathogens. *Clin Microbiol Rev* 2020;**33**:e00181–19. https://doi.org/10.1128/cmr.00181-19.
8. Huemer M, Mairpady Shambat S, Brugger SD. *et al.* Antibiotic resistance and persistence—implications for human health and treatment perspectives. *EMBO Rep* 2020;**21**:e51034.
9. Frieri M, Kumar K, Boutin A. Antibiotic resistance. *J Infect Public Health* 2017;**10**:369–78. https://doi.org/10.1016/j.jiph.2016.08.007.
10. Lei J, Sun L, Huang S. *et al.* The antimicrobial peptides and their potential clinical applications. *Am J Transl Res* 2019;**11**:3919–31.
11. de Kraker MEA, Stewardson AJ, Harbarth S. Will 10 million people die a year due to antimicrobial resistance by 2050? *PLoS Med* 2016;**13**:e1002184.
12. Chen CH, Lu TK. Development and challenges of antimicrobial peptides for therapeutic applications. *Antibiotics* 2020;**9**:24.
13. Mookherjee N, Anderson MA, Haagsman HP, Davidson DJ. Antimicrobial host defence peptides: functions and clinical potential. *Nat Rev Drug Discov* 2020;**19**:311–32. https://doi.org/10.1038/s41573-019-0058-8.
14. Diamond G, Beckloff N, Weinberg A, Kisich KO. The roles of antimicrobial peptides in innate host defense. *Curr Pharm Des* 2009;**15**:2377–92.
15. Wang G, Li X, Wang Z. APD3: the antimicrobial peptide database as a tool for research and education. *Nucleic Acids Res* 2016;**44**:D1087–93. https://doi.org/10.1093/nar/gkv1278.
16. Hiemstra PS, Amatngalim GD, Van Der Does AM. *et al.* Antimicrobial peptides and innate lung defenses. *Chest* 2016;**149**:545–51. https://doi.org/10.1378/chest.15-1353.
17. Silva ON, De La Fuente-Núñez C, Haney EF. *et al.* An anti-infective synthetic peptide with dual antimicrobial and immunomodulatory activities. *Sci Rep* 2016;**6**:35465.
18. Frohm M, Agerberth B, Ahangari G. *et al.* The expression of the gene coding for the antibacterial peptide LL-37 is induced in human keratinocytes during inflammatory disorders. *J Biol Chem* 1997;**272**:15258–63.
19. Liang W, Diana J. The dual role of antimicrobial peptides in autoimmunity. *Front Immunol* 2020;**11**:545577.
20. De La Fuente-Núñez C, Silva ON, Lu TK. *et al.* Antimicrobial peptides: role in human disease and potential as immunotherapies. *Pharmacol Ther* 2017;**178**:132–40. https://doi.org/10.1016/j.pharmthera.2017.04.002.
21. Li C, Zhu C, Ren B. *et al.* Two optimized antimicrobial peptides with therapeutic potential for clinical antibiotic-resistant *Staphylococcus aureus*. *Eur J Med Chem* 2019;**183**:111686.
22. Fan L, Wei Y, Chen Y. *et al.* Epinecidin-1, a marine antifungal peptide, inhibits Botrytis cinerea and delays gray mold in postharvest peaches. *Food Chem* 2023;**403**:134419.
23. Adade CM, Oliveira IR, Pais JA, Souto-Padrón T. Melittin peptide kills *Trypanosoma cruzi* parasites by inducing different cell death pathways. *Toxicon* 2013;**69**:227–39. https://doi.org/10.1016/j.toxicon.2013.03.011.
24. Huan Y, Kong Q, Mou H. *et al.* Antimicrobial peptides: classification, design, application and research progress in multiple fields. *Front Microbiol* 2020;**11**:582779.
25. Wachinger M, Kleinschmidt A, Winder D. *et al.* Antimicrobial peptides melittin and cecropin inhibit replication of human immunodeficiency virus 1 by suppressing viral gene expression. *J Gen Virol* 1998;**79**:731–40.
26. Cortes C, Vapnik V. Support-vector networks. *Mach Learn* 1995;**20**:273–97.
27. Breiman L. Random forests. *Mach Learn* 2001;**45**:5–32.

28. Tolles J, Meurer WJ. Logistic regression: relating patient characteristics to outcomes. *JAMA* 2016;**316**:533–4. https://doi.org/10.1001/jama.2016.7653.
29. Wang G, Vaisman II, Van Hoek ML. Machine learning prediction of antimicrobial peptides. *Comput Pept Sci* 2022;**2405**: 1–37.
30. Huang J, Xu Y, Xue Y. *et al.* Identification of potent antimicrobial peptides via a machine-learning pipeline that mines the entire space of peptide sequences. *Nat Biomed Eng* 2023;**7**:797–810. https://doi.org/10.1038/s41551-022-00991-2.
31. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: Krishnapuram B, Shah M, Smola AJ, Aggarwal CC, Shen D, Rastogi R (eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, San Francisco, California, USA, 2016, 785–94.
32. LeCun Y, Bottou L, Bengio Y. *et al.* Gradient-based learning applied to document recognition. *Proc IEEE* 1998;**86**:2278–324.
33. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;**9**:1735–80.
34. Ma Y, Guo Z, Xia B. *et al.* Identification of antimicrobial peptides from the human gut microbiome using deep learning. *Nat Biotechnol* 2022;**40**:921–31. https://doi.org/10.1038/s41587-022-01226-0.
35. Devlin J, Chang M-W, Lee K. *et al.* Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 2018.
36. Veltri D, Kamath U, Shehu A. Deep learning improves antimicrobial peptide recognition. *Bioinformatics* 2018;**34**:2740–7. https://doi.org/10.1093/bioinformatics/bty179.
37. Yan J, Bhadra P, Li A. *et al.* Deep-AmPEP30: improve short antimicrobial peptides prediction with deep learning. *Mol Ther-Nucleic Acids* 2020;**20**:882–94. https://doi.org/10.1016/j.omtn.2020.05.006.
38. Lee KJ. Architecture of neural processing unit for deep neural networks. *Adv Comput* 2021;**122**:217–45.
39. Meher PK, Sahu TK, Saini V. *et al.* Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC. *Sci Rep* 2017;**7**:42362.
40. García-Jacas CR, Pinacho-Castellanos SA, García-González LA. *et al.* Do deep learning models make a difference in the identification of antimicrobial peptides? *Brief Bioinform* 2022; **23**:bbac094.
41. Bingham E, Mannila H. Random projection in dimensionality reduction: applications to image and text data. In: Provost F, Srikant R, Schkolnick M, Lee D (eds.), *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, San Francisco, California, 2001, 245–50.
42. Wan S, Mak M-W, Kung S-Y. Ensemble linear neighborhood propagation for predicting subchloroplast localization of multi-location proteins. *J Proteome Res* 2016;**15**:4755–62.
43. Verma R, Varshney GC, Raghava GPS. Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile. *Amino Acids* 2010;**39**:101–10. https://doi.org/10.1007/s00726-009-0381-1.
44. Hayat M, Khan A, Yeasin M. Prediction of membrane proteins using split amino acid and ensemble classification. *Amino Acids* 2012;**42**:2447–60. https://doi.org/10.1007/s00726-011-1053-5.
45. Wang Z. APD: the antimicrobial peptide database. *Nucleic Acids Res* 2004;**32**:590D-2.
46. Wang G. The antimicrobial peptide database is 20 years old: recent developments and future directions. *Protein Sci* 2023; **32**:e4778.
47. Lata S, Sharma B, Raghava G. Analysis and prediction of antibacterial peptides. *BMC Bioinformatics* 2007;**8**:263.
48. Jhong J-H, Chi Y-H, Li W-C. *et al.* dbAMP: an integrated resource for exploring antimicrobial peptides with functional activities and physicochemical properties on transcriptome and proteome data. *Nucleic Acids Res* 2019;**47**:D285–97. https://doi.org/10.1093/nar/gky1030.
49. Osorio D, Rondón-Villarreal P, Torres R. Peptides: a package for data mining of antimicrobial peptides. *Small* 2015;**12**:44–444.
50. Nakai K. Protein sorting signals and prediction of subcellular localization. *Adv Protein Chem* 2000;**54**:277–344.
51. Emanuelsson O. Predicting protein subcellular localisation from amino acid sequence information. *Brief Bioinform* 2002;**3**:361–76.
52. Johnson WB, Lindenstrauss J, Schechtman G. Extensions of Lipschitz maps into Banach spaces. *Isr J Math* 1986;**54**:129–38.
53. Li P, Hastie TJ, Church KW. Very sparse random projections. In: Ungar L, Craven M, Gunopulos D, Eliassi-Rad T (eds.), *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Philadelphia, PA, USA, , 2006, 287–96.
54. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 2020;**21**:1–13.
55. Mishra B, Reiling S, Zarena D, Wang G. Host defense antimicrobial peptides as antibiotics: design and application strategies. *Curr Opin Chem Biol* 2017;**38**:87–96. https://doi.org/10.1016/j.cbpa.2017.03.014.
56. Wang G, Li X, Wang Z. APD2: the updated antimicrobial peptide database and its application in peptide design. *Nucleic Acids Res* 2009;**37**:D933–7. https://doi.org/10.1093/nar/gkn823.
57. Reim K. Complexins. In: Stein J (ed.), *Reference Module in Neuroscience and Biobehavioral Psychology*, Elsevier, Amsterdam, Netherlands, 2017.
58. Zlotnick A, Cheng N, Stahl SJ. *et al.* Localization of the C terminus of the assembly domain of hepatitis B virus capsid protein: implications for morphogenesis and organization of encapsidated RNA. *Proc Natl Acad Sci U S A* 1997;**94**:9556–61.
59. Hsin J-P, Manley JL. The RNA polymerase II CTD coordinates transcription and RNA processing. *Genes Dev* 2012;**26**:2119–37. https://doi.org/10.1101/gad.200303.112.
60. Frankl P, Maehara H. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J Comb Theory Ser B* 1988;**44**:355–62.
61. Wan S, Kim J, Won KJ. SHARP: hyperfast and accurate processing of single-cell RNA-seq data via ensemble random projection. *Genome Res* 2020;**30**:205–13. https://doi.org/10.1101/gr.254557.119.
62. Wang S-C. E-CLEAP: an ensemble learning model for efficient and accurate identification of antimicrobial peptides. *PloS One* 2024;**19**:e0300125.