Analytical Biochemistry 436 (2013) 168–177

Contents lists available at SciVerse ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio

# iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types

Xuan Xiao [a,c,*], Pu Wang [a], Wei-Zhong Lin [a], Jian-Hua Jia [a], Kuo-Chen Chou [b,c,*]

[a] Computer Department, Jing-De-Zhen Ceramic Institute, Jing-De-Zhen 333403, China
[b] King Abdulaziz University, Jeddah, Saudi Arabia
[c] Gordon Life Science Institute, 53 South Cottage Road, Belmont, Massachusetts 02478, USA

ARTICLE INFO

*Article history:*
Received 30 October 2012
Received in revised form 10 January 2013
Accepted 21 January 2013
Available online 6 February 2013

*Keywords:*
Antimicrobial peptide
Pseudo amino acid composition
Physicochemical properties
Fuzzy *K*-nearest neighbor
Multi-label classification

ABSTRACT

Antimicrobial peptides (AMPs), also called host defense peptides, are an evolutionarily conserved component of the innate immune response and are found among all classes of life. According to their special functions, AMPs are generally classified into ten categories: Antibacterial Peptides, Anticancer/tumor Peptides, Antifungal Peptides, Anti-HIV Peptides, Antiviral Peptides, Antiparasital Peptides, Anti-protist Peptides, AMPs with Chemotactic Activity, Insecticidal Peptides, and Spermicidal Peptides. Given a query peptide, how can we identify whether it is an AMP or non-AMP? If it is, can we identify which functional type or types it belong to? Particularly, how can we deal with the multi-type problem since an AMP may belong to two or more functional types? To address these problems, which are obviously very important to both basic research and drug development, a multi-label classifier was developed based on the pseudo amino acid composition (PseAAC) and fuzzy *K*-nearest neighbor (FKNN) algorithm, where the components of PseAAC were featured by incorporating five physicochemical properties. The novel classifier is called **iAMP-2L**, where "2L" means that it is a 2-level predictor. The 1st-level is to answer the 1st question above, while the 2nd-level is to answer the 2nd and 3rd questions that are beyond the reach of any existing methods in this area. For the conveniences of users, a user-friendly web-server for **iAMP-2L** was established at http://www.jci-bioinfo.cn/iAMP-2L.



## 1. Introduction

Found among all classes of life, antimicrobial peptides, also called host defense peptides, are an evolutionarily conserved component of the innate immune response. These peptides are generally between 12 and 50 amino acids, including two or more positively charged residues provided by arginine, lysine or, in acidic environments, histidine, and a large proportion (generally >50%) of hydrophobic residues [1,2]. It has a special meaning for drug design as well as basic research to study antimicrobial peptides (AMPs) at a deeper level. The reasons are as follows. (**1**) AMPs are potent and broad spectrum antibiotics that have been demonstrated to kill Gram negative and Gram positive bacteria (including strains that are resistant to conventional antibiotics), mycobacteria (including mycobacterium tuberculosis), enveloped viruses, fungi and even transformed or cancerous cells. (**2**) With the broad range of activity and the short contact time required for inducing killing, AMPs have been considered as excellent candidates for developing novel therapeutic agents [3,4]. With the growing microbial resistance to conventional antimicrobial agents [5] as well as the avalanche of protein sequences generated in the postgenomic age, it is highly desirable to develop sequence-based computational tools for rapidly and accurately identifying AMPs and their types for helping design new and more effective antimicrobial agents, it is highly desirable to develop computational tools for rapidly and accurately identifying AMPs and their types for helping design new and more effective antimicrobial agents.

Actually, considerable efforts have been made in this regard. For instances, Wang et al. constructed the antimicrobial peptide database (APD) [6] and the updated antimicrobial peptide database (APD2) [7], accessible at http://aps.unmc.edu/AP/main.php and aimed to be a useful tool for naming (nomenclature), classification, information search, statistical analysis, prediction, and design of antimicrobial peptides. Their prediction interface allows users to input a query peptide sequence for predicting whether it has the potential to be antimicrobial. In 2007, by means of the hidden Markov models (HMMs), Fjell et al. [8] proposed the AMPer method for identifying AMPs. Meanwhile, Lata et al. successively developed the AntiBP predictor [9] and AntiBP2 predictor [10] for

* Corresponding authors. Addresses: Computer Department, Jing-De-Zhen Ceramic Institute, Jing-De-Zhen 333403, China (X. Xiao); Gordon Life Science Institute, 53 South Cottage Road, Belmont, Massachusetts 02478, USA (K.C. Chou).

*E-mail addresses:* xxiao@gordonlifescience.org, xiaoxuan0326@yahoo.com.cn (X. Xiao), wp3751@163.com (P. Wang), lin_weizhong@yahoo.com.cn (W.-Z. Lin), jjh163yx@163.com (J.-H. Jia), kcchou@gordonlifescience.org (K.-C. Chou).

0003-2697/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ab.2013.01.019

identifying antibacterial peptide, one of the subtypes of AMPs according to the amino acid sequence information. Thomas et al. [11] established a useful resource called CAMP (Collection of Anti-Microbial Peptides) for studying AMPs. Based on the experimentally validated data in CAMP, these authors further used various machine-learning algorithms such as Random Forests (RF), Support Vector Machines (SVM) and Discriminant Analysis (DA) to identify AMPs [11]. Subsequently, Wang et al. [12] proposed a new method for predicting AMPs by integrating the sequence alignment method with the feature selection method. Recently, Mohabatkar and coworkers proposed a new method for predicting AMPs peptides based on the concept of Chou's pseudo-amino acid composition and machine learning methods [13].

Although the aforementioned methods each have their own advantages and did play a role in stimulating the development of this area, they were only focused on identifying whether a query peptide was AMP, or limited at identifying one of its subtypes, without considering various possible different functional types of AMPs. In fact many AMPs have different functions or belong to two or more functional types. It can be seen by a comparison of the sequences in APD database [6] that a same sequence may occur in different subclasses; e.g., the antimicrobial peptide with the code "AMP AP00012" is not only an antibacterial peptide but also anticancer/tumor peptide and antifungal peptide. Actually, this kind of phenomenon is very common, as can be seen through a statistic analysis conducted on the APD entries. Accordingly, the AMP prediction should be a task of two-level multi-label classification. In view of this, the present study was initiated in an attempt to develop a two-level multi-label predictor for AMP, in which the 1st level is to identify whether a query peptide is AMP, and the 2nd-level is to identify which functional type(s) the peptide belongs to if it turns out to be an AMP in the 1st-level prediction.

To establish a really useful prediction method for a biological system based on the sequence information, we need to accomplish the following procedures [14]: (1) construct or select a valid benchmark dataset to train and test the predictor; (2) formulate the biological sequences with an effective mathematical expression that can truly reflect the intrinsic correlation with the target to be predicted; (3) introduce or develop a powerful algorithm to operate the prediction; (4) properly perform a cross-validation test to objectively evaluate the anticipated accuracy; (5) establish a user-friendly web-server for the predictor that can be easily used by most experimental scientists. Below, let us describe how to realize these procedures one by one.

## 2. Materials and Methods

### 2.1. Benchmark Dataset

For the convenience of later description, the benchmark dataset is expressed by

$$\mathbb{S} = \mathbb{S}^{\text{AMP}} \cup \mathbb{S}^{\text{non-AMP}} \tag{1}$$

where $\mathbb{S}^{\text{AMP}}$ is the AMP dataset consisting of AMP sequences only, $\mathbb{S}^{\text{non-AMP}}$ the non-AMP dataset with non-AMP sequences only, and $\cup$ is the symbol for union in the set theory. The peptide sequences in $\mathbb{S}^{\text{AMP}}$ were fetched from the APD database [6,7]. According to their different functional types, the AMP sequences can be further classified into ten categories; i.e.,

$$\mathbb{S}^{\text{AMP}} = \mathbb{S}_1^{\text{AMP}} \cup \mathbb{S}_2^{\text{AMP}} \cup \mathbb{S}_3^{\text{AMP}} \cup \mathbb{S}_4^{\text{AMP}} \cup \mathbb{S}_5^{\text{AMP}} \cup \cdots \cup \mathbb{S}_{10}^{\text{AMP}} \tag{2}$$

where the subscripts 1, 2, 3, 4, 5,...,10 represent "Antibacterial", "Anticancer/tumor", "Antifungal", "Anti-HIV", "Antiviral", "Antiparasital", "Anti-protist", "AMPs with chemotactic activity", "Insecticidal" and "Spermicidal" peptides, respectively. As shown in Fig. 1, the lengths of AMPs are varying within the region from 5 to 100 amino acids. Of the aforementioned 10 subsets, the subsets for "Antiparasital", "Anti-protist", "AMPs with chemotactic activity", "Insecticidal" and "Spermicidal" contained too few peptides (less than 50) to have statistical significance, and hence were not further considered in this study. Thus, Eq. (2) can be reduced to

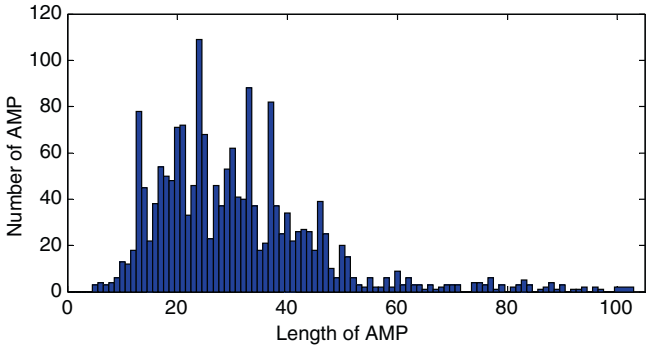


**Fig.1.** A histogram to show the distribution of the lengths of AMPs versus their numbers. The drawing was made based on the data from [6,7].

$$\mathbb{S}^{\text{AMP}} = \mathbb{S}_1^{\text{AMP}} \cup \mathbb{S}_2^{\text{AMP}} \cup \mathbb{S}_3^{\text{AMP}} \cup \mathbb{S}_4^{\text{AMP}} \cup \mathbb{S}_5^{\text{AMP}} \tag{3}$$

Furthermore, to reduce homology bias and redundancy, the program CD-HIT [15] was utilized to winnow those sequences that have $\geqslant$40% pairwise sequence identity to any other in a same subset. However, to ensure each of the subsets have sufficient samples for statistical treatment, the cutoff procedure was only imposed to those subsets that contained more than 150 samples. Finally, we obtained 878 AMPs, of which 454 belong to one functional attribute, 296 to two different functional attributes, 85 to three different functional attributes, 30 to four different functional attributes, and 13 to five different functional attributes. Because some AMPs may belong to two or more functional attributes [6,7], it is instructive to introduce the concept of "virtual AMP" as done in [16,17] when dealing with proteins with multiple location sites. The concept of virtual AMP can be briefed as follows. If an AMP possesses two different attributes of function, it will be counted as two virtual AMPs; if it possesses three attributes, it will be counted as three virtual AMPs; and so forth. Thus, the number of total virtual AMPs can be expressed as [17]

$$N(\text{vir}) = N(\text{seq}) + \sum_{m=1}^{M} (m-1)N(m) \tag{4}$$

where $N$(vir) is the number of total virtual AMPs, $N$( seq) the number of total different AMP sequences, $N(1)$ the number of AMPs with one functional type, $N(2)$ the number of AMPs with two functional types, and so forth; while $M$ is the number of total functional types investigated. Substituting the aforementioned data into Eq. (4), we obtained

$$\begin{aligned} N(\text{vir}) &= N(\text{seq}) + (1-1)\times 454 + (2-1)\times 296 + (3-1)\times 85 \\ &\quad + (4-1)\times 30 + (5-1)\times 13 \\ &= 878 + 0 + 296 + 170 + 90 + 52 = 1,486 \end{aligned} \tag{5}$$

meaning that the current benchmark dataset $\mathbb{S}^{\text{AMP}}$ contains 1,486 virtual AMPs, of which 770 belong to "Antibacterial", 140 to "Anticancer/tumor", 366 to "antifungal", 86 to ant-HIV, and 124 to "Antiviral" (see Table 1).

The peptide sequences in $\mathbb{S}^{\text{non-AMP}}$ were constructed according to the following procedures.

**Table 1**
Breakdown of the benchmark dataset $\mathbb{S}$.

| Attribute | Dataset | Functional attribute | Subset | Number of sequences |
|---|---|---|---|---|
| AMP | $\mathbb{S}^{AMP}$ | Antibacterial | $\mathbb{S}_1^{AMP}$ | 770 |
| | | Anticancer/tumor | $\mathbb{S}_2^{AMP}$ | 140 |
| | | Antifungal | $\mathbb{S}_3^{AMP}$ | 366 |
| | | Anti-HIV | $\mathbb{S}_4^{AMP}$ | 86 |
| | | Antiviral | $\mathbb{S}_5^{AMP}$ | 124 |
| | | Total virtual AMPs | | 1,486[a] |
| | | Total different AMPs | | 878[a] |
| Non-AMP | $\mathbb{S}^{non\text{-}AMP}$ | N/A | N/A | 2,405 |

[a] See Eqs. (4), (5).

*Step 1.* All the peptides, protein fragments and protein sequences with length 5 to 100 residues were collected from the UniProt (release 2012_08).
*Step 2.* Removed were those sequences with any of the following annotations: "Antimicrobial", "Antibiotic", "Fungicide", or "Defensin".
*Step 3.* Removed were those sequences that contained any code other than the 20 native amino acid codes.
*Step 4.* To reduce homology bias and redundancy, the CD-HIT program was utilized to remove those sequences that had ⩾ 40% pairwise sequence identity to any other.

Finally, we obtained 2,405 sequences, which were used to form the non-AMP dataset $\mathbb{S}^{non\text{-}AMP}$ (Table 1).

The sequences of the 1,486 virtual AMPs classified into five functional types and the sequences of the 2,405 non-AMPs are given Online Supporting Information S1 (in FASTA format).

### 2.2. Sequence Encoding Schemes

To develop a powerful method for identifying AMP peptides and their functional types according to the sequence information, one of the first important things is to formulate the peptide samples with an effective mathematical expression that can truly reflect the intrinsic correlation with the target to be identified [14]. However, it is by no means an easy job to realize this because this kind of correlation is usually deeply hidden or "buried" into piles of complicated sequences.

Obviously, the most straightforward formulation for a peptide sample **P** of $L$ amino acids is its entire amino acid sequence; i.e.,

$$\mathbf{P} = \mathrm{R}_1\mathrm{R}_2\mathrm{R}_3\mathrm{R}_4\cdots\mathrm{R}_L \tag{6}$$

where $\mathrm{R}_1$ represents the 1st residue, $\mathrm{R}_2$ the 2nd residue, ...,$\mathrm{R}_L$ the $L$-th residue, and they each belong to one of the 20 native amino acids. In order to identify its attribute(s), the sequence-similarity-search-based tools, such as BLAST [18,19], was utilized to search the peptide database for those peptides that have high sequence similarity to the query peptide **P**. Subsequently, the attribute(s) of the peptides thus found were used to deduce the attribute(s) for the query **P**. Unfortunately, this kind of straightforward sequential model, although quite intuitive and able to contain the entire information of a peptide sequence, failed to work when the query peptide **P** did not have significant sequence similarity to any attribute-known peptides.

Thus, various non-sequential or feature vector models were proposed in hopes to establish some sort of correlation or cluster manner by which to enhance the prediction power.

Among the discrete models for a protein or peptide sample, the simplest one is its amino acid (AA) composition or AAC [20]. According to the AAC-discrete model, the peptide **P** of Eq. (6) can be formulated by [21]

$$\mathbf{P} = [f_1 \quad f_2 \quad \cdots \quad f_{20}]^{\mathbf{T}} \tag{7}$$

where $f_i$ ($i$ = 1,2, ⋯,20) are the normalized occurrence frequencies of the 20 native amino acids in peptide **P**, and **T** the transposing operator. Many methods for predicting protein attributes were based on the AAC-discrete model (see, e.g., [22–25]). However, as we can see from Eq. (7), if using the ACC model to represent the peptide **P**, all its sequence-order effects would be lost, and hence the prediction quality might be considerably limited.

To avoid completely losing the sequence-order information, the pseudo amino acid composition (PseAAC) was proposed, as formulated by [26]

$$\mathbf{P} = [p_1 \quad p_2 \quad \cdots \quad p_{20} \quad p_{20+1} \quad \cdots \quad p_{20+\lambda}]^{\mathbf{T}} \tag{8}$$

where

$$p_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\theta_j}, & (1 \leqslant u \leqslant 20) \\ \dfrac{w\theta_{u-20}}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\theta_j}, & (20+1 \leqslant u \leqslant 20+\lambda; \lambda < L) \end{cases} \tag{9}$$

where

$$\begin{cases} \theta_1 = \frac{1}{L-1}\sum_{i=1}^{L-1}\Theta(\mathrm{R}_i, \mathrm{R}_{i+1}) \\ \theta_2 = \frac{1}{L-2}\sum_{i=1}^{L-2}\Theta(\mathrm{R}_i, \mathrm{R}_{i+2}) \\ \vdots \qquad\qquad \vdots \\ \theta_\lambda = \frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}\Theta(\mathrm{R}_i, \mathrm{R}_{i+\lambda}) \end{cases} \quad (\lambda < L) \tag{10}$$

where $\theta_1$ is the first-tier correlation factor, $\theta_2$ the 2nd-tier correlation factor, and so forth, while the correlation function is

$$\Theta(\mathrm{R}_i, \mathrm{R}_j) = H(\mathrm{R}_i)\cdot H(\mathrm{R}_j) \tag{11}$$

where $H(\mathrm{R}_i)$ is the physicochemical property score of the amino acid $\mathrm{R}_i$, while $H(\mathrm{R}_j)$ the corresponding value for the amino acid $\mathrm{R}_j$.

In this study, the following five physical-chemical properties were taken into account: (1) hydrophobicity [27]; (2) pK1 ($C^{\alpha}$-COOH) [28]; (3) pK2 (NH3) [28]; (4) PI (25°C) [29]; (5) molecular weight. It is instructive to point out that many preliminary tests had been performed for a series of other physicochemical properties, but better outcomes were observed for the current case by using the aforementioned five properties (see Online Supporting Information S2 for the details).

The numerical values of the five physical-chemical properties for each of the 20 native amino acids can be obtained from [27–29] and most biochemistry text books (see, e.g., [30]). Note that before submitting these physicochemical quantities into Eq. (11), they were each subject to a standard conversion according to the following equation:

$$y_i = \frac{x_i - \mathrm{mean}(x)}{\mathrm{std}(x)} \tag{12}$$

where $x_i$ ($i$ = 1,2, ⋯,20) is the original physicochemical score of the $i$th amino acid, mean($x$) the average of such score over the 20 native amino acids, and std($x$) the corresponding standard deviation. The converted values thus obtained will have a zero mean value over the 20 amino acids, and will remain unchanged if they go through the same conversion procedure again [31].

As we can see from the above equations, the first 20 elements in Eq. (8) actually reflect the conventional amino acid composition

AAC; while the additional $\lambda$ factors reflect some sequence-order information via a series of rank-different correlation factors (cf. Eq. (10)).

Since in the current study there are five physical-chemical properties to be taken into consideration, similar to the dimension-augmenting approach in [32], we should use a $(20+5\lambda)$ - D feature vector to represent a sample; i.e., instead of Eq. (8), we should have

$$\mathbf{P} = [p_1 \quad p_2 \quad \cdots \quad p_{20} \quad p_{20+1} \quad \cdots \quad p_{20+5\lambda}]^{\mathbf{T}} \tag{13}$$

In this study, we choose $w = 0.1$ (cf. Eq. (9)) and $\lambda = 4$ for getting the optimal results.

It is instructive to note that, since the concept of PseAAC was introduced in 2001 [26], it has penetrated into almost all the fields of protein attribute predictions, such as predicting metalloproteinase family [33], predicting GABA(A) receptor proteins [34], predicting enzyme subfamily classes [35], predicting allergenic proteins [36], predicting cyclin proteins [37], predicting protein structural class [38], identifying bacterial virulent proteins [39], predicting DNA-binding proteins [40], predicting protein subcellular location [41], identifying protein submitochondrial localization [42], predicting apoptosis protein subcellular localization [43], predicting outer membrane proteins [44], predicting protein quaternary structure attribute [45,46], classifying amino acids [47], predicting G-protein-coupled receptor classes [48], predicting risk type of human papillomaviruses [49], predicting cyclin proteins [37], predicting protein folding rates [50], predicting protein supersecondary structure [51], among many others. Actually, the concept of PseAAC was not only limited for protein and peptide sequences; recently it was also extended to represent the feature vectors of DNA and nucleotides [52,53], as well as other biological samples (see, e.g., [54,55]). Because it has been widely used, in 2012 a powerful soft-ware called PseAAC-Builder [56] was established for generating various special Chou's pseudo-amino acid compositions, in addition to the web-server PseAAC [57] built in 2008.

### 2.3. Prediction Engine

An improved fuzzy *K*-nearest neighbor (F*K*NN) algorithm was used in this study. The F*K*NN classification method [58] is a variation of the *K*NN classifier. The latter is quite popular in pattern recognition community owing to its good performance and simple-to-use feature. According to the *K*NN rule [59,60], named also as the "voting *K*NN rule", a query sample should be assigned to the subset represented by a majority of its *K* nearest neighbors, as illustrated in Fig. 5 of [14]. However, in the F*K*NN classifier, it was the membership values that would be used to determine which class the query sample should belong to, as formulated below.

Suppose $\mathbb{S}(N) = \{\mathbf{P}_1, \mathbf{P}_1, \cdots, \mathbf{P}_N\}$ is a set of vectors representing $N$ peptides in a training set classified into $M$ classes $\{C_1, C_2, \cdots, C_M\}$, where $C_i$ denotes the $i$-th class; $\mathbb{S}^*(\mathbf{P}) = \{\mathbf{P}_1^*, \mathbf{P}_2^*, \cdots, \mathbf{P}_K^*\} \subset \mathbb{S}(N)$ is the subset of the $K$ nearest neighbor peptides to the query peptide $\mathbf{P}$. Thus, the fuzzy membership value for the query peptide $\mathbf{P}$ in the $i$-th class of $\mathbb{S}(N)$ is given by [61]

$$\mu_i(\mathbf{P}) = \frac{\sum_{j=1}^{K} \mu_i\left(\mathbf{P}_j^*\right) \mathrm{d}\left(\mathbf{P}, \mathbf{P}_j^*\right)^{-2/(\varphi-1)}}{\sum_{j=1}^{K} d\left(\mathbf{P}, \mathbf{P}_j^*\right)^{-2/(\varphi-1)}} \tag{14}$$

where $K$ is the number of the nearest neighbors counted for the query peptide $\mathbf{P}$; $\mu_i\left(\mathbf{P}_j^*\right)$, the fuzzy membership value of the training sample $\mathbf{P}_j^*$ to the $i$-th class as will be further defined below; $\mathrm{d}\left(\mathbf{P}, \mathbf{P}_j^*\right)$, the Euclidean distance between $\mathbf{P}$ and its $j$th nearest peptide $\mathbf{P}_j^*$ in the training dataset $\mathbb{S}(N)$; $\varphi(>1)$, the fuzzy coefficient for determining how heavily the distance is weighted when calculating each nearest neighbor's contribution to the membership value. Note that the parameters $K$ and $\varphi$ will affect the computation result of Eq. (14), and they will be optimized by a grid-search as will be described later.

For the 1st-level prediction in identifying a query peptide $\mathbf{P}$ as an AMP or non-AMP, a task of single-label classification, the quantitative definition for the aforementioned $\mu_i\left(\mathbf{P}_j^*\right)$ in Eq. (14) is given by

$$\begin{cases} \mu_i\left(\mathbf{P}_j^*\right) = 1, & \text{if } \mathbf{P}_j^* \in C_i \\ \mu_i\left(\mathbf{P}_j^*\right) = 0, & \text{otherwise} \end{cases} \quad \text{(for single label classification)} \tag{15}$$

After calculating all the memberships for a query peptide via Eqs. (15) and (14), it is assigned to the class with which it has the highest membership value; i.e., the predicted class for $\mathbf{P}$ should be

$$C_u = \mathbf{argmax}_i\{\mu_i(\mathbf{P})\} \tag{16}$$

where $u$ is the argument of $i$ that maximizes $\mu_i(\mathbf{P})$.

For the 2nd-level prediction in identifying which functional type(s) the query AMP peptide belongs to, a task of multi-label classification, an ingenious scheme will be used to replace Eq. (15); i.e.,

$$\begin{cases} \mu_i\left(\mathbf{P}_j^*\right) = \dfrac{1}{n(\text{hit})}, & \text{if } \mathbf{P}_j^* \in C_i \\ \mu_i\left(\mathbf{P}_j^*\right) = 0, & \text{otherwise} \end{cases} \quad \text{(for multi-label classification)} \tag{17}$$

where $n$(hit) is the number of different classes that were hit by $\mathbf{P}_j^*$ during the prediction. For instance: if only $C_1$ was hit by $\mathbf{P}_j^*$, then we have $n(\text{hit}) = 1$ and $\mu_1\left(\mathbf{P}_j^*\right) = 1$ and $\mu_{i\neq 1}\left(\mathbf{P}_j^*\right) = 0$; if only $C_1$ and $C_3$ were hit by $\mathbf{P}_j^*$, then $n(\text{hit}) = 2, \mu_1\left(\mathbf{P}_j^*\right) = \mu_3\left(\mathbf{P}_j^*\right) = 0.5$, and $\mu_{i\neq 1,3}\left(\mathbf{P}_j^*\right) = 0$; and so forth.

After calculating all the memberships for a query peptide via Eqs. (17) and (14), its functional type(s) will be predicted by

$$C_u = \{C_i \,|\, \mu_i(\mathbf{P}) \geq \Psi\} \tag{18}$$

where $u$ is the argument(s) of $i$ satisfying the condition $\mu_i(\mathbf{P}) \geqslant \Psi$, where $\Psi$ is a threshold. For example, if both $\mu_1(\mathbf{P})$ and $\mu_3(\mathbf{P})$ are equal to or greater than $\Psi$ but $\mu_{i\neq 1,3}(\mathbf{P}) < \Psi$, then $u = \{1,3\}$ and $\mathbf{P}$ will be predicted as "antibacterial" peptide and "antifungal" peptide (cf. Table 1), and so forth. The value of $\Psi$ will also be determined by an optimal procedure via the grid-search.

The classifier thus established is called **iAMP-2L**, where "**i**" means identifying, and "2L" means the identification consisting of two layers. The 1st layer is to identify a query peptide as AMP or not; if it is an AMP, the 2nd layer will be automatically continued to further identifying the AMP among the five functional attributes (cf. Table 1). To provide an intuitive picture, a flowchart to show the process of how the classifier works is given in Fig. 2.

### 2.4. Web Server

For practical applications, a user-friendly web-server for **iAMP-2L** was established at http://www.jci-bioinfo.cn/iAMP-2L, by which users can easily obtain their desired results without the need to follow the complicated mathematical equations involved for developing the predictor.

### 2.5. Performance Metrics

The ways to calculate the success rates for the single-label and multi-label classification should not be the same [31].

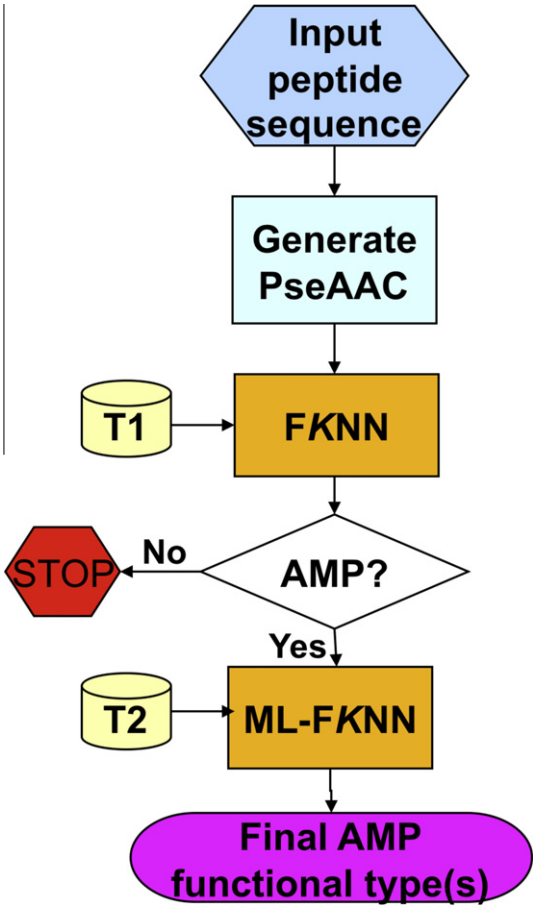


**Fig.2.** A flowchart to show the operation process of iAMP-2L. T1 represents the data taken from the dataset $\mathbb{S}$ (cf. Supporting Information S1) for training the 1st-level predictor; T2 represents those from the dataset $\mathbb{S}^{AMP}$ for training the 2nd-level predictor. ML-*FK*NN represents the multi-label fuzzy *K*-nearest neighbor classifier (cf. Eq. (17)). See the text for further explanation.

The 1st-level prediction in identifying a query peptide as an AMP or non-AMP belongs to the case of single-label classification. The following equation is often used in literature for examining the performance quality of a single-label predictor

$$\begin{cases} \mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \\ \mathrm{Acc} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \\ \mathrm{MCC} = \frac{(\mathrm{TP}\times\mathrm{TN})-(\mathrm{FP}\times\ \mathrm{FN})}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \end{cases} \tag{19}$$

where TP represents the true positive; TN, the true negative; FP, the false positive; FN, the false negative; Sn, the sensitivity; Sp, the specificity; Acc, the accuracy; MCC, the Mathew's correlation coefficient. For an intuitive and easy-to-understand explanation of Eq. (19), see a recent paper [52].

The 2nd-level prediction in identifying a query AMP among its five functional types (cf. Table 1) belongs to the case of multi-label classification, and its quality should be evaluated as follows. For a multi-label system consisting of *N* samples, suppose *M* is the number of all possible different categories, $\mathbb{L}$ the label set that contains the labels for all the possible categories concerned. Thus, the *i*-th peptide sample $\mathbf{P}_i$ and the category or categories it belongs to can be expressed by

$$\{\mathbf{P}_i, \mathbb{L}_i\}(i = 1, 2, \cdots, N) \tag{20}$$

where $\mathbb{L}_i$ is the subset that contains all the functional type label(s) for the *i*-th peptide sample. Obviously, we have

$$\mathbb{L}_1 \cup \mathbb{L}_2 \cup \cdots \cup \mathbb{L}_N \subseteq \mathbb{L} = \{\ell_1, \ell_2, \cdots \ell_M\} \tag{21}$$

where $\ell_i (i = 1,2, \cdots, M)$ is the label for the *i*-th functional type. For the current study, $N = 878$ and $M = 5$ (cf. Table 1). Suppose $\mathbb{L}_i^*$ represents the subset that contains all the predicted functional type label(s) for the *i*-th peptide. Thus, we have the following three metrics to measure the prediction quality for the multi-label system [17,62].

$$\begin{cases} \text{Hamming loss} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\| - \left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{M}\right) \\ \text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\|}\right) \\ \text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i^*\right\|}\right) \\ \text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i\right\|}\right) \\ \text{Absolute} - \text{True} = \frac{1}{N}\sum_{k=1}^{N}\Delta\left(\mathbb{L}_k, \mathbb{L}_k^*\right) \end{cases} \tag{22}$$

where $M = 5$ (cf. Table 1) is the total number of AMPs functional types covered by the current benchmark dataset $\mathbb{S}^{AMP}$, $\cup$ the symbol of union in the set theory, $\cap$ the intersection symbol, $\|\ \|$ the operator acting on the set therein to count the number of its elements, and

$$\Delta\left(\mathbb{L}_k, \mathbb{L}_k^*\right) = \begin{cases} 1, & \text{if all the labols in } \mathbb{L}_k \text{ are identical to those in } \mathbb{L}_k^* \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

Among the above five metrics, the rate for "Hamming loss" [62] reflects the rate of absolute false, which is opposite to those of the other four. As can be easily seen from Eq. (22), when the multi-labels for all the samples are correctly predicted, i.e., $\mathbb{L}_i \equiv \mathbb{L}_i^*$ or $\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\| = \left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\| \quad (i = 1, 2, \cdots, N)$, the rate of Hamming loss is equal to 0. When each of $\mathbf{P}_i \quad (i = 1,2, \cdots, N)$ is predicted completely wrong, i.e., belonging to all the possible categories except its own true category or categories; i.e., $\mathbb{L}_i \cup \mathbb{L}_i^* = \mathbb{L}$ and $\mathbb{L}_i \cap \mathbb{L}_i^* = \varnothing$, or $\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\| = M$ and $\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\| = 0$, the rate of Hamming loss is equal to 1. Therefore, the lower the Hamming loss is, the better the prediction quality will be. However, for the other four metrics, the meanings of their rates are just opposite; i.e., the higher their rates are, the better the prediction quality will be. As we can see from the above, it is much more complicated to evaluate the quality of a classifier on a multi-label system, just like in predicting protein subcellular localization for a system containing both single-location and multiple-location proteins, as elaborated in [31] and described by Eqs. (43)–(48) and Fig. 4 therein.

## 3. Results and Discussion

To validate a predictor, the following three cross-validation methods are often used in literatures: independent dataset test, subsampling test, and jackknife test [63]. However, as elaborated in [64] and demonstrated by Eqs. (28)–(30) in [14], considerable arbitrariness exists in the independent dataset test and subsampling test (or K-fold cross-over), and only the jackknife test is the least arbitrary that can always yield a unique result for a given benchmark dataset. Therefore, the jackknife test, also called Leave-One-Out (LOO) cross-validation [65], has been widely recognized and increasingly adopted by investigators to examine the quality of various predictors (see, e.g., [17,49,51,53,66–69]). In view of this, the jackknife test was also adopted in this study to examine the prediction quality of **iAMP-2L**.

For the 1st-level prediction, the values of parameter $K$ and $\varphi$ in Eq. (14) were determined by maximizing Acc of Eq. (19) using the jackknife test on the dataset $\mathbb{S}$ of Eq. (1) thru a 2-D grid search. For the 2nd-level prediction, the values of parameters $K$ and $\varphi$ in Eq. (14) and $\Psi$ in Eq. (18) were determined by minimizing the Hamming loss of Eq. (22) with the jackknife validation on the dataset $\mathbb{S}^{\text{AMP}}$ of Eq. (2) thru a 3-D grid search. The optimal parameter values thus determined for $K$ and $\varphi$ in Eq. (14) and $\Psi$ in Eq. (18) are summarized as follows

$$\begin{cases} K = 19, \varphi = 1.8 \quad \text{(for 1st-level operation)} \\ K = 21, \varphi = 1.5, \Psi = 0.31 \quad \text{(for 2nd-level operation)} \end{cases} \tag{24}$$

The single-label prediction quality achieved by the 1st-level of **iAMP-2L** for identifying AMPs and non-AMPs in the benchmark dataset $\mathbb{S}$ is measured by the four metrics as defined in Eq. (19). And their rates are given in Table 2, from which we can see the overall success rate achieved by **iAMP-2L** in identifying AMP or non-AMP is over 86%.

The multi-label prediction quality achieved by the 2nd-level of **iAMP-2L** is measured by the five metrics as defined in Eq. (22), and their outcomes for the benchmark dataset $\mathbb{S}^{\text{AMP}}$ (cf. Eq. (3)) are given below

$$\begin{cases} \text{Hamming loss} = 0.1640 \\ \text{Accuracy} = 0.6687 \\ \text{Precision} = 0.8331 \\ \text{Recall} = 0.7570 \\ \text{Absolute true} = 0.4305 \end{cases} \tag{25}$$

from which we can see the overall absolute-false or Hamming-loss (or absolute false) rate is very low (16.40%), while the absolute-true rate is much higher (43.05%), indicating the **iAMP-2L** is quite a promising multi-label predictor in identifying the functional types of AMPs as elucidated below.

It is instructive to point out that, for a multi-label system like the current one, the absolute-true success rate for each of the individual AMP functional types is meaningless and misleading [62,70]. Therefore, rather than the absolute-true success rate for each of the individual functional types, provided in Table 3 are the absolute true success rates for AMPs with different numbers of labels (or functional types). Furthermore, for facilitating comparison, listed in that table are also the corresponding rates by the completely random guess and weighted random guess, as defined below.

The completely random guess (CRG) rates were calculated according to the following equation

$$P(\text{CRG}) = \frac{1}{M} \cdot \frac{1}{C(M,m)} = \frac{1}{M \cdot \frac{M!}{(M-m)!m!}} \quad (m \leqslant M) \tag{26}$$

where $M$ is the total number of all the AMP functional types investigated that is equal to 5 for the current benchmark dataset $\mathbb{S}^{\text{AMP}}$, $m$ has the same meaning as in Eq. (3), and the symbol $C(M,m)$ represents the number of combinations of $M$ distinct things (or functional types) taken $m$ at a time.

The weighted random guess (WRG) rates were calculated according to the following equation

$$P(\text{WRG}) = \frac{N(m)}{N(\text{seq})} \cdot \frac{1}{C(M,m)} = \frac{N(m)}{\frac{N(\text{seq})M!}{(M-m)!m!}} \quad (m \leqslant M) \tag{27}$$

where $N(\text{seq})$ and $N(m)$ have the same meanings as in Eq. (3).

From Table 3 we can see the following: (**1**) absolute true rates for the AMPs with only one functional type is much higher than those with multiple functional types, indicating that it is much more difficult to predict the latter functional types exactly without any over-or under-prediction; (**2**) although for the small numbers of AMPs with 4 and 5 functional types the absolute-true rates by the **iAMP-2L** are about the same or slightly higher than those by the completely random guess, for most AMPs with 1 to 3 functional types the absolute-true rates achieved by **iAMP-2L** are about 12–15 times higher than those by the completely random guess; (**3**) particularly, for all the five cases, the absolute true rates by **iAMP-2L** are about 5–100 times higher than those by the weighted random guess.

To further demonstrate the power of the **iAMP-2L** predictor, let us compare it with some existing methods. As mentioned in the Introduction section, all the existing methods can only be used to identify a query peptide as an AMP or non-AMP, i.e., the 1st-level job by **iAMP-2L**; none of the existing methods can be used to deal with the 2nd-level job of **iAMP-2L**. Accordingly the comparison was limited in identifying AMPs or non-AMPs only. Also, the methods proposed in [12] and [13] did not provide any web-server, while the method in [9,10] were limited for antibacterial peptides only. To make it feasible and meaningful, the comparison was performed with the **CAM**P method [11], which contained three different algorithms or operation engines: the Support Vector Machine, Random Forests, and Discriminant Analysis.

Listed in Table 4 are the results obtained by **iAMP-2L** and **CAMP** [11] on an independent dataset $\mathbb{S}^{\text{Ind}}$, which contains 920 AMPs and 920 non-AMPs randomly picked from the removed sequences in the cutoff procedure (see Online Supporting Information S3). None of the peptide samples in the independent dataset $\mathbb{S}^{\text{Ind}}$occurred in the dataset used to train the two predictors. As we can see from the table, the rates for all metrics (Sn, Sp, Acc and MCC) achieved by

**Table 2**
Performance metrics (see Eq. (19)) achieved by **iAMP-2L** in identifying AMP and non-AMP.[a]

| Sn | Sp | Acc | MCC |
|---|---|---|---|
| 87.13% | 86.03% | 86.32% | 0.7265 |

[a] The rates reported here were based on the jackknife test on the benchmark dataset $\mathbb{S}$ of Eq. (1) and Online Supporting Information S1.

**Table 3**
A comparison of the absolute true success rates by different methods for the AMPs with different numbers of functional types.

| Number of functional types or labels | Number of AMPs | Absolute-true rate | | |
|---|---|---|---|---|
| | | **iAMP-2L** | Completely random guess[a] | Weighted random guess[b] |
| 1 | 454 | $\frac{276}{454} = 60.79\%$ | 4.00% | 10.34% |
| 2 | 296 | $\frac{71}{296} = 23.99\%$ | 2.00% | 0.81% |
| 3 | 85 | $\frac{27}{85} = 31.76\%$ | 2.00% | 0.31% |
| 4 | 30 | $\frac{1}{30} = 3.33\%$ | 4.00% | 0.68% |
| 5 | 13 | $\frac{3}{13} = 23.08\%$ | 20.00% | 2.96% |

[a] The completely random guess was calculated according to Eq. (26).
[b] The weighted random guess was calculated according to Eq. (27).

**Table 4**
Comparison of iAMP-2L with CAMP [11] on the independent dataset $\mathbb{S}^{Ind}$ containing 920 AMPs and 920 non-AMPs verified by experiments that were outside the dataset used to train the predictors. The detailed peptide sequences in $\mathbb{S}^{Ind}$ are given in Online Supporting Information S3.

| Method | Algorithm | Sn | Sp | Acc | MCC |
|---|---|---|---|---|---|
| CAMP | Support vector machine | 88.37% | 66.63% | 77.50% | 0.55 |
| | Random forest | 89.67% | 25.98% | 57.83% | 0.1565 |
| | Discriminant analysis | 86.63% | 64.13% | 75.38% | 0.5076 |
| iAMP-2L | Fuzzy *K*-nearest neighbor | 97.72% | 86.74% | 92.23% | 0.8446 |

**iAMP-2L** are remarkably higher than those by **CAMP** regardless which of its three operation engines was used for the prediction.

Why could the overall success rate be improved so remarkably by introducing the PseAAC? To address this problem, let us carry out a graphical analysis. Using graphic approaches to study biological systems can provide an intuitive vision and useful insights for helping analyze complicated relations therein, as indicated by many previous studies on a series of important biological topics, such as enzyme-catalyzed reactions [71–73], protein folding kinetics and folding rates [74], inhibition of HIV-1 reverse transcriptase [75], inhibition kinetics of processive nucleic acid polymerases and nucleases [76], protein sequence evolution [77], drug metabolism systems [78], and recently using wenxiang diagrams or graphs [79] to analyze protein-protein interactions [80].

To perform graphic analysis for the current case, let us consider the standard vectors [81,82] or norms [83] for AMPs and non-AMPs that were originally introduced for studying protein structural classification [81–83]. According to Eq. (13), when $\lambda = 4$ the standard vector $\mathbf{P}^{AMP}$ for AMPs and the standard vector $\mathbf{P}^{non\text{-}AMP}$ for non-AMPs can be respectively formulated as

$$\mathbf{P}^{AMP} = \left[\bar{p}_1^{AMP} \quad \bar{p}_2^{AMP} \quad \cdots \quad \bar{p}_u^{AMP} \quad \cdots \quad \bar{p}_{40}^{AMP}\right] \tag{28}$$

and

$$\mathbf{P}^{non\text{-}AMP} = \left[\bar{p}_1^{non\text{-}AMP} \quad \bar{p}_2^{non\text{-}AMP} \quad \cdots \quad \bar{p}_u^{non\text{-}AMP} \quad \cdots \quad \bar{p}_{40}^{non\text{-}AMP}\right] \tag{29}$$

The components in Eq. (28) are given by

$$\bar{p}_u^{AMP} = \frac{1}{40}\sum_{k=1}^{N^{AMP}} p_{u,k}^{AMP} \quad (u = 1, 2, \ldots, 40) \tag{30}$$

where $p_{u,k}^{AMP}$ is the $u$-th PseAAC component of the $k$-th peptide in the training dataset $\mathbb{S}^{AMP}$, while $N^{AMP}$ the total number of the AMP samples in $\mathbb{S}^{AMP}$.

The components in Eq. (29) are given by

$$\bar{p}_u^{non\text{-}AMP} = \frac{1}{40}\sum_{k=1}^{N^{non\text{-}AMP}} p_{u,k}^{non\text{-}AMP} \quad (u = 1, 2, \ldots, 40) \tag{31}$$

where $p_{u,k}^{non\text{-}AMP}$ is the $u$-th PseAAC component of the $k$-th peptide in the training dataset $\mathbb{S}^{non\text{-}AMP}$, while $N^{non\text{-}AMP}$ the total number of the non-AMP samples in $\mathbb{S}^{non\text{-}AMP}$. According to Eqs. (28) and (29), the standard vectors for AMPs and non-AMPs are two 40-D vectors. To provide an intuitive picture, let us project the 40 components in each of the two standard vectors onto a 2-D radar graph or diagram [84]. The radar diagrams thus obtained are shown in Fig. 3, where panel A is the standard vector for AMPs, and panel B for non-AMPs. As we can see from the figure, the radar diagram for the PseAAC of AMPs is remarkably different with that of non-AMPs. In other words, incorporating the aforementioned five physical-chemical properties into PseAAC can significantly enhance the distinction between AMPs and non-AMPs. This is the key why the success rate achieved by **iAMP-2L** in identifying AMPs and non-AMPS was so high. This is the essence why we choose to use the five physical-chemical properties.

By following the similar procedures, the radar diagrams for the five different functional types of AMPs can also be generated, as shown in Panels A, B, C, D, and E of Fig. 4. Compared with Fig. 3, the distinction among the five radar graphs in Fig. 4 is less remarkable. This is because the classification of the AMP functional types is actually a multi-label problem since an AMP may belong to two or more functional types. That is also why the overall absolute-true success rate achieved by **iAMP-2L** in identifying the functional types of AMPs (cf. Eq. (25)) is less than 86%, the overall success rate achieved by **iAMP-2L** in identifying AMP or non-AMP. Nevertheless, the absolute true success rates achieved by **iAMP-2L** for all the five cases listed in Table 3 are about 5–100 times higher than those by the weighted random guess.

## 4. Conclusion

The ability of AMPs to kill multidrug-resistant microorganisms has gained them considerable attention and clinical interest. With the growing microbial resistance to conventional antimicrobial agents, the demand for unconventional and efficient AMPs has become urgent. The results reported in this study indicate that the new predictor **iAMP-2L** holds very high potential to become a use-

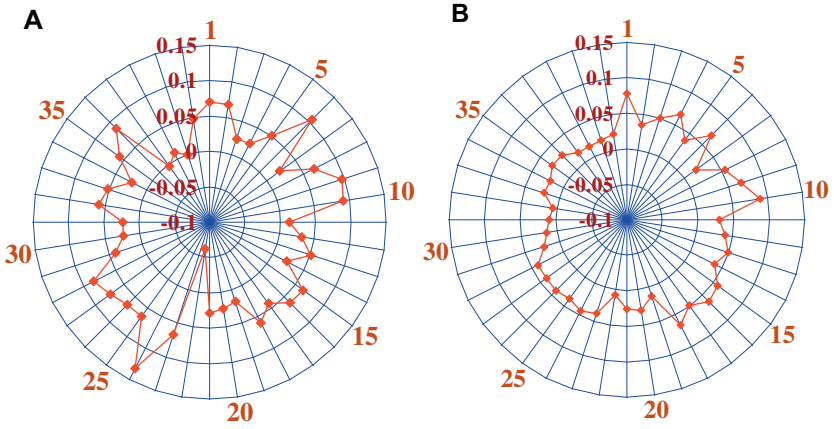


**Fig.3.** The radar diagram (or graph) to show the difference between (A) AMPs and (B) non-AMPs via their 40-D PseAAC standard vectors as defined by Eqs. (28) and (29), respectively.

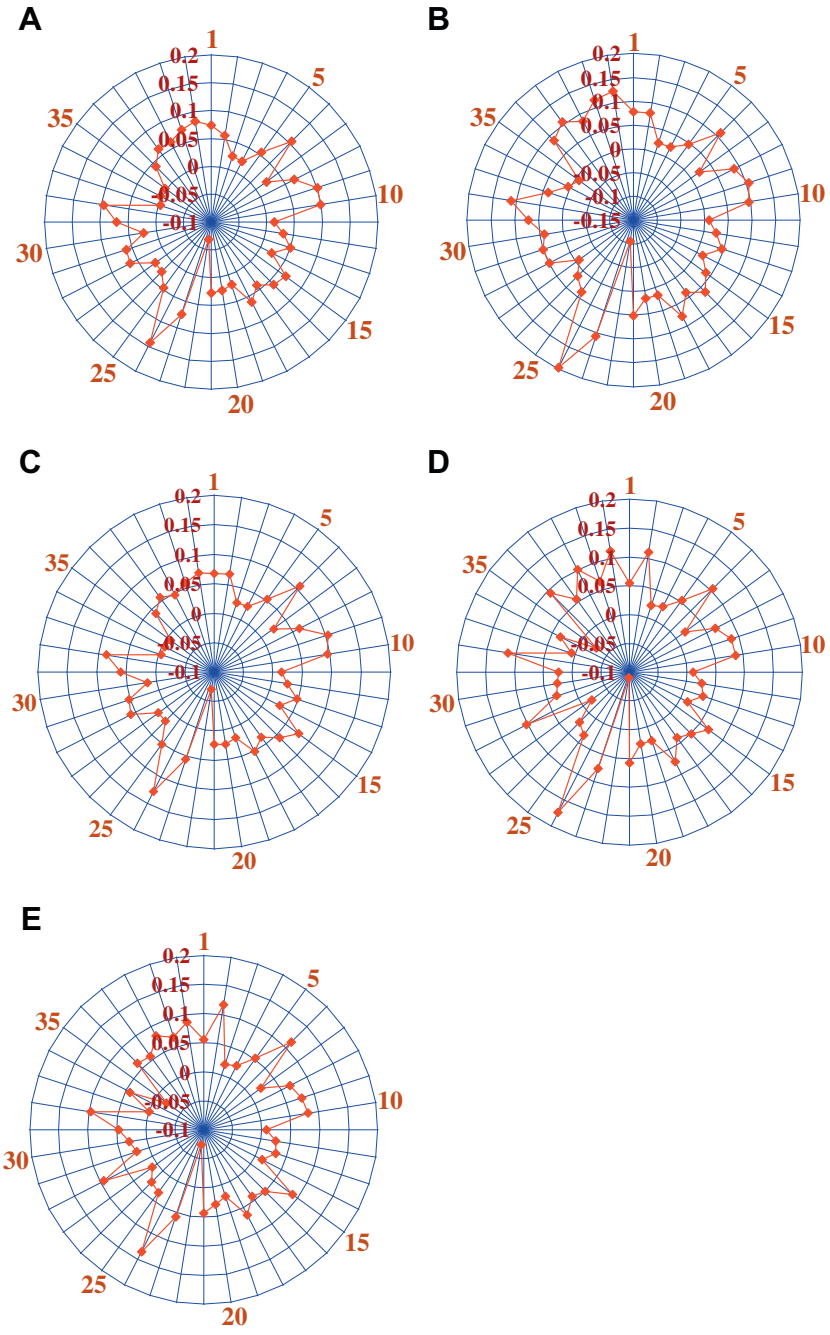


**Fig.4.** The radar diagram to show the difference among the following five different functional types of AMPs: (A) Antibacterial, (B) Anticancer/tumor, (C) Antifungal, (D) Anti-HIV, and (E) Antiviral. See the legend of Fig. 3 for more explanation.

ful high throughput tool for identifying AMPs and its functional types. Or at the very least, it may play an important complementary role to the existing predictors in this area. It has not escaped our notice that, with more data available for "Antiparasital", "Anti-protist", "AMPs with chemotactic activity", "Insecticidal" and "Spermicidal" in future, the current method can be straightforwardly extended to also cover these five AMP functional types. By that time, **iAMP-2L** will be able to identify AMPs and all their ten possible functional types as well. An announcement will be made either by a new publication or by the webpage of **iAMP-2L** when its coverage scope has been significantly enhanced.

## Acknowledgments

This work was supported by the grants from the National Natural Science Foundation of China (No.60961003, No.6121027 and No.31260273), the Key Project of Chinese Ministry of Education (No. 210116), the Province National Natural Science Foundation of JiangXi (No.2010GZS0122, No.20114BAB211013 and No. 20122BAB201020), the LuoDi plan of the Department of Education of JiangXi Province (KJLD12083), the Jiangxi Provincial Foreign Scientific and Technological Cooperation Project (No.20120 BDH80023), the Department of Education of JiangXi Province (GJJ12490), and the JiangXi Provincial Foundation for Leaders of Disciplines in Science (20113BCB22008).

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ab.2013.01.019.

## References


[1] N. Sitaram, R. Nagaraj, Host-defense antimicrobial peptides: importance of structure for activity, Current Pharmaceutical Design 8 (2002) 727–742.
[2] M. Papagianni, Ribosomally synthesized peptides with antimicrobial properties: biosynthesis, structure, function, and applications, Biotechnology Advances 21 (2003) 465–499.
[3] R.E.W. Hancock, A. Patrzykat, Clinical development of cationic antimicrobial peptides: from natural to novel antibiotics, Current Drug Targets – Infectious Disorsers 2 (2002) 79–83.
[4] A. Giuliani, G. Pirri, S. Nicoletto, Antimicrobial peptides: an overview of a promising class of therapeutics, Central European Journal of Biology 2 (2007) 1–33.
[5] H. Riadh, F. Ismail, Current trends in antimicrobial agent research: chemo-and bioinformatics approaches, Drug Discovery Today 15 (2010) 540–546.
[6] Z. Wang, G. Wang, APD: the antimicrobial peptide database, Nucleic Acids Research 32 (2004) D590–D592.
[7] G. Wang, X. Li, Z. Wang, APD2: the updated antimicrobial peptide database and its application in peptide design, Nucleic Acids Research 37 (2009) D933–D937.
[8] C.D. Fjell, R.E. Hancock, A. Cherkasov, AMPer: a database and an automated discovery tool for antimicrobial peptides, Bioinformatics 23 (2007) 1148–1155.
[9] S. Lata, B.K. Sharma, G.P.S. Raghava, Analysis and prediction of antibacterial peptides, BMC Bioinformatics 8 (2007) 263.
[10] S. Lata, N. Mishra, G. Raghava, AntiBP2: improved version of antibacterial peptide prediction, BMC Bioinformatics 11 (2010) S19.
[11] S. Thomas, S. Karnik, R.S. Barai, V.K. Jayaraman, S. Idicula-Thomas, CAMP: a useful resource for research on antimicrobial peptides, Nucleic Acids Research 38 (2010) D774–80.
[12] P. Wang, L. Hu, G. Liu, N. Jiang, X. Chen, J. Xu, W. Zheng, L. Li, M. Tan, Z. Chen, H. Song, Y.-D. Cai, K.-C. Chou, Prediction of antimicrobial peptides based on sequence alignment and feature selection methods, PLoS ONE 6 (2011) e18476.
[13] M. Khosravian, F.K. Faramarzi, M.M. Beigi, M. Behbahani, H. Mohabatkar, Predicting antibacterial peptides by the concept of Chou's pseudo-amino acid composition and machine learning methods, Protein and Peptide Letters (2012), doi: PPL-EPUB-20120807-7 [pii]..
[14] K.C. Chou, Some remarks on protein attribute prediction and pseudo amino acid composition (50th Anniversary Year Review), Journal of Theoretical Biology 273 (2011) 236–247.
[15] Y. Huang, B. Niu, Y. Gao, L. Fu, W. Li, CD-HIT Suite: a web server for clustering and comparing biological sequences, Bioinformatics 26 (2010) 680–682.
[16] K.C. Chou, Z.C. Wu, X. Xiao, iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins, PLoS One 6 (2011) e18258.
[17] K.C. Chou, Z.C. Wu, X. Xiao, iLoc-Hum: using accumulation-label scale to predict subcellular locations of human proteins with both single and multiple sites, Molecular Biosystems 8 (2012) 629–641.
[18] S.F. Altschul, Evaluating the statistical significance of multiple distinct local alignments, in: S. Suhai (Ed.), Theoretical and Computational Methods in Genome Research, Plenum, New York, 1997, pp. 1–14.
[19] J.C. Wootton, S. Federhen, Statistics of local complexity in amino acid sequences and sequence databases, Computational Chemistry 17 (1993) 149–163.
[20] H. Nakashima, K. Nishikawa, T. Ooi, The folding type of a protein is relevant to the amino acid composition, Journal of Biochemistry 99 (1986) 152–162.
[21] K.C. Chou, A novel approach to predicting protein structural classes in a (20-1)-D amino acid composition space, Proteins: Structure, Function and Genetics 21 (1995) 319–344.
[22] H. Nakashima, K. Nishikawa, Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies, Journal of Molecular Biology 238 (1994) 54–61.
[23] J. Cedano, P. Aloy, J.A. P'erez-Pons, E. Querol, Relation between amino acid composition and cellular location of proteins, Journal of Molecular Biology 266 (1997) 594–600.
[24] A. Reinhardt, T. Hubbard, Using neural networks for prediction of the subcellular location of proteins, Nucleic Acids Research 26 (1998) 2230–2236.
[25] G.P. Zhou, K. Doctor, Subcellular location prediction of apoptosis proteins, PROTEINS: Structure, Function, and Genetics 50 (2003) 44–48.
[26] K.C. Chou, Prediction of protein cellular attributes using pseudo amino acid composition, PROTEINS: Structure, Function, and Genetics (Erratum: ibid. 44 (2001) 60) 43 (2001) 246–255.
[27] C. Tanford, Contribution of hydrophobic interactions to the stability of the globular conformation of proteins, Journal of American Chemical Society 84 (1962) 4240–4274.
[28] C.W. Robert, CRC Handbook of Chemistry and Physics, 66th ed., CRC Press, Boca Raton, Florida, 1985.
[29] R.M.C. Dawson, D.C. Elliott, W.H. Elliott, K.M. Jones, Data for Biochemical Research, third ed., Clarendon Press, Oxford, 1986.
[30] D. Voet, J.G. Voet, C.W. Pratt, Fundamentals of Biochemistry, John Wiley & Sons, New York, 2002 (Chapter 13).
[31] K.C. Chou, H.B. Shen, Review: recent progresses in protein subcellular location prediction, Analytical Biochemistry 370 (2007) 1–16.
[32] K.C. Chou, Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes, Bioinformatics 21 (2005) 10–19.
[33] M. Mohammad Beigi, M. Behjati, H. Mohabatkar, Prediction of metalloproteinase family based on the concept of Chou's pseudo amino acid composition using a machine learning approach, Journal of Structural and Functional Genomics 12 (2011) 191–197.
[34] H. Mohabatkar, M. Mohammad Beigi, A. Esmaeili, Prediction of GABA(A) receptor proteins using the concept of Chou's pseudo-amino acid composition and support vector machine, Journal of Theoretical Biology 281 (2011) 18–23.
[35] X.B. Zhou, C. Chen, Z.C. Li, X.Y. Zou, Using Chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes, Journal of Theoretical Biology 248 (2007) 546–551.
[36] H. Mohabatkar, M.M. Beigi, K. Abdolahi, S. Mohsenzadeh, Prediction of allergenic proteins by means of the concept of Chou's pseudo amino acid composition and a machine learning approach, Medicinal Chemistry 9 (2013) 133–137.
[37] H. Mohabatkar, Prediction of cyclin proteins using Chou's pseudo amino acid composition, Protein and Peptide Letters 17 (2010) 1207–1214.
[38] S.S. Sahu, G. Panda, A novel feature representation method based on Chou's pseudo amino acid composition for protein structural class prediction, Computational Biology and Chemistry 34 (2010) 320–327.
[39] L. Nanni, A. Lumini, D. Gupta, A. Garg, Identifying bacterial virulent proteins by fusing a set of classifiers based on variants of Chou's pseudo amino acid composition and on evolutionary information, IEEE/ACM Transactions on Computational Biology and Bioinformatics 9 (2012) 467–475.
[40] Y. Fang, Y. Guo, Y. Feng, M. Li, Predicting DNA-binding proteins: approached from Chou's pseudo amino acid composition and other specific sequence features, Amino Acids 34 (2008) 103–109.
[41] S.W. Zhang, Y.L. Zhang, H.F. Yang, C.H. Zhao, Q. Pan, Using the concept of Chou's pseudo amino acid composition to predict protein subcellular localization: an approach by incorporating evolutionary information and von Neumann entropies, Amino Acids 34 (2008) 565–572.
[42] L. Nanni, A. Lumini, Genetic programming for creating Chou's pseudo amino acid based features for submitochondria localization, Amino Acids 34 (2008) 653–660.
[43] Y.S. Ding, T.L. Zhang, Using Chou's pseudo amino acid composition to predict subcellular localization of apoptosis proteins: an approach with immune genetic algorithm-based ensemble classifier, Pattern Recognition Letters 29 (2008) 1887–1892.
[44] H. Lin, The modified Mahalanobis discriminant for predicting outer membrane proteins by using Chou's pseudo amino acid composition, Journal of Theoretical Biology 252 (2008) 350–356.
[45] J.D. Qiu, S.B. Suo, X.Y. Sun, S.P. Shi, R.P. Liang, OligoPred: a web-server for predicting homo-oligomeric proteins by incorporating discrete wavelet transform into Chou's pseudo amino acid composition, Journal of Molecular Graphics and Modelling 30 (2011) 129–134.
[46] X.Y. Sun, S.P. Shi, J.D. Qiu, S.B. Suo, S.Y. Huang, R.P. Liang, Identifying protein quaternary structural attributes by incorporating physicochemical properties into the general form of Chou's PseAAC via discrete wavelet transform, Molecular BioSystems 8 (2012) 3178–3184.
[47] D.N. Georgiou, T.E. Karakasidis, J.J. Nieto, A. Torres, Use of fuzzy clustering technique and matrices to classify amino acids and its impact to Chou's pseudo amino acid composition, Journal of Theoretical Biology 257 (2009) 17–26.
[48] J.D. Qiu, J.H. Huang, R.P. Liang, X.Q. Lu, Prediction of G-protein-coupled receptor classes based on the concept of Chou's pseudo amino acid composition: an approach from discrete wavelet transform, Analytical Biochemistry 390 (2009) 68–73.
[49] M. Esmaeili, H. Mohabatkar, S. Mohsenzadeh, Using the concept of Chou's pseudo amino acid composition for risk type prediction of human papillomaviruses, Journal of Theoretical Biology 263 (2010) 203–209.
[50] J. Guo, N. Rao, G. Liu, Y. Yang, G. Wang, Predicting protein folding rates using the concept of Chou's pseudo amino acid composition, Journal of Computational Chemistry 32 (2011) 1612–1617.
[51] D. Zou, Z. He, J. He, Y. Xia, Supersecondary structure prediction using Chou's pseudo amino acid composition, Journal of Computational Chemistry 32 (2011) 271–278.
[52] W. Chen, P.M. Feng, H. Lin, K.C. Chou, iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition, Nucleic Acids Research (2013), http://dx.doi.org/10.1093/nar/gks1450.
[53] W. Chen, H. Lin, P.M. Feng, C. Ding, Y.C. Zuo, K.C. Chou, iNuc-PhysChem: a sequence-based predictor for identifying nucleosomes via physicochemical properties, PLoS ONE 7 (2012) e47843.
[54] B.Q. Li, T. Huang, L. Liu, Y.D. Cai, K.C. Chou, Identification of colorectal cancer related genes with mRMR and shortest path in protein–protein interaction network, PLoS ONE 7 (2012) e33393.
[55] T. Huang, J. Wang, Y.D. Cai, H. Yu, K.C. Chou, Hepatitis C virus network based classification of hepatocellular cirrhosis and carcinoma, PLoS ONE 7 (2012) e34460.

[56] P. Du, X. Wang, C. Xu, Y. Gao, PseAAC-Builder: a cross-platform stand-alone program for generating various special Chou's pseudo-amino acid compositions, Analytical Biochemistry 425 (2012) 117–119.
[57] H.B. Shen, K.C. Chou, PseAAC: a flexible web-server for generating various kinds of protein pseudo amino acid composition, Analytical Biochemistry 373 (2008) 386–388.
[58] J.M. Keller, M.R. Gray, J.A.J. Givens, A fuzzy K-nearest neighbor algorithm, IEEE tTransactions on Systems, Man, and Cybernetics 15 (1985) 580–585.
[59] T. Denoeux, k-nearest neighbor classification rule based on Dempster-Shafer theory, IEEE Transactions on Systems, Man and Cybernetics 25 (1995) 804–813.
[60] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy k-nearest neighbours algorithm, IEEE Transactions on Systems, Man and Cybernetics 15 (1985) 580–585.
[61] P. Wang, X. Xiao, K.C. Chou, NR-2L: a two-level predictor for identifying nuclear receptor subfamilies based on sequence-derived features, PLoS ONE 6 (2011) e23505.
[62] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, second ed., Springer, Heidelberg, 2010, pp. 1–19.
[63] K.C. Chou, C.T. Zhang, Review: prediction of protein structural classes, Critical Reviews in Biochemistry and Molecular Biology 30 (1995) 275–349.
[64] K.C. Chou, H.B. Shen, Cell-PLoc: a package of web servers for predicting subcellular localization of proteins in various organisms (updated version: Cell-PLoc 2.0: An improved package of web-servers for predicting subcellular localization of proteins in various organisms, Natural Science 2 (2010) 1090–1103; doi:10.4236/ns.2010.210136), Nature Protocols 3 (2008) 153–162.
[65] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, IJCAI (1995) 1137–1145.
[66] C. Chen, Z.B. Shen, X.Y. Zou, Dual-layer wavelet SVM for predicting protein structural class via the general form of Chou's pseudo amino acid composition, Protein and Peptide Letters 19 (2012) 422–429.
[67] M. Hayat, A. Khan, Discriminating outer membrane proteins with fuzzy K-nearest neighbor algorithms based on the general form of Chou's PseAAC, Protein and Peptide Letters 19 (2012) 411–421.
[68] S. Mei, Multi-kernel transfer learning based on Chou's PseAAC formulation for protein submitochondria localization, Journal of Theoretical Biology 293 (2012) 121–130.
[69] W.Z. Lin, J.A. Fang, X. Xiao, K.C. Chou, iLoc-Animal: A multi-label learning classifier for predicting subcellular localization of animal proteins, Molecular BioSystems 9 (2013) 634–644.
[70] G. Tsoumakas, L. Katakis, Multi-label classification: an overview, International Journal of Data Warehousing and Mining 3 (2007) 13.
[71] G.P. Zhou, M.H. Deng, An extension of Chou's graphic rules for deriving enzyme kinetic equations to systems involving parallel reaction pathways, Biochemical Journal 222 (1984) 169–176.
[72] K.C. Chou, Graphic rules in steady and non-steady enzyme kinetics, Journal of Biological Chemistry 264 (1989) 12074–12079.
[73] J. Andraos, Kinetic plasticity and the determination of product ratios for kinetic schemes leading to multiple products without rate laws: new methods based on directed graphs, Canadian Journal of Chemistry 86 (2008) 342–357.
[74] K.C. Chou, Review: applications of graph theory to enzyme kinetics and protein folding kinetics. Steady and non-steady state systems, Biophysical Chemistry 35 (1990) 1–24.
[75] I.W. Althaus, J.J. Chou, A.J. Gonzales, M.R. Diebel, K.C. Chou, F.J. Kezdy, D.L. Romero, P.A. Aristoff, W.G. Tarpley, F. Reusser, Steady-state kinetic studies with the non-nucleoside HIV-1 reverse transcriptase inhibitor U-87201E, Journal of Biological Chemistry 268 (1993) 6119–6124.
[76] K.C. Chou, F.J. Kezdy, F. Reusser, Review: steady-state inhibition kinetics of processive nucleic acid polymerases and nucleases, Analytical Biochemistry 221 (1994) 217–230.
[77] Z.C. Wu, X. Xiao, K.C. Chou, 2D-MH: a web-server for generating graphic representation of protein sequences based on the physicochemical properties of their constituent amino acids, Journal of Theoretical Biology 267 (2010) 29–34.
[78] K.C. Chou, Graphic rule for drug metabolism systems, Current Drug Metabolism 11 (2010) 369–378.
[79] K.C. Chou, W.Z. Lin, X. Xiao, Wenxiang: a web-server for drawing wenxiang diagrams (doi:10.4236/ns.2011.310111), Natural Science 3 (2011) 862–865. openly accessible at <http://www.scirp.org/journal/NS/>.
[80] G.P. Zhou, The disposition of the LZCC protein residues in wenxiang diagram provides new insights into the protein-protein interaction mechanism, Journal of Theoretical Biology 284 (2011) 142–148.
[81] K.C. Chou, W. Liu, G.M. Maggiora, C.T. Zhang, Prediction and classification of domain structural classes, PROTEINS: Structure, Function, and Genetics 31 (1998) 97–103.
[82] K.C. Chou, G.M. Maggiora, Domain structural class prediction, Protein Engineering 11 (1998) 523–538.
[83] K.C. Chou, Does the folding type of a protein depend on its amino acid composition?, FEBS Letters 363 (1995) 127–131
[84] K.C. Chou, A vectorized sequence-coupling model for predicting HIV protease cleavage sites in proteins, Journal of Biological Chemistry 268 (1993) 16938–16948.