Bioinformatics, 34(23), 2018, 4007–4016
doi: 10.1093/bioinformatics/bty451
Advance Access Publication Date: 1 June 2018
Original Paper

Sequence analysis

ACPred-FL: a sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides

Leyi Wei[1,5,*], Chen Zhou[1], Huangrong Chen[1], Jiangning Song[2,3,*] and Ran Su[4,5,*]

[1]School of Computer Science and Technology, Tianjin University, Tianjin, China, [2]Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, [3]Monash Centre for Data Science, Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia, [4]School of Computer Software, Tianjin University, Tianjin, China and [5]State Key Laboratory of Medicinal Chemical Biology, Nankai University, Tianjin, China

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on April 7, 2018; revised on May 14, 2018; editorial decision on May 25, 2018; accepted on May 29, 2018

Abstract

Motivation: Anti-cancer peptides (ACPs) have recently emerged as promising therapeutic agents for cancer treatment. Due to the avalanche of protein sequence data in the post-genomic era, there is an urgent need to develop automated computational methods to enable fast and accurate identification of novel ACPs within the vast number of candidate proteins and peptides.

Results: To address this, we propose a novel predictor named Anti-Cancer peptide Predictor with Feature representation Learning (ACPred-FL) for accurate prediction of ACPs based on sequence information. More specifically, we develop an effective feature representation learning model, with which we can extract and learn a set of informative features from a pool of support vector machine-based models trained using sequence-based feature descriptors. By doing so, the class label information of data samples is fully utilized. To improve the feature representation, we further employ a two-step feature selection technique, resulting in a most informative five-dimensional feature vector for the final peptide representation. Experimental results show that such five features provide the most discriminative power for identifying ACPs than currently available feature descriptors, highlighting the effectiveness of the proposed feature representation learning approach. The developed ACPred-FL method significantly outperforms state-of-the-art methods.

Availability and implementation: The web-server of ACPred-FL is available at http://server.malab.cn/ACPred-FL.

Contact: weileyi@tju.edu.cn or jiangning.song@monash.edu or ran.su@tju.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Cancer is one of the most devastating diseases accounting for millions of deaths worldwide each year (Ferlay *et al.*, 2010; Jemal *et al.*, 2010). Traditional chemotherapy is currently the major strategy for cancer treatment. Anti-cancer chemotherapeutic drugs are capable of killing cancer cells effectively. However, despite the advances in cancer therapy, such drugs have certain drawbacks, such as adverse effects on normal cells and resistance of cancer cells (Holohan *et al.*, 2013). Therefore, discovery and rational design of more effective therapeutic drugs is urgently needed. In the last few decades, peptides have emerged as alternative anti-cancer therapeutic agents. Compared with conventional chemotherapy, peptide-based therapy has a number of attractive advantages, such as high specificity, low production cost, high tumor penetration and ease of synthesis and modification (Otvos, 2008). Anti-cancer peptides (ACPs), are typically short peptides with a length of 10–50 amino acids and have been extensively explored as one of the most reliable anti-cancer therapeutics over the years (Barras and Widmann, 2011; Boohaker *et al.*, 2012). ACPs show a broad spectrum of cytotoxicity against various cancer cells but not to normal cells; the cancer-selective toxicity of ACPs is believed to be closely associated with the electrostatic interaction of ACPs with negatively charged components of the plasma membrane of cancer cells (Mader and Hoskin, 2006). Currently, many peptide-based therapies are being evaluated in terms of their efficacy to treat various tumor types across different phases of preclinical and clinical trials. In this context, discovery of novel ACPs and characterization of their functional mechanisms, have important implications in the success of these peptides in clinical settings.

Over the last decade, identification of ACPs has become a hot research topic in bioinformatics and computational biology. An increasing number of ACPs have been identified and experimentally validated (Tyagi *et al.*, 2015). Notably, most ACPs are found to be derived from protein sequences (Tyagi *et al.*, 2015). With the increasing availability of proteins as a result of high-throughput sequencing projects, it is expected that the number of potential ACPs will rapidly grow. As for the discovery of ACPs from protein sequence data, experimental methods are lab-intensive, time-consuming, expensive and difficult to be applied in a high-throughput manner. Accordingly, recent efforts have focused mainly on the development of computational methods, especially machine learning-based methods in order to expedite the identification of ACPs. However, a major challenge for such methods is how to effectively describe ACPs with informative feature representations, since ACPs are usually 10–50 residues long, too short to capture the specificity information of ACPs. To solve this problem, many efforts have been attempted in the literature. For example, Tyagi and his colleagues (Tyagi *et al.*, 2013) proposed to encode peptides with sequence-based feature descriptors, such as amino acid composition (AAC) and di-peptide composition. Using these feature descriptors, they developed a predictor called Anti-CP based on Support Vector Machine (SVM), the first computational tool established for identifying ACPs. Vijayakumar *et al.* (Vijayakumar and Ptv, 2015) reported that there was no significant difference in AACs observed between ACPs and non-ACPs. They thus proposed a new feature encoding method, incorporating not only compositional information but also centroidal and distributional measures of amino acids. Their method has been shown to successfully improve the predictive performance and compare favorably to AAC-based features (Vijayakumar and Ptv, 2015). In another work, Hajisharifi *et al.* developed also a SVM-based predictor, but used Chou's pseudo acid amino composition, which considers both the local correlation and sequence-order information of residues to improve the prediction of ACPs (Hajisharifi *et al.*, 2014). More recently, Chen *et al.* described a sequence-based predictor called iACP based on the optimal g-gap dipeptide components, by exploring the correlation between long-range residues and sequence-order effects (Chen *et al.*, 2016). This predictor exhibited the best predictive performance among several existing predictors (Chen *et al.*, 2016).

As aforementioned, last few years have witnessed the development of machine learning-based methods, especially related to effective feature representation algorithms. Currently there are different types of sequence-based feature descriptors available. A straightforward way is to integrate different types of features as the input to train a classifier and build a predictive model. However, this will lead to the dimensional disaster if the dimension of integrated features is too high, or on the other hand, simply integrating different feature types can also cause information redundancy and thereby influence the predictive performance. A more efficient way is required in order to achieve the maximum use of the information embedded in these feature descriptors. In addition, the majority of the existing feature descriptors use only sequential information, such as AAC, to build predictive models. This might be not informative enough to accurately discriminate true ACPs from non-ACPs, as the intrinsic properties of ACPs are not considered. Previous studies have suggested that ACPs significantly differ from non-ACPs in terms of physicochemical properties (Diana *et al.*, 2013). Therefore, integration of physicochemical information might be useful.

In this paper, we propose a novel feature representation learning scheme to integrate the class information of data into features and effectively explore a set of more informative features. To further improve the representation ability of features, we employ a well-known feature selection technique namely minimum Redundancy Maximum Relevance (mRMR; Ding and Peng, 2003; Peng *et al.*, 2005) to generate an optimal feature set containing only five most discriminative features. Importantly, the number of our generated features is the fewest among existing features, yet they can achieve significantly better performance than currently available feature descriptors reported in the literature, indicating the effectiveness of the proposed feature algorithm. Using the proposed features as the input to train the SVM classifier, we develop a high-throughput sequence-based predictor for the identification of anti-cancer peptides from protein sequences at a large scale. As an implementation of this approach, we establish a user-friendly online web server of **A**nti-**C**ancer peptide **Pred**ictor with **F**eature representation **L**earning (ACPred-FL), which is publicly available at http://server.malab.cn/ACPred-FL. It is noteworthy that the web server provides two modes to facilitate users' different needs, i.e. the classification mode and prediction mode. The former is for the purpose of identifying peptide sequences as ACPs or non-ACPs, while the latter is aimed at providing users with the option of mining potential ACPs from protein sequences. The detailed guideline of ACPred-FL can be found in the Supplementary Material.

2 Materials and methods

2.1 Datasets

Previous studies have suggested that a well-established dataset is crucial for building a robust and reliable predictive model (Wei *et al.*, 2017a, b, c; Xing *et al.*, 2017). In this study, we constructed a new dataset for ACP identification. A standard dataset, used for construction of binary predictive models, usually comprises of a positive dataset and a negative dataset. As previously described (Chen *et al.*, 2016; Tyagi *et al.*, 2013; Vijayakumar and Ptv, 2015), experimentally validated ACPs are used as positive samples, while anti-microbial peptides (AMPs) with no anti-cancer activity (Tyagi *et al.*, 2013) are collected as negative samples. For our positive set, we also collected experimentally validated ACPs as positive samples as previously suggested (Chen *et al.*, 2016; Tyagi *et al.*, 2013; Vijayakumar and Ptv, 2015). In order to collect a sufficient number of positive samples, we extracted experimentally validated ACPs from three main resources: Chen *et al.*' s work (Chen *et al.*, 2016), Tyagi *et al.*' s work (Tyagi *et al.*, 2013) and a public ACP database

CancerPPD (Tyagi *et al.*, 2015). It is noteworthy that CancerPPD is a recently released ACP database encompassing 2849 ACPs, which is the largest ACP database to date. As a result, a total of 3212 ACPs were extracted as the initial positive samples, of which 138 were from Chen *et al.*'s work, 225 from Tyagi *et al.*'s work and 2849 from the CancerPPD database, respectively. Thus, it is reasonable to assume that our initial positive dataset included the majority of experimentally validated ACPs to date in the literature. For collection of negative samples, we employed the negative dataset containing 2250 samples prepared by Tyagi *et al.* This negative dataset included the extracted AMPs from the above databases like APD, CAMP and DADP for which no anti-cancer activity had been reported in the literature and thus were considered as non-ACPs. To avoid performance over-estimation introduced by the homology bias, peptide sequences in both the positive and negative datasets with more than 90% sequence identity were removed using the CD-HIT program with the threshold set at 0.9 (Li and Godzik, 2006). Finally, 332 ACPs and 1023 non-ACPs were retained in the positive and negative datasets, respectively.

Training dataset

The purpose of constructing a training dataset is to train a predictive model and evaluate the model performance. We randomly selected 250 ACPs and an equal number of non-ACPs from the positive and negative datasets, respectively. The rationale for choosing an equal number of negative samples with positive samples is based on a well-known fact that classification techniques, particularly machine learning techniques generally perform better on balanced datasets (Tyagi *et al.*, 2013). For brevity, we named the training dataset as *ACP500*. The predictive models were trained on this dataset.

Independent test dataset

To validate the generalization ability of predictive models, we further constructed an independent test dataset. This dataset included 82 experimentally validated ACPs from the positive dataset, and an equal number of non-ACPs selected from the negative dataset. Note that none of the peptides in this independent test dataset appeared in the training dataset, ensuring a fair assessment of model performance. This dataset was referred to as *ACP164*.

Note that the *ACP500* and *ACP164* datasets mentioned above can be downloaded from the website: http://server.malab.cn/ACPred-FL/.

2.2 Prediction framework of the proposed predictor

To identify novel potential ACPs within proteins, we present a machine learning-based method called ACPred-FL. Figure 1 illustrates the framework of our proposed approach. Here, we briefly describe the workflow of ACPred-FL in the following three major steps. Firstly, using protein primary sequences as the input, proteins are scanned residue by residue using a sliding peptide window with *m* residues long. In this way, numerous peptide sequences can be generated, while those peptides identical to precursors can be further filtered out. Secondly, the remaining peptides will be subjected to the feature representation learning scheme, for which each sequence is encoded with a five-dimensional feature vector. Refer to the 'Feature representation learning scheme' Section for more technical details of this scheme. Thirdly, the resulting feature vectors are fed into a well-trained predictive model, which was trained with an SVM classifier on the *ACP500* dataset (containing balanced positive and negative samples of 250 ACPs and 250 non-ACPs). Ultimately, the SVM model assigns a prediction score to each candidate peptide. The prediction score ranges from 0 to 1. A higher score a peptide achieves, a higher probability it is likely to be a true ACP. The predictor considers the peptides as potential ACPs if their prediction scores are higher than 0.5, and non-ACPs otherwise. The proposed feature representation methods and the trained classifier are described in detail in the following sections.

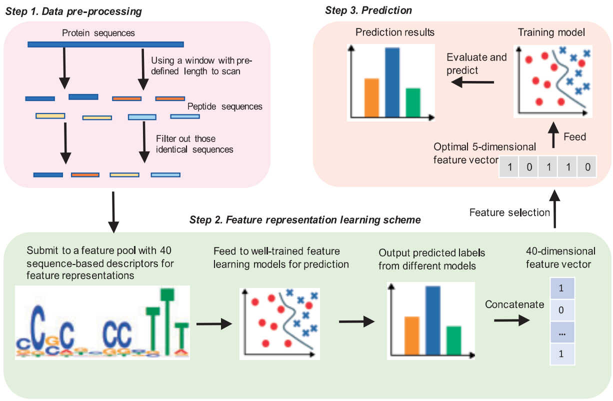
Step 1. Data pre-processing
Protein sequences
Using a window with pre-defined length to scan
Peptide sequences
Filter out those identical sequences
Step 3. Prediction
Prediction results
Training model
Evaluate and predict
Feed
Optimal 5-dimensional feature vector
1 0 1 1 0
Feature selection
Step 2. Feature representation learning scheme
Submit to a feature pool with 40 sequence-based descriptors for feature representations
Feed to well-trained feature learning models for prediction
Output predicted labels from different models
40-dimensional feature vector
Concatenate
1
0
...
1

Fig. 1. Flowchart of ACPred-FL. There exist three major steps: Firstly, given protein primary sequences as the input, they are scanned residue by residue using a peptide window with *m* residues to generate numerous peptides; those peptides that are identical to others will be filtered out; Secondly, the remaining peptides are subjected to the feature representation learning scheme, and each of them is encoded with a five-dimensional feature vector; Thirdly, the resulting feature vectors are fed into a predictive model, which is trained with a SVM classifier on the *ACP500* dataset. Ultimately, the SVM model generates a prediction score for each peptide in the range from 0 to 1. The predictor considers the peptides as potential ACPs if their prediction scores are higher than 0.5, and non-ACPs otherwise

2.3 Feature representation of peptides

A peptide sequence can be represented as:

$$\mathbf{P} = p_1 p_2 p_3 \ldots p_L$$

where p_1 denotes the 1st residue in the peptide **P**, p_2 denotes the 2nd residue in the peptide **P** and so forth. L denotes the length of **P**. Note that the residue p_i is an element of the standard amino acid alphabet {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. To train a machine learning model, the first step is to formulate diverse-length peptides as fixed-length feature vectors. In this study, we exploited multiple feature representation methods, as described below.

Binary profile Features

As described above, there are 20 different amino acids in the standard amino acid alphabet. Each amino acid type is encoded with the following feature vector composed of 0/1. More specifically, the first amino acid type A in the alphabet is encoded as b(A)=(1, 0,. . ., 0), the second amino acid type C is encoded as b(C)=(0, 1,. . ., 0), and so forth. Subsequently, for a given peptide sequence **P**, its N-terminus with the length of k amino acids was encoded as the following feature vector:

$$\text{BPF}(k) = [b(p_1), b(p_2), \ldots, b(p_\text{k})]$$

where k denotes the length of the N-terminus of the peptide **P**. Thus, the dimension of $BPF(k)$ is $20 \times k$.

Overlapping Property Features

The amino acids are classified into 10 groups based on their physicochemical properties (Dou *et al.*, 2014; see Supplementary Table S2 of Supplementary Material). Note that different groups might overlap, since a specific amino acid type may have two or more physicochemical properties. To reflect the correlation of different properties, we calculated a 10-bit vector (composed of 0/1 elements) to represent each amino acid of the N-terminus of a peptide. Similarly, the position of the bit of this 10-bit vector is set to 1, if the amino acid belongs to a corresponding group and 0 otherwise (Dou *et al.*, 2014).

Twenty-One-Bit Features

Alternatively, the standard amino acid alphabet can be also categorized based on the following seven physicochemical properties: polarity, normalized Van der Waals volume, hydrophobicity, secondary structures, solvent accessibility, charge and polarizability (Govindan and Nair, 2011). For each physicochemical property, the standard alphabet is clustered into three groups. Note that for this encoding scheme, any two groups are not overlapped. Supplementary Table S3 in the Supplementary Material shows the divisions of a standard alphabet based on the seven physicochemical properties. We ended up with a total of 21 amino acid groups for all the seven physicochemical properties. Similar to the overlapping property feature (OPF) encoding, each residue of the N-terminus of **P** is encoded as a 21-bit vector composed of 0/1 elements, where the position of the bit is set to 1 if the amino acid belongs to the corresponding group, and 0 otherwise. The dimensionality of feature vector is $21 \times k$.

Composition-Transition-Distribution (CTD)

This method is used to describe the global composition of amino acid property for each peptide sequence (Li *et al.*, 2011). It contains three feature descriptors: Composition (C), Transition (T) and Distribution (D) (Dubchak *et al.*, 1999). The composition descriptor computes the percentage frequency of a particular amino acid property group in the peptide sequence; the transition descriptor characterizes the percent frequency of amino acids of a particular property to be followed by amino acids of another property; the distribution descriptor describes the fractions of the entire peptide sequence where the first, 25, 50, 75 and 100% of amino acids of a particular property are placed within the peptide sequence, respectively. Refer to (Li *et al.*, 2011) for details on how to calculate the three descriptors. Using the composition-transition-distribution (*CTD)* method described above, 21 descriptors can be calculated for each of the three amino acid property groups. This leads to a total of 63 features to represent each peptide sequence.

AAC

The AAC is based on calculation of the occurrence frequency of each amino acid type in a peptide sequence. AAC can be formulated as follows:

$$\mathrm{AAC}(\mathbf{P}) = (f_1,\ f_2, \ldots, f_{20})$$

where $f_i = R_i/L$ ($i = 1, 2, \ldots, 20$) is the percent composition of amino acid type i, R_i is the number of type i appearing in the peptide, while L is the length of the peptide. The dimension of the *AAC* descriptor is 20.

G-gap dipeptide composition

Dipeptide composition is defined as the fraction of any two adjacent residues (p_ip_{i+1}) as a dipeptide pair. It measures the correlation of any two adjacent residues in a sequence. However, it is obvious that the correlation information of those intervening (non-adjacent) residues ($p_ip_j; j-i>1$) would be lost. Thus, the g-gap dipeptide composition (GDC) is used in this study. *GDC* encapsulates the composition and local order information of any two interval residues within a peptide sequence. It is represented as:

$$GDC(g) = (fv_1^g, fv_2^g, \ldots, fv_{400}^g)$$

where fv_i^g is the occurrence frequency of the i-th ($i = 1, 2, \ldots, 400$) g-gap dipeptide. It is computed as:

$$fv_i^g = \frac{\mathrm{O}_i^g}{\sum_{i=1}^{400} \mathrm{O}_i^g}$$

where O_i^g represents the occurrence number of the i-th g-gap dipeptide in a peptide sequence. The dimension of the *GDC* descriptor is 400.

Adaptive skip dipeptide composition

We have recently presented a modified dipeptide composition, called adaptive skip dipeptide composition (ASDC; Wei *et al.*, 2017a, b, c). This descriptor sufficiently considers the correlation information present not only between adjacent residues but also between intervening residues. For given a peptide sequence P, the feature vector for ASDC is represented by:

$$\mathrm{ASDC} = (fv_1,\ fv_2, \ldots, fv_{400})$$

where fv_i is calculated by

$$fv_i = \frac{\sum_{g=1}^{L-1} \mathrm{O}_i^g}{\sum_{i=1}^{400} \sum_{g=1}^{L-1} \mathrm{O}_i^g}$$

where fv_i denotes the occurrence frequency of all possible residue pairs with $\leq L-1$ intervening residues. In the case of $k = 1$, the feature vector would exactly resemble the dipeptide composition. The dimension of the ASDC descriptor is 400.

2.4 Feature representation learning scheme

In the present study, we propose a new feature representation learning scheme as illustrated in Figure 2. The procedures of this scheme are described in the following sub-sections.

Step 1. Construct an initial feature pool

As described in the preceding section, we extracted seven encoding schemes of feature descriptors with respect to AAC and physicochemical properties, including binary profile feature (*BPF*), *OPF*, twenty-one-bit feature (*TBF*), *CTD*, *AAC*, *GDC* and *ASDC*. Note that two parameters k and g have to be fine-tuned for specific feature types. The parameter k corresponds to the first three feature types: *BPF*, *OPF* and *TBF.* In this study, the value of k was set from 1 to 11, where 11 was the minimal length of the peptides in the dataset. By varying the k value from 1 to 11, for each of the three encoding scheme *BPF*, *OPF* and *TBF*, we generated 11 feature groups. On the other hand, g is the parameter of the *GDC* descriptor. The value of g was chosen from 1 to 4, as high-dimensional feature vector would be generated if $g > 4$. As a result, we obtained four feature groups for *GDC.* Finally, we obtained an initial feature set including 40 feature groups based on the seven feature encodings. For the sake of brevity, the j-th feature group is denoted as FG_j ($j = 1, 2, \ldots, 40$). The purpose for using a wider range of parameter values of k and g is to include as much sufficient information as possible in the initial feature fool. See Supplementary Material (Supplementary Table S4) for details.

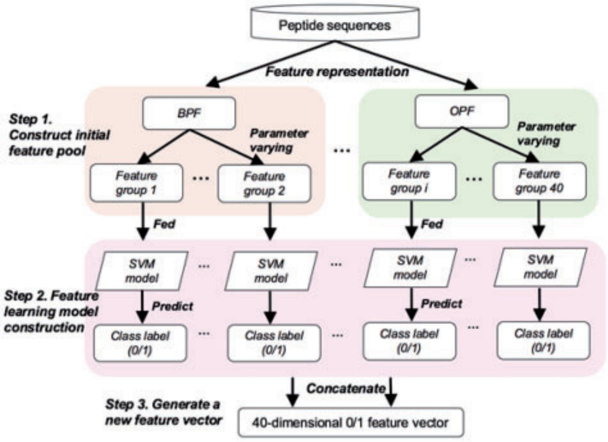

Fig. 2. The proposed feature representation learning scheme. First, peptide sequences are subjected to feature presentation using seven feature descriptors. To incorporate sufficient information, we alter the parameters of the feature descriptors, and then generate 40 feature groups to form the initial feature pool; Second, the resulting feature groups are then fed into well-trained SVM models for predicting the class labels, and finally, the predicted labels (0/1) from the SVM models are concatenated to generate a new feature vector for representation of peptide sequences

Step 2. Construct a feature representation learning model

For each FG$_j$ ($j = 1, 2, \ldots, 40$) in the feature pool, we subsequently trained a corresponding SVM model on the *ACP500* dataset, denoted as $M(\mathrm{FG}_j)$. We obtained totally 40 SVM models. Each model was denoted as a baseline model.

Step 3. Learn a new feature vector

For a given peptide sequence **P**, we used each baseline model [$M(\mathrm{FG}_j)$] to predict its class label. The predicted label by each model was then used as a 'feature'. In our experiments, the positive samples (i.e. ACPs) are labeled as 0, whereas the negative samples (i.e. non-ACPs) are labeled as 1. Therefore, if the baseline model predicts the sequence **P** as a true ACP, then the feature value is assigned as 0; otherwise assigned as 1. Finally, the sequence **P** is encoded with a new feature vector by concatenating all the features generated by all the 40 models, which is presented by:

$$FV_{\mathrm{SVM}}(\mathbf{P}) = \left(\Upsilon\left(\mathbf{P},\ M(FG_1)\right),\ \Upsilon\left(\mathbf{P},\ M(FG_2)\right), \ldots, \left(\mathbf{P},\ M(FG_{40})\right)\right)$$

where FV_{SVM} (**P**) is the feature vector of the sequence **P**; $Y(\mathbf{P}, M(FG_j))$ is the prediction output of each model for the sequence **P**. Clearly the dimension of the new feature vector is 40. In other words, the peptide sequence **P** is eventually represented by a 40-dimensional feature vector.

2.5 Feature selection

To improve the feature representation ability, we utilized a two-step feature selection strategy to optimize the selection of features. The first step is to rank the 40 features based on their classification importance, and the second is to search for the optimal feature subset from the ranked features. This two-step feature selection strategy is described as follows.

For the first step, we adopted a well-known feature selection technique namely mRMR (Ding and Peng, 2003; Peng *et al.*, 2005) to rank the features in our feature set. Using the mRMR method, we generated a ranking list of features with respect to their classification importance. Higher ranked features in this list have a better trade-off between the maximum relevance and the minimum redundancy, indicating that it is more informative. More technical details can be referred to Supplementary Material.

For the second step, we used the sequential forward search (SFS) to select the optimal feature subset from the ranked feature set (Whitney, 2006). For SFS, features from the ranked feature set were added one-by-one from lower rank (higher index) to higher rank (lower index) each time, and were used to re-construct the SVM-based prediction model on the 10-fold cross validation test. Finally, the feature subset, with which the prediction model would achieve the best performance, was recognized as the optimal set. The details of the feature selection results are discussed in Section 3.3.

2.6 Support vector machine

SVM is a powerful machine learning algorithm for binary classification (Furey *et al.*, 2000). In this work, we employed the SVM to train our predictive models. The basic idea of SVM is to map the input data onto a high-order feature space and then construct an optimal separating hyperplane to maximize the margin between the positive and negative samples. However, it often occurs that the mapped feature space is not linearly separable. To solve this, the kernel function is used to transform the original non-separating feature space into the linearly separating feature space, where the optimal classification hyperplane can be established. There are three types of kernel functions commonly used in SVM, including Polynomial, Radial basis function (RBF) and Gaussian. Through a comparative study of three kernels (Supplementary Material), we finally used the RBF kernel as the kernel function of SVM due to its better performance. The RBF kernel is defined as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{e}^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}$$

where γ is a kernel parameter, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the feature vectors of peptide sequences i and j, respectively. To implement the SVM, we used the LIBSVM package (version 3.22). To achieve the optimal classification performance, the regularization parameter C and kernel parameter γ in SVM were optimized using a grid search approach. The search ranges for the above two parameters are given as follows:

$$\begin{cases} 2^{-5} \le C \le 2^{15}, & \textit{with step size of } 2 \\ 2^{-15} \le \gamma \le 2^{-5}, & \textit{with step size of } -2 \end{cases}$$

2.7 Performance measurement

For performance evaluation, we used four metrics commonly used in binary classification tasks, including Sensitivity (SE), Specificity (SP), Accuracy (ACC) and Mathew's correlation coefficient (MCC). They are calculated as follows:

$$\begin{cases} SE = \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} * 100\% \\ SP = \dfrac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} * 100\% \\ ACC = \dfrac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} * 100\% \\ MCC = \dfrac{\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FN})(\mathrm{TP} + \mathrm{FP})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \end{cases}$$

where TP, TN, FP and FN represent the numbers of true positives, true negatives, false positives and false negatives, respectively. The two metrics, SE and SP, measure the prediction ability of a predictor for the positives and negatives, respectively (Wu *et al.*, 2013), while the other two metrics, ACC and MCC, are used to evaluate the overall performance of a predictor.

Moreover, we performed 10-fold cross validation to measure the performance of the model. The procedure of this validation method can be found in Supplementary Material.

2.8 Receiver operating characteristic (ROC) curve

ROC curve (Hsieh and Turnbull, 1996) is often used to measure the overall performance of a binary classifier system. It is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) under different classification thresholds. TPR is also known as sensitivity described in the above section, while FPR can be calculated as (1-specificity). We also calculated the area under ROC curve (AUC) to evaluate the predictive performance. The value of AUC ranges from 0.5 to 1. When the AUC score of a predictor is close to 1, the predictor is considered as a perfect predictor; when the AUC score is 0.5, it corresponds to a random predictor. A larger AUC value indicates that the model achieves a better and more robust predictive performance.

3 Results and discussion

3.1 Impact of different classifiers on feature representation learning models

In our feature representation learning scheme, we used the SVM classifier to build the feature representation learning model, and generate a feature descriptor FV_{SVM}. In fact, the learned feature descriptor is closely related with the prediction of the classifiers. Thus, one might wonder whether use of different classifiers to build feature representation learning models impacts the representation ability of the feature descriptor, and also influences the predictor's performance. For this purpose, we used two classifiers, Random Forest (RF) and Naïve Bayes (NB), to substitute the original SVM classifier to build new feature representation learning models. The resulting two new feature descriptors were denoted as FV_{RF} and FV_{NB}, respectively. To measure the quality of the two feature descriptors, we compared FV_{RF} and FV_{NB} with our original feature descriptor (FV_{SVM}). In order to make a fair comparison, all the three feature descriptors were evaluated using the SVM algorithm by performing both 10-fold cross validation on the *ACP500* dataset and the independent test on the *ACP164* dataset.

Table 1 provides the 10-fold cross validation and independent test results of the three feature descriptors. In the case of 10-fold cross-validation results, we observed that the descriptor (FV_{SVM}) achieved a similar SE to other two descriptors (FV_{RF} and FV_{NB}); the SEs of the three descriptors were 84.8%, 84.0% and 84.8%, respectively, while the SP (96.0% for FV_{SVM}) was significantly higher than that of the other two descriptors (93.2% for FV_{RF}, and 90.8% for FV_{NB}, respectively). Thus, the higher SP contributed to the improvement of the overall performance of our descriptor. Specifically, our descriptor achieved an ACC of 90.4% and an MCC of 0.813, which are 1.4—3% and 3—6.3% higher than that of the other two descriptors, respectively. As for the independent test results shown in Table 1, we observed a similar tendency to the cross-validation test. Our descriptor also exhibited a better performance that the other two on the independent *ACP164* dataset, achieving 1.8—2.4% and 5.5—5.6% improvement in terms of ACC and MCC, respectively.

Moreover, we also plotted the ROC curves of different descriptors in Figure 3. We can see that the AUC of SVM-based descriptor was similar to that of the RF-based descriptor, both of which were higher than that of the NB-based descriptor. Altogether, the observed results suggest that the proposed SVM-based feature descriptor was capable of generating more informative features for representing ACPs in contrast to other classification algorithm-based feature descriptors (e.g. NB and RF). Using our feature descriptor for representation, ACPs and non-ACPs were distributed more differentially in the feature space, thereby leading to an improved performance. Therefore, it can be concluded that the classifiers used for building the feature representation learning models are closely associated with the resulting predictive performance. This is presumably because the quality of feature representation highly relies on the classifier. If the classifier performs well, it generates high-quality feature representation; otherwise, it results in low-quality feature representation, which is considered as the noise by the classifier and thus has an impact on the predictive performance.

Table 1. Predictive performance of different feature descriptors generated by different machine learning classifiers, including RF, NB and SVM

Feature descriptor	SE (%)	SP (%)	ACC (%)	MCC	AUC	TP	FP	TN	FN
10-fold cross-validation test on the ACP500 dataset									
FV_{RF}	84.8	93.2	89.0	0.783	0.94	212	17	233	38
FV_{NB}	84.0	90.8	87.4	0.750	0.92	210	23	227	40
FV_{SVM}	84.8	96.0	90.4	0.813	0.94	212	10	240	38
Independent test on the ACP164 dataset									
FV_{RF}	84.1	91.5	87.8	0.758	0.96	69	7	75	13
FV_{NB}	86.6	90.2	88.4	0.759	0.93	71	8	74	11
FV_{SVM}	82.9	97.6	90.2	0.814	0.95	68	2	80	14

Notes: we used the SVM algorithm to evaluate the quality of feature descriptors with the 10-fold cross-validation test on the ACP500 dataset and the independent test on the ACP164 dataset.

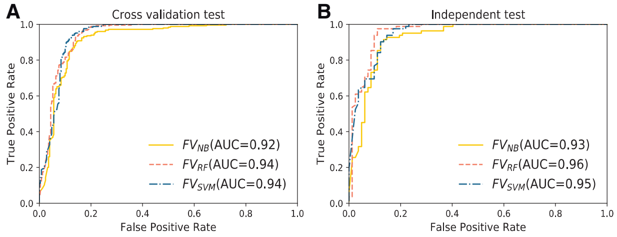

Fig. 3. Predictive performance of different feature descriptors on the 10-fold cross-validation and independent tests. (**A**) The ROC curves illustrating the 10-fold cross-validation performance of three types of feature descriptors. (**B**) The ROC curves illustrating the independent test performance of the three types of feature descriptors

3.2. Impact of different machine learning algorithms on the predictive performance

In this study, we used *SVM* as the underlying classification algorithm to build the prediction engine of our predictor. To investigate the effect of different machine learning algorithms on the performance, we also compared the performance of *SVM* with another two commonly used algorithms (i.e. *RF* and *NB*). To make an objective comparison, we respectively trained the classifiers of the three algorithms with the FV_{SVM} descriptor on the same *ACP500* dataset. The performance of the three classifiers on the 10-fold cross validation is illustrated in Figure 4. As shown in Figure 4A, *SVM* achieved a better performance than the other two algorithms in terms of three metrics: SP, ACC and MCC, while performing slightly worse than RF in

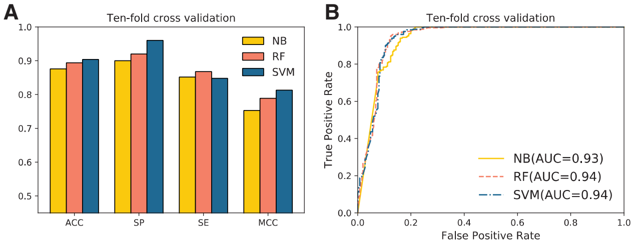

Fig. 4. Predictive performance of models based on different classifiers: (**A**) The ROC curve illustrating the 10-fold cross validation performances of the proposed features but with three different classifiers (NB, RF and SVM). (**B**) The ROC curve illustrating the independent test performances of three types of the proposed features but with three different classifiers (NB, RF and SVM)

terms of SE. In addition, we plotted the ROC curves of the three classifiers in Figure 4B. As can be seen, the *SVM* and *RF* achieved a similar AUC of 0.94, which was slightly higher than that of *NB* (AUC = 0.93). In summary, the better performance of SVM in terms of the three metrics (ACC, MCC and AUC) indicates that it provides a more discriminative power than the other two common algorithms and hence is more suitable for dealing with the task of discriminating true ACPs from non-ACPs. Additionally, choosing a more informative or powerful classifier to build the model might be potentially useful for further improving the predictive performance.

3.3 Effect and analysis of feature selection

In an effort to build an optimal predictive model, we used a two-step feature selection technique to identify a feature subset from the 40-dimensional feature vector that led to the best performance. The first step was to rank the features according to their classification importance using the mRMR program, and the second step was to use the SFS approach to select the optimal feature subset from the ranked feature list.

Applying the mRMR to the 40-dimensional vector, we obtained a ranked feature list, where the features were sorted according to their classification importance. The importance scores of all the 40 features are presented in Supplementary Table S5 of Supplementary Material. We plotted the distribution of the features in Figure 5A, where the classification importance decreased from the left to the right along the *x*-axis. As can be seen, 'fea2' is located at the leftmost of the *x*-axis, suggesting that it is the most important feature for the classification amongst all the analyzed features. It is worth mentioning that in Figure 5A, 'fea2' denotes the 2nd feature in our original feature vector. Afterwards, we used the SFS approach to select the optimal feature subset from the ranked feature list. The SFS curve is depicted in Figure 5B. The detailed results of feature selection using the SFS approach can be found in Supplementary Material (Supplementary Table S6). As shown in Figure 5B, there was no significant change in the performance with the increase of feature numbers at the very beginning. When the feature number increased to 5, the ACC and MCC of the predictive model significantly were improved to 91.4% and 0.835, respectively. Further, when the feature number increased to 15, the model achieved the best performance with an ACC of 92.0% and MCC of 0.849, respectively. It can be clearly seen from Figure 5B that these two were the two best predictive models amongst all 40 models. We further compared these two models and found that the former model only used five features. Although the feature number of the former model was one third of that of the latter model, its performance was only

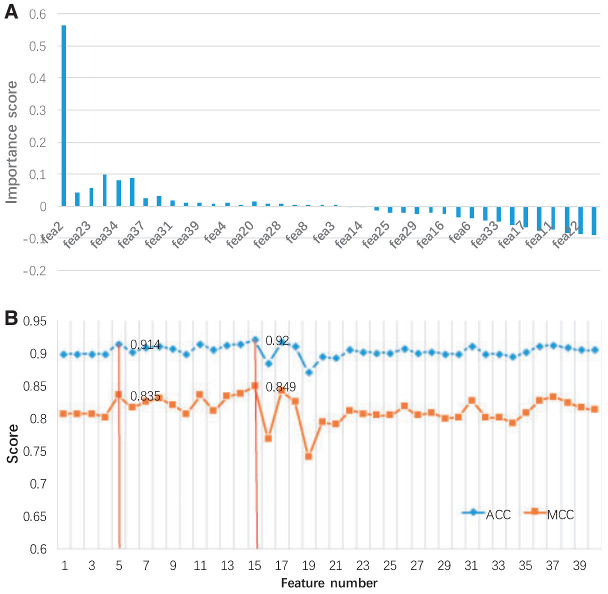

Fig. 5. mRMR feature selection of the proposed features. (**A**) The classification importance scores for the 40 generated features. Note that 'fea1' denotes the 1st feature among all the generated features. (**B**) SFS curve for the predictive model with respect to the ACC and MCC. The *x*- and *y*-axis represent the feature number *t* (ranging from 1 to 40) and the predictive performance, respectively. The blue and orange plots represent the SFS curves of ACC and MCC, respectively (Color version of this figure is available at *Bioinformatics* online.)

slightly worse than that of the latter. Therefore, we considered that this model with five features only was the optimal. Accordingly, the first five features in the ranked feature list were concatenated to constitute our final feature vector in the final predictor.

To analyze the selected optimal features, we investigated their composition. These five optimal features were 'fea2', 'fea38', 'fea23', 'fea7' and 'fea34'. In fact, these features were derived from SVM models with five feature descriptors: BPF ($k = 2$), GDC ($g = 3$), OPF ($k = 1$), BPF ($k = 7$) and *CTD*, respectively. Of the five feature descriptors, four out of five used the physicochemical properties, while the other one used the sequential information to encode peptide sequences. This suggests that the physicochemical information has the strongest feature representation ability of ACPs.

3.4 Comparison between the optimal features and individual feature descriptors

As aforementioned, our optimal features were generated from five individual feature descriptors: BPF ($k = 2$), GDC ($g = 3$), OPF ($k = 1$), BPF ($k = 7$) and *CTD*. To investigate the effectiveness of our proposed feature representation learning strategy, we compared the predictive performance between our features and the five individual feature descriptors. For the purpose of making a fair comparison, we performed 10-fold cross validation tests using the training dataset. The results are presented in Table 2. It can be clearly seen that compared with the other feature descriptors, our features achieved the best predictive performance in terms of ACC and MCC. Specifically, our features achieved an ACC of 91.4% and an MCC of 0.835, which were 1.6–18.8% and 2.8–29.5% higher than that of the other feature descriptors, respectively. In terms of SP, the OPF ($k = 1$) descriptor obtained the highest SP of 100%, which was slightly higher than our SP (98%). However, on the other hand, this descriptor obtained the worst performance (ACC = 72.6% and MCC = 0.540) amongst the compared feature descriptors, because

Table 2. Ten-fold cross-validation results of our optimal feature descriptor with individual feature descriptors

Feature descriptors	SE (%)	SP (%)	ACC (%)	MCC	TP	FP	TN	FN
BPF($k = 2$)	81.6	98.0	89.8	0.807	204	5	245	46
GDC ($g = 3$)	79.2	78.0	78.6	0.572	198	55	195	52
OPF ($k = 1$)	45.2	100	72.6	0.540	113	0	250	137
BPF ($k = 7$)	82.8	90.4	86.6	0.734	207	24	226	43
CTD	78.0	86.0	82.0	0.642	195	35	215	55
The proposed features	84.8	98.0	91.4	0.835	212	5	245	38

its SE (45.2%) was significantly lower than that of other the feature descriptors (78–82.8%) and our optimal features (84.8%). Apparently, the best performance of our optimal features contributed to the best SE and the runner-up SP. This suggests that the optimal features could identify more true ACPs (true positives) and at the meanwhile resulted in fewer false positives. Notably, our feature set contained only five features, which is far fewer than that of the five individual feature descriptors.

To investigate the effectiveness of our features, we further calculated and compared the distribution of positive and negative samples by mapping them to the feature space of the optimal five-dimensional features and the five individual feature descriptors, which is illustrated in Figure 6. On each feature dimension, we computed the mean and SD. An important observation is that the positive and negative samples were distributed more differentially in the feature space that was constructed by our optimal feature descriptor as compared to that by other feature descriptors. This might explain why our feature descriptor led to the most informative prediction of ACPs.

3.5 Comparison of the proposed predictor and the state-of-the-art predictors

To evaluate the predictive performance of the proposed ACPred-FL predictor, we benchmarked its performance against three other state-of-the-art predictors, including Anti-CP, Hajisharifi's method and iACP. It is noteworthy that Anti-CP had two predictive models, termed Anti-CP_AAC (trained with AAC) and Anti-CP_DC (trained with dipeptide composition). Both models were included in the comparison. Thus, a total of four predictive models from the three predictors were included to compare with the proposed ACPred-FL. Note that all the predictors above are based on SVM. To make a fair comparison, all predictors were performed using their optimal SVM parameters.

Performance comparison on the 10-fold cross validation using the ACP500 dataset

We performed the proposed ACPred-FL and the four predictive models on the same *ACP500* dataset (containing 250 ACPs and 250 non-ACPs) and evaluated their performances with 10-fold cross validation. Figure 7A illustrates the predictive performance in terms of four metrics (SE, SP, ACC and MCC) on this dataset. As shown in Figure 7A, we observed that the performance of the proposed ACPred-FL was significantly better than the other predictors. For example, the SE, SP, ACC and MCC of our ACPred-FL were 84.8%, 98.0%, 91.4% and 0.835, respectively (Supplementary Table S7 in the Supplementary Material), which were 12.0—17.6%, 11.6—13.8%, 12.0—14.8% and 24.2—29.3% higher than that of existing predictors, respectively. It can be seen that the improvement of the performance by our predictor is significant. Another interesting observation is that the SEs for all predictors were somehow lower than their SPs. This suggests that they effectively reduce the generation of false positives, but likely miss some true positives due to their low sensitivity. Furthermore, we compared ACPred-FL with the other state-of-the-art predictors using ROC curves (Fig. 7B). As can be seen, ACPred-FL also outperformed the other predictors in terms of AUC (0.94). It achieved a remarkable improvement of 7% (compared to Anti-CP_DC)–10% (compared to Anti-CP_AAC and Hajisharifi's method). These results indicate that ACPred-FL has a stronger capability than the existing predictors for identifying true ACPs from non-ACPs.

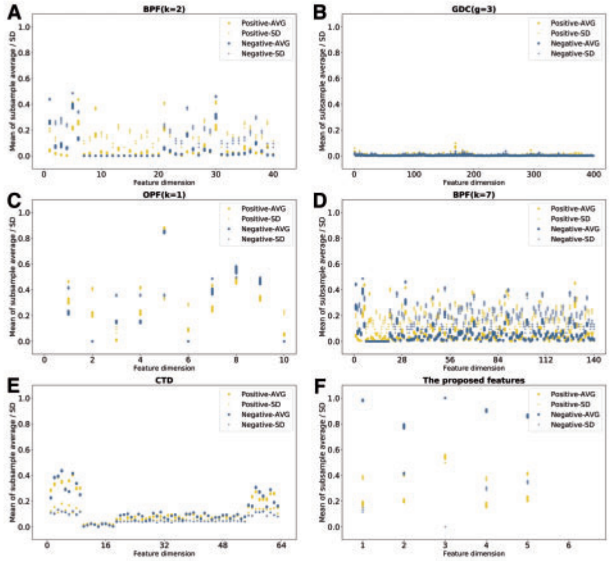

Fig. 6. Distribution of the positive and negative samples with respect to different feature descriptors. (A) - (F) represent the distributions of BPF (k=2), GDC (g=3), OPF (k=1), BPF (k=7), CTD, and the proposed features, respectively. 90% of the positive samples (ACPs) and negative samples (non-ACPs) were randomly selected at each sampling step. This sampling procedure was repeated 20 times to obtain sub-sample average feature vectors. On each feature dimension, we calculated the mean and SDs of the feature vectors

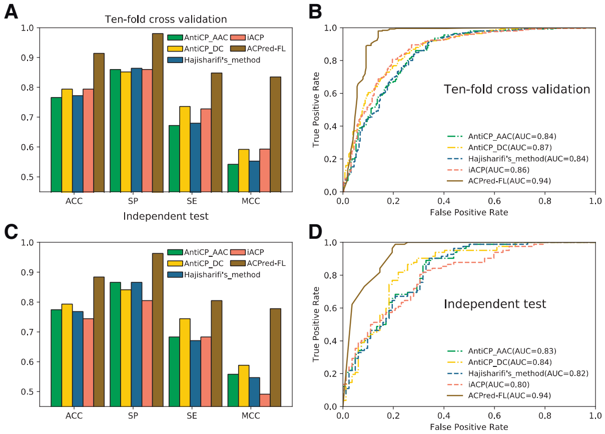

Fig. 7. Performance comparison of the proposed ACPred-FL and four state-of-the-art predictors. (**A**) Ten-fold cross validation results of the proposed ACPred-FL and the existing four predictive models on the *ACP500* dataset. (**B**) ROC curves of the proposed ACPred-FL and existing four predictive models on the ACP500 dataset. (**C**) Independent test results of the proposed ACPred-FL and existing four predictive models on the ACP164 dataset. (**D**) ROC curves of the proposed ACPred-FL and existing four predictive models on the *ACP164* dataset

Performance comparison on the independent test using the ACP164 dataset

In order to validate the robustness of ACPred-FL, we further compared it with the above three predictors on the *ACP164* dataset (containing 82 ACPs and 82 non-ACPs). In order to make a fair comparison, two important conditions should be applied to all the compared predictors. The first condition is to use the same dataset (*ACP500*) for model training. The other condition is to ensure lower sequence identities between the testing dataset and the training dataset, otherwise it will lead to an overestimation of the performance if the sequences in the testing dataset had higher identities with those in the training dataset. As described in the 'Datasets' Section, the *ACP164* and *ACP500* datasets shared no more than 90% sequence identity, thus excluding the possibility of performance overestimation introduced by sequence identities. Figure 7C shows the performance of ACPred-FL and the four predictive models, while Figure 7D presents their corresponding ROC curves. As can be seen from Figure 7C and D, we observed similar results as discussed in the previous sub-section. Our predictor again outperformed the existing methods on the independent test dataset, indicating that our model was robust for predicting ACPs. The detailed prediction results on the independent test are provided in Supplementary Table S8 in the Supplementary Material.

Performance comparison on other benchmarking datasets

To further compare our results with the existing methods, we applied our proposed ACPred-FL to a previous benchmark dataset from Tyagi *et al.* (Tyagi *et al.*, 2013). This dataset contained a training set (225 positives and 225 negatives) and an independent testing set (50 positives and 50 negatives). To make a fair comparison, all the compared methods were trained on the training set and tested on the testing set. The performance of all the compared methods is illustrated in Figure 8. Similar to the performance on our dataset, consistent results can be observed from Figure 8. Our predictor exhibited a better performance than existing predictors on both cross-validation test (Fig. 8A and B) and the independent test (Fig. 8C and D) using Tyagi's benchmarking dataset. Compared with existing predictors, our predictor achieved a performance gain of 2–3% and 2–7% in terms of AUC on the cross-validation and independent tests, respectively. The detailed performance of different methods is presented in Supplementary Tables S9 and S10 of the Supplementary Material.

Discussion of performance comparison results

As illustrated above, we performed a comprehensive performance comparison of our proposed ACPred-FL and the four state-of-the-art predictors. We benchmarked the different predictors using two datasets (i.e. our dataset and Tyagi's benchmarking dataset) on two performance evaluation settings: cross-validation test and independent test. With regard to the performance comparison results shown in the three preceding sub-sections, the consistently competitive performance on both 10-fold cross-validation and independent tests demonstrate that the proposed ACPred-FL method is capable of achieving an effective and promising performance to identify true ACPs from non-ACPs. In particular, its high specificity indicates that this new method can help reduce the number of false positives and narrow down experimental efforts, which is particularly important for predicting and prioritizing ACPs from a huge amount of protein sequence data at a large scale.

It is worth noting that all the compared predictors are primarily based on SVM. This means that the main difference between our predictor and existing predictors is reflected by the features used. As our predictor performed better than existing predictors, it is reasonable to assume that our features are more discriminative than the previously used features, thereby able to capture the key characteristics between true ACPs and non-ACPs and achieve an improved performance. There exist three main factors that contribute to the performance improvement of our method. First, our feature representation learning model integrates multiple feature descriptors, including not only residue composition and sequence-order information but also physiochemical properties and residue position-specific information. This provides us sufficiently informative and diverse information to build an effective feature representation model. Second, our model used the class information learning from the original feature descriptors. This helped map the original high-dimensional and complicated feature space into low-dimensional but more discriminative one, and third, the two-step selection technique enabled the selection of informative feature subsets and optimization of the feature representation ability of our model. Additionally, as ACPred-FL involved a fewer number of features (only five features used in ACPred-FL, compared with hundreds of features used by other predictors), it can significantly reduce the computational cost. In summary, we have proposed a more accurate and promising predictor for ACPs. ACPred-FL is expected to be a useful tool that complements with existing predictors. More importantly, we have also provided a new effective feature representation model; its design principle and strategy might inspire researchers' ideas to develop improved methods and can be applied to other research topics of sequence analysis.

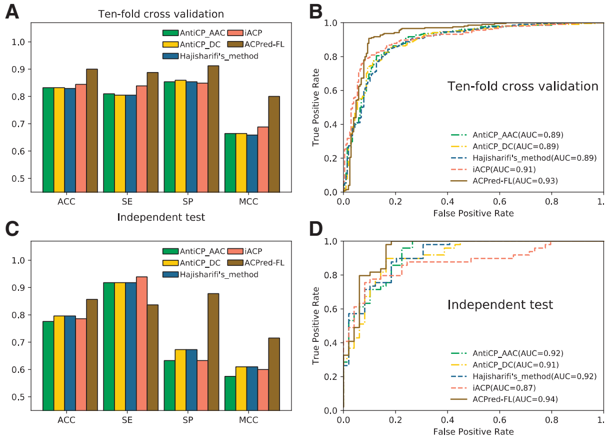

Fig. 8. Performance comparison of the proposed ACPred-FL and four state-of-the-art predictors on Tyagi's dataset. (**A**) Ten-fold cross validation results of the proposed ACPred-FL and the existing four predictive models on the training set of Tyagi's dataset. (**B**) ROC curves of the proposed ACPred-FL and existing four predictive models on the training set of Tyagi's dataset. (**C**) Independent test results of the proposed ACPred-FL and existing four predictive models on the training set of Tyagi's dataset. (**D**) ROC curves of the proposed ACPred-FL and existing four predictive models on the testing set of Tyagi's dataset

4 Conclusion

In this work, we have developed a novel ACP predictor called ACPred-FL. To build an effective predictive model, we propose a novel feature representation learning strategy to automatically learn discriminative features. Extensive benchmarking experiments demonstrated that the proposed features could effectively discriminate

ACPs from non-ACPs in the feature space, thereby providing a significant improvement of the predictive performance as compared to several currently available feature descriptors. Moreover, we examined the discriminative power of our feature vector, and found that its predictive ability was presumably attributed to the integration of the label information predicted by well-trained classifiers based on individual feature descriptors. Furthermore, this feature representation ability is closely associated with the quality of the predicted label information. High-quality predicted labels generated by feature representation learning models are likely to lead to a better predictive performance. Importantly, we have demonstrated that our proposed feature representation learning strategy has the potential to reveal important sequence-level clues that can guide feature extraction algorithms.

Furthermore, to validate the performance of our ACPred-FL predictor, we have conducted a comparative study of state-of-the-art predictors. Experimental results on both the 10-fold cross-validation and independent tests show that this proposed predictor is more effective and promising for identification of ACPs. As an implementation of this work, we have also made available a web server of ACPred-FL for the wider research community to use. We anticipate that ACPred-FL will be a useful tool for discovering novel potential ACPs in a high-throughput and cost-effective manner, thereby facilitating the characterization of their functional mechanisms and accelerating their applications in cancer therapy.

Acknowledgement

The authors would like to thank Prof. Quan Zou for kind suggestions and discussions that have helped improve the presentation of this manuscript.

Funding

The work was supported by the National Natural Science Foundation of China (No. 61701340 and 61702361), the State Key Laboratory of Medicinal Chemical Biology in China, the Australian Research Council (ARC) (LP110200333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965) and a Major Inter-Disciplinary Research (IDR) project awarded by Monash University.

Conflict of Interest: none declared.

References

Barras,D. and Widmann,C. (2011) Promises of apoptosis-inducing peptides in cancer therapeutics. *Curr. Pharm. Biotechnol.,* **12**, 1153–1165.

Boohaker,R.J. *et al.* (2012) The use of therapeutic peptides to target and to kill cancer cells. *Curr. Med. Chem.,* **19**, 3794.

Chen,W. *et al.* (2016) iACP: a sequence-based tool for identifying anticancer peptides. *Oncotarget,* **7**, 16895.

Diana,G. *et al.* (2013) From antimicrobial to anticancer peptides. *A review.* Front. Microbiol., **4**, 294.

Ding,C. and Peng,H. (2003) Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.*, **3**, 185–205.

Dou,Y. *et al.* (2014) PhosphoSVM: prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine. *Amino Acids,* **46**, 1459–1469.

Dubchak,I. *et al.* (1999) Recognition of a protein fold in the context of the SCOP classification. *Prot. Struct. Funct. Bioinform.*, **35**, 401–407.

Ferlay,J. *et al.* (2010) Estimates of worldwide burden of cancer in 2008: GLOBOCAN 2008. *Int. J. Cancer,* **127**, 2893–2917.

Furey,T.S. *et al.* (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics,* **16**, 906.

Govindan,G. and Nair,A.S. (2011) Composition, Transition and Distribution (CTD)—a dynamic feature for predictions based on hierarchical structure of cellular sorting. In: *IEEE 2011 Annual IEEE India Conference*, IEEE, Hyderabad, India, pp. 1–6.

Hajisharifi,Z. *et al.* (2014) Predicting anticancer peptides with Chou's pseudo amino acid composition and investigating their mutagenicity via Ames test. *J. Theor. Biol.,* **341**, 34.

Holohan,C. *et al.* (2013) Cancer drug resistance: an evolving paradigm. *Nat. Rev. Cancer,* **13**, 714–726.

Hsieh,F.S. and Turnbull,B.W. (1996) Nonparametric and semiparametric estimation of the receiver operating characteristic curve. *Ann. Stat.,* **24**, 25–40.

Li,W. and Godzik,A. (2006) Cd-Hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics,* **22**, 1658.

Li,Z.R. *et al.* (2011) PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res.,* **39**, W385.

Mader,J.S. and Hoskin,D.W. (2006) Cationic antimicrobial peptides as novel cytotoxic agents for cancer treatment. *Expert Opin. Investig. Drugs,* **15**, 933.

Otvos,L. (2008) Peptide-based drug design: here and now. *Methods Mol. Biol.,* **494**, 1.

Peng,H. *et al.* (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intel.*, **27**, 1226.

Jemal,A. *et al.* (2010) Cancer statistics, 2013. *CA Cancer J. Clin.,* **60**, 277.

Tyagi,A. *et al.* (2013) In silico models for designing and discovering novel anticancer peptides. *Sci. Rep.,* **3**, 10–2984.

Tyagi,A. *et al.* (2015) CancerPPD: a database of anticancer peptides and proteins. *Nucleic Acids Res.,* **43**, D837.

Vijayakumar,S. and Ptv,L. (2015) ACPP: a web server for prediction and design of anti-cancer peptides. *Int. J. Pept. Res. Ther.,* **21**, 99–106.

Wei,L. *et al.* (2017a) SkipCPP-Pred: an improved and promising sequence-based predictor for predicting cell-penetrating peptides. *BMC Genomics,* **18**, 1–11.

Wei,L. *et al.* (2017b) Fast prediction of methylation sites using sequence-based feature selection technique. *IEEE/ACM Trans. Comput. Biol. Bioinform*, doi: 10.1109/TCBB.2017.2670558.

Wei,L. *et al.* (2017c) CPPred-RF: a sequence-based predictor for identifying cell-penetrating peptides and their uptake efficiency. *J. Proteome Res.,* **16**, 2044.

Whitney,A.W. (2006) A direct method of nonparametric measurement selection. *IEEE Trans. Computers*, **C-20**, 1100–1103.

Wu,Y. *et al.* (2013) Classification of knee joint vibration signals using bivariate feature distribution estimation and maximal posterior probability decision criterion. *Entropy,* **15**, 1375–1387.

Xing,P. *et al.* (2017) Identifying N6-methyladenosine sites using multi-interval nucleotide pair position specificity and support vector machine. *Sci. Rep.,* **7**, 46757.