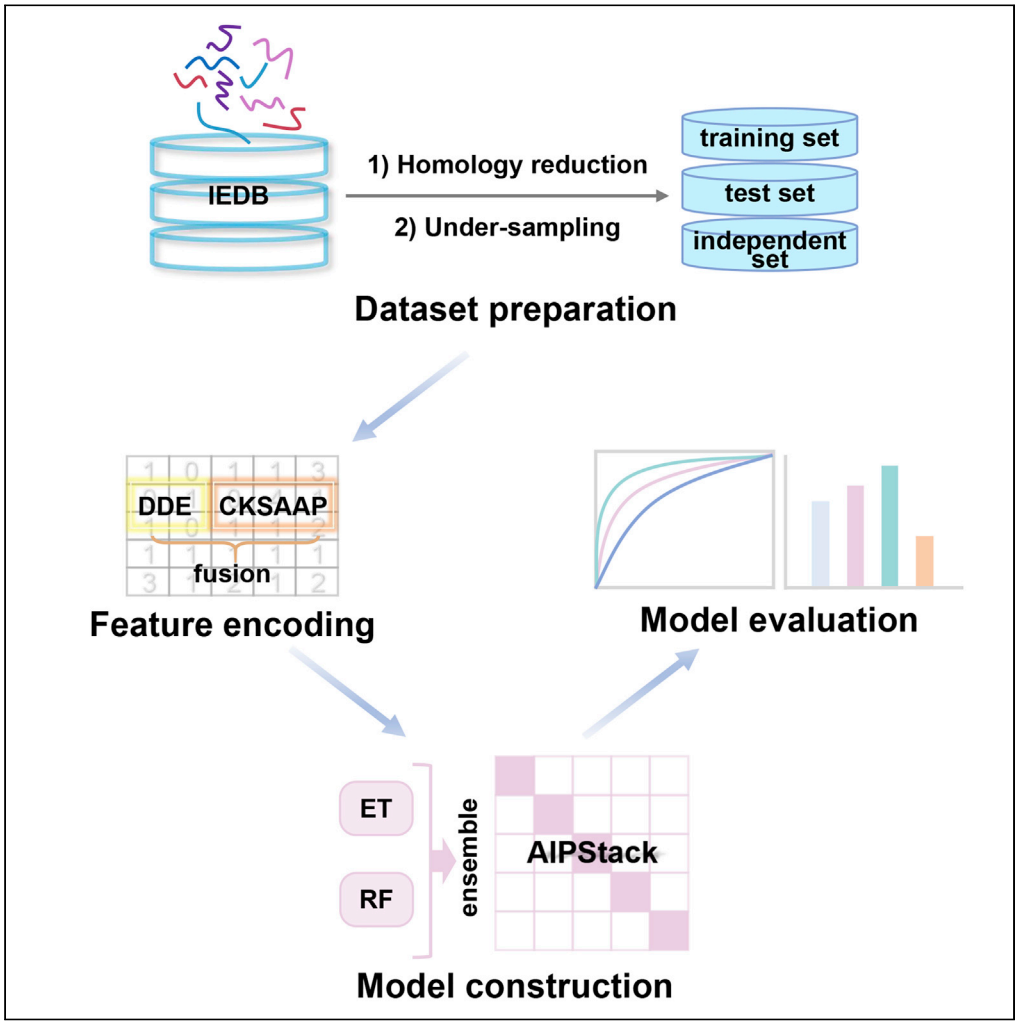
iScience

CellPress
OPEN ACCESS

Article

Prediction of anti-inflammatory peptides by a sequence-based stacking ensemble model named AIPStack



Hua Deng, Chaofeng Lou, Zengrui Wu, Weihua Li, Guixia Liu, Yun Tang

ytang234@ecust.edu.cn

Highlights

AIPStack model was developed for the prediction of anti-inflammatory peptides

The hybrid features were used to describe the peptide sequences

The proposed model AIPStack outperformed existing ones

SHAP was used to highlight the essential features required for AIP prediction

Deng et al., iScience 25, 104967
September 16, 2022
https://doi.org/10.1016/j.isci.2022.104967

iScience

CellPress
OPEN ACCESS

Article

Prediction of anti-inflammatory peptides by a sequence-based stacking ensemble model named AIPStack

Hua Deng,[1] Chaofeng Lou,[1] Zengrui Wu,[1] Weihua Li,[1] Guixia Liu,[1] and Yun Tang[1,2,*]

SUMMARY

Accurate and efficient identification of anti-inflammatory peptides (AIPs) is crucial for the treatment of inflammation. Here, we proposed a two-layer stacking ensemble model, AIPStack, to effectively predict AIPs. At first, we constructed a new dataset for model building and validation. Then, peptide sequences were represented by hybrid features, which were fused by two amino acid composition descriptors. Next, the stacking ensemble model was constructed by random forest and extremely randomized tree as the base-classifiers and logistic regression as the meta-classifier to receive the outputs from the base-classifiers. AIPStack achieved an AUC of 0.819, accuracy of 0.755, and MCC of 0.510 on the independent set 3, which were higher than other AIP predictors. Furthermore, the essential sequence features were highlighted by the Shapley Additive exPlanation (SHAP) method. It is anticipated that AIPStack could be used for AIP prediction in a high-throughput manner and facilitate the hypothesis-driven experimental design.

INTRODUCTION

Inflammation response is essentially the natural defense to injury, infection, or stimuli in the body, which helps to maintain tissue homeostasis under noxious conditions (Medzhitov, 2010). Usually, inflammation is classified into acute inflammation and chronic inflammation. Acute inflammation is a kind of innate immunity response, while chronic inflammation persists for a long time and results in various devastating chronic diseases, such as neurodegenerative diseases, cardiovascular diseases, cancer, and autoimmune disorders. According to statistics, three of five people die due to chronic inflammatory diseases in the world (Tsai et al., 2019; Deepak et al., 2019; Barcelos et al., 2019). There is no doubt that chronic inflammation has been a great threat to human health. Nonsteroidal anti-inflammatory drugs (NSAIDs), glucocorticoids, and some biologicals are the primary treatments for chronic inflammation and autoimmune disorders (Tabas and Glass, 2013; Vandewalle et al., 2018; Bindu et al., 2020; Chan and Carter, 2010). However, multiple adverse effects (Harirforoosh et al., 2013; Schäcke et al., 2002) and drug resistance (Dendoncker and Libert, 2017) pose challenges to the development of anti-inflammatory small molecular drugs. Thus, there is an urgent need for the discovery and rational design of novel effective anti-inflammatory drugs.

In recent years, peptide therapeutics, like antibacterial peptides and anticancer peptides, have brought a lot of attention due to their attractive advantages including safety, efficacy, high selectivity, and ease of synthesis (Muttenthaler et al., 2021). Anti-inflammatory peptide (AIP) is a type of therapeutic peptides that exhibit anti-inflammatory properties. Generally, AIPs are short linear peptides composed of 10–50 amino acids. Among AIPs discovered so far, most of them are endogenous peptides or derived from natural sources, such as endogenous neuropeptide vasoactive intestinal peptide (Jiang et al., 2016), melittin (Lee et al., 2014) from bee venom, and hydrostatin-SN1 (Zhang et al., 2020a) isolated from the sea snake. Some synthetic peptides were also explored to inhibit inflammatory responses, for example, the BCL-3-mimetic (Collins et al., 2015). The mechanisms of AIPs include modulation of immune cell differentiation, inducing anti-inflammatory responses, and prevention of excessive pro-inflammatory responses (Sun et al., 2018; Heinbockel et al., 2021). Recently, several AIPs have been approved by the U.S. Food and Drug Administration (FDA) to prevent and control inflammation (Usmani et al., 2017). These related studies show that AIPs have great therapeutic potentials and are likely to be a new alternative therapy for inflammation treatment.

[1]Shanghai Frontiers Science Center of Optogenetic Techniques for Cell Metabolism, School of Pharmacy, East China University of Science and Technology, Shanghai 200237, China

[2]Lead contact

*Correspondence: ytang234@ecust.edu.cn

https://doi.org/10.1016/j.isci.2022.104967

Table 1. List of currently available methods for AIP prediction

Methods	Year	Feature encoding	Model	Evaluation strategy	Web server	Standalone
AntiInflam	2017	TPC, motif features	SVM	10-fold CV, IT	http://metagenomics.iiserb.ac.in/antiinflam/	no
AIPpred	2018	DPC	RF	5-fold CV, IT	http://www.thegleelab.org/AIPpred/	no
PreAIP	2019	AAindex, KSAAP, structural features, pKSAAP	RF	10-fold CV, IT	http://kurata14.bio.kyutech.ac.jp/PreAIP/	no
PEPred-Suite	2019	89 class features	RF	10-fold CV, IT	http://server.malab.cn/PEPred-Suite	yes
AIEpred	2020	AAC, PSSM, PP	RF	LOOCV, IT	no	no
iAIPs	2021	AAC, DDE, GDC	RF	5-fold CV, IT	no	no
PreTP-EL	2021	Kmer, PPCT, Tng, DT, DR, PCG, PSDT, PSBT, DP	SVM, RF	10-fold CV, IT	http://bliulab.net/PreTP-EL	no
PreTP-Stack	2022	Kmer, PCG, Tng, DT, DR, AAC, BIT20, PPCT, PSDT, PSBT	SVM, RF, LDA, XGBoost, AMV	10-fold CV, IT	http://bliulab.net/PreTP-Stack	no

Feature abbreviations: TPC (tripeptide composition), DPC (dipeptide composition), AAindex (amino acid index), KSAAP (*k*-spaced amino acid pairs), pKSAAP (*k*-spaced amino acid pairs from position-specific scoring matrix), AAC (amino acid contact), PSSM (position-specific scoring matrix), PP (physicochemical property), DDE (dipeptide deviation from the expected mean), and GDC (*g*-gap dipeptide composition), PPCT (position-specific scoring matrix and position-specific frequency matrix cross transformation), Tng (top-*n*-gram), DT (distance-based top-*n*-gram), DR (distance-based residue), PCG (parallel correlation pseudo amino acid composition general), PSDT (PSSM distance transformation), PSBT (position-specific frequency matrix with distance bigram transformation), DP (distance amino acid pair or just distance-pair), BIT20 (twenty-bit feature).
Model abbreviations: SVM (support vector machine), RF (random forest), LDA (linear discriminant analysis), XGBoost (extreme gradient boosting), AMV (auto-weighted multi-view learning).
Evaluation strategy abbreviations: *k*-fold CV (*k*-fold cross-validation), IT (independent test), LOOCV (leave-one-out cross-validation).

The discovery of AIPs through wet-lab experiments is labor-intensive, expensive, and time-consuming, so it is difficult to be applied in a high-throughput manner. Moreover, with the rapid development and wide applications of next-generation sequencing techniques, there is an increasing demand for fast, cheaper, and efficient computational methods to annotate the enormous amount of protein sequences. Machine learning (ML) algorithms represent such kinds of computational methods that can efficiently predict the peptide properties based on the sequence profiles and are hoped to expedite the process of AIP discovery.

So far, several ML methods have been developed for the identification of potential AIPs. Tables 1 and 2 summarized the existing ML methods from a wide range of aspects, including the datasets, applied ML algorithms, feature encoding schemes, evaluation strategies, and the availability of web servers or standalone software. As shown in Table 1, Gupta2017 (Gupta et al., 2017) is the first dataset used for AIP prediction, whose positive and negative assaying epitopes were collected from the Immune Epitope Database (IEDB) (Vita et al., 2019). In 2018, Manavalan et al. (2018) recollected data from the IEDB with more peptide sequences than Gupta2017. The other six datasets were mostly derived from Manavalan2018. Relationships between datasets of the eight existing AIP prediction methods were presented in Figure S1. Early methods like AntiInflam (Gupta et al., 2017) and AIPpred (Manavalan et al., 2018) only utilized a single algorithm. Recently, two excellent methods, PreTP-EL (Guo et al., 2021) and PreTP-Stack (Yan et al., 2022), were constructed by integrating several ML algorithms for therapeutic peptide prediction. Random forest (RF) (Breiman, 2001) is the most popular algorithm; seven out of the eight methods employed it. Meantime, three out of the existing methods used support vector machine (SVM) (Grigoroiu et al., 2020).

How to effectively describe AIPs with informative feature representations is a major challenge for prediction models. The existing methods adopt various feature encoding schemes which can be classified into four groups, i.e. sequence composition features (e.g. dipeptide composition (DPC) (Saravanan and Gautham, 2015)), physicochemical property (e.g. amino acid index (AAindex) (Kawashima et al., 2007)), structure features (e.g. motif features), and evolution features (e.g. position-specific scoring matrix (PSSM) (Cai et al., 2012)). All the existing methods except AIPpred applied multiple feature encoding schemes to incorporate more information for the description of the sequences. Four evaluation strategies were

Table 2. A summary of the datasets used in currently available methods

		Benchmark sets		Independent sets			
Methods	Datasets	Number of AIPs	Number of non-AIPs	Number of AIPs	Number of non-AIPs	CD-HIT threshold	Datasets availability
AntiInflam	Gupta2017	690	1,009	173	253	–	yes
AIPpred	Manavalan2018	1,258	1,887	420	629	0.8	yes
PreAIP[a]	Khatun2019	1,258	1,887	420	629	0.8	yes
PEPred-Suite	Wei2019	1,258	1,887	420	629	0.8	yes
				420	2,000	–	
AIEpred	Zhang2020	690	1,009	173	253	–	yes
				420	629	0.8	
iAIPs[a]	Zhao2021	1,258	1,887	420	629	0.8	yes
PreTP-EL[a]	Guo2021	1,258	1,887	420	629	0.8	yes
PreTP-Stack[a]	Yan2022	1,258	1,887	420	629	0.8	yes

[a]Method whose dataset is the same as Manavalan2018.

used to estimate the performance of the existing methods, including leave-one-out cross-validation (CV), 5/10-fold CV, and independent test. Among the eight existing methods, six have been implemented as web servers or standalone software, but one of them, namely PEPred-Suite (Wei et al., 2019), cannot be accessed now.

These ML-based tools indeed have made great progress in the identification of AIPs and provided a rational basis for the selection of AIP candidates. However, two issues remain to be addressed. First, as presented in Table 2, most existing methods used the same and a small number of sequence samples, which limited the model performance and generalization capability. Second, three-quarters of existing methods only applied a single algorithm. However, lots of studies have proven that the ensemble learning model usually outperforms the single-algorithm-based model (Guo et al., 2021; Jiang et al., 2021; Basith et al., 2022; Liang et al., 2021; Mishra et al., 2019). Thus, the utilization of an ensemble learning strategy might improve the performance of AIP identification. Recently, two ensemble learning methods, namely PreTP-EL and PreTP-Stack, were reported, but their performance in AIP prediction might be still limited. Accordingly, it is essential to develop a new prediction model with higher accuracy, which could not only help improve our understanding of the association between peptide sequence and anti-inflammatory activity but also provide a reference for the rational design of AIPs based on the important features given by the model explanation.

Keeping these issues in mind, we first constructed a new dataset for model building and validation in this study. Then, we explored different feature encoding schemes and ML algorithms to further improve the prediction performance. We proposed a stacking ensemble model called AIPStack to predict AIPs. In brief, the AIPStack was composed of a two-layer framework. The first layer consisted of two popular ML algorithms (extremely randomized tree (ET) (Geurts et al., 2006) and RF), and used feature vectors fused by two feature representations, namely dipeptide deviation from expected mean (DDE) (Saravanan and Gautham, 2015) and composition of *k*-spaced amino acid pairs (CKSAAP) (Chen et al., 2013). And the second layer adopted the prediction probabilities from the first layer as the inputs of the meta-classifier (logistic regression, LR) (LaValley, 2008). The systematic workflow of AIPStack was depicted in Figure 1. We also evaluated the generalization capability of the proposed AIPStack and compared it with several state-of-the-art models by independent sets. Moreover, to implement the model interpretability, we leveraged the Shapley Additive exPlanation (SHAP) (Lundberg and Lee, 2017) method to highlight the most important and contributing sequence features. The proposed approach is potentially useful for AIP research.

RESULTS

Dataset preparation

Dataset preparation is the first step in an ML endeavor and is important to build a predictive model with strong generalization capability. In this study, at first, we obtained a total of 2,642 AIPs (positive samples)

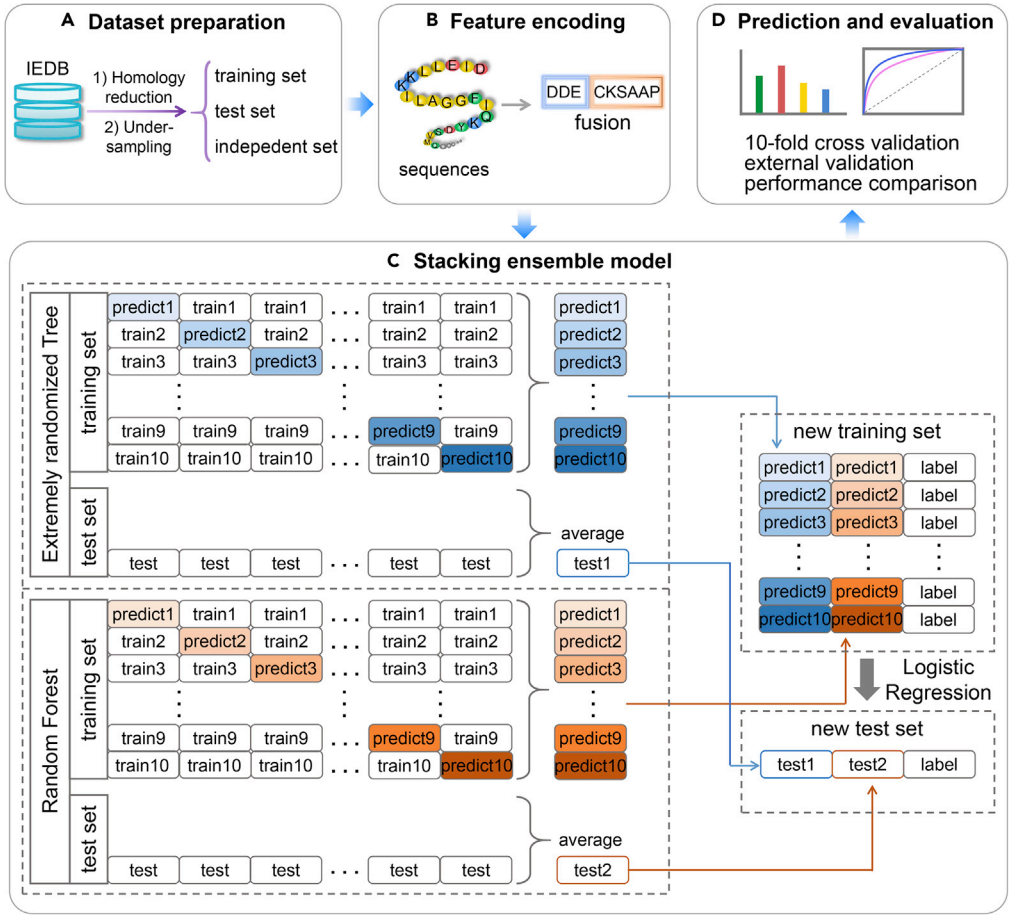


Figure 1. The overall framework of our AIPStack

(A) Dataset preparation. The dataset used here was collected from the IEDB. After reducing sequence redundancy, the undersampling approach was used to handle the imbalanced dataset.

(B) Feature encoding. The hybrid features fused by the DDE descriptor and CKSAAP descriptor were used to represent the peptide sequences.

(C) Model construction. A two-layer stacking ensemble model, called AIPStack, was developed.

(D) Model evaluation and prediction. We evaluated the AIPStack by the 10-fold cross-validation, internal and external validation. It was also compared with the existing methods.

and 3,704 non-AIPs (negative samples) from the IEDB. To avoid the evaluation bias caused by sequence homology, we excluded the sequences that shared >80% similarity with others, which resulted in a non-redundant dataset containing 1,866 AIPs and 2,845 non-AIPs. Since the dataset was imbalanced, we adopted the random undersampling technique to solve the issue, whereby a balanced dataset was obtained. The sampling process was repeated five times to generate five balanced datasets. Each balanced dataset contained 1,866 AIPs and the equal number of non-AIPs. Then according to a ratio of 8:1:1, each balanced dataset was randomly divided into training set, test set, and independent set (hereinafter referred to as independent set 1), which were used for model building, internal validation, and external validation, respectively.

Considering that there might be some overlaps between our independent set 1 of the final model and the datasets used for training in other methods, we constructed independent set 2 (difference of AntiInflam's benchmark dataset with independent set 1) and independent set 3 (difference of AIPpred's benchmark dataset with independent set 1) for unbiased comparison with other methods, which led to 135 AIPs and 138 non-AIPs in independent set 2, and 71 AIPs and 72 non-AIPs in independent set 3.

The length distribution of AIPs and non-AIPs can be seen in Figure 2A. The positive samples and negative ones had a similar distribution of sequence length, and most sequence lengths ranged from 15 to 20 amino

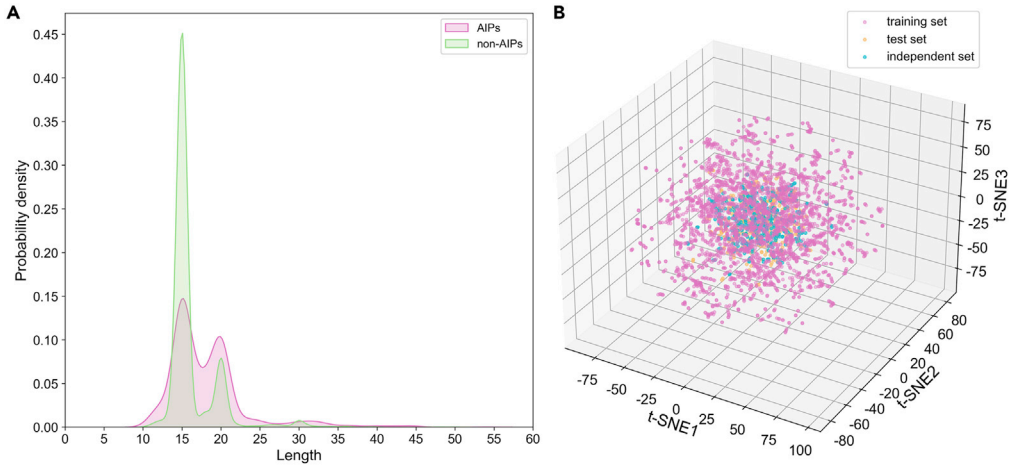


Figure 2. Sequence length and spatial distribution

(A) Sequence length distribution of AIPs and non-AIPs in the whole dataset.

(B) A *t*-SNE plot for sequence spatial distribution of one of the balanced datasets. See also Figure S2.

acids. Additionally, the *t*-distributed stochastic neighbor embedding (*t*-SNE) plot in Figures 2B and S2, illustrated that the created test sets and independent sets covered most sequence space occupied by the training set. The results implied the rationality of dataset partitioning.

Composition analysis of AIPs and non-AIPs

We performed composition information analysis for AIPs and non-AIPs. Each of the amino acid composition (AAC) (Bhasin and Raghava, 2004) descriptor and DPC descriptor was calculated on the whole dataset. Figure 3A showed the average composition of natural amino acids in AIP and non-AIP sequences. Amino acids with the four highest absolute difference scores were Leu, Asp, Arg, and Pro. The absolute difference values were 0.016, 0.009, 0.007, and 0.006, respectively (see Table S1). However, there was little difference in the composition of residue Trp and Met between AIPs and non-AIPs. Furthermore, a two-sided Mann-Whitney U test was used to evaluate the statistically significant difference between each amino acid for AIPs and non-AIPs. As shown in Figure 3A (see also Table S1), the composition of four amino acids (Leu, Asp, Arg, and Pro) was significantly different between AIPs and non-AIPs. Among them, the abundance of Leu and Arg (positively charged) was higher in AIPs than that in non-AIPs, whereas Pro and Asp (negatively charged) were more abundant in non-AIPs compared with AIPs. Similarly, Figure 3B and Table S2 showed the 20 top-ranked dipeptides which had the highest composition absolute differences between AIPs and non-AIPs. It was observed that dipeptides Leu-Leu, Ser-Leu, Leu-Ser, Leu-Glu, Leu-Lys, Leu-Ile, Ser-Val, Tyr-Leu, Glu-Arg, Arg-Ile, and Val-Leu were significantly dominant in AIPs, while dipeptides Gln-Gln, Asp-Asp, Gln-Pro, and Val-Asp were significantly dominant in non-AIPs. Namely, the most abundant dipeptides in AIPs were primarily composed of apolar-apolar, polar uncharged-apolar, apolar-polar uncharged, apolar-negatively charged, apolar-positively charged, negatively charged-positively charged, and positively charged-apolar amino acid pairs; the most abundant dipeptides in non-AIPs were primarily composed of polar uncharged-polar uncharged, negatively charged-negatively charged, polar uncharged-apolar, and apolar-negatively charged amino acid pairs.

These results suggested that the significant differences in the composition of amino acids and dipeptides between AIP and non-AIP sequences might be important factors for governing the activity of inducing the release of anti-inflammatory cytokines or not. Consequently, in this study, composition descriptors were mainly considered when performing feature extraction.

Conserved residues in the terminal regions of AIPs and non-AIPs

In the above composition analysis, it was observed that certain amino acids were abundant in AIPs. However, it was unclear whether the dominant amino acids were evenly distributed or preferred at a certain region in the sequence. To study the positional preference of amino acids, Two Sample Logo (TSL) (Crooks

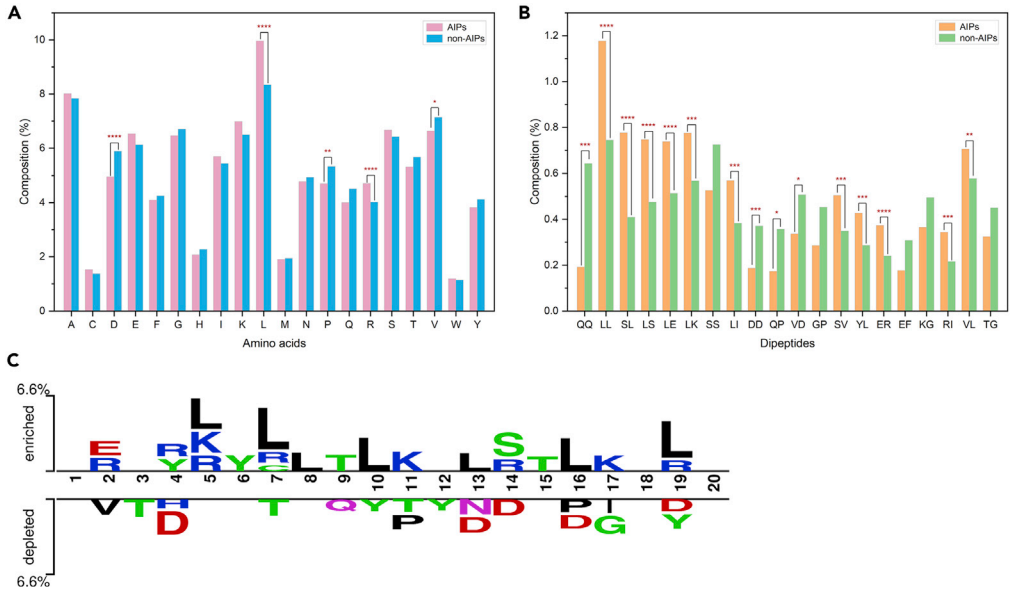


Figure 3. The different residue compositions of AIPs and non-AIPs

(A) Average amino acid compositions of 20 natural amino acids for AIPs and non-AIPs. See also Table S1.

(B) Dipeptide compositions of 20 top-ranked dipeptides that have the highest absolute differences between AIPs and non-AIPs. See also Table S2.

(C) Residue positional preference of AIPs and non-AIPs. The upper portion and the lower portion of the sequence logo graph represent the conserved residues of AIPs and non-AIPs, respectively. The first ten positions represent the N-terminus of peptides, and the last ten positions represent the C-terminus of peptides. p-values were calculated by the two-sided Mann-Whitney U test. The asterisks represent the statistical p-values (*p-value < 0.05; ** p-value < 0.01; *** p-value < 0.001; **** p-value < 0.0001).

et al., 2004) analysis was conducted for 10 residues from the N-terminus and C-terminus, separately, in the sequences of AIPs and non-AIPs.

As shown in Figure 3C, the first ten positions represent the N-terminus of peptides, and the last ten positions represent the C-terminus of peptides. Residue Leu was mostly preferred at positions 5, 7, 8, 10, 13, 16, and 19 of AIPs sequences. Moreover, other significantly preferred amino acids in the terminal regions of AIPs were listed as follows, Arg at 2, 4, 5, 7, 14, and 19; Lys at positions 5, 11, and 17; Glu at positions 2; and Thr at positions 9 and 15. Similarly, Asp, Pro, and Gly were dominant in the terminal regions of non-AIPs. The analysis showed that AIPs and non-AIPs had different preference for amino acids in the terminal regions.

Model construction of AIPStack

The whole process of model construction included two subsequent steps: first to build baseline models, then to build optimal models.

Baseline models

Totally eight ML algorithms and thirteen descriptors of peptide sequences were used in model building. To assess the capability of these algorithms and descriptors in distinguishing AIPs, for each balanced dataset, we built 104 (13 × 8) baseline models without hyperparameter tuning on the training set and evaluated them by the 10-fold CV and the test set. We displayed the average results of five balanced datasets in Figures 4A and 4B and Tables S3 and S4. As shown in Figure 4A and Table S3, decision tree-based models (i.e. ET, RF, light gradient boosting machine (LightGBM) (Friedman, 2001), and eXtreme gradient boosting (XGBoost) (Chen and Guestrin, 2016)) achieved better AUC (area under the receiver operating characteristic curve) values (Fawcett, 2006) than models based on other algorithms (i.e. *k*-nearest neighbor (KNN) (Weinberger and Saul, 2009), LR, naive Bayes (NB) (Rish, 2001), and SVM) on the training set in most cases. It was observed that a baseline model which was ET-based and developed by DDE descriptor attained the maximum performance with AUC = 0.789 on the training set. Followed by an RF-based baseline model, which also adopted the DDE descriptor and achieved the second-best performance with AUC = 0.784

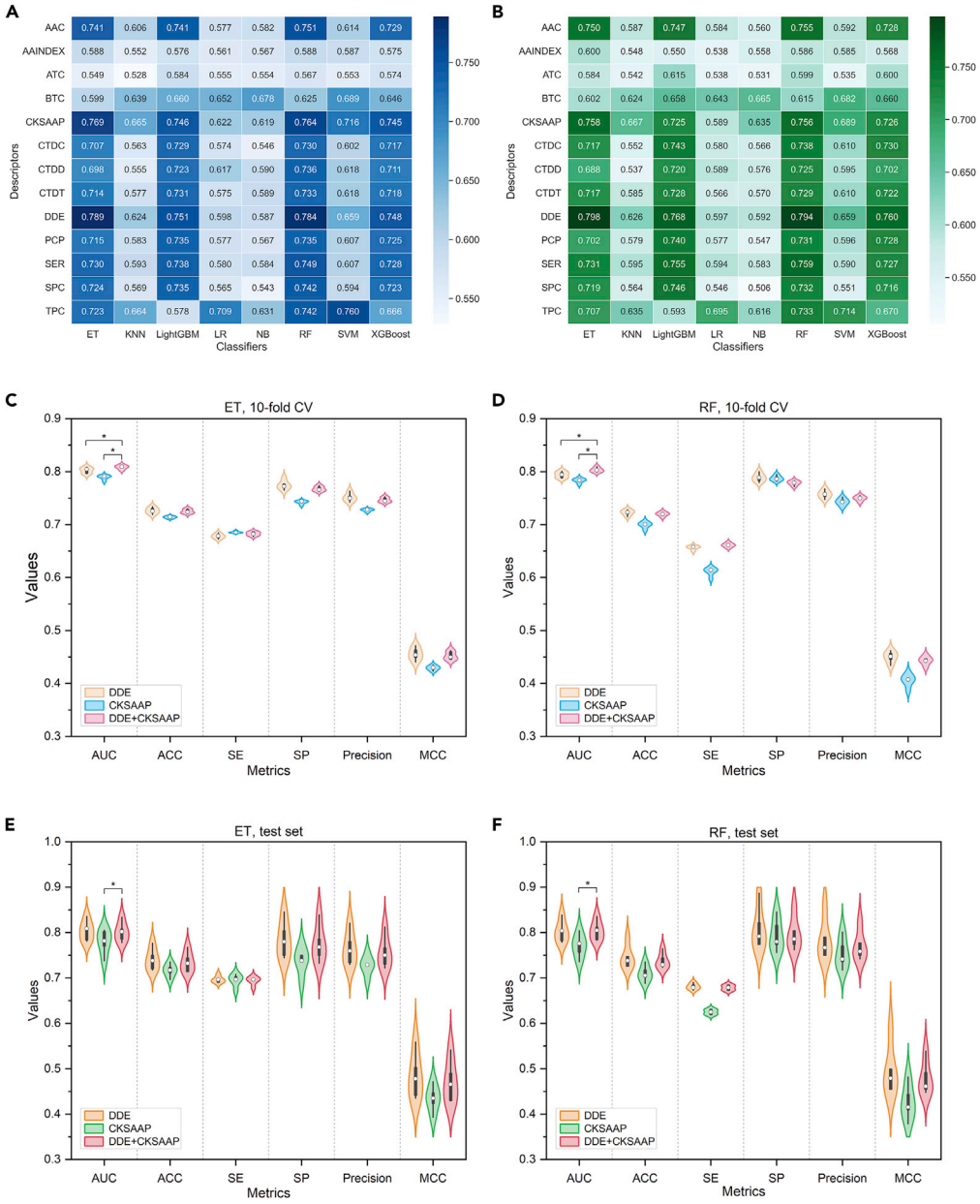


Figure 4. Performance evaluation of different ML algorithms and descriptors

(A) A heatmap showing the average AUCs of baseline models on the training set of five balanced datasets. See also Table S3.

(B) A heatmap showing the average AUCs of baseline models on the test set of five balanced datasets. See also Table S4.

Performance comparison of individual descriptors and the hybrid features. Results of ET-based models on (C) the training set and (E) the test set, respectively. Results of RF-based models on the training set (D) and (F) the test set, respectively. The "DDE + CKSAAP" denotes the hybrid features. The white dots in the violin plots represented the median values. p-values were calculated by the one-sided Wilcoxon signed-rank test and were also annotated. One asterisk represents p-value < 0.05 and the hybrid features performed better than the individual descriptor. See also Table S5 and Table S6.

on the training set. Interestingly, LightGBM and XGBoost in conjunction with the DDE descriptor also achieved higher performance compared with the combination with other descriptors. In addition, baseline models which combined the CKSAAP descriptor with ET or RF algorithm also obtained higher AUC values on the training set. Meanwhile, we observed a similar tendency in the results on the test set (see Figure 4B and Table S4). ET or RF-based baseline models in conjunction with DDE or CKSAPP descriptor also

performed better than other baseline models on the test set. The results above indicated that the DDE descriptor and CKSAAP descriptor were relatively informative for the identification of AIPs, and ET or RF-based models might be more suitable for AIP prediction. Consequently, DDE and CKSAAP were chosen for further study; ET and RF were used as base-classifiers in the stacking ensemble learning.

Optimal models

Feature fusion was employed to check whether the models built by hybrid features would achieve better performance. The two selected meta-classifiers (ET and RF) and three descriptors (CKSAAP, DDE, and the hybrid features) were combined to develop six prediction models. A one-sided Wilcoxon signed-rank test was also carried out to statistically compare the performance. On the training set, for both ET-based and RF-based models, the models with hybrid features significantly outperformed the ones with a single descriptor at a p-value threshold of 0.05 in terms of AUC values (see Figures 4C and 4D and Table S5). It suggested that feature fusion contributed to a more accurate classification model than relying on individual features. On the test set, for RF-based models, the hybrid features resulted in the best AUC of 0.804 and a lower SD of 0.022. And the models with hybrid features significantly outperformed the models with CKSAAP descriptor in terms of AUC values (p-value < 0.05) (see Figures 4E and 4F and Table S6). For ET-based models, the hybrid features led to a significant improvement of AUC values compared with the CKSAAP descriptor, but it was somewhat worse than the DDE descriptor. We inferred from the above results that the DDE descriptor might make the main contribution in the hybrid features to the model performance. Taking the results of the training set and the test set into consideration, the hybrid features were employed in our proposed method.

Evaluation of the AIPStack model

From the models constructed through five balanced datasets, we chose the one with the highest AUC value on the training set as the final predictive model, called AIPStack. To investigate the effectiveness of the AIPStack model, we carried out comparative experiments on the training set and test set. All the base-classifiers and meta-classifier were developed using the hybrid features and finely tuned. The corresponding results are provided in Figures 5A and 5B and Table S7. Using 0.05 as the p-value cutoff value, AIPStack significantly outperformed the base-classifier RF in terms of AUC on the training set. When compared with the base-classifier ET, though there was no significant difference in AUC, AIPStack achieved a slightly higher average AUC (0.808 versus 0.797) and lower SD(0.025 versus 0.027) in 10-fold CV. As for the meta-classifier, AIPStack significantly performed better than LR in all evaluation metrics on the training set (p-value of AUC <0.001). From the results on the test set (see Figure 5B), AIPStack and all three constituent classifiers did not suffer from overfitting. Besides, the AIPStack model was superior to its three constituent classifiers in all evaluation metrics.

Generalization capability of AIPStack

To examine the generalization capability of our model, independent set 1 was used to assess and validate the robustness of the AIPStack model. Our method achieved good performance with AUC = 0.797, accuracy (ACC) = 0.701, sensitivity (SE) = 0.658, specificity (SP) = 0.743, precision = 0.719, and Matthews correlation coefficient (MCC) = 0.403. As can be seen, AIPStack still performed well on the unseen data. It proved that our model had good transferability and was capable of distinguishing AIPs in practical prediction.

Comparison with existing methods

AntiInflam is the first model for the identification of AIPs (Gupta et al., 2017). It was based on the SVM algorithm and was trained on a relatively small dataset (690 AIPs and 1,009 non-AIPs). The independent set 2 was employed for the comparison of AIPStack with AntiInflam. The prediction results of AntiInflam were obtained from its web server by uploading the dataset. As presented in Figure 5C, AIPStack obviously outperformed AntiInflam. The AntiInflam performed not well in terms of SE (0.156 or 0.081) and MCC (0.081 or 0.181), whether using the less accurate or more accurate model. It tended to predict all samples as negative ones. The results indicated that AntiInflam might have a poor generalization capability.

Manavalan et al. constructed a benchmark set and an independent set that contain more AIPs (1,678) and non-AIPs (2,516), which was called Manavalan2018 here. They then proposed AIPpred. Later, PreAIP (Khatun et al., 2019) was developed and evaluated on the basis of Manavalan2018. It is the best prediction model among the existing methods which specifically predict AIPs. Most recently, Liu's group proposed

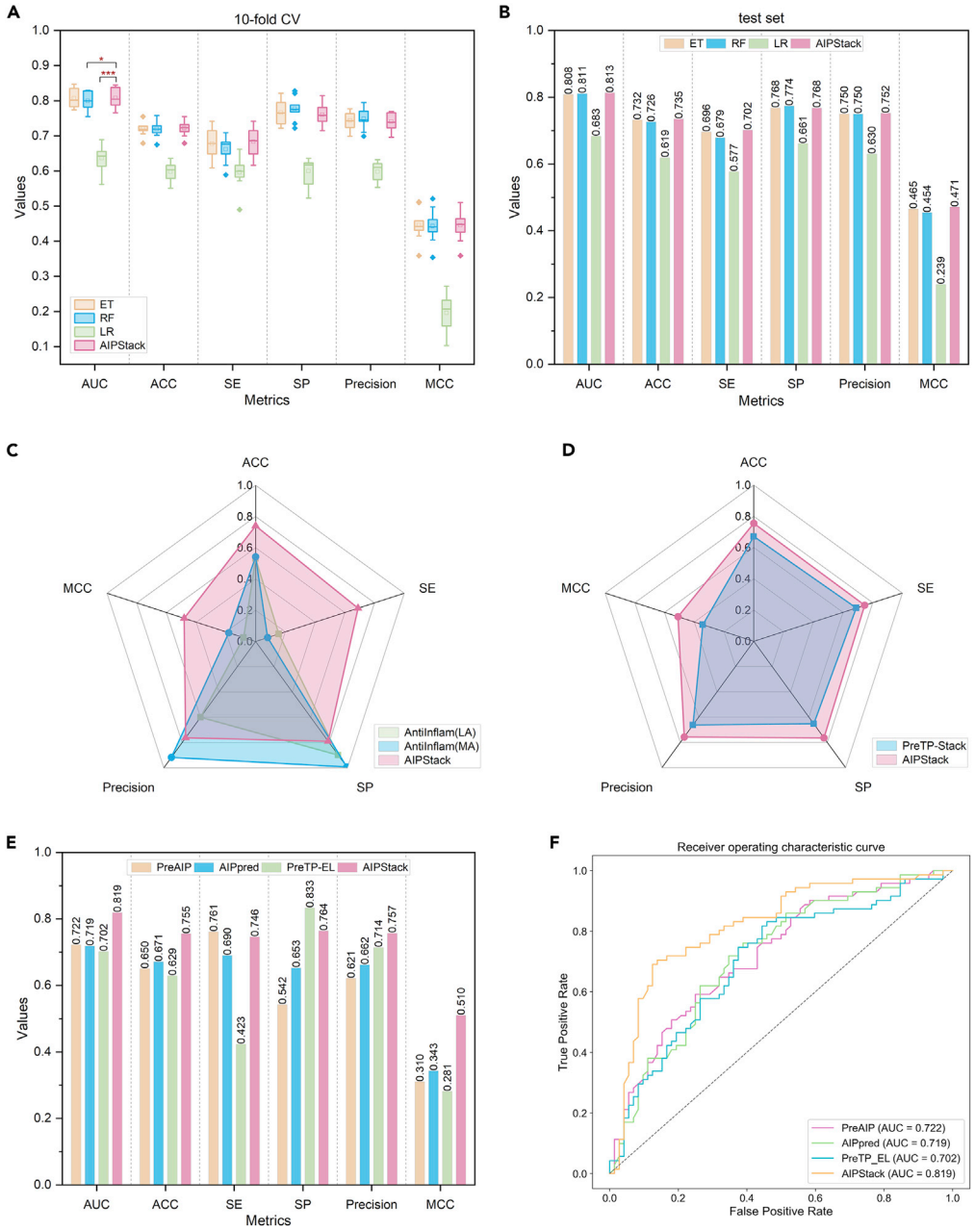


Figure 5. Performance comparison of AIPStack with its constituent classifiers and the existing methods

(A) Average performance of the AIPStack and its constituent classifiers on the training set using the 10-fold CV. ET and RF are base-classifiers, and LR is the meta-classifier. The lines in the boxes represent the median value, and the diamonds show outliers. p-values were calculated by the one-sided Wilcoxon signed-rank test and were also annotated. The asterisks represent the statistical p-value (* p-value < 0.05; *** p-value < 0.001). See also Table S7.

(B) Performance of the AIPStack and its constituent classifiers on the test set.

(C) Performance of the AntiInflam and AIPStack on the independent set 2. AntiInflam provided two models which used different feature encodings and showed different accuracy. "LA" and "MA" stood for less accurate and more accurate model, respectively.

(D) Performance of the PreTP-Stack and AIPStack on the independent set 3.

(E) Performance of the PreAIP, AIPpred, PreTP-EL, and AIPStack on the independent set 3.

(F) ROC curves and AUCs of the PreAIP, AIPpred, PreTP-EL, and AIPStack. In panels (C) and (D), AntiInflam and PreTP-Stack did not provide predicted probabilities on their web servers, hence the AUCs cannot be computed and were not shown.

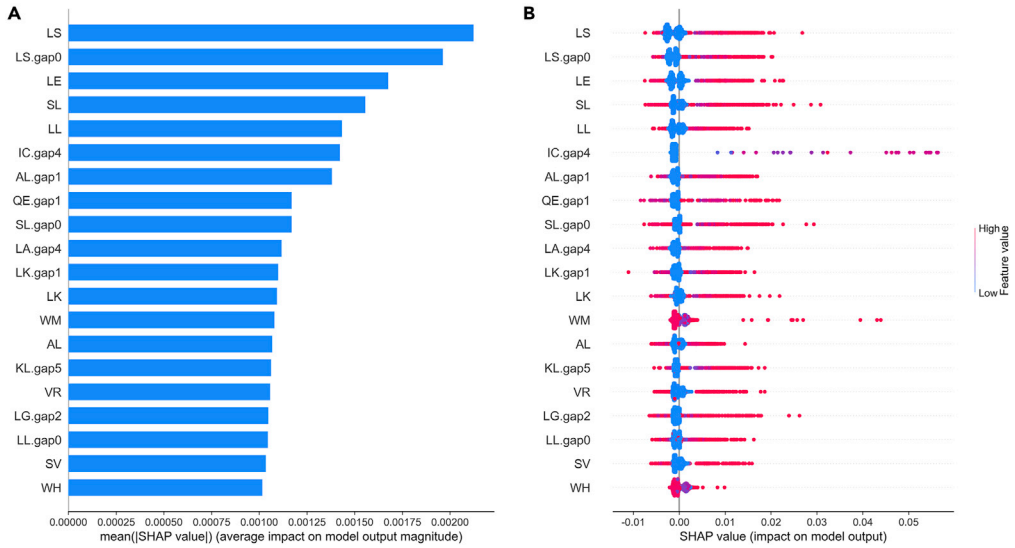


Figure 6. SHAP analysis results

(A) A standard bar plot showing the mean absolute value of SHAP values for the top 20 features.

(B) Distribution of SHAP values for the top 20 features. Feature values are indicated by different colors (red: high and blue: low). A positive SHAP value indicates it is AIP, while a negative SHAP value indicates it is non-AIP.

two ensemble learning models for identifying all types of therapeutic peptides, i.e. PreTP-EL and PreTP-Stack. These two methods also used Manavalan2018 for model construction and validation. The independent set 3 was employed for the comparison of AIPStack with AIPpred, PreAIP, PreTP-EL, and PreTP-Stack. Likewise, prediction results were obtained from their web servers. AIPStack outperformed PreTP-Stack in all five evaluation metrics (see Figure 5D). Meanwhile, as illustrated in Figures 5E and 5F, AIPStack achieved the highest AUC of 0.819, as well as the maximum ACC of 0.755, precision of 0.757, and MCC of 0.510, respectively. Although our AIPStack achieved a lower SE than PreAIP and a lower SP than PreTP-EL, it showed a more balanced performance in terms of SE and SP.

The AIPStack was not compared with the other three methods, i.e. PEPred-Suite, AIEpred (Zhang et al., 2020b), and iAIPs (Zhao et al., 2021). The reasons are as follows: i) the PEPred-Suite server is no longer functional; ii) AIEpred and iAIPs do not provide web servers or standalone software, and the models are unavailable, too. However, it can be inferred that AIPStack will perform better than these three models in the external validation. Because according to the literature, these three models underperformed AIPpred on the same independent set (Zhang et al., 2020b; Zhao et al., 2021), while our AIPStack achieved better performance than AIPpred in all six metrics.

Model interpretation

In this section, we employed the SHAP method to analyze the contributions of sequence features during the prediction. In current work, due to the good interpretability and relatively high performance, the ET model based on the hybrid features was analyzed, instead of the stacked model which is complicated to interpret. We visualized the importance of each feature in each sample of the training set, and ranked features according to their SHAP values.

Figure 6A provided the mean absolute values of the SHAP values for the top 20 features, it was obvious that the five most influential features for the prediction of samples were "LS", "LS.gap0", "LE", "SL", and "LL". Figure 6B illustrated the distribution of SHAP values for the 20 most influential features. From Figure 6B, we could figure out the relationships between SHAP values and the positive influence or negative influence of these features. Positive SHAP values indicated the prediction of AIPs with a high probability. On the contrary, negative ones indicated the prediction of non-AIPs with a high probability. Taking the feature "LS" in Figure 6B as an example, feature values of "LS" were relatively low for most negative samples. Hence, for an unknown sample, if its feature value of "LS" is high, then the model will tend to predict it as an AIP; otherwise, the model will tend to predict it as a non-AIP.

DISCUSSION

The accurate identification of potential AIPs via computational methods remains one of the most challenging problems. In this study, we presented a new method, called AIPStack, which allowed us to predict whether a given peptide could induce any anti-inflammatory cytokine or not, based on the sequence features.

First, we constructed a non-redundant dataset whose size was increased by approximately 12.6% compared with the dataset used in state-of-the-art methods (e.g. AIPpred, PreAIP, and PreTP-EL). Analysis on the composition information and positional preference suggested that residues Leu and Arg were significantly abundant in the terminal regions of AIPs but not the case in non-AIPs, which was consistent with the results of previous studies (Manavalan et al., 2018; Khatun et al., 2019). Also, there is some experimental evidence to support our findings. For example, it was reported that Leu had a major influence on the anti-inflammatory activity of peptides (Wang et al., 2010; Nan et al., 2007). Another example is that a lupin protein hydrolysat (LPH) peptide (GPETAFLR) derived from plants exerts anti-inflammatory activity by promoting the expression of the anti-inflammatory cytokine IL-10 and reducing the expression of pro-inflammatory cytokine TNF and IL-1β (Montserrat-de la Paz et al., 2019). In the sequence of this LPH peptide, residues Leu and Arg are at the C-terminus. Thus, introducing a Leu or Arg mutation to the terminal regions of peptides may improve the anti-inflammatory efficacy. For other dipeptides with a significant difference, further wet-lab experiments are needed to prove their effects on anti-inflammatory activity. Consequently, these observations of residue composition might shed light on the redesign and the *de novo* design of AIPs.

We explored various algorithms and encoding schemes for AIP identification, while six of eight existing methods only used RF to construct their models. Likewise, we found RF algorithm in conjunction with DDE and CKSAAP achieved good performance. However, ET algorithm has not been employed in any existing methods; when it was combined with the DDE descriptor, the model achieved top performance among all baseline models in this work. Next, we demonstrated the effectiveness of feature fusion by evaluating the performance of ET-based and RF-based models. The concatenation of feature vectors led to higher dimensional vectors, someone may argue that the hybrid descriptor may contain redundant or noisy features that potentially lead to the decreased predictive performance of the trained model. But because there were only two types of descriptors used in the fusion, feature selection is likely to cause an overfitting problem. Therefore, we did not perform feature selection in this work.

The final AIPStack model achieved an average AUC of 0.808 on the training set, representing an improvement of AUC of 1.4%–26.9% compared with the three constituent models. It also outperformed the constituent models in terms of all six evaluation metrics on the test set. Overall, our observations demonstrated the effectiveness of the stacking ensemble strategy. Ideally, it is desirable to apply classifiers that have different underlying operating principles as the base-classifiers, to enrich the meta-classifier with more information on the solution space. In our study, we found that tree-based models generally performed better, so we just chose the two best tree-based models as the base-classifiers. This might be the reason why there was only a slight improvement of the AIPStack and no statistical difference when compared with the base-classifier ET.

Moreover, we constructed three independent sets to assess the generalization capability of our method and objectively compared it with eight state-of-the-art methods. We found that the AIPStack performed well on all three independent sets, which demonstrated the stability and reliability of our method. Furthermore, AIPStack outperformed the existing methods. First, our AIPStack achieved much better performance compared with AntiInflam. The latter was built on a much smaller dataset, therefore showed worse performance of AIP prediction on the independent set 2 constructed by us. Second, on the independent set 3, AIPStack outperformed AIPpred and PreTP-Stack on all evaluation metrics, and it achieved a more balanced performance compared with PreAIP and PreTP-EL. These results indicated that AIPStack had great capacity and utility.

To further investigate the relationships between sequence features and anti-inflammatory properties of peptides, we applied the SHAP algorithm for model interpretation. It revealed some essential features for AIP optimization, such as "LS", "LS.gap0", "LE", "SL", and "LL". The "LS.gap0" belongs to the CKSAAP descriptor and the others come from the DDE descriptor. According to the definition of these

two descriptors, "LS.gap0" represents the composition of dipeptide Leu-Ser in a sequence, while the rest four features reflect the deviation of the corresponding dipeptide frequencies from expected mean values. Consistently, in the composition analysis, we also found dipeptides Leu-Ser, Leu-Glu, Ser-Leu, and Leu-Leu were significantly different between AIPs and non-AIPs. Peptides with higher values of the above five features are more likely to be AIPs. Though there is no direct evidence to prove the importance of features identified in this work, some studies proved the dipeptides we mentioned here are indispensable for AIPs. For example, Lin et al. found that tripeptide LSW showed anti-inflammatory activity on vascular smooth muscle cells (Qinlu et al., 2017), from which we can infer the important role of dipeptide Leu-Ser in the anti-inflammatory activity.

Taken together, AIPStack is a promising method for distinguishing AIPs; it should be helpful for large-scale AIP screening and facilitating hypothesis-driven experimental design.

Limitations of the study

One limitation of the current study is that we did not provide a user-friendly web interface; alternatively, we shared our final model on the well-known GitHub. It is convenient for researchers to download and use.

Another limitation is that all the existing predictors including AIPStack are ML-based approaches. ML-based predictors typically need sufficient data for training. However, the number of currently available AIPs still cannot meet the needs and limited the performance of existing predictors. In addition, ML-based predictors need to extract sequence information by third-party software in advance. Though we had explored several sequence representation schemes in the current study, the existing schemes might be not informative enough. To solve the issue, on the one hand, developing new and informative sequence representation schemes even AIP-specific feature representation schemes will be helpful. On the other hand, using more elaborate deep learning algorithms and thus extracting features through network layers automatically could avoid the problem.

The third limitation is that we did not consider the AIPStack's capability in distinguishing AIPs from other types of peptides, since the dataset did not include peptides with other functions as negative samples. As suggested by Manavalan et al., in future work, a two-step framework can be employed to overcome the limitation (Manavalan et al., 2021). In the first step, the predictor will be developed using the dataset containing experimentally validated AIPs as positive samples, and other therapeutic peptides and random peptides as negative samples. In the second step, the predictor will be constructed using a dataset containing the same AIPs as positive samples and experimentally validated non-AIPs as negative samples. Model robustness and practical applicability will be enhanced through such a framework.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Dataset preparation
  - Position conservation analysis
  - Selection of feature encoding schemes
  - Feature fusion
  - Selection of ML algorithms
  - Framework of AIPStack
  - Model evaluation
  - Model interpretation
- QUANTIFICATION AND STATISTICAL ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2022.104967.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (Grant 2019YFA0904800), the National Natural Science Foundation of China (Grants 81872800, 82173746, and 82104066), and Shanghai Frontiers Science Center of Optogenetic Techniques for Cell Metabolism (Shanghai Municipal Education Commission, Grant 2021 Sci & Tech 03-28).

AUTHOR CONTRIBUTIONS

Conceptualization, H.D. and Y.T.; Formal Analysis, H.D.; Investigation, H.D.; Visualization, H.D. and C.L.; Writing - Original Draft, H.D.; Writing - Review & Editing, Y.T., G.L., W.L., and Z.W.; Supervision, Y.T.; Funding Acquisition, Y.T.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: June 21, 2022
Revised: August 9, 2022
Accepted: August 12, 2022
Published: September 16, 2022

REFERENCES

Banchereau, J., Pascual, V., and O'garra, A. (2012). From IL-2 to IL-37: the expanding spectrum of anti-inflammatory cytokines. Nat. Immunol. *13*, 925–931.

Barcelos, I.P.d., Troxell, R.M., and Graves, J.S. (2019). Mitochondrial dysfunction and multiple sclerosis. Biology *8*, 37–53.

Basith, S., Lee, G., and Manavalan, B. (2022). STALLION: a stacking-based ensemble learning framework for prokaryotic lysine acetylation site prediction. Brief. Bioinform. *23*, bbab376.

Bhasin, M., and Raghava, G.P.S. (2004). Classification of nuclear receptors based on amino acid composition and dipeptide composition. J. Biol. Chem. *279*, 23262–23266.

Bindu, S., Mazumder, S., and Bandyopadhyay, U. (2020). Non-steroidal anti-inflammatory drugs (NSAIDs) and organ damage: a current perspective. Biochem. Pharmacol. *180*, 114147–114167.

Boughorbel, S., Jarray, F., and El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PLoS One *12*, e0177678.

Breiman, L. (2001). Random forests. Mach. Learn. *45*, 5–32.

Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X., and Chen, Y.Z. (2003). SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res. *31*, 3692–3697.

Cai, Y., Huang, T., Hu, L., Shi, X., Xie, L., and Li, Y. (2012). Prediction of lysine ubiquitination with mRMR feature selection and analysis. Amino Acids *42*, 1387–1395.

Chan, A.C., and Carter, P.J. (2010). Therapeutic antibodies for autoimmunity and inflammation. Nat. Rev. Immunol. *10*, 301–316.

Charoenkwan, P., Chiangjong, W., Nantasenamat, C., Hasan, M.M., Manavalan, B., and Shoombuatong, W. (2021). StackIL6: a stacking ensemble model for improving the prediction of IL-6 inducing peptides. Brief. Bioinform. *22*, bbab172.

Chen, T., and Guestrin, C. (2016). XGBoost: a scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Association for Computing Machinery).

Chen, X., Qiu, J.-D., Shi, S.-P., Suo, S.-B., Huang, S.-Y., and Liang, R.-P. (2013). Incorporating key position and amino acid residue features to identify general and species-specific Ubiquitin conjugation sites. Bioinformatics *29*, 1614–1622.

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T.T., Wang, Y., Webb, G.I., Smith, A.I., Daly, R.J., Chou, K.-C., and Song, J. (2018). iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics *34*, 2499–2502.

Collins, P.E., Grassia, G., Colleran, A., Kiely, P.A., Ialenti, A., Maffia, P., and Carmody, R.J. (2015). Mapping the interaction of B cell leukemia 3 (BCL-3) and nuclear factor κB (NF-κB) p50 identifies a BCL-3-mimetic anti-inflammatory peptide. J. Biol. Chem. *290*, 15687–15696.

Collison, L.W., Workman, C.J., Kuo, T.T., Boyd, K., Wang, Y., Vignali, K.M., Cross, R., Sehy, D., Blumberg, R.S., and Vignali, D.A.A. (2007). The inhibitory cytokine IL-35 contributes to regulatory T-cell function. Nature *450*, 566–569.

Crooks, G.E., Hon, G., Chandonia, J.-M., and Brenner, S.E. (2004). WebLogo: a sequence logo generator. Genome Res. *14*, 1188–1190.

Deepak, P., Axelrad, J.E., and Ananthakrishnan, A.N. (2019). The role of the radiologist in determining disease severity in inflammatory bowel diseases. Gastrointest. Endosc. Clin. N. Am. *29*, 447–470.

Dendoncker, K., and Libert, C. (2017). Glucocorticoid resistance as a major drive in sepsis pathology. Cytokine Growth Factor Rev. *35*, 85–96.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognit. Lett. *27*, 861–874.

Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. Ann. Statist. *29*, 1189–1232.

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics *28*, 3150–3152.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. Mach. Learn. *63*, 3–42.

Grigoroiu, A., Yoon, J., and Bohndiek, S.E. (2020). Deep learning applied to hyperspectral endoscopy for online spectral classification. Sci. Rep. *10*, 3947.

Guo, Y., Yan, K., Lv, H., and Liu, B. (2021). PreTP-EL: prediction of therapeutic peptides based on ensemble learning. Brief. Bioinform. *22*, bbab358.

Gupta, S., Sharma, A.K., Shastri, V., Madhu, M.K., and Sharma, V.K. (2017). Prediction of anti-inflammatory proteins/peptides: an in silico approach. J. Transl. Med. *15*, 7–11.

Harirforoosh, S., Asghar, W., and Jamali, F. (2013). Adverse effects of nonsteroidal antiinflammatory drugs: an update of gastrointestinal, cardiovascular and renal complications. J. Pharm. Pharm. Sci. *16*, 821–847.

Heinbockel, L., Weindl, G., Correa, W., Brandenburg, J., Reiling, N., Wiesmüller, K.H., Schürholz, T., Gutsmann, T., Martinez de Tejada, G., Mauss, K., and Brandenburg, K. (2021).

Anti-infective and anti-inflammatory mode of action of peptide 19-2.5. Int. J. Mol. Sci. *22*, 1465.

Jiang, M., Zhao, B., Luo, S., Wang, Q., Chu, Y., Chen, T., Mao, X., Liu, Y., Wang, Y., Jiang, X., et al. (2021). NeuroPpred-Fuse: an interpretable stacking model for prediction of neuropeptides by fusing sequence information and feature selection methods. Brief. Bioinform. *22*, bbab310.

Jiang, W., Wang, H., Li, Y.S., and Luo, W. (2016). Role of vasoactive intestinal peptide in osteoarthritis. J. Biomed. Sci. *23*, 63.

Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M. (2008). AAindex: amino acid index database, progress report 2008. Nucleic Acids Res. *36*, D202–D205.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree (Curran Associates, Inc).

Khatun, M.S., Hasan, M.M., and Kurata, H. (2019). PreAIP: computational prediction of anti-inflammatory peptides by integrating multiple complementary features. Front. Genet. *10*, 129–139.

LaValley, M.P. (2008). Logistic regression. Circulation *117*, 2395–2399.

Lee, W.R., Kim, K.H., An, H.J., Kim, J.Y., Chang, Y.C., Chung, H., Park, Y.Y., Lee, M.L., and Park, K.K. (2014). The protective effects of melittin on Propionibacterium acnes-induced inflammatory responses in vitro and in vivo. J. Invest. Dermatol. *134*, 1922–1930.

Liang, X., Li, F., Chen, J., Li, J., Wu, H., Li, S., Song, J., and Liu, Q. (2021). Large-scale comparative review and assessment of computational methods for anti-cancer peptide identification. Brief. Bioinform. *22*, bbaa312.

Lundberg, S.M., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. *30*, 4765–4774.

Manavalan, B., Basith, S., and Lee, G. (2022). Comparative analysis of machine learning-based approaches for identifying therapeutic peptides targeting SARS-CoV-2. Brief. Bioinform. *23*, bbab412.

Manavalan, B., Shin, T.H., Kim, M.O., and Lee, G. (2018). AIPpred: sequence-based prediction of anti-inflammatory peptides using random forest. Front. Pharmacol. *9*, 276–287.

Marie, C., Pitton, C., Fitting, C., and Cavaillon, J.M. (1996). Regulation by anti-inflammatory cytokines (IL-4, IL-10, IL-13, TGFβ) of interleukin-8 production by LPS-and/or TNFα-activated human polymorphonuclear cells. Mediators Inflamm. *5*, 334–340.

Medzhitov, R. (2010). Inflammation 2010: new adventures of an old flame. Cell *140*, 771–776.

Mishra, A., Pokhrel, P., and Hoque, M.T. (2019). StackDPPred: a stacking based prediction of DNA-binding protein from sequence. Bioinformatics *35*, 433–441.

Montserrat-de la Paz, S., Lemus-Conejo, A., Toscano, R., Pedroche, J., Millan, F., and Millan-Linares, M.C. (2019). GPETAFLR, an octapeptide isolated from Lupinus angustifolius L. protein hydrolysate, promotes the skewing to the M2 phenotype in human primary monocytes. Food Funct. *10*, 3303–3311.

Muttenthaler, M., King, G.F., Adams, D.J., and Alewood, P.F. (2021). Trends in peptide drug discovery. Nat. Rev. Drug Discov. *20*, 309–325.

Nan, Y.H., Park, K.H., Jeon, Y.J., Park, Y., Park, I.S., Hahm, K.S., and Shin, S.Y. (2007). Antimicrobial and anti-inflammatory activities of a Leu/Lys-rich antimicrobial peptide with Phe-peptoid residues. Protein Pept. Lett. *14*, 1003–1007.

Pande, A., Patiyal, S., Lathwal, A., Arora, C., Kaur, D., Dhall, A., Mishra, G., Kaur, H., Sharma, N., Jain, S., et al. (2019). Computing wide range of protein/peptide features from their sequence and structure. Preprint at BioRxiv. https://doi.org/10.1101/599126.

Paul, W.E. (2015). History of interleukin-4. Cytokine *75*, 3–7.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: machine learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

Lin, Q., Liao, W., Bai, J., Wu, W., and Wu, J. (2017). Soy protein-derived ACE-inhibitory peptide LSW (Leu-Ser-Trp) shows anti-inflammatory activity on vascular smooth muscle cells. J. Funct.Foods *34*, 248–253.

Raschka, S. (2018). MLxtend: providing machine learning and data science utilities and extensions to Python's scientific computing stack. J. Open Source Softw. *3*, 638.

Rish, I. (2001). An Empirical Study of the Naive Bayes Classifier. In IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence (August 2001 Seattle), pp. 41–46.

Saravanan, V., and Gautham, N. (2015). Harnessing computational biology for exact linear B-cell epitope prediction: a novel amino acid composition-based feature descriptor. OMICS *19*, 648–658.

Schäcke, H., Döcke, W.D., and Asadullah, K. (2002). Mechanisms involved in the side effects of glucocorticoids. Pharmacol. Ther. *96*, 23–43.

Sun, G.Y., Yang, H.H., Guan, X.X., Zhong, W.J., Liu, Y.P., Du, M.Y., Luo, X.Q., Zhou, Y., and Guan, C.X. (2018). Vasoactive intestinal peptide overexpression mediated by lentivirus attenuates lipopolysaccharide-induced acute lung injury in mice by inhibiting inflammation. Mol. Immunol. *97*, 8–15.

Tabas, I., and Glass, C.K. (2013). Anti-inflammatory therapy in chronic disease: challenges and opportunities. Science *339*, 166–172.

Tsai, D.-H., Riediker, M., Berchet, A., Paccaud, F., Waeber, G., Vollenweider, P., and Bochud, M. (2019). Effects of short-and long-term exposures to particulate matter on inflammatory marker levels in the general population. Environ. Sci. Pollut. Res. Int. *26*, 19697–19704.

Usmani, S.S., Bedi, G., Samuel, J.S., Singh, S., Kalra, S., Kumar, P., Ahuja, A.A., Sharma, M., Gautam, A., and Raghava, G.P.S. (2017). THPdb: database of FDA-approved peptide and protein therapeutics. PLoS One *12*, e0181748.

Vandewalle, J., Luypaert, A., De Bosscher, K., and Libert, C. (2018). Therapeutic mechanisms of glucocorticoids. Trends Endocrinol. Metab. *29*, 42–54.

Verma, A., and Mehta, S. (2017). A comparative study of ensemble learning methods for classification in bioinformatics. In 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence (IEEE), pp. 155–158.

Vita, R., Mahajan, S., Overton, J.A., Dhanda, S.K., Martini, S., Cantrell, J.R., Wheeler, D.K., Sette, A., and Peters, B. (2019). The immune epitope database (IEDB): 2018 update. Nucleic Acids Res. *47*, D339–D343.

Wang, P., Nan, Y.H., Yang, S.T., Kang, S.W., Kim, Y., Park, I.S., Hahm, K.S., and Shin, S.Y. (2010). Cell selectivity and anti-inflammatory activity of a Leu/Lys-rich alpha-helical model antimicrobial peptide and its diastereomeric peptides. Peptides *31*, 1251–1261.

Wei, L., Zhou, C., Su, R., and Zou, Q. (2019). PEPred-Suite: improved and robust prediction of therapeutic peptides using adaptive feature representation learning. Bioinformatics *35*, 4272–4280.

Weinberger, K.Q., and Saul, L.K. (2009). Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. *10*, 207–244.

Yan, K., Lv, H., Wen, J., Guo, Y., Xu, Y., and Liu, B. (2022). PreTP-Stack: prediction of therapeutic peptide based on the stacked ensemble learning. IEEE/ACM Trans. Comput. Biol. Bioinform. *14*, 1–10. https://doi.org/10.1109/tcbb.2022.3183018.

Yoshida, H., and Hunter, C.A. (2015). The immunobiology of interleukin-27. Annu. Rev. Immunol. *33*, 417–443.

Zhang, C., Guo, S., Wang, J., Li, A., Sun, K., Qiu, L., Li, J., Wang, S., Ma, X., and Lu, Y. (2020a). Anti-inflammatory activity and mechanism of hydrostatin-SN1 from hydrophis cyanocinctus in interleukin-10 knockout mice. Front. Pharmacol. *11*, 930.

Zhang, J., Zhang, Z., Pu, L., Tang, J., and Guo, F. (2021). AIEpred: an ensemble predictive model of classifier chain to identify anti-inflammatory peptides. IEEE/ACM Trans. Comput. Biol. Bioinform. *18*, 1831–1840.

Zhao, D., Teng, Z., Li, Y., and Chen, D. (2021). iAIPs: identifying anti-inflammatory peptides using random forest. Front. Genet. *12*, 773202–773210.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Dataset of AntiInflam	(Gupta et al., 2017)	http://metagenomics.iiserb.ac.in/antiinflam/data.php
Dataset of AIPpred	(Manavalan et al., 2018)	http://www.thegleelab.org/AIPpred/AIPpredData.html
Software and algorithms		
Two Sample Logo	(Crooks et al., 2004)	http://www.twosamplelogo.org/
Python version 3.7.10	Python Software Foundation	https://www.python.org
iFeature	(Chen et al., 2018)	http://iFeature.erc.monash.edu/
Pfeature	(Pande et al., 2019)	https://webs.iiitd.edu.in/raghava/pfeature/
scikit-learn version 1.0.2	(Pedregosa et al., 2011)	https://scikit-learn.org/stable/index.html
XGBoost version 1.4.2	(Chen and Guestrin, 2016)	https://pypi.org/project/xgboost/1.4.2/
LightGBM version 3.2.1	(Ke et al., 2017)	https://pypi.org/project/lightgbm/3.2.1/
mlxtend version 0.19.0	(Raschka, 2018)	https://pypi.org/project/mlxtend/
shap version 0.40.0	(Lundberg and Lee, 2017)	https://pypi.org/project/shap/
PowerPoint version 18.2110.13110.0	Microsoft Corporation	https://www.microsoft.com/zh-cn/microsoft-365/microsoft-office?rtc=1

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Dr. Yun Tang (ytang234@ecust.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The datasets of AIPStack are made available on Github: https://github.com/Nicole-DH/AIPStack. We also shared the main codes of the AIPStack at that link. Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Dataset preparation

In this work, we constructed a new dataset for AIP prediction. Firstly, we collected linear peptides from the IEDB which captured large amounts of data from the literature, making it a reliable and popular database. The following criteria were used to separate positive samples from negative ones. If a linear peptide could induce any one of the anti-inflammatory cytokines including interleukin (IL)-4, IL-10, IL-13, IL-22, IL-27, IL-35, IL-37, transforming growth factor β (TGF-β), and interferon (IFN)-α/β, it was considered to be an AIP; if it cannot induce any anti-inflammatory cytokines above, it was regarded as a non-AIP (Marie et al., 1996; Paul, 2015; Yoshida and Hunter, 2015; Collison et al., 2007; Banchereau et al., 2012). Subsequently, to reduce homology bias and prevent overfitting, a CD-HIT (Fu et al., 2012) threshold of 0.8 among the whole dataset was applied to exclude redundant sequences.

We employed the random undersampling technique to obtain a new balanced dataset by randomly selecting samples from the majority class (non-AIPs). The random undersampling was repeated five times. Next, for each balanced dataset, 80% of the data was randomly selected as the training set for model training and hyperparameter optimization, 10% of the data was randomly selected as the test set for internal validation, and the remaining 10% was the independent set 1 used for external validation.

Furthermore, two additional independent sets named independent set 2 and independent set 3 were constructed. Independent set 2 was the difference set of independent set 1 and AntiInflam's benchmark dataset. Similarly, independent set 3 was the difference set of independent set 1 and AIPpred's benchmark dataset. It is worth noting that independent set 1 used here derived from the balanced dataset on which the model achieved the highest AUC among the five balanced datasets.

Position conservation analysis

Ten residues were extracted from the N-terminus and the C-terminus of each peptide sequence, respectively. The two terminal regions were joined to create a new sequence of 20 residues. The following example shows the process of creating a new sequence from a peptide of length 22 residues.

Original peptide sequence (N - > C): DIELLKKILAGGFIQKYDSVMQ

Ten residues in N-terminus (N - > C): DIELLKKILA

Ten residues in C-terminus (C - > N): QMVSDYKQIF

Created sequence (N-terminus (N - > C) + C-terminus (C - > N)): DIELLKKILAQMVSDYKQIF.

The created sequences of 20 residues were used as inputs for TSL software for generating logo representation of sequences. The first ten positions represented the N-terminus of peptides, and the last ten positions represented the C-terminus of peptides. In the logo graph, the height of each letter is proportional to the frequency of the corresponding residue at that position. To test the statistical significance of AIPs and non-AIPs, the height of the logo was scaled according to the statistical significance threshold of p-value < 0.05 (Welch *t*-test). By the logo representation, we could visualize the differences between the terminal residues of AIPs and non-AIPs.

Selection of feature encoding schemes

In this study, to select the optimal encoding schemes, we firstly employed 13 encoding schemes and combined them with several ML algorithms (see the subsection Selection of ML algorithms below) to construct baseline models. Then, these baseline models were built on the training set and were evaluated by a 10-fold CV and the test set. Encoding schemes used by the models with top performance were selected. The employed 13 encoding schemes can be grouped into three major types: simple composition descriptors (AAC, ATC (Pande et al., 2019), BTC (Pande et al., 2019), DDE, CKSAAP, and TPC (Bhasin and Raghava, 2004)), physicochemical descriptors (PCP (Pande et al., 2019), AAINDEX and C/T/D (Cai et al., 2003)), and Shannon entropy (SER and SPC (Pande et al., 2019)). Details about these descriptors are shown in Table S8. These descriptors were calculated by the *iFeature* toolkit and the *Pfeature* toolkit. A brief introduction to descriptors used in the final model was provided below.

DDE. The DDE descriptor was first proposed by Saravanan et al. in 2015 and was used for linear B-cell epitope prediction (Saravanan and Gautham, 2015). A 400-dimensional feature vector is generated by calculating 3 parameters, that is dipeptide composition measure (D_c), theoretical mean (T_m), and theoretical variance (T_v). D_c is defined as:

$$D_c = \frac{N_{ij}}{L - 1}$$

where N_{ij} is the occurrence time of amino acid pair ij in a given peptide or protein sequence, and L is the sequence length.

T_m is defined as:

$$T_m = \frac{C_i}{C_N} \times \frac{C_j}{C_N}$$

where C_i and C_j stand for the numbers of codons encoding amino acid i and amino acid j in the dipeptide, while C_N stands for the number of all possible codons except the termination codon, that is C_N is equal to 61.

T_v is defined as

$$T_v = \frac{T_m \times (1 - T_m)}{L - 1}$$

So, DDE can be calculated as below:

$$DDE = \frac{D_c - T_m}{\sqrt{T_v}}$$

CKSAAP. The CKSAAP encoding strategy is widely employed in bioinformatics research, and it was also successfully applied in AIP prediction (Khatun et al., 2019). CKSAAP calculates the occurrence frequencies of k (k = 0, 1, 2, 3, 4, 5) -spaced amino acid pairs in a protein or peptide sequence, which can reflect the short-range interactions of amino acids within a sequence or sequence fragment. Taking k = 0 as an example, there are 400 amino acid pairs in 0-space, such as AA, AC, and AD. The composition of 0-spaced amino acid pairs is defined as:

$$\left(\frac{N_{AA}}{N_{total}}, \frac{N_{AC}}{N_{total}}, \frac{N_{AD}}{N_{total}}, \ldots, \frac{N_{YY}}{N_{total}}\right)_{400}$$

where N_{YY} denotes the occurrence number of amino acid pair YY, and the value of N_{total} is equal to sequence length L minus k, namely L - k.

A sequence represented by the CKSAAP encoding is a 400 × (k + 1)-dimensional feature vector. When k = 0, the CKSAAP descriptor is the same as the DPC descriptor. Here, we set k = 5. As a result, this forms a 2400-dimensional feature vector for each peptide sequence.

Feature fusion

Feature fusion was applied here to check whether the performance was improved or not. Briefly, the DDE descriptor (400-dimensional vector) and CKSAAP descriptor (2,400-dimensional vector) were concatenated in a row, so each peptide sequence was converted into a 2,800-dimensional feature vector. After that, the MinMaxScaler normalization technique by the scikit-learn v1.0.2 package was used to scale the hybrid feature vectors into the range of 0 and 1.

Selection of ML algorithms

As mentioned in the subsection Selection of feature encoding schemes, we constructed several baseline models by combining 8 ML algorithms with 13 sequence descriptors to select the optimal ML algorithms simultaneously. ML algorithms used by the models with top performance were selected as the base-classifiers for the AIPStack. The employed ML algorithms included ET, KNN, LightGBM, LR, NB, RF, SVM, and XGBoost. The XGBoost Python package (v1.4.2) and LightGBM Python package (v3.2.1) were used to build the XGBoost model and LightGBM model, respectively. Other classifiers were implemented by scikit-learn v0.24.2 in Python v3.7.10. During the process of algorithm selection, default parameters were used for all models. But before constructing the final model, two selected ML algorithms were fine-tuned using the grid search technique. The two selected algorithms are introduced below, and the description of the other algorithms is presented in Table S9.

RF

RF is a powerful ensemble-based algorithm, which has been successfully utilized in various classification and regression tasks due to its advantages of simplicity, high efficiency, and high accuracy. RF model combines multiple classification and regression trees (CART) to create a "forest". And it improves the prediction performance of CART classifiers by growing numbers of weak CART classifiers. In the classification task, the final results are obtained by simply voting from the classification results of each independent tree. In the regression task, the final results are the average of the outputs of each independent tree. In the RF model construction here, four hyperparameters were optimized, including the number of trees used for constructing the RF classifier (n_estimators), the number of features to consider when looking for the best split (max_features), class weights (class_weight), and the split criterion (criterion). These hyperparameters were tuned using a grid search method implemented by scikit-learn v0.24.2 within the following ranges: n_estimators from 500 to 2,000, with a step size of 50; "auto" or "log2" for max_features; "balanced" or "None" for class_weight; and "Gini" or "entropy" for criterion.

ET

ET was proposed in 2006. It also belongs to a kind of ensemble method and can be applied to classification and regression tasks. ET is very similar to RF, but there are also differences between them. ET fits each decision tree on the whole training dataset, while RF uses the bootstrap sample to grow decision trees. Additionally, ET selects a split point at random, but RF picks an optimal split point according to Gini impurity, information gain, or mean square error. The optimization procedure of hyperparameters in ET was the same as that in the RF method.

Framework of AIPStack

Many previous studies have shown that the ensemble model can achieve better predictive performance than single models in the ensemble, and reduce the generalization error of the prediction (Charoenkwan et al., 2021; Mishra et al., 2019; Basith et al., 2022; Liang et al., 2021; Jiang et al., 2021; Guo et al., 2021). The existing ensemble learning strategies include boosting, bagging, and stacking (Verma and Mehta, 2017). The stacking strategy integrates information from a range of base models to generate a new model, which reflects the idea of seeking the wisdom of crowds. Stacking ensemble models have been successfully applied in many biological tasks so far, such as the classification of therapeutic peptides (Charoenkwan et al., 2021), the prediction of DNA-binding protein (Mishra et al., 2019), and the recognition of non-coding RNA (Mishra et al., 2019).

In this study, a two-layer stacking model named AIPStack was proposed. Firstly, base-classifiers in the first layer were trained on the input dataset, and then the output probabilities from the base-classifiers were used for the second-layer classifier namely the meta-classifier. To avoid overfitting, our stacking model was implemented using the stacking cross-validation algorithm provided in the mlxtend (v0.19.0) package. In the stacking model, as shown in Figure 1, the 10-fold CV was used to split the training set into ten subsets with equal size. In ten successive rounds, each subset was used as the validation set and the remaining nine subsets as the training set in turn. Then, each base-classifier was used for model fitting and output the prediction results of the validation set and the prediction results of the test set. Ten prediction results for the validation set and ten prediction results for the test set were generated in this way. Furthermore, the former was merged into a feature vector matrix for the new training set and the average value of the latter was also merged into a feature vector matrix for the new test set. The new feature matrices together with labels were the final training set and test set for the meta-classifier. Secondly, the new datasets from the first layer were provided as the input datasets of the meta-classifier. LR algorithm was employed as the meta-classifier here.

Model evaluation

To evaluate the performance of the classifiers and compare our model with other existing models, several widely used evaluation metrics for binary classification were employed, i.e. ACC, SE (also called recall), SP, Precision, MCC, and AUC. They are defined as follows.

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

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

$$SP = \frac{TN}{TN + FP}$$

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

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

where TP, TN, FP, and FN represent the numbers of true positives, true negatives, false positives, and false negatives, respectively.

The six different metrics evaluate the performance of the classifier from different perspectives. Precision and SE focus on the classifier's ability to predict positive samples, while SP focuses on the ability to predict negative samples. MCC measures the correlation of the actual classes with the predicted labels. It has a range of −1 to +1, where +1 represents a perfect prediction, 0 means no better than random prediction and −1 indicates a completely wrong prediction. The AUC is a threshold-dependent metric that

summarizes the overall performance of the classifier. For imbalanced datasets, MCC and AUC usually perform better in evaluating the model performance than other metrics (Boughorbel et al., 2017).

It is worth mentioning that the hyperparameter tuning for two base-classifiers was guided by the AUC values, which were calculated through the 10-fold CV on the training set. A parameter combination that achieved the highest AUC was considered the optimal parameter.

Model interpretation

In the present study, we used a powerful and famous framework, SHAP, to help understand the relationships between each feature and positive or negative sample prediction. SHAP method applies a cooperative game theory to calculate the marginal contribution for each feature in a sample, and it reflects how and to what extent a feature affects the final prediction. Here, we calculated the SHAP value for each feature in each sample of the training set. The calculation was conducted via the Python package shap v0.40.0. Then, the features were ranked according to feature importance scores, namely SHAP values. SHAP values with the higher absolute value indicated features with greater overall contributions (either negatively or positively).

QUANTIFICATION AND STATISTICAL ANALYSIS

All computations were performed in the Python programming language. We used the two-sided Mann-Whitney U test to evaluate the statistically significant difference in residue composition for AIPs and non-AIPs. The one-sided Wilcoxon signed-rank test was carried out to statistically compare the performance of the hybrid features and individual features, and compare the performance of the AIPStack and its constituent models. Details pertaining to significance have been also noted in the respective figure legends or table footnotes. The graphic abstract and Figure 1 were generated by Microsoft PowerPoint, other plots appearing in this study were generated by the Python package.