ProsperousPlus: a one-stop and comprehensive platform for accurate protease-specific substrate cleavage prediction and machine-learning model construction

Article *in* Briefings in Bioinformatics · September 2023

DOI: 10.1093/bib/bbad372

CITATIONS
3

READS
127

8 authors, including:

Fuyi Li
Northwest A & F University
104 PUBLICATIONS **4,130** CITATIONS
SEE PROFILE

Cong Wang
Northwest A & F University
7 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Tatsuya Akutsu
Kyoto University
515 PUBLICATIONS **11,885** CITATIONS
SEE PROFILE

Geoffrey I Webb
Monash University (Australia)
508 PUBLICATIONS **22,137** CITATIONS
SEE PROFILE

OXFORD

Briefings in Bioinformatics, 2023, **24(6)**, 1–14
https://doi.org/10.1093/bib/bbad372
Problem Solving Protocol

ProsperousPlus: a one-stop and comprehensive platform for accurate protease-specific substrate cleavage prediction and machine-learning model construction

Fuyi Li, Cong Wang, Xudong Guo, Tatsuya Akutsu, Geoffrey I. Webb, Lachlan J.M. Coin, Lukasz Kurgan and Jiangning Song

Corresponding authors: Fuyi Li, 22 Xinong Road, College of Information Engineering, Yangling, Shaanxi 712100, China. E-mail: Fuyi.Li@nwsuaf.edu.cn; Jiangning Song, Monash Biomedicine Discovery Institute and Data Futures Institute, 19 Innovation Walk, Monash University, Clayton campus, Victoria 3800, Australia. Tel.: +61 3 9902 9304; E-mail: Jiangning.Song@monash.edu; Lukasz Kurgan, Department of Computer Science, Virginia Commonwealth University, Department of Computer Science, 401 West Main Street, Room E4225, P.O. Box 843019, Richmond, Virginia 23284-3019, USA. Tel.: (804) 827-3986; Fax: (804) 828-2771. E-mail: lkurgan@vcu.edu

Abstract

Proteases contribute to a broad spectrum of cellular functions. Given a relatively limited amount of experimental data, developing accurate sequence-based predictors of substrate cleavage sites facilitates a better understanding of protease functions and substrate specificity. While many protease-specific predictors of substrate cleavage sites were developed, these efforts are outpaced by the growth of the protease substrate cleavage data. In particular, since data for 100+ protease types are available and this number continues to grow, it becomes impractical to publish predictors for new protease types, and instead it might be better to provide a computational platform that helps users to quickly and efficiently build predictors that address their specific needs. To this end, we conceptualized, developed, tested and released a versatile bioinformatics platform, *ProsperousPlus*, that empowers users, even those with no programming or little bioinformatics background, to build fast and accurate predictors of substrate cleavage sites. *ProsperousPlus* facilitates the use of the rapidly accumulating substrate cleavage data to train, empirically assess and deploy predictive models for user-selected substrate types. Benchmarking tests on test datasets show that our platform produces predictors that on average exceed the predictive performance of current state-of-the-art approaches. *ProsperousPlus* is available as a webserver and a stand-alone software package at http://prosperousplus.unimelb-biotools.cloud.edu.au/.

Keywords: protease; cleavage site prediction; scoring function; machine learning; ensemble learning; model construction; high-throughput prediction

INTRODUCTION

Proteases are enzymes that cleave their target proteins' peptide backbone, contributing to various cellular processes, including protein degradation, signal transduction and immune response [1–3]. Many proteases are highly specific, cleaving only those target substrates that present certain amino acid sequence patterns. When dysregulated, proteases' actions are closely associated with numerous diseases [2–7], motivating their roles in drug design and disease diagnosis efforts [4–6, 8]. Since current data on the cleavage sites and substrates are incomplete, sequence-based substrate cleavage site predictors can be used to support efforts to understand protease function and substrate specificity. Some of these tools provide accurate predictions [9] and can be applied to identify previously unknown and physiologically relevant cleavage sites, thus providing insights into biological processes and guiding hypothesis-driven experiments to verify protease–substrate interaction and cleavage.

Fuyi Li is a professor at the College of Information Engineering, Northwest A&F University, China. His research interests are bioinformatics, computational biology, machine learning and data mining.
Cong Wang is a PhD student in the College of Information Engineering, Northwest A&F University. Her research interests are bioinformatics and machine learning.
Xudong Guo received his MEng degree from Ningxia University, China. He is currently a research assistant at the College of Information Engineering, Northwest A&F University, China. His research interests are bioinformatics and data mining.
Tatsuya Akutsu received his DEng degree in Information Engineering in 1989 from the University of Tokyo, Japan. Since 2001, he has been a professor at the Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan. His research interests include bioinformatics and discrete algorithms.
Geoffrey I. Webb received his PhD degree in 1987 from La Trobe University. He is the director of the Monash Centre for Data Science and a Professor in the Faculty of Information Technology at Monash University, Australia. He is a leading data scientist and has been the Program Committee Chair of two leading data mining conferences, ACM SIGKDD and IEEE ICDM. His research interests include machine learning, data mining, computational biology and user modelling.
Lachlan J.M. Coin is a professor and group leader in the Department of Microbiology and Immunology at the University of Melbourne. He is also a member of the Department of Clinical Pathology, University of Melbourne. His research interests are bioinformatics, machine learning, transcriptomics and genomics.
Lukasz Kurgan is a fellow of AIMBE and AAIA, Member of European Academy of Sciences and Arts and Robert J. Mattauch Endowed Professor of Computer Science at Virginia Commonwealth University. His research encompasses the structural and functional characterization of proteins. He serves on the Editorial Board of Bioinformatics and as associate editor-in-chief of Biomolecules. More details at http://biomine.cs.vcu.edu/.
Jiangning Song is an associate professor and a group leader at the Monash Biomedicine Discovery Institute, Monash University. He is also affiliated with the Monash Data Futures Institute, Monash University. His research interests include bioinformatics, computational biomedicine, machine learning, data mining and pattern recognition.

Received: July 30, 2023. **Revised:** August 30, 2023. **Accepted:** September 29, 2023

We systematically reviewed and evaluated 19 state-of-the-art computational approaches for the cleavage site prediction [9]. We classified the existing tools into two main categories according to the prediction algorithms that they utilize: (i) scoring function-based predictors, such as PeptideCutter [10], PoPS [11], SitePrediction [12] and GPS-CCD [13], and (ii) machine learning-based tools, such as Cascleave [14], PROSPER [15], Cascleave2.0 [16], iProt-Sub [17], PROSPERous [18], DeepCleave [19] and Procleave [20]. The scoring function-based tools rely on a limited number of scoring functions (i.e. typically just one or two), and thus, they cannot comprehensively encode the amino acid sequence information for a given substrate. While machine learning-based tools do not share this drawback, they require considerable time and availability of high-end computing hardware to compute and tune/optimize their predictive models, especially for the intrinsically large deep learning models. In addition, formulation and calculation of sequence-derived features are often required when developing machine learning models, which requires strong programming skills and may lead to overfitting/overtraining the resulting models if the number of these features is excessive. Moreover, another major problem is that users cannot effectively train models that target specific protease types, and instead, they have to rely on the current tools that may miss some proteases that the users want to target. The latter issue is compounded by the fast pace with which protease substrate cleavage data have been generated and will continue to grow, which means that predictors for the proteases that were recently added will inevitably be missing. In addition, considering data privacy and intellectual property aspects, some labs may prefer to develop predictors using their own in-house generated data instead of relying on the published methods that are trained using public data. These efforts might be rather challenging because of the machine learning and programming skills that are needed to develop accurate predictive tools. Correspondingly, we argue that it might be more practical to provide an easy-to-use computational platform that helps users quickly and efficiently build predictors that address the needs of specific users rather than developing a new predictor for a specific collection of protease types.

Drawing from our extensive experience with building predictors of protease-specific substrates and cleavage sites [17–21], we conceptualized, developed, tested and released *ProsperousPlus*, a versatile bioinformatics platform that can be used to develop fast and accurate predictors for a user-defined collection of protease types. *ProsperousPlus* leverages both the sequence scoring function-based and the machine learning-based approaches to generate accurate predictions. Our platform can be used by users with little to no programming and bioinformatics expertise and is conveniently available as a webserver and a standalone code. Its current version supports the development of models for up to 110 protease types, and this can be easily extended by the inclusion of additional datasets for the other proteases.

MATERIALS AND METHODS

Overview of *ProsperousPlus*

Figure 1A illustrates the four major activities that were needed to develop *ProsperousPlus*, including data collection, sequence scoring, model construction and evaluation and webserver development. As the first activity, we collected and pre-processed training and independent test datasets from the MEROPS resource for 110 protease types [22]. These datasets are needed to train/compute predictors and assess their predictive performance. Second, we implemented several sequence scoring functions and sequence encodings that can be used to generate a diverse and large collection of features from the input protein sequences, which in turn are used as inputs to predictive models. Third, we formulated and implemented the AutoML framework, which supports the construction/training, evaluation and selection of accurate protease-specific prediction models. The training allows for selecting a specific collection of features, predictive model types and protease types. The selection relies on empirical comparison of the predictive performance of predictors that are trained using different features and predictive models. Finally, we developed an online webserver and local standalone software for *ProsperousPlus* using empirically optimized models of each protease type that is covered by our datasets.

Figure 1B illustrates the three functional modules, '*Prediction*', '*TrainYourModel*' and '*UseYourOwnModel*', that are available in both the webserver and local standalone software for *ProsperousPlus*. The '*Prediction*' module provides access to pre-trained prediction models for the 110 protease types, allowing users to conduct protease-specific substrate cleavage site prediction easily. With the '*TrainYourModel*' module, users can train their own prediction models based on their in-house dataset, which can cover proteases outside of the 110 types, using the AutoML framework of *ProsperousPlus*. This module also facilitates comparative evaluation of predictive performance for the generated models. The selected pre-trained and user-generated models can be used to make predictions with the help of the '*UseYourOwnModel*' module.

Dataset collection and pre-processing

We curated experimentally validated protease substrate cleavage annotation data for model training and validation from release 12.4 of the MEROPS database, which is a comprehensive database for protease substrates and their cleavage events [22]. We removed highly homologous sequences (>70% sequence identity) from the initial substrate datasets. This aligns with previous studies [18, 20], and it facilitates the generation of more robust models that will not be skewed towards the prediction of over-represented homologues. Next, we selected proteases with over 30 cleavage sites and split their datasets into training and independent test subsets (the latter is not used for training) using the 7:3 ratio. We excluded proteases with 30 or fewer sites since this amount of data will not be sufficient to perform model training and testing activities. We represent each cleavage site by a window of 20 amino acids, i.e. 10 amino acids in the upstream and 10 amino acids in the downstream of the cleavage site. Using this protocol, we obtained the training and independent test datasets for 110 protease types summarized in Tables S1 and S2. We trained and optimized the *ProsperousPlus* models for each of the 110 protease types using the 10-fold cross-validation on the training datasets, and afterwards, we evaluated their predictive performance utilizing the independent test datasets.

Sequence scoring functions in *ProsperousPlus*

ProsperousPlus relies on the two-layer prediction architecture, which uses multiple sequence scoring functions in the first layer to produce inputs for the second layer that utilizes a machine learning algorithm to make predictions. The first layer employs eight different types of scoring functions, which were found to be useful for substrate cleavage site and protein post-translational modification site predictions [18, 23–25]. These functions calculate scores that quantify the likelihood of a cleavage site, and we describe them in the following subsections.

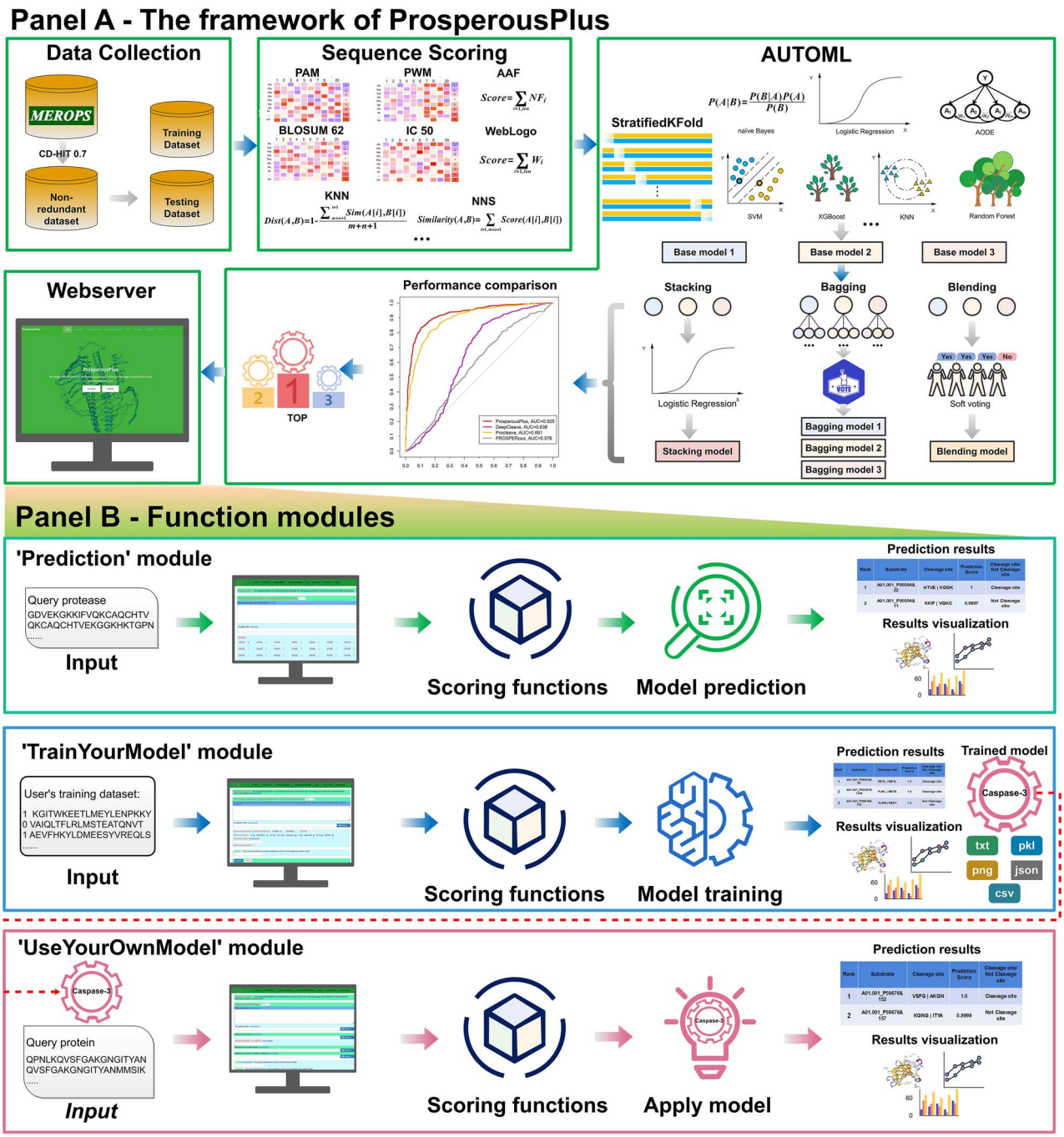

Figure 1. The *ProsperousPlus* framework. (**A**) Graphical illustration of four major activities that were needed to develop *ProsperousPlus*, which include data collection, sequence scoring, AutoML module that facilitates training and evaluation of ML models and webserver development; (**B**) graphical illustration of three functional modules in *ProsperousPlus* that facilitate selection of pre-trained prediction models, training of new models and use of the selected/trained models.

Amino acid frequency

Amino acid frequency (AAF) is a popular sequence scoring function [23, 25, 26] which is defined as

$$\mathrm{AAF} = \sum_{i=P(-10),\ P(+10)} \mathrm{NF_i} \tag{1}$$

where $\mathrm{NF_i}$ is the normalized relative AAF and P denotes the amino acid position surrounding the cleavage site (i.e. P =[−10, −9, −8, −7, −6, −5, −4, −3, −2, −1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]). $\mathrm{NF_i}$ is defined as follows:

$$\mathrm{NF_i} = \frac{f_i}{f_{i,\max}} \tag{2}$$

where $f_i = n_i/N$ represents the frequency value of the amino acid at the position i (e.g. N is the total number of residues of all known cleavage peptide sequences, and n_i is the number of each amino acid type at position i) and $f_{i,\max}$ is the frequency value of the most common amino acid at the same position.

WebLogo-based sequence conservation

WebLogo [27] is a widely used approach based on the calculation of the sequence conservation score (W) [18, 23, 25]. We apply the conservation scores generated by WebLogo to rank the potential cleavage sites. The conservation score of a peptide is calculated as

$$\mathrm{WebLogo}_{\mathrm{Score}} = \sum_{i=P(-10),P(+10)} W_i \tag{3}$$

where W_i denotes the conservation score of the amino acid at the position i (P =[−10, −9, −8, −7, −6, −5, −4, −3, −2, −1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).

Nearest neighbour similarity

Nearest neighbour similarity (NNS) evaluates the similarity between two amino acid sequences $A(m,n)$ and $B(m,n)$, where m and n are the numbers of amino acids flanking the upstream and downstream of the cleavage site [18, 23]. The NNS between two cleavage site sequences $A(m,n)$ and $B(m,n)$ is defined as

$$\mathrm{NNS}(A,B) = \sum_{i=-m,n} \mathrm{Score}(A[i], B[i]) \tag{4}$$

where i runs from $-m$ to n and $\mathbf{Score}(a,b)$ is the corresponding element value in the BLOSUM62 substitution matrix. $A[i]$ and $B[i]$ denote the ith amino acids of sequences $A(m,n)$ and $B(m,n)$. We use $m = n = 10$, which indicates the cleavage sites [$P(-10)$, $P(-9)$, $P(-8)$, $P(-7)$, $P(-6)$, $P(-5)$, $P(-4)$, $P(-3)$, $P(-2)$, $P(-1)$, $P(1)$, $P(2)$, $P(3)$, $P(4)$, $P(5)$, $P(6)$, $P(7)$, $P(8)$, $P(9)$, $P(10)$].

For a given protease type, the putative substrate cleavage site sequence is compared against all training cleavage sites to calculate the sequence similarity scores and the final score is calculated as the average of these similarity scores.

K-Nearest Neighbours

The K-Nearest Neighbour (KNN) identifies the k most similar known cleavage site sequences of a potential cleavage site sequence by calculating the distance between them [23, 25]. The distance between two amino acid sequences $A(m,n)$ and $B(m,n)$ is calculated as follows:

$$\mathrm{Distance}(A,B) = 1 - \frac{\sum_{i=1}^{m+n} \mathrm{Sim}(A[i], B[i])}{m+n} \tag{5}$$

where m and n have the same meaning as NNS described in the Nearest Neighbour Similarity section, and $m = n = 10$ (window size = 20, $i \neq 0$). The $\mathbf{Sim}()$ function calculates the similarity between the amino acids $A[i]$ and $B[i]$ as follows:

$$\mathrm{Sim}(a,b) = \frac{\mathrm{BLOSUM62}(a,b) - \min\{\mathrm{BLOSUM62}\}}{\max\{\mathrm{BLOSUM62}\} - \min\{\mathrm{BLOSUM62}\}} \tag{6}$$

where a and b denote the amino acids from the sequences A and B, respectively. We use $k = 0.005 \times N$ (N is the total number of sequences) to calculate the KNN score.

Position probability matrix-based scoring function

The position probability matrix (PPM) is a 21×20 matrix, where '21' denotes the 20 types of common amino acids and X (undefined amino acids and gap '-'), and '20' is the length of cleavage site sequences. The PPM captures positional preference of amino acids in the cleavage site sequence by calculating probability of each amino acid occurring at different positions within the sequence [28]. The PPM is calculated as follows:

$$F_{a,i} = \frac{C_{a,i} + 1}{M}, i \in [1, 20] \tag{7}$$

where $F_{a,i}$ is the probability of amino acid a at the position i in the PPM. $C_{a,i}$ represents the number of residues of amino acid type a at position i, and M is the number of samples. Given a cleavage site $S = \{a_1 a_2 \ldots a_{20}\}$, the average of the PPM elemental values corresponding to all amino acids in the sequence is calculated as the feature of the sequence. The score of a sequence is computed as follows:

$$\mathrm{PPM}_{\mathrm{score}} = \frac{\sum_{i=1}^{20} F_{S[i],i}}{20} \tag{8}$$

where$S[i]$ is the ith amino acid in the sequence S.

Position-specific scoring matrix-based IC50 scoring function

The position-specific scoring matrix (PSSM) is also a 21×20 matrix, similar to the PPM. The PSSM quantifies evolutionary information of amino acids occurring at different positions in a given sequence [29], and it has been widely used in many related bioinformatics prediction tasks [30–32]. PSSMs are generated by first calculating the sequence-weighted frequency of the amino acids at each position on the cleavage site sequence, then the frequency is normalized by the background frequency of corresponding amino acids and finally transformed by a log transformation [33]. The elements of a PSSM are calculated as follows:

$$P_{a,i} = \log \frac{F_{a,i} + \omega}{BG_a} \tag{9}$$

where $P_{a,i}$ represents the element value of amino acid a at position i in the PSSM; $F_{a,i}$ is the frequency of amino acid a at position i; $\mathbf{BG}_a$ denotes the background frequency corresponding to the amino acid a, and ω takes a value from the 0 to 1 range. We obtain the background frequencies of amino acids from the UniProt database [34].

The PSSM cannot be directly used as the scoring function to calculate the score for the cleavage site sequence. We use the half-maximum inhibitory concentration (IC50) score, which is widely used for the binding affinity prediction [32, 35], to produce the PSSM-based score for the cleavage site sequences. The IC50 score is computed as follows:

$$\mathrm{Seq\_score} = \frac{\sum_{i=1}^{20} P_{a,i}}{20} \tag{10}$$

$$\mathrm{IC50} = 5000^{\mathrm{Max} - \mathrm{Seq\_score}} / \mathrm{Max} - \mathrm{Min} \tag{11}$$

where we set $\mathbf{Max} = 0.8$ and $\mathbf{Min} = -0.8$ based on Liu *et al.* [30].

Position weight matrix-based scoring function

The position weight matrix (PWM) is a 21×20 matrix, similar to the PPM. The PWM is used to identify sequence patterns or motifs with specific functions in the sequences, by identifying regions with higher position weights [25, 36, 37]. The PWM is calculated as follows:

$$\mathrm{PW}_{a,i} = \log_2 \frac{F_{a,i}}{0.05} \tag{12}$$

where $\mathbf{PW}_{a,i}$ denotes the position weight of amino acid a at position i in PWM, and $F_{a,i}$ is the element value in the PPM. For a given cleavage site sequence $S = \{a_1 a_2 \ldots a_{20}\}$, the score is determined

by taking the average position weight of all amino acids within the sequence. The formula is as follows:

$$\text{Score}_{\text{PWM}} = \frac{\sum_{i=1}^{20} \text{PW}_{S[i],i}}{20} \tag{13}$$

where $S[i]$ denotes the ith amino acid in the sequence S.

Substitution matrix index-based scoring function

The substitution matrix index (SMI) is utilized to compute similarity amongst sequences and is extensively employed for extracting features from sequences [38–41]. We use five different versions of BLOSUM scoring matrices, including BLOSUM100, BLOSUM75, BLOSUM62, BLOSUM45 and BLOSUM30, and five point accepted mutation (PAM) matrices, including PAM500, PAM400, PAM300, PAM120 and PAM30, to evaluate similarity between the query cleavage site and other cleavage site sequences from the training dataset. For a given query sequence $S = \{a_1 a_2 \ldots a_{20}\}$, the SMI score is calculated as

$$\text{Score}_{\text{SMI}} = \max\left(\left\{\text{Score}_{Sj} = \frac{\sum_{i=1}^{20} \text{Matrix}(\text{index}(S[i]), \text{index}(St_j[i]))}{\sum_{i=1}^{20} \text{Matrix}(\text{index}(St_j[i]), \text{index}(St_j[i]))} : j = 1, \ldots, m\right\}\right)$$

where $\textbf{Matrix}(x, y)$ is the corresponding element value with index (x, y) of the BLOSUM or PAM scoring matrix, and $S[i]$ is the ith amino acid of the query sequence S and $St_j[i]$ is the ith amino acid of the sequence St_j, which is the jth sequence in the training dataset (training dataset has m sequence in total). Besides, the $\textbf{index}(a)$ function can get the corresponding index of amino acid a in the matrix. We use the maximum $\textbf{Score}_{Sj}$ value as the SMI score, and we generate 10 SMI scores for the 10 different substitution matrices.

AutoML framework

The AutoML framework automates the entire process of designing and selecting an accurate predictor, making it easy for users who lack machine learning and coding expertise. In particular, AutoML implements and considers multiple versions of the two-layered predictors, performs advanced machine learning modelling and comparative analysis and selects the most accurate solution. The predictors generate the 17 scores in the first layer by combining the outputs of the eight scoring functions that are produced from the input sequence. The second layer applies the 17 scores to train predictors by utilizing nine popular machine learning algorithms, including Logistic Regression (LR), Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), KNN, CatBoost [42], Extreme Gradient Boosting (XGBoost) [43], Light Gradient Boosting Machine (LightGBM) [44] and Averaged One-Dependent Estimator (AODE) [45]. The framework considers using each of the nine algorithms, performs a robust comparative analysis of their predictive performance using multiple performance metrics, attempts to improve the most promising models using popular feature selection and ensemble algorithms and ultimately outputs the model that secures the most accurate predictions.

We detail this process in Algorithm 1: First, the inputs that include the 17-dimensional feature sets for the training dataset S and the number of cross-validation folds k are collected. We use the stratified sampling strategy (*StratifiedKFold*) to divide the feature set $\boldsymbol{S}$ into the k cross-validation folds. Steps 2–7 involve applying the stratified $\boldsymbol{k}$-fold cross-validation to train and validate the nine machine learning models: LR, NB, SVM, RF, KNN, CatBoost, XGBoost, LightGBM and AODE. Next, the accuracy and area under the receiver-operating characteristic (ROC) curves (AUC) values of each model on $\boldsymbol{S_i^{valid}}$ are obtained to evaluate and select the top three best models (denoted as m_1, m_2 and m_3). The top three are selected by sorting the nine models by their average AUC values across the k validation folds set. If selected models have the same average AUC value, then we use the average accuracy value to break the tie. In step 9, we attempt to optimize the top three models using the incremental feature selection algorithm [46] (this method was shown to produce strong results in several related studies [47–53]), which results in models ($\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$, $\boldsymbol{m_3'}$). Then, we generate five ensemble models of $\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$ and $\boldsymbol{m_3'}$ using three different popular ensemble learning strategies, including stacking [54], blending [55] and three versions of bagging [56]. Subsequently, we obtain eight models, which include the three base models $\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$ and $\boldsymbol{m_3'}$ and five ensemble models ***stacked_m'***, ***blended_m'***, ***bagging_m$_1$'***, ***bagging_m$_2$'*** and ***bagging_m$_3$'***. Finally, in line 13, we select the *optimised_model*, which is the model that secures the highest AUC and accuracy values amongst the eight alternatives.

Algorithm 1 The AutoML pipeline of *ProsperousPlus*

Input:
  $\boldsymbol{S}$: feature set of the training dataset;
  $\boldsymbol{k}$: the cross-validation fold number;
Output:
  optimised_model: the optimal prediction model;
01: *cv* = *StratifiedKFold* ($\boldsymbol{S}$, $\boldsymbol{k}$);
02: **for i,** ($\boldsymbol{S_i^{train}}$, $\boldsymbol{S_i^{valid}}$) in enumerate(*cv.split* ($\boldsymbol{S}$)):
03: for *j*, ***m* in enumerate** ([*LR, NB, SVM, RF, KNN, CatBoost, XGBoost, LightGBM, AODE*]):
04: $m_{i,j}$ = *trainModel*(m, $\boldsymbol{S_i^{train}}$);
05: *ACCvalues*[i][j], *AUCvalues*[i][j] = *predictModel*($m_{i,j}$, $\boldsymbol{S_i^{valid}}$);
06: end for;
07: end for;
08: m_1, m_2, m_3 = *rank* (*ACCvalues*[i][j].*mean*, *AUCvalues*[i][j].*mean*);
09: $\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$, $\boldsymbol{m_3'}$ = *IFS* (m_1, m_2, m_3, *cv*, *S*);
10: ***stacked_m'*** = *stack_models* ($\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$, $\boldsymbol{m_3'}$);
11: ***blended_m'*** = *blend_models* ($\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$, $\boldsymbol{m_3'}$);
12: ***bagging_m$_1$'***, ***bagging_m$_2$'***, ***bagging_m$_3$'*** = *bagging_model* ($\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$, $\boldsymbol{m_3'}$);
13: *optimised_model* = *compare_models* ($\boldsymbol{m_1'}$, $\boldsymbol{m_2'}$, $\boldsymbol{m_3'}$, ***stacked_m'***, ***blended_m'***, ***bagging_m$_1$'***, ***bagging_m$_2$'***, ***bagging_m$_3$'***)
14: return *optimised_model*;

Performance evaluation

AutoML calculates several widely used performance measures including accuracy, sensitivity, precision, F1 and Matthew's correlation coefficient (MCC) [57]. These measures are calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{15}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{17}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \times (\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TN} + \text{FN})}} \tag{18}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

where TP, TN, FP and FN denote the numbers of true positives, true negatives, false positives and false negatives, respectively. Moreover, the AutoML framework also plots the ROC curves and calculates the AUC values [58].

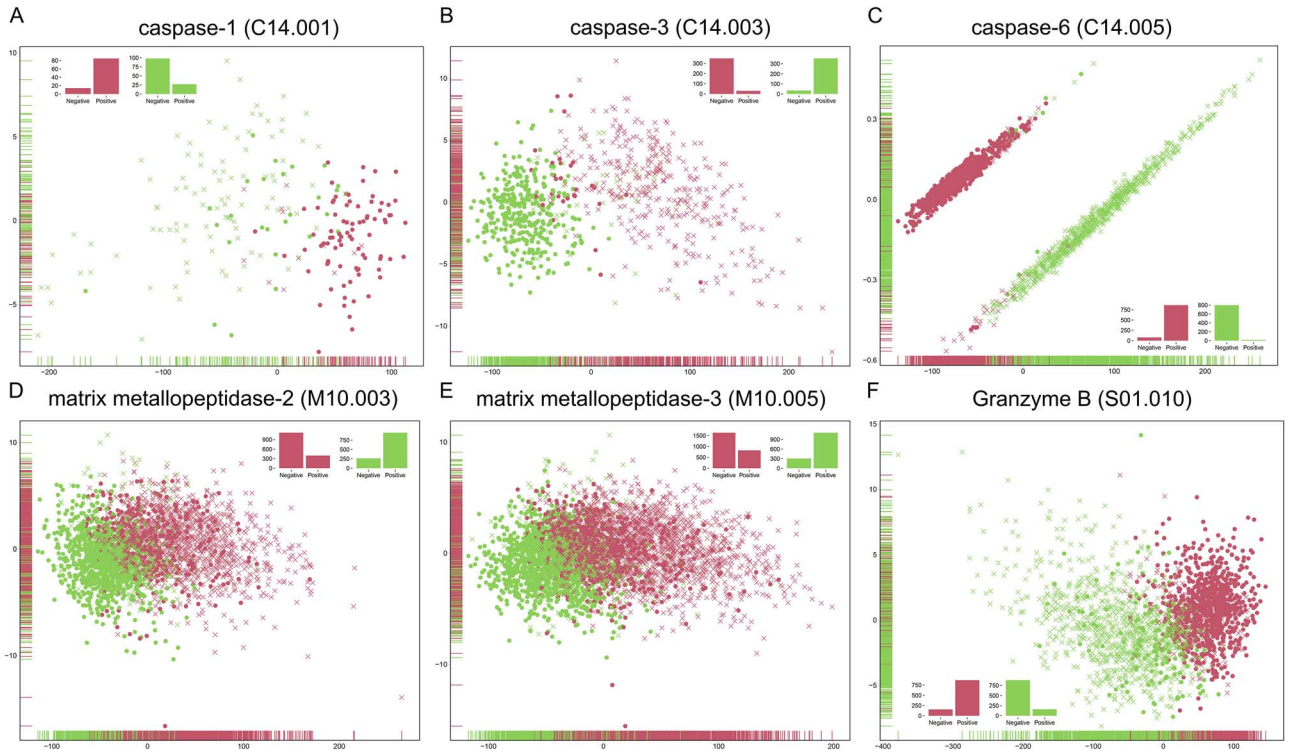

Figure 2. Distribution and clustering of the cleavage site and non-cleavage sites based on the sequence scores selected by the IFS algorithm for six proteases: (**A**) caspase-1, (**B**) caspase-3, (**C**) caspase-6, (**D**) MMP-2, (**E**) MMP-3 and (**F**) Granzyme B. For each protease, the samples are clustered into two groups using the K-means algorithm, and clusters are colour-coded. The cleavage and non-cleavage sites are presented by different markers, where dots are for the cleavage sites and '×' for the non-cleavage sites. The inset bar charts show the fractions of cleavage versus non-cleavage sites in each cluster.

RESULTS AND DISCUSSION

Predictive quality of the sequence scoring functions

We use the training datasets to investigate the ability of the 17 sequence scores generated by eight scoring functions to differentiate cleavage from non-cleavage sites in two complementary ways. First, we perform unsupervised clustering to see whether these scores can be used for the natural grouping of the sequences (without using the cleavage annotation) and whether these groups align with the annotations. Second, we use Algorithm 1 to quantify the performance of the predictive models that rely on these scores. Both experiments focus on six diverse proteases, including caspase-1, caspase-3, caspase-6, matrix metallopeptidase-2, matrix metallopeptidase-3 and Granzyme B. These proteases cover a wide range of training dataset sizes, from 112 cleavage sites for caspase-1 to 1600 for matrix metallopeptidase-3 (Table S2), and they are frequently targeted by the existing predictors.

We apply the popular *K*-means clustering [59] with $K=2$ to mimic the presence of the two populations (cleavage sites versus non-cleavage sites) in the training datasets. We process the datasets for each protease using the IFS algorithm before running clustering to accommodate for the fact that this simulates the processing done in the first layer of the two-layered predictive model that we use (section Sequence Scoring Functions in *ProsperousPlus* and Algorithm 1). Figure 2 shows the resulting clusters for the six proteases, including caspase-1 (Figure 2A), caspase-3 (Figure 2B), caspase-6 (Figure 2C), matrix metallopeptidase-2 (Figure 2D), matrix metallopeptidase-3 (Figure 2E) and Granzyme B (Figure 2F). The clustering results are provided in Table S3. To ease the visualization, this figure relies on the two-dimensional feature space that applies the two dominant dimensions extracted using the principal component analysis. We represent the cleavage sites using the dot markers and the non-cleavage sites using the x markers, while the two clusters are colour-coded in red and green.

The colour-coded clusters reveal relatively clear natural groupings. We find particularly well-separated clusters for caspase-1, caspase-3, caspase-6 and Granzyme B (Figure 2A–C and F, respectively). Moreover, these natural clusters (i.e. clusters obtained without using the cleavage site annotations) are in good agreement with the cleavage site annotations, where cleavage sites (positives) are primarily grouped in the red clusters while the non-cleavage sites (negatives) dominate the green clusters in caspase-1, caspase-6 and Granzyme B. In contrast, cleavage sites are primarily grouped in the green cluster, and non-cleavage sites dominate the red cluster in caspase-3. For instance, for caspase-6, 91.59% of the cleavage sites are in the red cluster, and 97.80% of the non-cleavage sites are in the green cluster. While the scores of MMP-2 and MMP-3 achieved relatively poorer clustering performance compared with the other four proteases, but still showed far better results than random classification. These results reveal that the sequence scoring functions that we employ produce relatively well-defined natural clusters that align with the native annotations of cleavage sites, suggesting that these features should be useful for predicting cleavage sites across different types of proteases.

We also evaluate the predictive performance of the complete two-layer models using the 10-fold cross-validation on the training dataset for the same six proteases (Table 1). We quantify the predictive quality with a comprehensive collection of metrics,

Table 1: Performance evaluation of *ProsperousPlus* based on the 10-fold cross-validation on the training dataset for six proteases: C14.001 (caspase-1), C14.003 (caspase-3), C14.005 (caspase-6), M10.003 (MMP-2), M10.005 (MMP-3) and S01.010 (Granzyme B). The results are presented as the averages over the 10-folds ± the corresponding standard deviations

Protease	Accuracy	AUC	Sensitivity	Precision	F1	MCC
C14.001	0.875 ± 0.058	0.910 ± 0.053	0.846 ± 0.112	0.913 ± 0.067	0.872 ± 0.062	0.762 ± 0.111
C14.003	0.970 ± 0.015	0.989 ± 0.011	0.902 ± 0.047	0.924 ± 0.050	0.913 ± 0.044	0.895 ± 0.053
C14.005	0.970 ± 0.006	0.992 ± 0.006	0.974 ± 0.014	0.968 ± 0.015	0.971 ± 0.005	0.941 ± 0.011
M10.003	0.863 ± 0.023	0.922 ± 0.022	0.879 ± 0.026	0.847 ± 0.024	0.863 ± 0.023	0.728 ± 0.045
M10.005	0.869 ± 0.019	0.931 ± 0.015	0.881 ± 0.025	0.860 ± 0.026	0.870 ± 0.019	0.738 ± 0.038
S01.010	0.909 ± 0.030	0.953 ± 0.015	0.887 ± 0.041	0.925 ± 0.033	0.906 ± 0.033	0.819 ± 0.060

including accuracy, AUC, sensitivity, precision, F1 and MCC. We find that combining the sequence scoring functions with the machine learning classifiers produces relatively strong predictions, with AUCs ranging between 0.86 (for metallopeptidase-2) and 0.97 (for caspase-6 and caspase-3). The values of the other metrics are also high, with the MCC values between 0.73 and 0.95 that correspond to strong correlations, sensitivities >0.87 and precision scores >0.84. Altogether, these results suggest that the machine learning algorithms that are used produce accurate predictions, which means they can take advantage of the high-quality features produced by the sequence scoring functions.

Comparison with other modern predictors

We used the independent test datasets to compare the predictive performance of models generated by *ProsperousPlus* against several state-of-the-art approaches that include SitePrediction [12], PROSPERous [18], DeepCleave [19] and Procleave [20]. We collected predictions of these four tools using their corresponding webservers. We compare results for 12 proteases (caspase-1, caspase-3, caspase-6, caspase-7, MMP-2, MMP-3, MMP-7, MMP-8, MMP-9, MMP-12, Granzyme B and thrombin) that are commonly predicted by these tools. We summarize the results in Figures 3 and 4 and Table S4.

Figure 3 and Table S4 show the performance comparison results between *ProsperousPlus* and state-of-the-art approaches regarding three major evaluation metrics (accuracy, MCC and F1) for 12 proteases on the independent test datasets. The comparison results demonstrated that *ProsperousPlus* achieved optimal predictive performance in most of the proteases. More specifically, *ProsperousPlus* performed best on five out of 12 proteases in all three metrics (accuracy, MCC and F1), including caspase-1 (C14.001), caspase-6 (C14.005), MMP-8 (M10.002), MMP-2 (M10.003) and MMP-12 (M10.009). In addition, *ProsperousPlus* achieved the best performance on MMP-3 (M10.005) and thrombin (S01.217) in two of three metrics. To be precise, *ProsperousPlus* achieved the best accuracy and F1 and the second-best MCC (Prosperous achieved the best MCC) on MMP-3. For thrombin, *ProsperousPlus* secured the best accuracy and MCC and the second-best F1 (Procleave achieved the best F1). Besides, *ProsperousPlus* ranked second on four proteases regarding these three metrics, including caspase-3 (C14.003), MMP-9 (M10.004), MMP-7 (M10.008) and Granzyme B (S01.010), with slightly lower performance compared with Procleave. However, *ProsperousPlus* ranked 4th on caspase-7 (C14.004) in terms of three performance metrics.

Figure 4 shows the ROC curves of *ProsperousPlus* and state-of-the-art methods on these 12 proteases. The AUC value is a threshold-independent metric that provides an aggregate measure of a model's ability to distinguish between positive and negative samples across all possible thresholds, with a value ranging from 0 to 1. AUC values closer to 1 indicate better performance. *ProsperousPlus* performed best regarding the AUC with eight out of the 12 proteases, including caspase-1 (C14.001), caspase-3 (C14.003), caspase-7 (C14.004), caspase-6 (C14.005), MMP-2 (M10.003), MMP-9 (M10.004), MMP-3 (M10.005) and MMP-12 (M10.009), while on the MMP-8 (M10.002), *ProsperousPlus* and Procleave both achieved the best AUC value. In addition, *ProsperousPlus* achieved the second-best AUCs on MMP-7 (M10.008), Granzyme B (S01.010) and thrombin (S01.217), and Procleave performed best on these two proteases. The average AUC across all considered proteases for *ProsperousPlus* was 0.966, surpassing the second-best AUC of 0.950 achieved by Procleave and the third-best AUC of 0.923 achieved by DeepCleave. Overall, all these results demonstrated that ProsperousPlus achieved competitive predictive performance that exceeded the predictive performance of current state-of-the-art approaches on average.

Model interpretation

We employ the Shapley additive explanation (SHAP) algorithm [60] to perform an interpretability analysis of the *ProsperousPlus* models. SHAP assesses the importance of input features through the calculation of Shapley values. This tool is frequently applied to analyse related bioinformatics models [49, 50, 61–63]. While *ProsperousPlus* utilizes feature selection to develop a well-performing subset of features in Algorithm 1, here, we study the contributions of individual features to the predictive quality of the resulting predictive models.

Figure 5 shows the feature importance ranking generated by SHAP for the 12 proteases that we use in the comparative analysis in section Comparison with Other Modern Predictors. Each panel in this figure ranks the features from the most useful (at the top) to the least useful (at the bottom). Each feature is represented by a plot that shows the distribution of SHAP values (positive and negative values that quantify associations with predictions) for test sequences, while the colour gradient from blue to red visualizes the distribution of feature values from low to high. These plots provide useful insights into the impact and association of values of the features on the model performance.

We found that several features provide substantial contributions to the prediction of the cleavage sites across multiple proteases, including KNN, NNS, WLS and IC50. Notably, the KNN scores are identified by SHAP as the most important feature in six out of 12 proteases (caspase-1, caspase-3, MMP-3, MMP-9, MMP-12 and Granzyme B), the second important feature for three proteases (caspase-6, MMP-7 and MMP-8) and the third for two proteases (MMP-2 and thrombin), respectively. The SHAP values' distributions of KNN scores for all these 12 proteases indicate that *ProsperousPlus* is more likely to predict samples with larger KNN values as the cleavage sites. The WLS scores are also found to be useful, surpassing the KNN for three proteases (caspase-6, MMP-7 and thrombin). In addition, WLS ranked second for

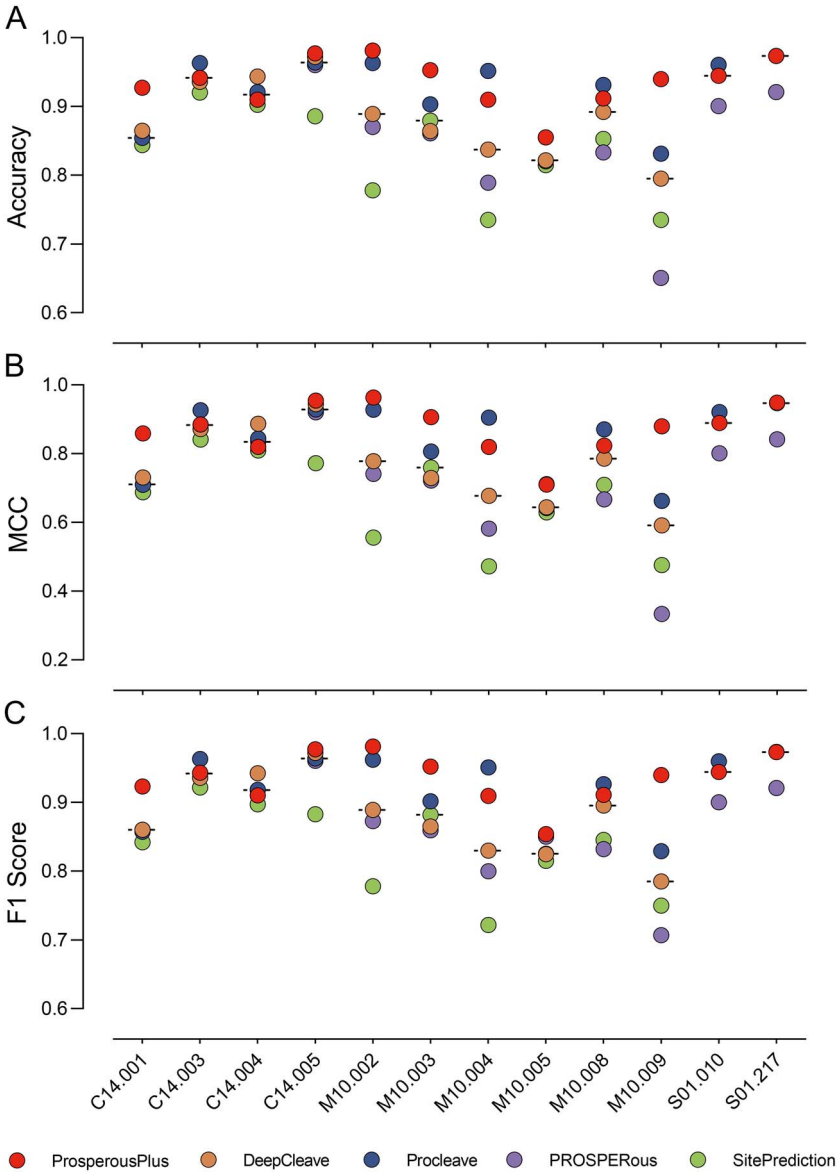

Figure 3. Performance comparison between *ProsperousPlus* and state-of-the-art approaches, including DeepCleave, Procleave, PROSPERous and SitePrediction, in terms of Accuracy, MCC and F1 on the independent test datasets.

caspase-3 and ranked third for five proteases, e.g. caspase-1, MMP-3, MMP-8, MMP-12 and Granzyme B. Generally, a larger value of the WLS feature promotes the prediction of cleavage sites, while a lower WLS value favours the prediction of non-cleavage sites. Conversely, IC50 (which ranked as the second important feature for MMP-2, MMP-3 and Granzyme B and ranked as the third important feature for caspase-3 and caspase-6) demonstrated an opposite effect compared to KNN and WLS features. SHAP plots revealed that larger values of the IC50 score in most proteases led to a higher likelihood of non-cleavage site prediction. In summary, KNN, NNS, WLS and IC50 features are arguably the key features that contribute to the cleavage site predictions by the models produced by *ProsperousPlus*. The significance of these scoring functions presumably arises from their adeptness at capturing specific motifs, short conserved sequences or functional domains that play pivotal roles in the biological activity of the sequences. These functions might excel in pinpointing such crucial regions that substantially contribute to the prediction. Furthermore, scoring functions such as KNN and NNS take into consideration the nearest neighbours of each sequence, which might aid in capturing context-specific insights. This might signify conserved functional contexts that underlie the observed biological activity. Moreover, KNN, NNS and WLS possess the capability to capture non-linear relationships within the sequence data through their proximity-based methodology, enabling them to grasp intricate interactions that collectively contribute to the prediction outcome.

Furthermore, we contrast distributions of values of the best four features for the cleavage sites (positives in red) and the non-cleavage sites (negative in green) in Figure S1. We find that these distributions are substantially different between the sequences of the cleavage versus the non-cleavage sites. Importantly, the differences are consistent across different protease types, i.e. the median IC50 values, are uniformly lower for the cleavage sites, and the median WLS, KNN and NNS values are higher for the cleavage sites. This explains why and how our models can so consistently produce highly accurate results over the different types of proteases.

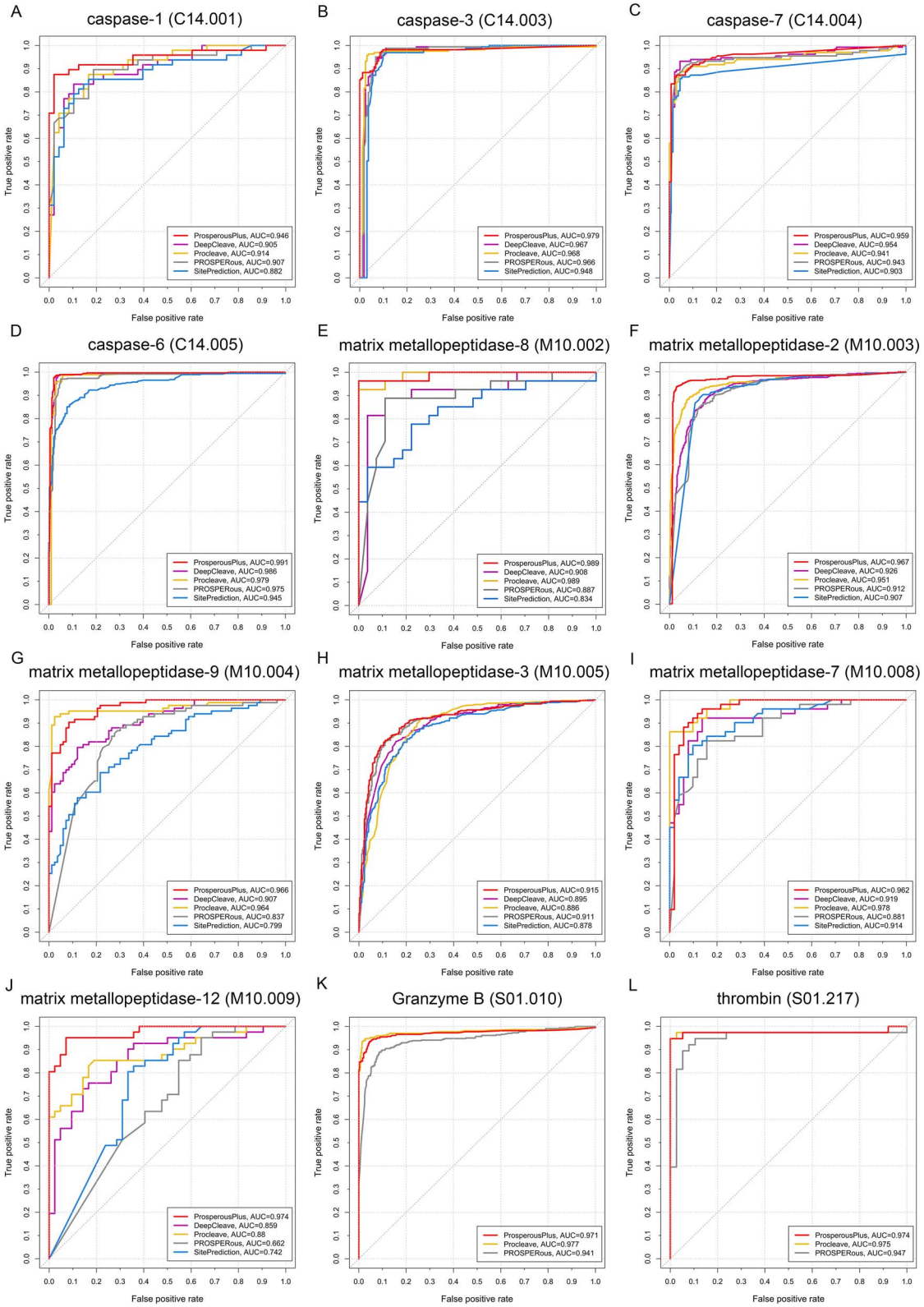

Figure 4. ROC curves of *ProsperousPlus* and other state-of-the-art approaches (including SitePrediction, PROSPERous, DeepCleave and Procleave) for the cleavage site prediction of caspase-1, caspase-3, caspase-6, caspase-7, MMP-2, MMP-3, MMP-7, MMP-8, MMP-9, MMP-12, Granzyme B and thrombin on the independent test datasets.

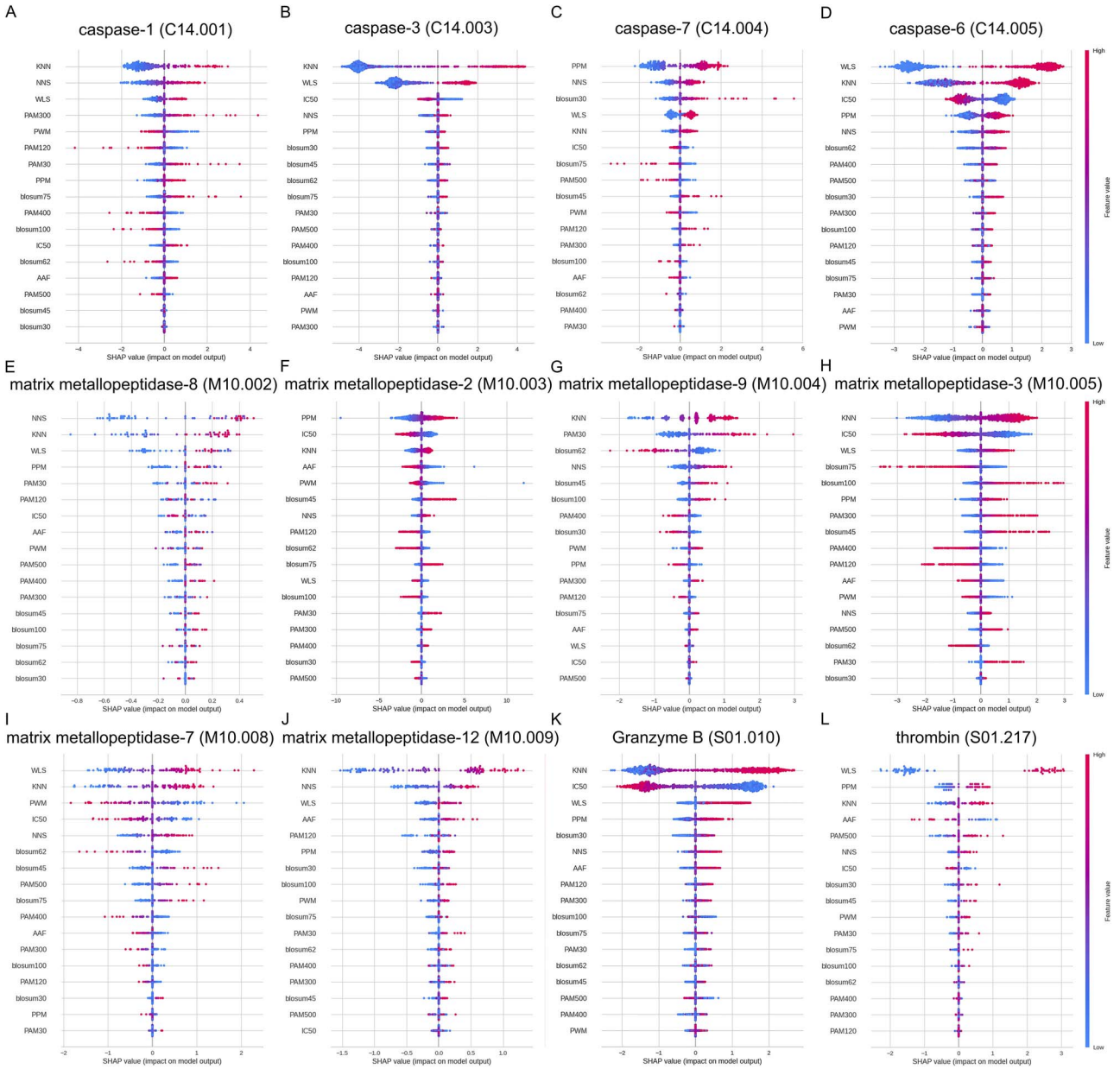

Figure 5. Feature interpretations according to SHAP values for *ProsperousPlus* prediction of protease-specific cleavage sites on the independent test datasets. The order from top to bottom represents the importance ranking of features. Colours indicate feature values (red: high; blue: low), and SHAP (positive or negative) values indicate the directionality of the top features. Positive SHAP values indicate positive predictions (cleavage sites), while negative SHAP values indicate negative predictions (non-cleavage sites).

Webserver and local standalone software for *ProsperousPlus*

We amplify the impact of the *ProsperousPlus* resource by implementing it as both an online webserver and standalone software that can be used locally. We rely on the PHP and Python programming languages, which are relatively popular and should be easy to use by the end users. The webserver is freely available at http://prosperousplus.unimelb-biotools.cloud.edu.au/, while the local standalone version is available at https://github.com/lifuyi774/ProsperousPlus. Both versions include the three modules that we introduce in section Overview of *ProsperousPlus*, i.e. '*Prediction*', '*TrainYourModel*' and '*UseYourOwnModel*' (Figure 6A). Here, we provide more details.

The '*Prediction*' module provides access to the pre-computed predictive models for the 110 protease types, allowing users to instantly make predictions for the corresponding protease-specific cleavage sites. Users can input or upload their substrate sequences in the FASTA format and then select the proteases of interest to make the prediction (Figure 6B-1). We limited each prediction job to <1000 sequences in the webserver to ensure that long jobs do not block this resource for other users. The standalone version does not have this limit and is recommended in cases where users want to process large datasets. Using the '*TrainYourModel*' and '*UseYourOwnModel*' modules, users can train new models based on their in-house data (including protease types outside of the list of the 110 for which we pre-computed the models) and then apply these trained models to make cleavage site predictions (Figure 6B-2). For the '*TrainYourModel*' module, users need to provide a training dataset file containing cleavage site sequences (with sequence lengths ranging from

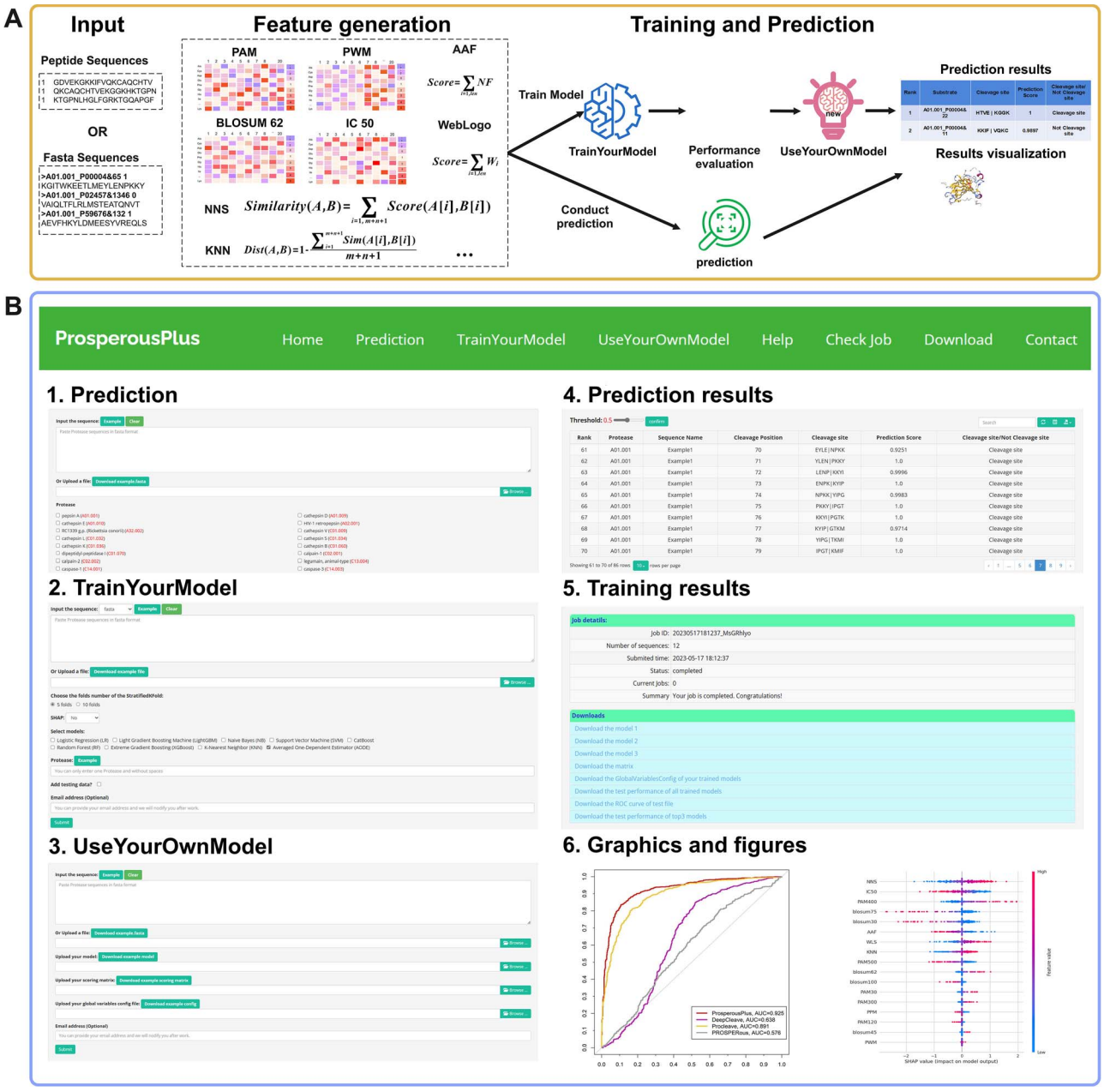

Figure 6. The *ProsperousPlus* webserver. (**A**) Intuitive graphical workflow of the webserver. (**B**) An example to showcase the usage of the *ProsperousPlus* webserver that covers the input page for the '*Prediction*', '*TrainYourModel*' and '*UseYourOwnModel*' modules (subpanels 1, 2 and 3), the prediction results (subpanel 4), the model training results (subpanel 5) and visualization of results (subpanel 6).

8 to 20) to train and build the model. In addition, users can optionally provide a test dataset file to test the performance of the trained model. Sequences of interest can also be optionally provided to make predictions by using the trained model. For the '*UseYourOwnModel*' module, users need to provide the trained model file and sequences of interest in the FASTA format to make the prediction.

Upon submission, a unique Job ID will be generated to refer to the job summary page during the job execution process. Users can use this Job ID to enquire and track the execution progress of their job and access and download their prediction results once they are ready (Figure 6B-4, B-5 and B-6). The '*Prediction*' module outputs consist of the prediction results and visualization plots of predicted cleavage site positions in the query FASTA sequences. The outputs of the '*TrainYourModel*' module include the trained model file, the performance evaluation results file, the ROC curves of the trained model and a graphical plot including predicted cleavage sites with corresponding positions (if the query sequence in the FASTA format is provided). The outputs of the '*UseYourOwnModel*' modules are the same as the '*Prediction*' module, including the prediction results and plots of the predicted cleavage sites.

DISCUSSION AND CONCLUSION

We introduce *ProsperousPlus*, a user-friendly and comprehensive bioinformatic platform for accurate protease-specific cleavage site prediction. *ProsperousPlus* generates well-designed two-layer prediction models that combine the strengths of scoring function-based features, machine learning algorithms and multiple model

ensemble strategies (stacking, blending and bagging). We empirically show that the scoring functions that *ProsperousPlus* uses provide high-quality inputs for the predictive models across different types of proteases. Further benchmarking using the independent test datasets reveals that models generated by *ProsperousPlus* are substantially more accurate than the results produced by modern predictors of cleavage sites on average. The key features of *ProsperousPlus* include

1) Comprehensive coverage of the protease-specific substrate and cleavage site predictions. While our platform is pre-loaded with models for the 110 protease types, its innovative AutoML pipeline facilitates an easy (even for users without programming and bioinformatics background) development of models for other protease types. This arguably makes *ProsperousPlus* the most comprehensive tool that is available to date.
2) State-of-the-art levels of predictive performance. This stems from the well-informed design of our platform, which benefits from our expertise as authors of PROSPERous, iProt-Sub and DeepCleave predictors.
3) Availability of both the webserver and standalone code versions. This amplifies the impact of our tools by catering to the needs of different types of users, e.g. occasional versus frequent users, users who want to run large jobs versus predictions for small datasets and bioinformaticians who want to use these predictors in other bioinformatics tools.

The combination of these features arguably makes *ProsperousPlus* an invaluable tool, empowering users to perform accurate protease-specific substrate cleavage site prediction and to train in-house models to meet the ever-expanding pool of substrate cleavage site data.

While *ProsperousPlus* represents a significant advancement in protease-specific cleavage site prediction, it has certain limitations: Firstly, although the platform encompasses an extensive array of protease types, the predictive performance might vary across different protease families due to variations in cleavage site characteristics. Additionally, while incorporating multiple scoring functions enhances the predictive power, the selection of an optimal set of features and models remains a challenge for some proteases, which might be addressed through ongoing refinement. In terms of future research, exploring cutting-edge deep learning architectures, such as large language models (LLMs), could be potentially leveraged to further enhance the predictive accuracy and uncover complex interactions governing protease substrate recognition.

Key Points

- Comprehensive coverage of the protease-specific substrate and cleavage site predictions. While *ProsperousPlus* is pre-loaded with models for the 110 protease types, its innovative AutoML pipeline facilitates the easy development of models for other protease types.
- State-of-the-art levels of predictive performance. This stems from the well-informed design of our platform, which benefits from our expertise as authors of PROSPERous, iProt-Sub and DeepCleave predictors.
- Availability of both webserver and standalone code versions at http://prosperousplus.unimelb-biotools.cloud.edu.au/ and https://github.com/lifuyi774/ProsperousPlus. This amplifies the impact of *ProsperousPlus* by catering to the needs of different types of users.

SUPPLEMENTARY DATA

Supplementary data are available online at https://academic.oup.com/bib.

FUNDING

The National Natural Scientific Foundation of China (No. 62202388); the National Key Research and Development Program of China (No. 2022YFF1000100); the Qin Chuangyuan Innovation and Entrepreneurship Talent Project (No. QCYRCXM-2022-230); Talent Research Funding at Northwest A&F University (No. Z1090222021); the Major and Seed Inter-Disciplinary Research Projects awarded by Monash University.

DATA AND CODE AVAILABILITY

The *ProsperousPlus* tool is freely available at http://prosperousplus.unimelb-biotools.cloud.edu.au/. The source code and datasets of *ProsperousPlus* are freely available on the Download page of the webserver and GitHub repository (https://github.com/lifuyi774/ProsperousPlus). In addition, detailed user instructions are accessible on the Help page at http://prosperousplus.unimelb-biotools.cloud.edu.au/index.php/help.

REFERENCES

1. Lopez-Otin C, Matrisian LM. Emerging roles of proteases in tumour suppression. *Nat Rev Cancer* 2007;**7**:800–8.
2. Dixit VM. The road to death: caspases, cleavage, and pores. *Sci Adv* 2023;**9**:eadi2011.
3. Han N, Jin K, He K, *et al.* Protease-activated receptors in cancer: a systematic review. *Oncol Lett* 2011;**2**:599–608.
4. Chary A, Holodniy M. Recent advances in hepatitis C virus treatment: review of HCV protease inhibitor clinical trials. *Rev Recent Clin Trials* 2010;**5**:158–73.
5. Pang X, Xu W, Liu Y, *et al.* The research progress of SARS-CoV-2 main protease inhibitors from 2020 to 2022. *Eur J Med Chem* 2023;**257**:115491.
6. Peach CJ, Edgington-Mitchell LE, Bunnett NW, *et al.* Protease-activated receptors in health and disease. *Physiol Rev* 2023;**103**:717–85.
7. Turk B. Targeting proteases: successes, failures and future prospects. *Nat Rev Drug Discov* 2006;**5**:785–99.
8. Yau MK, Liu L, Fairlie DP. Toward drugs for protease-activated receptor 2 (PAR2). *J Med Chem* 2013;**56**:7477–97.
9. Li F, Wang Y, Li C, *et al.* Twenty years of bioinformatics research for protease-specific substrate and cleavage site prediction: a comprehensive revisit and benchmarking of existing methods. *Brief Bioinform* 2019;**20**:2150–66.
10. Wilkins MR, Gasteiger E, Bairoch A, *et al.* Protein identification and analysis tools in the ExPASy server. *Methods Mol Biol* 1999;**112**:531–52.
11. Boyd SE, Pike RN, Rudy GB, *et al.* PoPS: a computational tool for modeling and predicting protease specificity. *J Bioinform Comput Biol* 2005;**3**:551–85.

12. Verspurten J, Gevaert K, Declercq W, *et al.* SitePredicting the cleavage of proteinase substrates. *Trends Biochem Sci* 2009;**34**: 319–23.
13. Liu Z, Cao J, Gao X, *et al.* GPS-CCD: a novel computational program for the prediction of calpain cleavage sites. *PloS One* 2011;**6**:e19001.
14. Song J, Tan H, Shen H, *et al.* Cascleave: towards more accurate prediction of caspase substrate cleavage sites. *Bioinformatics* 2010;**26**:752–60.
15. Song J, Tan H, Perry AJ, *et al.* PROSPER: an integrated feature-based tool for predicting protease substrate cleavage sites. *PloS One* 2012;**7**:e50300.
16. Wang M, Zhao XM, Tan H, *et al.* Cascleave 2.0, a new approach for predicting caspase and granzyme cleavage targets. *Bioinformatics* 2014;**30**:71–80.
17. Song J, Wang Y, Li F, *et al.* iProt-sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Brief Bioinform* 2019;**20**:638–58.
18. Song J, Li F, Leier A, *et al.* PROSPERous: high-throughput prediction of substrate cleavage sites for 90 proteases with improved accuracy. *Bioinformatics* 2018;**34**:684–7.
19. Li F, Chen J, Leier A, *et al.* DeepCleave: a deep learning predictor for caspase and matrix metalloprotease substrates and cleavage sites. *Bioinformatics* 2020;**36**:1057–65.
20. Li F, Leier A, Liu Q, *et al.* Procleave: predicting protease-specific substrate cleavage sites by combining sequence and structural information. *Genomics Proteomics Bioinformatics* 2020;**18**:52–64.
21. Wang Y, Song J, Marquez-Lago TT, *et al.* Knowledge-transfer learning for prediction of matrix metalloprotease substrate-cleavage sites. *Sci Rep* 2017;**7**:5755.
22. Rawlings ND, Bateman A. How to use the MEROPS database and website to help understand peptidase specificity. *Protein Sci* 2021;**30**:83–92.
23. Li F, Li C, Marquez-Lago TT, *et al.* Quokka: a comprehensive tool for rapid and accurate prediction of kinase family-specific phosphorylation sites in the human proteome. *Bioinformatics* 2018;**34**:4223–31.
24. Gao J, Xu D. The Musite open-source framework for phosphorylation-site prediction. *BMC Bioinformatics* 2010; **11**(Suppl 12):S9.
25. Mei S, Li F, Xiang D, *et al.* Anthem: a user customised tool for fast and accurate prediction of binding between peptides and HLA class I molecules. *Brief Bioinform* 2021;**22**(5):bbaa415.
26. Chen Z, Zhao P, Li F, *et al.* iFeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 2018;**34**:2499–502.
27. Crooks GE, Hon G, Chandonia JM, *et al.* WebLogo: a sequence logo generator. *Genome Res* 2004;**14**:1188–90.
28. Nishida K, Frith MC, Nakai K. Pseudocounts for transcription factor binding sites. *Nucleic Acids Res* 2009;**37**:939–44.
29. Andreatta M, Alvarez B, Nielsen M. GibbsCluster: unsupervised clustering and alignment of peptide sequences. *Nucleic Acids Res* 2017;**45**:W458–63.
30. Liu G, Li D, Li Z, *et al.* PSSMHCpan: a novel PSSM-based software for predicting class I peptide-HLA binding affinity. *GigaScience* 2017;**6**:1–11.
31. Zhang H, Lund O, Nielsen M. The PickPocket method for predicting binding specificities for receptors based on receptor pocket similarities: application to MHC-peptide binding. *Bioinformatics* 2009;**25**:1293–9.
32. Mei S, Li F, Leier A, *et al.* A comprehensive review and performance evaluation of bioinformatics tools for HLA class I peptide-binding prediction. *Brief Bioinform* 2020;**21**:1119–35.
33. Thompson JD, Higgins DG, Gibson TJ. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Bioinformatics* 1994;**10**:19–29.
34. Consortium U. UniProt: the universal protein knowledgebase. *Nucleic Acids Res* 2018;**46**:2699.
35. Li M, Lu Z, Wu Y, *et al.* BACPI: a bi-directional attention neural network for compound–protein interaction and binding affinity prediction. *Bioinformatics* 2022;**38**:1995–2002.
36. Gfeller D, Guillaume P, Michaux J, *et al.* The length distribution and multiple specificity of naturally presented HLA-I ligands. *J Immunol* 2018;**201**:3705–16.
37. Bassani-Sternberg M, Chong C, Guillaume P, *et al.* Deciphering HLA-I motifs across HLA peptidomes improves neo-antigen predictions and identifies allostery regulating HLA specificity. *PLoS Comput Biol* 2017;**13**:e1005725.
38. Jurtz V, Paul S, Andreatta M, *et al.* NetMHCpan-4.0: improved peptide–MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. *J Immunol* 2017;**199**: 3360–8.
39. Rasmussen M, Fenoy E, Harndahl M, *et al.* Pan-specific prediction of peptide–MHC class I complex stability, a correlate of T cell immunogenicity. *J Immunol* 2016;**197**:1600582.
40. Hu Y, Wang Z, Hu H, *et al.* ACME: pan-specific peptide-MHC class I binding prediction through attention-based deep neural networks. *Bioinformatics* 2019;**35**:4946–54.
41. Reynisson B, Alvarez B, Paul S, *et al.* NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Res* 2020;**48**: W449–54.
42. Dorogush AV, Ershov V, Gulin A. CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363 2018.
43. Chen T, He T, Benesty M, *et al.* Xgboost: extreme gradient boosting. R package version 0.4-2 2015:1-4.
44. Ke G, Meng Q, Finley T, *et al.* Lightgbm: a highly efficient gradient boosting decision tree. *Adv Neural Inf Process Syst* 2017;**30**: 3146–54.
45. Webb GI, Boughton JR, Wang Z. Not so naive Bayes: aggregating one-dependence estimators. *Mach Learn* 2005;**58**:5–24.
46. Liu H, Setiono R. Incremental feature selection. *Appl Intell* 1998;**9**: 217–30.
47. Li F, Li C, Wang M, *et al.* GlycoMine: a machine learning-based approach for predicting N-, C-and O-linked glycosylation in the human proteome. *Bioinformatics* 2015;**31**:1411–9.
48. Li F, Li C, Revote J, *et al.* GlycoMine struct: a new bioinformatics tool for highly accurate mapping of the human N-linked and O-linked glycoproteomes by incorporating structural features. *Sci Rep* 2016;**6**:34595.
49. Li F, Guo X, Jin P, *et al.* Porpoise: a new approach for accurate prediction of RNA pseudouridine sites. *Brief Bioinform* 2021;**22**:bbab245.
50. Chen R, Li F, Guo X, *et al.* ATTIC is an integrated approach for predicting A-to-I RNA editing sites in three species. *Brief Bioinform* 2023;**24**(3):bbad170.
51. Li F, Chen J, Ge Z, *et al.* Computational prediction and interpretation of both general and specific types of promoters in Escherichia coli by exploiting a stacked ensemble-learning framework. *Brief Bioinform* 2021;**22**: 2126–40.
52. Jia C, Bi Y, Chen J, *et al.* PASSION: an ensemble neural network approach for identifying the binding sites of RBPs on circRNAs. *Bioinformatics* 2020;**36**:4276–82.

53. Wang M, Zhao XM, Takemoto K, *et al*. FunSAV: predicting the functional effect of single amino acid variants using a two-stage random forest model. *PloS One* 2012;**7**: e43847.
54. MJVD L, Polley EC, Hubbard AE. Super learner, statistical applications in genetics and molecular biology. 2007;6.
55. Zhou Z-H. *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.
56. Altman N, Krzywinski M. Ensemble methods: bagging and random forests. *Nat Methods* 2017;**14**:933–4.
57. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta Protein Struct* 1975;**405**:442–51.
58. Fawcett T. An introduction to ROC analysis. *Pattern Recogn Lett* 2006;**27**:861–74.
59. Hartigan JA, Wong MA. A K-means clustering algorithm. *J R Stat Soc Ser C Appl Stat* 2018;**28**:100–8.
60. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. *Adv Neural Inf Process Syst* 2017;**30**.
61. Bi Y, Li F, Guo X, *et al*. Clarion is a multi-label problem transformation method for identifying mRNA subcellular localizations. *Brief Bioinform* 2022;**23**(6):bbac467.
62. Wang R, Jiang Y, Jin J, *et al*. DeepBIO: an automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis. *Nucleic Acids Res* 2023;**51**:3017–29.
63. Wei L, He W, Malik A, *et al*. Computational prediction and interpretation of cell-specific replication origin sites from multiple eukaryotes by exploiting stacking framework. *Brief Bioinform* 2020;**22**(4):bbaa275.