Briefings in Bioinformatics, 2025, **26(2)**, bbaf189
https://doi.org/10.1093/bib/bbaf189
Problem Solving Protocol

MlyPredCSED: based on extreme point deviation compensated clustering combined with cross-scale convolutional neural networks to predict multiple lysine sites in human

Yun Zuo[1,*], Xingze Fang[1], Jiankang Chen[1], Jiayi Ji[1], Yuwen Li[1], Zeyu Wu[1], Xiangrong Liu[2], Xiangxiang Zeng[3], Zhaohong Deng[1,*], Hongwei Yin[4], Anjing Zhao[4,*]

[1]School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214000, China
[2]Department of Computer Science and Technology, National Institute for Data Science in Health and Medicine, Xiamen Key Laboratory of Intelligent Storage and Computing, Xiamen University, Xiamen 361005, China
[3]School of Information Science and Engineering, Hunan University, Changsha, China
[4]Department of Oncology, The First Affiliated Hospital of Naval Military Medical University, Shanghai 200000, China

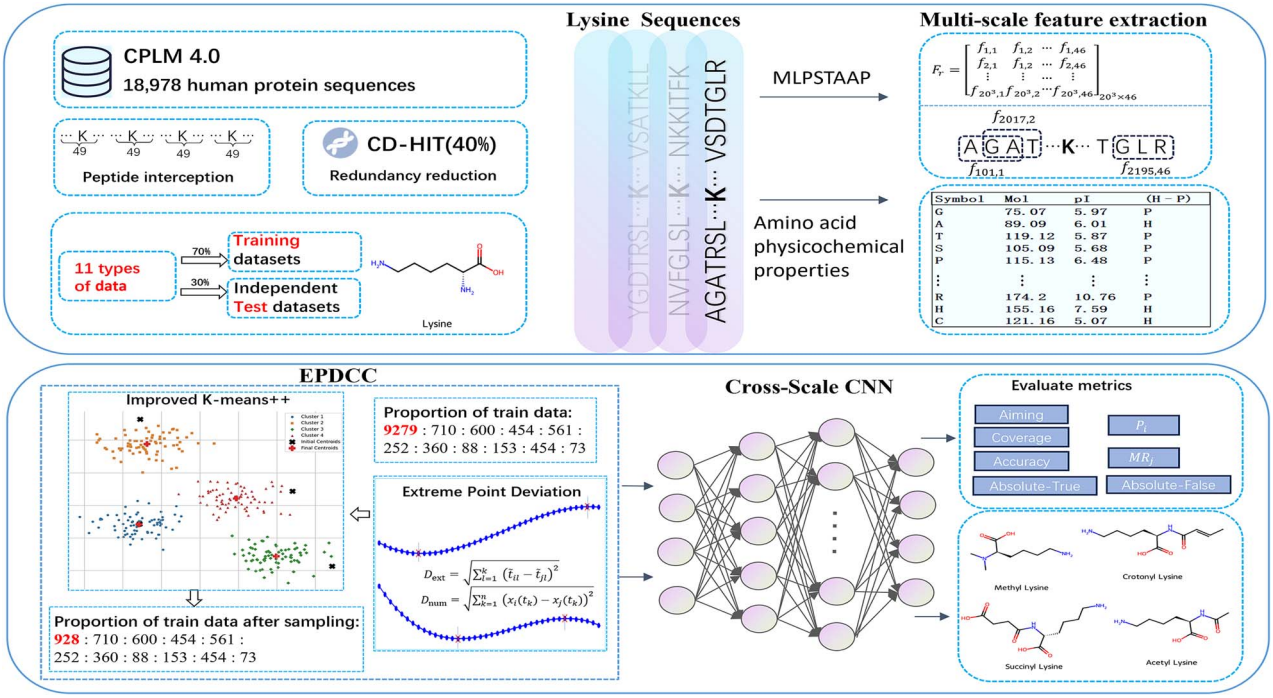
*Corresponding authors. Yun Zuo, E-mail: zuoyun@jiangnan.edu.cn; Zhaohong Deng, dengzhaohong@jiangnan.edu.cn; Anjing Zhao, dr_zhaoaj@smmu.edu.cn

Abstract

In post-translational modification, covalent bonds on lysine and attached chemical groups significantly change proteins' physical and chemical properties. They shape protein structures, enhance function and stability, and are vital for physiological processes, affecting health and disease through mechanisms like gene expression, signal transduction, protein degradation, and cell metabolism. Although lysine (K) modification sites are considered among the most common types of post-translational modifications in proteins, research on K-PTMs has largely overlooked the synergistic effects between different modifications and lacked the techniques to address the problem of sample imbalance. Based on this, the Extreme Point Deviation Compensated Clustering (EPDCC) Undersampling algorithm was proposed in this study and combined with Cross-Scale Convolutional Neural Networks (CSCNNs) to develop a novel computational tool, MlyPredCSED, for simultaneously predicting multiple lysine modification sites. MlyPredCSED employs Multi-Label Position-Specific Triad Amino Acid Propensity and the physicochemical properties of amino acids to enhance the richness of sequence information. To address the challenge of sample imbalance, the innovative EPDCC Undersampling technique was introduced to adjust the majority class samples. The model's training and testing phase relies on the advanced CSCNN framework. MlyPredCSED, through cross-validation and testing, outperformed existing models, especially in complex categories with multiple modification sites. This research not only provides an efficient method for the identification of lysine modification sites but also demonstrates its value in biological research and drug development. To facilitate efficient use of MlyPredCSED by researchers, we have specifically developed an accessible free web tool: http://www.mlypredcsed.com.

Received: September 26, 2024. **Revised:** March 27, 2025. **Accepted:** April 3, 2025

Graphical Abstract



Keywords: multiple lysine modification site prediction; extreme point deviation compensated clustering Undersampling; cross-scale convolutional neural networks; sequence analysis

Introduction

Once proteins are translated from RNA, they undergo a series of processes involving the binding of chemical small molecular groups, the process known as post-translational modifications (PTMs) [1]. These modifications can alter the structure, stability, function, and localization of proteins, thereby influencing numerous biological processes within cells. To date, hundreds of PTM types have been identified. These modifications not only play crucial regulatory roles within cells but also are closely associated with the development and progression of various diseases, such as cancer, neurodegenerative disorders, and cardiovascular diseases [2]. Therefore, studying PTMs is of significant importance for understanding cell biology and disease mechanisms [3, 4]. Lysine is one of the common amino acids in protein synthesis, with a side chain containing an amino group (−NH2). Lysine post-translational modifications involve the addition or removal of chemical groups on lysine residues through enzymatic reactions, which play a special role in drug development and disease prediction. Therefore, there is an urgent need for research methods to identify lysine post-translational modification sites. Lysine acetylation, crotonylation, methylation, and succinylation modifications play critical roles in many biological processes [5–8]. Traditionally, the prediction of these PTM sites often relied on biological experimental methods such as mass spectrometry analysis and phospho-specific antibodies. While these traditional approaches can provide accurate results, they also have significant drawbacks: the required equipment is expensive, the operations are complex, maintenance costs are high, and the experimental cycle is long, with a risk of sample contamination. With the advancement of technology and the emergence of a large number of protein sequences, these traditional methods no longer meet the need for quickly processing massive amounts of data. Therefore, there is an urgent need to develop a new method that is convenient, safe, fast, and reliable to overcome these challenges.

As the accumulation of experimentally verified PTM data continues, an increasing number of machine learning techniques have emerged to predict various post-translational modification sites. These modifications include methylation [9–17], succinylation [18–20], acetylation [21, 22], and phosphorylation [23–25]. However, existing methods primarily focus on identifying sites for a single type of modification. Studies have shown that there is a high degree of overlap between different types of modifications. For instance, some lysine sites are modified by both succinylation and acetylation [26]. Moreover, research has found that in *Corynebacterium glutamicum* and *Bacillus subtilis*, the overlap rates of these two modifications reach 40% and 35%, respectively [27–29]. To address this issue, researchers have proposed various computational methods. In 2016, Qiu *et al.* [30] introduced the first computational method, iPTM-mLys, for identifying four types of lysine PTM sites. In 2018, Hasan *et al.* [31] employed a different error-cost approach to develop a multilabel prediction model, mLysPTMpred, for predicting multiple lysine PTM sites. In 2021, Ahmed's research team [32, 33] launched two prediction tools: predML-Site and iMul-kSite, aimed at predicting multilabel K-PTM sites. Although these methods can identify multiple modification sites, they treat the four types of lysine PTMs separately and decompose the problem into four binary classification tasks, thereby ignoring the interactions among different lysine PTM sites. Based on this, in 2022, we proposed a predictive model for multiple lysine sites, MLysPRED, which is based on multiview clustering and multidimensional normal distribution resampling techniques [34]. However, the problem of extreme data imbalance still persists within lysine PTM site data, and the efficacy of the

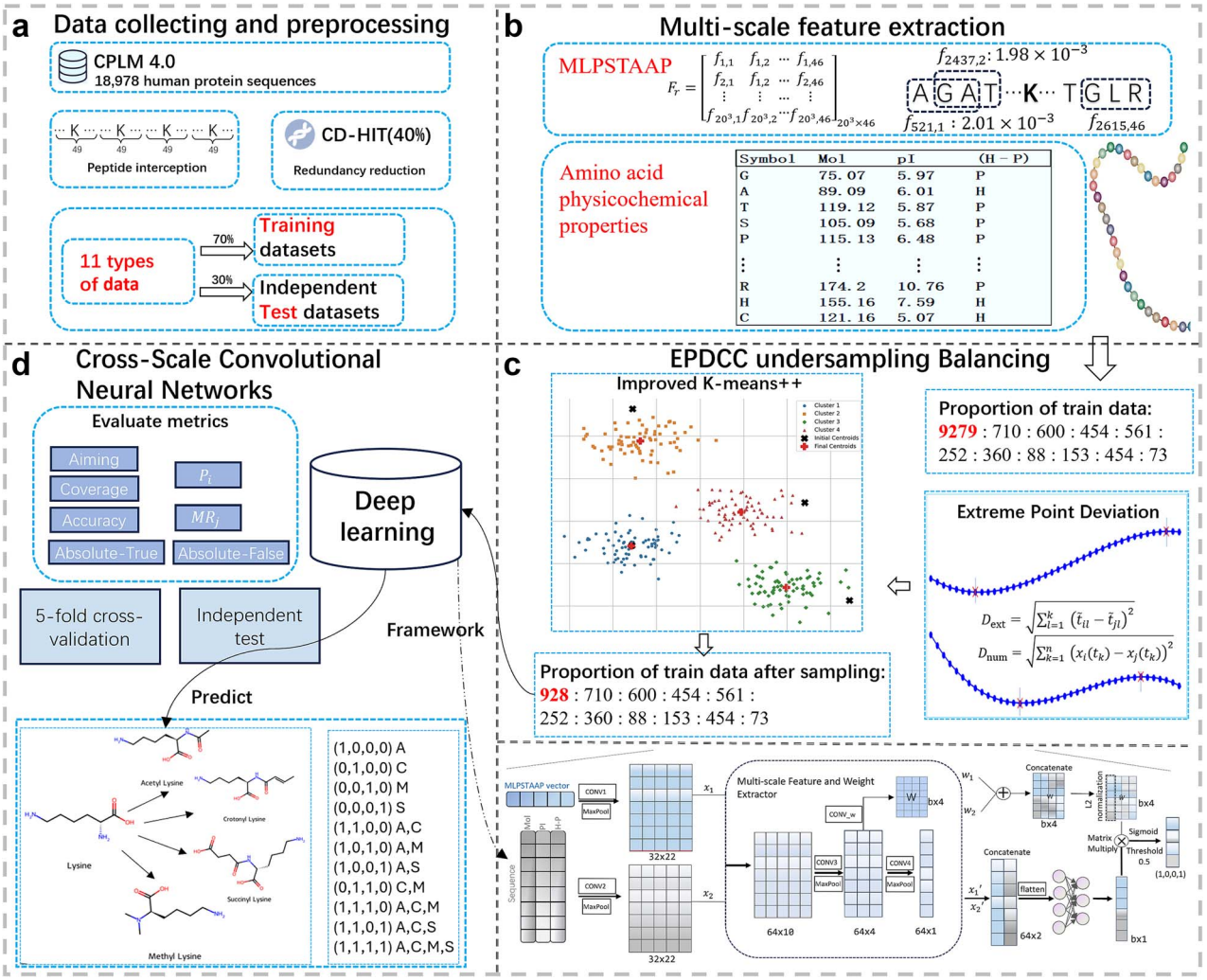


Figure 1. The flowchart of the overall framework of this study. (a) Data collection and preprocessing. (b) Multi-scale feature extraction. (c) Cross-Scale Convolutional Neural Networks. (d) EPDCC undersampling Balancing.

above models in handling such highly imbalanced data requires further improvement. Crosstalk in biology typically refers to the interaction and influence between different signaling pathways or molecular modifications, a phenomenon that is very common within cells, especially in PTMs [5]. Most previous research methods have focused on predicting only a single type of modification, whereas potential synergistic effects between different modifications, which may jointly regulate protein function, have often been overlooked. For example, the phosphorylation and ubiquitination modifications of certain proteins may collectively affect their degradation processes.

This study delves into the interactions between different types of modifications and successfully develops an advanced prediction tool, MlyPredCSED, capable of accurately predicting four types of lysine post-translational modification sites. Furthermore, in the face of the extreme sample imbalance challenge in the training data (a ratio exceeding 100:1), we employed a cluster-based undersampling algorithm with extreme value point deviation compensation to effectively balance the training dataset. To enhance the model's ability to interpret data, we also designed a cross-scale feature extraction method and constructed an end-to-end prediction model, MlyPredCSED. This model can not only capture the dependencies between sequences but also integrate features across different scales for prediction, with its architectural details shown in Fig. 1. The development process of the model includes several key steps:

(1) Data preparation phase: We carefully collected and filtered a large volume of protein sequence data and removed redundant information and types of modification sites with very low data volumes. Ultimately, a set of data with clear classification imbalance characteristics was obtained as the starting point of our research. The selection of acetyllysine, crotonyllysine, methyllysine, and succinyllysine as the four lysine modification sites was based on their relatively large sample sizes in the Compendium of Protein Lysine Modifications (CPLM) dataset, as well as their frequent use in previous studies, facilitating comparative analysis. The availability of sufficient data provides the model with rich feature information and a diverse sample distribution, contributing to improved prediction accuracy and generalization capability. Due to the limited availability of multilabel data (i.e. data containing two or more modifications simultaneously) for other lysine modifications, such as ubiquitination and sumoylation, which could result in insufficient model training, these modifications were not included in the present study.

(2) Feature enrichment processing: To enhance the semantic expressiveness of the data, we combined amino acid physicochemical properties with Multi-Label Position-Specific Tri-Amino Acid Propensity (MLPSTAAP) technology to conduct detailed feature encoding of lysine sequences, thereby extracting multidimensional feature information.

(3) Data balancing strategy: We adopted an undersampling method based on extreme value point deviation compensation, utilizing the optimized Kmeans++ algorithm to accurately calculate the cluster distribution of the majority class, effectively achieving undersampling adjustment of the dataset.
(4) Network mode learning and output: After processing the data through a multilabel multiscale learning network, an output matrix containing 4632 samples was generated, each sample corresponding to four label values. A threshold of 0.5 was set, categorizing label values below 0.5 as 0 and those equal to or above 0.5 as 1.

$$Pre(x) = \begin{cases} 1, Output(x) \geq 0.5 \\ 0, Output(x) < 0.5 \end{cases} \quad (1)$$

(5) Model optimization: The difference between the output matrix and the target matrix is evaluated by calculating the loss function. This difference is then used for backpropagation to precisely adjust the parameters in the convolutional neural network, thereby enhancing the overall performance of the model.

By adopting the advanced strategies mentioned above, this study successfully constructed the MlyPredCSED prediction model, effectively overcoming the issue of data imbalance. The model integrates a multiscale learning network, significantly enhancing the precision and stability of predictions. Moreover, by combining feature encoding with undersampling techniques, the model's generalization ability and robustness are further strengthened, ensuring accurate prediction of multiple lysine modification sites.

Materials and methods

Dataset description

To construct an efficient statistical prediction model, it is essential to establish a high-quality training and testing dataset [35–37]. This study constructs the dataset following the method used in previous research [34]. The human protein lysine sequences utilized in this research all originate from the CPLM 4.0 database (https://cplm.biocuckoo.cn/Browse.php) [38]. CPLM 4.0 is a comprehensive data resource focused on various PTMs on the amino side chain of protein lysine residues, integrating data from 10 public databases. All lysine modification sites in proteins were remapped to the UniProt database to eliminate redundant samples. In total, 18 978 human protein sequences were collected from the CPLM 4.0 database, including 11 863 acetyllysine sequences, 2151 crotonyllysine sequences, 3152 methyllysine sequences, and 1812 succinyllysine sequences. The specific steps for constructing the training and testing datasets are as follows:

Step 1: Peptide fragment extraction: From the collected 18 978 human protein sequences, extract peptide fragments that have been experimentally verified to be modified by acetyllysine, crotonyllysine, methyllysine, or succinyllysine, with a length of 49 centered on lysine (K).

Step 2: Data classification: The preprocessed lysine modification data were classified into a total of 15 categories. After excluding categories with fewer than 60 sequences, the dataset was ultimately divided into 11 categories:

$$\begin{cases} \phi_1 : accetyllysine \\ \phi_2 : crotonyllysine \\ \phi_3 : methyllysine \\ \phi_4 : succinyllysine \\ \phi_5 : accetyllysine + crotonyllysine \\ \phi_6 : accetyllysine + methyllysine \\ \phi_7 : accetyllysine + succinyllysine \\ \phi_8 : crotonyllysine + methyllysine \\ \phi_9 : accetyllysine + crotonyllysine + methyllysine \\ \phi_{10} : accetyllysine + crotonyllysine + succinyllysine \\ \phi_{11} : accetyllysine + crotonyllysine + methyllysine + succinyllysine \end{cases} \quad (2)$$

Step 3: Redundancy removal: After eliminating peptide fragments that were 100% identical, the following numbers of data points were obtained for each of the 11 categories: Category (1): 39 938; Category (2): 2463; Category (3): 3107; Category (4): 1019; Category (5): 4695; Category (6): 972; Category (7): 1220; Category (8): 138; Category (9): 277; Category (10): 1161; and Category (11): 125.

Step 4: For the resulting 11 categories, 70% of the data from each category were randomly selected as training data, while the remaining 30% were allocated as testing data. To avoid problems with sequence redundancy and homology affecting the accuracy of the multilabel prediction model evaluations, the Cluster Database at High Identity with Tolerance (CD-HIT) program [39] was employed with a threshold set at 0.4 to eliminate potential homologous sequences and redundant samples, thereby forming a de-duplicated dataset. The specific results of this segmentation are as follows:

Training Set: Category (1): 9279; Category (2): 710; Category (3): 600; Category (4): 454; Category (5): 561; Category (6): 252; Category (7): 360; Category (8): 88; Category (9): 153; Category (10): 454; Category (11): 73. Independent Test Set: Category (1): 4062; Category (2): 304; Category (3): 257; Category (4): 194; Category (5): 240; Category (6): 107; Category (7): 154; Category (8): 42; Category (9): 73; Category (10): 191; and Category (11): 36.

These steps ensured the comprehensiveness and accuracy of the data, providing a solid foundation for subsequent model training and validation. For each sequence, a multilabel annotation approach was employed, with the detailed methodology comprehensively described in the supplementary section titled "Multi-label Labeling Methodology."

Furthermore, to thoroughly demonstrate the exceptional performance of the model developed in this study, an additional dataset—the Qiu dataset—was incorporated. This dataset employs the same annotation methodology as described in detail in the supplementary titled "Introduction to the Qiu Dataset."

Cross-scale feature construction

In practical applications, multiscale data are prevalent and are considered as data describing the same sample from different perspectives. The core concept of this data analysis is to comprehensively understand the characteristics and behaviors of the research subject through different scales or data features. This approach allows researchers to capture complex relationships and potential patterns that a single data source might not reveal, thereby enhancing the depth and breadth of data analysis. In the field of bioinformatics, data features can be obtained from

multiple scales such as sequence, structure, dynamic properties, and physicochemical properties [40–42]. This not only enriches the dimensions of data analysis but also improves the accuracy of biological data analysis. Therefore, the prediction of lysine modification site categories in this study was performed using two scales: lysine sequence information and amino acid physicochemical properties.

Sequence-scale feature: Multilabel Position–Specific Tri-Amino Acid Propensity

According to our previous research [34], three sequence-scale encoding methods—Multi-Label Position-Specific Amino Acid Propensity (MLPSAAP), Multi-Label Position-Specific Di-Amino Acid Propensity (MLPSDAAP), and MLPSTAAP—were found to effectively extract hidden information from lysine sequences. Further experiments indicated that the MLPSTAAP encoding method has the highest compatibility when constructing predictive models; hence, the MLPSTAAP encoding method was chosen as the sequence-scale feature in this research. The feature extraction steps of the algorithm are as follows:

Step 1: The frequency matrix F_t for each tri-amino acid in the lysine sequences of class t was computed. The matrix, with a size of $20^3 \times 46$, represents the occurrence frequency of each tri-amino acid at every position within the sequences.

$$F_t = \begin{bmatrix} F_t(TAA_1/1) & F_t(TAA_1/2) & \cdots & F_t(TAA_1/46) \\ F_t(TAA_2/1) & F_t(TAA_2/2) & \cdots & F_t(TAA_2/46) \\ \vdots & \vdots & \cdots & \vdots \\ F_t(TAA_{20^3}/1) & F_t(TAA_{20^3}/2) & \cdots & F_t(TAA_{20^3}/46) \end{bmatrix}_{20^3 \times 46}, \quad t \in \{\phi_1, \ldots, \phi_{11}\} \tag{3}$$

In this matrix, $F_t(TAA_i/j)$ represents the frequency of the tri-amino acid TAA_i occurring at position j in the lysine sequences of class t. Here, TAA_i belongs to the set $\{AAA, AAC, AAD, \ldots YYY\}$, where $TAA_1 = AAA, TAA_2 = AAC, \cdots, TAA_{20^3} = YYY, i = 1, 2, 3, \ldots, 20^3, j = 1, 2, 3, \ldots, 46$.

Step 2: The frequency matrix FF_k was computed for each tri-amino acid occurring at each position in the lysine sequences of the other 10 classes, excluding class k. The matrix, with a size of $20^3 \times 46$, represents these occurrence frequencies.

$$FF_k = \begin{bmatrix} FF_k(TAA_1/1) & FF_k(TAA_1/2) & \ldots & FF_k(TAA_1/46) \\ FF_k(TAA_2/1) & FF_k(TAA_2/2) & \ldots & FF_k(TAA_2/46) \\ \vdots & \vdots & \cdots & \vdots \\ FF_k(TAA_{20^3}/1) & FF_k(TAA_{20^3}/2) & \ldots & FF_k(TAA_{20^3}/46) \end{bmatrix}_{20^3 \times 46}, \quad k \in \{\phi_1 \cup \phi_2 \cup \ldots \cup \phi_{k-1} \cup \phi_{k+1} \cup \ldots \cup \phi_{11}\} \tag{4}$$

In this matrix, $FF_k(TAA_i/j)$ represents the frequency of the tri-amino acid TAA_i occurring at position j in the lysine sequences of the other 10 classes, excluding class k. These classes are denoted as $(\varnothing_1 \cup \varnothing_2 \cup \cdots \cup \varnothing_{k-1} \cup \varnothing_{k+1} \cup \cdots \cup \varnothing_{11})$.

Step 3: The matrices FF and F were obtained by averaging the 11 matrices F_t and the 11 matrices FF_k, respectively. The difference between these matrices was then computed to obtain the matrix F_r, as follows:

$$F_r = F - FF = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,46} \\ f_{2,1} & f_{1,2} & \cdots & f_{2,46} \\ \vdots & \vdots & \cdots & \vdots \\ f_{20^3,1} & f_{20^3,2} & \cdots & f_{20^3,46} \end{bmatrix}_{20^3 \times 46} \tag{5}$$

in equation (5), $f_{i,j} = F_t(TAA_i/j) - FF_t(TAA_i/j)$, $F = (F_1 + F_2 + \cdots + F_{11})/11$, $FF = (FF_1 + FF_2 + \cdots + FF_{11})/11$.

Step 4: For each position in the sequence, feature encoding is performed based on the type of tri-amino acid found at that position in the F_r matrix. As illustrated in Fig. 1b, the numerical value corresponding to the tri-amino acid "AGA" at the first position is $f_{101,1}$; for "GAT" at the second position, it is $f_{2017,2}$; and for "GLR" at the last position, it is $f_{2195,46}$. This process continues until all 46 positions in a sequence have been successfully encoded, ultimately forming a 46-dimensional feature vector that encompasses both sequence order information and the frequency of amino acids between classes.

Amino acid physicochemical properties feature scale

According to Yao's research [43], the physicochemical properties of amino acids are crucial for protein structure. Molecular weight, isoelectric point, and hydrophobicity/hydrophilicity collectively influence the structure and function of proteins. For instance, larger amino acids like tryptophan and phenylalanine help in forming specific folds and domains, while high-molecular-weight amino acids can alter the spatial structure of proteins and their interactions, thereby adjusting their overall morphology. The isoelectric point is the pH value at which a protein has no net charge. Near the isoelectric point, the protein's solubility is lowest, making it prone to aggregation, which is crucial for purification and crystallization processes. Moreover, the hydrophobicity and hydrophilicity of amino acids are essential for the tertiary structure of proteins: hydrophobic amino acids are usually located internally, facilitating the spontaneous folding of proteins and enhancing stability; hydrophilic amino acids are mostly distributed on the surface, improving solubility and functionality through interactions with water. Therefore, these characteristics of amino acids play a decisive role in maintaining protein structure and function. Table 1 details the relevant physicochemical properties of the 20 natural amino acids. In analyzing each protein sequence, we encode hydrophilic amino acids from Table 1 as 1 and hydrophobic ones as −1. For instance, glycine (G) is encoded as (75.07, 5.97, −1) and alanine (A) as (89.09, 6.01, 1). Consequently, each protein sequence is transformed into a matrix of 49 rows and 3 columns, where each row represents the encoding of a specific amino acid.

Overview of imbalanced data issues

In classification task, the number of samples in different categories is often unequal, a phenomenon known as imbalanced data. This type of data is prevalent in various fields, including bioinformatics [44, 45]. In these datasets, the category with a larger number of samples is referred to as the majority class, whereas the class with fewer samples is known as the minority class. This imbalance can have several adverse effects on the predictive performance of deep learning models: firstly, the model may be biased toward the majority class because its samples have a greater weight in the loss function, leading to the optimization process focusing mainly on the accuracy of the majority class while neglecting the minority class [46]. Secondly, imbalanced data can affect gradient updates, causing the model to prioritize reducing errors for the majority class at a faster rate than for the minority class. This phenomenon has been observed in both multilayer perceptron and convolutional neural networks (CNNs) [47]. Moreover, imbalanced datasets can lead to poor model performance on the minority class, thereby reducing overall generalization capability. In some extreme cases, models may almost fail to correctly classify samples from the minority class. This issue is particularly pronounced in image classification and text classification tasks [47, 48]. Lastly, traditional evaluation metrics, such

Table 1. Physicochemical properties of the 20 natural amino acids.

Amino acid	Mol	PI	H-P
G	75.07	5.97	P
A	89.09	6.01	H
T	119.12	5.87	P
S	105.09	5.68	P
P	115.13	6.48	P
V	117.15	5.97	H
L	131.18	5.98	H
I	131.18	6.02	H
M	149.21	5.74	H
F	165.19	5.48	H
Y	181.19	5.66	H
W	204.23	5.89	H
D	133.10	2.77	P
E	147.13	3.22	P
N	132.12	5.41	P
Q	146.15	5.65	P
K	146.19	9.74	P
R	174.20	10.76	P
H	155.13	7.59	H
C	121.13	5.07	H

as accuracy, can be misleading when dealing with imbalanced datasets; therefore, it is necessary to employ evaluation metrics that are more suitable for imbalanced data.

A significant challenge faced by this research is the marked imbalance in the training dataset for lysine modification sites, which is characterized by a substantial disparity in the number of samples across different categories, with ratios of 9279:710:600:454:561:252:360:88:153:454:73. Based on prior research experience, employing resampling algorithms to adjust the number of samples in each category to achieve a balanced state has proven to be an effective strategy. Consequently, this study utilized the Extreme Point Deviation Compensated Clustering Undersampling (EPDCC) algorithm to balance the training dataset across these 11 categories, aiming to overcome the aforementioned issues. The framework details of the EPDCC undersampling algorithm will be elaborated upon.

The core concept of the EPDCC undersampling algorithm lies in adopting a distance-based metric function for clustering functional data. In the field of bioinformatics, clustering algorithms are often utilized to identify similarities among various biological data, thereby facilitating various predictions. However, most methods to data treat collected sample data as individual vectors and then perform clustering analysis based on these vectors. This approach, focusing solely on numeric clustering, tends to overlook the complex structure and potential dynamic property inherent in the data. Especially in the domain of bioinformatics, many datasets possess time-series characteristics or other types of functional properties, such as gene expression data, protein structure data, and metabolite concentration data. These datasets are not merely simple numerical vectors but rather curves or functions that contain rich information.

To address this issue, the study initially employs the EPDCC method to fit discrete data into a continuous function. Subsequently, this clustering algorithm is applied to undersample the training data. Specifically, samples near the cluster center are selected as new representatives for the majority class, replacing the original majority class samples, thereby achieving effective undersampling. The EPDCC undersampling algorithm primarily consists of two stages: function fitting and clustering analysis. The following sections will provide a detailed explanation of these two stages.

Function fitting

Step 1: Nondimensionalization process. To eliminate the magnitude differences between data, this study adopts normalization methods for nondimensionalization. This allows data of different dimensions to be compared on the same scale. This step is crucial for ensuring that variations in magnitude do not disproportionately influence the function-fitting process.

Step 2: Selection of basis functions for fitting. Function fitting reconstructs discrete data to obtain a continuous and smooth curve. That is, for a given samplei, the extracted discrete feature values $[y_{i1}, y_{i2}, \ldots, y_{i46}]$ are fitted to obtain the function curve $x_i(t)$, whose basic form is represented as shown in Equation (6):

$$y_{ij} = x_i(t_j) + \varepsilon(t_j) \tag{6}$$

In this context, t_j denotes the horizontal coordinate corresponding to the current location (In this study, 45 equally spaced points are used to partition the interval [0, 1], with each point's coordinate serving as the horizontal coordinate for the corresponding data point, where $j = 1, 2, \cdots, 46$, and $\varepsilon(t_j)$ represents the error term. To deeply explore the dynamic changes in data, B-spline basis functions are utilized as a tool during the data clustering analysis process, which are particularly suited to reveal the inherent patterns of nonperiodic data.

Step 3: Selection of data nodes. In the process of selecting nodes, we adopted two methods based on the distribution of data points: one is to choose at equal intervals, and the other is to adaptively select according to the density of data points. According to experimental results, this study employed an equidistant selection method to uniformly generate 10 nodes between the minimum and maximum values of the data's horizontal coordinates.

Step 4: Model construction. Subsequently, we constructed a k-order B-spline basis function as shown in Equation (7), proving a solid mathematical foundation for data analysis.

$$x_i(t) = \sum_{k=0}^{K} c_{ik}\phi_k(t) \quad , K = 3 \tag{7}$$

where c_{ik} represents the coefficients to be estimated, and $\phi_k(t)$ denotes the k-th order B-spline basis functions. To ensure the fitting function is both smooth and accuracy, it is crucial to choose an appropriate order. Based on the characteristics of the data and experimental results, an order of 3 was ultimately selected for the base function in this study.

Step 5: The parameters c_{ik} are estimated by minimizing the objective function as shown in Equation (8) using the least squares method:

$$\min_{c_{ik}} \sum_{j=1}^{T} \left(y_{ij} - \sum_{k=0}^{K} c_{ik}\phi_k(t_j) \right)^2 + \lambda \int \left(x_i^{(p)}(t) \right)^2 dt \tag{8}$$

where λ is the smoothing factor, and $x_i^{(p)}(t)$ represents the p-th derivative of $x_i(t)$. By solving the above optimization problem, the parameters c_{ik} can be obtained, which allows for the derivation of the fitted function curve $x_i(t)$.

Clustering analysis

In exploring the similarity measurement of curve shapes, we typically rely on the positions of extreme points to evaluate curve shapes. Penalties are imposed by calculating the deviations between these extreme points to measure their similarity. If the extreme points of two curves are very close, their distance will naturally be smaller; conversely, if the positions of the extreme points differ significantly, the calculated distance will correspondingly increase. However, relying solely on the deviation distance between extrema points does not fully reflect the overall differences between curves. This method primarily focuses on local shape comparisons and fails to accurately capture the horizontal differences across the entire curve. On the other hand, purely numerical distance–based measurement methods, while capable of reflecting the absolute level differences of curves, neglect the differences in curve shapes. In light of this, the present study proposes an innovative approach that integrates a penalty mechanism for extremum point deviation into numerical distance measurement, resulting in a new similarity measurement method based on the compensation for deviation at curve extremum points. The essence of this method lies in incorporating the distance of extremum point deviation as a compensatory factor into the overall distance calculation between two curves, thereby adjusting the impact caused by extremum point deviation. The specific steps are as follows:

Initially, the first derivatives of functions $x_i(t)$ and $x_j(t)$ are calculated to identify their extremum points, which are those where the derivatives equal zero. Let these extremum points form sets $E_i = \{t_{i1}, t_{i2}, \ldots, t_{im}\}$ and $E_j = \{t_{j1}, t_{j2}, \ldots, t_{jn}\}$, respectively. Next, interpolation is applied to the set of extreme points. If the number of elements in E_i and E_j differs, adjust through linear interpolation to ensure both have the same number of elements, resulting in new sets of extreme points denoted as $\widetilde{E_i} = \{\tilde{t}_{i1}, \tilde{t}_{i2}, \ldots, \tilde{t}_{ik}\}$ and $\widetilde{E_j} = \{\tilde{t}_{j1}, \tilde{t}_{j2}, \ldots, \tilde{t}_{jk}\}$. Subsequently, the deviation distance D_{ext} between the extreme points is calculated according to Equation (9) defined as follows:

$$D_{\text{ext}} = \sqrt{\sum_{l=1}^{k} \left(\tilde{t}_{il} - \tilde{t}_{jl}\right)^2} \tag{9}$$

Additionally, the Euclidean distance between the two functions $x_i(t)$ and $x_j(t)$ needs to be calculated using the formula shown in Equation (10):

$$D_{\text{num}} = \sqrt{\sum_{k=1}^{n} \left(x_i\left(t_k\right) - x_j\left(t_k\right)\right)^2} \tag{10}$$

Finally, by combining the numerical distance and the deviation distance of extreme points, the total distance D_{total} is calculated, which serves as the final similarity measurement criterion.

$$D_{\text{total}} = D_{\text{num}} + D_{\text{ext}} \tag{11}$$

This measurement method significantly enhances the accuracy and reliability of similarity assessments by thoroughly analyzing differences in function values and integrating deviations in extreme point positions.

This study employed an advanced *K*-means++ clustering analysis technique for data grouping. The basic idea of this technique is to first calculate the total distance D_{total} between all data points and then select two data points that are farthest apart as the initial cluster centers. Subsequently, after determining n cluster centers, the selection of the $(n+1)$th center involves choosing the data point with the maximum total distance from the existing n centers. Through this recursive approach, K initial cluster centers are ultimately determined. Compared to the traditional *K*-means++ method, which randomly chooses a starting point, this improved method completely eliminates randomness and significantly enhances clustering efficiency. Moreover, the samples selected after undersampling can better represent the characteristics of the original dataset. The pseudocode for the EPDCC undersampling algorithm is presented in Table 2.

Multilabel cross-scale convolutional neural network learning architecture

The multilabel cross-scale convolutional neural network (CSCNN) learning architecture proposed in this study is depicted in Fig. 2. This model is capable of fully leveraging the potential interaction relationships among PTM sites. At the same time, it utilizes a CNN to extract the implicit relationships among labels. By fusing features and weights at different scales, it enriches the dimensions of data analysis, thereby achieving superior classification performance. This learning architecture mainly consists of the following components:

(1) We have developed an advanced multisource feature alignment technique, as shown in Fig. 2a. This technique ensures the dimensional consistency between MLPSTAAP features and the physicochemical properties of amino acids. As a result, it generates more comprehensive and enriched feature representations, thereby facilitating a better understanding of the input data.

(2) A cross-scale feature and weight extractor was designed, as illustrated in Fig. 2b, to progressively delve into the intricate and complex information of deep features. During this process, it generates corresponding feature weights. By deriving weights from intermediate features, the risk of overfitting is effectively mitigated. This system consists of two CNN modules with identical kernel sizes. It is designed to comprehend features under varying contextual dependencies, fully integrating information from different positions, in order to accurately predict interactions.

(3) An Adaptive Normalization Weights mechanism was designed, as shown in Fig. 2c. During the feature fusion and prediction phase, the model effectively integrates latent information from different scales and combines it with the obtained weights to obtain the prediction results. This method prevents labels with larger numerical ranges from dominating the training process, thereby achieving a balanced classifier output across different labels. After normalization, all features are processed through a weighted calculation to produce the network output. A Sigmoid activation function is applied to the network output, with a threshold set at 0.5 to predict lysine post-translational modification sites. This process reduces bias toward majority class samples and can visually display the predicted scores for different modification sites.

The predictive model MlyPredCSED developed in this study not only enhances classification performance but also deepens the understanding of the relationship between lysine post-translational modification sites and sequences as well as amino acid physicochemical properties, offering new perspectives and methods for further research.

Table 2. The pseudo-code of EPDCC undersampling.

Algorithm 1: EPDCC Undersampling

Input: The original data of the first class after MLPSDAAP feature extraction: $x_1, x_2, x_3, \cdots, x_{9279}$.
The number of clusters: k.
The number of samples: n.
Output: The n samples after sampling.
1. Normalize $x_1, x_2, x_3, \cdots, x_{9279}$ to obtain *X-normalized*.
2. For EACH x_i **in** *X-normalized*:
Fit x_i using a cubic B-spline function to f_i.
Store f_i in *functions*.
END FOR
3. Calculate the D_{total} distance between all functions in the *functions*.
4. Add the two functions with the largest D_{total} distance to the initial *clustering centers*.
5. WHILE the number of clusters $< k$:
For EACH function f_i **not in** *clustering centers*:
Calculate the sum of D_{total} distances from f_i to all current *clustering centers*.
END FOR
Select the function f_{max} with the maximum distance sum and add it to *clustering centers*.
END WHILE
6. For EACH f_i **in** *functions*:
Calculate the distance from f_i to the clustering centers and assign it to the nearest cluster.
Record the distances to the nearest centroids as *distances-to-centroid*.
END FOR
7. Sort the samples in ascending order based on *distances-to-centroid*.
8. RETURN the top n samples from the sorted list.

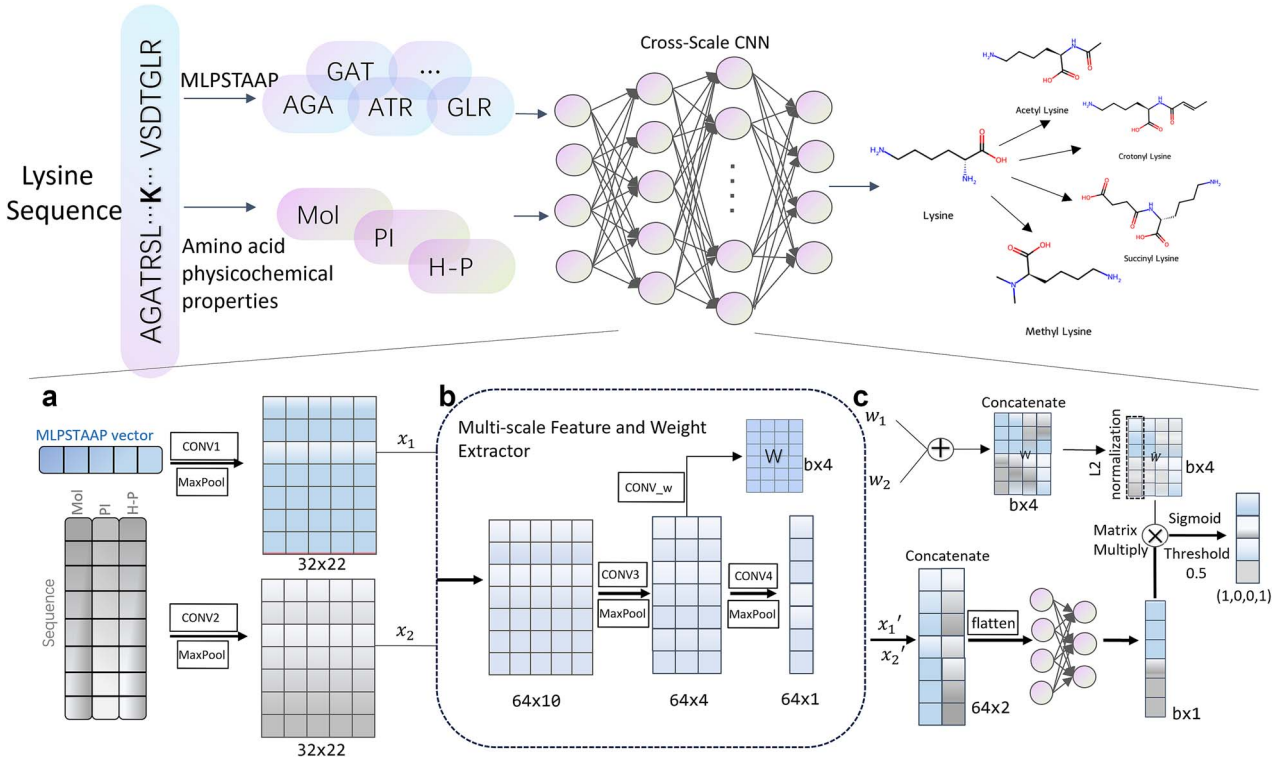


Figure 2. The network architecture of the CSCNN. (a) Multisource feature alignment technique for ensuring dimensional consistency between MLPSTAAP features and amino acid physicochemical properties. (b) Cross-scale feature and weight extractor using CNN modules to enhance feature representation and mitigate overfitting. (c) Adaptive Normalization Weights mechanism for balanced prediction and improved model performance.

Adaptive Normalization Weights

In addressing the issue of data imbalance, models tend to favor classes with larger sample sizes. According to related research [49, 50], we have discovered that the norm distribution in the final layer of classifiers may not be uniform, especially showing that the prediction norms for classes with more samples significantly exceed those with fewer samples. To address this challenge, our study introduces an adaptive normalization weight mechanism aimed at reducing the model's overreliance on categories with larger sample sizes, thereby enhancing overall classification performance.

In form, the classifier projects each sample feature x from a D-dimensional vector to a d-dimensional vector x^d and multiplies it with an $L \times d$ weight matrix W. Through matrix operations $W \times x^d$,

the final predicted label result of $L \times 1$ is obtained. Based on this process, an adaptive normalization weight method is proposed in this study. In the final layer of the model, a d-dimensional vector and an $L \times d$ weight matrix W are outputted. Subsequently, this weight matrix is normalized using the L_2 norm, ensuring that the sum of the squares of each vector element equals 1:

$$\overline{W_l} = \frac{w_l}{\|w_l\|_2}, \forall l \in L \tag{12}$$

In this formula, w_l denotes the l-th row vector of W. Subsequently, the normalized weight matrix $\overline{W} = [\overline{w_1}; \overline{w_2}; \ldots; \overline{w_l};]^T$ is used to replace the original weight matrix W and perform operations with x^d to obtain the final prediction results.

The weight matrix W can be dynamically adjusted during the model training process to ensure that the weight scales of all predicted labels are uniform. This approach prevents labels with larger numerical ranges from dominating the training, achieving a balanced classifier output across different labels. It also helps mitigate the model's preference for categories with a larger sample size, promoting an overall performance enhancement.

Evaluation metrics

Based on Chou's research [51], this study employs five fundamental metrics for multilabel classification tasks to evaluate the performance of different methods: Aiming, Coverage, Accuracy, Absolute-True, and Absolute-False. The definitions of these metrics are as follows:

$$Aiming = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|\ Y_i \cap Y_i^*\ |}{|\ Y_i^*\ |}\right) \tag{13}$$

$$Coverage = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|\ Y_i \cap Y_i^*\ |}{|\ Y_i\ |}\right) \tag{14}$$

$$\mathbf{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|\ Y_i \cap Y_i^*\ |}{|\ Y_i \cup Y_i^*\ |}\right) \tag{15}$$

$$\mathbf{Absolute-True} = \frac{1}{n}\sum_{i=1}^{n}\Delta\left(Y_i, Y_i^*\right) \tag{16}$$

$$\mathbf{Absolute-False} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|\ Y_i \cup Y_i^*\ | - |\ Y_i \cap Y_i^*\ |}{M}\right) \tag{17}$$

In the given dataset, n represents the total number of samples. For each individual sample$i$, Y_i is the actual label of that sample, while Y_i^* is the corresponding predicted label. Here, the $\cap$ symbol performs an intersection operation, while the $\cup$ symbol performs a union operation. When we refer to $|Y_i|$, we mean the number of elements in Y_i that have a value of 1; for example, $|(1,0,1,0)|$ equals 2. Additionally, the function $\Delta(a,b)$ indicates whether a and b are identical; it returns 1 if they are identical and 0 otherwise. M represents the total number of labels in the system.

During the data analysis process, we observed that the sample sizes of certain categories significantly outnumbered others. In this scenario, if a model were to predict all samples as these dominant categories, it could still achieve high results on traditional evaluation metrics. Therefore, to more accurately evaluate our predictive model's generalization ability across different categories, especially those with fewer samples, this study introduces a new evaluation metric—the absolute accuracy p_i for each category i. The calculation method for this metric is as follows:

$$p_i = \frac{c_i}{n_i} \tag{18}$$

In the sample set being discussed, n represents the total number of all samples. For any given category$i$, c_i is the number of correctly predicted samples in that category$i$, and n_i represents the total number of samples in category i. The label vector associated with the current category is denoted as $Label_i$. Specifically, the value of c_i is obtained by accumulating all samples j from 1 to n, where each sample's calculation is based on the comparison results between $\Delta\left(Y_j, Y_j^*\right)$ and $\Delta\left(Y_j, Label_i\right)$, namely, $c_i = \sum_{j=1}^{n}\Delta\left(\Delta\left(Y_j, Y_j^*\right), \Delta\left(Y_j, Label_i\right)\right)$. Similarly, the calculation of n_i also involves applying $\Delta\left(Y_j, Label_i\right)$ to all samples j and summing up the results. The function $\Delta(a,b)$ is used to determine if a and b are completely identical, returning 1 if they are and 0 otherwise. This procedure applies to all 11 categories, that is, for $i \in \{1, 2, \cdots, 11\}$.

In multilabel tasks, a model's effectiveness should not be solely judged on its ability to accurately predict all labels. In fact, if a model can effectively identify multiple related labels, even if not all labels are correct, the model should still be considered effective. Therefore, evaluating the model solely based on absolute accuracy may be unfair. Given this, the study introduces a new evaluation metric—MR_j (Match Rate), to more comprehensively measure the performance of multilabel classification systems.

$$MR_j = \frac{\sum_{k=j}^{max} P_k}{\sum_{k=j}^{max} C_k} \tag{19}$$

At the computation level of k, P_k represents the total number of samples where the actual and predicted labels match (considering only labels with a value of 1, ignoring labels with a value of 0) exactly k times. Specifically, it counts the number of samples where the number of label 1 in the intersection of actual and predicted labels is exactly k. This is expressed mathematically as $P_k = \sum_{j=1}^{n}\Delta\left(\left|Y_j \cap Y_j^*\right|, k\right)$. For example, for the intersection (1,1,1,0) ∩ (1,0,1,0), when k equals 2, the value of the Δ function is 1, namely, $\Delta\left(|(1,1,1,0) \cap (1,0,1,0)|, 2\right) = 1$. On the other hand, C_k refers to the total number of samples in the true labels that contain exactly k labels marked as 1, which can be expressed by the formula $C_k = \sum_{j=1}^{n}\Delta\left(\left|Y_j\right|, k\right)$. For instance, for the set (1,1,1,1), when k equals 4, the value of the Δ function is 1, namely, $\Delta\left(|(1,1,1,1)|, 4\right) = 1$. In this study, **max** is defined as the maximum number of actual labels marked as 1 in the dataset, with a value of 4.

For instance, when $k = 3$, the numerator $P_3 + P_4$ represents the total number of samples where the predicted labels match the actual labels at least 3. Meanwhile, the denominator $C_3 + C_4$ indicates the total number of samples where the true labels contain at least three labels with a value of 1. Therefore, the calculation formula for MR_3 can be derived as $MR_3 = \frac{P_3+P_4}{C_3+C_4}$.

Results and discussion

Feature ablation

This study focuses on exploring the impact of different feature dimensions on the performance of prediction models. Specifically, we compared the effects of using only sequence-scale features (MLPSTAAP), relying solely on the amino acid physicochemical property features, and a fusion of both feature types on the accuracy of model predictions. To ensure fairness in the experiment, we set other feature values to zero when testing a single feature set. The results of the comparative analysis are detailed in Fig. 3. The experimental results indicate that the model integrating

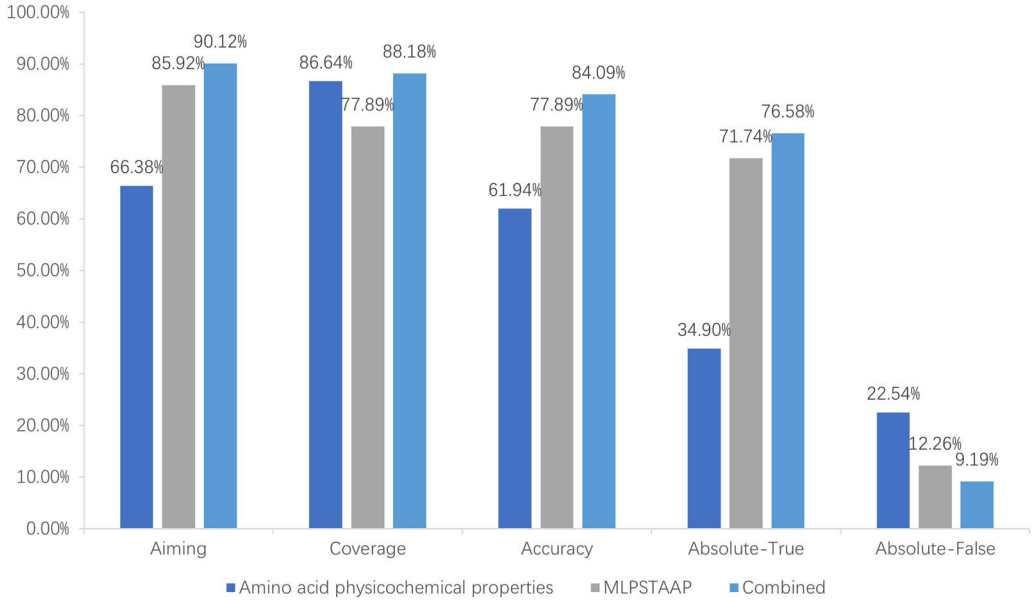


Figure 3. Comparison of performance metrics for different feature scales on independent test set.

multiview data performed significantly better than single-feature models in terms of Aiming, Coverage, Accuracy, Absolute-True, and Absolute-False, with respective values of 90.12%, 88.18%, 84.09%, 76.58%, and 9.19% on the independent test set. Notably, compared to models that rely solely on the physicochemical properties of amino acids, this comprehensive model has seen at least a 20% improvement in Aiming, Accuracy, and Absolute-True, with a slight increase of ~2% in coverage and a reduction of over 10 percentage points in Absolute-False. These results fully demonstrate the importance of integrating multiview features for enhancing model prediction accuracy. This study further confirmed the significant advantages of the constructed multiview features through feature ablation experiments. These features not only enrich the dimensions of semantic information but also enable the model to effectively learn the interdependencies between different perspectives, thereby significantly improving overall performance.

Determining the number of clusters for undersampling

Cluster-based undersampling is an effective strategy for addressing data imbalance by reducing the number of majority class samples, but its success largely depends on the reasonable selection of the number of clusters.

When the number of clusters is insufficient, samples from different categories may be grouped into the same cluster, leading to excessive heterogeneity within the cluster. This can cause the selected representative samples to fail to fully reflect the diversity of each category, limiting the model's comprehensive learning of the characteristics of most categories. Additionally, fewer clusters mean more samples are selected from each cluster, resulting in an insufficient undersampling effect and failing to effectively reduce the number of majority class samples, thus still causing a significant imbalance in the dataset. Conversely, when the number of clusters is excessive, the number of samples within each cluster significantly decreases, potentially leading to oversegmentation where a single sample might form a cluster. This can cause the selected samples to be too dispersed, making it difficult to fully reflect the characteristics of the majority categories during model training. Furthermore, an excessive number of clusters increases computational costs, especially with large datasets, and can lead to insufficient sample representativeness during resampling, thereby affecting the model's generalization performance.

In this study, we focused on reducing the number of samples in the majority class while preserving their representativeness. To achieve this goal, we compared the test results under different cluster numbers k on the independent test set. The comparison results are presented in Fig. 4. Observing Fig. 4 reveals that when the cluster number is set to 4, the constructed dataset's predictive performance reaches its optimal level. Specifically, the Aiming, Coverage, Accuracy, Absolute-True, and Absolute-False values at this point are 90.12%, 88.18%, 84.09%, 76.58%, and 0.0919, respectively. Compared to the cluster number with the worst predictive performance ($k = 15$), the Aiming, Coverage, Accuracy, and Absolute-True values are improved by 18.23%, 3.87%, 17.4%, and 32.06%, respectively, while the Absolute-False value decreased by 9.63%. Based on these comparative results, it can be concluded that the prediction effect is indeed influenced by the number of clusters. Therefore, in order to achieve optimal predictive performance, a clustering number of 4 was selected for undersampling of the majority class data.

Influence of the undersampling ratio

In this study, we explored a dataset that encompasses 11 distinct categories with extremely uneven data distribution, specifically in the ratio of 9279:710:600:454:561:252:360:88:153:454:73. It is clear that the first category significantly leads in volume, accounting for over 71% of the total sample size. Therefore, the focus of this research was to implement undersampling strategies in this primary category. The undersampling ratio significantly impacts data representation and model training effectiveness. A low ratio can lead to a substantial loss of original samples, inaccurately reflecting the original data distribution, introducing prediction bias, and hindering model learning, causing under-fitting. Conversely, a high ratio better retains majority class information,

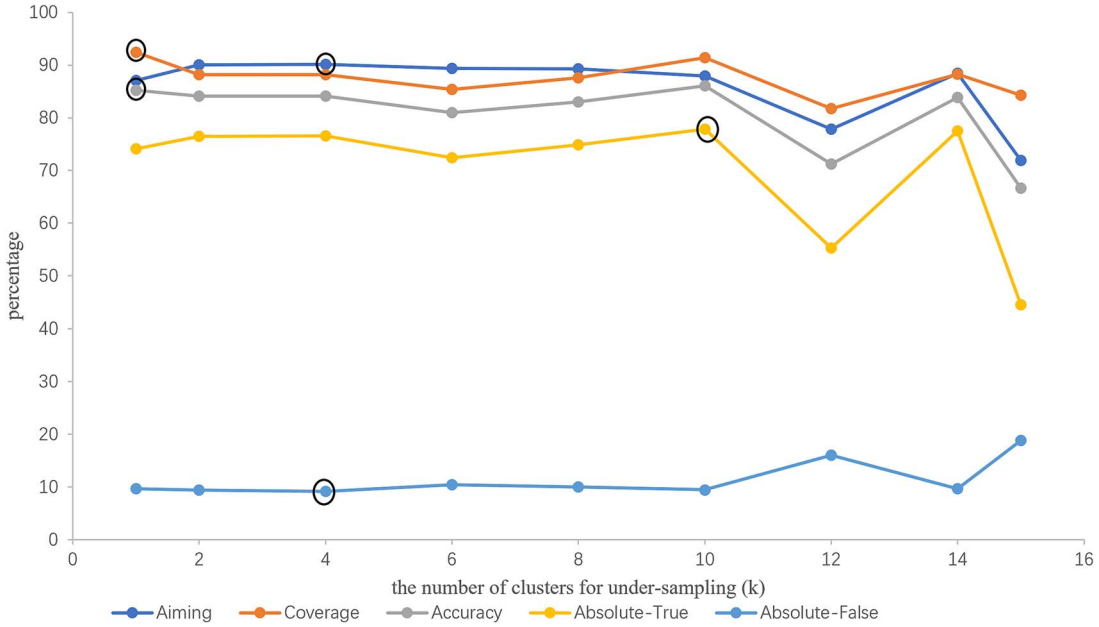


Figure 4. Effect of different number of clusters (k) on independent test set metrics.

Table 3. Absolute accuracy p_i for 11 different categories on the independent test set under various undersampling ratios (%).

Ratios	A	C	M	S	AC	AM	AS	CM	ACM	ACS	ACMS
0.5	99.58	23.03	26.85	25.77	7.92	16.82	55.19	0.00	22.22	21.65	11.11
0.4	99.53	28.95	30.85	32.99	13.33	19.63	44.81	11.90	86.11	29.90	2.78
0.3	98.47	0.00	0.00	0.00	29.58	80.37	41.56	0.00	43.06	19.07	2.78
0.2	97.19	0.00	0.00	0.00	20.00	19.63	62.34	0.00	66.67	5.15	41.67
0.1	97.91	**37.50**	**31.91**	20.10	27.92	**50.47**	33.12	7.14	52.78	11.34	13.89
0.05	17.58	11.84	17.90	14.95	37.50	2.80	11.04	0.00	0.00	23.20	0.00

In the table, A, C, M, and S represent four different types of lysine post-translational modifications. Specifically, A stands for acetyllysine, C for crotonyllysine, M for methyllysine, and S for succinyllysine. Bold and underlined values indicate the highest accuracy for each category across different undersampling ratios.

enhancing data expressiveness. However, due to extreme category imbalance, a high ratio may still result in overfitting to majority classes, making undersampling less effective [52].Thus, selecting an appropriate intermediate undersampling ratio is crucial. It ensures the dataset retains enough information postsampling while alleviating class imbalance and preventing model bias toward a single class, maintaining generalization capability. Table 3 provides a detailed presentation of the absolute accuracy p_i for 11 different categories on the independent test set under various undersampling ratios.

In the data analysis of Table 3, it is clear that as the undersampling ratio decreases, the model's predictive capability for samples containing only acetyllysine modifications tends to decline within a certain range. When the undersampling rate is set to 0.1, the model's generalization performance is significantly better than other ratios when dealing with sample types containing only crotonyllysine, methyllysine, and both acetyllysine and methyllysine modifications. Moreover, at an undersampling rate of 0.1, the model exhibits good generalization ability in all classes, effectively preventing the creation of a model that only generalizes well for a few classes. Therefore, a final undersampling rate of 0.1 was chosen in this study. This rate ensures that the undersampled dataset maintains good representation after undersampling but also mitigates the issue of class imbalance, avoiding excessive bias toward any particular class, thereby ensuring the accuracy and reliability of the prediction results.

Selection of data fitting functions

In the exploration of undersampling algorithms for discrete data fitting, selecting an appropriate fitting function is critical for enhancing the effectiveness of the algorithm. This study provides a detailed analysis of two main fitting techniques: B-spline fitting and polynomial fitting. The B-spline method, as a commonly used nonparametric fitting method, is particularly suitable for smooth and nonperiodic data; whereas polynomial fitting, as a traditional parametric fitting technique, approximates data points through polynomial functions.

The extremum point deviation compensation clustering undersampling algorithm employed in this study calculates the similarity between samples based on the deviation of the function's extremum points after different sample fittings. Therefore, the closer the extremum points are, the smaller the distance between the samples. For a deeper analysis, we randomly selected three samples from the dataset: Sample 1 and Sample 2 belong to the second category, while Sample 3 belongs to the third category. As shown in Fig. 5, a comparison was made (a) the fitting effect of B-spline functions on data from the same category, (b) the fitting effect of B-spline functions on data from different categories, (c) the fitting effect of polynomial functions on data from the same category, and (d) the fitting effect of polynomial functions on data from different categories.

As shown in Fig. 5, when fitting samples of the same category, the extremum points of the B-spline function are closer together;

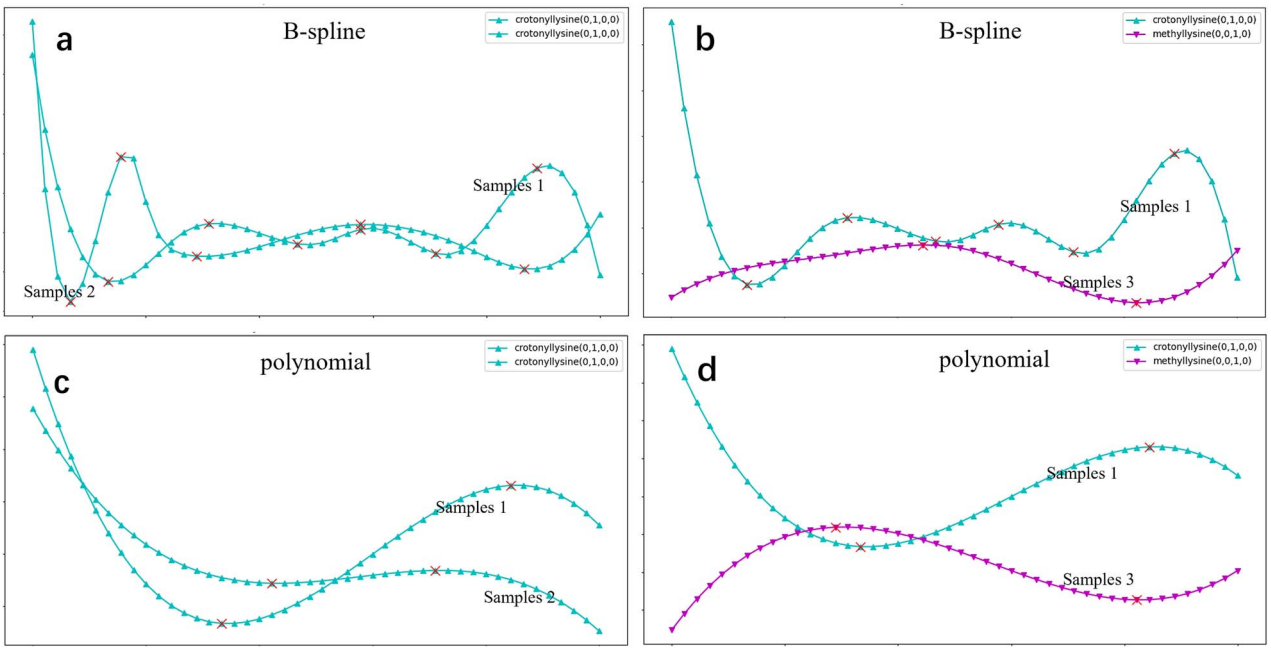


Figure 5. The fitting effect of different functions on the randomly selected three samples.

whereas the extremum points of polynomial fitting are further apart, indicating larger differences. In processing samples from different categories, the extremum points of B-spline functions are further apart, displaying greater differences; in contrast, the extremum points of polynomial fitting are relatively closer, revealing smaller differences. Given that the clustering algorithm used in this study requires a clear distinction between data from different categories, the B-spline function was chosen for sample fitting.

Furthermore, to deeply explore the differences in data fitting effects between B-spline functions and polynomial functions, a comparative study was conducted. Experimental results on an independent test dataset, as illustrated in Fig. 6, indicate that when using B-spline functions for data fitting, their performance in terms of Aiming, Coverage, Accuracy, and Absolute-True metrics is superior to that of polynomial functions, with improvements of 3%, 6%, 7%, and 7%, respectively. However, in terms of measuring the Absolute-False metric, the performance of B-spline function fitting is 2% lower than that of polynomial function fitting, further confirming the superiority of B-spline functions in data fitting.

Comparison of clustering distance measurement methods

While exploring the effects of various similarity measurement methods on clustering results, observed that these results could manifest significant differences. This study conducted an in-depth comparison of five distinct distance measurement techniques: extremum point deviation compensation, Minkowski distance, a method based on the expansion coefficients of basis functions, traditional Euclidean distance, and traditional Mahalanobis distance. These measurement methods not only encompass the realm of function fitting but are also applicable to strategies for processing discrete data.

The Minkowski distance is a widely used method for measuring the differences between functions. It calculates the distance between two functions by directly comparing their values across the entire domain of definition. For two functions f and g defined on the interval$[a, b]$, the formula for calculating the Minkowski distance is as follows:

$$D(f,g) = \left(\int_a^b |f(x) - g(x)|^p dx \right)^{1/p} \tag{20}$$

where p is a positive integer or a positive real number.

The method of distance measurement based on the expansion coefficients of basic functions involves expressing a function as a linear combination of a set of basic functions and evaluating the similarity or difference between functions based on these expansion coefficients. This technique transforms the challenge of comparing functions into a comparison in the coefficient space, making it particularly suitable for measuring complex functions. The specific definition is as follows:

$$D(f,g) = \left(\sum_{i=1}^{m} (c_i - d_i)^2 \right)^{1/2} \tag{21}$$

where c_i and d_i are the expansion coefficients for functions f and g, respectively.

Additionally, our study also delved into two nonparametric methods of distance calculation: the traditional Euclidean distance and Mahalanobis distance. Euclidean distance, as a widely recognized metric, primarily assesses the distance between data points by calculating the length of the direct line connecting them. This method is popular for its simplicity and intuitiveness. However, it is sensitive to scale variations and distribution differences in data, which could affect its accuracy. In contrast, Mahalanobis distance is more complex. It not only considers the straight-line distance between data points but also incorporates the covariance structure of the data points, thereby more effectively handling issues related to different scales and correlations among variables. This makes Mahalanobis distance exhibit stronger performance when analyzing multidimensional datasets with inherent correlations.

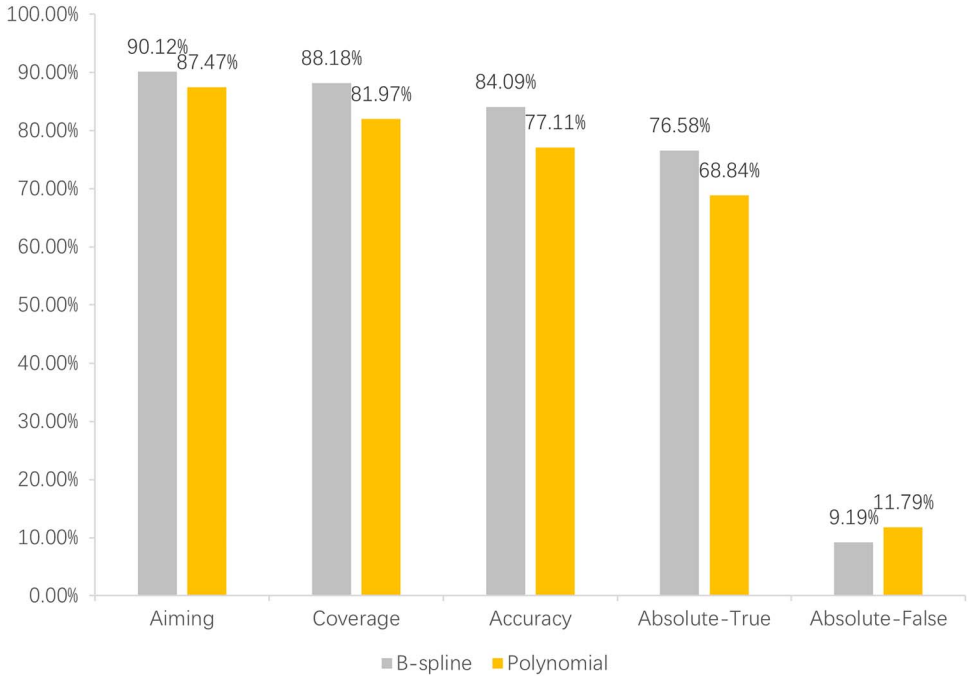


Figure 6. Comparison of the performance of different fitting functions in independent test sets.

Table 4. The results of different distance measurement methods on an independent test set.

Measurement methods	Aiming (%)	Coverage (%)	Accuracy (%)	Absolute-True (%)	Absolute-False
Extreme point deviation	**90.12**	88.18	84.09	**76.58**	**0.0919**
Minkovsky distance	86.45	87.11	81.55	73.10	0.1092
Function expansion coefficient	89.54	86.08	82.42	74.09	0.0994
Traditional European distances	87.76	**88.88**	**85.32**	74.83	0.1038
Traditional Mahalanobis Distance	87.97	87.81	82.36	73.98	0.1065

Bold values indicate optimal values for different metrics.

The experimental results are shown in Table 4, where the distance measurement method with extreme point deviation compensation outperforms other methods in three indicators: Aiming, Absolute-True, and Absolute-False. Additionally, this method is nearly as effective as the best method in two other indicators: Coverage and Accuracy. Therefore, this study ultimately selected the distance measurement method with extreme point deviation compensation.

Validation of sampling algorithm effectiveness

Under the condition of an undersampling ratio of 0.1, we conducted a comprehensive evaluation of the effectiveness of the EPDCC algorithm. We designed and carried out four control experiments: in the first experiment, the EPDCC algorithm was applied for training and testing the dataset; in the second experiment, the centroid-based clustering technology was firstly used to undersample the original training data, followed by training and testing; in the third experiment, we preprocessed the training data using random undersampling strategy, followed by training and testing on this part of the data; and, in the final experiment, training and testing were performed using the original data without any sampling processing.

Figure 7a displays a comparison of different sampling algorithms on the independent test set across five performance metrics. The results indicate that the EPDCC method proposed in this study outperforms other compared methods in terms of Aiming, Absolute-True, and Absolute-False metrics. Figure 7b further illustrates the prediction accuracy of these methods on 11 different categories of data. It can be seen from the results that, except for classes 5, 9, 10, and 11, the EPDCC sampling algorithm achieves higher prediction accuracy in all other categories, with each category achieving complete accuracy. These comparative results fully demonstrate the efficiency and superiority of the EPDCC sampling method proposed in this study.

Normalization Weight Ablation

To explore the effectiveness of the normalized weight method proposed in this study, we conducted a series of ablation experiments. The specific arrangements for these experiments are as follows: after the completion of weight fusion, L2 normalization was avoided while ensuring that the remaining steps remain unchanged. The prediction probability of each label was calculated through the sigmoid function and a threshold of 0.5. We compared and analyzed the performance metrics of the EVBCC dataset before and after the ablation experiment and observed the changes before and after ablation without sampling. The data in Table 5 clearly shows that the experimental results on the EVBCC dataset using the normalized weight method significantly outperform those without using this method, with all evaluation metrics exceeding the results after ablation.

In order to thoroughly assess the effectiveness of this method, an analysis of a test set comprising 5662 independent samples

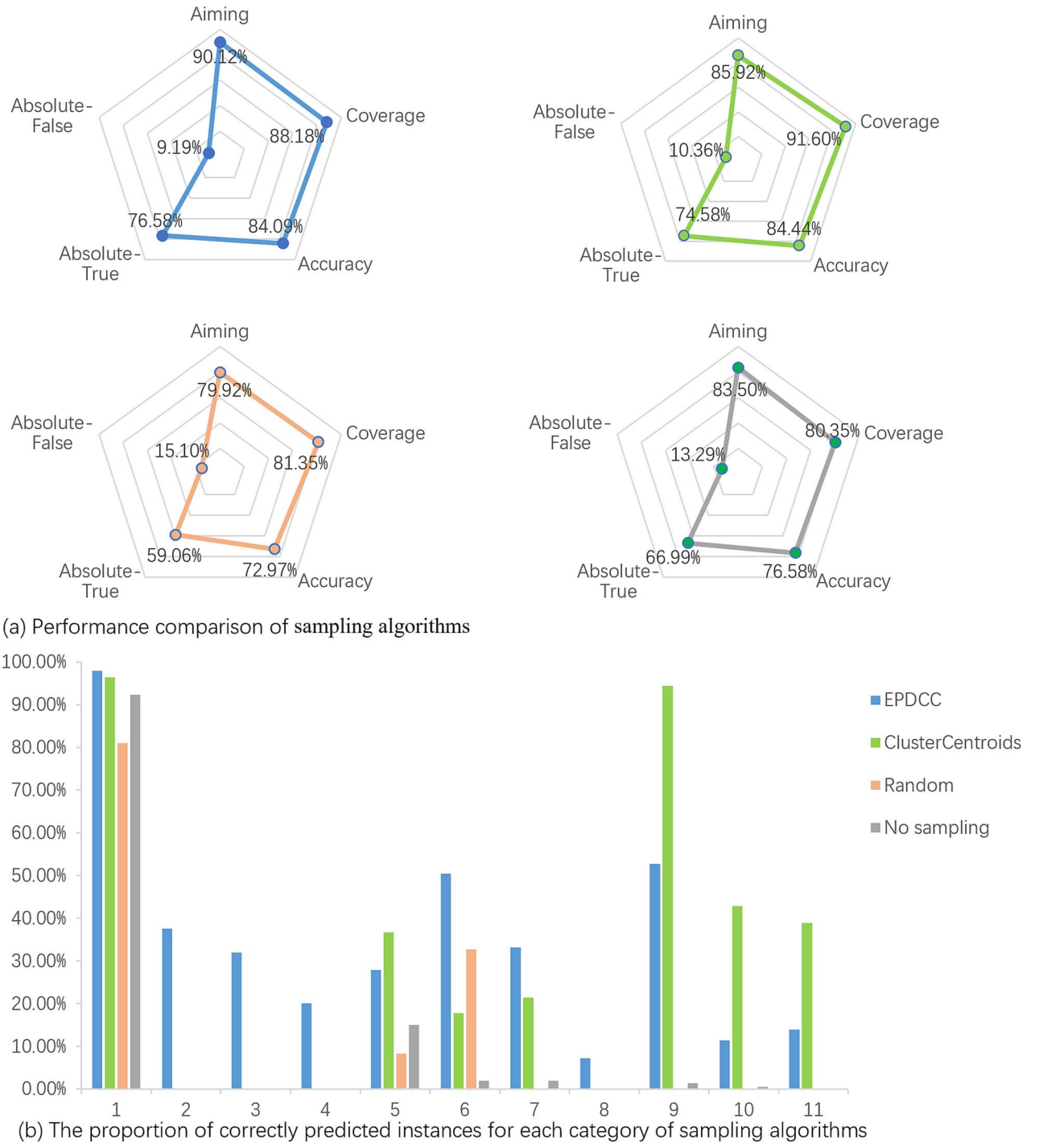


Figure 7. Comparison of indicators of each sampling algorithm on an independent test set. (a) Performance comparison of sampling algorithms. (b) The proportion of correctly predicted instances for each category of sampling algorithms.

Table 5. The results of normalized weight ablation experiments on the independent test set; "+" stands for before ablation, and "−" stands for after ablation.

Methods	Aiming (%)	Coverage (%)	Accuracy (%)	Absolute-True (%)	Absolute-False
EPDCC +	**90.12**	**88.18**	**84.09**	**76.58**	**0.0919**
EPDCC −	76.80	78.67	63.35	44.90	0.1852
No sampling +	83.50	80.35	76.58	66.99	0.1329
No sampling −	86.01	78.10	78.01	71.78	0.1220

Bold values indicate optimal values for different metrics.

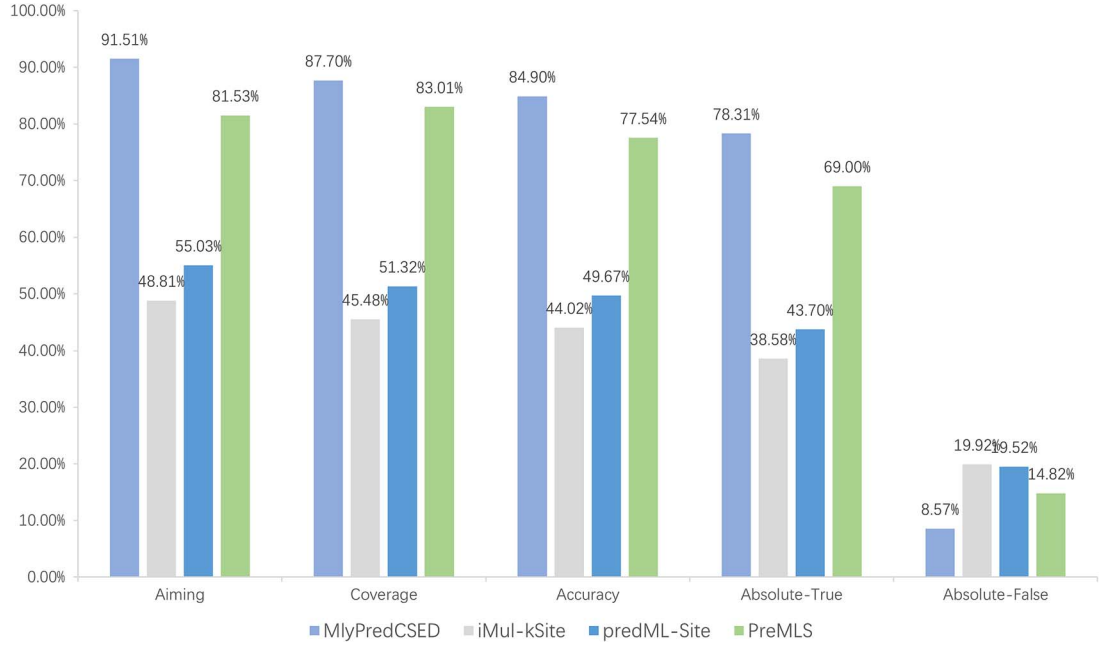


Figure 8. Performance comparison with previous models.

was conducted. The number of samples with predicted labels of (1,*,*,*) (where * represents either 1 or 0) were recorded. Among these, without the application of normalized weights, the first predicted label for all samples was predicted as 1. However, after implementing normalized weights, the number decreased to 4725. This significant change further underscores the effectiveness and reliability of our approach. Without conducting sample collection, we observed that most performance metrics (except for Coverage) improved after the ablation operation compared to before. However, through detailed statistics of the Absolute-True rate for each of the 11 different categories, we found that all samples processed by the ablation method were predicted as (1, 0, 0, 0), meaning that they were all categorized into the first category, whereas before, samples from categories 1, 5, 6, 7, 9, and 10 had Absolute-True predictions. This finding further confirms the effectiveness of our proposed method in reducing model bias.

Comparison with previous models

As illustrated in Fig. 8, we compared the model MlyPredCSED constructed in this study with predML-Site [32], iMul-kSite [33], and PreMLS [52] on an independent test dataset. The existing prediction tools, iMul-kSite and predML-Site, process the original multilabel prediction problem by decomposing it into multiple binary classification problems for each label. Specifically, these tools partition the dataset into positive and negative class samples, based on whether the label is present or absent, and then use an improved Support Vector Machine (SVM) [53, 54] algorithm for binary classification, and finally integrate the individual binary classification models to construct a multilabel prediction model. In contrast, PreMLS employs a deep learning CNN to directly construct a multilabel prediction tool. The experimental results reveal that MlyPredCSED improves by ~10% in the Aiming and Absolute-True metrics and achieves ~7% and 4% improvements in Accuracy and Coverage metrics, respectively. Additionally, it has reduced by ~6% on the Absolute-False metric. Clearly, the overall performance of MlyPredCSED on the test set is superior to the aforementioned models mentioned above. Furthermore, to assess the discriminative capability of our model for the four distinct modifications, we conducted a comprehensive analysis by constructing receiver operating characteristic (ROC) curves. The detailed graphical representations and corresponding analytical results are provided in the supplementary section titled "ROC Curve Analysis of MlyPredCSED."

Table 6. The results of comparison with previous models on the independent test sets (%).

Methods	MR_1	MR_2	MR_3	MR_4
MlyPredCSED	**92.43**	**60.86**	16.8	2.7
PreMLS	88.99	51.9	**25.44**	**100**
predML-Site	59.26	6.93	0.29	0
iMul-kSite	52.64	8.45	0.88	0

Bold values indicate optimal values for different metrics.

As shown in Table 6, we also compared MlyPredCSED with the aforementioned three models on four metrics: MR_1,MR_2, MR_3, and MR_4 . The results demonstrate that MlyPredCSED outperforms other models in both MR_1 and MR_2 , exceeding the worst-performing model by ~40% and 50% in these two metrics, respectively. However, its performance on the MR_4 metric is inferior to that of the PreMLS model. This discrepancy is attributed to PreMLS achieving a 100% prediction accuracy for the label (1, 1, 1, 1), which is a very rare category in the entire dataset. This fully demonstrates that MlyPredCSED is superior in performance to methods based on ensemble binary classification and the PreMLS model.

To further validate the predictive performance of MlyPredCSED, a systematic comparative analysis was conducted between MlyPredCSED and existing state-of-the-art models using two evaluation schemes on the Qiu dataset: five-fold cross-validation and independent testing. Detailed experimental results can be found in the supplementary under the section titled "Performance Comparison of MlyPredCSED on the Qiu Dataset."

Conclusion and prospects

This study delves into data processing, multilabel multiscale learning architectures, and the analysis of experimental results,

significantly enhancing the effectiveness of multiscale feature fusion and undersampling techniques in the field of biosequence analysis. During the data preprocessing phase, the dataset was split into training and testing sets at a ratio of 7:3 and employed two different feature extraction algorithms, MLPSTAAP and amino acid physicochemical properties, to convert the data into numerical vectors. At the same time, the EPDCC algorithm was used for undersampling the majority class, effectively resolving the issue of class imbalance. Through a meticulous experiment, we determined the optimal undersampling ratio, further verifying the effectiveness of the proposed method.

In the application study of MlyPredCSED, we innovatively proposed a normalization weight strategy and presented the adopted neural network architecture in detail along with corresponding figures. The experimental results showed that the proposed method significantly outperformed existing models such as predML-Site, iMul-kSite, and PreMLS in multilabel prediction performance. Moreover, ablation experiments further validated the effectiveness of the normalization weight and undersampling algorithms. The research found that the extremum point deviation compensation-based clustering undersampling algorithm better represents the distribution of the original data compared to random undersampling, thereby reducing information loss.

Although our model has demonstrated impressive predictive capabilities on the test set, there is still room for performance improvement. Specifically, future research efforts can be optimized from the following perspectives:

1. **Exploration resampling algorithms**: The current algorithm primarily addresses the issue of class imbalance by reducing the majority class samples, which may result in the loss of important information. Therefore, future studies should consider employing oversampling techniques to increase the number of minority class samples.
2. **In-depth study of multiscale features**: Existing experimental results support that the fusion of features at different scales can significantly enhance prediction accuracy. Accordingly, subsequent work can explore introducing new scale features, such as protein tertiary structures or 3D image data, to further enhance the model's predictive capabilities [55, 56].
3. **Expanding application scenarios**: The method proposed in this study is not limited to its current applications but can also be extended to other areas of bioinformatics, such as protein function prediction, gene expression data analysis, and drug target prediction, thereby verifying its applicability and stability across different field [57].
4. **Network architecture improvement:** One drawback of using CNNs to process protein sequences is the potential loss of information, as their local receptive fields may fail to capture the global structure of the protein. If key features are widely distributed, CNNs may miss them by focusing only on local areas. Future work will combine CNNs with Graph Convolutional Networks (GCNs) to utilize both protein structure and sequence information, thus more comprehensively capturing protein information and improving structural understanding.

In summary, this study demonstrates the immense potential of multiscale feature fusion and undersampling techniques in the field of biological sequence analysis, providing valuable references for future research and applications. We look forward to refining and expanding our model in future work, enhancing its applicability and depth across a broader range of fields.

Key Points

- Lysine post-translational modifications are a common phenomenon in the field of biology, and their identification and understanding have far-reaching implications for biomedical research and drug development. However, existing studies predominantly focus on single modification types, neglecting potential interactions between different PTM sites. Based on this, our team has developed an innovative multi-label prediction tool that can directly predict multiple lysine modification sites, offering researchers a more comprehensive and precise platform for study.
- This study effectively balanced the dataset by fitting discrete data into functional forms and employing the EPDCC undersampling algorithm for clustering, which significantly reduced the bias toward majority classes during model training.
- By integrating MLPSTAAP with amino acid physicochemical properties, multiscale data were constructed. This not only enriched the semantic depth of the data but also significantly improved the accuracy of predicting post-translational modification sites on lysine.
- The MlyPredCSED model successfully constructed in this study precisely captures the dependencies between elements by effectively integrating information at different scales. The model introduces an adaptive weight normalization technique that adjust the weight matrix using the L_2 norm, effectively mitigating prediction bias towards majority class samples and significantly improving prediction accuracy for minority class samples.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Funding

This work is supported by the National Natural Science Foundation of China [62302198]; Natural Science Foundation of Jiangsu Province of China [BK20231035]; Fundamental Research Funds for the Central Universities [JUSRP124014]; National Key Research and Development Program of China [2021YFE010178]; National Natural Science Foundation of China [62176105]; and Hong Kong Research Grants Council [PolyU152006/19E].

Data availability

The related code and datasets for this study can be obtained at the following locations: https://github.com/xzfang00/MlyPredCSED.

References

1. Xu H, Zhou J, Lin S. *et al*. PLMD: an updated data resource of protein lysine modifications. *J Genet Genomics* 2017;**44**:243–50. https://doi.org/10.1016/j.jgg.2017.03.007

2. Saraswathy N, Ramalingam P. (eds.), *Concepts and Techniques in Genomics and Proteomics*. Oxford: Woodhead Publishing, Elsevier, 2011.
3. Ao C, Yu L, Zou Q. Prediction of bio-sequence modifications and the associations with diseases. *Brief Funct Genomics* 2021;**20**:1–18. https://doi.org/10.1093/bfgp/elaa023
4. 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**:1–10. https://doi.org/10.1093/bib/bbaa275
5. Xu HD, Wang LN, Wen PP. *et al.* Site-specific systematic analysis of lysine modification crosstalk. *Proteomics* 2018;**18**:1–13. https://doi.org/10.1002/pmic.201700292
6. Verdin E, Ott M. 50 years of protein acetylation: from gene regulation to epigenetics, metabolism and beyond. *Nat Rev Mol Cell Biol* 2015;**16**:258–64. https://doi.org/10.1038/nrm3931
7. Lanouette S, Mongeon V, Figeys D. *et al.* The functional diversity of protein lysine methylation. *Mol Syst Biol* 2014;**10**:724. https://doi.org/10.1002/msb.134974
8. Jiang Y, Wang R, Feng J. *et al.* Explainable deep hypergraph learning modeling the peptide secondary structure prediction. *Adv Sci* 2023;**10**:2206151. https://doi.org/10.1002/advs.202206151
9. Sereshki S, Lee N, Omirou M. *et al.* On the prediction of non-CG DNA methylation using machine learning. *NAR Genom Bioinform* 2023;**5**:lqad045. https://doi.org/10.1093/nargab/lqad045
10. Wang Z, He W, Tang J. *et al.* Identification of highest-affinity binding sites of yeast transcription factor families. *J Chem Inf Model* 2020;**60**:1876–83. https://doi.org/10.1021/acs.jcim.9b01012
11. Ilyas S, Hussain W, Ashraf A. *et al.* iMethylK_pseAAC: improving accuracy of lysine methylation sites identification by incorporating statistical moments and position relative features into general PseAAC via Chou's 5-steps rule. *Curr Genomics* 2019;**20**:275–92. https://doi.org/10.2174/1389202920666190809095206
12. Chen Z, Liu X, Li F. *et al.* Large-scale comparative assessment of computational predictors for lysine post-translational modification sites. *Brief Bioinform* 2019;**20**:2267–90. https://doi.org/10.1093/bib/bby089
13. Luo X, Li Q, Tang Y. *et al.* Predicting active enhancers with DNA methylation and histone modification. *BMC Bioinformatics* 2023;**24**:414. https://doi.org/10.1186/s12859-023-05547-y
14. Tang W, Wan S, Yang Z. *et al.* Tumor origin detection with tissue-specific miRNA and DNA methylation markers. *Bioinformatics* 2018;**34**:398–406. https://doi.org/10.1093/bioinformatics/btx622
15. Wang L, Ding Y, Tiwari P. *et al.* A deep multiple kernel learning-based higher-order fuzzy inference system for identifying DNA N4-methylcytosine sites. *Inf Sci* 2023;**630**:40–52. https://doi.org/10.1016/j.ins.2023.01.149
16. Qiao J, Jin J, Yu H. *et al.* Towards retraining-free RNA modification prediction with incremental learning. *Inf Sci* 2024;**660**:120105. https://doi.org/10.1016/j.ins.2024.120105
17. Jin J, Yu Y, Wang R. *et al.* iDNA-ABF: multi-scale deep biological language learning model for the interpretable prediction of DNA methylations. *Genome Biol* 2022;**23**:1–23. https://doi.org/10.1186/s13059-022-02780-1
18. Odeyomi O, Zaruba G. Predicting Succinylation sites in proteins with improved deep learning architecturearXiv preprint arXiv:2201.11215. 2021, 1–5.
19. Huang G, Shen Q, Zhang G. *et al.* LSTMCNNsucc: a bidirectional LSTM and CNN-based deep learning method for predicting lysine succinylation sites. *Biomed Res Int* 2021;**2021**:9923112. https://doi.org/10.1155/2021/9923112
20. Huang K-Y, Hsu JB-K, Lee T-Y. Characterization and identification of lysine succinylation sites based on deep learning method. *Sci Rep* 2019;**9**:16175. https://doi.org/10.1038/s41598-019-52552-4
21. Meng L, Chen X, Cheng K. *et al.* TransPTM: a transformer-based model for non-histone acetylation site prediction. *Brief Bioinform* 2024;**25**:bbae219. https://doi.org/10.1093/bib/bbae219
22. Yu K, Zhang Q, Liu Z. *et al.* Deep learning based prediction of reversible HAT/HDAC-specific lysine acetylation. *Brief Bioinform* 2020;**21**:1798–805. https://doi.org/10.1093/bib/bbz107
23. Hong X, Lv J, Li Z. *et al.* Sequence-based machine learning method for predicting the effects of phosphorylation on protein-protein interactions. *Int J Biol Macromol* 2023;**243**:125233. https://doi.org/10.1016/j.ijbiomac.2023.125233
24. Luo F, Wang M, Liu Y. *et al.* DeepPhos: prediction of protein phosphorylation sites with deep learning. *Bioinformatics* 2019;**35**:2766–73. https://doi.org/10.1093/bioinformatics/bty1051
25. Wang B, Wang M, Li A. Prediction of post-translational modification sites using multiple kernel support vector machine. *PeerJ* 2017;**5**:e3261. https://doi.org/10.7717/peerj.3261
26. Weinert BT, Schölz C, Wagner SA. *et al.* Lysine succinylation is a frequently occurring modification in prokaryotes and eukaryotes and extensively overlaps with acetylation. *Cell Rep* 2013;**4**:842–51. https://doi.org/10.1016/j.celrep.2013.07.024
27. Mizuno Y, Nagano-Shoji M, Kubo S. *et al.* Altered acetylation and succinylation profiles in Corynebacterium glutamicum in response to conditions inducing glutamate overproduction. *Microbiology* 2016;**5**:152–73. https://doi.org/10.1002/mbo3.320
28. Kosono S, Tamura M, Suzuki S. *et al.* Changes in the acetylome and succinylome of Bacillus subtilis in response to carbon source. *PLoS One* 2015;**10**:e0131169. https://doi.org/10.1371/journal.pone.0131169
29. Rardin MJ, He W, Nishida Y. *et al.* SIRT5 regulates the mitochondrial lysine succinylome and metabolic networks. *Cell Metab* 2013;**18**:920–33. https://doi.org/10.1016/j.cmet.2013.11.013
30. Qiu W-R, Sun BQ, Xiao X. *et al.* iPTM-mLys: identifying multiple lysine PTM sites and their different types. *Bioinformatics* 2016;**32**:3116–23.
31. Hasan MAM, Ahmad S. Mlysptmpred: multiple lysine ptm site prediction using combination of svm with resolving data imbalance issue. *Nat Sci* 2018;**10**:370–84. https://doi.org/10.4236/ns.2018.109035
32. Ahmed S, Rahman A, Hasan MAM. *et al.* predML-site: predicting multiple lysine PTM sites with optimal feature representation and data imbalance minimization. *IEEE/ACM Trans Comput Biol Bioinform* 2021;**19**:3624–34. https://doi.org/10.1109/TCBB.2021.3114349
33. Ahmed S, Rahman A, Hasan MAM. *et al.* Computational identification of multiple lysine PTM sites by analyzing the instance hardness and feature importance. *Sci Rep* 2021;**11**:18882. https://doi.org/10.1038/s41598-021-98458-y
34. Zuo Y, Hong Y, Zeng X. *et al.* MLysPRED: graph-based multi-view clustering and multi-dimensional normal distribution resampling techniques to predict multiple lysine sites. *Brief Bioinform* 2022;**23**:bbac277. https://doi.org/10.1093/bib/bbac277
35. Liu M, Li C, Chen R. *et al.* Geometric deep learning for drug discovery. *Expert Syst Appl* 2023;**240**:122498. https://doi.org/10.1016/j.eswa.2023.122498
36. Liu Y, Shen X, Gong Y. *et al.* Sequence alignment/map format: a comprehensive review of approaches and applications. *Brief Bioinform* 2023;**24**:bbad320. https://doi.org/10.1093/bib/bbad320
37. Liu B, Gao X, Zhang H. BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level

and residue level based on machine learning approaches. *Nucleic Acids Res* 2019;**47**:e127. https://doi.org/10.1093/nar/gkz740
38. Zhang W, Tan X, Lin S. *et al.* CPLM 4.0: an updated database with rich annotations for protein lysine modifications. *Nucleic Acids Res* 2022;**50**:D451–9. https://doi.org/10.1093/nar/gkab849
39. Huang Y, Niu B, Gao Y. *et al.* CD-HIT suite: a web server for clustering and comparing biological sequences. *Bioinformatics* 2010;**26**:680–2. https://doi.org/10.1093/bioinformatics/btq003
40. Zulfiqar H, Guo Z, Ahmad RM. *et al.* Deep-STP: a deep learning-based approach to predict snake toxin proteins by using word embeddings. *Front Med* 2024;**10**:10. https://doi.org/10.3389/fmed.2023.1291352
41. Zhu W, Yuan SS, Li J. *et al.* A first computational frame for recognizing heparin-binding protein. *Diagnostics (Basel)* 2023;**13**:1–13. https://doi.org/10.3390/diagnostics13142465
42. Li H, Pang Y, Liu B. BioSeq-BLM: a platform for analyzing DNA, RNA, and protein sequences based on biological language models. *Nucleic Acids Res* 2021;**49**:e129. https://doi.org/10.1093/nar/gkab829
43. Yao Y, Yan S, Han J. *et al.* A novel descriptor of protein sequences and its application. *J Theor Biol* 2014;**347**:109–17. https://doi.org/10.1016/j.jtbi.2014.01.001
44. Zou X, Ren L, Cai P. *et al.* Accurately identifying hemagglutinin using sequence information and machine learning methods. *Front Med (Lausanne)* 2023;**10**:1281880. https://doi.org/10.3389/fmed.2023.1281880
45. Dao FY, Liu ML, Su W. *et al.* AcrPred: a hybrid optimization with enumerated machine learning algorithm to predict anti-CRISPR proteins. *Int J Biol Macromol* 2023;**228**:706–14. https://doi.org/10.1016/j.ijbiomac.2022.12.250
46. Rawat SS, Mishra AK. "Review of methods for handling class imbalance in classification problems." *International Conference on Data, Engineering and Applications*. Singapore: Springer Nature Singapore, 2022,1–617.
47. Ghosh K, Bellinger C, Corizzo R. *et al.* The class imbalance problem in deep learning. *Mach Learn* 2024;**113**:4845–901. https://doi.org/10.1007/s10994-022-06268-8
48. Ai C, Yang H, Liu X. *et al.* MTMol-GPT: De novo multi-target molecular generation with transformer-based generative adversarial imitation learning. *PLoS Comput Biol* 2024;**20**:e1012229. https://doi.org/10.1371/journal.pcbi.1012229
49. Wei T, Tu W.-W., Li Y.-F., Yang G.-P. Towards robust prediction on tail labels. In Zhu F, Ooi BC, Miao C. (eds.), *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21)*, Virtual Event, 2021, 1812–1820.
50. Wei T, Shi JX, Tu WW. *et al.* Robust long-tailed learning under label noise[J]. arXiv preprint arXiv:2108.11569, 2021, 1–14.
51. Chou K-C. Some remarks on predicting multi-label attributes in molecular biosystems. *Mol BioSyst* 2013;**9**:1092–100. https://doi.org/10.1039/c3mb25555g
52. Zuo Y, Fang XZ, Wan JY. *et al.* PreMLS: the undersampling technique based on ClusterCentroids to predict multiple lysine sites. *PLoS Comput Biol* 2024;**20**:e1012544. https://doi.org/10.1371/journal.pcbi.1012544
53. Wang Y, Zhai Y, Ding Y. *et al.* SBSM-Pro: support bio-sequence machine for proteins[J]. *Science China Information Sciences*, 2024;**67**:212106.
54. Wang Y, Zhang W, Yang Y. *et al.* Survival prediction of Esophageal squamous cell carcinoma based on the prognostic index and sparrow search algorithm-support vector machine. *Curr Bioinforma* 2023;**18**:598–609. https://doi.org/10.2174/1574893618666230419084754
55. Zhu H, Hao H, Yu L. Identifying disease-related microbes based on multi-scale variational graph autoencoder embedding Wasserstein distance. *BMC Biol* 2023;**21**:294. https://doi.org/10.1186/s12915-023-01796-8
56. Zhu H, Hao H, Yu L. Identification of microbe–disease signed associations via multi-scale variational graph autoencoder based on signed message propagation. *BMC Biol* 2024;**22**:172. https://doi.org/10.1186/s12915-024-01968-0
57. Ren S, Chen L, Hao H. *et al.* Prediction of cancer drug combinations based on multidrug learning and cancer expression information injection. *Futur Gener Comput Syst* 2024;**160**:798–807. https://doi.org/10.1016/j.future.2024.06.039

Briefings in Bioinformatics, 2025, **26(2)**, bbaf189
https://doi.org/10.1093/bib/bbaf189
Problem Solving Protocol