Computational and Structural Biotechnology Journal 20 (2022) 3268–3279

COMPUTATIONAL ANDSTRUCTURAL BIOTECHNOLOGY JOURNAL

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

iKcr_CNN: A novel computational tool for imbalance classification of human nonhistone crotonylation sites based on convolutional neural networks with focal loss

Lijun Dou [a,b,c], Zilong Zhang [b], Lei Xu [d,*], Quan Zou [a,b,*]

[a] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324000, China
[b] Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China
[c] School of Automotive and Transportation Engineering, Shenzhen Polytechnic, Shenzhen 518055, China
[d] School of Electronic and Communication Engineering, Shenzhen Polytechnic, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 20 March 2022
Received in revised form 13 June 2022
Accepted 13 June 2022
Available online 16 June 2022

Keywords:
Nonhistone crotonylation modification
Imbalance classification
Convolutional neutral networks
Focal loss function

ABSTRACT

Lysine crotonylation (Kcr) is a newly discovered protein post-translational modification and has been proved to be widely involved in various biological processes and human diseases. Thus, the accurate and fast identification of this modification became the preliminary task in investigating the related biological functions. Due to the long duration, high cost and intensity of traditional high-throughput experimental techniques, constructing bioinformatics predictors based on machine learning algorithms is treated as a most popular solution. Although dozens of predictors have been reported to identify Kcr sites, only two, nhKcr and DeepKcrot, focused on human nonhistone protein sequences. Moreover, due to the imbalance nature of data distribution, associated detection performance is severely biased towards the major negative samples and remains much room for improvement. In this research, we developed a convolutional neural network framework, dubbed iKcr_CNN, to identify the human nonhistone Kcr modification. To overcome the imbalance issue (Kcr: 15,274; non-Kcr: 74,018 with imbalance ratio: 1:4), we applied the focal loss function instead of the standard cross-entropy as the indicator to optimize the model, which not only assigns different weights to samples belonging to different categories but also distinguishes easy- and hard-classified samples. Ultimately, the obtained model presents more balanced prediction scores between real-world positive and negative samples than existing tools. The user-friendly web server is accessible at ikcrcnn.webmalab.cn/, and the involved Python scripts can be conveniently downloaded at github.com/lijundou/iKcr_CNN/. The proposed model may serve as an efficient tool to assist academicians with their experimental researches.

1. Introduction

As one of the important post-translational modifications (PTMs), lysine crotonylation (Kcr) was first discovered by Tan et al. in 2011 [1]. It usually exists on histone proteins in the chromatin region with active transcription processes, thereby playing significant roles in reproductive regulation, gene expression, and chromatin structure [2–4]. In 2017, Xu et al. first confirmed the occurrence of Kcr modification on nonhistone proteins [5], and proved that it is widely localized in the cytoplasm and nucleus of H1299 and HeLa cells, as well as a variety of mouse tissues. Subsequent studies elucidated crucial roles in various physiological and pathological processes, indicating its significance in academic researches and medical applications [6,7].

Predicting Kcr positions is the first step to proceed with mechanism investigation. Experimentally, multiple high-throughput techniques have been developed, such as high-performance liquid chromatography fractionation (HPLC), HPLC-tandem mass spectrometry (MS) [6–8]. However, these methods are time-consuming, expensive and labour-intensive, bringing great difficulties to large-scale analysis. Therefore, developing bioinformatics tools based on mathematics and statistics theories become a promising alternative to address this issue. In particular, machine-learning (ML)-based predictors exhibited considerable advantages in terms of time cost and budget, and presented satisfactory prediction results.

[*] Corresponding authors at: Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324000, China.
E-mail addresses: csleixu@szpt.edu.cn (L. Xu), zouquan@nclab.net (Q. Zou).

https://doi.org/10.1016/j.csbj.2022.06.032
2001-0370/© 2022 The Authors. Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology.

To date, a total of eleven predictors have been published for the identification of protein Kcr sites. As summarized in Table 1, the first tool, called CrotPred, applied the discrete hidden Markov model (DHMM) to explore histone Kcr sites in 2016 [9]. Later, five traditional ML-based models were developed, including support vector machine (SVM) model by Qiu et al. [10], CKSAAP_CrotSite (SVM) [11], iKcr-PseEns (ensemble random forest, RF) [12], LightGBM-CroSite (LightGBM) [13], and random forest (RF)/SVM classifiers by Wang et al. [14]. The remaining five were all built on the frame of deep learning (DL), including iCrotoK-PseAAC [15], pKcr [16], Deep-Kcr [17], DeepKcrot [18] and nhKcr [19]. Throughout these predictors, we can observed that: in terms of datasets, the early tools mainly concentrated on histone or mixed data, whereas the recent works began to study nonhistone proteins due to the enrichment of high-throughout nonhistone data. Accordingly, the number of samples sharply increased from 34/90 (the number of positive samples over negative samples) in the first model CrotPred [9] to 15,605/75,111 in the latest model nhKcr [19], which can effectively guarantee the statistical significance of the constructed models; in terms of protein features, it roughly covered several classical easy-interpreted descriptors (i.e., composition of k-spaced amino acid pairs (CKSAAP), one-hot, enhanced amino acid composition (EAAC), pseudo-amino acid composition (PseAAC), pseudo-position specific scoring matrix (PsePSSM), etc) and deep learning representation embedding methods (i.e., wording embedding (WE)); in terms of algorithms, researchers were more inclined to choose deep learning techniques rather than conventional classifiers; as for model evaluation, recent works strictly completed cross validation and independent tests to get objective and reliable results.

In machine learning fields, another challenge is data imbalance, indicating the uneven distribution of samples belonging to different classes. The degree of imbalance can be expressed by the imbalance ratio (IR),

$$IR = \frac{N_{majority}}{N_{minority}} \tag{1}$$

where N_{majority} and N_{minority} indicate the number of samples in the majority and minority classes, respectively. The data imbalance (columns 3 and 4 in Table 1) is actually a common problem in machine learning fields and usually leads to the prediction preference on the samples in the majority class. For instance, the imbalance ratio (IR) in nhKcr tool achieves 12262/60101≈1/5 for the training dataset. As can be seen in the column "(Imbalance) Classifiers", two imbalance strategies of random under-sampling (RUS) [20] and synthetic minority oversampling technique (SMOTE) [21] were implemented to balance the datasets before modelling. Considering the limited amount of data available and the lack of independent test for the early models, here we only discussed two newest tools focused on nonhistone Kcr modification. For the DeepKrot model, the prediction specificity (Sp) for negative samples over 5-fold cross validation (5-CV) and independent tests both achived 90.00%, whereas the associated sensitivity (Sn) for positive samples were only 53.70% and 52.40%, respectively. Similarly, nhKcr separately presented unsatisfied Sp values of 62.86% and 58.90% over 5-CV and independent tests. Generally, compared with the average prediction efficiency (~60%) of positive segments, these two tools are obviously skewed towards negatives (~90%). In the prediction process using above models, the true Kcr sites what experts are actually more concerned about are often negleted. Therefore, it is urgent to develop new Kcr predictors with better and balanced detection performance.

In this research, applying the large-scale nonhistone Kcr data, we developed a novel computational tool based on the neural convolutional networks (CNNs), named iKcr_CNN, to determine moified

Table 1
Description of eleven computational tools for protein Kcr prediction.

Tools	Species	Train	Test	Features(selection)	(Imbalance) Classifier	Results	Sn	Sp	MCC	AUC	Year
nhKcr	nonhistone-human	12,262/ 60,101	3,343/ 15,010	BE, AAINDE, BLOSUM62	CNN	5-cv	**62.86**	**90.00**	**0.51**	**0.88**	2021
						Inde	**58.90**	**90.00**	**0.48**	**0.88**	
DeepKcrot	nonhistone-human	6,687/ 67,106	1,483/ 16,497	WE	CNN	5-cv	53.70	90.00	0.34	0.86	2021
						Inde	52.40	90.00	0.34	0.86	
	nonhistone-papaya	2,742/ 29,676	711/ 7,458			Inde	–	–	–	0.88	
	nonhistone-rice	909/ 11,780	225/ 2,734			Inde	–	–	–	0.86	
	nonhistone-tabacum	1,643/ 9,696	451/ 2,449			Inde	–	–	–	0.84	
Wang's work	histone-mammalian	167/167	44/95	AAC, AAPC, BE, CKSAAP, EAAC, EGAAC(CHI2)	(RUS)RF.SVM	Inde	92.00	88.00	0.80	–	2020
	nonhistone-plant	2,548/ 2,548	669/ 6,720			Inde	83.00	70.00	0.54	0.84	
LightGBM-CroSite	histone-human,mouse	159/847	–	BE, PWAA, EBGW, KNN, PsePSSM (Elastic)	(SMOTE) LightGBM	Jack	98.86	99.11	0.98	1.00	2020
Deep-Kcr	mixed-human	All: 9,964/9,964 (Train/Test = 7:3)		CSKAAP, PWAA, AAIndex, CTD, EBGW	CNN	10-cv	–	–	–	0.86	2020
pKcr	nonhistone-papaya	2,742/ 29,676	711/ 7,458	WE	CNN	10-cv	51.69	90.00	0.34	0.86	2019
						Inde	53.67	90.00	0.34	0.85	
iCrotoK-PseAAC	mixed-mixed	378/500	–	SVV, SM, PRIM, RPRIM, FV, AAPIV, RAAPIV	ANN	Jack	99.17	99.40	98.00		2019
iKcr-PseEns	histone-human,mouse	169/866	–	PseAAC	ensemble RF	Jack	90.53	95.27	81.26	0.98	2018
CKSAAP_CrotSite	histone-human,mouse	169/847	–	CKSAAP, PseAAC	SVM	Jack	92.45	99.17	92.83		2017
Qiu's work	histone protein	159/847	–	PWAA	SVM	Jack	71.69	98.70	77.80		2017
CrotPred	histone-mixed	34/90	–	DHMM		Jack	79.41	77.78	52.59		2016

Asbbreviations: EBGW, encoding based on grouped weight; kNN, k nearest neighbors; PsePSSM, pseudo-position specific scoring matrix; PWAA: position weight amino acid composition; CTD, composition, transition and distribution; RPRIM, position relative incidence matrix; SVV, site vicinity vector; FV, frequency vector; AAPIV, accumulative absolute position incidence vector; RAAPIV, reverse accumulative absolute position incidence vector; CNNs: convolutional neural networks; RF: random forest; SVM: support vector machine; DHMM: discrete hidden Markov model.

Kcr positions. As illustrated in Fig. 1, the modelling process includes five main parts: (1) nonhistone Kcr data collection; (2) protein sequence encoding; (3) one-dimensional CNNs architecture with FL loss; (4) performance evaluation; and (5) web server establishment. Meanwhile, we implemented the powerful dimension reduction method of uniform manifold approximation and projection (UMAP [22]) to visually analyze the data distribution. We demonstrate that the proposed predictor is a reliable tool (available at ikcrcnn.webmalab.cn/.) to recognize potential Kcr sites in nonhistone proteins.

2. Materials and methods

2.1. Benchmark datasets

In this research, we used the benchmark datasets from the related literature and online datasets by Chen et al. for modelling [19], which includes large number of non-redundant experimentally verified Kcr sites on human nonhistone sequences. In short, it was collected through the following steps: first, 19,287 Kcr sites involved in 4230 proteins were filtered from the UniProt database [23]; then, the threshold of 0.3 was set in the CD-HIT program [24] to remove redundant segments, which can effectively avoid the overfitting problem caused by sample similarity; later, centered on the residue K, these protein sequences were further split into 15,603 Kcr and 164,709 non-Kcr segments with a fixed window length of 29, i.e. 14 acids on upstream and 14 acids on downstream; finally, 12,262 positive and 60,101 negative samples were collected (marked as 12,262/60,101) for the training dataset and 3,343/15,010 for the testing dataset. Besides, several samples with auto-filled residue “O” or sparse amino acid “U” were further deleted to avoid potential interference or feature extraction error. Ultimately, the training and testing datasets contained a total of 12,022/59,226 and 3,252/14,792 samples, respectively. Corresponding *IR* between Kcr and non-Kcr samples is approximately 1:5. More details on data preparation process can be found in Ref. [19].

2.2. Protein descriptors

Translating the protein fragments into a computer-recognizable numeric vector is a fundamental step to construct a superior model. In the current study, we explored the effectiveness of eleven common descriptors, which can be generally grouped into four categories: (1) sequential information: one-hot, EAAC and CKSAAP; (2) evolutionary information: bi-profile Bayes (BPB), BLOSUM62 and position-specific scoring matrix (PSSM); (3) physicochemical information: AAIndex, 188D; and (4) deep representation learning embedding information: bi-directional long short-term memory (BiLSTM) [25], word2vec (W2V) [26], and encoder representation from transformers (BERT) [27]. Given a protein sample R = $R_1R_2...R_{l-1}R_l$, the mentioned descriptors can be briefly described as follows:

(1) Sequential information.

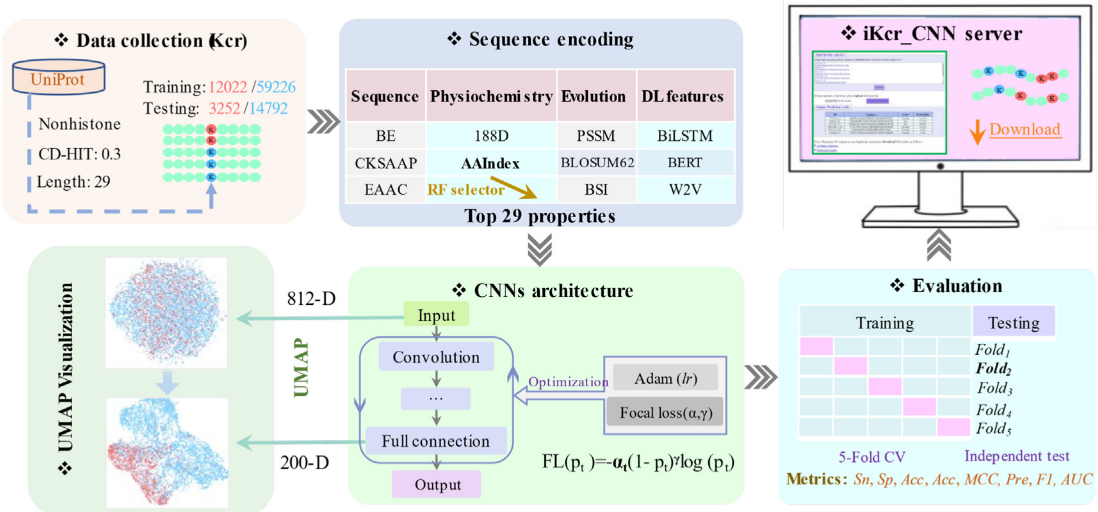
In the one-hot method, 20 different residues, ‘ARNDCQEGHILKMFPSTWYV’, are separately encoded as **1**0000000000000000000, 0**1**000000000000000000, ..., 0000000000000000000**1**, generating a 20**l*-dimensional sparse matrix. EAAC calculates the frequencies of individual residues in the fixed *k*-length segments (default *k* = 5), inducing (*l*-*k* + 1)*20-D features. Similarly, CKSAAP computes the occurrence probabilities of *k*-spaced amino acid pairs. When *k* = 0, it includes 20*20 0-spaced amino acid pairs (i.e., AA, AC, AD, ...). When *k* = 1, it counts 400 1-spaced pairs (i.e., A-A, A-C, A-D, ...), etc. Finally, it generates a (k_{max} + 1)*400-D feature vector.

(2) Evolutionary information.

BPB calculates the position-specific probabilities of 20 amino acids of positive and negative subsets, and applies them directly to define 2**l*-D features. BLOSUM62 implements the simple protein evolutionary matrix BLOSUM62 through the basic local alignment search tool (BLAST) to encode protein sequences with the dimension of 20**l*. Furthermore, PSSM strictly aligns the sequence with large protein sequences in NR database (ftp://download.nmdc.cn/blast/db/nr) to obtain a 20**l*-D scoring matrix locally.

(3) Physicochemical information.

AAIndex incorporates 531 physicochemical properties of amino acids to obtain *l**531-D peptide features by iLearn toolkit [28]. 188D features can be mainly divided into two parts. The first one calculates the amino acid composition (abbreviated as AAC), which



Sequence	Physiochemistry	Evolution	DL features
BE	188D	PSSM	BiLSTM
CKSAAP	AAIndex	BLOSUM62	BERT
EAAC	RF selector	BSI	W2V

Fig. 1. Flowchart of the iKcr_CNN predictor, including data collection, sequence encoding, CNNs-based model development, performance evaluation, web server establishment as well as UMAP visualization of data distribution.

forms a 20-D feature vector. The second part comprehensively integrates composition, transition, and distribution values of eight important physicochemical attributes (CTD) and forms a 168-D feature vector [29,30]. More specific, these physiochemical properties include hydrophobicity, normalized Vander Waals volume, polarity, polarizability, charge, surface tension, secondary structure and solvent accessibility. Ultimately, it generates a 188-D protein feature vector in total to encode protein sequences.

(4) Deep representation learning embedding methods.

By analogy of biological sequences as text, bioinformatics experts successfully applied multiple advanced natural language processing (NLP) techniques into proteomics and genomics with outstanding results [31,32]. Here, we exploited three deep learning embedding methods of BiLSTM [25], BERT [27], and W2V [26]. BiLSTM combines forward and backward LSTM based on the bidirectional propagation mechanisms to extract contextual information [25], well reflecting the global structure similarity between proteins and pairwise residue contact maps for every segment [33]. BERT uses the transformer attention mechanism to realize multiple focus points for the same sentence simultaneously [27]. W2V characterizes residues by the similarity of the involved context/similarity [26]. These features mentioned above can be conveniently acquired through state-of-the-art toolkits, such as iLearn, eFeature (lab.malab.cn/soft/eFeature/), BioSeq-Analysis [34–36].

2.3. Classification algorithms

In recent years, DL technology has made remarkable achievements in image recognition, autonomous driving, NLP, etc. It also demonstrates excellent performance on bioinformatics subjects, such as the prediction of protein structure, protein-protein interactions (PPIs), drug design, and disease treatment [37,38]. Regarding PTMs detection, dozens of state-of-the-art tools have been proposed in the application of CNNs, transfer learning (TL), LSTM and attention-based networks, such as MusiteDeep [39], CapsNet_PTM [40], MultiRM [41].

As illustrated in Fig. 2, our CNNs architecture consists an input layer, three 1D convolutional layers, two fully connected layers, and an output layer. Combining the involved model parameters in Table 2, 812-D AAindex features were served as input and fed into the network; Then, three sequentially connected blocks (i.e., convolutional with a rectified linear unit (RELU) activation function, max pooling, and dropout layers) were followed to extract hidden discriminative patterns. In specific, the number of filters was separately set as 200, 100 and 50 for three convolutional layers with kernel sizes of 7, 5 and 3, as well as the same pooling sizes of 2. Next, the output of Dropout_3 with a size of (98,50) was flattened into a 4,900-D vector and continuously fed into two dense layers to finally generate a 1-D matrix, which was activated by the sigmoid function to output the prediction probability. Here the dropout rates were set as 0.25 for the first three dropout layers and 0.5 for the last, and the batch size as 150 to avoid out of memory error. During the fitting process, we chose fashionable optimizer of Adam with the default learning rate of 0.001 to optimize the involved 1,097,151 parameters in total. This model is implemented in TensorFlow (v1.12.0) and Keras (v2.2.4) [42]. For comparison, we also investigated several traditional machine learning methods in Scikit-Learn package (v0.24.2) [43], including RF [44], SVM [45] and naïve Bayes (NB) [46].

Table 2
Parameters involved in this CNNs model.

Layers	Settings	Output shape	Parameters
Input	selected AAindex	(-,812)	–
Conv1D_1	filters = 200, kernel_size = 7, activation='relu'	(-,860,200)	1600
Maxpooling_1	pool_size = 2	(-,403,200)	–
Dropout_1	rate = 0.25	(-,403,200)	–
Conv1D_2	filters = 100, kernel_size = 5, activation='relu'	(-,399,100)	100,100
Maxpooling_2	pool_size = 2	(-,199,100)	–
Dropout_2	rate = 0.25	(-,199,100)	–
Conv1D_3	filters = 50, kernel_size = 3, activation='relu'	(-,197,50)	15,050
Maxpooling_3	pool_size = 2	(-,98,50)	–
Dropout_3	rate = 0.25	(-,98,50)	–
Flatten		(-,4900)	–
Dense_1	unit = 200, activation='relu'	(-,200)	980,200
Dropout_4	rate = 0.5	(-,200)	–
Dense2	unit = 1, activation='sigmoid'	(-,1)	201
Adam	lr = 0.001	–	–
Total	–	–	1,097,151

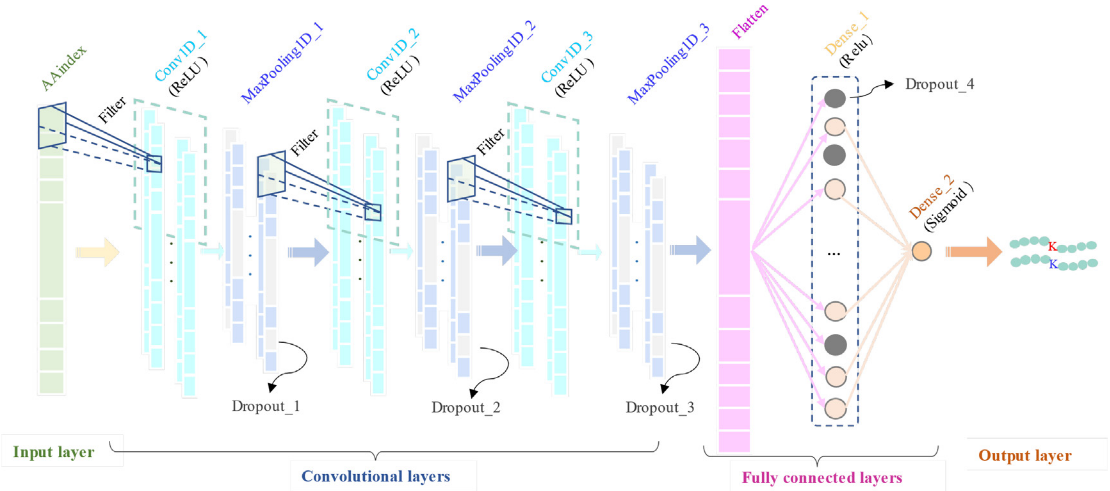


Fig. 2. CNNs architecture of this work. AAindex features are continuously fed into three 1D convolutional blocks (convolution, max pooling and dropout layers) to extract informative attributes. Then, one flatten and two dense layers followed by a sigmoid activation function are incorporated to output the prediction results.

2.4. Imbalance strategy

Data imbalance exists in almost all machine learning fields, such as credit fraud, information security, image processing, bioinformatics [47]. It is known that traditional algorithms proceed optimization by maximizing the overall prediction accuracy. As a result, classification model usually illustrates severe skewness towards the majority class, which is actually converse to the samples of interest to experts, seriously limiting the reliability and applicability of proposed tools. Fortunately, various imbalance strategies have been proposed, such as SMOTE [21], RUS [20], cost-sensitive, ensemble classifiers, which can be roughly grouped into three parts: data-, algorithm-, and hybrid-level methods [48,49]. Taking data-level SMOTE as an example [21], it synthesizes new samples $X\prime$ according to the data similarity of the k-nearest neighbor (KNN) samples X_n randomly chosen from the minority samples $\tilde{X}$, formulated as.

$$X\prime = X_n + rand(0,1) \times (\tilde{X} - X_n) \tag{2}$$

Here, $rand(0,1)$ will produce a random number between 0 and 1. Ultimately, we can generate a large number of new minority class samples to form a balanced dataset and applied to build model.

In the frame of DL, the imbalance issue can be similarly addressed by adjusting class weights, evaluation matrices, and loss functions. For instance, the online hard example mining (OHEM) method increases the weight of mis-predicted samples to obtain better results. However, it neglects the influence of easy-classified samples. As an improvement, Lin et al. proposed focal loss in 2017 when comparing one- and two-stage object detection techniques and presented excellent results [50]. It is mainly benefited from the initial precise identification of valid target areas, which in fact is an imbalance issue. In the past five years, FL has become a powerful way to address imbalance issue [50–53], which not only assigns asymmetrical weights to samples in different classes but also sets different weights on easy- and hard-classified samples. Specifically, the standard cross entropy (CE) in two-class classification is expressed as.

$$CE(p,y) = \begin{cases} -log(p) & if\ y = 1 \\ -log(1-p) & otherwise \end{cases} \tag{3}$$

Here, y indicates the true label of considered samples (1: positive; 0: negative), and p is the predicted probability of the query samples for category $y = 1$ in the range 0–1. Conveniently, p can be writing as.

$$p_t = \begin{cases} p, & if\ y = 1 \\ 1-p, & otherwise \end{cases} \tag{4}$$

Then, CE can be shortened as.

$$CE(p,y) = CE(p_t) = -\log(p_t) \tag{5}$$

FL improves the loss function from two hands as follows:

(1) Distinguishing the contribution of samples of different categories to the loss function by introducing the weight parameter $\alpha \in [0,1]$ for class 1 and $1-\alpha$ for class -1. We define α_t analogously to p_t, then the α-balanced CE loss can be written as,

$$CE(p_t) = -\alpha_t \log(p_t) \tag{6}$$

(2) Distinguishing the contribution of easy- and hard-classified samples to the loss function by introducing the focusing parameter $\gamma(\gamma \geq 0)$, CE will be rewritten as.

$$CE(p_t) = -(1-p_t)^{\gamma}\log(p_t) \tag{7}$$

Ultimately, FL can be ultimately written as.

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{8}$$

In summary, α exposes different weights of samples belonging to positive and negative classes, and $(1-p_t)^{\gamma}$, called the modulating factor, decreases the contributions of easy-classified samples. As a result, FL makes classifiers mostly focused on the minority class and hard-classified samples, as we expected. The best values of α and γ can be optimized on the integrated deep learning platform using grid search using the grid search method of $\alpha = [0.6, 0.75, 0.8, 0.83, 0.85, 0.9]$ and $\gamma = [0.5, 1, 1.5, 2, 3, 5]$.

2.5. Performance evaluation

Here, 5-fold CV and independent tests were both conducted to evaluate model performance. In 5-fold CV, the training dataset is randomly divided into five sub-datasets, in which four are used to train the model and the remaining one to test. It is not completed until all five sub-datasets are applied for training once. In addition, independent test is performed to check model generalizability. Six criteria are used to quantitively measure model efficiency, including the sensitivity (Sn), specificity (Sp), accuracy (Acc), Matthew's correlation coefficient (MCC), precision (Pre) and $F1$ score, formulated as follows,

$$S_n = \frac{TP}{TP + FN} \tag{9}$$

$$S_p = \frac{TN}{TN + FP} \tag{10}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{12}$$

$$Pre = \frac{TP}{TP + FP} \tag{13}$$

$$F1 = 2 \times \frac{P_{re} \times R_e}{P_{re} + R_e} \tag{14}$$

Here, TP = true positive, TN = true negative, FP = false positive, and FN = false negative. As supplements, the receiver operating characteristic (ROC, false-positive rate (FPR) vs. true-positive rate (TPR)) curve, PR curves (precision vs. recall) and relevant areas under curves ($AUROC$, $AUPRC$) are also illustrated as objective induces because of independence of the threshold.

3. Results and discussion

In the current study, we applied the incorporated features of selected AAIndex (812-D) to represent proteins and developed an novel convolutional neural network framework to find potential human nonhistone Kcr sites. To eliminate the prediction bias on negative samples, we applied the focal loss function instead of the standard cross-entropy as the indicator to guide the optimization process. This concise model ultimately demonstrated 77.31% prediction score for true Kcr sites, and 78.62% for false Kcr sites with AUC value of 0.86 over 5-fold CV, as well as 77.87% for true Kcr sites, 76.61% for false Kcr sites with AUC of 0.85 over independent test. Our model presented more well results with real-world datasets. The user-friendly web server is accessible at ikcrcnn.webmalab.cn/.

3.1. Analysis of residues distribution

At first, we analyzed the statistical distribution characteristics of position-specific residues between the positive and negative subsets (*t*-test: $p < 0.05$) in Fig. 3 using the Two Sample Logo platform [54]. Different residues are colored by the charge property, i.e., blue, red, and black mean the positively, negatively charged, and neutral residues, respectively. Overall, apparent differences can be observed between Kcr fragments (upper panel) and non-Kcr fragments (lower panel), where the charged residues (blue and red) were significantly enriched in the modified samples, and contrarily neutral residues (black) in the nonmodified samples. Combining the position preference, we can observe that for the residues close to the center K (−4–4, circled by a green box), negatively charged residues (E, D) and neutral residues (A, N, V, Y, etc) are more likely located in the positive set, whereas the positively charged amino (R, K) as well as multiple neutral residues (P, S, F, etc) in the negative set. In particular, six residues close to the center show higher preference values than 4%, including E (6.9%) in the positive set, K (8.1%) and R (5.8%) in the negative set in the upstream position −1, as well as E (6.3%), D (4.0%) in the positive set, P (4.4%) in the negative set in the downstream position 1. The noticeable position-specific distinguishment is fundamental to building reliable computational Kcr tools.

3.2. Preliminary results of multiple protein descriptors and imbalance/classification algorithms

As depicted in Fig. 1, the modelling process is a complex process where the performance depends on many factors (features, algorithms, hyperparameters, etc.). Thus, it is hard to obtain the absolutely best-performed model, and we can only screen out the locally best one based on the limited considerations in the current study. Here, we proceeded a series of preliminary experiments to assess the performance of several commonly used feature extraction methods (including One-hot, EAAC, CKSAAP, PSSM, AAIndex, etc.). Notably, for each feature representation, we applied multiple imbalance strategies/classifiers (including SMOTE, RF, SVM, CNN (CE), CNN(FL), etc.). After a robust/systematic comparison, we filtered out the best-performed one as the candidate model to carry out further optimization and analysis, which integrates the protein feature AAIndex_nhKcr and the classifier CNNs with FL.

Among these preliminary experiments, we summarized parts of sequence representation results based on CNNs (FL) in Table 3 and different classifiers based on AAIndec_nhKcr features in Table 4 to discuss, respectively. Since the strict implement of standard 5-CV experiment with a large amount of training samples is time-consuming, we split the raw training dataset into two sub-datasets with a ratio of 8:2, marked as data_training (9,599/47,399) and data_validation (2,423/11,827), to finish validation experiment. Ultimately, the associated results of the validation and independent tests were obtained, where second column "Num_Feas" indicates the number of features involved.

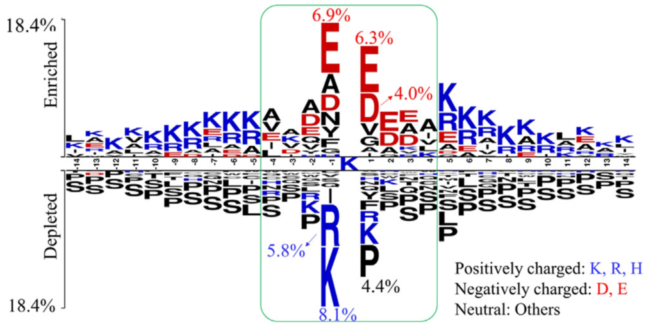


Fig. 3. Two Sample Logo of human nonhistone Kcr segments (p < 0.05), where the positively charged, negatively charged and neutral residues are separately indicated in blue, red, and black [54]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In Table 3, we compared effectiveness of 17 single/combined feature encodings based on the CNNs architecture with FL function ($\alpha = 0.8$, $\gamma = 1$, batch_size = 150, epoch = 10). Among the first six types of sequence information, the 500-D EAAC vector presented good results (validation: *Sn* = 74.24%, *Sp* = 74.74%; independent: *Sn* = 76.73%, *Sp* = 75.83%). For the evolutionary information, except for the BPB descriptor, BLOSUM62 and PSSM both gave higher prediction scores of *Sn* and *Sp* than 70%. Regarding physicochemical methods, three encoding approaches of AAIndex, AAIndex_bhKcr and 188D were considered. Compared with the results of whole AAIndex features, AAIndex_nhKcr displayed more exciting results of 75.24% *Sn*, 78.73% *Sp*, 78.14% *Acc*, 0.44 *MCC*, 0.85 *AUC* over validation, and 81.43% *Sn*, 73.36% *Sp*, 74.80% *Acc*, 0.44 *MCC*, 0.84 *AUC* over independent test. As supplements, we also investigated three types of embedded features of BiLSTM, TAPE_BERT, and W2V using the eFeature toolkit. However, these features all presented unsatisfied scores of 60%$\sim$70%. The combined features in the nhKcr predictor, marked as nhKcr, consisting of one-hot, BLOSUM62, and selected AAIndex, also gave good results [19]. Further incorporation of effective EAAC and PSSM, forming a 3603-D vector didn't display obvious enhancement. In general, these descriptors of EAAC, BLOSUM62, PSSM, AAIndex, AAindex_nhKcr, nhKcr and nhKCR + EAAC + PSSM presented well recognition scores of approximately 75% based on the CNNs (FL) algorithm (marked in bold).

As we all know, AAIndex is one of the classical physiochemistry-related methods considering 531 physicochemical properties in total to induce a 15399-D vector. Furthermore, Chen et al. [19] applied RF as a feature selection approach to construct an effective feature subject with 29 top-ranked properties (marked as AAIndex_nhKcr). After deleting the specific columns with same values for all samples related to K residue at the center, there are (29–1)*29 = 812-D features remained to feed into the CNNs model. Typically, feature extraction methods of PSSM and deep representation learning requires large memory as well as high time cost. In addition, the feature combination process is often accompanied by the curse of dimensionality and information redundancy. By balancing the model performance and computational cost, we ultimately determined to only choose AAIndex_nhKcr features to construct a simple yet efficient computational model. To better illustrate the effectiveness of AAIndex features, we carried out an *t*-test experiment with another representative descriptors of EAAC. Specifically, we repeated validation tests of these two models ten times and calculated the *p*-values, respectively. Taking comprehensive metrics of MCC and AUC as examples, corresponding *p*-values reached 2.19eE-11 and 3.368eE-11. Because *p* was much smaller than 0.05, we can say that the AAIndex_nhKcr encoder is obviously better than EAAC encodings with $\sim$100% confidence intervals.

In Table 4, we summarized the results of a series of (imbalance) algorithms based on the AAIndex_nhKcr features, including three conventional ML classifiers of RF, SVM, NB, and DL algorithm of CNNs. Moreover, we equipped classical SMOTE and different loss functions (CE, αCE and FL) to these classifiers to explore the effect of data imbalance. Among the data in rows 1–4, we can find the serve overfitting problem for the RF model, corresponding to lowest *Sn* of 4.24%. For the SVM and CNN (CE) models, imbalance issue seriously influenced the model performance, where the validation *Sn* scores were only 44.58% and 22.79%, respectively. As for the NB model, although the prediction precision was relatively balanced for samples in two classes, it still cannot meet the experimental requirements. Next four rows of 5–8 indicated the results of typical under-sampling strategy SMOTE implemented with above four

Table 3
Performance of different protein representation approaches based on the CNNs(FL) algorithm.

Features	Num_Feas	Validation					Independent				
		Sn (%)	*Sp* (%)	*Acc* (%)	*MCC*	*AUC*	*Sn* (%)	*Sp* (%)	*Acc* (%)	*MCC*	*AUC*
One-hot	580	56.20	79.82	75.83	0.31	0.76	62.10	75.80	73.33	0.31	0.77
EAAC	500	**74.24**	**74.74**	**74.66**	**0.39**	**0.82**	**76.73**	**75.83**	**75.99**	**0.43**	**0.84**
CKASSP (k_{max} = 0)	400	69.23	62.02	63.24	0.24	0.71	67.23	62.51	63.36	0.23	0.70
CKSAAP (k_{max} = 1)	800	64.79	66.07	65.85	0.24	0.71	60.65	70.60	68.81	0.25	0.72
CKSAAP (k_{max} = 2)	1200	56.94	73.36	70.59	0.24	0.72	61.45	68.77	67.45	0.24	0.71
CKSAAP (k_{max} = 3)	1600	65.32	67.12	66.82	0.25	0.72	62.04	70.63	69.08	0.26	0.72
BPB	58	59.16	72.56	70.30	0.25	0.73	66.92	67.31	67.24	0.27	0.74
BLOSUM62	580	**81.40**	**71.89**	**73.50**	**0.41**	**0.84**	**77.47**	**74.78**	**75.27**	**0.42**	**0.83**
PSSM	580	**75.08**	**74.20**	**74.34**	**0.39**	**0.82**	**70.95**	**74.42**	**73.79**	**0.37**	**0.81**
AAIndex	15,399	**71.64**	**73.92**	**73.53**	**0.36**	**0.81**	**78.79**	**72.35**	**73.51**	**0.41**	**0.83**
AAIndex_nhKcr	812	**75.24**	**78.73**	**78.14**	**0.44**	**0.85**	**81.43**	**73.36**	**74.80**	**0.44**	**0.84**
188D	188	76.82	60.19	63.00	0.28	0.75	72.52	67.72	68.58	0.32	0.77
BiLSTM	3605	55.95	69.62	67.31	0.20	0.69	56.93	67.05	65.23	0.19	0.68
TAPE_BERT	768	74.49	56.62	59.64	0.23	0.71	68.89	63.23	64.25	0.25	0.72
W2V	300	61.52	62.23	62.11	0.18	0.67	59.76	63.95	63.20	0.19	0.67
nhKcr	2523	**75.60**	**78.40**	**77.93**	**0.44**	**0.85**	**74.30**	**76.53**	**76.13**	**0.42**	**0.83**
nhKCR + EAAC + PSSM	3603	**80.57**	**74.16**	**75.24**	**0.43**	**0.85**	**82.69**	**71.91**	**73.85**	**0.43**	**0.85**

Num_Feas: the number of features.
AAIndex_nhKcr: the AAindex-related 841-D features by considering top 29 most important physicochemical properties in the predictor nhKcr [19].
nhKcr: the combined protein features proposed in the predictor nhKcr [19].

Table 4
Comparison of different imbalance strategies/classifiers based on the selected AAindex_nhkcr features.

Classifiers	Validation					Independent				
	Sn (%)	*Sp* (%)	*Acc* (%)	*MCC*	*AUC*	*Sn* (%)	*Sp* (%)	*Acc* (%)	*MCC*	*AUC*
RF	81.38	99.68	90.53	0.82	0.97	4.24	99.44	82.28	0.13	0.77
SVM	44.58	73.64	68.73	0.15	0.64	38.86	75.68	69.04	0.13	0.63
NB	61.99	66.96	66.12	0.22	0.71	62.22	67.13	66.25	0.23	0.71
CNN (CE)	22.79	97.02	84.49	0.30	0.85	36.80	93.97	83.66	0.37	0.85
SMOTE + RF	81.53	99.59	90.56	0.82	0.97	4.15	99.47	82.29	0.13	0.77
SMOTE + SVM	85.71	70.58	78.14	0.57	0.89	50.11	67.63	64.47	0.14	0.63
SMOTE + NB	78.31	73.41	75.86	0.52	0.84	43.77	73.51	68.15	0.15	0.65
SMOTE + CNN (CE)	82.58	97.10	89.84	0.81	0.97	33.75	93.99	83.13	0.34	0.84
CNN (αCE)	77.93	77.76	77.79	0.45	0.85	80.51	74.30	75.42	0.44	0.85
CNN (FL)	75.24	78.73	78.14	0.44	0.85	81.43	73.36	74.80	0.44	0.84

classifiers separately (marked as SMOTE + RF, SMOTE + SVM, SMOTE + NB, and SMOTE + CNN (CE)). Overall, the noticeable overestimation problem can be observed, where the average *Sn* score of 32.85% over the independent test was much lower than that of 82.03% over the validation test. At last, we adapted the weighted CE (marked as αCE) and FL function to the CNN architecture to enhance performance. Once different weights were assigned to the involved classes, the recognition skew on major samples can be significantly eliminated. Accordingly, the average identification scores of αCE- and FL-based CNNs models were sharply increased to 78.78%, contributing to 20%~80% improvement than the previous results.

For a clear comparison of imbalance strategies, radar plot of the independent results of the last six models were depicted in Fig. 4. The SMOTE-based four models were all presented poor results for positive samples. In machine learning research, a very important assumption is that the training and testing datasets share same data distribution characteristics. However, it is a challenging task to keep the data spatial properties unchanged when dealing with the imbalance datasets. Although SMOTE is a classical oversampling technique, it synthesizes new samples based on the data similarity of the *k*-nearest neighbor (KNN) samples by $X\prime = X_n + rand(0,1) \times (\tilde{X} - X_n)$ (more details can be seen in Eq. (2) and Ref. [21]). In our tests, the balanced training dataset-based model performed well over the 5-CV experiment but badly over independent test, especially for the true Kcr sites. We believed that the new balanced training dataset can't precisely keep the origin properties by the simple linear interpolation in SMOTE method. Related classifiers were still dedicated to distinguishing negative samples, and needed to be improved to concentrate on more critical positive samples. Fortunately, *Sn* values were sharply increased to 80.51% for CNN (αCE, cyan line) and 81.43% for CNN (FL, red line). Compared with αCE, FL additionally distinguish the

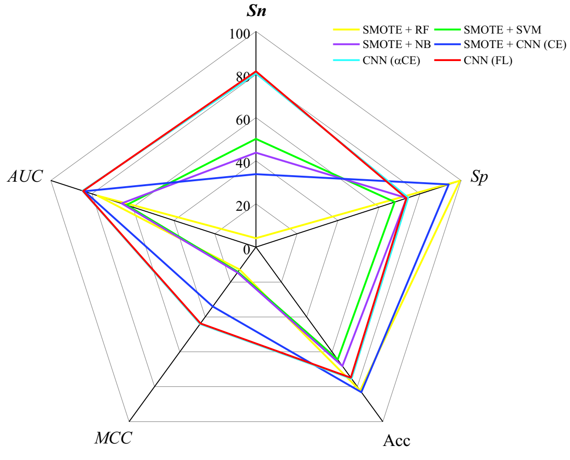


Fig. 4. Radar plot of independent prediction results of six different imbalance classifiers using selected AAIndex_nhKcr features.

easy- and hard- classified samples by the focusing parameter γ, which provides more opportunity to enhance the model power and understand data characteristics. We concluded that the CNN scheme implemented with class weights is superior to the application of SMOTE for true Kcr detection.

3.3. Model optimization and data visualization

After a series of preliminary investigation, we determined the general framework of the model, which integrates the protein feature AAIndex_nhKcr and the classifier CNN with FL. As illustrated in *Imbalance strategies* section, FL function concludes two parameters of α and γ. First, α is introduced to assigning different class weights for samples of different categories. Thus, we can take values according to the imbalance ratio. As for another focusing parameter $\gamma(\gamma \geq 0)$, it is used to distinguish the contribution of easy- and hard-classified samples to the loss function. As depicted in Ref. [50], the contribution of well-classified examples will decrease with γ increasing. Thus, we gradually increased γ values to find best one. Comprehensively, we applied the grid search method of $\alpha = [0.6, 0.75, 0.8, 0.83, 0.85, 0.9]$ and $\gamma = [0.5, 1, 1.5, 2, 3, 5]$ to find best values. After comprehensive comparison, we finally obtained the best model with $\alpha = 0.8$, $\gamma = 0.25$. Based on the optimized parameters, we carried out standard 5-CV and independent experiments to fairly compare with proposed tools. Meanwhile, we recorded the changes of FL values and accuracy scores with epoch increasing in Fig. 5A and B, respectively. More specific, FL at first dropped rapidly within the epochs of 1 ~6, then slowly decreased towards a steady value of ~0.12. We can observe a weak increasing trend in epoch of 16–20, indicating potential overestimation problem. Correspondingly, accuracy also showed an exciting increasement and gradually tended to stabilize. Here, four-fifths (57,000 samples) of the training samples were used to train during the fitting process of 5-fold CV, which means that more than 380 interactions performed in each epoch with the batch size of 150. Comprehensively considering the detection capability and computational costs, we finally used the model with epoch of 16 as our final model (marked by the pale purple area in Fig. 5).

Finally, our model was constructed using the whole training data with selected 821-D AAIndex_nhKcr features based on the CNNs method implemented with FL loss ($\alpha = 0.8$, $\gamma = 0.25$, epoch = 16, batch_size = 150). Fig. 6 plotted the evaluated metrics of optimized model (A. *Sn*, *Sp*, *Acc*, *MCC*, *Pre*, *F1*; B. *ROC* curves; C. *PR* curves). More specific, 5-CV illustrated the results of *Sn* = 77.13%, *Sp* = 78.26%, *MCC* = 0.45, *AUC* = 0.85 and *AUPRC* = 0.52, and correlated independent results of *Sn* = 81.46%, *Sp* = 74.23%, *MCC* = 0.45, *AUC* = 0.87 and *AUPRC* = 0.52. It is found that 5-CV and independent tests gave comparable and balanced results. Thus, we believed that our model is an efficient tool to detect existed Kcr sites.

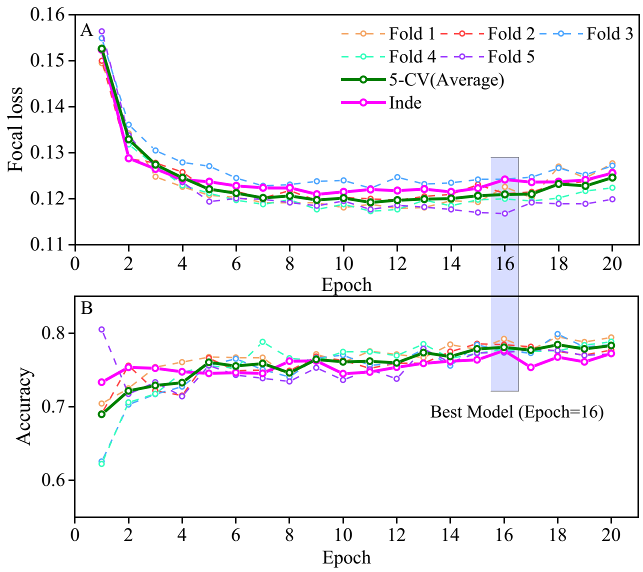


Fig. 5. The changes of FL (A) and Accuracy (B) with increasing epoch (1 ~ 20) over 5-CV and independent experiments.

At last, we visually analyzed the distribution characteristics of the training samples using the popular dimension reduction method UMAP. Because of more than 70,000 samples involved in the training dataset, we randomly selected 10% of all sequences with the same *IR* of 1:5 to perform clear illustration. Fig. 7A plotted the UMAP results of the original 812-D AAIndex protein features, i.e., the input data for the convolution model. It can be observed that the positive (red dot) and negative samples (purple dot) were strongly mixed together and evenly scattered throughout the entire feature space. Furthermore, we extracted the output of hidden fully connected layer ("Dense1" in Table 2) formed 200-D hidden attributes. Similarly, Fig. 7B displayed the UAMP results of CNN-learned hidden features. Remarkably, the positive samples (red dot) were mainly concentrated in the lower-left area, whereas the negative samples in the opposite upper-right area. Compared with original AAIndex features, the mined patterns by a series of convolution, pooling and dropout operations are more discriminative to distinguish Kcr and non-Kcr proteins, which well proved/reflected the effectiveness of the CNNs framework as a feature extraction approach to mine hidden distinctive attributes from inputted simple categories.

3.4. Comparison of different tools and discussion

Only two state-of-the-art computational tools, nhKcr and DeepKcrot, are concentrated on human nonhistone Kcr sites. Therefore, Table 5 compared the prediction performance of iKcr_CNN with these two tools. Of note, nhKcr and iKcr_CNN used the same datasets to train model. Despite ~12% drop in *Sp* of iKcr_CNN than that of nhKcr, the rapid increase in *Sn* from ~60% to ~77% provides greater opportunities to identify hidden real true Kcr sites, which is more crucial for the future application. In terms of DeepKcrot, it illustrated ~37% skew on false Kcr sequences compared to true Kcr sites. As can be seen in Table 1, the datasets used in DeepKcrot are different from this work. Thus, we reperformed the independent test using their testing dataset (containing 1,483 Kcr and 16,497 non-Kcr segments), which can not only check our model generalizability but also be treated as a more fair comparison between DeepKrot and iKcr_CNN. We found that the prediction efficiency of positive samples are sharply improved from 52.40% to 85.7, corresponding to 33.3% improvements, which well reflected the precise identification of true positive samples in our model. Related *MCC*, *AUC*, *Pre* and *F1* separately achieved 0.36, 0.87, 23.37% and 36.72%. Additionally, we further calculated 95% confidence interval results of 5-CV experiments and listed in 3rd row: *Sn* = 77.13% ± 4.51%, *Sp* = 78.26% ± 2.46%, *Acc* = 78.06% ± 1.47%, *MCC* = 0.45 ± 0.02, *AUC* = 0.86 ± 0.00, etc. Taking *Sn* as an example, the average value is 77.13% and will be in the range of 72.62%~81.64% with 95% probability. Similarly, *Sp* will be located in the range of 75.8%~80.72% with 95% probability. In summary, relative to other two tools, iKcr_CNN presented well-balanced prediction results for samples belonging to different classes, especially for the positive samples of interest to biologists. Therefore, it is an efficient bioinformatics tool with well robustness and generalizability, and expected to offer reliable guidance for future

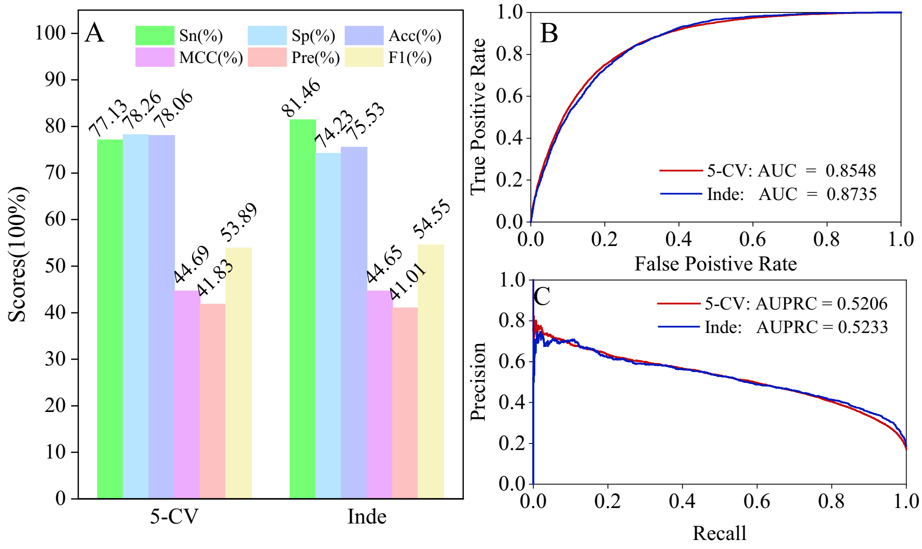


Fig. 6. Evaluated metrics of optimized model over 5-CV and independent tests (A. *Sn*, *Sp*, *Acc*, *MCC*, *Pre*, *F1*; B. *ROC* curves; C. *PR* curves).

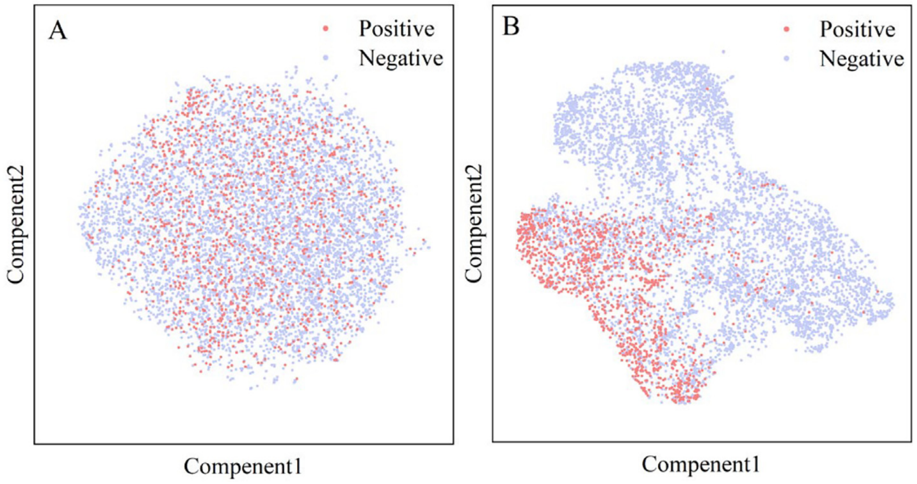


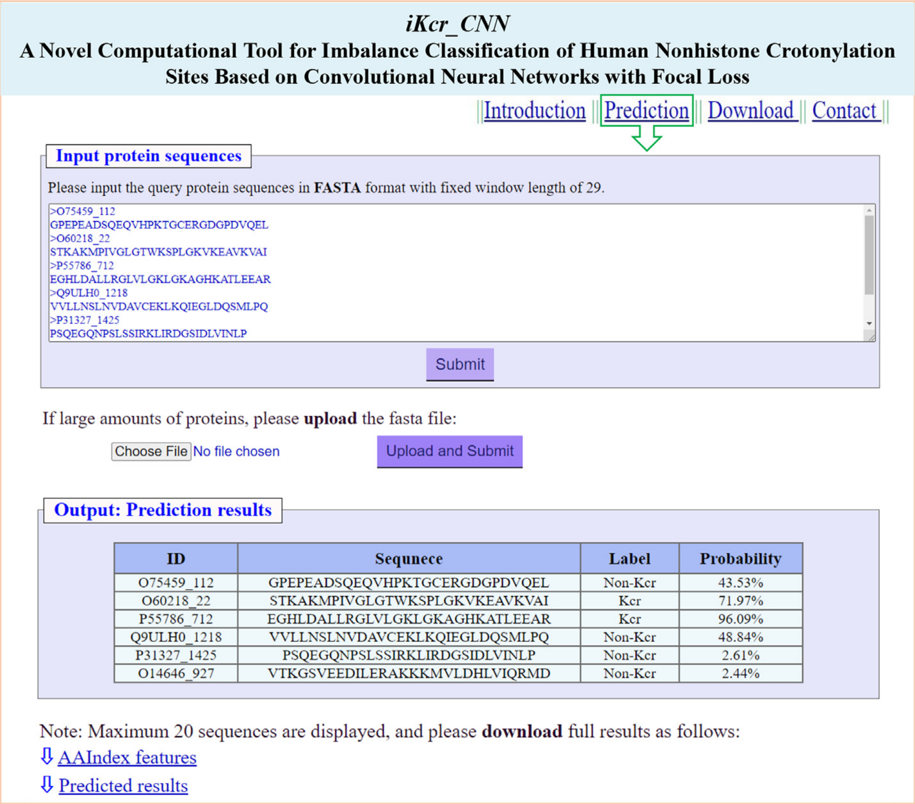
Fig. 7. UMAP visualization results of training samples, where the protein sequences were formulated by the original AAIndex features (A) and mined attributes outputted from the constructed CNNs architecture (B).

Table 5
Comparison of state-of-the-art tools for human nonhistone Kcr sites, where "±" indicated 95% confident interval results.

Tools	Features	Classifier	Results	*Sn* (%)	*Sp* (%)	*Acc* (%)	*MCC*	*AUC*	*Pre*	*F1*
nhKcr	BE, AAIndex, BLOSUM62	CNN	5-CV	62.86	90.00	85.40	0.51	0.88	–	–
			Inde	58.90	90.00	84.33	0.48	0.88	–	–
iKcr_CNN	AAIndex	CNN (FL)	5-CV	77.13 ± 4.51	78.26 ± 2.46	78.06 ± 1.47	0.45 ± 0.02	0.86 ± 0.00	41.83 ± 1.71	53.89 ± 1.35
			Inde	81.46	74.23	75.53	0.45	0.85	41.01	54.55
DeepKcrot	WE	CNN	5-cv	53.70	90.00	87.10	0.34	0.86	–	–
			Inde	52.40	90.00	86.90	0.34	0.86	–	–

researches on associated molecular mechanisms, disease treatments, and drug developments.

It should be noted that these three predictors were all built on the frame of CNNs. Neglecting the protein representations and hyperparameters setting, the biggest difference is the implement of FL function instead of CE to guide optimization. The prediction gap (~40%) between *Sn* and *Sp* in previous tools caused by the imbalance challenge was successfully narrowed to be negligible. During the construction process, we have tried to take into account many factors, including well-known feature extraction methods, feature incorporation, complex feature selection strategies (i.e., F-score-based feature ranking followed by incremental feature selection (IFS)), grid search of hyperparameters in CNN architecture (the number of layers, and involved number of filters, kernel size, dropout rate, learning rate, etc., see Table 2). The evaluated metrics was basically swung at 75%, and hard to obtain remarkable enhancement. Therefore, we implied that, with the appropriate/necessary imbalance strategy, the most basic and crucial method to increase performance is to find discriminative protein representations.


||Introduction ||Prediction|| Download || Contact ||
Submit
Choose File No file chosen
Upload and Submit

iKcr_CNN

A Novel Computational Tool for Imbalance Classification of Human Nonhistone Crotonylation Sites Based on Convolutional Neural Networks with Focal Loss

Input protein sequences

Please input the query protein sequences in **FASTA** format with fixed window length of 29.

```
>O75459_112
GPEPEADSQEQVHPKTGCERGDGPDVQEL
>O60218_22
STKAKMPIVGLGTWKSPLGKVKEAVKVAI
>P55786_712
EGHLDALLRGLVLGKLGKAGHKATLEEAR
>Q9ULH0_1218
VVLLNSLNVDAVCEKLKQIEGLDQSMLPQ
>P31327_1425
PSQEGQNPSLSSIRKLIRDGSIDLVINLP
```

If large amounts of proteins, please **upload** the fasta file:

Output: Prediction results

ID	Sequnece	Label	Probability
O75459_112	GPEPEADSQEQVHPKTGCERGDGPDVQEL	Non-Kcr	43.53%
O60218_22	STKAKMPIVGLGTWKSPLGKVKEAVKVAI	Kcr	71.97%
P55786_712	EGHLDALLRGLVLGKLGKAGHKATLEEAR	Kcr	96.09%
Q9ULH0_1218	VVLLNSLNVDAVCEKLKQIEGLDQSMLPQ	Non-Kcr	48.84%
P31327_1425	PSQEGQNPSLSSIRKLIRDGSIDLVINLP	Non-Kcr	2.61%
O14646_927	VTKGSVEEDILERAKKKMVLDHLVIQRMD	Non-Kcr	2.44%

Note: Maximum 20 sequences are displayed, and please **download** full results as follows:

⇩ AAIndex features

⇩ Predicted results

Fig. 8. Screenshot of the iKcr_CNN web-server (available at ikcrcnn.webmalab.cn/), where the user can input or upload the query proteins (see upper panel) and then results can be presented or download (see lower panel) a few minutes later.

3.5. Construction of the iKcr_CCN web server

The essential purpose of developing prediction models is to assist biological/experimental researchers find potential modified sites. However, local implementation is cumbersome and challenging, especially the configuration of the DL environment of TensorFlow, Keras, etc. Thereby, we built a user-friendly web server to predict Kcr sites online (accessible at ikcrcnn.webmalab.cn/). As illustrated in Fig. 8, four underlined buttons of "Introduction", "Prediction", "Download" and "Contact" are separately linked to brief sever introduction, Kcr prediction interface, resource download and contact details. When predicting potential sites (green box), the user only needs to prepare and input the query proteins in FASTA format with a fixed window length of 29 (the lysine (K) is in center and 14 acids on both sides, see 2.1) or directly upload the FASTA file if large amounts of proteins. After submitting, the predicted results will be illustrated in tabular form, including sample ID, protein sequence, predicted label with default threshold of 0.5 as well as specific probabilities predicted to be Kcr sites a few seconds later (abbreviated as "ID", "Label", "Sequence" and "Probability", respectively). It is noted that the output box can only display up to 20 samples. All prediction results and related AAIndex features can be quickly download below in csv format (↓ icon). All benchmark datasets and python codes can be obtained on the "Download" page as well as the public Git-Hub platform github.com/lijundou/iKcr_CNN/.

4. Conclusion

Precise identification of Kcr sites can facilitate the research progress of the involved modification mechanisms, cellular activities, and medical developments. Although two state-of-the-art prediction tools have been proposed, they presented low efficiencies of positive samples that experimenters are more interested in. In this work, we built a novel computational tool in the deep learning frame, dubbed iKcr_CNN, to predict Kcr site on human nonhistone proteins. To eliminate the prediction preference for major samples caused by imbalance distribution, we implemented the powerful focal loss function as the indicator to optimize the constructed deep learning classifier. Unlike the classical binary cross-entropy (CE), FL not only implements different class weights but also distinguishes the well- and hard-classified samples to fairly treat the positive and negative subsets. Based on only 29 important physicochemical patterns in the AAIndex descriptor, this concise model ultimately demonstrated 77.31% prediction score for true Kcr sites, and 78.62% for false Kcr sites with *AUC* value of 0.86 over 5-fold CV, as well as 77.87% for true Kcr sites, 76.61% for false Kcr sites with AUC of 0.85 over independent tests. Compared to the previous tools, it reported the highest prediction precision of positive samples and showed more balanced performance, which indicated the high efficiency of FL on data imbalance issue. Furthermore, we built an online web-server, named iKcr_CNN, to help scientific researchers conveniently perform Kcr detection (available at ikcrcnn.webmalab.cn/). We anticipate that the proposed model is a reliable tool to detect potential Kcr sites, and provides bioinformatics guidance for further laboratory researches.

Although our model showed balanced prediction scores for positive and negative samples, there is still some room existed to improve its performance. A series of complex optimization experiments involving different protein representation approaches, feature combination/selection, and classification algorithms, can only bring ~5% improvement. Therefore, we believed that developing effective protein representation method is still the fundamental/significant way to build advanced bioinformatics tools. Undeniably, designing novel and effective representation methods is a challenging and meaningful task, which mainly depends on the profound realization of modification mechanisms and biological sequences, combined with certain statistical knowledge. Besides, DL-based multiple mainstream techniques, such as the Autoencoder (AE), transfer learning (TL), generative adversarial network (GANs), are also the promising direction for us to mine more powerful features. In addition, previous studies were basically concentrated on improving the overall prediction performance during the modelling process. It is also valuable to give precise explanation for the specific example which is detected by FL but not detected by other methods. It hopefully brings new insights to the difference of Kcr and non-Kcr samples in sequence level, even helps to uncover the mechanisms of modification. In addition, it is highly encouraged to implement several interpretable algorithms to help biologist understand the model and analyze the contribution of individual features to the predicted results, such as the Shapley Additive explanation (SHAP) [20], Local Interpretable Model-agnostic Explanations (LIME) [22], etc. The mentioned above will be important aspects for us to enhance model performance and conduct more in-depth research in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work was supported by the National Natural Science Foundation of China (Nos. 62002242, 61902259, 61922020), Natural science foundation of Guangdong province (No. 2018A0303130084), Scientific Research Foundation in Shenzhen (Nos. JCYJ20170818100431895, JCYJ20180306172207178), Foundation of Shenzhen Polytechnic (No. 6021310018 K).

References

[1] Tan M, Luo H, Lee S, et al. Identification of 67 Histone Marks and Histone Lysine Crotonylation as a New Type of Histone Modification. Cell 2011;146:1016–28.
[2] Sabari Benjamin R, Tang Z, Huang H, et al. Intracellular Crotonyl-CoA Stimulates Transcription through p300-Catalyzed Histone Crotonylation. Mol Cell 2015;58:203–15.
[3] Montellier E, Rousseaux S, Zhao Y, et al. Histone crotonylation specifically marks the haploid male germ cell gene expression program. BioEssays 2012;34:187–93.
[4] Wei W, Liu X, Chen J, et al. Class I histone deacetylases are major histone decrotonylases: evidence for critical and broad function of histone crotonylation in transcription. Cell Res 2017;27:898–915.
[5] Xu W, Wan J, Zhan J, et al. Global profiling of crotonylation on non-histone proteins. Cell Res 2017;27:946–9.
[6] Lu Y, Xu Q, Liu Y, et al. Dynamics and functional interplay of histone lysine butyrylation, crotonylation, and acetylation in rice under starvation and submergence. Genome Biol 2018;19:144.
[7] Yu H, Bu C, Liu Y et al. Global crotonylome reveals CDYL-regulated RPA1 crotonylation in homologous recombination–mediated DNA repair, Science Advances 2020;6:eaay4697.
[8] Sabari BR, Zhang D, Allis CD, et al. Metabolic regulation of gene expression through histone acylations. Nat Rev Mol Cell Biol 2017;18:90–101.
[9] Huang G, Zeng W. A Discrete Hidden Markov Model for Detecting Histone Crotonyllysine Sites. Match-Communications in Mathematical and in Computer Chemistry 2016;75:717–30.
[10] Qiu W-R, Sun B-Q, Tang H, et al. Identify and analysis crotonylation sites in histone by using support vector machines. Artif Intell Med 2017;83:75–81.
[11] Ju Z, He J-J. Prediction of lysine crotonylation sites by incorporating the composition of k-spaced amino acid pairs into Chou's general PseAAC. J Mol Graph Model 2017;77:200–4.
[12] Qiu W-R, Sun B-Q, Xiao X, et al. iKcr-PseEns: Identify lysine crotonylation sites in histone proteins with pseudo components and ensemble classifier. Genomics 2018;110:239–46.
[13] Liu Y, Yu Z, Chen C, et al. Prediction of protein crotonylation sites through LightGBM classifier based on SMOTE and elastic net. Anal Biochem 2020;609:113903.
[14] Wang RL, Wang Z, Wang HF, et al. Characterization and identification of lysine crotonylation sites based on machine learning method on both plant and mammalian. Sci Rep 2020;10:12.
[15] Malebary SJ, Rehman MSu, Khan YD. iCrotoK-PseAAC: Identify lysine crotonylation sites by blending position relative statistical features according to the Chou's 5-step rule, Plos One 2019;14:e0223993.
[16] Zhao Y, He N, Chen Z, et al. Identification of Protein Lysine Crotonylation Sites by a Deep Learning Framework With Convolutional Neural Networks. IEEE Access 2020;8:14244–52.
[17] Lv H, Dao F-Y, Guan Z-X, et al. Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method. Briefings Bioinf 2020.
[18] Wei XL, Sha YT, Zhao YM, et al. DeepKcrot: A Deep-Learning Architecture for General and Species-Specific Lysine Crotonylation Site Prediction. IEEE Access 2021;9:49504–13.
[19] Chen Y-Z, Wang Z-Z, Wang Y, et al. nhKcr: a new bioinformatics tool for predicting crotonylation sites on human nonhistone proteins based on deep learning. Briefings Bioinf 2021.
[20] Lemaitre G, Nogueira F, Aridas CK. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. 2016, arXiv:1609.06570.
[21] Chawla NV, Bowyer KW, Hall LO, et al. SMOTE: synthetic minority over-sampling technique. J Artif Int Res 2002;16:321–57.
[22] McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2018, arXiv:1802.03426.
[23] The UniProt Consortium. UniProt: the universal protein knowledgebase, Nucleic Acids Research 2016;45:D158-D169.
[24] Fu L, Niu B, Zhu Z, et al. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics (Oxford, England) 2012;28:3150–2.
[25] Alzaidy R, Caragea C, Giles C. Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. 2019.
[26] Asgari E, Mofrad M. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. PLoS ONE 2015;10:e0141287.
[27] Devlin J, Chang M-W, Lee K et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018, arXiv:1810.04805.
[28] Chen Z, Zhao P, Li F, et al. iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA. RNA and protein sequence data, Briefings in bioinformatics 2019;21:1047–57.
[29] Quan Z, Xubin L, Yi J, et al. BinMemPredict: a Web Server and Software for Predicting Membrane Protein Types. Curr Proteomics 2013;10:2–9.
[30] Lin C, Zou Y, Qin J, et al. Hierarchical classification of protein folds using a novel ensemble classifier. PLoS ONE 2013;8:e56499–e.
[31] Bepler T, Berger B. Learning protein sequence embeddings using information from structure. 2019.
[32] Cui FF, Zhang ZL, Zou Q. Sequence representation approaches for sequence-based protein prediction tasks that use deep learning. Briefings in Functional Genomics 2021;20:61–73.
[33] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80.
[34] 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 Research 2019;1:e127.
[35] Chen Z, Zhao P, Li F et al. iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA. RNA and protein sequence data, Briefings in bioinformatics 2019.
[36] Muhammod R, Ahmed S, Md Farid D, et al. PyFeat: a Python-based effective feature generation tool for DNA. RNA and protein sequences, Bioinformatics 2019;35:3831–3.
[37] Lv Z, Ding H, Wang L, et al. A Convolutional Neural Network Using Dinucleotide One-hot Encoder for identifying DNA N6-Methyladenine Sites in the Rice Genome. Neurocomputing 2021;422:214–21.
[38] Huaixu Z, Xiuquan D, Yu Y. ConvsPPIS: Identifying Protein-protein Interaction Sites by an Ensemble Convolutional Neural Network with Feature Graph. Curr Bioinform 2020;15:368–78.
[39] Wang D, Liu D, Yuchi J et al. MusiteDeep: a deep-learning based webserver for protein post-translational modification site prediction and visualization, Nucleic Acids Research 2020;48:W140-W146.
[40] Wang D, Liang Y, Xu D. Capsule network for protein post-translational modification site prediction. Bioinformatics 2018;35:2386–94.
[41] Song Z, Huang D, Song B, et al. Attention-based multi-label neural networks for integrated prediction and interpretation of twelve widely occurring RNA modifications. Nat Commun 2021;12:4011.
[42] Abadi M, Barham P, Chen J, et al. In: TensorFlow: a system for large-scale machine learning. Savannah, GA, USA: USENIX Association; 2016. p. 265–83.
[43] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 2011;12:2825–30.
[44] Breiman L. Random forests. Machine Learning 2001;45:5–32.
[45] Cortes C, Vapnik V. Support-vector networks. Machine Learning 1995;20:273–97.
[46] Zhang H. The Optimality of Naive Bayes. 2004.
[47] Branco P, Torgo L, Ribeiro RP. A Survey of Predictive Modeling on Imbalanced Domains. ACM Comput Surv 2016;49:31.
[48] Kaur H, Pannu HS, Malhi AK. A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. ACM Comput Surv 2019;52:79.

[49] Chawla NV. Data Mining for Imbalanced Datasets: An Overview. In: Maimon O, Rokach L, editors. Data Mining and Knowledge Discovery Handbook. Boston, MA: Springer US; 2010. p. 875–86.

[50] Lin T, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection. In: In: 2017 IEEE International Conference on Computer Vision (ICCV). p. 2999–3007.

[51] Rao S, Narayanaswamy V, Esposito M et al. Deep Learning with hyper-parameter tuning for COVID-19 Cough Detection. In: 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA). 2021, p. 1-5.

[52] Wang Z, Dong Q, Guo W, et al. Geometric imbalanced deep learning with feature scaling and boundary sample mining. Pattern Recogn 2022;126:108564.

[53] Johnson JM, Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data 2019;6:27.

[54] Vacic V, Iakoucheva LM, Radivojac P. Two Sample Logo: a graphical representation of the differences between two sets of sequence alignments. Bioinformatics 2006;22:1536–7.