Briefings in Bioinformatics, 2022, **23**(**2**), 1–9

https://doi.org/10.1093/bib/bbac037

Problem Solving Protocol

Adapt-Kcr: a novel deep learning framework for accurate prediction of lysine crotonylation sites based on learning embedding features and attention architecture

Zutan Li[†], Jingya Fang[†], Shining Wang, Liangyun Zhang, Yuanyuan Chen and Cong Pian

Corresponding authors: Cong Pian, College of Sciences, Nanjing Agricultural University, Nanjing, Jiangsu, China; Simcere Diagnostics Co., Ltd., Nanjing, Jiangsu, China. Tel.: (86)0258-3360006; E-mail: piancong@njau.edu.cn; Liangyun Zhang, College of Sciences, Nanjing Agricultural University, Nanjing, Jiangsu, China. Tel.: (86)0258-3360006; E-mail: zlyun@njau.edu.cn; Yuanyuan Chen, Department of Mathematics, College of Science, Nanjing Agricultural University, Nanjing, China. Tel.: (86)0258-3360006; E-mail: chenyuanyuan@njau.edu.cn

[†]These authors contributed equally to this work.

Abstract

Protein lysine crotonylation (Kcr) is an important type of posttranslational modification that is associated with a wide range of biological processes. The identification of Kcr sites is critical to better understanding their functional mechanisms. However, the existing experimental techniques for detecting Kcr sites are cost-ineffective, to a great need for new computational methods to address this problem. We here describe Adapt-Kcr, an advanced deep learning model that utilizes adaptive embedding and is based on a convolutional neural network together with a bidirectional long short-term memory network and attention architecture. On the independent testing set, Adapt-Kcr outperformed the current state-of-the-art Kcr prediction model, with an improvement of 3.2% in accuracy and 1.9% in the area under the receiver operating characteristic curve. Compared to other Kcr models, Adapt-Kcr additionally had a more robust ability to distinguish between crotonylation and other lysine modifications. Another model (Adapt-ST) was trained to predict phosphorylation sites in SARS-CoV-2, and outperformed the equivalent state-of-the-art phosphorylation site prediction model. These results indicate that self-adaptive embedding features perform better than handcrafted features in capturing discriminative information; when used in attention architecture, this could be an effective way of identifying protein Kcr sites. Together, our Adapt framework (including learning embedding features and attention architecture) has a strong potential for prediction of other protein posttranslational modification sites.

Keywords: protein lysine crotonylation, phosphorylation, learning embedding features, convolutional neural networks, bidirectional LSTM, attention mechanism, PTMs prediction

Introduction

Posttranslational modifications (PTMs) are reversible or irreversible covalent processing events in the later stages of protein biosynthesis that change a protein's properties through proteolytic cleavage and addition of a modifying group [1, 2]. PTMs have important implications in many biological processes, including cell cycle modulation, DNA repair, gene activation, gene regulation and signaling processes [3–5]. With advances in modern proteomics technologies, over 400 different types of PTMs have been identified. These include the addition of small chemical or complex groups, e.g. phosphorylation, ubiquitination, crotonylation, acetylation, benzoylation or succinylation [6], which can occur on single or multiple amino acid residues [7]. Based on statistics from a leading PTM database, dbPTM, Ser (S), Lys (K) and Thr (T) are the three most frequently modified amino acids, and Lys is the most different amino acid based on the PTM pattern [8]. Lysine crotonylation (Kcr) is a newly discovered PTM found in several types of eukaryotes; it is a dynamic process that is regulated by the co-regulation of crotonyl-transferases and debatotylatase [9]. Currently, it is known that Kcr is involved in the normal processes of DNA replication, cell cycle, spermatogenesis and embryonic stem cell differentiation, among other biological mechanisms. It is also known to be associated with disease states, acute kidney injury, HIV latency and colon cancer [10–14].

Due to recent advances in proteomics technologies, various experimental techniques have been reported

Zutan Li is a bioinformatics doctoral student at Nanjing Agricultural University. His research interest is in epigenetics and deep learning.
Jingya Fang is a bioinformatics doctoral student at Nanjing Agricultural University. Her research interest is in Computational Biology.
Shining Wang is an undergraduate student in the Department of Mathematics at Nanjing Agricultural University. Her research interest is in deep learning.
Liangyun Zhang is a professor of College of Sciences at Nanjing Agricultural University. His research area is Computational Biology.
Yuanyuan Chen is an associate professor of College of Sciences at Nanjing Agricultural University. Her research area is Computational Biology.
Cong Pian is an associate professor of College of Sciences at Nanjing Agricultural University. His research area is Computational Biology.

Received: January 1, 2022. **Revised:** January 18, 2022. **Accepted:** January 25, 2022

that could promote the study of Kcr distribution and function. These techniques include stable isotope labeling, high-performance liquid chromatography fractionation, affinity enrichment, specific antibodies and high-resolution liquid chromatography–tandem mass spectrometry [15]. However, all of these methods are time consuming and labor intensive, particularly for large-scale datasets.

Computational tools have previously been developed to predict Kcr modification sites. For instance, Qiu et al. [16] used a new encoding scheme, position weight amino acid composition (PWAA), to identify Kcr sites using the support vector machine (SVM). Ju *et al.* [17] introduced a model called CKSAAP CrotSite, which uses the composition of k-spaced amino acid pairs (CKSAAP) as the input coding and SVM as the algorithm for classification. Liu *et al.* [18] incorporated five kinds of amino encoding methods (binary encoding (Binary), PWAA, encoding based on grouped weight, k-nearest neighbors and pseudo-position-specific scoring matrix) into their model, LightGBM-CroSite, to characterize protein sequence feature information and predict Kcr sites with the LightGBM algorithm. Lv *et al.* [19] built a convolutional neural network (CNN)-based predictor called Deep-Kcr that integrates sequence-based features, physicochemical property-based features and numerical space-derived information with information gain feature selection. The Kcr site predictors described above are highly dependent on handcrafted features (HF), which require engineering by researchers, as input to train predictive models. This can easily lead to dimensional disasters and can affect high-dive information capture. More recently, Qiao et al. established a Kcr site predictor [20] using the BERT model to extract high-dimensional features of protein sequences for input into a bidirectional long short-term memory network (BLSTM)-based classifier. The resulting model, BERT-Kcr, shows good classification performance on Kcr data.

The models discussed above show that CNNs, recurrent neural networks (RNNs), and BERT can be used to accurately identify protein modification sites. However, the current methods still have two main disadvantages: (1) BERT involves tens of millions of parameters that need to be trained, which requires too much time and computational resources for feasibly extracting the characteristics of protein sequences, especially for large-scale input, and (2) not all features along a contextual (protein) sequence contribute to the final prediction in classification tasks. Thus, to solve these two issues, we developed a novel deep learning framework named Adapt-Kcr to identify Kcr sites. We used an adaptive embedding algorithm, which adapts to the characteristics of protein modification by adjusting to specific tasks through reverse propagation during model training. We also used a CNN module, a BLSTM module, and an attention mechanism to better capture the latent information of adaptive embedding. Compared to the methods discussed above, our deep learning model performed significantly better in predicting Kcr sites. Adapt-Kcr additionally has more robust performance in distinguishing between different types of lysine modifications. Using the same framework, we also trained a new model called Adapt-ST on a serine/threonine (S/T) phosphorylation dataset of SARS-CoV-2, which also outperformed state-of-the-art models on S/T datasets. Our method therefore shows strong generalization potential for prediction of other protein modifications.

Materials and methods

Benchmark dataset

Three benchmark datasets were used in this study. The first, denoted as the Kcr dataset, was the same dataset used by Lv *et al.* [19] and Qiao *et al.* [20]; this allowed us to fairly compare the models. Lv *et al.* downloaded above protein sequences from the UniProt database, then used the CD-HIT program [21] to remove redundant sequences by setting the threshold of sequence identity to 30%. The final Kcr dataset included 9964 Kcr sites and 9964 non-Kcr sites. Each sample contained 31 amino acids with the lysine in the middle. To compare the state-of-the-art model on the Kcr dataset, we used the same data segmentation method as Bert-Kcr and Deep-Kcr. The non-redundant dataset was randomly divided into the training and independent testing sets at a ratio of 7:3. The second dataset comprised the Kgly and Kace sites recorded in PLMD and was used to test the performance of multiple models in discriminating between Kcr and other lysine modification sites. Redundant negative samples were removed using CD-HIT with a threshold of 40%. The final dataset included 2989 Kcr sites and 4041 non-Kcr sites (2556 Kgly sites and 1485 Kace sites, which are available at http://zhulab.org.cn/BERT-Kcr_models/data). The third dataset comprised the experimentally verified phosphorylation sites of human A549 cells infected with SARS-CoV-2, which were collected from the literature [22]. CD-HIT was used with a sequence identity threshold of 30% to reduce redundancy. The phosphorylation sequences were truncated into 33-residue-long sequence segments with S/T located at the center. To balance the positive and negative data, the negative samples were selected randomly to match the number of positive samples. A total of 5387 positive samples and 5387 negative samples of S/T sites were obtained. Similarly, in order to compare the best models on the phosphorylation dataset, we used the same data segmentation method as Deep-Ips, which were randomly separated into training and independent testing sets at a ratio of 8:2.

Adaptive embedding module

In the adaptive embedding module, we focused on the token vector information and position information of 20 amino acid types in each protein sequence. First, we mapped each of 20 amino acids to a vector by summing up a specific random initialized vector using a lookup table and the position of the letter in the whole sequence. When the model was being trained, each fusion vector could adjust adaptively according to the task with backpropagation. The description of the embedding is as follows:

$$\text{Embedding} = \text{embed}_{\text{token}} + \text{embed}_{\text{position}} \tag{1}$$

CNNs, long short-term memory networks and attention mechanism

Due their high learning efficiency, CNNs are widely used in image processing, speech recognition, and image semantic segmentation. CNNs contain a set of learnable filters, each of which is convolved with the input of the layer to encode the local knowledge of a small receptive field. To successfully capture the spatial and temporal dependence in an image, a CNN block generally consists of three parts: convolution layers, pooling layers, and fully connected (FC) layers. In the picture field, the convolution layer can extract high-level features such as edges, color, and gradient orientation through multiple feature mapping. A pooling layer is often used to compress the resolution of feature mapping, extracting dominant features of rotational and positional invariant and decreasing the computational cost. CNN has low power in sequence analyses, such as natural language processing, because it does not consider dependence between inputs. RNNs can overcome this shortcoming, but due to the problems of gradient vanishing and gradient exploding, RNN training is very difficult and its application is limited. LSTMs, a special type of RNN, are designed to solve gradient explosion and disappearance [23]. LSTMs have greatly improved the early RNN structure, broadened the application range of RNN and laid the foundation for the development of subsequent sequence modeling. An LSTM layer consists of a set of recurrently connected blocks, which contain one or more recurrently connected memory cells and multiplicative units. In LSTM, a storage mechanism is used to replace the hidden function used in traditional RNN, which consists of a set of recurrently connected blocks. The recurrently connected memory cells and multiplicative units enhance the learning ability of LSTM for long-distance dependency. Compared with unidirectional LSTM, BLSTM better captures the information of sequence context. In addition to the BLSTM architecture, the attention mechanism can also be employed to capture positional information. It was originally proposed to solve machine translation tasks [24] and has proven to be capable of distinguishing between more and less important information [25]. In the field of natural language processing and image identification, an increasing number of studies have recently been conducted to explore the application of advanced deep learning techniques such as the attention mechanism in an effort to improve model interpretability [26–29]. The attention mechanism is often used in bioinformatics in conjunction with RNN [30] and has been shown to achieve a competitive performance in a wide range of biological sequence analysis problems [31–33]. It was therefore used in this study to identify the key information that affects Kcr ~~site~~ sites prediction.

The Adapt-Kcr model

To fully capture the information in protein sequences, we used a deep learning network, Adapt-Kcr. This network has an adaptive embedding module to embed protein sequence data based on amino acid token and position information, then continuously update the embedding value using the gradient. Adapt-Kcr also includes one CNN layer to extract high-level features in the sequence, one BLSTM layer to learn dependence structure along the sequence, one attention mechanism layer to identify key information within input data and one FC layer. The convolution layer in the CNN collocated 256 filters, with each filter size set to 10. Rectified linear activation unit (ReLU) was used as the activation function in the CNN layer as follows:

$$\text{ReLU}\,(\chi) = \begin{cases} 0, & \textbf{if } x < 0 \\ x, & \textbf{else} \end{cases} \tag{2}$$

where x is the feature map from the convolution operation.

The convolution layer was used to capture higher level features underlying each sequence. To minimize feature redundancy and prevent overfitting, a pooling layer with Max Pooling was added after the convolution layer. One BLSTM layer with a hidden unit size of 32 was added after the CNN to learn the dependence structure in the sequence. Next, one attention layer with hidden size 10 was added after BLSTM to identify the key information in the feature matrix. The attention layer computed the weight coefficient matrix using the following formula:

$$T = \textbf{softmax}\,(s\,(M, Q)) \tag{3}$$

where M is the input matrix, Q represents the weight matrix of attention and $s(M, Q)$ is the attention scoring function represented as $s(M, Q) = MT \times Q$.

In addition, a FC layer with 32 hidden units was used in this model, with the size of output equal to 2. Final classifications were made using a sigmoid activation function to combine the outputs from the FC layer. The framework of Adapt-Kcr is shown in Figure 1.

Adapt-Kcr uses binary cross-entropy as the loss function, which measures the difference between the target and the predicted output as follows:

$$\text{L}\,(\mathbf{w}) = -\sum_{i=1}^{N} y_i \log\left(y_i'\right) + \left(1 - y_i\right) \log\left(1 - y_i'\right) + \alpha \|w\|_2 \tag{4}$$

where y_i is the true label, y_i' is the corresponding predicted value from Adapt-Kcr and $\alpha\|w\|_2$ is a regularization term to avoid overfitting.

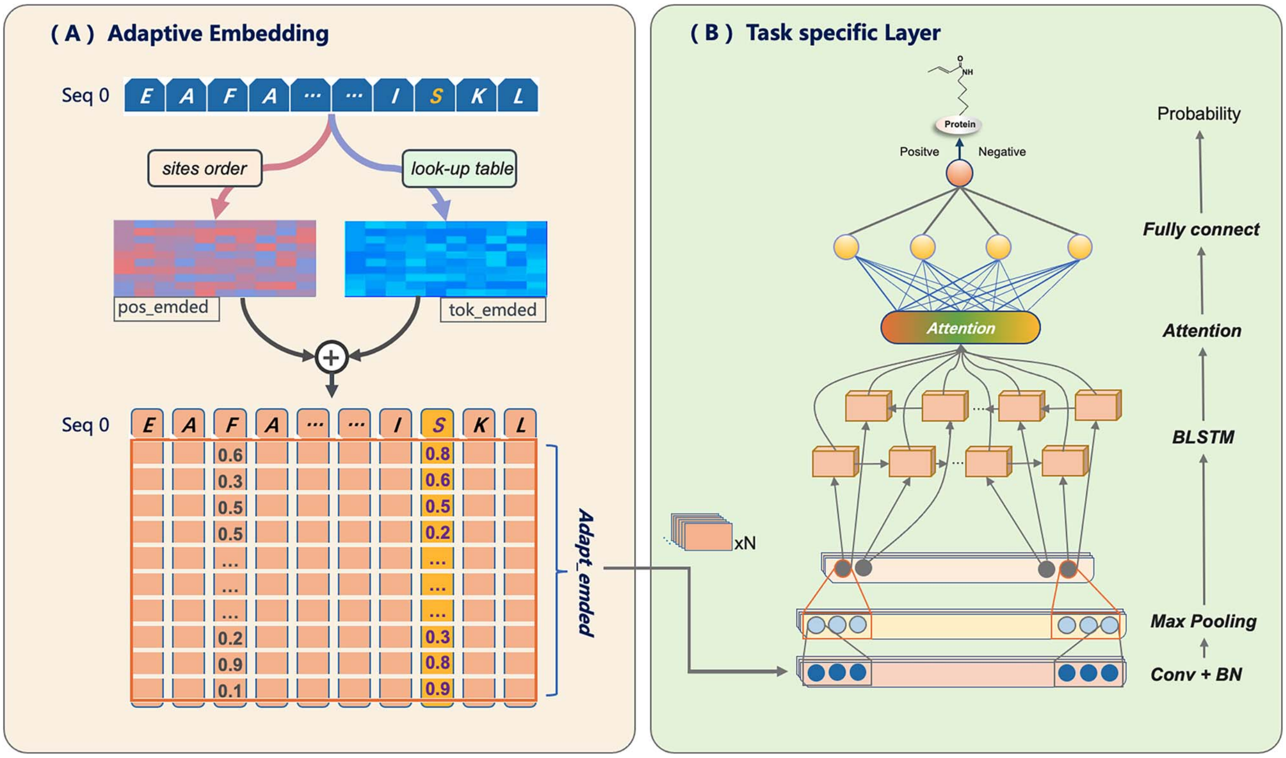

Figure 1. The flowchart of Adapt-Kcr. Adapt-Kcr consists of two main modules: (**A**) an adaptive embedding module that focuses on both token embedding and position embedding and adjusts with backpropagation, and (**B**) a task-specific layer consisting of CNN, BLSTM, attention and full connection layers to the responding probability distribution of a specific class.

Adam [34], Batch normalization and dropout [35] were applied in the Adapt-Kcr training procedure to accelerate training and avoid overfitting. When training the model, the dropout rate, learning rate and reduced factor were set to 0.5, 0.001 and 0.5, respectively. The maximum training epoch was 50 and the batch size was set at 256. An early stopping strategy was used in the training process, meaning training was stopped when prediction performance did not improve on the validation set; the patience was set to 20. The whole framework of this model was implemented in Pytorch (https://pytorch.org).

Prediction accuracy assessment

Prediction accuracy (ACC), Matthews correlation coefficient (MCC), sensitivity (SEN) and specificity (SPE), defined below, were used to evaluate the performance of different models:

$$\mathrm{SEN} = \frac{T_P}{T_P + F_N} \tag{5}$$

$$\mathrm{SPE} = \frac{T_N}{T_N + F_P} \tag{6}$$

$$\mathrm{Precision} = \frac{T_P}{T_P + F_P} \tag{7}$$

$$\mathrm{ACC} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{8}$$

$$\mathrm{MCC} = \frac{T_P \times T_N - F_P \times F_N}{\sqrt{(T_P + F_P) \times (T_N + F_N) \times (T_P + F_N) \times (T_N + F_P)}} \tag{9}$$

where the true positive (T_P) is the number of correctly predicted Kcr sequences, true negative (T_N) is the number of correctly predicted non-Kcr sequences, false negative (F_N) is the number of Kcr sequences incorrectly predicted as non-Kcr and false positive (F_P) is the number of non-Kcr sequences incorrectly predicted as Kcr. The area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve were used to comprehensively evaluate and compare the performance of different models.

Results

Model performance comparison for adaptive embedding versus handcrafted feature encodings

We evaluated and compared the prediction performance of models trained on features generated with eight different protein embedding methods or with adaptive embedding used by basic CNN architecture. The convolution layer in the CNN collocated 256 filters, with each filter size set to 10. ReLU was used as the activation function in the CNN layer. Finally, a FC layer with 32 hidden units was used, with the size of output equal to 2. The eight common HF encodings tested were amino acid composition (AAC), Binary, CKSAAP, Kmer dipeptides composition (DPC), composition (CTDC), normalized moreau-broto (NMBroto), quasi-sequence-order descriptors (QSOrder) and pseudo-amino acid composition (PAAC) according to iLearnPlus [36]. Binary performed better in eight HF embedding methods (Figure 2). However, adaptive embedding ranked first in terms of all evaluation metrics; the model built with

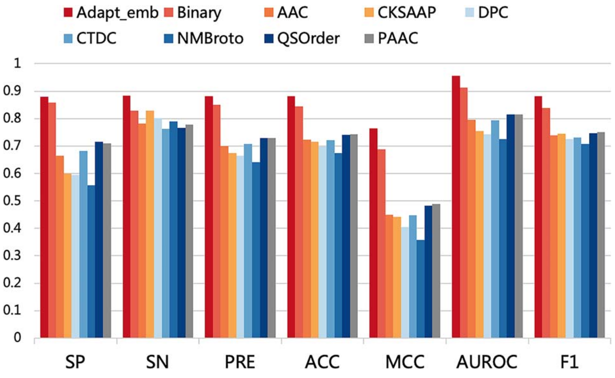

Figure 2. Performance comparison between adaptive embedding and eight HF encodings.

these features had values that were higher than the binary model by 3.84%, 7.6%, 4.3%, 4.4%, 2.1% and 5.6% in ACC, MCC, AUROC, F1, SPE and SEN, respectively.

In addition, we plotted the two-dimensional output feature vectors in the plane (Figure 3) by setting the output dimension of the penultimate neural network to 2 in the classification module. The positive and negative samples of the adaptive embedding model are clearly separated, the performance of the binary model is second-best, and the others are intermixed. This may be because adaptive embedding methods have the ability to capture information hidden in protein sequences by mapping truncated proteins from high-dimensional to low-dimensional spaces. Thus, we found that adaptive learning features were more effective than ~~HF~~ these handcrafted features in the prediction of lysine crotonylation.

Selecting model architecture of Adapt-Kcr

We compared multiple model architectures by analyzing the training data (described in the 'Benchmark dataset' section of the Materials and methods) using adaptive embedding with different deep learning feature extractors, namely, CNN, long short-term memory network (LSTM), CNN–LSTM, LSTM–attention and CNN–LSTM–attention. All detailed evaluation indicators for the resulting models are shown in Table 1. In general, the overall performance of each deep learning model was good using adaptive embedding; in particular, CNN–LSTM–attention achieved higher performance than the other deep-learning architectures. Compared to the CNN model, the ACC, MCC and AUROC values of the CNN–LSTM model were higher by 1.0%, 1.7% and 0.5%, respectively. This is because LSTM can consider the information of the sequence context, thereby slightly enhancing the performance of the constructed model. In addition, CNN–LSTM–attention achieved higher performance in terms of SEN, Pre, ACC, MCC and AUROC than CNN–LSTM. This demonstrates the capacity of the attention mechanism to identify key information and thus improve model performance. Furthermore, we found that the LSTM and LSTM–attention models did not perform as well as the other deep-learning approaches in predicting Kcr sites, indicating that LSTM may not be an ideal architecture for Kcr site prediction. Based on these results, CNN–LSTM–attention was chosen to build the final Adapt-Kcr model.

Comparison with other leading methods on the independent testing dataset

To further assess the performance of Adapt-Kcr, we compared our model with two other existing Kcr site prediction tools, Deep-Kcr and BERT-Kcr, using an independent testing set (Table 2). Adapt-Kcr showed better predictive performance than BERT-Kcr and Deep-Kcr. More specifically, the SEN, Pre, ACC, MCC and AUROC for the Adapt-Kcr model outperformed BERT-Kcr, the state-of-the-art model, by 5.3%, 2.2%, 3.2%, 6.6% and 1.9%, respectively.

In addition, using the receiver operating characteristic and precision-recall curves as metrics, our model performed better in predicting Kcr sites than the other methods did on the independent test set (Figure 4A and B). The attention vectors of Adapt-Kcr show that the weights of the central regions are larger than those of the marginal regions (Figure 4C), i.e. the positions near the modification site have made more contributions to the final prediction. In total, these results indicated that Adapt-Kcr has excellent predictive ability for Kcr sites compared with other existing tools.

Validation on other lysine modifications from Kcr

We additionally tested whether Adapt-Kcr could discriminate between Kcr sites and other types of lysine modifications, namely, glycation (Kgly) and acetylation (Kace) sites. Using Kgly and Kace sites as the negative samples and Kcr sites from the independent testing data as positive samples, we tested the performance of several models in discriminating between lysine modifications (Table 3). None of the candidate models performed well. This is likely because of the large differences between the negative samples in the benchmark datasets compared to the negative samples from other proteins, meaning that the effective features of other proteins had not been seen during model training. Nevertheless, our approach performed better than the other five methods, demonstrating its superior robustness and effectiveness compared to the other methods.

To facilitate understanding, the highest value in each column is shown in bold.

S/T phosphorylation site prediction with the adaptive method

To further verify the scalability of the adaptive method, we tested the performance of the model on other PTM data. We used serine and threonine (S/T) phosphorylation sites for this purpose, because lysine, serine and threonine are the three most frequently modified amino acid residues. We used experimentally verified phosphorylation sites of human A549 cells infected with SARS-CoV-2 collected from literature [22] as the positive samples. We used the same architecture and parameters as those used for Adapt-Kcr to train a new model, Adapt-ST, on S/T data. To fairly compare methods, we rebuilt

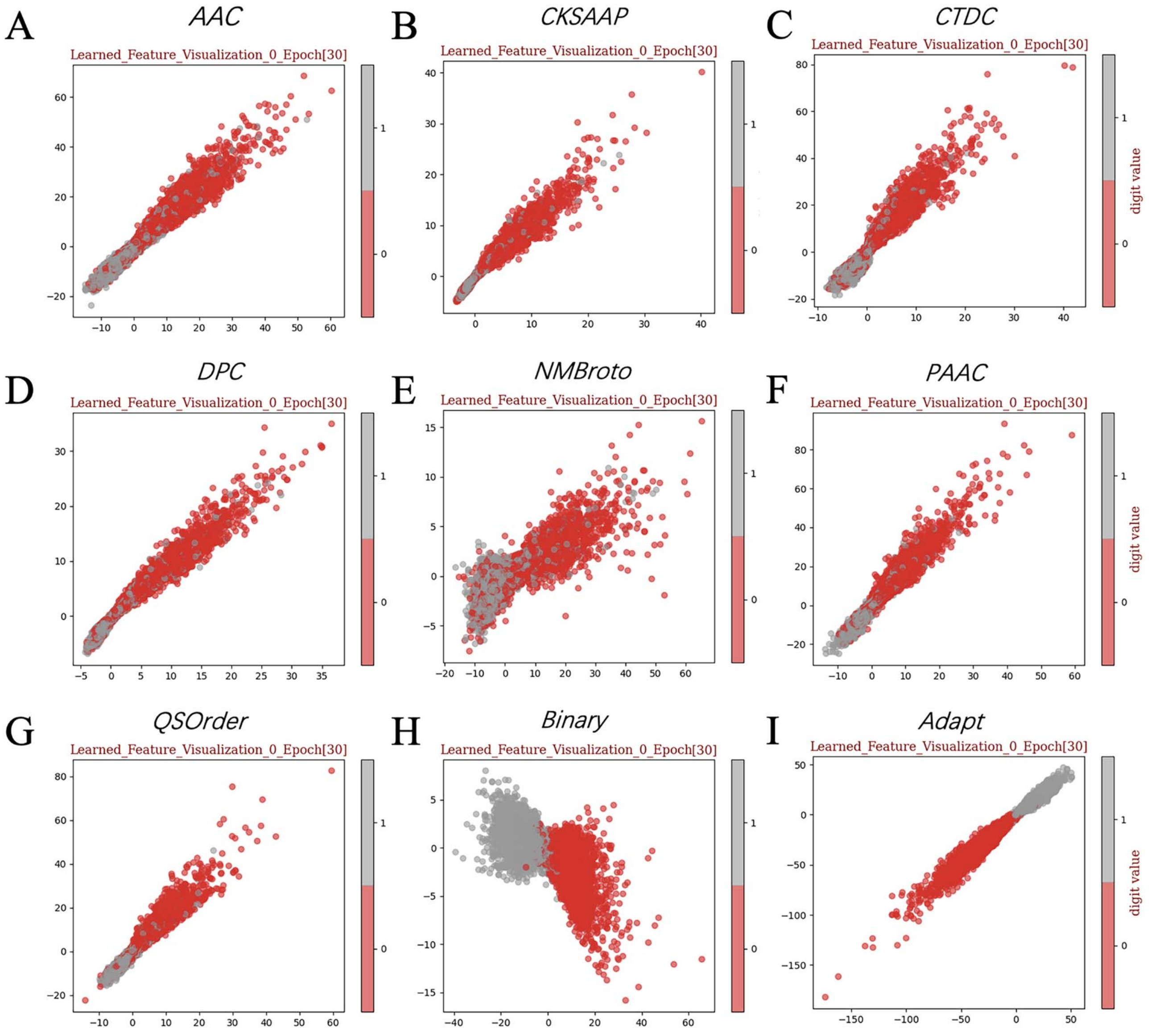

Figure 3. Visualization of learned features between adaptive embedding and eight handcrafted feature encodings in the Kcr dataset. 0 and 1 represent non-crotonylation and crotonylation, respectively, in a training set. A-I show visualizations for Kcr at epoch 30 with AAC, CKSAAP, CTDC, DPC, NMBroto, PAAC, QSOrder, Binary and Adapt, respectively.

Table 1. Comparison of performance between different deep learning network architectures using 10-fold cross-validation

Model	SPE(%)	SEN(%)	PRE(%)	ACC(%)	MCC	AUROC
CNN	88.02	88.38	88.13	88.20	0.7645	0.9552
CNN–LSTM	87.37	**91.00**	87.85	89.18	0.7813	0.9600
CNN–LSTM–Att	**88.87**	90.76	**89.09**	**89.82**	**0.7966**	**0.9636**
LSTM	85.83	89.26	86.30	87.54	0.7526	0.9474
LSTM–Att	85.51	90.36	86.22	87.94	0.7601	0.9449

the BERT-Kcr model on the S/T phosphorylation data (BERT-ST), and compared several models recently published using this dataset (Table 4). The BERT-ST model performance was very close to that of the state-of-the-art model, DeepIPS. However, Adapt-ST surpassed even DeepIPS on the S/T data set; the ACC, MCC, SPE, SEN and AUROC values of Adapt-ST were 2.7%, 3.5%, 2.2%, 1.3% and 1.8% higher, respectively, compared to DeepIPS. This result suggests that learning embedding features and attention architecture (i.e. our Adapt framework) has a strong potential for application in other PTM prediction tasks.

Table 2. Performance evaluation of the Kcr site prediction tools using an independent testing set. To facilitate understanding, the highest value in each column is shown in bold

Model	SPE	SEN	PRE	ACC	MCC	AUROC
Adapt-Kcr	0.849	**0.854**	**0.850**	**0.852**	**0.706**	**0.924**
BERT-Kcr	0.838	0.801	0.832	0.820	0.640	0.905
Deep-Kcr	**0.871**	0.630	0.830	0.751	0.516	0.859

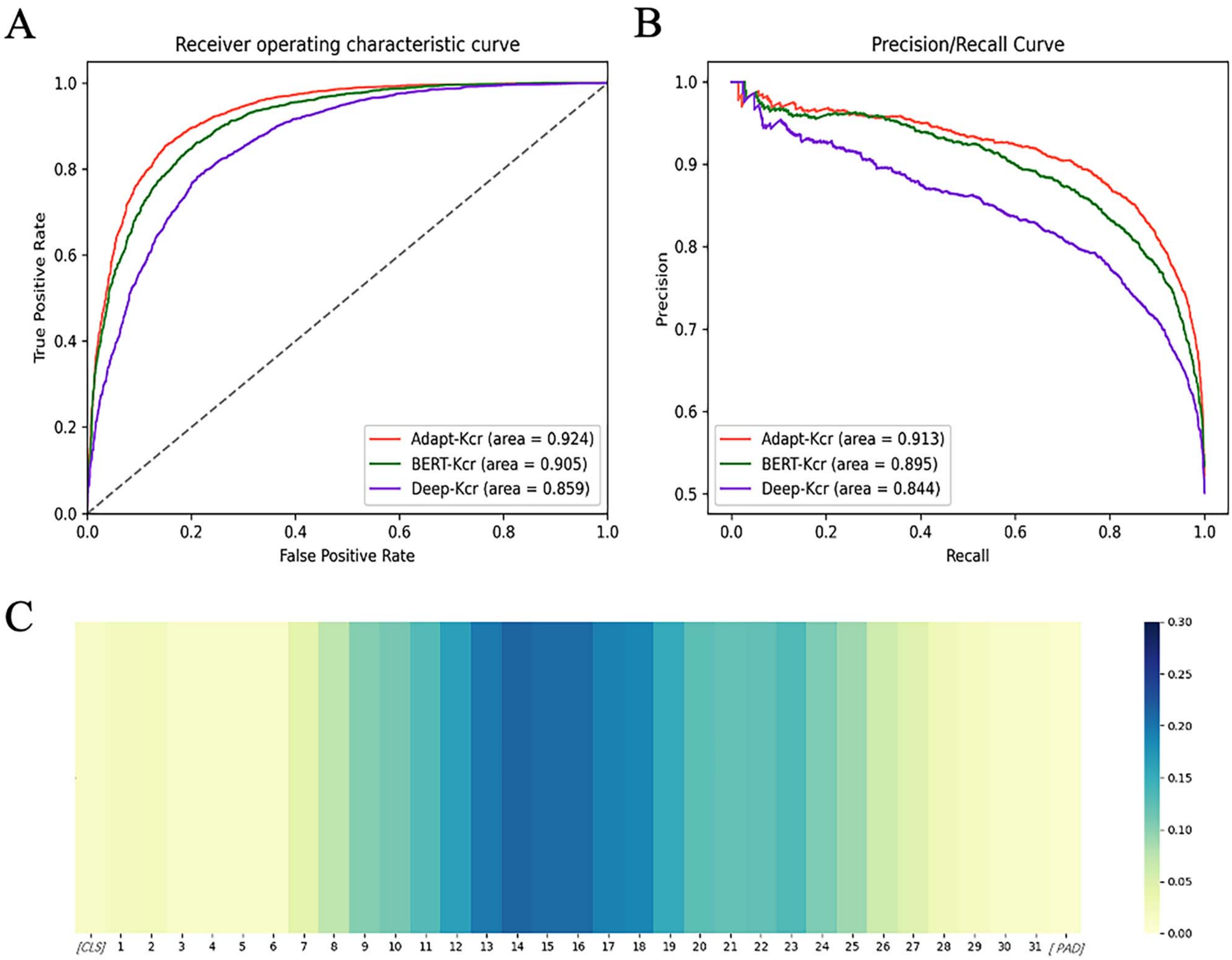

Figure 4. Performance analysis of Kcr models. (**A**) receiver operating characteristic curves of Adapt-Kcr based on the Kcr dataset. (**B**) Precision-recall curves of Adapt-Kcr based on the Kcr dataset. (**C**) Characterization of the attention vectors of Adapt-Kcr for predicting Kcr sites.

Table 3. Performance evaluation of Kcr site prediction models in distinguishing between Kcr sites and other types of lysine modification (glycation and acetylation) sites

Methods	SPE	SEN	PRE	ACC	MCC	AUROC
Adapt-Kcr	0.514	**0.854**	**0.639**	**0.685**	**0.393**	**0.767**
BERT-Kcr	**0.567**	0.801	0.578	0.667	0.370	0.758
Deep-Kcr	0.376	0.630	0.428	0.484	0.006	0.497
Position_weight	0.268	0.725	0.423	0.463	-0.007	0.493
CKSAPP_CrotSite	0.208	0.853	0.443	0.482	0.078	0.592
LightGBM-CrotSite	0.477	0.806	0.533	0.617	0.292	0.697

To facilitate understanding, the highest value in each column is shown in bold.

In addition, ROC and PR curves were plotted to demonstrate the performance of Adapt-ST, DeepIPS and BERT-ST (Figure 5). The Adapt-ST model had much higher performance than the other methods, further demonstrating the stability and generalization ability of the Adapt framework.

Table 4. Performance evaluation of S/T site prediction tools using an independent testing set

Methods	SPE(%)	SEN(%)	ACC(%)	MCC	AUROC
Adapt-ST	**85.72**	80.90	**83.32**	**0.667**	**0.912**
DeepIPs	83.50	79.61	80.63	0.632	0.894
Bert-ST	74.60	80.07	79.84	0.600	0.889
DeepPSP	83.78	76.65	80.21	0.606	0.876
MusiteDeep2020	78.96	**82.95**	80.95	0.620	0.887
MusiteDeep2017	81.46	78.87	80.17	0.604	0.880

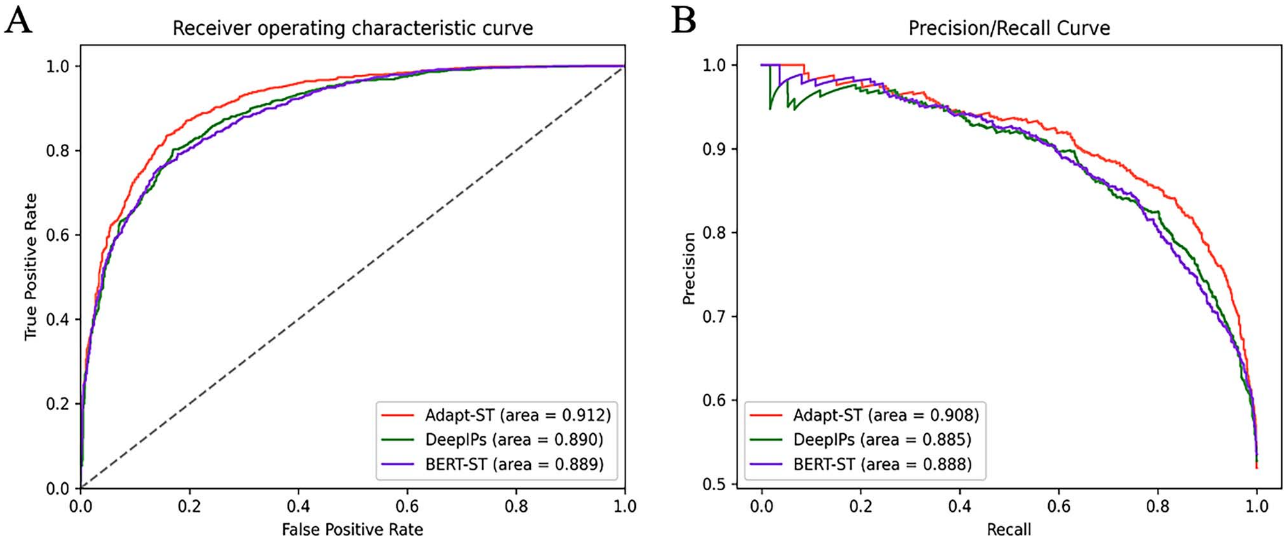

Figure 5. ROC (**A**) and PR (**B**) curves for Adapt-ST on the S/T dataset.

Discussion and Conclusions

In this study, we constructed an end-to-end deep learning-based model, Adapt-Kcr, that integrates several deep learning methods to efficiently predict Kcr sites. Adapt-Kcr uses an adaptive embedding layer to characterize protein sequence information, followed by a CNN layer to extract local protein sequence characterization, then a BLSTM layer to capture context-dependency information of Kcr sites. Adapt-Kcr also uses an attention mechanism to select the information that is most critical to predicting Kcr sites, followed by a FC layer to make the final determination for each site. Experimental results showed that this model outperformed existing models in most metrics on the benchmark datasets, demonstrating higher ACC and robustness. Importantly, we found that this deep learning architecture with adaptive embedding features can also be applied to better predict other PTM data, such as serine and threonine phosphorylation sites, compared to existing methods. This framework thus shows good scalability and application potential for PTM prediction tasks. There are some limitations of this study that should be noted. The model performance on unbalanced data is slightly inferior to that on balanced data, which is a problem to be solved by bioinformatics data engineering and model architecture design. Due to the relative complexity of the calculation time, the framework and parameter design of Adapt-Kcr may only achieve a local optimum. In addition, the length of the protein sequence may limit the performance of the model; theoretically, a longer sequence provides more information. However, all previous studies on Kcr site recognition are based on sequences with a length of 31 nt. Our future work will incorporate longer sequence information and aim to build an effective machine learning framework for more single class PTM predictions or multi-label prediction tasks.

Key Points

- Adaptive learning features, which use position embedding and token embedding, are more effective than HF in the prediction of Kcr sites.
- We provide a new computing framework, termed Adapt-Kcr, to automatically capture local and long-range information from protein sequences using CNN and LSTM. The attention mechanism is employed to effectively capture the position information from protein sequences.
- Attention vectors of the trained model show that the region near modifications are weighted higher in the final predictions.
- The learning embedding features and attention architecture (the adaptive framework) has a strong potential for application in other PTM prediction tasks.

Authors' contributions

Zutan Li, Jingya Fang and Cong Pian conceived and designed the study; Zutan Li, Jingya Fang and Shining Wang performed the experiments, analyzed the data, prepared figures and tables; Liangyun Zhang, Yuanyuan Chen and Cong Pian authored or reviewed drafts of the paper, and approved the final draft.

Supporting information

Source code, all training data and testing data of Adapt-Kcr are found at https://github.com/Marscolono/Adapt-Kcr

Funding

The Fundamental Research Funds for the Central Universities (No. JCQY202108) and Startup Foundation for Advanced Talents at Nanjing Agricultural University (No. 050/804009).

References

1. Soffer RL. Post-translational modification of proteins catalyzed by aminoacyl-tRNA-protein transferases. *Mol Cell Biochem* 1973;**2**(1):3–14.
2. Wold F. In vivo chemical modification of proteins (post-translational modification). *Annu Rev Biochem* 1981;**50**:783–814.
3. Fu J, Wu M, Liu X. Proteomic approaches beyond expression profiling and PTM analysis. *Anal Bioanal Chem* 2018;**410**(17):4051–60.
4. Huang H, Sabari BR, *et al.* Snap shot: histone modifications. *Cell* 2014;**159**(2):458–8.
5. Wang Y, Jin J, Chung MWH, *et al.* Identification of the YEATS domain of GAS41 as a pH-dependent reader of histone succinylation. *Proc Natl Acad Sci U S A* 2018;**115**(10):2365–70.
6. Ramazi S, Allahverdi A, Zahiri J. Evaluation of post-translational modifications in histone proteins: a review on histone modification defects in developmental and neurological disorders. *J Biosci* 2020;**45**:135.
7. Krishna RG, Wold F. Post-translational modification of proteins. *Adv Enzymol Relat Areas Mol Biol* 1993;**67**:265–98.
8. Lee TY, Huang HD, Hung JH, *et al.* dbPTM: an information repository of protein post-translational modification. *Nucleic Acids Res* 2006;**34**:D622–7.
9. 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**(6):1016–28.
10. Fellows R, Denizot J, Stellato C, *et al.* Microbiota derived short chain fatty acids promote histone crotonylation in the colon through histone deacetylases. *Nat Commun* 2018;**9**(1):105.
11. Huang H, Zhang D, Wang Y, *et al.* Lysine benzoylation is a histone mark regulated by SIRT2. *Nat Commun* 2018;**9**(1):3374.
12. Jiang G, Nguyen D, Archin NM, *et al.* HIV latency is reversed by ACSS2-driven histone crotonylation. *J Clin Invest* 2018;**128**(3): 1190–8.
13. Liu S, Yu H, Liu Y, *et al.* Chromodomain protein CDYL acts as a crotonyl-CoA hydratase to regulate histone crotonylation and spermatogenesis. *Mol Cell* 2017;**67**(5):853–866e855.
14. Ruiz-Andres O, Sanchez-Niño MD, Cannata-Ortiz P, *et al.* Histone lysine crotonylation during acute kidney injury in mice. *Dis Model Mech* 2016;**9**(6):633–45.
15. Yu H, Bu C, Liu Y, *et al.* Global crotonylome reveals CDYL-regulated RPA1 crotonylation in homologous recombination-mediated DNA repair. *Sci Adv* 2020;**6**:eaay4697.
16. Qiu WR, Sun BQ, Tang H, *et al.* Identify and analysis crotonylation sites in histone by using support vector machines. *Artif Intell Med* 2017;**83**:75–81.
17. Ju Z, He JJ. 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.
18. 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.
19. Lv H, Dao FY, Guan ZX, *et al.* Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method. *Brief Bioinform* 2020;**22**(4):bbaa255.
20. Qiao Y, Zhu X, Gong H. BERT-Kcr: prediction of lysine crotonylation sites by a transfer learning method with pre-trained BERT models. *Bioinformatics* 2021;**38**:648–654.
21. 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.
22. Stukalov A, Girault V, Grass V, *et al.* Multi-level proteomics reveals host-perturbation strategies of SARS-CoV-2 and SARS-CoV. *Nature* 2021;**594**:246–52.
23. Greff K, Srivastava RK, Koutnik J, *et al.* LSTM: a search space odyssey. *IEEE Trans Neural Netw Learn Syst* 2017;**28**:2222–32.
24. Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. *arXiv preprint*, arXiv:1706.03762, 2017.
25. Lin Z, Feng M, Santos C, *et al.* A structured self-attentive sentence embedding. arXiv preprint, arXiv:1703.03130v1, 2017.
26. Zhong R, Shao S, Mckeown K. Fine-grained sentiment analysis with faithful attention. arXiv preprint, arXiv:1908.06870v1, 2019.
27. Wiegreffe S, Pinter Y. Attention is not not explanation. arXiv preprint, arXiv:1908.04626v1, 2019.
28. Clark K, Khandelwal U, Levy O, *et al.* What does BERT look at? An analysis of BERT's attention. arXiv preprint, arXiv:1906.04341v1, 2019.
29. Htut PM, Phang J, Bordia S, *et al.* Do attention heads in BERT track syntactic dependencies?. arXiv preprint, arXiv: 1911.12246v1, 2019.
30. Li H, Tian S, Li Y, *et al.* Modern deep learning in bioinformatics. *J Mol Cell Biol* 2021;**12**(11):823–7.
31. Park S, Koh Y, Jeon H, *et al.* Enhancing the interpretability of transcription factor binding site prediction using attention mechanism. *Sci Rep* 2020;**10**(1):13413.
32. Zou Z, Tian S, Gao X, *et al.* mlDEEPre: multi-functional enzyme function prediction with hierarchical multi-label deep learning. *Front Genet* 2019;**9**:714.
33. Hong Z, Zeng X, Wei L, *et al.* Identifying enhancer-promoter interactions with neural network based on pretrained DNA vectors and attention mechanism. *Bioinformatics* 2020;**36**(4):1037–43.
34. Kingma D, Ba J. Adam: a method for stochastic optimization. *In: International Conference on Learning Representations, San Diego, CA, USA,* OpenReview.net. International Conference on Representation Learning, La Jolla, CA, USA. 2015.
35. Srivastava N, Hinton G, Krizhevsky A, *et al.* Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014;**15**(1):1929–58.
36. 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. *Brief Bioinform* 2020;**21**(3):1047–57.