BMC Genomics

RESEARCH

Open Access

DeepHLAPred: a deep learning-based method for non-classical HLA binder prediction

Guohua Huang[1,2*], Xingyu Tang[2] and Peijie Zheng[2]

Abstract

Human leukocyte antigen (HLA) is closely involved in regulating the human immune system. Despite great advance in detecting classical HLA Class I binders, there are few methods or toolkits for recognizing non-classical HLA Class I binders. To fill in this gap, we have developed a deep learning-based tool called DeepHLAPred. The DeepHLAPred used electron-ion interaction pseudo potential, integer numerical mapping and accumulated amino acid frequency as initial representation of non-classical HLA binder sequence. The deep learning module was used to further refine high-level representations. The deep learning module comprised two parallel convolutional neural networks, each followed by maximum pooling layer, dropout layer, and bi-directional long short-term memory network. The experimental results showed that the DeepHLAPred reached the state-of-the-art performanceson the cross-validation test and the independent test. The extensive test demonstrated the rationality of the DeepHLAPred. We further analyzed sequence pattern of non-classical HLA class I binders by information entropy. The information entropy of non-classical HLA binder sequence implied sequence pattern to a certain extent. In addition, we have developed a user-friendly webserver for convenient use, which is available at http://www.biolscience.cn/DeepHLApred/. The tool and the analysis is helpful to detect non-classical HLA Class I binder. The source code and data is available at https://github.com/tangxingyu0/DeepHLApred.

Keywords Non-classical HLA class I, Deep learning, Representation, Information entropy, Convolutional neural network

Introduction

Human leukocyte antigen (HLA) genes are located at the human histocompatibility complex (MHC) region on the short arm of chromosome 6 [1, 2]. HLA genes have more than one different allele, which are encoded into cell-surface glycoproteins which play a key role in the immune system [3, 4]. Generally, HLA genes are classified into three categories, class I, class II, and class III [5], while HLA class I genes are further divided into two subcategories: classical (HLA-A, HLA-B, HLA-C) and non-classical (HLA-E, HLA-G, HLA-F) [6]. As of Feb 2023, the IPD-IMGT/HLA database deposited 25,228 HLA Class I alleles, including 7712 HLA-A, 9164 HLA-B, 7672 HLA-C, and 10,592 HLA Class II alleles [7, 8]. The non-classical HLA class I genes are different from classical I ones in a wide range of respects including specific patterns of transcription, protein expression, and immunological functions [9]. For example, non-classical HLA class I genes are less polymorphic than classical, characterized by a low genetic diversity and by a particular expression pattern, structural organization, and functional profile [10–13].

An adaptive immune response was activated by binding of peptides from antigenic pathogens to HLA and

*Correspondence:
Guohua Huang
guohuahhn@163.com
[1] School of Information Technology and Administration, Hunan University of Finance and Economics, Changsha, Hunan 410215, China
[2] College of Information Science and Engineering, Shaoyang University, Shaoyang, Hunan 422000, China

then eliminated the source pathogens [14]. Therefore, identifying the HLA binding peptides not only helps understand the immune mechanism, but also facilitates rational subunit vaccine design. However, this is still a bottleneck to precisely recognize the non-classical HLA binders at present [15]. Hannoun et al. employed the biochemical methodology to identify 4 HIV-1-derived HLA-E-binding peptides in assays [16]. This methodology is very complex, time-consuming, and laborious [17]. Over the recent twenty years, computational methods have attracted more attention due to simplicity and effectiveness. No less than ten computational methods have been proposed for predicting HLA binders [15, 18–25].

In 1993, Bisset et al. employed the neural network to determine HLA-DR1 binding peptides [18]. Trained by the peptide segments known to bind to HLA-DR1, the neural network was able to learn representations relating to HLA-DR1-binding capacity to a certain extent. Singh et al. developed a graphical web tool to identify HLA-DR binder [15] and an online web tool to predict peptides binding to MHC class-I alleles [19]. Nielsen et al. utilized the stabilization matrix method to develop a quantitative MHC class II binding prediction [26]. Lata et al. created a support vector machine-based method for prediction of promiscuous binders of MHC class II alleles [27]. Wang et al. combined multiple machine learning algorithms to explore HLA-peptide binding affinities for HLA DR, DP, and DQ alleles [28]. Peters et al. set up a benchmark dataset for detecting peptide binding to MHC-I alleles, and compared the neural network-based and two matrix-based predictions [29]. Lin et al. compared and evaluated thirty prediction servers for seven human MHC-I molecules and argued that non-linear predictors were superior to matrix-based ones [30]. Nielsen et al. developed a pan-specific HLA-DR prediction [31], while Jurtz et al. fused the eluted ligand and peptide binding affinity data to promote prediction of peptide-MHC class I interaction [20]. Most of computational methods above were based on the traditional machine learning (shallow learning), which were restricted to the small number of leaning samples. The generalization ability of the model was sometimes not as good as expected. Ye et al. [22] employed long short-term memory (LSTM) and multiple head attentions to build a deep learning-based method (MATHLA) for classical HLA class I binding peptide prediction. The MATHLA showed the improved accuracy of prediction for HLA-C alleles and depicted some HLA-ligand binding patterns [22]. Zhang et al. proposed a complex model (HLAB) for HLA class I binding peptide prediction [23]. The HLAB used the pre-trained Protein Bidirectional Encoder Representations (ProBERT) [32] to extract initial representations from peptides,, which is a BERT model [33–35] trained by the protein sequences from the UniRef100 [36] as well as BFD [37] databases then employed bi-directional LSTM (Bi-LSTM) to refine contextual semantics, utilized the Umap [38] to reduce the dimensions, and finally built seven binary classification models. Chu et al. [24] proposed a transformer-based method for peptide-HLA binding prediction. The experiments showed superior performance over 14 state of art methods.

More attentions were paid to classical HLA genes than non-classical HLA class I genes in the past ten years [39]. However, the recent studies have demonstrated that non-classical HLA class I alleles play equally important roles in transcription, protein expression, and immune regulation [9, 13, 40–45]. To best of our knowledge, only the HLAncPred [6] was explicitly intended to predict binders for non-classical HLA class I alleles. The HLAncPred was a feature engineering and traditional machine learning-based method, which used different machine learning algorithms with different representations to construct the predicting models. Although the HLAncPred obtained the quite high performance, it was inconvenient to choose a specified model for multiple-type datasets. Hence, it is necessary to develop a more efficient method for non-classical HLA binder prediction. Here, we developed a deep learning-based method for non-classical HLA binder prediction, called DeepHLAPred. The DeepHLAPred first extracted initial representations of non-classical HLA binding and non-binding peptide sequences by three encoding methods, and then fed them into an embedding layer followed by a deep leaning module which consisted of two parallel sequences. Each sequence comprised mainly convolutional neural network (CNN) at different scale and Bi-LSTM. The two fully connected layers were attached to the deep leaning module for the decision. To validate the effectiveness and efficiency of the DeepHLAPred, we tested it extensively on the balanced, the unbalanced, and the independent datasets.

Materials and methods

Materials

Adequate and reliable data is crucial for building a robust predictive model. We used the non-classical class I HLA binding peptides collected by Dhall et al. [6] as the benchmark datasets. All the binding peptides were experimentally validated by the fluorescence-based, and the mass spectrometry or the X-ray crystallography, which were of 8 to 15 amino acid residues. Dhall et al. [6] grouped the peptides into two categories: the balanced and the imbalanced, each with five datasets. In the balanced category, each dataset included the equal numbers of the positive and the negative samples, while the number of the negative samples was ten times more than the number of

positive ones for each dataset in the imbalanced category. The positive samples were identical for both the balanced and the imbalanced category. The binding peptides (positive samples) for HLA-E∗01:01, HLA-E ∗01:03, HLA-G∗01:01, HLA-G∗01:03, and HLA-G∗01:04 alleles were 142, 632, 2633, 751, and 812, respectively. Peptides of all the binders were downloaded from the website: https://webs.iiitd.edu.in/raghava/hlancpred.

DeepHLAPred framework

Figure 1 showed the schematic framework of DeepHLAPred. The binding peptides were first encoded by electron-ion interaction pseudo potential (EIIP), integer numerical mapping (INM), and accumulated amino acid frequency (AAAF), which then passed through the embedding layer. Two parallel CNNs were employed to further refine high-level abstract information, each followed by max pooling, by Batch Normalization, by Dropout, and by Bi-LSTM. The Bi-LSTM was intended to learn the dependency relationship in the peptides. Lastly, the fully connected layer was attached to the Bi-LSTM layer. The sigmoid activation function was used for decision in the last fully connected layer, which outputted a probability value between 0 and 1. If the probability value was greater than 0.5, it was determined as non-classical HLA class I binders, and otherwise it was non-classical HLA non-binders. The detailed model parameters were shown in the Supplementary Table 1. The formula of the sigmoid function was expressed as:

$$Sigmoid(x) = \left(1 + e^{-x}\right)^{-1} \quad (1)$$

EIIP

The EIIP was defined as the energy of delocalized electrons of amino acid [46], which is one of the most important physical property of amino acid. We used the EIIP to encode each amino acid (Table 1). For example, the peptide sequence "CEFSQC" was encoded by the EIIP into (0.08292, 0.00580, 0.09460, 0.08292, 0.07606, 0.08292). The EIIP of a peptide reflected the distribution of the free electron energies.

INM

In order to solve the problem of sparse dimension caused by one-hot encoding, we assigned different positive integer values to twenty amino acids (Table 1). We used MathFeature [47] to compute the INM. The MathFeature is a python package which is able to compute up to 37 categories of representations for DNA, RNA or protein sequences. For example, the sequence "CEFSQC" was mapped into a numeric vector (5, 7, 14, 16, 6, 5).

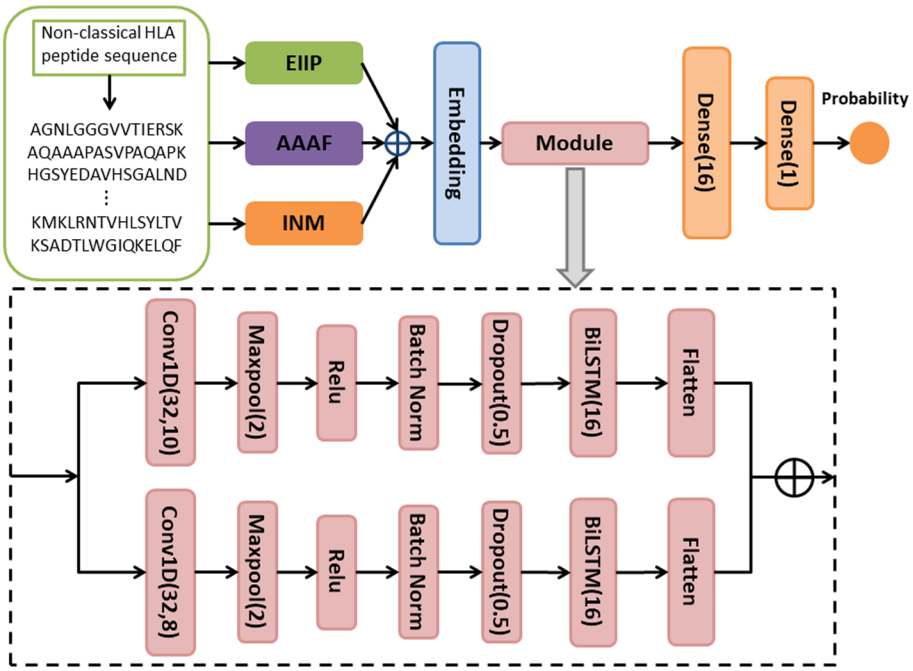

Fig. 1 The flowchart of DeepHLAPred. Dense stands for fully-connected layer. The numbers in the bracket represent value of corresponding parameters

Table 1 The EIIP and INM value of each amino acid

Amino Acid	EIIP	INM	Amino Acid	EIIP	INM
Alanine(A)	0.37100	1	Leucine(L)	0.00000	11
Arginine(R)	0.95930	2	Lysine(K)	0.37100	12
Asparagine(N)	0.00359	3	Methionine(M)	0.08226	13
Asparticacid(D)	0.12630	4	Phenylalanine(F)	0.09460	14
Cystine(C)	0.08292	5	Proline(P)	0.01979	15
Glutarnine(Q)	0.07606	6	Serine(S)	0.08292	16
Glutamicacid(E)	0.00580	7	Threonine(T)	0.09408	17
Glycine(G)	0.00499	8	Tryptophan(W)	0.05481	18
Histidine(H)	0.02415	9	Tyrosine(Y)	0.05159	19
Isoleucine(I)	0.00000	10	Valine(V)	0.00569	20

AAAF

The AAAF [47] reflected the distribution density of amino acid in a protein sequence. Assuming a non-classical HLA Class I binding peptide sequence $S = s_1 s_2 \cdots s_n$, where n denoted the length of the sequence S. The AAAF was computed by

$$f(s_j) = \frac{1}{j} \sum_{t=1}^{j} T(s_t) \tag{2}$$

$$T(s_t) = \begin{cases} 1, & s_t = s_j \\ 0, & s_t \neq s_j \end{cases} \tag{3}$$

A peptide sequence of n residues was of n dimensional AAAF feature. For example, the AAAF of the sequence "CEFSQC" was (1.0000, 0.50000, 0.33333, 0.25000, 0.20000, 0.33333). We also used the MathFeature [47] to compute the AAAF.

CNN

The CNN is a feed-forward neural network [48, 49] that is one of the most popular algorithms in the area of deep learning. It significantly reduces the number of training parameters [48, 50]. The CNN consists mainly of convolutional and pooling operation. The convolutional operation is called also the filter operation. In order to refine multiple-view representations, the CNN uses more than a filter (kernel). The pooling operation is a down-sampling technique, which reduces computations and overfitting to a certain extent. Compared with traditional neural networks, the CNN is characterized by weight sharing and local connectivity. Over the past decades, CNN has achieved remarkable success in various fields, such as medical image analysis [51, 52], speech recognition [53], target detection [54], natural language processing [55–58]. We applied two parallel one-dimensional convolutional operations which are of different scale. One was with the kernel size of 10 and another was with the kernel size of 8. The max pooling operation with a pooling window size of 2 was attached to the corresponding convolution. RELU was used as the activation function. The batch normalization and the dropout were used to reduce overfitting. The dropout rate was set to 0.5.

Bi-LSTM

The LSTM is actually a kind of recurrent neural network (RNN), which is of gate mechanism [59–61]. Each repeated module in the common LSTM consists of the input gate, the output gate, forget gate and the cell state. At the heart of LSTM is the cell state, which preserves previous record. The forget gate determines what information of previous state cell is forgot or remembered. The input gate determines what new information is added to the cell state. The candidate value is created by the tanh function. The forget gate, the candidate value and the input gate jointly update the cell state. The hidden state is updated by the output gate and the cell state. The LSTM well solve the long-term dependence, gradient vanishing, or gradient exploding problems [62–64]. The Bi-LSTM captures bidirectional relationship between words (token). In this study, we used the Bi-LSTM.

Model evaluation

We used the following five evaluation metrics: SN(sensitivity), SP(specificity), ACC (accuracy), MCC (Matthews correlation coefficient) to measure the performance [65, 66]. Their formulas were expressed as:

$$SN = \frac{T_P}{T_P + F_N} \tag{4}$$

$$SP = \frac{T_N}{T_N + F_P} \tag{5}$$

$$ACC = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{6}$$

$$MCC = \frac{T_P \times T_N - F_P \times F_N}{\sqrt{(T_P + F_N)(T_P + F_P)(T_N + F_P)(T_N + F_N)}} \tag{7}$$

In addition, we used ROC curves (receiver operating characteristic curves) to visualize the performance. The ROC curve is to link true positive rate (TPR) against false positive rate (FPR) under various threshold. TPR and FPR were defined by

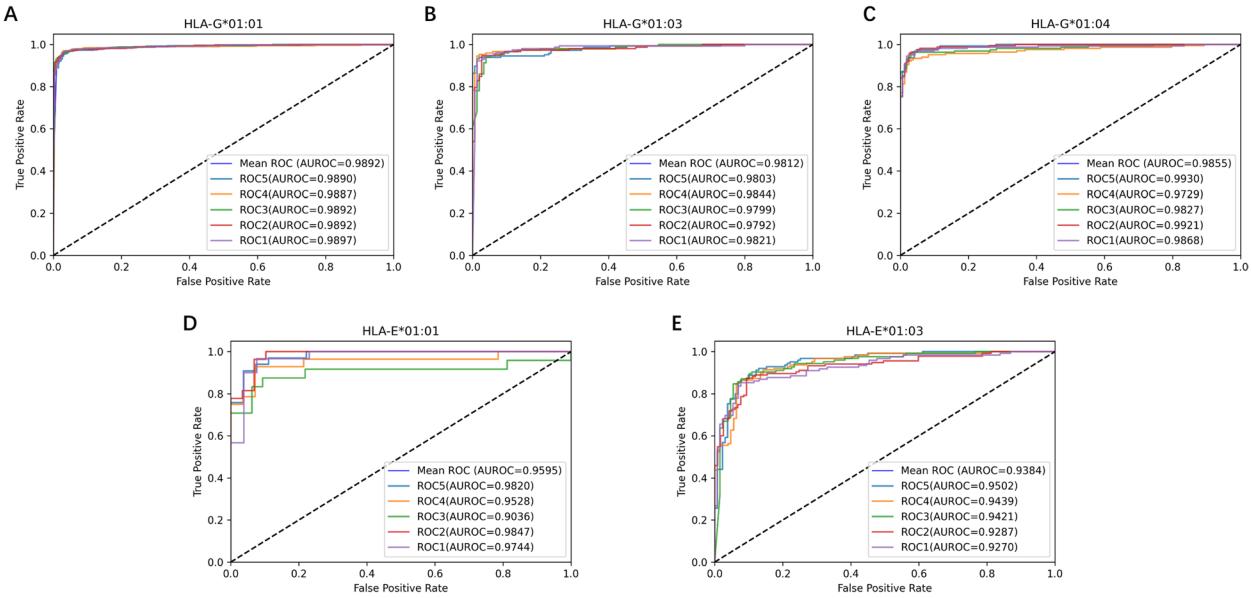

Fig. 2 The ROC curves and AUC values on the five-fold cross validation

$$TPR = \frac{T_P}{T_P + F_N} \tag{8}$$

$$FPR = \frac{F_P}{F_P + T_N} \tag{9}$$

The area under the ROC curve (AUC) was employed to quantitatively assess performance. In the above equations, T_P, T_N, F_P, and F_N were denoted as true positive (number of samples correctly as positive), true negative (number of samples correctly predicted as negative), false positive (number of samples incorrectly predicted as positive), and false negative (number of samples incorrectly predicted as negative), respectively.

Results and discussions

Cross validation on the balanced category

We conducted five-fold cross-validation on five balanced datasets (HLA-G*01:01, HLA-G*01:03, HLA-G*01:04, HLA-E*01:01, HLA-E*01:03) to examine the DeepHLAPred. Five-fold cross-validation is to randomly split the dataset into five parts, of which four parts are used for training the model and the other is used for testing the model. The training and testing process is repeated five times to ensure that each is trained four times and tested only a time. As shown in Fig. 2, the DeepHLAPred achieved excellent performance, with AUC reaching 98.92%, 98.12%, 98.55%, 95.95%, and 93.84% on five datasets of HLA-G*01:01, HLA-G*01:03, HLA-G*01:04, HLA-E*01:01, and HLA-E*01:03, respectively. For intuitively contrasting the DeepHLAPred to the HLAncPred which is the latest method for non-classical HLA Class I binder prediction, we draw histograms of SN, SP, ACC, MCC, and AUC (Fig. 3). Except for the SN on the datasets HLA-G*01:04 and HLA-E*01:03, and the AUC on the datasets HLA-G*01:01 and HLA-E*01:01, the DeepHLAPred obviously outperformed the HLAncPred. The DeepHLAPred improved SN by 1.70%, SP by 1.02%, ACC by 1.37%, and MCC by 3.05% on the dataset HLA-G*01:01. The DeepHLAPred increased SN by 5.21%, SP by 2.22%, ACC by 3.72%, MCC by 6.83%, and AUC by 1.12% on the dataset HLA-G*01:03. The DeepHLAPred promoted SP by 2.43%, ACC by 0.48%, MCC by 0.79%, and AUC by 0.55% on the dataset HLA-G*01:04. The DeepHLAPred raised SN by 1.27%, SP by 3.92%, ACC by 2.79%, and MCC by 5.55% on the dataset HLA-E*01:01. The DeepHLAPred elevated SP by 8.61%, ACC by 2.35%, MCC by 3.31%, and AUC by 0.84% on the dataset HLA-E*01:03. We performed 5-fold cross validations 5 times and used T-test to compare difference between DeepHLAPred and the HLAncPred. As shown in Table 2, most metrics were significantly improved excluding AUC on the HLA-E*01:01, SN on the HLA-E*01:03, and SN on the HLA-G*01:04.

Validation on the imbalanced category

To further validate the effectiveness and efficiency of the DeepHLAPred, we amplified the numbers of negative samples ten times, which along with positive samples were called the imbalanced category (see the section Materials and methods). We shuffled samples

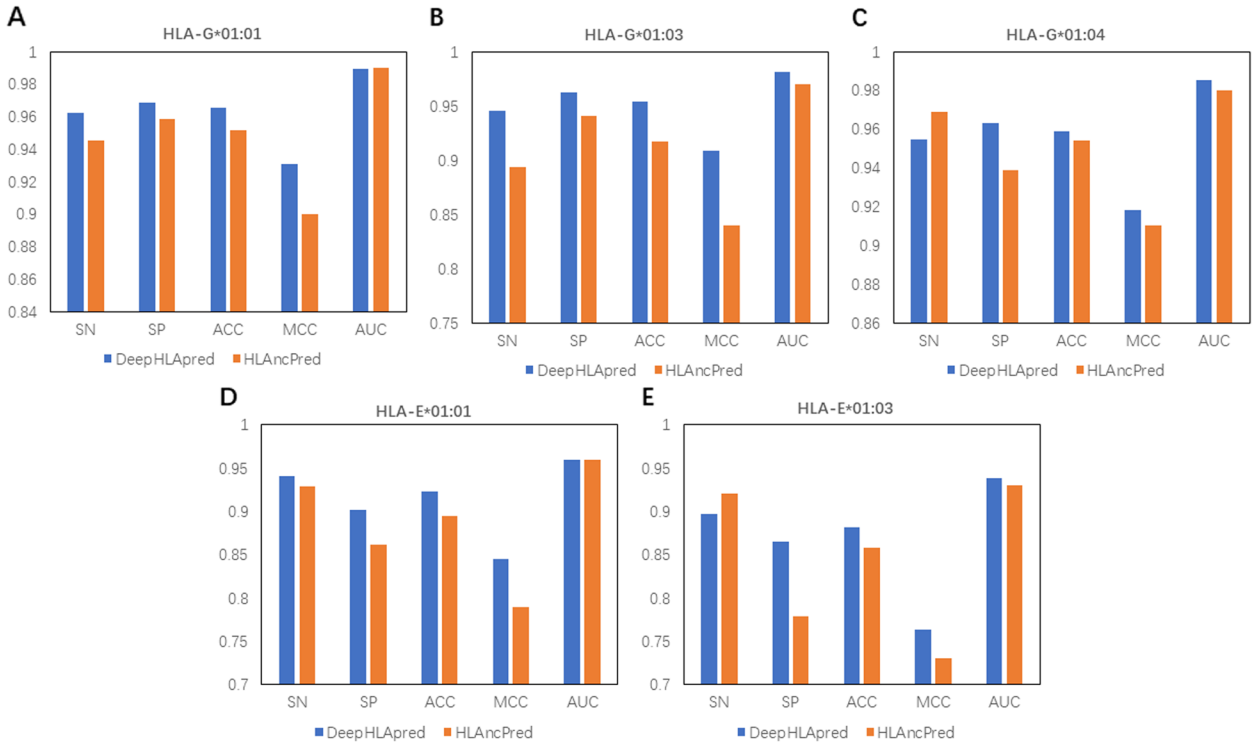

Fig. 3 Comparison with state-of-the-art methods on five-fold cross-validation in balanced datasets

Table 2 The *P*-values by T-test

P-Value	SN	SP	ACC	MCC	AUC
HLA-E*01:01	**0.0048**	**0.0036**	**0.00002**	**0.00004**	0.0610
HLA-E*01:03	0.9318	**0.0000004**	**0.00002**	**0.00003**	**0.0014**
HLA-G*01:01	**0.00000002**	**0.0004**	**0.000001**	**0.000001**	**0.0002**
HLA-G*01:03	**0.00000007**	**0.011**	**0.000007**	**0.000008**	**0.0002**
HLA-G*01:04	0.7370	**0.0025**	**0.0098**	**0.0105**	**0.0056**

in each dataset and randomly chose 10% samples for testing. We repeated this operation ten times. Figure 4 showed the ROC curves and the average ROC curves. The DeepHLAPred obtained the average AUC of 98.78% ± 0.003 on the HLA-G*01:01, 97.91% ± 0.003 on the HLA-G*01:03, 98.22% ± 0.005 on the HLA-G*01:04, 97.49% ± 0.013 on the HLA-E*01:01, and 94.69% ± 0.013 on the HLA-E*01:03 respectively. By contrast with Fig. 3, AUC was generally stable on the whole.

Comparison with the state-of-the-art methods

It's crucial to examine the performance of the DeepHLAPred on the independent datasets so as to objectively estimate its generalization ability. We retrieved 82 positive samples for HLA-E*01:01 and 67 positive ones for HLA-E*01:03 from the IEDB database [67], We randomly selected an equal number of negative samples from the imbalanced category, and none of these data were previously present in the training datasets. The positive along with negative samples constituted two independent datasets. We compared the DeepHLAPred with the state-of-the-art methods: HLAncPred ((https://webs.iiitd.edu.in/raghava/hlancpred) [6], MHCflurry 2.0 [21], NetMHCpan 4.1 (https://services.healthtech.dtu.dk/services/NetMHCpan-4.1/) [68]). As shown in the Table 3, DeepHLAPred demonstrated stable and excellent performance on the independent datasets. Although it was inferior to other three methods in terms of SP, DeepHLAPred exhibited greater stability in the prediction of different allele types, and it significantly outperformed MHCflurry 2.0 and NetMHCpan 4.1 in terms

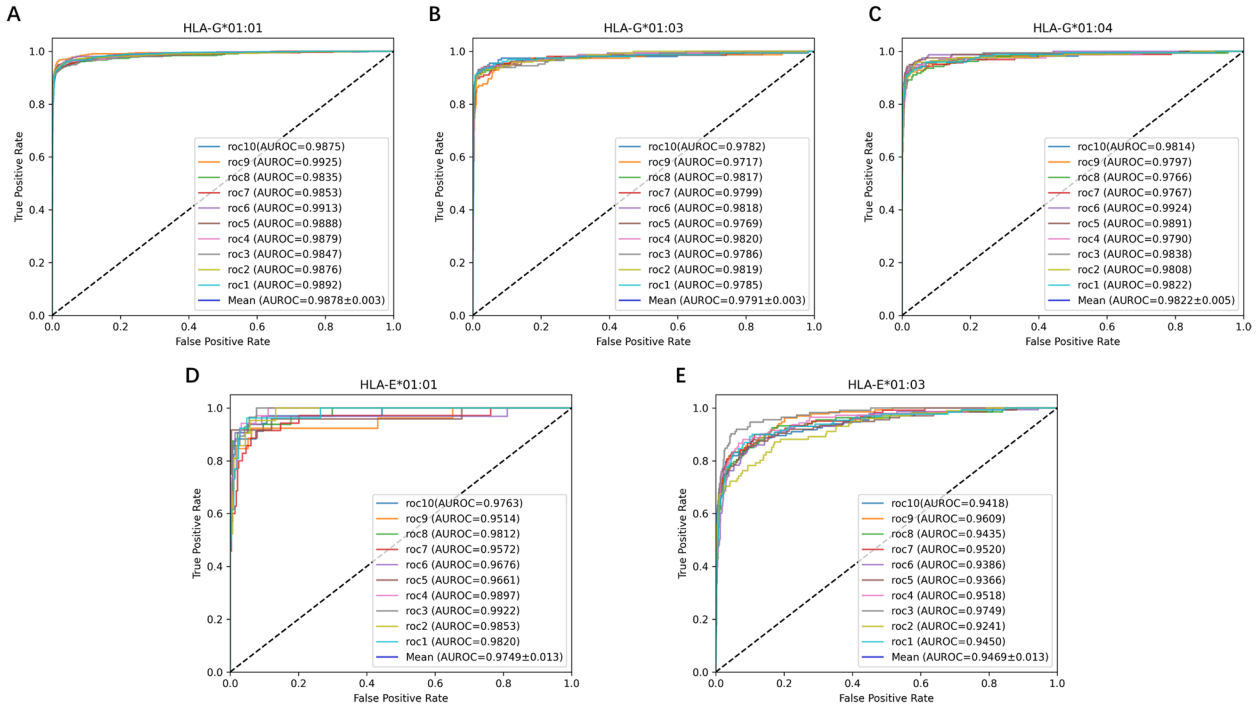

Fig. 4 The ROC curves of 10-times shuffle validation on the imbalanced category

Table 3 Comparisons with the state-of-the-art methods on independent datasets

Datasets	DeepHLAPred				HLAncPred				MHCflurry 2.0				NetMHCpan 4.1			
	SN	SP	ACC	MCC	SN	SP	ACC	MCC	SN	SP	ACC	MCC	SN	SP	ACC	MCC
HLA-E*01:01	0.7195	0.8780	0.7988	0.6052	0.5854	0.9268	0.7561	0.5449	0.2804	0.9634	0.6220	0.3339	0.4756	0.9390	0.7073	0.4679
HLA-E*01:03	0.9402	0.8507	0.8955	0.7942	0.9552	0.8358	0.8955	0.7967	0.1493	0.9552	0.5522	0.1765	0.3880	0.9851	0.6866	0.4651

of SN, ACC, and MCC. Compared to HLAncPred, DeepHLAPred achieved a notable improvement on the HLA-E*01:01 dataset, increasing SN by 13.41%, ACC by 4.27%, and MCC by 6.03%. On the HLA-E*01:03 dataset, DeepHLAPred achieved performance comparable to HLAncPred, with a slight decreased SN by 1.5% but an increase of 1.49% of SP. Additionally, ACC and MCC were very close between the two methods.

Discussion

Generally speaking, a single category of representation was inadequate to represent a protein sequence to full

Table 4 The performance of single representation and combinations on HLA-G*01:01

HLA-G*01:01	SN	SP	ACC	MCC	AUC
AAAF	0.6010	0.5382	0.5699	0.1397	0.5965
EIIP	0.8465	0.8304	0.8386	0.6774	0.9069
INM	0.9167	0.8529	0.8851	0.7720	0.9488
AAAF + EIIP	0.9102	0.8619	0.8859	0.7729	0.9532
AAAF + INM	0.9325	0.8504	0.8914	0.7856	0.9572
INM + EIIP	**0.9465**	0.9052	0.9253	0.8523	**0.9788**
INM + EIIP + AAAF	0.9407	**0.9145**	**0.9276**	**0.8557**	0.9762

Table 5 The performance of single representation and combinations on HLA-G*01:03

HLA-G*01:03	SN	SP	ACC	MCC	AUC
AAAF	0.5854	0.5440	0.5653	0.1299	0.5738
EIIP	0.8365	0.7814	0.8089	0.6190	0.8874
INM	0.8800	0.8298	0.8549	0.7107	0.9235
AAAF + EIIP	0.8525	0.8079	0.8296	0.6618	0.9116
AAAF + INM	0.9001	0.8083	0.8542	0.7115	0.9227
INM + EIIP	0.8990	**0.8685**	0.8835	0.7683	0.9537
INM + EIIP + AAAF	**0.9160**	0.8682	**0.8922**	**0.7852**	**0.9556**

Table 6 The performance of single representation and combinations on HLA-G*01:04

HLA-G*01:04	SN	SP	ACC	MCC	AUC
AAAF	0.5666	0.5646	0.5659	0.1315	0.5748
EIIP	0.8485	0.7964	0.8226	0.6462	0.9012
INM	0.8905	0.8470	0.8688	0.7389	0.9342
AAAF + EIIP	0.8630	0.8066	0.8350	0.6710	0.9079
AAAF + INM	0.9025	0.8419	0.8725	0.7471	0.9269
INM + EIIP	**0.9198**	0.8721	0.8960	0.7929	0.9547
INM + EIIP + AAAF	0.9137	0.**8875**	**0.9008**	**0.8022**	**0.9586**

Table 7 The performance of single representation and combinations on HLA-E*01:01.

HLA-E*01:01	SN	SP	ACC	MCC	AUC
AAAF	0.6595	0.6514	0.6559	0.3102	0.6815
EIIP	0.6848	0.7229	0.7041	0.4086	0.7781
INM	0.7977	0.7903	0.7917	0.5867	0.8534
AAAF + EIIP	0.7848	0.7251	0.7541	0.5099	0.8130
AAAF + INM	0.8447	0.7624	0.8029	0.6092	0.8430
INM + EIIP	**0.8458**	0.7807	**0.8144**	**0.6293**	**0.8512**
INM + EIIP + AAAF	0.8037	**0.7989**	0.7995	0.6022	0.8463

Table 8 The performance of single feature and combinations of features on HLA-E*01:03

HLA-E*01:03	SN	SP	ACC	MCC	AUC
AAAF	0.6150	0.5221	0.5689	0.1386	0.5867
EIIP	0.6870	0.6345	0.6605	0.3224	0.6979
INM	0.7059	0.6755	0.6946	0.3892	0.7392
AAAF + EIIP	0.7114	0.6560	0.6843	0.3687	0.7327
AAAF + INM	0.7217	0.6723	0.6969	0.3945	0.7481
INM + EIIP	0.7387	0.7109	0.7246	0.4498	0.7815
INM + EIIP + AAAF	**0.7451**	**0.7261**	**0.7358**	**0.4712**	**0.7901**

advantage. To validate this view, we experimented with single category of representation and their combination. As listed in Tables 4, 5, 6, 7 and 8, the INM performed best, followed by the EIIP, and the AAAF performed worst among the single category of representation. For example, the INM exceeded the AAAF by 31.52% ACC, the EIIP by 4.65% ACC on the dataset HLA-G*01:01. Difference in the performance between the EIIP and the INM was not too much. This indicated that the EIIP and INM better represent the peptide sequence. The combination of the AAAF, the INM and the EIIP reached the best ACC among the combination of any two and any single category of representation. This indicated that this combination enables complementation of different information.

In the context of deep learning, the embedding layer is primarily intended to transform high dimensional discrete inputs into low dimensional continuous vectors. The embedding layer captures the correlation within the inputs, reduces computational complexity, and enhance the generalization ability. Therefore, the embedding layer is popularly used in the deep learning model. Figure 5 showed the performance of the DeepHLAPred with the embedding layer and without the embedding layer. The inclusion of the embedding layer significantly improved performance on each dataset. Take for example the dataset HLA-E*01:03, the DeepHLAPred without the Embedding layer obtained an SN of 74.51%, SP of 72.61%, ACC of 73.58%, MCC of 47.12%, and AUC of 79.01%, respectively, while the DeepHLAPred with the embedding layer, reached SN of 89.71%, SP of 86.56%, ACC of 88.12%, MCC of 76.31% and AUC of 93.84%, respectively. The inclusion of the embedding layer improved SN by 15.20%, SP by 13.95%, ACC by 14.54%, MCC by 29.19%, and AUC by 14.83%, respectively. Similar phenomenon was observed in the other datasets.

The DeepHLAPred comprised mainly two scales of CNN and Bi-LSTM. To demonstrate the superiority of the DeepHLAPred, we compared it with models with a single CNN, a single Bi-LSTM, a CNN followed by Bi-LSTM, two paralleling CNNs with different scales, and two paralleling Bi-LSTMs, their performance were shown in Tables 9, 10, 11, 12 and 13. The DeepHLAPred reached the better performance on the five datasets. We found that a single CNN model or single Bi-LSTM model is not as good as the CNN + Bi-LSTM combination. The above results demonstrated the soundness of the DeepHLAPred architecture.

The discriminating ability of representations plays crucial roles in predictive performance. We used the Umap [38] to visualize the initial representations and the ones learned by the DeepHLAPred. As shown in Fig. 6, the

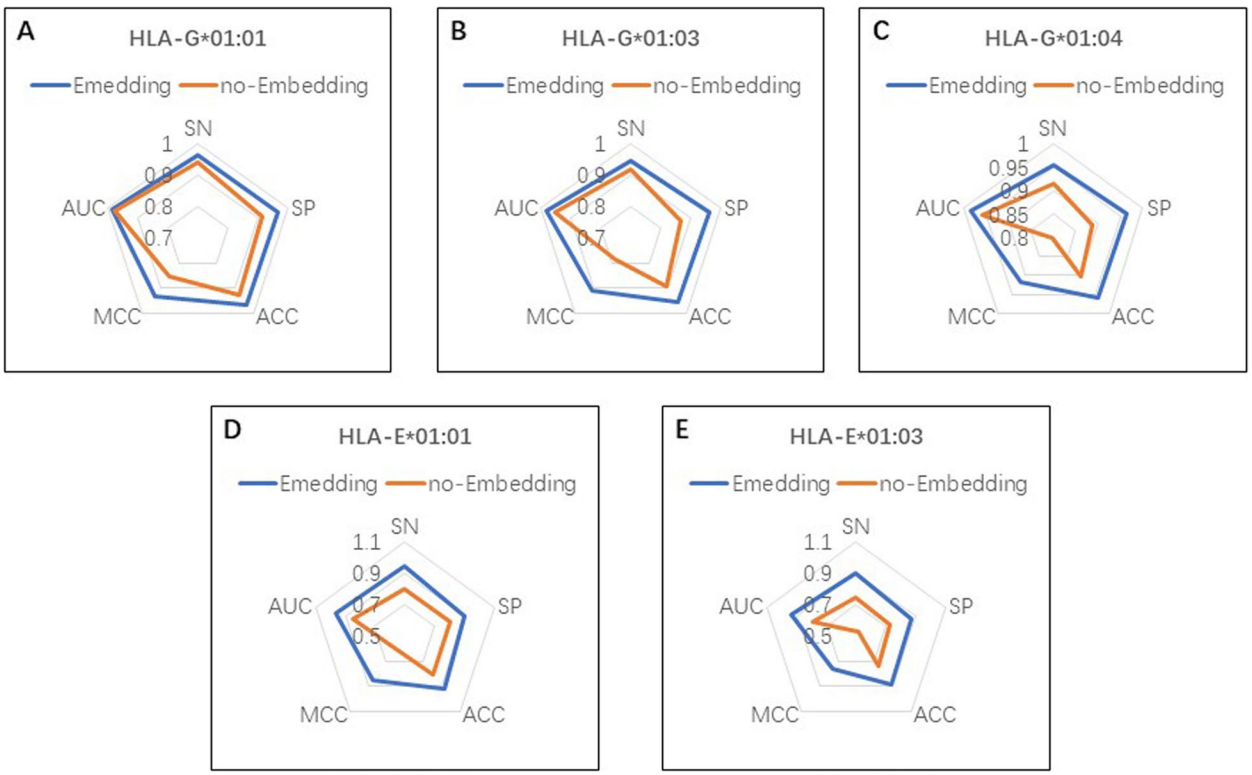

Fig. 5 The radar chart of the performance Embedding layer

Table 9 The performance of different modules on HLA-G*01:01 dataset

HLA-G*01:01	SN	SP	ACC	MCC	AUC
Model					
CNN	0.9467	0.9626	0.9548	0.9099	0.9872
Bi-LSTM	0.9482	0.9615	0.9550	0.9101	0.9863
CNN + Bi-LSTM (In series)	0.9533	0.9619	0.9577	0.9154	0.9873
CNN + CNN (In parallel)	0.9582	0.9593	0.9588	0.9176	0.9864
Bi-LSTM + Bi-LSTM (In parallel)	0.9529	0.9528	0.9529	0.9058	0.9861
DeepHLAPred	**0.9620**	**0.9685**	**0.9653**	**0.9305**	**0.9892**

Table 10 The performance of different modules on HLA-G*01:03 dataset

HLA-G*01:03	SN	SP	ACC	MCC	AUC
Model					
CNN	0.9372	0.9039	0.9208	0.8424	0.9726
Bi-LSTM	0.9257	0.9241	0.9248	0.8499	0.9716
CNN + Bi-LSTM (In series)	0.9281	0.9402	0.9341	0.8686	0.9786
CNN + CNN (In parallel)	0.9321	0.9333	0.9328	0.8656	0.9748
Bi-LSTM + Bi-LSTM (In parallel)	0.9309	0.9437	0.9374	0.8749	0.9703
DeepHLAPred	**0.9454**	**0.9626**	**0.9541**	**0.9083**	**0.9812**

Table 11 The performance of different modules on HLA-G*01:04 dataset

HLA-G*01:04	SN	SP	ACC	MCC	AUC
Model					
CNN	0.9264	0.9365	0.933	0.8627	0.9782
Bi-LSTM	0.9236	0.9234	0.9236	0.8475	0.9729
CNN + Bi-LSTM (In series)	0.9457	0.9394	0.9386	0.8858	0.9836
CNN + CNN (In parallel)	0.9310	0.9396	0.9354	0.8727	0.9802
Bi-LSTM + Bi-LSTM (In parallel)	0.9211	0.9434	0.9323	0.8657	0.9760
DeepHLAPred	**0.9545**	**0.9630**	**0.9587**	**0.9179**	**0.9855**

Table 12 The performance of different modules on HLA-E*01:01 dataset

HLA-E*01:01	SN	SP	ACC	MCC	AUC
Model					
CNN	0.8497	0.8512	0.8521	0.7050	0.9540
Bi-LSTM	0.8666	0.852	0.8588	0.7174	0.9209
CNN + Bi-LSTM (In series)	0.8475	0.8730	0.8629	0.7234	0.9356
CNN + CNN (In parallel)	0.9012	0.8588	0.8802	0.7605	0.9509
Bi-LSTM + Bi-LSTM (In parallel)	0.8721	0.8665	0.8690	0.7378	0.9391
DeepHLAPred	**0.9413**	**0.9013**	**0.9226**	**0.8455**	**0.9595**

Table 13 The performance of different modules on HLA-E*01:03 dataset

HLA-E*01:03	SN	SP	ACC	MCC	AUC
Model					
CNN	0.8507	0.8221	0.8377	0.6760	0.9130
Bi-LSTM	0.8448	0.8063	0.8259	0.6518	0.9053
CNN + Bi-LSTM (In series)	0.8905	0.8235	0.8575	0.7168	0.9287
CNN + CNN (In parallel)	0.8509	0.8464	0.8488	0.6974	0.9164
Bi-LSTM + Bi-LSTM (In parallel)	0.8511	0.8167	0.8346	0.6694	0.9031
DeepHLAPred	**0.8971**	**0.8656**	**0.8812**	**0.7631**	**0.9384**

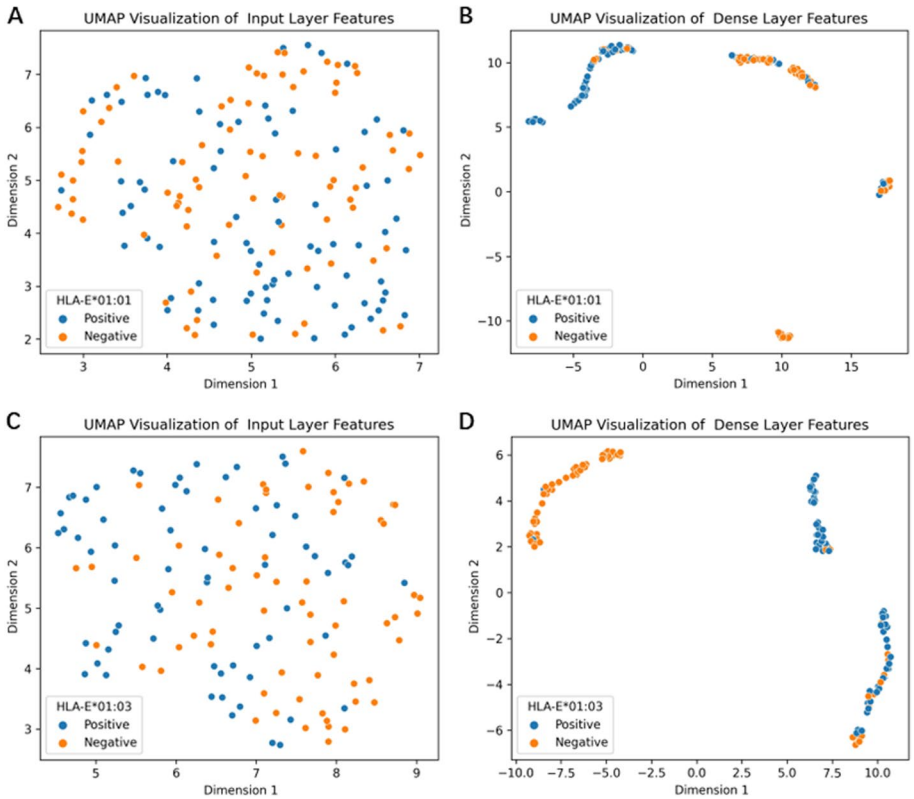

Fig. 6 The Umap visualization for (**A**) initial representations, (**B**) learned representation on the HLA-E*01:01 dataset, (**C**) initial representations, (**D**) learned representation on the HLA-E*01:03 dataset. The learned representations refer to output of the first fully-connected layer

DeepHLAPred remarkably improved the discriminating ability of representations.

Information entropy analysis

We explored further potential sequence patterns of non-classical class-I HLA binding peptides from two perspectives: amino acid information entropy and positional information entropy. The position specific amino acid matrix was defined by:

$$Z = \begin{pmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,n} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ z_{20,1} & z_{20,2} & \cdots & z_{20,n} \end{pmatrix} \tag{10}$$

where $z_{i,j}$ stood for the probability of the amino acid i at the position j and n represented the length of the sequence. The position specific amino acid matrix was estimated in practice by calculating all the samples in the balanced datasets. The amino acid information entropy and the position information entropy were calculated as:

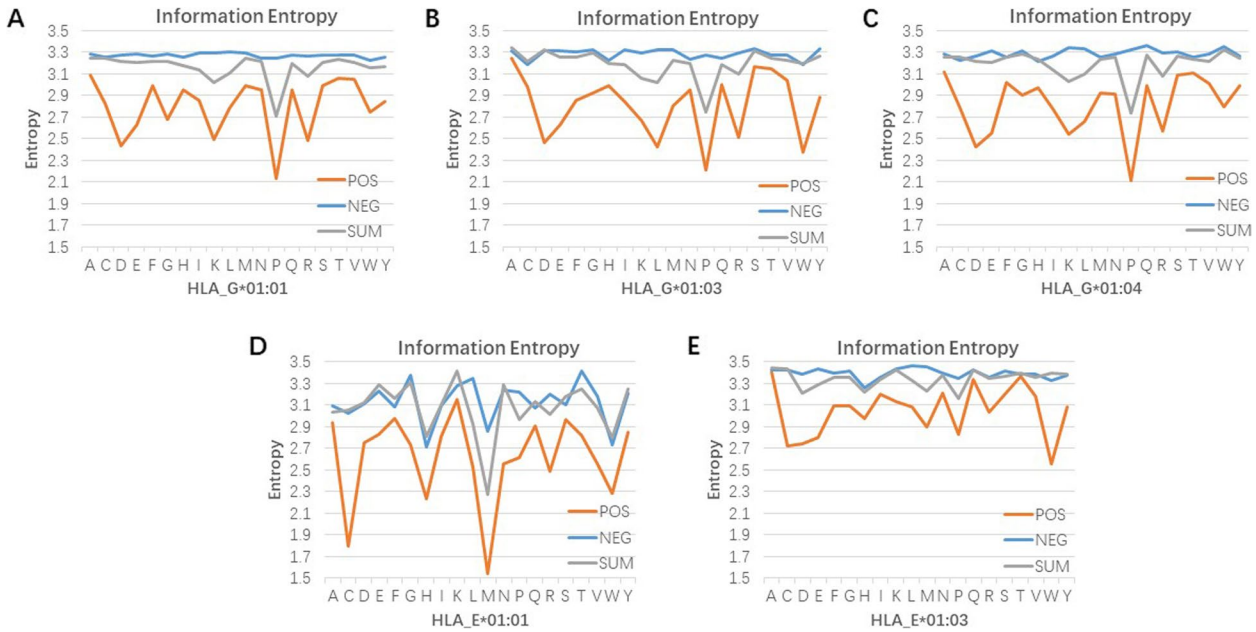

Fig. 7 Amino acids information entropy. "POS", "NEG", and "SUM" represent positive samples, negative samples, and the total samples, respectively

$$AP^i = \sum_{j=1}^{n} - Z_{i,j} log\left(Z_{i,j}\right) \tag{11}$$

and

$$PP^j = \sum_{i=1}^{20} - Z_{i,j} \log\left(Z_{i,j}\right) \tag{12}$$

The lower the information entropy was, the more certain the distribution of amino acid and position was. Figure 7 showed the amino acid information entropy on five balanced datasets. Evidently, HLA binding peptides generally have lower entropy values than the non-HLA binding peptides, indicating that the distribution of amino acid was not completely random. Amino acid information entropy exhibited specificity to the type of HLA binding peptides. The HLA-G binding peptides

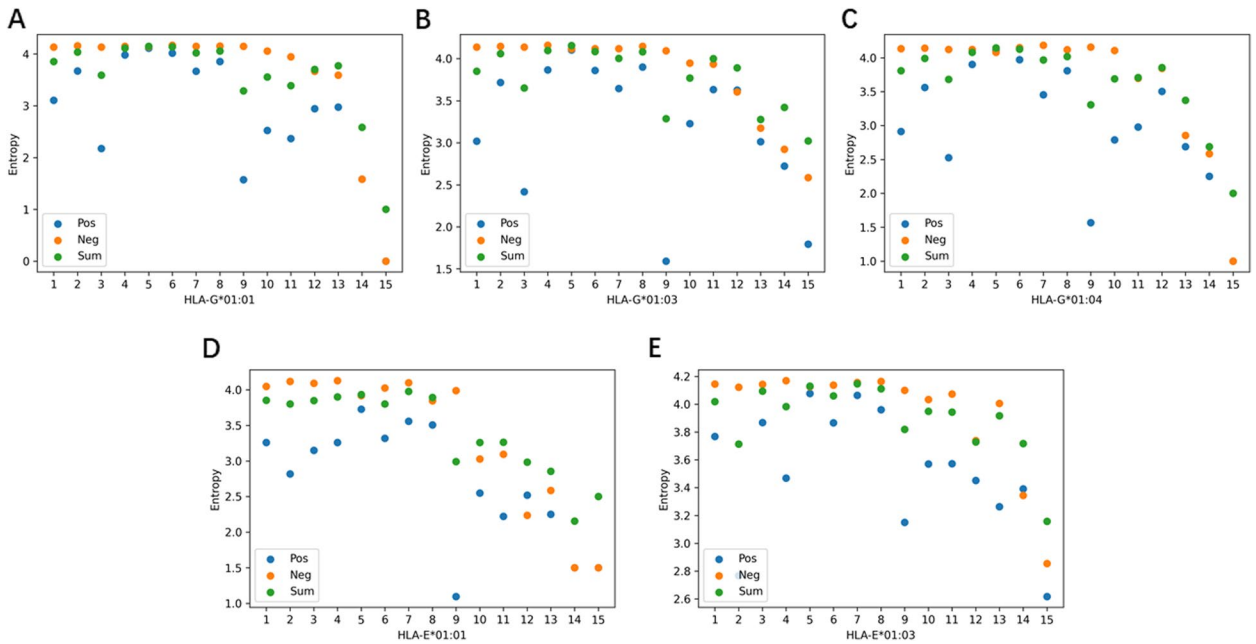

Fig. 8 The position information entropy of non-classical HLA peptide sequences

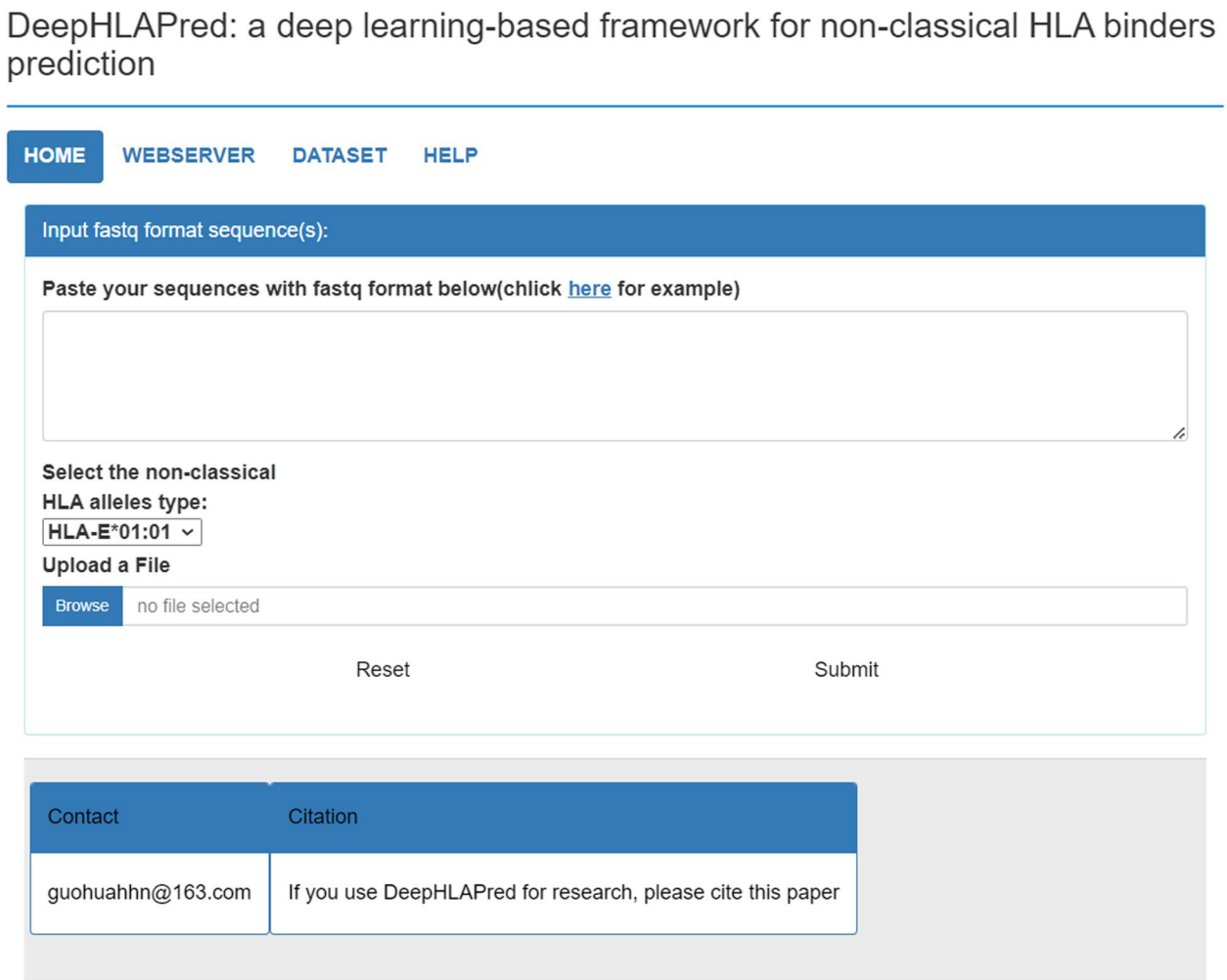

Fig. 9 The web server page of the DeepHLAPred

have lower value of amino acid information entropy at the Asparticacid (D) and Proline (P), while the HLA-E binding peptides have lower value at Cystine (C), Methionine (M), and Tryptophan (W). This implied that these amino acids were not distributed randomly. As shown in Fig. 8, we found that the positional information entropy of peptide sequences also was specific to type of HLA-binding peptides. Interestingly, positional information entropy at the 9-th position in the HLA-E*01:01, HLA-G*01:03, and HLA-G*01:04 were lower than others, indicating specificity of amino acid distribution at this position. These findings help us understand the sequence pattern of non-classical HLA I binding peptides [6, 21].

Webserver

To facilitate to predict non classical HLA class I binders, we developed a user-friendly webserver which is available at http:/www.biolscience.cn/DeepHLApred/. The webserver interface was shown in the Fig. 9. Users who utilize this webserver hardly require any prior knowledge about biology or deep learning. The only done is three steps. Firstly, users either input sequences in FASTA format into the inputting box or choose to upload a FASTA sequence file. Secondly, users select types of the non-classical HLA Class I allele which they want to predict. Finally, by clicking the submit button, users will get the prediction results on the webpage.

Conclusion

HLA is closely related to the human immune system. Precisely identifying the HLA binding peptides is still challenging. We used three feature extraction methods, EIIP, AAAF, and INM to encode peptide sequences, and proposed a CNN and Bi-LSTM-based deep learning model (DeepHLAPred) for non-classical HLA Class I binder prediction. The DeepHLAPred was extensively tested by

datasets of non-classical HLA I binder. It was well demonstrated that our method achieved state of the art performance on nearly all the datasets. The information entropy analysis implied the sequence pattern of non-classical binder to a certain extent. Though the DeepHLAPred demonstrated satisfactory performance in the prediction of non-classical HLA class I binding peptides. However, there still exists considerable room for improvement. In addition, the model interpretability need improving. In the future work, we shall focus on large language mode to improve prediction accuracy and interpretability.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12864-023-09796-2.

Additional file 1: Supplementary Table 1. The hyper-parameters of the DeepHLAPred. **Supplementary Table 2.** The performance on the HLA-G*01:01 dataset at different dropout rate. **Supplementary Table 3.** The performance on the HLA-G*01:03 dataset at different dropout rate. **Supplementary Table 4.** The performance on the HLA-G*01:04 dataset at different dropout rate. **Supplementary Table 5.** The performance on the HLA-E*01:01 dataset at different dropout rate. **Supplementary Table 6.** The performance on the HLA-E*01:03 dataset at different dropout rate. **Supplementary Table 7.** Comparison with state-of-the-art methods on five-fold cross-validation.

Acknowledgements
Not applicable.

Authors' contributions
GH conceived the experiments, analyzed the results, and reviewed the manuscript. XT collected the dataset, performed the experiments, analyzed results, and drafted the manuscript. PZ developed the software. All authors read and approved the final manuscript.

Funding
This work was supported by National Natural Science Foundation of China (62272310), by Hunan Province Natural Science Foundation of China (2022JJ50177), and by Shaoyang University Innovation Foundation for Postgraduate (CX2022SY041).

Availability of data and materials
The experimental data was available at https://github.com/tangxingyu0/DeepHLApred.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 20 June 2023 Accepted: 8 November 2023
Published online: 23 November 2023

References

1. Jia X, Han B, Onengut-Gumuscu S, et al. Imputing amino acid polymorphisms in human leukocyte antigens. PLoS ONE. 2013;8:e64683. https://doi.org/10.1371/journal.pone.0064683.
2. Moyer AM, Gandhi MJ. Human Leukocyte Antigen (HLA) Testing in Pharmacogenomics. Pharmacogenomics in Drug Discovery and Development. Volume 2547. Springer; 2022. pp. 21–45. https://doi.org/10.1007/978-1-0716-2573-6_2.
3. Mosaad Y. Clinical role of human leukocyte antigen in health and Disease. Scand J Immunol. 2015;82:283–306. https://doi.org/10.1111/sji.12329.
4. Choo SY. The HLA system: genetics, immunology, clinical testing, and clinical implications. Yonsei Med J. 2007;48:11–23. https://doi.org/10.3349/ymj.2007.48.1.11.
5. Medhasi S, Chantratita N. Human leukocyte antigen (HLA) system: genetics and association with bacterial and viral Infections. J Immunol Res. 2022;2022:9710376. https://doi.org/10.1155/2022/9710376.
6. Dhall A, Patiyal S, Raghava GP. HLAncPred: a method for predicting promiscuous non-classical HLA binding sites. Brief Bioinform. 2022;23:bbac192. https://doi.org/10.1093/bib/bbac192.
7. Robinson J, Barker DJ, Georgiou X, et al. Ipd-imgt/hla database. Nucleic Acids Res. 2020;48:D948–55. https://doi.org/10.1093/nar/gkz950.
8. Barker DJ, Maccari G, Georgiou X, et al. The IPD-IMGT/HLA database. Nucleic Acids Res. 2023;51:D1053–60. https://doi.org/10.1093/nar/gkac1011.
9. Paul P, Rouas-Freiss N, Moreau P, et al. HLA-G,-E,-F preworkshop: tools and protocols for analysis of non-classical class I genes transcription and protein expression. Hum Immunol. 2000;61:1177–95. https://doi.org/10.1016/S0198-8859(00)00154-3.
10. Wyatt RC, Lanzoni G, Russell MA, et al. What the HLA-I!—Classical and non-classical HLA class I and their potential roles in type 1 Diabetes. Curr Diab Rep. 2019;19:159. https://doi.org/10.1007/s11892-019-1245-z.
11. McCusker CT, Singal DP. The human leukocyte antigen (HLA) system: 1990. Transfus Med Rev. 1990;4:279–87. https://doi.org/10.1016/S0887-7963(90)70270-2.
12. Kochan G, Escors D, Breckpot K, et al. Role of non-classical MHC class I molecules in cancer immunosuppression. Oncoimmunology. 2013;2:e26491. https://doi.org/10.4161/onci.26491.
13. Moscoso J, Serrano-Vela J, Pacheco R, et al. HLA-G,-E and-F: allelism, function and evolution. Transpl Immunol. 2006;17:61–4. https://doi.org/10.1016/j.trim.2006.09.010.
14. Zhang L, Udaka K, Mamitsuka H, et al. Toward more accurate pan-specific MHC-peptide binding prediction: a review of current methods and tools. Brief Bioinform. 2012;13:350–64. https://doi.org/10.1093/bib/bbr060.
15. Singh H, Raghava G. ProPred: prediction of HLA-DR binding sites. Bioinformatics. 2001;17:1236–7. https://doi.org/10.1093/bioinformatics/17.12.1236.
16. Hannoun Z, Lin Z, Brackenridge S, et al. Identification of novel HIV-1-derived HLA-E-binding peptides. Immunol Lett. 2018;202:65–72. https://doi.org/10.1016/j.imlet.2018.08.005.
17. Finton KA, Brusniak M-Y, Jones LA, et al. ARTEMIS: a novel mass-spec platform for HLA-restricted self and disease-associated peptide discovery. Front Immunol. 2021;12:658372. https://doi.org/10.3389/fimmu.2021.658372.
18. Bisset LR, Fierz W. Using a neural network to identify potential HLA-DR1 binding sites within proteins. J Mol Recognit. 1993;6:41–8. https://doi.org/10.1002/jmr.300060105.
19. Singh H, Raghava G. ProPred1: prediction of promiscuous MHC Class-I binding sites. Bioinformatics. 2003;19:1009–14. https://doi.org/10.1093/bioinformatics/btg108.
20. Jurtz V, Paul S, Andreatta M, et al. NetMHCpan-4.0: improved peptide–MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. J Immunol. 2017;199:3360–8. https://doi.org/10.4049/jimmunol.1700893.
21. O'Donnell TJ, Rubinsteyn A, Laserson U. MHCflurry 2.0: improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. Cell Syst. 2020;11:42–8. https://doi.org/10.1016/j.cels.2020.06.010.
22. Ye Y, Wang J, Xu Y, et al. MATHLA: a robust framework for HLA-peptide binding prediction integrating bidirectional LSTM and multiple head

attention mechanism. BMC Bioinformatics. 2021;22:7. https://doi.org/10.1186/s12859-020-03946-z.
23. Zhang Y, Zhu G, Li K, et al. HLAB: learning the BiLSTM features from the ProtBert-encoded proteins for the class I HLA-peptide binding prediction. Brief Bioinform. 2022;23:bbac173. https://doi.org/10.1093/bib/bbac173.
24. Chu Y, Zhang Y, Wang Q, et al. A transformer-based model to predict peptide–HLA class I binding and optimize mutated peptides for vaccine design. Nat Mach Intell. 2022;4:300–11. https://doi.org/10.1038/s42256-022-00459-7.
25. Mei S, Li F, Leier A, et al. A comprehensive review and performance evaluation of bioinformatics tools for HLA class I peptide-binding prediction. Brief Bioinform. 2020;21:1119–35. https://doi.org/10.1093/bib/bbz051.
26. Nielsen M, Lundegaard C, Lund O. Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. BMC Bioinformatics. 2007;8:238. https://doi.org/10.1186/1471-2105-8-238.
27. Lata S, Bhasin M, Raghava GP. Application of machine learning techniques in predicting MHC binders. Methods Mol Biol. 2007;409:201–15. https://doi.org/10.1007/978-1-60327-118-9_14.
28. Wang P, Sidney J, Kim Y, et al. Peptide binding predictions for HLA DR, DP and DQ molecules. BMC Bioinformatics. 2010;11:568. https://doi.org/10.1186/1471-2105-11-568.
29. Peters B, Bui H-H, Frankild S, et al. A community resource benchmarking predictions of peptide binding to MHC-I molecules. PLoS Comput Biol. 2006;2:e65. https://doi.org/10.1371/journal.pcbi.0020065.
30. Lin HH, Ray S, Tongchusak S, et al. Evaluation of MHC class I peptide binding prediction servers: applications for vaccine research. BMC Immunol. 2008;9:8. https://doi.org/10.1186/1471-2172-9-8.
31. Nielsen M, Lundegaard C, Blicher T, et al. Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan. PLoS Comput Biol. 2008;4:e1000107. https://doi.org/10.1371/journal.pcbi.1000107.
32. Elnaggar A, Heinzinger M, Dallago C, et al. Prottrans: toward understanding the language of life through self-supervised learning. IEEE Trans Pattern Anal Mach Intell. 2021;44:7112–27. https://doi.org/10.1109/tpami.2021.3095381.
33. Devlin J, Chang M-W, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv Preprint. 2018. https://doi.org/10.48550/arXiv.1810.04805. arXiv:1810.04805.
34. Le NQK, Ho Q-T, Nguyen T-T-D, et al. A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. Brief Bioinform. 2021;22:bbab005. https://doi.org/10.1093/bib/bbab005.
35. Le NQK, Ho Q-T, Nguyen V-N, et al. BERT-Promoter: an improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection. Comput Biol Chem. 2022;99:107732. https://doi.org/10.1016/j.compbiolchem.2022.107732.
36. Suzek BE, Wang Y, Huang H, et al. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics. 2015;31:926–32. https://doi.org/10.1093/bioinformatics/btu739.
37. Steinegger M, Söding J. Clustering huge protein sequence sets in linear time. Nat Commun. 2018;9:2542. https://doi.org/10.1038/s41467-018-04964-5.
38. McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv Preprint. 2018. https://doi.org/10.48550/arXiv.1802.03426. arXiv:1802.03426.
39. Alvaro-Benito M, Morrison E, Wieczorek M, et al. Human leukocyte Antigen-DM polymorphisms in autoimmune Diseases. Open Biology. 2016;6:160165. https://doi.org/10.1098/rsob.160165.
40. Foroni I, Couto AR, Bettencourt BF, et al. HLA-E, HLA-F and HLA-G—the non-classical side of the MHC cluster. HLA and Associated Important Diseases. 2014;3:61–109. https://doi.org/10.5772/57507.
41. Crux NB, Elahi S. Human leukocyte antigen (HLA) and immune regulation: how do classical and non-classical HLA alleles modulate immune response to human immunodeficiency virus and Hepatitis C virus Infections? Front Immunol. 2017;8:832. https://doi.org/10.3389/fimmu.2017.00832.
42. Carlini F, Ferreira V, Buhler S, et al. Association of HLA-A and non-classical HLA class I alleles. PLoS ONE. 2016;11:e0163570. https://doi.org/10.1371/journal.pone.0163570.
43. Bukur J, Jasinski S, Seliger B. The role of classical and non-classical HLA class I antigens in human tumors. Sem Cancer Biol. 2012;22:350–8. https://doi.org/10.1016/j.semcancer.2012.03.003.
44. Ferns DM, Heeren AM, Samuels S, et al. Classical and non-classical HLA class I aberrations in primary cervical squamous-and adenocarcinomas and paired lymph node metastases. J Immunother Cancer. 2016;4:78. https://doi.org/10.1186/s40425-016-0184-3.
45. Murdaca G, Contini P, Negrini S et al. Immunoregulatory role of HLA-G in allergic Diseases. J Immunol Res. 2016;2016:6865758. https://doi.org/10.1155/2016/6865758.
46. Bloch KM, Arce GR. Analyzing protein sequences using signal analysis techniques, in Computational and Statistical Approaches to Genomics. 2006, 137–161. https://doi.org/10.1007/0-387-26288-1_9.
47. Bonidia RP, Domingues DS, Sanches DS, et al. MathFeature: feature extraction package for DNA, RNA and protein sequences based on mathematical descriptors. Brief Bioinform. 2022;23:bbab434. https://doi.org/10.1093/bib/bbab434.
48. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network, in 2017 international conference on engineering and technology (ICET), Ieee, (2017), 1–6. https://doi.org/10.1109/icengtechnol.2017.8308186.
49. Sazli MH. A brief review of feed-forward neural networks. Commun Fac Sci Univ Ankara Ser A2-A3 Phys Sci Eng. 2006;50. https://doi.org/10.1501/commua1-2_0000000026.
50. Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks. Pattern Recogn. 2018;77:354–77. https://doi.org/10.1016/j.patcog.2017.10.013.
51. Tajbakhsh N, Shin JY, Gurudu SR, et al. Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE Trans Med Imaging. 2016;35:1299–312. https://doi.org/10.1109/tmi.2016.2535302.
52. Li Q, Cai W, Wang X et al. Medical image classification with convolutional neural network, in 2014 13th international conference on control automation robotics & vision (ICARCV). 2014 IEEE, 844–848. https://doi.org/10.1109/icarcv.2014.7064414.
53. Passricha V, Aggarwal RK. A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition. J Intell Syst. 2020;29:1261–74. https://doi.org/10.1515/jisys-2018-0372.
54. Khan MJ, Yousaf A, Javed N, et al. Automatic target detection in satellite images using deep learning. J Space Technol. 2017;7:44–9. https://doi.org/10.3390/s22031147.
55. Britz D. 2015. Understanding convolutional neural networks for NLP. Available from: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp.
56. Rehman AU, Malik AK, Raza B, et al. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools and Applications. 2019;78:26597–613. https://doi.org/10.1007/s11042-019-07788-7.
57. Nguyen QH, Nguyen-Vo T-H, Le NQK, et al. iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks. BMC Genomics. 2019;20:1–10. https://doi.org/10.1186/s12864-019-6336-3.
58. Le NQK, Ho QT, Ou YY. Incorporating deep learning with convolutional neural networks and position specific scoring matrices for identifying electron transport proteins. J Comput Chem. 2017;38:2000–6. https://doi.org/10.1002/jcc.24842.
59. Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D. 2020;404:132306. https://doi.org/10.1016/j.physd.2019.132306.
60. Yu Y, Si X, Hu C et al. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 2019;31:1235–1270. https://doi.org/10.1162/neco_a_01199.
61. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
62. Rashid TA, Fattah P, Awla DK. Using accuracy measure for improving the training of LSTM with metaheuristic algorithms. Procedia Comput Sci. 2018;140:324–33. https://doi.org/10.1016/j.procs.2018.10.307.
63. Jin N, Wu J, Ma X, et al. Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification. IEEE Access. 2020;8:77060–72. https://doi.org/10.1109/access.2020.2989428.
64. Jing R. A self-attention based LSTM network for text classification. J Physics Conference Series. 2019;1207:012008. https://doi.org/10.1088/1742-6596/1207/1/012008.
65. Le N-Q-K, Ou Y-Y. Incorporating efficient radial basis function networks and significant amino acid pairs for predicting GTP binding sites in

transport proteins. BMC Bioinformatics. 2016;17:183–92. https://doi.org/10.1186/s12859-016-1369-y.
66. Le NQK, Yapp EKY, Ho Q-T, et al. iEnhancer-5Step: identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding. Anal Biochem. 2019;571:53–61. https://doi.org/10.1016/j.ab.2019.02.017.
67. Vita R, Mahajan S, Overton JA, et al. The immune epitope database (IEDB): 2018 update. Nucleic Acids Res. 2019;47:D339–43. https://doi.org/10.1093/nar/gky1006.
68. Reynisson B, Alvarez B, Paul S, et al. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. Nucleic Acids Res. 2020;48:W449–54. https://doi.org/10.1093/nar/gkaa379.

Publisher's Note