Bioinformatics, 35(23), 2019, 4946–4954
doi: 10.1093/bioinformatics/btz427
Advance Access Publication Date: 23 May 2019
Original Paper

Sequence analysis

ACME: pan-specific peptide–MHC class I binding prediction through attention-based deep neural networks

Yan Hu[1], Ziqiang Wang[2], Hailin Hu [3], Fangping Wan[4], Lin Chen[5], Yuanpeng Xiong[6,7], Xiaoxia Wang[2], Dan Zhao[4], Weiren Huang[2,*] and Jianyang Zeng[4,8,*]

[1]School of Life Sciences, Tsinghua University, Beijing 100084, China, [2]Department of Urology, Shenzhen Second People's Hospital, The First Affiliated Hospital of Shenzhen University, International Cancer Center, Shenzhen University School of Medicine, Shenzhen 518039, China, [3]School of Medicine, Tsinghua University, Beijing 100084, China, [4]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China, [5]Turing AI Institute of Nanjing, Nanjing 210000, China, [6]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, [7]Bioinformatics Division, BNRIST/Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China and [8]MOE Key Laboratory of Bioinformatics, Tsinghua University, Beijing 100084, China

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on February 7, 2019; revised on April 12, 2019; editorial decision on May 11, 2019; accepted on May 19, 2019

Abstract

Motivation: Prediction of peptide binding to the major histocompatibility complex (MHC) plays a vital role in the development of therapeutic vaccines for the treatment of cancer. Algorithms with improved correlations between predicted and actual binding affinities are needed to increase precision and reduce the number of false positive predictions.

Results: We present ACME (Attention-based Convolutional neural networks for MHC Epitope binding prediction), a new pan-specific algorithm to accurately predict the binding affinities between peptides and MHC class I molecules, even for those new alleles that are not seen in the training data. Extensive tests have demonstrated that ACME can significantly outperform other state-of-the-art prediction methods with an increase of the Pearson correlation coefficient between predicted and measured binding affinities by up to 23 percentage points. In addition, its ability to identify strong-binding peptides has been experimentally validated. Moreover, by integrating the convolutional neural network with attention mechanism, ACME is able to extract interpretable patterns that can provide useful and detailed insights into the binding preferences between peptides and their MHC partners. All these results have demonstrated that ACME can provide a powerful and practically useful tool for the studies of peptide–MHC class I interactions.

Availability and implementation: ACME is available as an open source software at `https://github.com/HYsxe/ACME`.

Contact: pony8980@163.com or zengjy321@tsinghua.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Presentation of antigen peptides by major histocompatibility complex (MHC) class I molecules plays a vital role in initiating an immune response to identify and kill cancer cells. Peptides derived from mutant proteins in cancer cells can be engineered into cancer vaccines to stimulate a specific immune response against the cancer cells presenting these antigen peptides (Carreno *et al.*, 2015; Ott *et al.*, 2017; Walter *et al.*, 2012). However, in most cases, for a peptide to effectively elicit an immune response, it needs to bind to an MHC molecule with a sufficiently high affinity ($IC_{50} < 500$ nM), making peptide–MHC binding the most crucial and selective step in antigen presentation (Yewdell and Bennink, 1999). Hence, identifying the peptides that can be presented by MHC molecules is a vital step in developing effective therapeutic cancer vaccines (Hu *et al.*, 2018). This underscores the importance of developing *in silico* algorithms that can accurately predict the binding affinities of peptides to MHC molecules.

In the past two decades, two types of computational methods, i.e. allele-specific and pan-specific predictors, have been proposed for this purpose. The allele-specific predictors generally train one model for each MHC allele (Andreatta and Nielsen, 2016; Lundegaard *et al.*, 2008; Peters and Sette, 2005; Vang and Xie, 2017). For example, NetMHC, one of the most widely used allele-specific models, employs a conventional feed-forward neural network with a single hidden layer to predict peptide–MHC interactions for individual alleles (Andreatta and Nielsen, 2016; Lundegaard *et al.*, 2008). When training these allele-specific methods, we have to split the binding data into individual smaller datasets, each specific for a single allele, which makes it difficult to derive accurate predictions for those data-deficient alleles. On the other hand, the pan-specific methods take both peptide and MHC sequences as input features, thus allowing to pool all the data of different alleles together and train one common model for all alleles (Han and Kim, 2017; Hoof *et al.*, 2009; Jurtz *et al.*, 2017; Karosiene *et al.*, 2012; Nielsen *et al.*, 2007; Nielsen and Andreatta, 2016). For instance, the widely used method, NetMHCpan, uses a simple neural network model with one hidden layer and takes information extracted from both peptide and MHC sequences as its input to perform the pan-specific prediction task (Hoof *et al.*, 2009; Jurtz *et al.*, 2017; Nielsen *et al.*, 2007; Nielsen and Andreatta, 2016). However, such neural networks with a single hidden layer are generally not sophisticated enough to capture all the intrinsically complex patterns in the data. The recently proposed method, ConvMHC, uses deep convolutional networks (Han and Kim, 2017) to address this problem. Unfortunately, it cannot integrate the binding data of peptides with different lengths, and is only applicable to 9-mer peptides.

Despite the progress made so far, there is still a need for developing a more precise method for peptide–MHC binding prediction to reduce the large number of false positives and thus improve the confidence of the predicted peptide–MHC interactions. In addition, improving the correlations between predicted and measured binding affinities may help quantify the binding advantage of neoantigens compared to the wild-type version, which can further facilitate vaccine development. Moreover, the prediction results from most of the previous methods lack interpretability. Although there existed several studies on the interpretable binding patterns between peptides and MHC modules (Bade-Doeding *et al.*, 2005; Huyton *et al.*, 2012; Yusim *et al.*, 2016), they only focused on a limited number of alleles. Thus, it becomes increasingly necessary to develop a model that can not only make accurate binding prediction but also reveal interpretable patterns to characterize peptide–MHC interactions. Recently, the attention mechanism has been used in natural language processing (Bahdanau *et al.*, 2014; Li *et al.*, 2015), computer vision (Mnih *et al.*, 2014) and bioinformatics (Hu *et al.*, 2019) to produce interpretable results for the proposed deep learning models. This strategy assigns different weights to individual positions of the input, so that the model can focus on the most crucial features to perform better prediction. In sequence analysis, the attention values (which can also be considered as weights) for individual sites (e.g. base pairs or amino acid residues) derived by the embedded attention mechanism allow one to spot those important sites that significantly contribute to the final predictions. Hence, here the attention mechanism is a suitable technique to aid the interpretation of the binding patterns of MHC molecules.

In this article, we develop a new deep learning-based method, called ACME (Attention-based Convolutional neural networks for MHC Epitope binding prediction), for the pan-specific prediction of the binding affinities between peptides and MHC class I molecules (Fig. 1a). We combine a deep convolutional neural network with an attention module to build an accurate and interpretable prediction model, in which features of different levels are extracted from multiple layers of the neural network and then integrated to effectively capture the intrinsic characteristics of peptide–MHC binding. Extensive tests on existing benchmark and external datasets have demonstrated that ACME can significantly outperform other state-of-the-art prediction methods with an increase of the Pearson correlation coefficient between predicted and measured binding affinities by up to 23 percentage points. More importantly, we have also shown that ACME can be effectively used to make accurate predictions for those new MHC alleles that are not seen in training data. In addition, the ability of ACME to identify strong binding peptides has been well validated experimentally. Moreover, comprehensive analyses have indicated that ACME can extract interpretable molecular interaction patterns and generate accurate peptide binding motifs for different MHC alleles, including those with scarce or no available experimental data. Overall, our model can provide a powerful and useful tool for predicting peptide–MHC binding as well as investigating the underlying molecular patterns of this biological process. ACME is available as an open source software at `https://github.com/HYsxe/ACME`.

2 Materials and methods

2.1 Datasets

The widely used IEDB MHC class I binding affinity dataset was downloaded from the IEDB website (`http://tools.immuneepitope.org/mhci/download/`) (Kim *et al.*, 2014; Vita *et al.*, 2015). We first carried out a five-fold cross-validation procedure on this dataset to evaluate the prediction performance of our model and other state-of-the-art prediction methods. Afterwards, we combined this IEDB dataset with another dataset obtained from (Pearson *et al.*, 2016), to train a final version of our model (redundant data were removed). To check if our final model can be generalized to other datasets, we further tested our model on the data obtained from the IEDB weekly benchmarking website (`http://tools.iedb.org/auto_bench/mhci/weekly/`) (Trolle *et al.*, 2015), which had also been previously used to evaluate other methods as well (Han and Kim, 2017; Vang and Xie, 2017). In particular, the datasets published within the last three years were chosen. These datasets were not used in parameter optimization and training of our final model. The sequences of different MHC molecules were obtained from the IPD-IMGT/HLA Database (`ftp://ftp.ebi.ac.uk/pub/databases/ipd/imgt/hla/hla_prot.fasta`) (Robinson *et al.*, 2014). More details about dataset preprocessing and filtering can be found in Supplementary Notes Section S1.1.

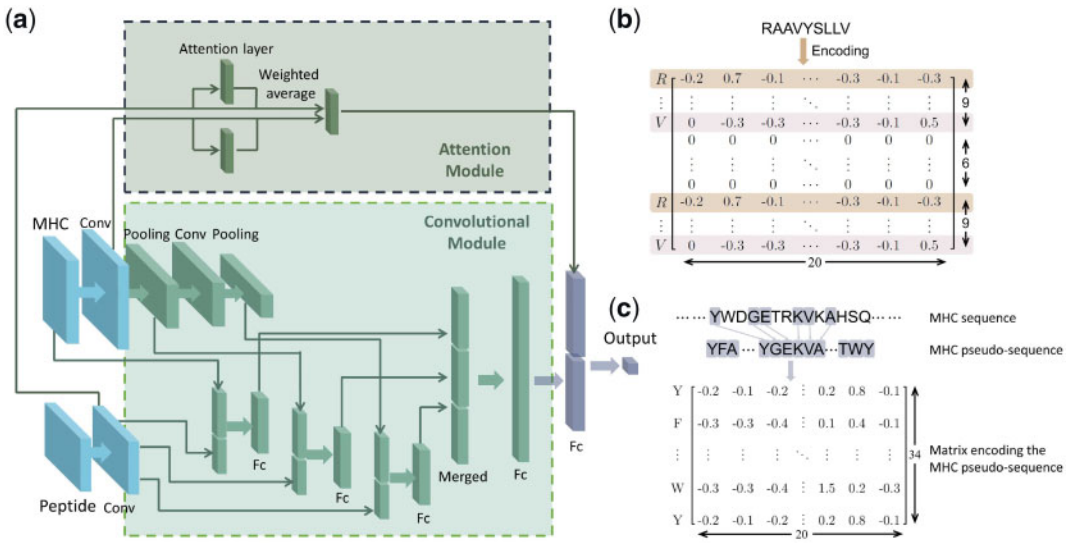

Fig. 1. Schematic illustration of the ACME framework. (**a**) The architecture of the deep neural network employed in ACME. MHC and peptide sequences are encoded and used as inputs. After initial feature extraction, the encoded feature representations are channeled into both attention and convolutional modules. The outputs of these two modules are then combined to make the final prediction. (**b**) Our scheme for encoding peptide sequences. Each peptide sequence is encoded into a 24×20 matrix in a paired-end manner. (**c**) Our scheme for encoding MHC sequences. For each MHC sequence, the 34 residues that are in close spatial proximity with the peptide partner (< 4.0 angstroms) are selected as a pseudo-sequence and then encoded into a 34×20 matrix. More details can be found in the text

2.2 Data encoding

The input data to our prediction framework ACME include peptide and MHC sequences, which are both encoded based on the standard BLOSUM50 scoring matrix (Henikoff and Henikoff, 1992), with each residue encoded by the corresponding row of this matrix. Here, BLOSUM50 was adopted following the same protocols as in previous studies (Jurtz *et al.*, 2017; Nielsen *et al.*, 2007; Nielsen and Andreatta, 2016). In our framework, the elements of the BLOSUM matrix are also divided by a scaling factor to facilitate model training, which was set to 10 after hyperparameter tuning using a grid search scheme. Each peptide is encoded into a 24×20 matrix in a pair-end manner (Fig. 1b). In particular, the first 12 rows of the input matrix encode the input peptide sequence with the N terminus aligned to the first row, while the last 12 rows encode the peptide sequence with the C terminus aligned to the last row. For those peptides shorter than 12 residues, the vacancies in the middle are filled with zero padding. For instance, a peptide sequence of length 9 is encoded into the 1st–9th and 16th–24th rows, in which the 1st and 16th rows encode the N terminus while the 9th and 24th rows encode the C terminus. In this case, the 10th to 15th rows are all padded with zeros. Such a design was mainly inspired by the structural studies of peptide–MHC complexes (Engelhard, 1994; Liu and Gao, 2001; Madden, 1995). These studies pointed out that the two ends of the peptides are usually inserted into two specific binding pockets in the peptide binding grooves and hence peptide termini often play a more important role in binding compared to the middle parts of the peptides. As a result, when peptides of different lengths bind to the same MHC molecule, the two ends of the peptides usually interact with the same pockets in similar ways, while the interactions between the middle parts of the peptides and the MHC molecules might vary to a greater extent depending on the lengths, sequences and geometries of the peptides. Therefore, we decided to fix the two ends of the peptides during feature encoding, and pad zeros to the middle, so that for peptides of different lengths, the most important features at the ends of the peptides are always well aligned.

When encoding the input MHC sequences, we only consider those residues that are in close contact with the peptide residues (i.e. within 4.0 angstroms). These residues form a short sequence, which is also called the MHC pseudo-sequence, as defined in (Nielsen *et al.*, 2007). In particular, 34 such residues are chosen and encoded into a 34×20 matrix, where each residue is represented by the corresponding row in the BLOSUM50 matrix (Fig. 1c).

The peptide–MHC binding affinities, which are represented by the IC_{50} values in nM units, are first transformed using the equation $score_{transformed} = 1 - \frac{\log(Affinity)}{\log(50000)}$, as previously described in (Nielsen *et al.*, 2003), and then used as the label values to train and test our model.

2.3 Network architecture

Our model ACME consists of four main components, including initial feature extraction, a convolutional module, an attention module and the final output (Fig. 1a). ACME first passes the encoded feature of peptide and MHC sequences through a convolutional layer for initial feature extraction, and then sends the extracted feature maps into both convolutional and attention modules (Supplementary Fig. S1). After that, the outputs of these two modules are combined to make the final prediction. The model is implemented using Keras (https://keras.io). Below we will describe each part of the model in detail.

2.3.1 Initial feature extraction

The encoded peptide and MHC feature representations as input to our model are first processed separately by a one-dimensional convolutional layer with rectified linear unit (ReLU) activation (Nair and Hinton, 2010) for initial feature extraction (Supplementary Fig. S2). The convolutional layers mentioned hereafter all use ReLU activation, unless otherwise noted. The convolutional layer outputs a feature map matrix F, in which each column corresponds to one convolutional filter and each row represents a position of the input, that is,

$$F_{jk} = f\left(b_k + \sum_i \sum_d M_{j+d,i} W_{d,i,k}\right), \quad (1)$$

where M stands for the input matrix (24×20 for the peptide sequence and 34×20 for the MHC pseudo-sequence), f stands for the ReLU activation function, i.e. $f(X_{jk}) = max\{X_{jk}, 0\}$, $W_{d,i,k}$ stands

for the element in the dth row and the ith column of the kth filter, and b_k stands for the corresponding bias. Note that the ranges of d and j may vary due to different padding results. The resulting feature maps for the peptide and MHC sequences, extracted by the first convolutional layers, are then separately forwarded into both convolutional and attention modules (Fig. 1a).

2.3.2 Convolutional module

The convolutional module is mainly designed to extract the binding features of different levels and then integrate them to make prediction (Supplementary Fig. S3). In our convolutional module (Fig. 1a), the MHC feature map $\boldsymbol{F}$ first goes through a max-pooling layer, followed by another round of one-dimensional convolution and max-pooling operations, while the peptide feature map does not go through further convolution and pooling operations mainly due to its short sequence length (i.e. 9–12 residues). The MHC features extracted after 0, 1 and 2 rounds of convolution and max-pooling operations are then concatenated separately with the peptide feature map after one convolutional layer, so that the model can learn how MHC features of different abstraction levels combine with peptide features to determine peptide–MHC binding. Here, each concatenated layer is also connected to a separate fully connected layer. The outputs of these three fully connected layers are then concatenated together and forwarded into two consecutive fully connected layers. The output of the latter is the output of the convolutional module. Note that similar designs have been adopted previously (Kim, 2014; Zhang *et al.*, 2017), in which convolutional filters of different sizes were used in parallel to extract features of different ranges, and the features extracted by different filters were first separately processed and then concatenated to generate the final output. More details on our network architecture can be found in Supplementary Notes Section S1.2.

2.3.3 Attention module

The attention module in our framework cooperates with the convolutional module to extract interpretable binding patterns (i.e. motifs). In particular, it assigns different weights to the feature vectors corresponding to individual residue positions and then computes their weighted average to facilitate prediction (Supplementary Fig. S4). As a result, it learns to assign higher weights to those residues that are more important in predicting peptide–MHC interactions. The structure of the attention module is shown in Supplementary Figure S5. Each row vector $\boldsymbol{F}_j$ in the feature map matrix $\boldsymbol{F}$ is taken as the input to the attention layer, and is then converted into a real number w_j. In other words, the attention layer maps the feature map matrix $\boldsymbol{F}$ into an attention weight vector $\boldsymbol{w}$, that is,

$$\boldsymbol{w} = \boldsymbol{b} + \boldsymbol{F}\boldsymbol{a}, \tag{2}$$

where $\boldsymbol{a}$ stands for the weights of the attention layer and $\boldsymbol{b}$ stands the corresponding bias. Then each element w_j of $\boldsymbol{w}$ is converted into w'_j through softmax normalization, that is,

$$w'_j = \frac{e^{w_j}}{\sum_i e^{w_i}}. \tag{3}$$

Next, we compute the weighted average $\boldsymbol{F}_{avg}$ over all row vectors, with each w'_j being the weight of the corresponding $\boldsymbol{F}_j$, that is,

$$\boldsymbol{F}_{avg} = \sum_j w_j' \boldsymbol{F}_j. \tag{4}$$

Here, $\boldsymbol{F}_{avg}$ stands for the output of the attention module, which is further combined with the output of the convolutional module to produce the final prediction.

2.3.4 Performing the final prediction

The outputs of both convolutional and attention modules are concatenated together and then forwarded into a fully connected layer with a sigmoid activation function to perform the final prediction (Fig. 1a), that is,

$$Output_{final} = \frac{1}{1 + \exp(Fc(Cat(Output_{Conv}, Output_{Att})))}, \tag{5}$$

where $Output_{Conv}$, $Output_{Att}$ and $Output_{final}$ stand for the outputs of convolutional and attention modules, and the final output of ACME, respectively, $Fc(\cdot)$ represents the linear transformation in the last fully connected layer, and $Cat(\boldsymbol{x}, \boldsymbol{y})$ stands for the concatenation of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

2.4 Network training

To determine the hyperparameters of our model, we split the IEDB dataset into a training set (95%) and a validation set (5%). The model was optimized using the training set, and its optimal hyperparameters were determined through a grid search approach on the validation set (Supplementary Notes Section S1.3). Such an experimental setup was adopted mainly because it is less time-consuming for our deep learning model (compared to the nested cross-validation based scheme). Redundant data were removed using the same approach as described previously when generating training and validation datasets (Andreatta and Nielsen, 2016). The performance was evaluated mainly using the Pearson correlation coefficient (PCC) between predicted and measured binding affinities, while the area under the receiver operating characteristics curve (AUROC) was also used as an additional metric to evaluate the prediction results. For the calculation of AUROC, the experimentally determined IC_{50} values were converted into one or zero labels using a threshold of 500 nM. Each model consists of an ensemble of n deep neural networks, where $n = 5$ in five-fold cross-validation and $n = 25$ in the final version of our model. The average prediction score over all the networks in the ensemble is used as the final output. We use the mean squared error as the loss function. In addition, the Adam optimizer (Kingma and Ba, 2014) is used to compute stochastic gradient descent in the training procedure. After hyperparameter calibration, we ran a five-fold cross-validation procedure on the training set mentioned above to evaluate the prediction performance of our model. Five trials of five-fold cross-validation were conducted, each with a different random initialization and their average result was reported. More details about the five-fold cross-validation procedure can be found in Supplementary Notes Section S1.4.

3 Results

3.1 Five-fold cross-validation on the IEDB MHC class I binding affinity dataset

To evaluate the prediction performance of ACME, we first carried out five-fold cross-validation on the IEDB dataset (Kim *et al.*, 2014; Vita *et al.*, 2015) for individual alleles and different peptide lengths, including 11-mers, 10-mers and 9-mers (Section 2.4). We also compared the performance of our model to that of the current state-of-the-art method NetMHCpan (Nielsen and Andreatta, 2016), which had been widely used for peptide–MHC binding prediction. Note that NetMHCpan 3.0 was used here because it was the latest version of NetMHCpan with detailed performance results published for individual alleles and peptide lengths (Nielsen and Andreatta, 2016).

Here we mainly used the Pearson correlation coefficient (PCC) between predicted and measured binding affinities to assess the

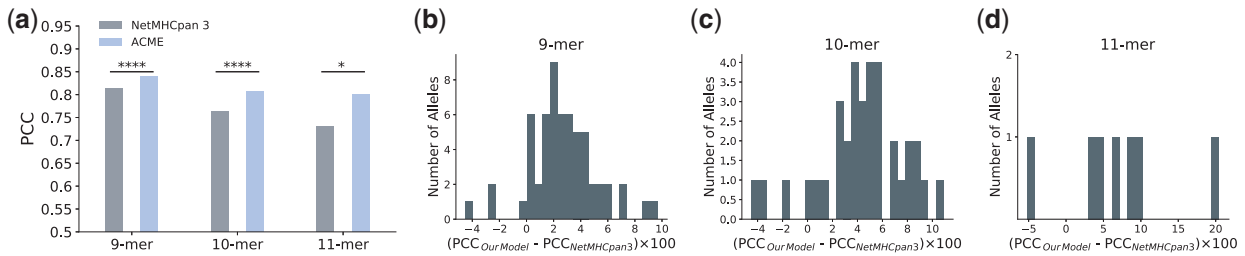

Fig. 2. Performance comparison between ACME and the state-of-the-art baseline model, NetMHCpan 3.0. (**a**) Performances of the two models on peptides of different lengths. ACME significantly outperformed NetMHCpan 3.0 on 9-mer, 10-mer and 11-mer peptides. *: $P < 0.05$, ****: $P < 0.0001$ (one-sample t-test based on the performance differences on individual alleles). The prediction results of NetMHCpan 3.0 were obtained from (Nielsen and Andreatta, 2016). (**b–d**) Performance improvement of ACME over NetMHCpan 3.0 for 11-mer, 10-mer and 9-mer peptides, respectively. The horizontal axis stands for the improvement (in terms of percentage points), while the vertical axis represents the number of alleles with such improvements

predictive abilities of different models. Compared to the previously reported performance of NetMHCpan 3.0, ACME achieved significantly better prediction results, with an average increase of PCC by 7.0, 4.4 and 2.8 percentage points for 11-mers, 10-mers and 9-mers, respectively (Fig. 2). More specifically, for 11-mers, ACME outperformed the baseline method on 6 out of the 7 alleles tested. On 4 alleles, ACME achieved improvement of over 5 percentage points in PCC, with the most significant case exceeding 20 percentage points. For 10-mers, ACME outperformed the baseline method on 31 out of the 35 alleles tested, and achieved improvement of over 5 percentage points on 15 alleles, with the highest one exceeding 10 percentage points. For 9-mers, ACME outperformed the baseline on 55 out of the 59 alleles tested, with improvement of over 5 percentage points on 9 alleles. More details can be found in Supplementary Data S1 and Supplementary Notes Section S1.4. In addition, we also formulated the peptide–MHC binding problem as a binary classification task and used the area under the receiver operating characteristic curve (AUROC) to evaluate the prediction performance of ACME. The comparison results based on this metric also demonstrated that ACME significantly outperformed NetMHCpan 3.0 (Supplementary Data S1).

Note that in the above comparison, ACME was trained mainly on human peptide–MHC binding data while the original training data of NetMHCpan 3.0 included additional binding data from other species (which consisted of only 18% of its training data). To show that the improvement of our model was not due to the difference in the training data used, we also reimplemented a model with the same model structure and encoding strategy as in NetMHCpan 3.0 (Nielsen and Andreatta, 2016) (since the source code of NetMHCpan 3.0 was not available), and trained it using the exact same human peptide–MHC binding data as in ACME. This additional comparison based on five-fold cross-validation also confirmed the significant performance improvement achieved by ACME (Supplementary Data S1). Overall, the above results indicated that ACME can provide significantly more accurate predictions of peptide–MHC binding than the other state-of-the-art baseline method.

In addition, we also carried out a series of ablation tests to study the contributions of different modules in affinity prediction (Supplementary Notes Section S1.4, Supplementary Table S1 and Supplementary Fig. S6). To briefly summarize the results, the convolutional module plays a primary role in affinity prediction, while the attention module can also possess sufficient information for peptide–MHC binding prediction, as long as it is integrated into a suitable network architecture. Among all the architectures that we tested, ACME achieved the best performance in five-fold cross-validation.

3.2 Testing the generalizability of ACME

We conducted two additional tests to further demonstrate the generalizability of ACME (Supplementary Notes Section S1.5). We first tested ACME on 30 other independent datasets, i.e. the most recent IEDB benchmark datasets and compared its prediction performance to those of previous methods, whose Spearman rank correlation coefficient (SRCC) values had been reported previously in (Trolle *et al.*, 2015).

Our comparison showed that the prediction performances of other methods were roughly on the same level, while ACME achieved significantly better performance, with around 5 percentage points improvement in terms of SRCC (Fig. 3a, Supplementary Data S2 and Supplementary Fig. S7). For example, three versions of the widely used NetMHCpan program, including NetMHCpan 2.8, NetMHCpan 3.0 and NetMHCpan 4.0, achieved SRCC values of 0.512, 0.522 and 0.521, respectively. In comparison, ACME achieved an average SRCC of 0.569, which was 4.8 percentage points higher than NetMHCpan 4.0. In addition, among all the 30 datasets tested, ACME generally yielded better performance than NetMHCpan 4.0. For example, ACME outperformed NetMHCpan 4.0 by over 3 percentage points on 14 alleles, whereas NetMHCpan 4.0 had such better performance on only four alleles (Fig. 3b). The recently proposed deep learning-based model, ConvMHC (Han and Kim, 2017) is only applicable to 9-mers, making it unable to make predictions for many of the IEDB benchmark datasets. Therefore, we only tested the ConvMHC web server (http://jumong.kaist.ac.kr: 8080/convmhc) on those IEDB benchmark datasets containing 9-mer data and included all the datasets compatible with the server. The results showed that ACME achieved improvement of 9.3 percentage points on average in SRCC, with the most significant case exceeding 30 percentage points (Fig. 3c). The above results indicated that our model can be generalized well on other independent data and has better prediction performance than other baseline methods.

Next, we examined whether ACME can make accurate binding predictions for those novel MHC alleles without any training data. Here, the data of a specific allele were removed from training data and used as a test set. We then trained a model using the remaining training set and evaluated the performance of our model on this test set. All the alleles that had > 10 samples with both binders and non-binders were included in the test. As shown in Figure 3d and e, although the tested alleles were not seen in the training set, ACME still achieved relatively high performance. More specifically, for HLA-A alleles, ACME achieved an average PCC of 0.77 and an average AUROC of 0.90. For HLA-B, ACME achieved an average PCC of 0.68, which is lower than that of HLA-A. However, the average AUROC of ACME on HLA-B alleles reached 0.88, which indicated

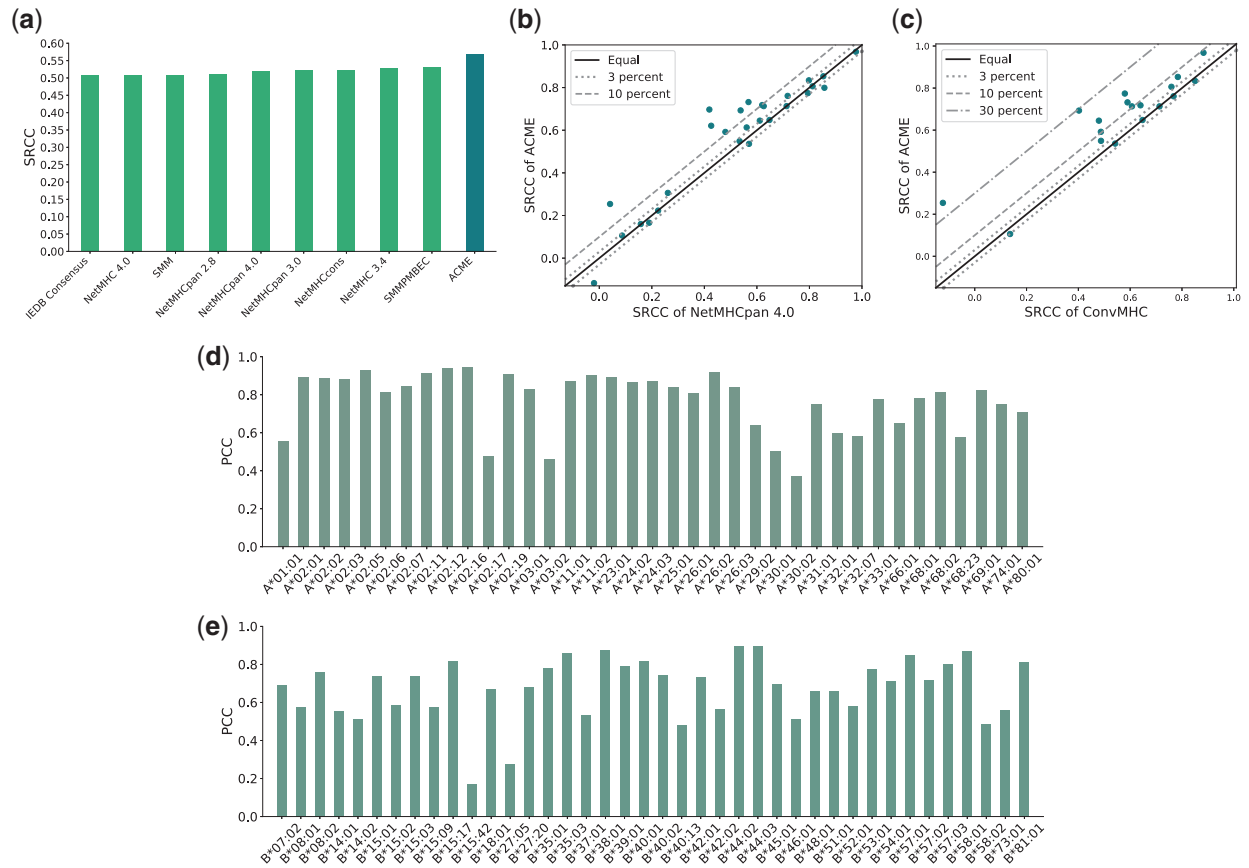

Fig. 3. Testing the generalizability of ACME. (**a**) The performances of different prediction algorithms on the most recent IEDB benchmark datasets, measured in terms of the Spearman rank correlation coefficient (SRCC). The SRCC scores of other prediction methods were obtained from (Trolle *et al.*, 2015). Each bar represents the averaged performance of a specific prediction algorithm over all the datasets tested. (**b**) Comparison of the performances of NetMHCpan 4.0 and ACME on the IEDB benchmark datasets. (**c**) Comparison of the performances of ConvMHC and ACME on the IEDB benchmark datasets. (**d–e**) The prediction performances of ACME for the novel alleles that were not seen in training data, evaluated in terms of the Pearson correlation coefficient (PCC) between predicted and measured binding affinities. (**d**) Results for HLA-A alleles. (**e**) Results for HLA-B alleles

that ACME still possessed relatively high generalizability for classifying peptides into binders and non-binders even for this challenging prediction task (Supplementary Notes Section S1.6, Supplementary Fig. S8 and Supplementary Data S3). This new result indicated that ACME can fully exploit the existing knowledge from the common alleles with available training data to make accurate predictions for those novel alleles with few or no training data.

3.3 Experimental validation

We selected the representative peptides with the highest and lowest predicted binding affinities and experimentally measured their binding affinities to further validate the predictive ability of our algorithm (Supplementary Notes Section S1.7). The human MHC alleles HLA-A*02: 01 and HLA-B*27: 05 were included in the validation as representative alleles of common and relatively rare alleles, respectively. For HLA-A*02: 01, we used flow-cytometry with T2 cells to measure the binding affinities of peptides. As shown in Figure 4a, the expression of HLA-A2 was increased when T2 cells were incubated with the peptides with high predicted binding affinities. Furthermore, some of the tested peptides with high predicted affinities even displayed equivalent or higher binding affinities when compared to the positive control peptide (OVAL235). More importantly, the predicted affinities agreed well with the experimentally measured values, with an SRCC of 0.73.

We also used the ProImmune REVEAL peptide–MHC binding assay to experimentally measure the binding affinities with HLA-B*27: 05. The experimental validation results (Fig. 4b) also showed that the binding affinities predicted by ACME matched the experimentally measured values, with an SRCC of 0.69. Detailed results of the above two experimental validation tests can also be found in Supplementary Data S4.

Taken together, the results above indicated that ACME can be adopted to accurately identify neoantigens. Admittedly, here the above experiments only focused on the relatively extreme cases, which might also be accurately predicted by other methods. Hence, we also included additional analyses to further investigate the cases where ACME and NetMHCpan 3.0 make very different predictions. More specifically, we downloaded the experimental dataset from another study (Khan *et al.*, 2017), which contained the peptide sequences derived from the HSV-1 genome and the corresponding experimentally measured binding affinities of these peptides with HLA-A*02: 01. We then compared the prediction results made by ACME and NetMHCpan 3.0 on this dataset. Among all the 434 9-mers in the dataset, 171 showed differences of more than 5 percentage points in the affinities predicted by ACME and NetMHCpan 3.0. For 149 out of these 171 peptides, ACME made more accurate predictions compared to NetMHCpan 3.0. Furthermore, there were 47 peptides showing differences of more than 10 percentage points in the affinities predicted by ACME and NetMHCpan 3.0, and

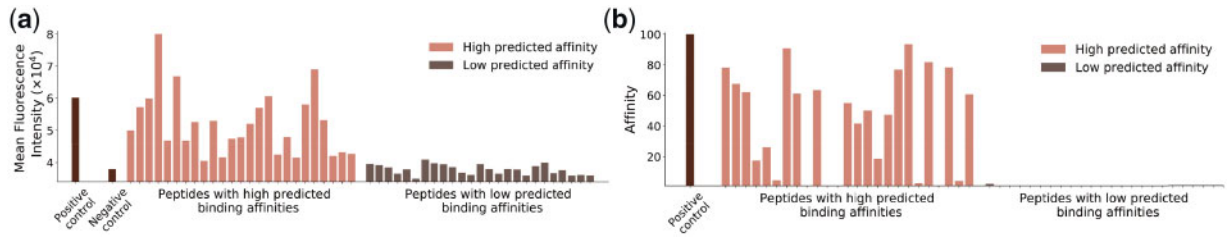

Fig. 4. Experimental validation on the selected representative peptides with the highest 25 and lowest 25 predicted binding affinities. (**a**) The flow cytometry results for allele HLA-A*02: 01, where a higher mean fluorescence intensity (MFI) value indicates a higher affinity. (**b**) The ProImmune REVEAL peptide–MHC binding assay results for allele HLA-B*27: 05. Synthesis of one peptide with low predicted binding affinity failed, and thus its binding affinity was not tested. Both experimental validation results showed that our peptide–MHC binding predictions are generally accurate

ACME made more accurate predictions on 45 of them. The information of these 45 peptides is listed in Supplementary Table S2 in the Supplementary Notes. More details of these additional analyses can also be found in Supplementary Notes Section S1.7 and Supplementary Figure S9.

All the above validation results further demonstrated that our algorithm can yield accurate predictions and thus provide a practically useful tool to identify neoantigens in human cancer.

3.4 ACME uncovers the underlying rules of peptide–MHC binding

To overcome the black-box nature of deep learning, here we combined our convolutional neural network with an attention module to enhance the explainability of our model (Supplementary Notes Section S1.8). To validate the quality of the attention values assigned to individual positions, we first conducted a masking test and confirmed that the positions with higher attention values tend to make a greater contribution to the prediction of peptide–MHC binding in our ACME framework (Supplementary Notes Section S1.9 and Supplementary Fig. S10).

As a first attempt, we visualized the motifs for several MHC alleles whose binding patterns had been well characterized previously (Fig. 5a). Expectedly, we observed similar preferences for amino acid types at different peptide positions as reported in previous studies (Yusim *et al.*, 2016). For instance, our model was able to identify the Lys at position 9 as an anchor residue (i.e. it makes a major contribution to binding), for the peptide bound to HLA-A*11: 01. Also, for HLA-B*40: 01, the previously reported anchor residues Glu at position 2 and Leu at position 9 were also successfully identified by our model as important residues.

In addition, by aligning the sequence motifs discovered by ACME with previously reported structural profiles of peptide–MHC complexes (Fig. 5b), we found that our model was also able to capture the difference of local biochemical environments caused by the polymorphisms in different MHC molecules, especially at the termini of the peptides. For instance, our analyses demonstrated that the Lys or Arg residue at position 9 of a peptide binder significantly contributes to the binding to HLA-A*68: 01, which can actually be well supported by a previously solved high-resolution structure of HLA-A*68: 01 complexed with a 9-mer peptide (PDB code 4HWZ) (Niu *et al.*, 2013), where the C terminus Lys of the peptide is involved in a hydrogen bond network with surrounding residues in the bound MHC molecule. In particular, the ϵ-amino group of this Lys residue forms a hydrogen bond with Asp116 of the MHC molecule. In contrast, for the HLA-B*57: 03 protein whose binding pocket is generally formed by hydrophobic residues, our motif analysis indicated a preference for hydrophobic residues (e.g. Leu, Phe or Trp) at the C terminus, which was consistent with the previous structural data (PDB code 2BVP) (Stewart-Jones *et al.*, 2005) showing that the C-terminus Trp of the peptide is involved in an aromatic stacking with Tyr123 and makes close contacts with several other residues in the MHC molecule, including Trp147, Tyr116, Ile95 and Ile80. In addition, we found that for our predicted peptides bound to allele HLA-B*44: 02, the anchor residue Glu2 near the N terminus is engaged in multiple hydrogen bonds with neighboring MHC residues, and its γ-carboxyl group forms a salt bridge with the ϵ-amino group of Lys45 in the MHC protein (PDB code 1M6O) (Macdonald *et al.*, 2003). All these analysis results based on the existing structural data strongly supported the biological relevance of the binding motifs revealed by ACME.

Next, we sought to extend the above analyses to the MHC alleles whose binding preferences lack experimental characterization. To test the validity of our results, we first confirmed that our model can produce the sequence motifs verified by previous studies for several alleles that were not included in training data (Supplementary Fig. S11a). For instance, the binding data of HLA-A*66: 02 and HLA-B*44: 09 were not included in our training set. Previous studies showed that the anchor residues of HLA-A*66: 02 are Val or Thr at position 2 and Arg at position 9 (Bade-Doeding *et al.*, 2005) and the anchors for HLA-B*44: 09 are Glu at position 2 and Phe or Leu at position 9 (Huyton *et al.*, 2012). These results were in accordance with the binding motif predicted by our model (Supplementary Fig. 11a).

Furthermore, to investigate the universal patterns in peptide–MHC interactions, we used ACME to generate motifs for a large number of alleles, including those that were not previously characterized. Overall, we found that positions 2 and 9 in the peptides generally play the most vital roles in peptide–MHC binding, which was consistent with the findings of previous studies (Liu and Gao, 2001). Our analyses also showed that the aliphatic amino acids Leu and Val, the aromatic amino acids Phe and Tyr and the basic amino acids Lys and Arg are among the most common anchor residues. In addition, we found that different alleles may employ different anchor strategies to facilitate specific peptide binding. More specifically, the peptides bound to some alleles, e.g. HLA-A*03: 01, HLA-A*02: 19 and HLA-A*24: 02, have anchor residues at both ends (Supplementary File 1), while peptides bound to other alleles, such as HLA-A*03: 02, HLA-A*11: 01 and HLA-A*11: 02, are mainly anchored at one end. We found that the residues in the middle of the peptides can also make a significant contribution to binding in some alleles like HLA-B*15: 42.

In addition to the discovery of the residues that strengthen binding, we were also interested in identifying the sequence features that may impair peptide–MHC binding. In contrast to the above analyses which focused on the high-affinity binding peptides, here we

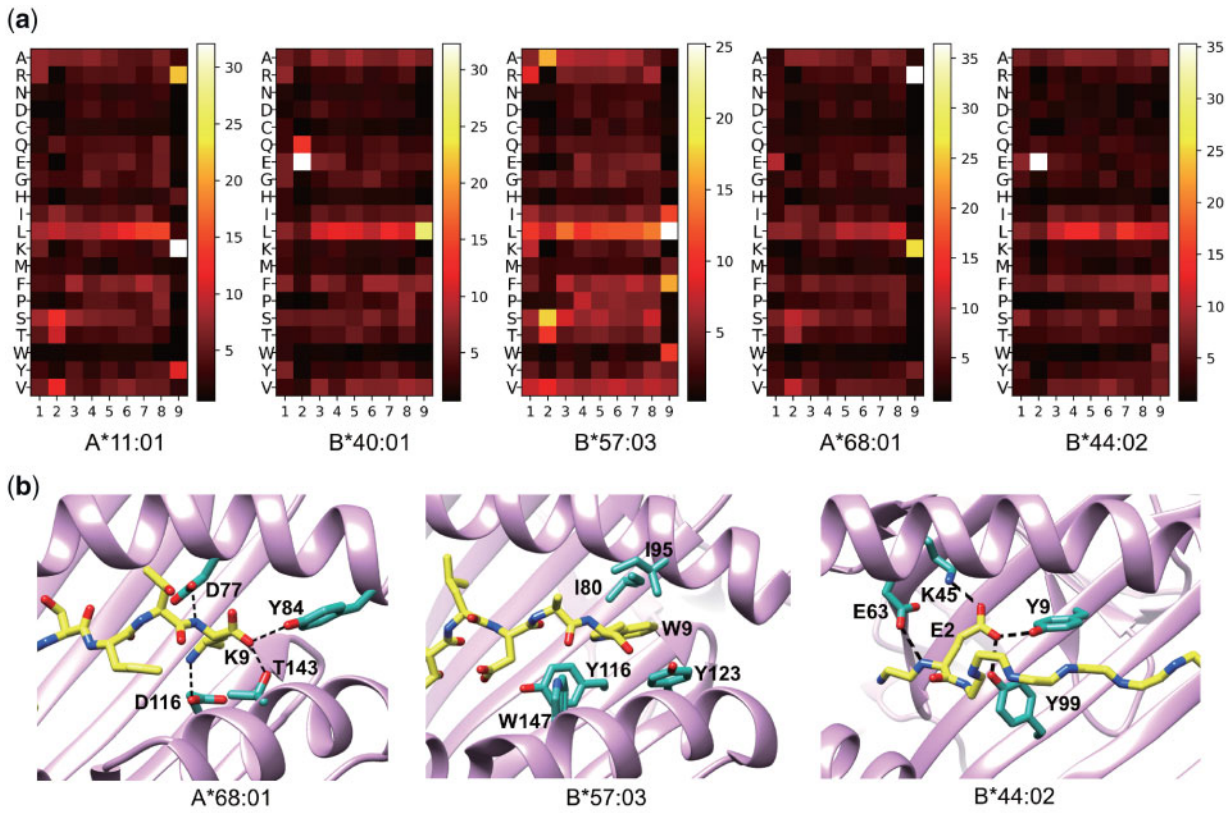

Fig. 5. Examples of sequence motifs for the peptides bound to different MHC alleles. (**a**) The heatmaps visualizing the representative motifs of several well-characterized MHC alleles. Only the motifs of 9-mers are shown here. Each column corresponds to a position in the peptide sequence and each row stands for an amino acid type. Brighter pixels mean that the corresponding residues have higher accumulated attention scores, and are thus considered to be more important in peptide–MHC binding (The attention scores weigh the importance of the features in corresponding positions, and the formal definition of attention scores can be found in Supplementary Notes Section S1.8). (**b**) Structural data supporting the peptide sequence motifs revealed by ACME. The structural profiles here are visualized using UCSF Chimera (Pettersen *et al.*, 2004)

generated a set of 'non-binder' motifs using the peptides with the lowest predicted affinities (Supplementary Fig. 11b). These non-binder motifs displayed several clear attention loci, e.g. Ser at position 9 in HLA-A*02: 01 and Lys at position 7 in HLA-B*35: 01, indicating that most likely these residues are detrimental to binding. To test whether or not these residues can actually impair peptide–MHC binding, we conducted an additional test in which we replaced the original residues of the peptides with random amino acids or the above found anti-binding amino acids, respectively. It turned out that replacing the corresponding residues with these anti-binding amino acids indeed resulted in significantly lower predicted binding affinities compared to the replacement by random amino acids (Supplementary Fig. S12). This additional test result can thus support our discovery of the non-binder motifs.

4 Discussion

In this study, we have developed a new pan-specific algorithm, called ACME, for peptide–MHC class I binding prediction. The algorithm uses a convolutional neural network to integrate the features extracted from both peptide and the MHC sequences and has a built-in attention module to help extract interpretable binding patterns. Comprehensive tests on available benchmark datasets have demonstrated the superior prediction performance of ACME over other state-of-the-art methods. In addition, we have shown that ACME can be used to accurately generate the binding motifs for different MHC alleles, which can be further used to investigate the patterns of peptide–MHC class I binding. On the other hand, there are still several future directions that can be explored to further improve our model. For example, as we have shown in the ablation tests (Supplementary Notes Section S1.4), the contribution of the attention module in affinity prediction is limited. Future studies are needed to find a better way to integrate attention into a model so that its functions in prediction can be further exploited. Besides, currently we only used limited experimental data to analyze the cases where different models make distinct predictions. Large-scale experimental validation of such cases will be helpful for developing better prediction algorithms. In the future, we will also incorporate more information such as peptide transportation and TCR recognition to further improve the prediction of antigen peptides. We expect that incorporating information about these biological processes into our prediction model will further extend the application potentials of our model in cancer immunotherapies.

Acknowledgements

We are grateful to Prof. Nielsen, M. and Prof. Shi, Y. for their helpful discussions.

Funding

This work was supported by the funds from the Turing AI Institute of Nanjing; the National Natural Science Foundation of China [61872216, 61472205, 81630103, 81772737]; National Science Foundation Projects of

Guangdong Province [2017B030301015]; and the Shenzhen Municipal Government of China [JCYJ20170413161749433].

Conflict of Interest: none declared.

References

Andreatta,M. and Nielsen,M. (2016) Gapped sequence alignment using artificial neural networks: application to the MHC class I system. *Bioinformatics*, **32**, 511–517.

Bade-Doeding,C. *et al*. (2005) Peptide-binding motif of HLA-a*6603. *Immunogenetics*, **56**, 769–772.

Bahdanau,D. *et al*. (2014) Neural machine translation by jointly learning to align and translate. *ICLR*, **1409**.

Carreno,B.M. *et al*. (2015) A dendritic cell vaccine increases the breadth and diversity of melanoma neoantigen-specific T cells. *Science*, **348**, 803–808.

Engelhard,V.H. (1994) Structure of peptides associated with class I and class II MHC molecules. *Annu. Rev. Immunol*., **12**, 181–207.

Han,Y. and Kim,D. (2017) Deep convolutional neural networks for pan-specific peptide–MHC class I binding prediction. *BMC Bioinformatics*, **18**, 585.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, **89**, 10915–10919.

Hoof,I. *et al*. (2009) Netmhcpan, a method for MHC class i binding prediction beyond humans. *Immunogenetics*, **61**, 1.

Hu,H. *et al*. (2019) Deephint: understanding HIV-1 integration via deep learning with attention. *Bioinformatics*, **35**, 1660–1667.

Hu,Z. *et al*. (2018) Towards personalized, tumour-specific, therapeutic vaccines for cancer. *Nat. Rev. Immunol*., **18**, 168–182.

Huyton,T. *et al*. (2012) Residue 81 confers a restricted c-terminal peptide binding motif in HLA-b*44: 09. *Immunogenetics*, **64**, 663–668.

Jurtz,V. *et al*. (2017) Netmhcpan-4.0: improved peptide–MHC class i interaction predictions integrating eluted ligand and peptide binding affinity data. *J. Immunol*., **199**, 3360–3368.

Karosiene,E. *et al*. (2012) Netmhccons: a consensus method for the major histocompatibility complex class i predictions. *Immunogenetics*, **64**, 177–186.

Khan,A.A. *et al*. (2017) Bolstering the number and function of hsv-1–specific cd8+ effector memory T cells and tissue-resident memory T cells in latently infected trigeminal ganglia reduces recurrent ocular herpes infection and disease. *J. Immunol*., **199**, 186–203.

Kim,Y. (2014) Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751. doi: 10.3115/v1/D14-1181.

Kim,Y. *et al*. (2014) Dataset size and composition impact the reliability of performance benchmarks for peptide–MHC binding predictions. *BMC Bioinformatics*, **15**, 241.

Kingma,D. and Ba,J. (2014) *Adam: A Method for Stochastic Optimization*. International Conference for Learning Representations (ICLR).

Li,J. *et al*. (2015) A hierarchical neural autoencoder for paragraphs and documents. *Int. Jt. Conf. Nat. Language Process*., **1**, 1106–1115.

Liu,J. and Gao,G.F. (2001) Major histocompatibility complex: interaction with peptides. *e LS*. doi: 10.1002/9780470015902.a0000922.pub2.

Lundegaard,C. *et al*. (2008) Netmhc-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8–11. *Nucleic Acids Res*., **36**, W509–W512.

Macdonald,W.A. *et al*. (2003) A naturally selected dimorphism within the HLA-b44 supertype alters class I structure, peptide repertoire, and T cell recognition. *J. Exp. Med*., **198**, 679–691.

Madden,D.R. (1995) The three-dimensional structure of peptide–MHC complexes. *Annu. Rev. Immunol*., **13**, 587–622.

Mnih,V. *et al*. (2014). Recurrent models of visual attention. *Neural information processing systems*, pp. 2204–2212.

Nair,V. and Hinton,G.E. (2010) Rectified linear units improve restricted Boltzmann machines. *ICML*, *Pp*. 807–814.

Nielsen,M. and Andreatta,M. (2016) Netmhcpan-3.0; improved prediction of binding to mhc class i molecules integrating information from multiple receptor and peptide length datasets. *Genome Med*., **8**, 33.

Nielsen,M. *et al*. (2003) Reliable prediction of t-cell epitopes using neural networks with novel sequence representations. *Protein Sci*., **12**, 1007–1017.

Nielsen,M. *et al*. (2007) Netmhcpan, a method for quantitative predictions of peptide binding to any hla-a and-b locus protein of known sequence. *PLoS One*, **2**, e796.

Niu,L. *et al*. (2013) Structural basis for the differential classification of hla-a*6802 and hla-a*6801 into the a2 and a3 supertypes. *Mol. Immunol*., **55**, 381–392.

Ott,P.A. *et al*. (2017) An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature*, **547**, 217–221.

Pearson,H. *et al*. (2016) Mhc class i-associated peptides derive from selective regions of the human genome. *J. Clin. Investig*., **126**, 4690–4701.

Peters,B. and Sette,A. (2005) Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. *BMC Bioinformatics*, **6**, 132.

Pettersen,E.F. *et al*. (2004) Ucsf chimera-a visualization system for exploratory research and analysis. *J.Comput. Chem*., **25**, 1605–1612.

Robinson,J. *et al*. (2014). The ipd and imgt/hla database: allele variant databases. **43**.

Stewart-Jones,G.B.E. *et al*. (2005) Structures of three hiv-1 hla-b*5703-peptide complexes and identification of related hlas potentially associated with long-term nonprogression. *J. Immunol*., **175**, 2459–2468.

Trolle,T. *et al*. (2015) Automated benchmarking of peptide–MHC class i binding predictions. *Bioinformatics*, **31**, 2174–2181.

Vang,Y.S. and Xie,X. (2017) Hla class i binding prediction via convolutional neural networks. *Bioinformatics*, **33**, 2658–2665.

Vita,R. *et al*. (2015) The immune epitope database (iedb) 3.0. *Nucleic Acids Res*., **43**, D405–D412.

Walter,S. *et al*. (2012) Multipeptide immune response to cancer vaccine ima901 after single-dose cyclophosphamide associates with longer patient survival. *Nature Med*., **18**, 1254–1261.

Yewdell,J.W. and Bennink,J.R. (1999) Immunodominance in major histocompatibility complex class I–restricted T lymphocyte responses. *Annu. Rev. Immunol*., **17**, 51–88.

Yusim,K. *et al*. (2016) Hiv molecular immunology 2015. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Zhang,S. *et al*. (2017) Analysis of ribosome stalling and translation elongation dynamics by deep learning. *Cell Syst*., **5**, 212–220.e6.