Bioinformatics, 37(22), 2021, 4172–4179
doi: 10.1093/bioinformatics/btab422
Advance Access Publication Date: 7 June 2021
Original Paper

Systems biology

BERTMHC: improved MHC–peptide class II interaction prediction with transformer and multiple instance learning

Jun Cheng [1,*], Kaïdre Bendjama [2], Karola Rittner[2] and Brandon Malone [1,*]

[1]NEC Laboratories Europe GmbH Kurfuersten-Anlage 36, 69115 Heidelberg, Germany and [2]Transgene, Boulevard Gonthier d'Andernach, 67400 Illkirch-Graffenstaden, France

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on November 24, 2020; revised on May 17, 2021; editorial decision on May 23, 2021; accepted on June 4, 2021

Abstract

Motivation: Increasingly comprehensive characterization of cancer-associated genetic alterations has paved the way for the development of highly specific therapeutic vaccines. Predicting precisely the binding and presentation of peptides to major histocompatibility complex (MHC) alleles is an important step toward such therapies. Recent data suggest that presentation of both class I and II epitopes are critical for the induction of a sustained effective immune response. However, the prediction performance for MHC class II has been limited compared to class I.

Results: We present a transformer neural network model which leverages self-supervised pretraining from a large corpus of protein sequences. We also propose a multiple instance learning (MIL) framework to deconvolve mass spectrometry data where multiple potential MHC alleles may have presented each peptide. We show that pretraining boosted the performance for these tasks. Combining pretraining and the novel MIL approach, our model outperforms state-of-the-art models based on peptide and MHC sequence only for both binding and cell surface presentation predictions.

Availability and implementation: Our source code is available at https://github.com/s6juncheng/BERTMHC under a noncommercial license. A webserver is available at https://bertmhc.privacy.nlehd.de/

Contact: s6juncheng@gmail.com or bmmalone@gmail.com

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

1 Introduction

The adaptive immune system plays a central role in immune response against foreign molecules, such as pathogens or cancerous cells. The adaptive immune system is classically divided into two components (Janeway *et al.*, 2001): humoral immunity, which concerns antibody generation by mature B cells, and cell-mediated immunity, which entails stimulation of cytotoxic CD8+ T cells among other things. The major histocompatibility complex (MHC) class II plays an important role in both humoral and cell-mediated immunity (Janeway *et al.*, 2001); the human leukocyte antigen (HLA) is the human version of MHC. The primary role of MHC class II is to bind to and then present peptide sequences from exogenous proteins on the cell surface. This MHC–peptide complex leads to the stimulation of CD4+ T cells, or 'helper T cells'. The helper T cells may then stimulate either the humoral or cell-mediated immune response pathways (Al-Daccak *et al.*, 2004). MHC class II molecules are mostly found in professional antigen presenting cells (Lang *et al.*, 2001), such as dendritic cells. Among the MHC class II molecules, each person typically has multiple alleles from HLA-DP, HLA-DQ and HLA-DR (Rock *et al.*, 2016). Importantly, different people have different MHC alleles, although some alleles are more common than others. Further, class II MHCs are encoded by genes characterized by a very high level of polymorphism (Neefjes *et al.*, 2011). The different MHC alleles have different amino acid sequences and structures, leading to the presentation of different specificity and different presented repertoire across individuals.

This high level of polymorphism is an evolutionary trait allowing the immune system to react against a very large number of pathogens at the individual level. The corollary implication of this high level of polymorphism is that each individual will eventually present different peptides. This has critical implications when attempting to design vaccines and select antigens. Historically, vaccine development involved the use of large antigens, representative of the target pathogen. Because of their size, these antigens are likely to contains peptides sequences that will be presented by a wide spectrum of HLA genotypes. More recent vaccines, and in particular vaccines directed to cancer cells, are targeted at restricted sequences as the

pathogen—the cancer cell—is minimally different from the host cell (Tanyi *et al.*, 2018). In order to increase vaccine efficiency, it is critical to identify sequences that are likely to be presented by the class II HLA in the individuals that will receive the vaccine.

The presentation of peptides to T cells involves a series of processes. Important steps include binding between MHC molecules and peptides, as well as presentation of the MHC–peptide complex to the cell surface. Experimental assays have been developed to study and quantify many of these processes. The binding affinities between MHC molecules and peptides can be measured by *in vitro* binding assays (Sidney *et al.*, 2013). Mass spectrometry can be used to detect peptides eluted from the cell surface to determine peptide presentation (Purcell *et al.*, 2019). Thousands of data points have been generated by such assays for hundreds of different MHC molecules (Vita *et al.*, 2019). However, all these wet lab methods remain labor intensive and subject to a number of biases.

Given the importance of the problem and the availability of the data, many methods have been developed to predict MHC–peptide binding and peptide presentation (Peters *et al.*, 2020). In some approaches, a single model is trained specifically for each MHC allele; other approaches instead train a single model (a *pan model*) covering all MHC alleles. The prediction performances of MHC class I models have reached a high level (auROC > 0.95, O'Donnell *et al.*, 2020; Peters *et al.*, 2020; Reynisson *et al.*, 2020b). On the other hand, models for class II still have limited performance. Despite recent progress (Reynisson *et al.*, 2020b), there is still a need for better performing models. One significant limiting factor for MHC class II models is the limited amount of training data compared to class I. Thus, models that can efficiently use all of the limited available data and transfer knowledge from other sources are extremely valuable.

Recent advances in natural language processing have enabled techniques to train complex models that understand semantics from text without labels (self-supervised learning) (Devlin *et al.*, 2019; Peters *et al.*, 2018). Such models are trained to predict words masked out in a sentence or to predict the next word or sentence following some context. Similar techniques have also been applied to proteins (Heinzinger *et al.*, 2019; Nambiar *et al.*, 2020; Rao *et al.*, 2019). Since these models do not require labels to train, they can be trained on very large corpora of protein sequences across many species. One example of these models is the TAPE model (Rao *et al.*, 2019), which was trained with 31 million protein sequences from the Pfam database (El-Gebali *et al.*, 2019). The model has been shown to be helpful in a variety of downstream tasks such as remote protein homology prediction and stability prediction. Detailed analysis of the model has shown that it captures long-range interactions in 3D structure (Vig *et al.*, 2020). It is highly relevant to explore whether pretrained protein sequence models can be helpful for MHC–peptide binding and presentation prediction, especially for MHC class II where much less data are available.

As mentioned, each person has multiple MHC class II molecules; thus, typical mass spectrometry experiments cannot precisely identify the MHC molecule to which a peptide was bound. In designing personalized vaccines, we are also interested in the likelihood of response given all the MHC alleles an individual carries. Therefore, it is important to develop algorithms to predict the likelihood of presentation for a peptide given a set of MHC molecules.

Here, we focus on developing models for predicting MHC–peptide binding and presentation for MHC class II. We show that models taking advantage of self-supervised pretraining from large corpora of protein sequences can achieve better performance on both binding and presentation prediction tasks. We found pretraining to be extremely valuable in the case where training data are limited. Additionally, we propose a novel multiple instance learning (MIL) algorithm to account for the limitation that mass spectrometry data often cannot precisely identify the exact MHC molecule to which a peptide was bound. We foresee our work to be valuable in T-cell-based immunotherapy and provide new directions for training peptide binding and presentation models with limited training data.

2 Materials and methods

2.1 MHC class II binding data

To train the MHC class II binding model, we used the data from Jensen *et al.* (2018), since it has been designed to minimize the overlap between the training and evaluation sets. The original data were collected from the Immune Epitope Database (IEDB, Vita *et al.*, 2019, accessed on June 30, 2020) up to the year 2016. The data consist of 134 281 data points and covers HLA-DR, HLA-DQ, HLA-DP and H-2 mouse MHC allele. The affinity labels were transformed from IC_{50} to values between 0 and 1 with the formula $1 - \log(IC_{50})/\log(50\,000)$.

2.2 Independent MHC class II binding data

The data from Jensen *et al.* were collected from IEDB up to the year 2016. To benchmark on an independent dataset where no model has been used for training or validation, we collected quantitative binding data from IEDB and filtered out data already used in Jensen *et al.* In addition, we collected further independent binding data from the Dana–Farber repository (Zhang *et al.*, 2011). In the end, we collected 2 413 additional MHC–peptide pairs covering 47 MHC class II alleles. The complete list of benchmark data is available in Supplementary Table S1.

2.3 MHC class II presentation data

To train the MHC class II mass spectrometry presentation model, we used the data curated from Reynisson *et al.* (2020b). The original data were curated from IEDB and other public sources. The data cover 41 MHC class II alleles with peptide lengths ranging from 13 to 21. Each data point consists of the peptide ligand, the source protein and list of possible MHC class II alleles bound to the peptide. The data points where only one MHC allele is unambiguously given are referred to as single-allele (SA) data, whereas the data points where multiple potential alleles are given are referred to as multiallele (MA) data. Reynisson *et al.*, selected negative peptides by randomly sampling from the UniProt database. Peptide lengths for the negatives are sampled uniformly from 13 to 21.

2.4 Independent MHC class II benchmark presentation data

We also filtered for further independent datasets from IEDB (accessed on June 30, 2020) on which no existing models had been trained or evaluated. All data presented in the training and evaluation set of Reynisson *et al.* were filtered out. We only kept 8 170 peptides with length between 13 and 21, in line with Reynisson *et al.*, and alleles which have more than 50 positive peptides. To generate negative mass spectrometry decoys, we randomly sampled 10× negative peptides from the human proteome with the same peptide length distribution as the positive peptides per allele. This evaluation data of 95 638 peptides is provided in Supplementary Table S2.

2.5 Patient mass spectrometry data

To evaluate our model on an independent set of data, we generated a dataset of HLA class II presented peptides from six patients with cancer. Briefly, cancer tissue was collected on surgical material in patients undergoing nonsmall cell lung cancer resection. The collection of biological material was performed in accordance to international and local clinical research regulation and subject to ethical review boards approvals. Accordingly, participation to the study was contingent to informed consent from individual patients. After tissue lysis, solubilized HLA complexes were purified with antibody-conjugated resin (clone L243). HLA-bound peptides were eluted in acidic condition, further purified and desalted before final mass spectrometric analysis. Eluted peptides were analyzed by data-dependent mass spectrometry on an HF-X hybrid quadrupole-Orbitrap mass spectrometer. Peptides were identified with MaxQuant (Cox and Mann, 2008) (Supplementary Methods). In total 15 277 unique peptides were identified. Between 1 500 and 7 000 unique peptide sequences were identified from each sample

(Supplementary Table S3). The detailed HLA types of all patients are provided in Supplementary Table S3.

2.6 Pretrained protein BERT model

We use the pretrained bidirectional encoder representations from transformers (BERT, Devlin *et al.*, 2019) neural network from the TAPE repository to model the input amino acid sequences. The TAPE model was trained with *self-supervised learning* from a dataset of over 31 million protein sequences. Briefly, each amino acid is encoded as a *token*. The input protein sequence is encoded as a sequence of tokens: $\mathbf{x} = (x_1, \ldots, x_L)$. Taking unlabeled protein sequence as input, the TAPE model was trained with two tasks. One task is bidirectional *next-token prediction* (predicting $p(x_i|x_1, x_2, \ldots, x_{i-1})$ and $p(x_i|x_{i+1}, x_{i+2}, \ldots, x_L)$), and the other task is *masked-token prediction* (predicting $p(x_{\text{masked}}|x_{\text{unmasked}})$). The BERT model masks 15% of the input tokens randomly for prediction. The model has 12 layers with 12 self-attention heads [Equation (1)] in each layer, which enables the model to learn long distance interactions. For an input amino acid sequence **x**, the outputs of the model are L continuous vectors of dimension 768 corresponding to the input amino acids. We refer to Rao *et al.* (2019), for details of the pretrained protein BERT model.

Self-Attention: Self-attention learns the interaction (*attention*) of all possible amino acid pairs in the input sequence. Specifically, for each input amino acid sequence $\mathbf{x} = (x_1, \ldots, x_L)$, self-attention learns an attention score $\alpha_{ij} > 0$ for each pair of amino acids i, j where $\sum_j \alpha_{ij} = 1$. The attention scores are computed from the normalized dot product of *query* vectors and *key* vectors followed by a softmax operation. The output of a self-attention layer is a weighted sum of the *value* vector by the attention weights. The operations of a self-attention layer written in matrix form are as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \qquad (1)$$

where d_k is the dimension of the key vectors (chosen as 64), Q is a query matrix, K is a key matrix and V is a value matrix. The query, key and value matrices in this case are different trainable linear projections of the layer input. Instead of a single attention function, 12 such attention heads are used. In each layer, the outputs from each of the 12 self-attention heads are concatenated to give a final continuous vector of dimension 768 for each amino acid. The query, key and value matrices of each self-attention head are independent. Each self-attention layer transforms the original input sequence into a deeper representation of the same length. We refer to the original publication (Vaswani *et al.*, 2017) for more details on the complete transformer architecture.

2.7 Supervised training of MHC class II models

We trained two BERT models, one each for MHC class II binding and presentation prediction. Both models were initialized with the BERT model parameters from the TAPE repository. Both models are pan models, and they take as input the concatenated amino acid sequences for the MHC pseudosequences (Jensen *et al.*, 2018; Reynisson *et al.*, 2020b) followed by the peptide. Because peptides have variable lengths, we pad all sequences to the same length with an input padding token. The target for the binding prediction model is the (real-valued) binding affinity between the peptide and MHC, while the target for the presentation models is binary (presented or not).

As shown in Figure 1, our model architecture consists of three main components: the BERT component, a pooling layer, and a final multilayer perceptron (MLP) block. The MHC pseudosequences all have the same length by design, and input peptides are padded or truncated to length of 24. The concatenated input sequence is tokenized and used as input for the BERT component. We inherited the BERT architecture from the pretrained TAPE model (Rao *et al.*, 2019) without modification. As described previously, the BERT model produces a set of continuous vectors. The vectors

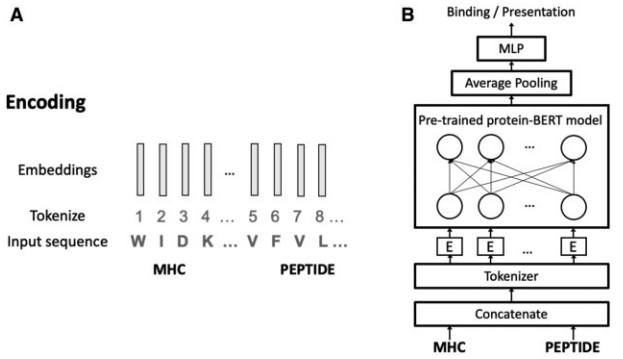

Fig. 1. (**A**) Encoding of MHC and peptide sequences. MHC and peptide sequences were first encoded into tokens with a tokenizer, where each token corresponds to a single amino acid. Peptide sequences are padded to the same length (24) with a pad token. Peptides longer than 24 are trimmed from the end. Each token is then embedded into a 768-dimensional vector with a trainable embedding layer. The tokens are integer values corresponding to the entries of the embedding matrix (parameters) of the embedding layer. (**B**) BERTMHC model architecture. The encoding layer is followed by 12 multihead self-attention layers with 12 attention heads at each layer. The outputs of the 12 heads are concatenated. Each self-attention layer transforms the input sequence into a deeper representation of the same 768 dimensions (each head transforms the input into dimension of 64). The output of the BERT architecture is then mean-pooled along the sequence dimension. A 2-layer feed forward neural network uses that to predict either binding affinity or presentation

corresponding to padding tokens are masked, and the remaining vectors are then pooled by taking the mean over the sequence dimension. This vector is then used as input for the MLP, which consists of two fully connected layers with hidden dimension of 512. The MLP is then used to predict either the MHC binding affinity (regression) or cell surface presentation (classification). We use the standard mean squared error loss function for the binding model and the weighted cross-entropy loss function (weight 10 for the positive class) for the presentation model when training with SA data. The loss function for training with MA data is described in Section 2.8. We train the model end-to-end with backpropagation; i.e. we also optimize the parameters from the pretrained TAPE model.

For both the binding and presentation models, we used random search to identify the best hyperparameter combinations on the first cross-validation fold. The random search for hyperparameters entailed randomly drawing a set of hyperparameters, including the learning rate, batch size, weight decay, MLP dimension and peptide encoding length. The best performing hyperparameters in the held-out evaluation set were used for the remaining analysis. We used an initial learning rate of 0.15 for the binding model and 0.01 (with weight decay 0.0001) for the presentation model. All models were trained with stochastic gradient descent with momentum of 0.9; the learning rate was reduced by a factor of 0.1 after 2 epochs of no performance improvement. After identifying high-quality hyperparameters, we use them to train three models with different initializations for each cross-validation fold for the binding model; only one model per fold was trained for the presentation task. The two tasks were trained independently.

2.8 MA deconvolution with MIL

In the MA presentation data, each peptide is associated with a bag of alleles. The bag is labeled as positive if at least one of the alleles presented the peptide; otherwise, the bag is labeled as negative. Training from such data has been performed in other research, and is sometimes referred to as *deconvolution* (Alvarez *et al.*, 2019; Bassani-Sternberg and Gfeller, 2016; Reynisson *et al.*, 2020a,b).

We model the training of the prediction model f_θ from the MA data as an MIL problem. We denote the ith bag with m alleles as $A_i = \{a_{i1}, a_{i2}, \ldots, a_{im}\}$ and the corresponding peptide sequence as s_i. At each training step, we predict the probability $p(y_{ij} = 1|x_{ij})$ for every instance $x_{ij} = (a_{ij}, s_i)$ in the bag as $\hat{y}_{ij} = f_\theta(a_{ij}, s_i)$ with our neural network model f_θ. A symmetric pooling operator **h** is used to

calculate the prediction of the bag from the predictions of instances within it.

We further propose to incorporate the confidence of the deconvolution operation. Since predicted probabilities by machine learning models are often uncalibrated, they do not correspond to the true model confidence (Niculescu-Mizil and Caruana, 2005). We calibrate the predicted probabilities $\hat{p}_i$ to $\mathcal{C}(\hat{p}_i)$. The probability calibration step is performed with isotonic regression (Barlow *et al.*, 1972, implemented in *scikit-learn*) at the beginning of each training epoch. Specifically, the isotonic regression model was fit with the predicted logits as the covariate and the labels on the training set as the response variable. At each training epoch, we weight each positive data point i from deconvolution by the calibrated predicted probability of being positive $\mathcal{C}(\hat{p}_i)$. We do not weight the negative sets since there is no ambiguity with their labels. Therefore, we compute the loss for the MA data as follows:

$$\begin{aligned}\hat{y}_i &= \mathbf{h}\{f_\theta(a_{i1}, s_i), f_\theta(a_{i2}, s_i), \ldots, f_\theta(a_{im}, s_i)\} \\ \ell(\theta) &= -\frac{1}{N_{\text{Pos}}}\sum_{i=0}^{N_{\text{Pos}}}(\mathcal{C}(\hat{p}_i)\cdot w\cdot \log(\hat{y}_i)) \\ &-\frac{1}{N_{\text{Neg}}}\sum_{i=0}^{N_{\text{Neg}}}(\mathbb{E}_{j\sim P_i(X_i)}\log(1-\hat{y}_{ij})),\end{aligned} \tag{2}$$

where $\hat{p}_i$ is the prediction of $p(y_i = 1|A_i, s_i)$ from the previous epoch of the model, $\mathcal{C}$ is the calibration function, and w is the weight for the mass spectrometry positive class for compensating class imbalance in the dataset. We use $w = 10$ for all presentation models. We used max pooling as the pooling operator $\mathbf{h}$, although other pooling operations, such as attention-based pooling (Ilse *et al.*, 2018), are also applicable. For computational reasons, we perform negative sampling with a probability distribution $P_i(X_i)$ instead of using all negative samples. We use the most likely positive example predicted by the current model from the negative bag. Therefore, the sampling distribution $P_i(X_i) = P_i(x_{i1}, x_{i2}, \ldots, x_{im})$ is defined as

$$P_i(x_{ij}) = \begin{cases} 1, & \text{if } f_\theta(x_{ij}) = \max(f_\theta(x_{i1}), f_\theta(x_{i2}), \ldots, f_\theta(x_{im})) \\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

In addition, we follow a two-stage training procedure. Before multiple instance training with MA data, we first train the model only with SA data. After eight epochs, we combine SA and MA data and train them jointly with MIL, where the bag for SA data has only one element. We stop the training process if the model performance [average precision (AP)] on the evaluation set does not improve after five epochs. Algorithm 1 summarizes the probability reweighted MIL algorithm.

Algorithm 1: Probability Reweighted Multiple Instance Learning

Input: Training data $\{X_i, y_i\}_{i\in 1\ldots N}$, where $X_i := (s_i, A_i)$, $y_i \in \{0, 1\}$;
Initialize the θ_0 as the model trained from SA data ;
$\theta_k \leftarrow \theta_0$, choose w ;
while *not converge* **do**
  for k *in* $i \in 1\ldots N_{EPOCH}$ **do**
    Predict bag labels with the current model
    $\hat{P} := \{h(f_{\theta_k}(x_{ij}), .., f_{\theta_k}(x_{im}))\}_{i\in 1\ldots N}$;
    Train a probability calibration model $\mathcal{C}_k$ with
    $\{y_i, \text{logit}(\hat{p}_i)\}_{i\in 1\ldots N}$ as input ;
    $\theta_t \leftarrow \theta_k$;
    for t *in* $1\ldots N_{BATCH}$ **do**
      $\hat{p}_i := \hat{P}[i]$, $\hat{y}_{ij} := f_{\theta_t}(x_{ij})$, $y_i := h(\{y_{ij}\}_{j\in\ldots m})$;
      $\mathcal{L}(\theta_t) = -\frac{1}{N_{\text{Pos}}}\sum_{i=0}^{N_{\text{Pos}}}\left(\mathcal{C}_k(\hat{p}_i)\cdot w\cdot\log(\hat{y}_i)\right) -$
      $\frac{1}{N_{\text{Neg}}}\sum_{i=0}^{N_{\text{Neg}}}\mathbb{E}_{j\sim P_i(X_i)}\log(1-\hat{y}_{ij}))$;
      $\theta_t \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t)$
    end
    $\theta_k \leftarrow \theta_t$;
  end
  return θ
end

Predictions on MA data in the test sets were performed as follows: for each bag of alleles and peptide $x_i = (A_i, s_i)$, predictions $\hat{y}_{ij}$ are made with each candidate allele separately. The label of the bag is predicted as

$$\hat{y}_i = \max\{\hat{y}_{ij}\}_{j\in 1\ldots m}, \tag{4}$$

where m is the number of potential alleles which may have presented the peptide. $\hat{y}_{ij} = f_\theta(a_{ij}, s_i)$ is the predicted score for peptide s_i and allele a_{ij} using model f_θ.

Meanwhile, the allele label for the bag is assigned as

$$a_{A_i} = \underset{j}{\text{argmax}}\,\{\hat{y}_{ij}\}_{j\in 1\ldots m}. \tag{5}$$

All of the following evaluation metrics are reported with the bag label y_i, the predicted bag label $\hat{y}_i$ and the assigned allele a_{A_i}. When evaluating the performance per allele, we exclude the alleles with <50 either positive or negative samples.

2.9 Evaluation metrics

Several evaluation metrics were used to compare models. For the binding affinity prediction task, Pearson correlation (R) and area under the receiver operating characteristic curve (auROC) were used. We used a cutoff of 500 nM to determine binding and non-binding. For the mass spectrometry presentation prediction task, auROC and AP were used for evaluation. AP is the weighted mean of precisions achieved at each threshold. It computes the area under the precision–recall curve without linear interpolation as with the trapezoidal rule. We used the python package *scikit-learn* to compute these metrics (Pedregosa *et al.*, 2011).

3 Results

3.1 Self-supervised pretraining improves the prediction of MHC–peptide interaction

Self-supervised pretraining has been shown to boost model performance for natural language processing and computer vision tasks (Chen *et al.*, 2020; Devlin *et al.*, 2019). Recent research has also shown the potential benefits of self-supervised pretraining on protein related tasks (Nambiar *et al.*, 2020; Rao *et al.*, 2019), such as contact prediction. However, the application of self-supervised pretraining on MHC–peptide related tasks has been less explored.

We first asked whether self-supervised pretraining helps with our prediction tasks. We investigated this in depth with the MHC–peptide binding affinity prediction task since all of the binding affinity values come from actual experiments, compared to presentation which entails sampling negatives. For this comparison, we binarized affinity values using the widely used threshold of 500 nM. We compared four strategies of training the same model architecture with the same hyperparameters:

- *Random*. Randomly initialized model trained end-to-end
- *Pretrain*. Model initialized with pretrained parameters from TAPE and trained end-to-end
- *Feature*. Model initialized with pretrained parameters from TAPE, but only the MLP classifier was trained. The BERT component was frozen during training and can be thought of as a kind of feature extractor
- *Random Feature*. Randomly initialized model, but only the MLP classifier was trained

When comparing *Pretrain* with *Random*, the pretrained model not only achieved a better performance (auROC 0.872 versus 0.853), but also converged much faster (Fig. 2). Indeed, the pretrained model performed similarly as the best randomly initialized model after only a single training epoch (auROC 0.851 versus 0.853). The pretrained model reached a good performance

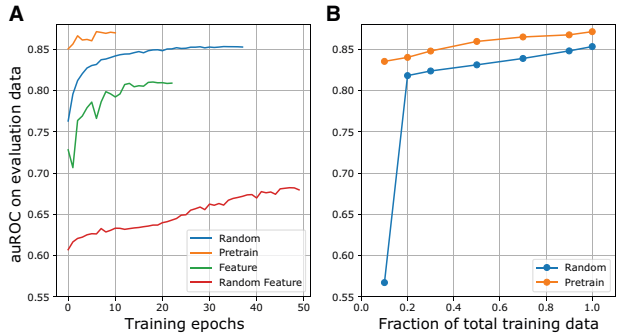

Fig. 2. Comparing model performance with and without pretraining. (**A**) Evaluation performance (auROC) at each training epoch for four different training strategies: *Random* (blue), *Pretrain* (yellow), *Feature* (green) and *Random Feature* (red). (**B**) Evaluation performance (auROC) at each training data subsampling fraction. Fraction of subsampled training data (*x*-axis) versus auROC on the evaluation set (*y*-axis) for *Random* (blue) and *Pretrain* (yellow). The reported performances are the best validation performances when the models of different fractions converge

(auROC > 0.85) even at the first epoch whereas the randomly initialized model had poor performance at the start of the training process.

Based on these results, we use *Pretrain* for the remaining analysis.

3.2 Improved MHC class II binding prediction with transformers

We trained our MHC class II binding model with the dataset curated by Jensen *et al.* We used the same cross-validation folds so that peptides appearing in both training and testing folds are minimized. We compare our method (BERTMHC) against the sate-of-the-art class II peptide binding model NetMHCIIpan3.2 (Jensen *et al.*, 2018), which was trained with the same cross-validation splits. Our model outperformed NetMHCIIpan3.2 in terms of auROC for most of the alleles (48 out of 61, Fig. 3). We also compared our model with PUFFIN (Zeng and Gifford, 2019b), which is a convolutional neural network model trained on the same dataset. BERTMHC ($\text{auROC} = 0.8822, R = 0.759$) outperformed both PUFFIN-mean ($\text{auROC} = 0.8774, R = 0.754$) and PUFFIN-BL ($\text{auROC} = 0.8795$) (Zeng and Gifford, 2019b).

We also used our independent MHC class II benchmark binding set, described in Section 2, to further benchmark the models. When evaluated on this independent data, our model ($\text{auROC} = 0.72, R = 0.39$) performed better than NetMHCIIpan3.2 ($\text{auROC} = 0.68, R = 0.30$), PUFFIN ($\text{auROC} = 0.69, R = 0.37$) and MHCnuggets ($\text{auROC} = 0.58, R = 0.15$) in terms of classifying peptides into binders and nonbinders.

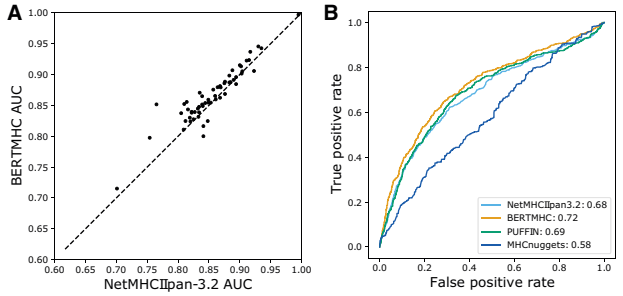

Fig. 3. Benchmarking BERTMHC on binding affinity prediction. (**A**) Comparing BERTMHC with NetMHCIIpan3.2 under cross-validation. Each dot represents one MHC class II allele. The area under the receiver operating characteristic curve (auROC) for the BERTMHC model (*y*-axis) are compared with the NetMHCIIpan3.2 model (*x*-axis). (**B**) Comparing BERTMHC (yellow) against NetMHCpan3.2 (cyan), PUFFIN (green) and MHCnuggets (blue) on independent binding affinity data from IEDB and the Dana–Farber repository. The number in the legend gives the auROC of the respective methods

Since our model takes both the MHC and peptide sequence as input, it can generalize to unseen MHC alleles. To test this generalizability, we evaluated our model under a leave-one-molecule-out (LOMO) setting. We performed the LOMO experiments with data from Jensen *et al.* (2018) under the same fivefold cross-validation split. To test the LOMO performance of an allele, a model was trained on the training set with data from the allele removed and evaluated on the evaluation set with only the test allele kept. Out-of-fold predictions from all fivefolds are combined to compute a final LOMO performance for each allele. We performed LOMO experiments for all 61 alleles in Jensen *et al.* (2018) and compare our performance against NetMHCIIpan3.2 with their self-reported auROC values. BERTMHC outperforms NetMHCIIpan3.2 for 35 out of 61 evaluated alleles (Fig. 4). Our mean LOMO auROC across alleles is 0.784, outperforming 0.775 for NetMHCIIpan3.2. The complete data including Pearson correlation per allele for BERTMHC can be found in Supplementary Table S4.

3.3 Improved MHC class II presentation prediction with transformers

We then asked whether the same approach can be applied to train a better model to predict peptide presentation. Mass spectrometry is typically used to detect presentation of peptides on the cell surface. We trained our model on all the curated mass spectrometry elution data from Reynisson *et al.* (2020b), using the same cross-validation splits. We also trained a model with SA data only (BERTMHC-SA) to compare with the version trained on the complete SA and MA dataset with MIL strategy (referred to as BERTMHC). We compared our models with the latest NetMHCIIpan4.0 model which was trained on the same dataset. Since the out-of-fold predictions of NetMHCIIpan4.0 were not provided, we compared our out-of-fold predictions against the prediction of the public stand-alone version of NetMHCIIpan4.0, which is an ensemble of models trained on all cross-validation folds. Note that this comparison is in favor of the competing model since its training used our evaluation data. We used *Score_EL* of NetMHCIIpan4.0 without the context encoding throughout this work.

For the MA data, we take the maximum prediction among the alleles in each bag of the prediction for that bag (see Section 2). Evaluated on both SA and MA data, NetMHCIIpan4.0 performed only slightly better in terms of both auROC and AP, even though the model was trained with the evaluation data while we performed out-of-fold prediction (Table 1). Both models outperform NetMHCIIpan3.2; this is not surprising, since NetMHCIIpan3.2 is only trained on binding data (Table 1).

We next evaluated the models on SA data only, where the labels are unambiguous. We observe that all models performed better in this setting compared with including the MA data in the evaluation (Table 2). BERTMHC is better or on par with NetMHCIIpan4.0 in terms of both auROC and AP but consistently better than NetMHCIIpan3.2 (Table 2). The version only trained on SA data (BERTMHC-SA) does not perform as well as BERTMHC which was trained on the complete dataset (Table 2).

To further investigate the potential pros and cons of our model compared with others, we investigated the results for each allele with the SA data. We outperform NetMHCIIpan4.0 for most alleles in terms of auROC (12 out of 19) while NetMHCIIpan4.0 is better in terms of AP (13 out of 19) (Fig. 5). Training BERTMHC including MA data with MIL again further improved the performance for most alleles both in terms of auROC and AP (Supplementary Fig. S2). We also show with an LOMO experiment on the SA data for 19 alleles that our presentation model generalize well to unseen alleles (Supplementary Table S5).

Overall we show that our model is able to perform well both when the allele labels are unambiguous and also when deconvolution is needed. Furthermore, our MIL strategy not only expands the model training to MA data but also effectively improves the performance on the SA data with a larger training set. We found that BERTMHC under out-of-fold prediction to have comparable

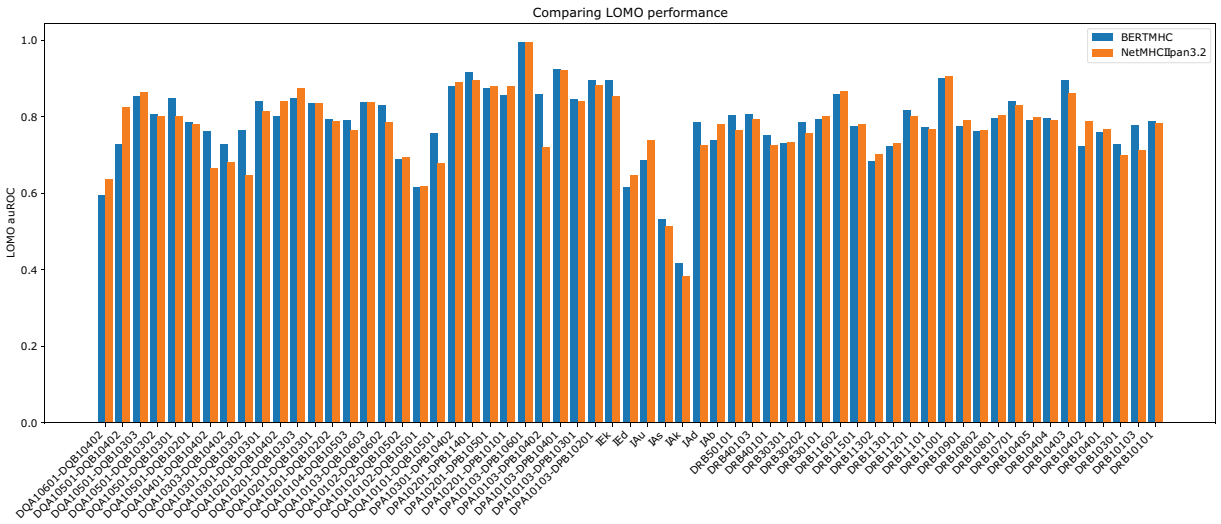

Fig. 4. Comparing with NetMHCIIpan3.2 on LOMO experiment. Barplot of auROCs for 61 MHC class II alleles from Jensen *et al.* (2018) data for BERTMHC (blue) and NetMHCIIpan3.2 (orange). The auROCs for NetMHCIIpan3.2 are self-reported by the authors. We provide LOMO Pearson correlations in Supplementary Table S4

Table 1. Presentation performance on both MA and SA data from Reynisson *et al.* (2020b), with all alleles combined

Method	auROC	AP
BERTMHC	0.954 ± 0.0004	0.807 ± 0.001
BERTMHC-SA	0.902 ± 0.0006	0.625 ± 0.002
NetMHCIIpan4.0	0.956 ± 0.0004	0.810 ± 0.001
NetMHCIIpan3.2	0.704 ± 0.001	0.257 ± 0.002

Note: The performances for BERTMHC (trained on MA and SA data) and BERTMHC-SA (trained on SA data) are reported by out-of-fold prediction under cross-validation while NetMHCIIpan4.0 was trained on all the data. ± indicates the range of 95% confidence interval estimated by bootstrapping.

Table 2. Presentation performance on SA data only from Reynisson *et al.* (2020b), with all alleles combined

Method	auROC	AP
BERTMHC	0.965± 0.0008	0.863± 0.002
BERTMHC-SA	0.960 ± 0.0009	0.847 ± 0.003
NetMHCIIpan4.0	0.961 ± 0.002	0.865± 0.004
NetMHCIIpan3.2	0.796 ± 0.001	0.385 ± 0.0025

Note: The performances for BERTMHC (trained on MA and SA data) and BERTMHC-SA (trained on SA data) are reported by out-of-fold prediction under cross-validation while NetMHCIIpan4.0 was trained on all the data. ± indicates the range of 95% confidence interval estimated by bootstrapping.

performance with NetMHCIIpan4.0 which was trained on all the data.

We next compared the methods on an independent mass spectrometry presentation dataset with no overlapping data from the training and evaluation set (see Section 2). Our model outperforms NetMHCIIpan4.0 in terms of both auROC (0.89 versus 0.83) and AP (0.60 versus 0.53) (Fig. 6). BERTMHC-SA performed better than NetMHCIIpan4.0 but worse than BERTMHC on this independent SA data (auROC = 0.87, AP = 0.57), further suggesting that our MIL strategy on MA data can further improve the performance on SA data.

To further evaluate our model under the multiple instance setting, which is often the case when applying our model to real patient data, we generated mass spectrometry data from peptides eluted from six patient samples (Section 2). In total, 15 277unique peptides

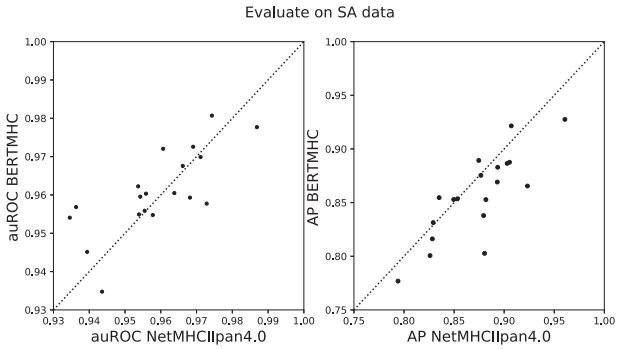

Fig. 5. Comparing BERTMHC with NetMHCIIpan4.0 under cross-validation on SA data from Reynisson *et al.* (2020b) only. Each dot represents one MHC class II allele. The auROC (left) and AP (right) for the BERTMHC model (*y*-axis) are compared with the NetMHCIIpan4.0 model (*x*-axis). BERTMHC was trained with both SA and MA data. Predictions of BERTMHC were made out-of-fold under cross-validation while NetMHCIIpan4.0 was trained on all data

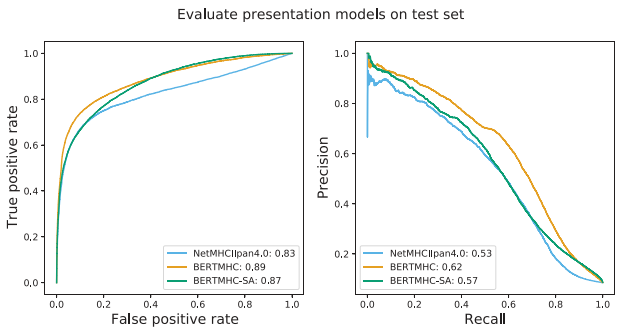

Fig. 6. Comparing BERTMHC, BERTMHC-SA and NetMHCIIpan4.0 on independent mass spectrometry data from IEDB. (**left**) ROC curve for BERTMHC (yellow), BERTMHC-SA (green) and NetMHCIIpan4.0 (blue). (**right**) Precision–recall curve for BERTMHC (yellow), BERTMHC-SA (green) and NetMHCIIpan4.0 (blue). The number in the legend gives the area under the curve for that method

presented by HLA-DR molecules were eluted from six patients, with a median of 3 964 peptides per patient. Each patient had two HLA-DR alleles determined. We compared BERTMHC against NetMHCIIpan4.0 on this MA data. Negative peptides were randomly sampled from the human proteome by matching the length distribution of positive peptides. We sampled 10 negative peptides

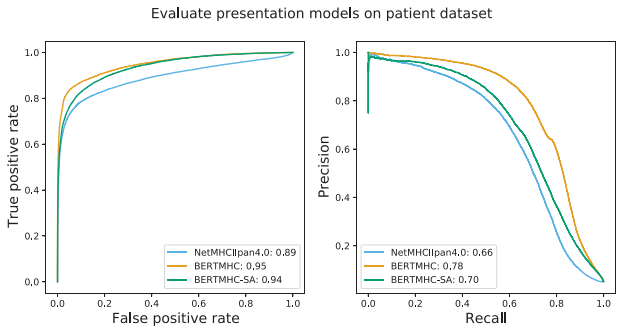

Fig. 7. Comparing BERTMHC, BERTMHC-SA and NetMHCIIpan4.0 on mass spectrometry eluted peptides from six patient tumor samples. (**left**) ROC curve for BERTMHC (yellow), BERTMHC-SA (green) and NetMHCIIpan4.0 (blue). (**right**) Precision–recall curve for BERTMHC (yellow), BERTMHC-SA (green) and NetMHCIIpan4.0 (blue). The number in the legend gives the area under the curve for that method

for each positive peptide. Peptide scores were predicted as described in Equation (4) for both models. When evaluated on all patients combined, BERTMHC outperforms NetMHCIIpan4.0 both in terms of auROC (0.95 versus 0.89) and AP (0.78 versus 0.66) (Fig. 7). When evaluated per patient, BERTMHC again outperforms NetMHCIIpan4.0 in all patients with maximum auROC improvement by 14.8% (Supplementary Fig. S3) and maximum AP improvement by 61.0% (Supplementary Fig. S4). BERTMHC-SA performs consistently better than NetMHCIIpan4.0 but worse than BERTMHC both overall (Fig. 7) and per patient (Supplementary Figs S3 and S4), again highlighting the importance of training including the MA data with MIL.

We note that the HLA-DR antibody used in the mass spectrometry experiment for patients binds to all HLA-DR alleles. We evaluated all model predictions based only on the HLA-DRB1 alleles of the patients (Supplementary Table S3), though some of the peptides in the dataset may have instead been bound and presented by other HLA-DR molecules, such as HLA-DRB3.

4 Discussion

Predicting MHC–peptide binding has been a long-standing problem, and many other approaches have also been developed. For example, embedding the input sequences for MHC-related tasks with self-supervised learning has also been explored in other directions. DeepLigand (Zeng and Gifford, 2019a) trained a language model from positive mass spectrometry peptides using the ELMo architecture (Peters *et al.*, 2018). The peptide embedding was shown to provide additional predictive information other than what binding assays provide. Similarly, using embedding layers pretrained via language modeling on peptides was shown to benefit MHC–peptide binding prediction (Phloyphisut *et al.*, 2019; Vielhaben *et al.*, 2020). Here, we used a pretrained network with a more effective architecture and trained from a large corpus of protein sequences instead of only positive peptides. Moreover, our pretrained embedding model was applied to a concatenated sequence from both MHC and peptides, so that the model can potentially capture long distance interactions between MHC and peptide sequences.

Models with attention mechanisms have also been applied to MHC–peptide interaction prediction tasks. ACME (Hu *et al.*, 2019) combines convolution and attention for MHC class I, MHCAttnNet (Venkatesh *et al.*, 2020) uses attention on top of a bidirectional LSTM (Venkatesh *et al.*, 2020) for both class I and class II. These models used one-layer attention, for which the attention weights are more interpretable. Here, we used multiple layers of self-attention, which is less interpretable but performed better in these tasks.

Despite the extensive existing work, previous models still have limited performance on MHC class II molecules. In this work, we have demonstrated that self-supervised pretraining of a transformer model leads to state-of-the-art performance for MHC class II binding and presentation prediction. We further provided a novel MIL strategy to address limitations of typical mass spectrometry assays for assessing eluted peptides. We conducted a thorough set of empirical experiments to compare the performance of the models for both binding and presentation tasks. For presentation, we consider both the SA setting, in which the exact MHC molecule to which a peptide is bound is known, as well as the multiple allele setting, in which only a set of possible MHCs to which a peptide was bound are known. In both cases, we show that our approach leads to state-of-the-art performance.

Our approach to predicting peptide presentation is solely based on the peptide and MHC sequences. Other studies have shown that the original context of the peptides and the peptide expression level are both important features for peptide presentation prediction (Chen *et al.*, 2019; Reynisson *et al.*, 2020b). Models using this additional information might be able to outperform our model on the presentation task. Moreover, T-cell epitope prediction is important in real applications. Future work could integrate BERTMHC with other relevant features such as the gene expression to have more precise epitope prediction.

The proposed approach is applicable to other sequence-based predictions as well. We anticipate this approach to be useful to the community not only as a strategy for training binding and presentation models, but also as an approach to train protein sequence-based models for other challenges in immunology, such predicting T-cell response. Considering that immune response assays are typically costly, a modeling strategy that improves data efficiency by performing self-supervised learning is valuable. The self-supervised pretraining step was not specific to downstream tasks, and was trained from generic protein sequences that may have very distinct biochemical properties compared to the sequences of our task. Nevertheless, the pretraining step was very beneficial. Preliminary work has been done to interpret the models trained from protein sequences with self-supervised learning (Heinzinger *et al.*, 2019; Vig *et al.*, 2020). This is a promising direction for future research in order to better understand what the models have learned and how that can guide better treatment decisions.

Acknowledgement

We thank Carolin Lawrence, Timo Sztyler and Martin Renqiang Min for the discussions.

Financial Support: none declared.

Conflict of Interest: J.C. and B.M. are employed by NEC Laboratories Europe. K.B. and K.R. are employed by Transgene SA.

Data availability

The independent binding data described in Section 2.2 are available in Supplementary Table S1. The independent mass spectrometry data described in Section 2.4 are available in Supplementary Table S2. The patient mass spectrometry data described in Section 2.5 cannot be shared for ethical/privacy reasons.

References

Al-Daccak,R. *et al.* (2004) MHC class II signaling in antigen-presenting cells. *Curr. Opin. Immunol.*, **16**, 108–113.

Alvarez,B. *et al.* (2019) NNAlign_MA: MHC peptidome deconvolution for accurate MHC binding motif characterization and improved T-cell epitope predictions. *Mol. Cell. Proteomics*, **18**, 2459–2477.

Barlow,R.E. *et al.* (1972) *Statistical Inference Under Order Restrictions: Theory and Application of Isotonic Regression*. John Wiley & Sons Ltd.

Bassani-Sternberg,M. and Gfeller,D. (2016) Unsupervised HLA peptidome deconvolution improves ligand prediction accuracy and predicts cooperative effects in peptide–HLA interactions. *J. Immunol.*, **197**, 2492–2499.

Chen,B. *et al.* (2019) Predicting HLA class II antigen presentation through integrated deep learning. *Nat. Biotechnol.*, **37**, 1332–1343.

Chen,T. *et al.* (2020) A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning,* PMLR 119:1597-1607, 2020.

Cox,J. and Mann,M. (2008) MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.*, **26**, 1367–1372.

Devlin,J. *et al.* (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

El-Gebali,S. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.

Heinzinger,M. *et al.* (2019) Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinform.*, **20**, 1–17.

Hu,Y. *et al.* (2019) ACME: pan-specific peptide–MHC class I binding prediction through attention-based deep neural networks. *Bioinformatics*, **35**, 4946–4954.

Ilse,M. *et al.* (2018) Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning*.

Janeway,C.A. Jr. *et al.* (2001) *Immunobiology: The Immune System in Health and Disease*. Garland Science.

Jensen,K.K. *et al.* (2018) Improved methods for predicting peptide binding affinity to MHC class II molecules. *Immunology*, **154**, 394–406.

Lang,P. *et al.* (2001) TCR-induced transmembrane signaling by peptide/MHC class II via associated Ig-α/β dimers. *Science*, **291**, 1537–1540.

Nambiar,A. *et al.* (2020) Transforming the language of life: transformer neural networks for protein prediction tasks. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–8.

Neefjes,J. *et al.* (2011) Towards a systems understanding of MHC class I and MHC class II antigen presentation. *Nature Reviews. Immunology*, **11**, 823–836. *10.1038*/nri3084 22076556

Niculescu-Mizil,A. and Caruana,R. (2005) Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 625–632.

O'Donnell,T.J. *et al.* (2020) MHCflurry 2.0: improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. *Cell Syst.*, **11**, 42–48.

Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Peters,B. *et al.* (2020) T cell epitope predictions. *Annu. Rev. Immunol.*, **38**, 123–145.

Peters,M.E. *et al.* (2018) Deep contextualized word representations. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Phloyphisut,P. *et al.* (2019) MHCSeqNet: a deep neural network model for universal MHC binding prediction. *BMC Bioinform.*, **20**, 1–10.

Purcell,A.W. *et al.* (2019) Mass spectrometry-based identification of MHC-bound peptides for immunopeptidomics. *Nat. Protoc.*, **14**, 1687–1707.

Rao,R. *et al.* (2019) Evaluating protein transfer learning with TAPE. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*.

Reynisson,B. *et al.* (2020a) Improved prediction of MHC II antigen presentation through integration and motif deconvolution of mass spectrometry MHC eluted ligand data. *J. Proteome Res.*, **19**, 2304–2315.

Reynisson,B. *et al.* (2020b) 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.*, **48**, W449–W454.

Rock,K.L. *et al.* (2016) Present yourself! by MHC class I and MHC class II molecules. *Trends Immunol.*, **37**, 724–737.

Sidney,J. *et al.* (2013) Measurement of MHC/peptide interactions by gel filtration or monoclonal antibody capture. *Curr. Protoc. Immunol.*, **100**, 18.3.1–18.3.36.

Tanyi,J.L. *et al.* (2018) Personalized cancer vaccine effectively mobilizes antitumor T cell immunity in ovarian cancer. *Sci. Transl. Med.*, **10**, eaao5931.

Vaswani,A. *et al.* (2017) Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

Venkatesh,G. *et al.* (2020) MHCAttnNet: predicting MHC–peptide bindings for MHC alleles classes I and II using an attention-based deep neural model. *Bioinformatics*, **36**, i399–i406.

Vielhaben,J. *et al.* (2020) USMPep: universal sequence models for major histocompatibility complex binding affinity prediction. *BMC Bioinform.*, **21**, 1–16.

Vig,J. *et al.* (2020) BERTology meets biology: interpreting attention in protein language models. arXiv:2006.15222 [cs.CL].

Vita,R. *et al.* (2019) The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Res.*, **47**, D339–D343.

Zeng,H. and Gifford,D.K. (2019a) DeepLigand: accurate prediction of MHC class I ligands using peptide embedding. *Bioinformatics*, **35**, i278–i283.

Zeng,H. and Gifford,D.K. (2019b) Quantification of uncertainty in peptide–MHC binding prediction improves high-affinity peptide selection for therapeutic design. *Cell Syst.*, **9**, 159–166.

Zhang,G.L. *et al.* (2011) Dana-Farber repository for machine learning in immunology. *J. Immunol. Methods*, **374**, 18–25.