Exploration of Neural Networks for Peptide-MHC Class I Binding Prediction

MAALOUF Andrew

University of Bern, MSc in Bioinformatics and Computational Biology, 23-117-583

KEYWORDS

MHC class I
Peptides
Neoantigens
Artificial Neural Networks
Hyperparameters
Recurrent Neural Network
Long Short-Term Memory
Bidirectional
Immunotherapy
Eluted Ligand
Binding Affinity
Performance Metrics
Optimizer

ABSTRACT

Peptide-MHC binding prediction is a rapidly advancing field in immunoinformatics, driving innovations in vaccine design and personalized immunotherapies. Many existing tools predict binding affinity and eluted ligand interactions using artificial neural networks, often relying on fixed-length input transformations through insertions or deletions. This project investigates the potential of bidirectional Long Short-Term Memory (biLSTM) networks to generalize across variable-length peptides without such transformations, focusing on the impact of different hyperparameters and encoding techniques.

Two models were developed: an allele-specific model to predict binding affinity and a pan-specific model for eluted ligand interactions. Performance was evaluated using metrics such as AUC, Pearson correlation, and sensitivity. While the models demonstrated promise, limitations in dataset quality and computational resources hindered further optimization and training. The report also shares insights into the effect of hyperparameters on model performance and discusses techniques to enhance the current approach.

Future directions include exploring convolutional neural networks, attention mechanisms, and dimensionality reduction techniques to address challenges related to input size and model complexity. Integrating sequence- and structure-based approaches could further refine predictions. This research highlights the potential of biLSTM architectures for pan-specific HLA-peptide binding prediction while emphasizing the importance of robust datasets, advanced methodologies, and sufficient computational resources for achieving reliable and generalizable results.

INTRODUCTION

Cancer treatments such as surgery, chemotherapy, and radiation therapy have unique benefits and limitations. Surgery often fails to entirely eliminate tumor cells, and its post-operative infections can increase the risk of cancer recurrence due to the release of proinflammatory mediators[1,2]. Moreover, radiation and chemotherapy may lead to acquired resistance through mechanisms like multidrug resistance and inhibition of apoptosis[3,4]. On the other hand, immunotherapy, leveraging the immune system to target cancer cells, has shown auspicious outcomes. The genetic instability in tumor cells results in non-synonymous somatic mutations and thus the production of abnormal proteins. These proteins are broken down by the proteasome and transported into the endoplasmic reticulum lumen where they bind to major histocompatibility complex class I (MHC-I) molecules, within the peptide loading complex[5]. These tumor-specific mutated peptides, known as neoantigens, are displayed on the surface of malignant cells through the MHC-I protein. If T cells recognize and bind to a peptide-MHC complex, an immune response is triggered, and the compromised cell undergoes lysis. Therefore, understanding the rules of this event is key in human health applications.

MHC class I molecules are highly selective and only bind a small fraction of presented peptides strongly enough to trigger an immune response, highlighting the importance of accurately predicting this peptide-MHC-I interaction[6]. To address this, some tools are allele-specific, trained for individual MHC molecules, while others are pan-specific, trained on data spanning multiple MHC molecules. Given the thousands of allelic variants identified for MHC, pan-specific methods, such as NetMHCpan, have been crucial in overcoming the polymorphism challenge[7,8]. Several computational methods have been developed to estimate peptide-HLA-I binding affinities based on MHC structure and empirical force fields[9-12]; however, these methods are constrained by the limited number of HLA-I molecules with solved structural data. In addition to MHC-I molecules showing significant allele-specific diversity in amino acid preferences, they also exhibit different preferences in peptide lengths. While most alleles favor 9-mer peptides, others prefer shorter or longer peptides. For instance, H-2-Kb favors 8-mers while HLA-A*01:01 frequently binds peptides longer than 9 amino acids[13,14]. Methods for predicting MHC-peptide binding rely heavily on data biased towards 9-mers, which make up over 72% of the training data in the IEDB, hindering thus the accuracy for other peptide lengths[15]. Neural network-based methods, including NNAlign, initially extrapolated from 9-mer data for other lengths while ignoring non-9-mer peptide data[16,17]. Consequently, a novel alignment step was used which allows the integration of peptide length variation into prediction models, capturing thus length preferences and improving accuracy[18-20]. This model, having a higher performance than extrapolations model especially for molecules with few measured experimental data of a given peptide length, introduce insertions and deletions which limit thus different lengths to a binding core of a common 9-mer size[18].

Advanced bioinformatics approaches like artificial neural networks (ANNs) have shown their ability to recognize non-linear patterns which are crucial to peptide-MHC-1 interactions[21,22]. In an ANN, information is trained and passed through a network with input layer, hidden layers, and output layer all connected in a specific architecture through weighted connections[23]. They are trained using an input such as peptide sequences associated with an output like binding affinity. Once the training is done and the weights are calculated, the network should be able to recognize complicated input patterns which allows the correct prediction of the output, as shown in predicting peptide-HLA-I interactions[24].

Different types of experimental data have been used to train these methods: first, some predictors are trained on binding affinity (BA) data which models the single event of peptide-MHC binding only. For that, IC_{50} concentrations in nanomolar are mapped onto a regression target between 0 and 1 using the formula $1 - \frac{\log IC_{50}}{\log 50000}$. Second, some predictors are trained with data received from mass spectrometry (MS) experiments, known as eluted ligands (EL)[25]. Third, some methods are trained integrating both BA and EL data, thus boosting predictive power[26-28]. The latest updated version of NetMHCpan differ from the previous versions in two aspects: the training data and the machine-learning modeling framework. The training data have been vastly extended by accumulating BA and EL data from the public domain. In particular, EL data were extended to include multi-allelic (MA) data. The combined dataset used for training, which forms the starting dataset for this research, consists of 13,245,212 data points covering 250 distinct MHC class I molecules. The machine learning NNAlign MA framework allows the integration of mixed data types (including EL MA) in the model training[29].

In order to prepare the input for the model, several encoding methods can be used to represent categorical data numerically: first, one-hot encoding where each amino acid is assigned a unique position in a binary vector of length 20 (corresponding to the 20 standard amino acids). For example, if a peptide contains the amino acid alanine (A), its one-hot representation would be [1, 0, 0, ..., 0], with the 1 in the position corresponding to alanine. Second, BLOSUM (BLOcks Substitution Matrix) encoding is another option where each amino acid in a peptide sequence is represented by a vector of substitution scores against all 20 standard amino acids[30]. These matrices are derived from empirical substitution probabilities of amino acids in conserved regions of protein families. BLOSUM62 and BLOSUM50 are among the most used variants, optimized for identifying sequence similarities at different evolutionary distances.

The NetMHCpan 4.1 tool incorporates a sophisticated encoding structure to predict peptide-MHC binding with high accuracy[28]. The neural network inputs included the peptide and the MHC molecule as a pseudo-sequence. Peptides were represented as 9-mer binding cores using insertions and deletions, as discussed earlier, and encoded with BLOSUM. Additional features included the length of insertions/deletions, peptide-flanking regions (non-zero if the peptide extended beyond the binding groove), and peptide length (encoded using four input neurons for lengths ≤8, 9, 10, and ≥11).

Furthermore, to process these peptides, NetMHCpan 4.1 uses a deep neural network (DNN) with 1 hidden layer as described in [31] to capture complex relationships between peptide sequences and MHC molecules, after representing all lengths of peptides as 9-mers.

In contrast, one potential alternative approach could be using Long Short-Term Memory (LSTM) networks, specifically a bidirectional LSTM[32,33]. LSTM networks are a type of recurrent neural network (RNN) that excels at processing sequential data by capturing long-term dependencies in sequences. The bidirectional aspect of an LSTM network is advantageous because it processes sequences in both forward and backward directions, allowing the model to capture both the preceding and succeeding context of each amino acid.

The key advantage of LSTM over a traditional DNN lies in its ability to model sequential dependencies, which is particularly relevant if the peptide sequence plays a significant role in binding, and subtle relationships between distant amino acids may influence the overall binding affinity. Moreover, while NetMHCpan 4.1 focuses on peptides of fixed length (9-mers), a bidirectional LSTM could theoretically process peptides of different lengths, without introducing insertions or deletions, making it more flexible and capable of incorporating a wider variety of peptide sequences.

While the DNN architecture used in NetMHCpan 4.1 and other tools has been highly effective for the task of peptide-MHC binding prediction, the use of a bidirectional LSTM offers a compelling alternative, an option which will be explored in this research.

METHODS

Dataset

2 different MHC peptide class I datasets, each containing both BA and EL data for numerous alleles, were downloaded to train the models in this research. Initially, all datasets were filtered to include only peptides of length 8–15 amino acids. However, for the final models, the data was length filtered to include only peptides of length 8-12.

All BA samples are labeled with IC50 binding affinity values in nanomolar, providing an essential metric for evaluating the strength of interaction between peptides and MHC molecules. These IC50 values were rescaled to the interval [0,1] as mentioned in the introduction. This normalization ensures that lower IC50 values, which correspond to stronger binding affinities, are assigned higher scores closer to 1, while weaker affinities approach 0. As for EL data, they are assigned binary target values: a value of 1 for binders and 0 for non-binders.

At first, DNNs were trained on the dataset BD2013, downloaded from the widely used IEDB3 database (http://tools.iedb.org/main/datasets). Given that the allele HLA-A*02:01 had the most data points (12160 assays), it was extracted for model training and hyperparameters exploration. Duplicate peptides existed, so they were removed from both the training and testing data. For DNN training only, a threshold of 0.426 was used to divide the peptides into binders and non-binders for binary output training.

To increase the dataset, the training data from NetMHCpan-4.1, which includes 13,245,212 data points representing 250 distinct MHC class I molecules was downloaded. Only MHC alleles beginning with "HLA" were retained, focusing specifically on human alleles. Although the dataset included both single allelic (SA) and multi-allelic (MA) data - valuable for their training strategy - duplicates were not removed in this case since NNs are known for being robust and tolerant to imperfect or redundant data, which can sometimes enhance their learning capacity[34].

For the final EL model, the pan method was employed, resulting in the selection of 51 human MHC alleles from the 57 initially available after excluding alleles that lacked an associated HLA peptide sequence or were represented by fewer than 50 sequences. In contrast, the final BA models did not adhere to the pan method due to performance considerations discussed later in this research. Instead, these BA models were specifically tailored to the alleles of interest, provided these alleles had data points within the dataset.

Dataset Imbalance

The distribution of BA and EL for the peptides in the dataset is extremely nonuniform which is why the network training should be done in a balanced manner. To address imbalances in the dataset when training the final models, different strategies were applied for EL and BA data. For the EL data, where the original dataset included 3,481,858 non-binders and only 197,547 binders, upsampling and downsampling were performed to achieve a balanced dataset. Specifically, the binders were upsampled to 1 million, and the non-binders were downsampled to 1,925,146. For the BA data, preprocessing was conducted for each allele separately before training. BA values were divided into bins, namely ['0-0.2', '0.2-0.4', '0.4-0.45', '0.45-0.65', '0.65-1'], and minority bins were upsampled to match the size of the largest bin. The '0.4-0.45' bin, however, was left unchanged due to its sensitivity around the 0.426 threshold, which marks the transition between binding and non-binding predictions. This way, data from each bin are presented to the neural network with equal frequency.

HLA Pseudo-Sequences

The MHC allele pseudo-sequences used in this study were those used for NetMHCpan-4.1 to leverage information between MHC molecules. These pseudo-sequences consist of 34 amino acids representing key polymorphic residues within 4.0 Å of the peptide-binding groove, as described in [17].

Encoding

2 distinct encoding techniques were explored to represent the input sequences for the NN: conventional one-hot encoding and BLOSUM encoding. Given that all BLOSUM encoding schemes, with clustering thresholds ranging from 30% to 70%, demonstrate comparable performance [35], BLOSUM50 encoding was selected for this study.

Hyperparameters Choice

Hyperparameters were not determined through an exhaustive grid search. Instead, various hyperparameters, including learning rate, batch size, number of layers, number of neurons per layer, and activation functions, were adjusted iteratively. Each parameter was modified individually to observe its direct impact on performance, and subsequent changes were informed by the outcomes of the previous adjustments.

Regularization Techniques

To mitigate overfitting, regularization techniques such as L2 regularization and dropout were explored. These methods were applied to prevent the model from relying excessively on specific features, thereby enhancing its generalization ability.

Optimization Techniques

Different optimization techniques were tested, including Stochastic Gradient Descent (SGD) and Adam. While both methods were evaluated, Adam was predominantly favored due to its faster convergence and the computational resource constraints inherent to this project.

Network Architecture Explored Ideas

Initially, when training DNNs for the prediction of binary output whether the peptide is a binder or not, the model architecture and training code were implemented manually, drawing insights from Andrew Ng's Coursera course on deep learning. Two architectures were explored: one with three hidden layers and another with five hidden layers (including the output layer). Various hyperparameters were tuned during this phase to optimize performance.

Later, for RNNs, TensorFlow was used to facilitate more complex architectures. Among the models tested, a bidirectional Long Short-Term Memory (biLSTM) network demonstrated high accuracy on the validation set. This model included a masking layer to handle variable input lengths, two bidirectional LSTM layers, a dense layer, and a final output layer. Given its strong performance, this architecture was selected for further exploration.

When attempting to replicate the NNAlign idea of using EL data to enhance BA predictions, several experimental setups were explored. These included incorporating techniques like embedding layers and transfer learning to improve model performance and generalization. Furthermore, the biLSTM architecture was modified to incorporate two separate output layers. The weights between the input and hidden layers are shared between the two input types (BA/EL), while the weights from the hidden to the output layer are specific to each input type. During the training process, an example is randomly selected from either dataset and processed through forward and backward propagation. A training epoch is completed once every data point in the smaller dataset (BA) has been used for training.

Final Model Architecture for BA Prediction

The final model designed for predicting BA utilizes a BiLSTM architecture tailored for sequence regression tasks. The input consists of peptide sequences encoded in a 12×20 matrix format, representing 12 amino acids and 20 encoded features per residue. If the length of the peptide sequence is shorter than 12, the rest is padded with zeros. To accommodate variable input lengths, a Masking layer is used to effectively ignore padding during training.

The model begins with two Bidirectional LSTM layers, the first with 32 units and the second with 64 units, each configured with an L2 regularizer to reduce overfitting. The second LSTM layer outputs a sequence representation in a fixed dimension. A Dropout layer with a rate of 0.5 follows to further mitigate overfitting. A Dense layer with 16 units and ReLU activation acts as a feature extractor, and the final output layer uses a linear activation function to generate continuous BA predictions. The loss function is a custom logarithmic loss as suggested in [44], which ensures robustness by penalizing errors in prediction logarithmically, preventing extreme deviations. The model is compiled with the Adam optimizer and uses Root Mean Squared Error (RMSE) as a metric to assess performance.

Final Model Architecture for EL Prediction

The EL prediction model is designed to classify peptide sequences as binders or non-binders using a bidirectional LSTM architecture. The input shape is a 46×20 matrix (this time the HLA pseudo-sequence is included after the peptide), where each peptide is encoded similarly to the BA model. The architecture is identical to the final model architecture for BA prediction, however the final output layer employs a sigmoid activation function to produce a probability score for classification. The model is compiled with the Adam optimizer and a binary cross-entropy loss function. Performance is measured using accuracy, which evaluates the model's capability to correctly classify binders and non-binders.

EL Model Training

To make efficient use of computational resources given the huge dataset, the EL model was trained for 10 epochs only using a balanced 5-fold stratified cross-validation strategy. For each fold, the dataset was split into training and test sets, and a separate model instance was trained and evaluated. Each model was evaluated on its respective test set, and accuracy scores were recorded for all folds. The model with the highest accuracy was saved as the best-performing EL model.

BA Model Training

The training process leveraged a single model instance and used a validation split of 5% of the training data. Additionally, callbacks such as learning rate decay and early stopping were integrated during training to dynamically adjust the learning rate and halt training when performance on validation data plateaued, thereby preventing unnecessary computations and overfitting.

Performance Evaluation

For testing the models, only the alleles of interest were considered. Performance evaluation on BA models was conducted using BA data downloaded from [26] for the HLA-A*26:01 and HLA-A*32-01 alleles, and from the python library “epitopepredict” for the HLA-A*74:01 and HLA-B*48:01 alleles. Several metrics were used for BA models evaluation, mainly the Area Under the Curve (AUC) and Pearson correlation[36,37]. The AUC was used to assess the model's ability to distinguish between positive and negative classes for each MHC allele separately. An AUC of 0.5 indicates random model performance, while an AUC of 1 represents a perfect model. For this evaluation, a threshold of 0.426 was applied, based on the dataset's characteristics; binding thresholds can vary across different HLA molecules, but this fixed approach represents a common simplification[31,43]. Additionally, Pearson correlation was used to measure the strength and direction of the linear relationship between predicted and actual values. A higher Pearson correlation indicates better model performance in predicting continuous values, providing a deeper understanding of the alignment between the predicted and true outcomes. Also, the F1 score, balancing the trade-off between false positives and false negatives, as well as Kendall’s Tau were explored.

For the EL pan model, evaluation was conducted using various performance metrics. The Anthem dataset, containing unseen peptides as well as unseen alleles, was downloaded along with performance metrics for comparison with other tools such as CapsNet-MHC, MixMHCpred-2.0.2, and NetMHCpan-4.1, as mentioned in [38]. The unseen peptides for trained alleles included HLA-A*26:01 and HLA-A*32:01, while the never-before-seen allele (not included in the training data) was HLA-B*48:01. A confusion matrix was generated to calculate the number of true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP). Sensitivity (recall) measured the proportion of actual binders correctly predicted as binders, while specificity quantified the proportion of non-binders correctly predicted as non-binders. Accuracy represented the overall proportion of correct predictions, and precision indicated the proportion of predicted binders that were actual binders. Additionally, the F1 score and the AUC were used, consistent with the evaluation approach for the BA models.

Further details on the architectures, hyperparameters, and results explored are provided in the supplementary materials.

RESULTS

*Evaluation of BA Model for HLA-A*26:01 Allele*

While the reported results indicate that the model clearly outperforms other tools such as NetMHCpan, NetMHC, and SMMPMBEC in all evaluated metrics, the evaluation suffers from a significant limitation: the dataset used for testing was also part of the training dataset. This overlap makes the results inherently biased and does not reflect the model's performance on unseen data. Although the architecture of the model was designed to minimize overfitting, such evaluations cannot conclusively demonstrate the model’s generalizability. The comparison with other tools was intended as a preliminary exercise due to the lack of an independent BA dataset for testing. Once new data becomes available, it will be essential to repeat the evaluation process to ensure fair and meaningful comparisons. Until then, these results should not be viewed as conclusive evidence of the model's superiority. Nevertheless, it is worth noting that the low root mean squared error and high Pearson correlation suggest that the model's architecture holds promise in effectively learning patterns. Detailed findings are provided in the evaluation notebook included in the supplementary materials, as well as in Table 1.

	Predictor	BiLSTM	NetMHC	NetMHCpan	SMMPMBEC_CPP
Allele	Metric				
HLA-A*26:01	AUC	0.936594	0.742109	0.791067	0.781818
	PC	0.879271	0.751937	0.789063	0.577492
	Tau	0.497056	0.436646	0.451666	0.422900
	F1	0.984564	0.951472	0.962748	0.956739
HLA-A*32:01	AUC	0.918263	0.856498	0.841015	0.702842
	PC	0.866021	0.711823	0.751433	0.613146
	Tau	0.652495	0.448284	0.480669	0.493693
	F1	0.949367	0.888889	0.897436	0.619958

Table 1 Comparison of different metrics evaluating the performance of different tools and the BA allele-specific model on the BLIND dataset.

Length	HLA-I	Tool	AUC	Sensitivity	Specificity	Accuracy
9	HLA-A*26:01	BiLSTM	0.9655	0.8586	0.9385	0.8986
	HLA-A*32:01	BiLSTM	0.9648	0.7012	0.9592	0.8303
10	HLA-A*26:01	BiLSTM	0.9245	0.7538	0.9545	0.8550
		CapsNet-MHC	0.9860	0.9385	0.9697	0.9542
		NetMHCpan-4.1	0.9770	0.9230	0.9580	0.9410
		MixMHCpred-2.0.2	0.9380	0.8970	0.8970	0.8970
	HLA-A*32:01	BiLSTM	0.9210	0.5304	0.9828	0.7576
		CapsNet-MHC	0.9518	0.8733	0.8966	0.8874
		NetMHCpan-4.1	0.9560	0.9210	0.9470	0.9340
		MixMHCpred-2.0.2	0.9390	0.8930	0.9010	0.8960
11	HLA-A*32:01	BiLSTM	0.9584	0.4400	1.0000	0.9608
		CapsNet-MHC	0.9878	0.9400	0.9608	0.9505
		NetMHCpan-4.1	0.9820	0.9380	0.9740	0.9560
		MixMHCpred-2.0.2	0.9850	0.9320	0.9780	0.9550

Table 2 Comparison of different metrics evaluating the performance of different tools and the EL BiLSTM on the Anthem dataset.

Evaluation of EL pan Model

The evaluation of the pan model on unseen peptides trained on known alleles and unseen alleles demonstrates differing levels of performance. In the first scenario, testing on unseen peptides for trained alleles such as HLA-A*26:01, which included 779 peptides of length 9 and 131 peptides of length 11, the model achieved strong results with an AUC of 0.9648, indicating excellent discrimination between binders and non-binders. The sensitivity (recall) was 0.7012, showing that the model correctly identified approximately 70% of the true binders, while specificity was high at 0.9592, reflecting accurate predictions for non-binders. The precision of 0.9450 highlighted the model's ability to confidently predict binders, and the F1 score of 0.8050 demonstrated a balanced trade-off between precision and recall. Looking at TPs and FNs, the confusion matrix indicates 383 TPs while misclassifying 71 FNs. The class distributions reveal that 500 predictions are class 0, and 410 predictions are class 1, closely aligned with the true distributions (456 negatives and 454 positives). These results, along with those for the HLA-A*32:01 allele (which was tested on datasets of 1,373 peptides of length 9, 231 of length 10, and 101 of length 11), are displayed in Figures 1 and 2.

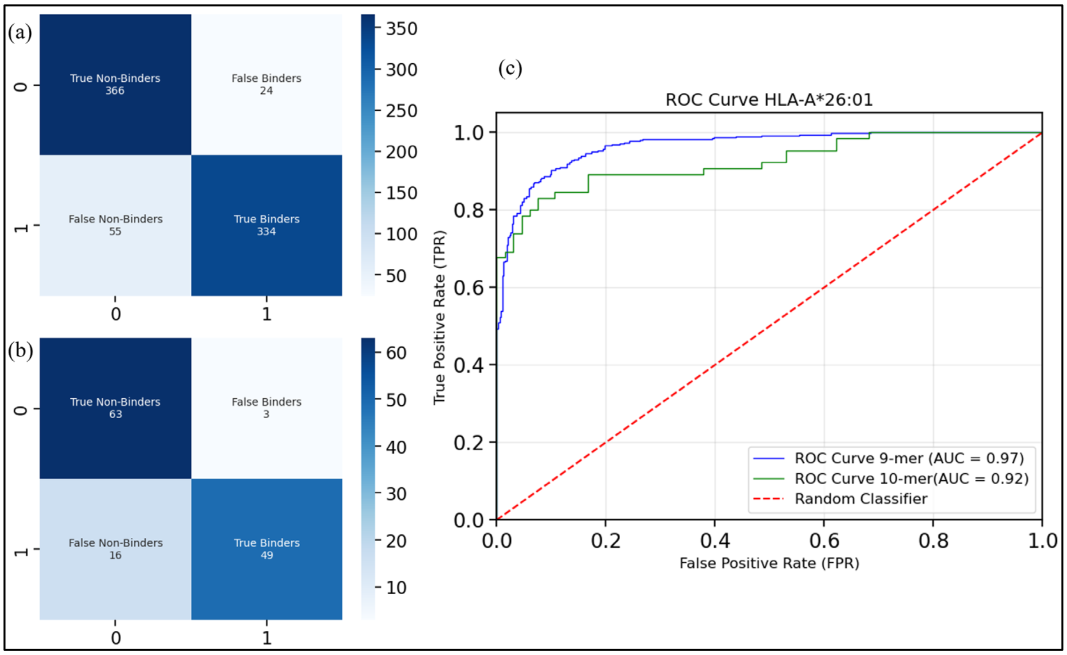

*Figure 1 (a) Confusion matrix showing the EL pan model performance on the HLA-A*26:01 allele for 9-mer peptides; (b) Confusion matrix for 10-mer peptides; (c) ROC curves comparing the model performance for 9-mer and 10-mer peptides.*

When compared to other tools (as seen in Table 2), the model's performance is comparable, although improvements are needed. Longer training times and the use of learning rate decay are necessary to enhance performance. The short training duration likely contributed to a bias toward predicting non-binding peptides. Further discussion on performance and critique is provided in the discussion section.

In contrast, when tested on unseen alleles to evaluate the pan method's generalizability, the model's performance declined significantly (Figure 3). The testing dataset consisted of 49 peptides of length 9. While the AUC remained high at 0.9367, suggesting good discrimination ability, sensitivity dropped to 0.25, indicating that only 25% of true binders were correctly identified. Specificity remained perfect at 1.00, as all non-binders were accurately classified. The precision of 1.00 reflected no false positives, but the low recall led to an F1 score of just 0.4, showing limited effectiveness in handling unseen alleles. Looking at TPs and FNs, only 6 binding peptides were correctly predicted while 18 were misclassified as non-binders.

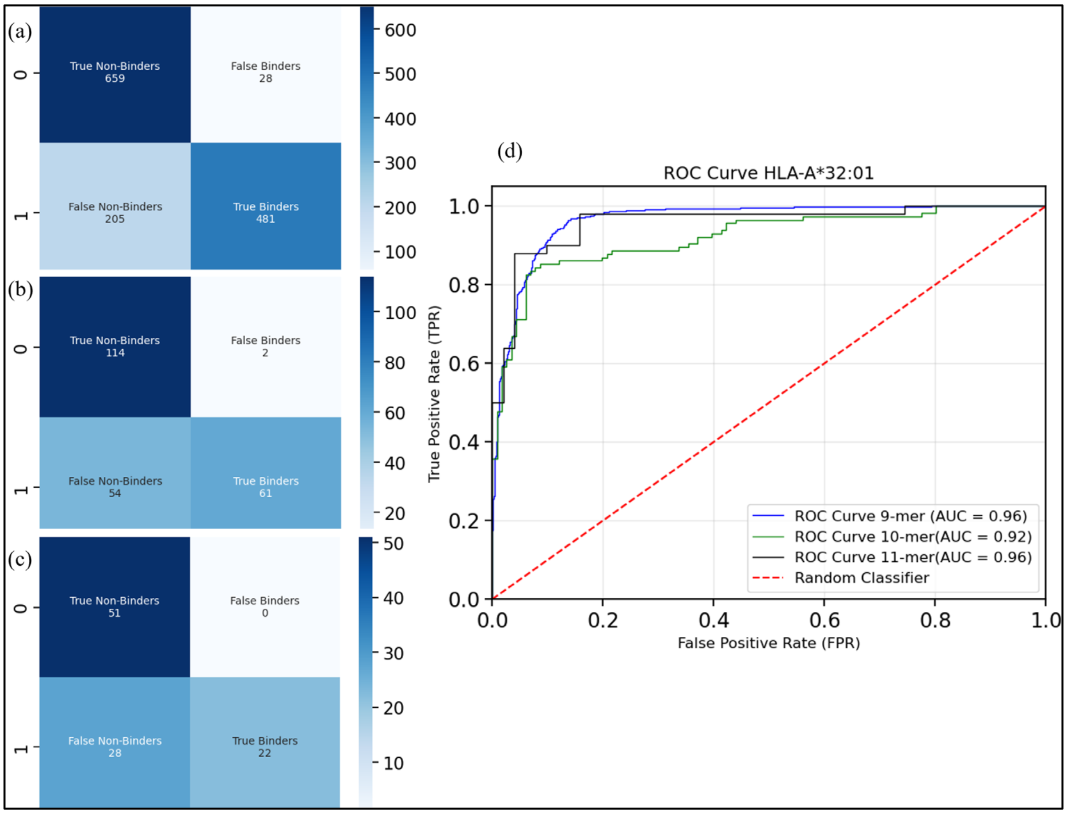

*Figure 1 (a) Confusion matrix showing the EL pan model performance on the HLA-A*32:01 allele for 9-mer peptides; (b) Confusion matrix for 10-mer peptides; (c) Confusion matrix for 11-mer peptides; (d) ROC curves comparing the model performance for 9-mer, 10-mer,, and 11-mer peptides.*

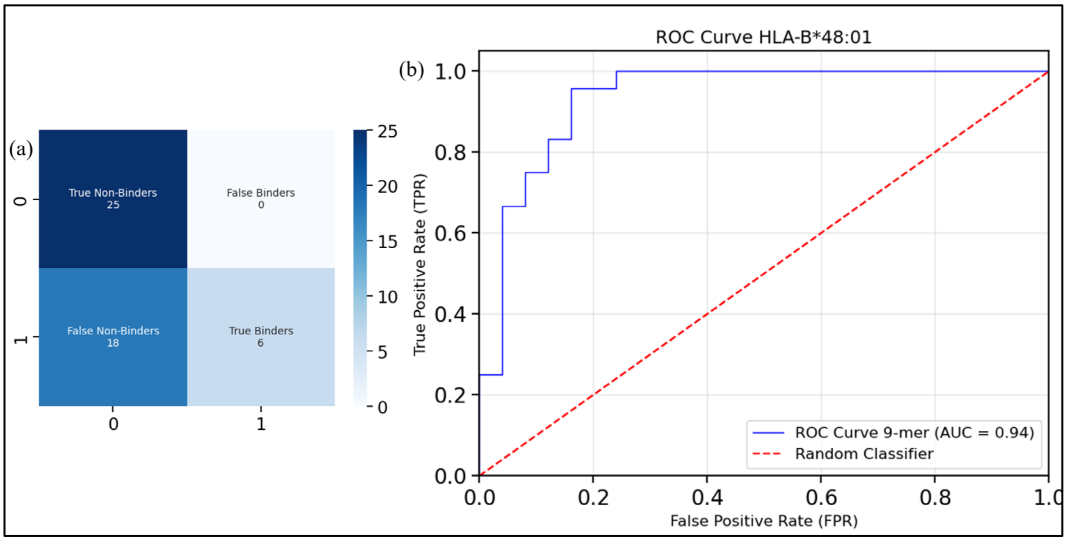

*Figure 2 (a) Confusion matrix showing the EL pan model performance on the unseen HLA-B*48:01 allele for 9-mer peptides; (b) ROC curve for 9-mer peptides.*

DISCUSSION

The Final EL Pan Model Evaluation

Although the accuracy on the validation set during training for the chosen model exceeded 95%, and despite upsampling binders and downsampling non-binders, the final model exhibits high specificity and precision (excelling in identifying negatives) but struggles with sensitivity (failing to capture many positives). While the model is effective at ruling out non-binders, its ability to identify binding peptides is less optimal. The high AUC indicates that the model can discriminate between classes across various thresholds, but its threshold bias strongly favors class 0. The perfect precision suggests that when the model predicts a positive (class 1), it is always correct; however, the number of positive predictions is always smaller. The accuracy of 63.27% for the unseen allele indicates that 63.27% of predictions are correct, but this metric can be misleading due to class imbalance and the model's bias toward class 0. This highlights room for improvement in the pan method, particularly for unseen alleles. While several metrics appear impressive at first glance, a closer inspection reveals significant limitations, suggesting the need for caution in interpreting these results. This raises important concerns about the metrics reported by well-known tools, which often highlight favorable results while omitting others that may reveal shortcomings. This selective presentation of metrics calls into question the reliability of such predictions and underscores the need for transparency in model evaluation.

Computational Resource Limitations and Choice of Optimization Technique

The scale of the dataset, especially after upsampling the binders, posed significant challenges to training efficiency. While Adam's faster convergence was advantageous, the model's training - limited to 10 epochs due to resource constraints - took over 3 days to complete. In contrast, other tools that employ SGD with momentum often train for up to 500 epochs, leveraging their computational efficiency to refine model weights thoroughly [39]. The reduced training time in this project likely hindered the model's ability to fully learn the complex relationships within the data. Future work would benefit from access to more advanced computational resources. On the other hand, comparative analyses indicate that Adam's adaptive learning rate may lead to convergence in narrow local minima, potentially compromising generalizability. In contrast, SGD with momentum can traverse such minima and settle in flatter regions of the loss surface, fostering better generalization[40]. Despite Adam's faster convergence, its limitations in escaping sharp minima suggest that future iterations of this project might benefit from exploring SGD with momentum, particularly if computational resources allow for extended training.

Coding Skills and Resource Constraints

The dual-dataset training approach inspired by NNAlign, aimed at leveraging EL data to improve BA predictions, did not yield significant performance gains to a standard BA model. This outcome reflects both the constraints of limited resources and the challenges posed by the project's timeline hindering the expansion of the coding expertise. Nonetheless, the exploration highlights the potential value of combining diverse data types to enhance predictive accuracy. Improved implementation and resource allocation in future projects may unlock the full potential of this method. Additionally, model performance is significantly impacted by the hyperparameters selection. Randomly chosen hyperparameters may not yield optimal results. Hence, a more systematic approach, such as grid search, is recommended to explore a wider range of hyperparameters combinations and ultimately improve performance and generalization.

Impact of Input Size on Learning Ability

Throughout this research, it has been observed that an increase in input size leads to a decrease in learning ability, assuming the dataset size and architecture remain constant. For example, when this model architecture was applied to inputs with a fixed peptide length (e.g., filtering out all peptides except those of length 9) or to a single allele (eliminating the need for MHC allele encoding, thereby reducing input size), performance metrics improved, and the loss function decreased.

When predicting binding affinity for a single allele, performance improved when the MHC allele sequence encoding was excluded from the input, using the same architecture, hyperparameters, and epochs. While the allele encoding is a constant addition to the input size, it may have contributed to longer convergence times, potentially impacting performance.

Notably, NetMHC operates as an allele-specific method, training a separate predictor for each allele's binding dataset. In contrast, NetMHCpan employs a pan-allele approach, utilizing vector encodings of both peptides and MHC molecule subsequences as inputs. Conventional wisdom holds that NetMHC outperforms for alleles with extensive ligand data, whereas NetMHCpan excels with less well-characterized alleles. Although reducing input size may seem advantageous, including MHC allele information is essential for the pan method to generalize effectively to unseen peptides.

Finally, attempts to incorporate embeddings from ESM2 for peptides and MHC sequences separately did not improve performance. Instead, the model's predictive ability approximated random chance. This likely resulted from the substantial increase in input size introduced by ESM2 embeddings, which overwhelmed the model and hindered effective learning. Previous studies suggest that excessive input dimensionality can obscure meaningful patterns and introduce noise [41]. Exploring dimensionality reduction techniques, such as principal component analysis (PCA) or autoencoders, could address this issue and improve model interpretability and performance.

Effect of Encoding Method

The choice of BLOSUM encoding has increased the model's ability to learn patterns and generalize from data. While one-hot encoding provides precise and direct sequence information, it lacks the ability to capture biochemical and evolutionary relationships between amino acids, limiting the model's capacity to generalize. In contrast, BLOSUM encoding leverages substitution matrices that encode evolutionary and biochemical similarities, allowing the model to infer relationships between similar amino acids even when they are not explicitly present in the training data.

Upsampling Before Cross-Validation

When training the final EL pan model using 5-fold cross-validation, upsampling prior to dataset splitting resulted in similar data distributions across the training and validation sets. While this approach may introduce a degree of data overlap, it is unlikely to significantly affect the model's generalization ability, as the validation sets still function as an internal control. Previous research indicates that neural network architectures designed to be robust against overfitting can effectively mitigate such concerns. However, future studies could address this potential limitation by employing more advanced data-splitting strategies to enhance the evaluation of model robustness.

Future Directions for Improvement

To enhance the predictive performance of the model and address its current limitations, several strategies could be explored in future work. First, the use of convolutional neural networks (CNNs) should be investigated as an alternative or complementary architecture. CNNs are well-suited for capturing spatial patterns and could efficiently learn sequence motifs or local interactions between peptide residues and MHC molecules.

Integrating attention mechanisms into the model architecture could further enhance its capabilities. Attention layers allow the model to dynamically focus on critical regions of input sequences, enabling it to prioritize the most relevant interactions between MHC alleles and peptides. This could improve generalization across diverse alleles and peptides, especially in a pan-allele framework, by reducing the influence of irrelevant input features.

Another promising direction involves incorporating an interaction matrix into the model, as done by tools like RPEMHC[42].

Additionally, a more detailed exploration of dimensionality reduction techniques for embeddings generated from tools like ESM2 could help reduce input size while enabling the model to focus on biologically relevant patterns. This approach could also serve as a method for transfer learning, allowing the model to refine its predictions effectively. As mentioned earlier in this report, these ideas have been partially explored, with the results included in the supplementary material. However, dedicating more time and effort to these strategies is warranted, as they appear to hold significant promise for improving model performance[38].

CONCLUSION

The model proposed in this project demonstrates potential for pan-specific HLA-peptide binding prediction, accurately handling peptides of varying lengths due to its BiLSTM architecture. The results emphasize the importance of adequate training data and computational resources for achieving high predictive performance in deep learning models. Future improvements could integrate sequence- and structure-based predictions to further enhance accuracy, advancing immunotherapy and immune response research.

SUPPLEMENTARY MATERIALS

All scripts, environment details, evaluations, and neural network architectures tested, along with their results, can be found in the “Z:\Andy\final” directory on chatprot.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Dr. Anne-Christine Uldry, whose guidance, regular discussions, and collaborative spirit were invaluable throughout this project. Her dedication and support greatly enriched my learning experience. I also extend my thanks to Prof. Dr. Manfred Heller for the insightful meeting that helped shape key aspects of this work. Lastly, I am grateful to Andrew Ng and his online Coursera course, which provided foundational knowledge that was instrumental in completing this research. I also acknowledge the use of ChatGPT for refining sentence structure and enhancing the clarity of the English in this report.

REFERENCES

[1] Beecher SM, O'Leary DP, McLaughlin R, Kerin MJ. The impact of surgical complications on cancer recurrence rates: a literature review. Oncol Res Treat. 2018;41:478-482.

[2] Alieva M, van Rheenen J, Broekman MLD. Potential impact of invasive surgical procedures on primary tumor growth and metastasis. Clin Exp Metastasis. 2018;35:319-331.

[3] Mansoori B, Mohammadi A, Davudian S, Shirjang S, Baradaran B. The different mechanisms of cancer drug resistance: a brief review. Adv Pharm Bull. 2017;7:339-348.

[4] Willers H, Azzoli CG, Santivasi WL, Xia F. Basic mechanisms of therapeutic resistance to radiation and chemotherapy in lung cancer. Cancer J. 2013;19:200-207.

[5] Joyce S. Immunoproteasomes edit tumours to escape immune recognition. Eur J Immunol. 2015;45:3241-3245.

[6] Yewdell JW, Bennink JR (1999) Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses. Annual Review of Immunology 17: 51–88.

[7] Hoof, I., Peters, B., Sidney, J., Pedersen, L. E., Sette, A., Lund, O., Buus, S., and Nielsen, M. (2009) NetMHCpan, a method for MHC class I binding prediction beyond humans. Immunogenetics 61, 1–13

[8] Sette A, Sidney J (1999) Nine major HLA class I supertypes account for the vast preponderance of HLA-A and –B polymorphism. Immunogenetics 50: 201–212.

[9] Doytchinova IA, Flower DR (2001) Toward the Quantitative Prediction of TCell Epitopes: CoMFA and CoMSIA Studies of Peptides with Affinity for the Class I MHC Molecule HLA-A*0201. J Med Chem 44: 3572–3581.

[10] Bordner AJ, Abagyan R (2006) Ab initio prediction of peptide-MHC binding geometry for diverse class I MHC allotypes. Proteins 63: 512–526.

[11] Antes I, Siu SW, Lengauer T (2006) DynaPred: A structure and sequence based method for the prediction of MHC class I binding peptide sequences andconformations. Bioinformatics 22: e16–24.

[12] Fagerberg T, Cerottini JC, Michielin O (2006) Structural prediction of peptides bound to MHC class I. J Mol Biol 356: 521–546.

[13] Deres K, Schumacher TN, Wiesmuller KH, Stevanovic S, Greiner G, Jung G, et al. Preferred size of peptides that bind to H-2 Kb is sequence dependent. Eur J Immunol. 1992;22(6):1603–8.

[14] Rammensee H, Bachmann J, Emmerich NP, Bachor OA, Stevanovic S. SYFPEITHI: database for MHC ligands and peptide motifs. Immunogenetics. 1999;50:213–9.

[15] Vita,R. et al. (2015) The immune epitope database (IEDB) 3.0. Nucleic Acids Res., 43, D405–D412.

[16] Lundegaard,C. et al. (2008) Accurate approximation method for prediction of class I MHC affinities for peptides of length 8, 10 and 11 using prediction tools trained on 9mers. Bioinformatics, 24, 1397–1398.

[17] 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.

[18] Andreatta M, Nielsen M. Gapped sequence alignment using artificial neural networks: application to the MHC class I system. Bioinformatics. 2016;32(4):511–7.

[19] Nielsen, M., and Andreatta, M. (2017) NNAlign: a platform to construct and evaluate artificial neural network models of receptor-ligand interactions. Nucleic Acids Res. 45, W344–W349.

[20] Bassani-Sternberg,M. et al. (2015)Mass spectrometry of human leukocyte antigen class I peptidomes reveals strong effects of protein abundance and turnover on antigen presentation. Mol. Cell. Proteomics MCP, 14, 658–673.

[21] Nielsen M, Lundegaard C, Worning P, Lauemoller SL, Lamberth K, et al. (2003) Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci 12: 1007–1017.

[22] Buus S, Lauemoller SL, Worning P, Kesmir C, Frimurer T, et al. (2003) Sensitive quantitative predictions of peptide-MHC binding by a ‘Query by Committee’ artificial neural network approach. Tissue Antigens 62: 378–384.

[23] Baldi P, Brunak S (2001) Bioinformatics: The Machine Learning Approach, 2nd edition. CambridgeMass.: MIT Press.

[24] Peters B, Bui HH, Frankild S, Nielson M, Lundegaard C, et al. (2006) A community resource benchmarking predictions of peptide binding to MHC-I molecules. PLoS Comput Biol 2: e65.

[25] Abelin,J.G., Keskin,D.B., Sarkizova,S., Hartigan,C.R., Zhang,W., Sidney,J., Stevens,J., Lane,W., Zhang,G.L., Eisenhaure,T.M. et al. (2017) Mass spectrometry profiling of HLA-associated peptidomes in mono-allelic cells enables more accurate epitope prediction. Immunity, 46, 315–326.

[26] O’Donnell,T.J., Rubinsteyn,A., Bonsack,M., Riemer,A.B., Laserson,U. and Hammerbacher,J. (2018) MHCflurry: open-source Class I MHC binding affinity prediction. Cell Syst., 7, 129–132.

[27] Jurtz,V., Paul,S., Andreatta,M., Marcatili,P., Peters,B. and Nielsen,M. (2017) NetMHCpan-4.0: improved peptide-MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. J. Immunol., 199, 3360–3368.

[28] Reynisson B, Alvarez B, Paul S et al. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration ofMS MHC eluted ligand data. Nucleic Acids Res 2020b;48:W449–W454.

[29] Alvarez,B., Reynisson,B., Barra,C., Buus,S., Ternette,N., Connelley,T., Andreatta,M. and Nielsen,M. (2019) NNAlign MA; MHC peptidome deconvolution for accurate mhc binding motif characterization and improved t-cell epitope predictions. Mol. Cell Proteomics, 18, 2459–2477.

[30] Nielsen, M., C. Lundegaard, P. Worning, S. L. Lauemøller, K. Lamberth, S. Buus, S. Brunak, and O. Lund. 2003. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci. 12: 1007–1017.

[31] Nielsen, M., and M. Andreatta. 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.

[32] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

[33] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673-2681.

[34] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[35] Nielsen M, Lundegaard C, Worning P, Lauemøller SL, Lamberth K, Buus S, Brunak S, Lund O. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci. 2003 May;12(5):1007-17. doi: 10.1110/ps.0239403. PMID: 12717023; PMCID: PMC2323871.

[36] Zhou, J., et al. (2018). Evaluation metrics for predictive models in bioinformatics. Bioinformatics, 34(5), 787-795.

[37] Press, W.H., et al. (1989). Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press.

[38] Kalemati, M., Darvishi, S. & Koohi, S. CapsNet-MHC predicts peptide-MHC class I binding based on capsule neural networks. Commun Biol 6, 492 (2023). https://doi.org/10.1038/s42003-023-04867-2.

[39] Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. Proceedings of the 30th International Conference on Machine Learning.

[40] ZHOU, Pan; FENG, Jiashi; MA, Chao; XIONG, Caiming; HOI, Steven C. H.; and E, Weinan. Towards theoretically understanding why SGD generalizes better than ADAM in deep learning. (2020). Proceedings of the 34th Conference on Neural Information Processing Systems, NeurIPS 2020, Vancouver, Canada, December 6-12. 1-12.

[41] Lin, Z., Akin, H., Kriechbaumer, V., & Koeppl, H. (2017). Dimensionality reduction for computational biology. Nature Methods, 14(8), 665-670.

[42] Xuejiao Wang, Tingfang Wu, Yelu Jiang, Taoning Chen, Deng Pan, Zhi Jin, Jingxin Xie, Lijun Quan, Qiang Lyu, RPEMHC: improved prediction of MHC–peptide binding affinity by a deep learning approach based on residue–residue pair encoding, Bioinformatics, Volume 40, Issue 1, January 2024, btad785, https://doi.org/10.1093/bioinformatics/btad785.

[43] Paul, S. et al. HLA class I alleles are associated with peptide-binding repertoires of different size, affinity, and immunogenicity. The Journal of Immunology, 1302101 (2013).

[44] Blom N, Hansen J, Blaas D, Brunak S. Cleavage site analysis in picornaviral polyproteins: discovering cellular targets by neural networks. Protein Sci. 1996 Nov;5(11):2203-16. doi: 10.1002/pro.5560051107. PMID: 8931139; PMCID: PMC2143287.