Authors
Daniel Fridman, Matthew Benstock, Jeremy Goldwasser

Title
Predicting Binding Affinity of Peptide-MHC Interactions with Structurally-Aware Embeddings

Abstract
The Major Histocompatibility Complex (MHC) plays a significant role in the human immune response. It is theorized that for peptide epitopes from either foreign pathogens or from cancerous cells to be recognized by the host immune system, they must bind with high affinity to the MHC allele. When these peptides bind to MHC molecules, they are recognized as potential antigens by T cell receptors (TCRs), which then trigger an immune response. An important area of computational immunology research has been devoted to the task of predicting the binding affinity between unique peptide signatures and MHC Class I alleles. This information is of crucial importance in both vaccine and tumor immunotherapy development, as the vaccine or immunotherapy uses peptides from the antigen known to have high MHC allele binding affinity in order to stimulate an immune response.

We developed an end-to-end neural network model for binding affinity prediction inspired by the MHCSeqNet neural network. However, while MHCSeqNet uses a simple Skip-Gram language model to embed amino acids in the peptide sequence based on their local linear context, we use the model proposed by Bepler and Berger (2019) to produce more robust embeddings that take proteins' 3D structure into account. Additionally, using the Bepler and Berger embedding model, we are able to produce embeddings for any peptide and MHC allele sequences, including sequences which contain gaps. We demonstrate that our model performs very well, attaining a training accuracy of 98%, a testing accuracy of 91.4%, and a testing ROC AUC of 0.97. Lastly, in response to the Coronavirus pandemic, we applied our binding affinity prediction model to peptide sequences obtained from the 2019 Novel Coronavirus. In doing so, we generated a list of coronavirus peptides that had the highest binding affinity to various MHC-I alleles, indicating potential targets for Coronavirus vaccine development.

1. Introduction
1.1 Background
Our immune systems are incredibly adept at detecting and fighting pathogens. In the case of many cancers and infectious diseases, they detect pathogens by way of their antigens, whose epitope peptides bind to the Major Histocompatibility Complex (MHC) Class I molecules that exist on the surface of all nucleated mammalian cells (1). MHC provides Cytotoxic T cells with a "window" through which to examine the afflicted cell. When they recognize the peptide fragments bound to the MHC, they bind to the peptide-MHC complex and kill the whole cell. They then alert the immune system of the presence of these antigens by causing Helper T cells to secrete a chemical called cytokines, which stimulates the differentiation of B cells into plasma cells; these plasma cells produce Y-shaped antibody proteins called immunoglobulin.

Immunoglobulin latches onto antigens, indicating that the immune system needs to attack them (1).

The binding of peptide chains with MHC is a key stage in the immune response. It is thought that the peptide chains are required to bind to the MHC complex with high binding affinity in order to initiate an effective immune response (2). These chains vary in length, although the majority of the peptides that bind to MHC-I consist of 8-10 amino acids. However, MHC-I can bind to certain peptides with 11-14 amino acids, and in rare cases to peptides with 15 or more (1). Regardless of length, only a tiny fraction of an antigen's peptides are likely to bind, or have high "binding affinity," with a given MHC-I allele. These are the epitopes that successfully activate the immune response and thus serve as effective targets for vaccine development. Selection of epitopes for vaccine design remains a challenging and slow component of the vaccine development process, but computational approaches to predicting peptide-MHC allele binding affinities may contribute to more efficient screening of promising vaccine candidates.

1.2 Related Work

To help select epitopes, neural networks can be used to predict binding affinity between peptide sequences and MHC-Class I alleles. This is a complicated task, as there are complicated nonlinear feature interactions within the sequences of amino acids. For example, these interactions cause the linear sequence to fold into a 3D structure. To account for them, *High-Order Neural Networks and Kernel Methods for Peptide-MHC Binding Prediction* (Kuska et al., 2014) uses semi-Restricted Boltzmann Machines to pre-train a feedforward high-order neural network and a high-order Kernel Support Vector Machine. While state-of-the-art for the time, this model was limited in the sense that it could not handle variable length input. It only focused on 9-mer peptides (5).

MHCFlurry: Open-Source Class I MHC Binding Affinity Prediction (O'Donnell et al., 2018) improved on this somewhat by training separate allele-specific models for input sequences of length 8 through 15. While the various models achieved impressive results, this solution is inefficient, inflexible, and inelegant (6).

MHCSeqNet (Phloyphisut et al., 2019) is a much more robust model that utilizes principles of Natural Language Processing. By using Recurrent Neural Networks (RNNs), it is able to process amino acid sequences of any length - both in the peptide and in the MHC-Class I allele. In addition, it generates a context-aware embedding vector for each amino acid in both input peptide and allele sequences, allowing the model to learn relationships between peptide and allele sequences (9).

To get strong peptide embeddings, it performed unsupervised learning on a Skip-Gram language model as well as the ProtVec model, training on millions of unique peptide sequences. Both Skip-gram (2013) and ProtVec predict the likelihood of the surrounding context words (or amino acids) of a center word (or amino acid). While this approach produces useful embeddings

which significantly improve model performance, they have significant drawbacks for embedding peptide sequences as described below.

Word and Peptide Embeddings (Mikolov, 2013; Ehsaneddin, 2015; Bepler, 2019)
Context-aware embedding vectors of sequences have been shown to successfully capture semantic relationships between words in natural language applications, grouping related words close to one another in vector spaces. The idea behind these so-called *word2vec* models is the concept of *distributed representation*, that information about an item can be encoded as a function of its interactions with its members. In the context of word embeddings, this means that related words would appear within similar contexts and thus models such as the skip-gram model and the continuous-bag-of-words (CBOW) model generate vector embeddings for words dependent on their direct local context (7).

Following the success of word embeddings in NLP tasks, the idea was extended to biological sequences, including peptides. Peptide sequences, which are represented as strings of amino acids, can be thought of as 'peptide sentences' consisting of 'amino acid words' in which related amino acids would be surrounded by similar amino acid contexts. This approach has been used to extract useful amino acid embeddings such as the ProtVec model, which is capable of encoding important features of amino acids including their biochemical and biophysical properties. Furthermore, these embeddings, which can be trained in an unsupervised fashion, can be used as pre-trained inputs into specific supervised biological tasks such as protein-protein interaction prediction (8).

Using the ideas of peptide embeddings, Phloyphisut et al. (2019) employed both the skip-gram model and ProtVec model approach as pre-trained inputs for the task of predicting peptide - MHC-allele binding affinities (*see above*). While this is a reasonable approach, embeddings based on language modelling techniques represent peptides as 1-dimensional linear sequences. However, peptides fold into complex 3-D structures. This results in amino acids which may be linearly distant to enter each other's local contexts in 3-dimensional space. Consequently, it is known that not only the primary sequence, but also the 3-dimensional structural properties (secondary and tertiary peptide structures) of proteins are essential in determining their biological functions. As a result, peptide sequence embeddings which only account for linear amino acid contexts may disregard important structural information that would contribute to the accuracy of predicting a peptide's function.

Recently, Bepler et al. (2019) showed that protein sequence embeddings can be learned using structural information. The peptide embedding model is capable of accounting for whole sequence similarity between different proteins as well as local contacts between residues in 3-dimensional protein structures, allowing for representation of both global structural similarity between proteins and residue-residue contacts within individual proteins. As a result, this model produces more structurally-conscious amino acid embeddings (3).

2. Methods

Our goal is to show that sequence embeddings that take into account 3-Dimensional structures of human peptides can accurately predict peptide MHC allele binding affinity.

2.1 Data and Preprocessing

We obtained two of the datasets used in the *MHCSeqNet* paper (*data available on MHCSeqNet GitHub, link below*). The authors preprocessed those data sets into the following cleaned versions: The first, from the Immune Epitope Database (IEDB), is a (255039 x 3) data frame. Each row represents a unique peptide-allele interaction. The three column variables are:

1. “Peptide” - A peptide sequence string (each letter is one of the amino acid characters).
2. “Allele” - A string of characters representing the codename of an MHC allele. There are both class I and class II alleles.
3. “Binding Quality” - A binary assessment of the peptide-allele binding affinity (positive if they interact, and negative if they do not).

The second, a cleaned set from the Immunopolymorphism database (IPD), is a (4867 x 4) data frame. Each row represents a unique MHC class I allele and the parts of its sequence that are known to interact with peptides. The four column variables are:

1. “MHC_allele” - A string of characters representing the MHC allele.
2. “Beta_sheet_res_3-125” - A string of characters representing the Beta sheet of the MHC protein that is known to interact with binding MHC peptides. The residue of these amino acids are 3-37 and 94-126.
3. “alpha_helix_res_140-179” - A string of characters representing the Alpha helix also known to interact with MHC peptides. The residue of these amino acids is 140-179.
4. “alpha_helix_res_50-84” - A string of characters representing another Alpha helix. The residue of these amino acids are 50-84.

Because the second data set only had sequences of class I alleles, we removed all rows in the first data set that had class II alleles. This changed the dimensionality of our first data set to (229758 x 3).

2.2 Sequence Embeddings

Using these two data sources, we constructed our own data embeddings. To start, in data set one, for each allele in column two, we find its corresponding “alpha_helix_res_140-179” and “alpha_helix_res_50-84” sequence in data set two (we ignored “Beta_sheet_res_3-125” following what was done in the *MHCSeqNet* paper). From here, we create two new data matrices for each of these sequences. Our first matrix is for the “alpha_helix_res_140-179” sequence. Each row of the matrix corresponds to an allele, and each column corresponds to a letter in its sequence. The columns are encoded based on the Uniprot alphabet. The second matrix is for the “alpha_helix_res_50-84” sequence and follows the same structures as just described. One note is that some of the sequences have gaps but this is accounted for in the

Uniprot encoding. Lastly, we created a third matrix for the peptides - again, using the same procedure.

From here, each of these matrices is converted into a 3-D tensor. The first dimension of the tensor corresponds to an allele (a row in the previous matrices). The second dimension corresponds to the letters in our sequences. Now, the dimension is equal to the maximum length sequence of an allele. If, for a given allele, it had fewer characters than the maximum sequence, we padded the corresponding missing row with zeros. The third dimension represents our feature embeddings and is fixed at length 100 as was done by *Bepler et al.* To summarize, we now have three 3-D tensors for the "alpha_helix_res_140-179," "alpha_helix_res_50-84," and our peptide sequences, where the dimensionality at a high level represents (allele/peptide, maximum length of a sequence, embedding length).

To create our embeddings, we used the pretrained full SSA sequence embedding model from *Bepler et al.* (*Pretrained model available on Bepler et al. GitHub, link below*). This model "maps sequences of amino acids to sequences of vector representations, such that residues occurring in similar structural contexts will be close in embedding space" (3). We also use these pretrained embeddings as input to our model to predict peptide-allele binding affinity.

2.3 Architecture:

2.3.1 Sequence Embedding Architecture

As described in the Bepler et al. paper, the structure-based embedding model model consists of an unsupervised bi-directional LSTM sequence-to-embedding encoder with two supervised feedback mechanisms. The first mechanism accounts for global structural similarity between different proteins by using a soft symmetric alignment (SSA) metric and the second accounted for pairwise residue contacts within the 3-dimensional contexts of individual proteins (3). A detailed schematic of the embedding model architecture is shown in *Additional Figures 1 and 2* (3). (*Note: we did not implement this architecture ourselves).

2.3.2 Peptide - MHC-allele Binding Affinity Prediction Model Architecture

In order to predict peptide-MHC allele binding affinities as a probability of binding, we implemented a modified version of the MHCSeqNet model architecture, shown in *Additional Figure 3* (9). Our model consists of 3 major components: a peptide processing layer, an allele processing layer, and a feedforward output layer which outputs a probability of peptide-MHC allele binding between 0 and 1 (ref. 5).

(I) *Peptide Processing Layer*

The peptide processing layer takes as input the 3-D peptide embedding tensors (batch_size x sequence_length x embedding_dimension) and processes the sequence embeddings through a bi-directional GRU. The GRU consists of a single layer with 100-dimensional hidden units. The hidden unit for the last time step incorporates information from the GRU forward pass for the entire sequence, while the hidden unit from the first time step incorporates information from the

GRU backward pass for the entire sequence. In order to incorporate information from both the forward and backward passes of our bi-directional GRU, the hidden units from the first time step and last time steps are concatenated to produce a 2-D tensor of size (batch_size x hidden_size*2). This tensor is the output of our peptide processing layer and is fed as input into the output layer.

(II) *MHC allele Processing Layer*
Similar to the peptide processing layer, the MHC allele processing layer takes as input the 3-D allele embedding tensors for both the alpha helix residue 140-179 embedding and the alpha helix residue 50-84 embedding. The embedding model is capable of accounting for gaps within sequences and thus can produce embeddings for all 229758 alleles in our dataset, some of which contain gaps. The alpha_res_140_179 and alpha_res_50_84 embedding tensors are then concatenated along the sequence_length dimension to create a single allele sequence embedding tensor of dimension (batch_size x (alpha_140_179 sequence length + alpha_50_84 sequence_length) x embedding_dimension).

This tensor is fed as input into a bi-directional GRU similarly to the peptide processing layer. The bi-directional GRU consists of a single layer with 100-dimensional hidden units. The hidden units from the first and last time step are concatenated to account for sequence information from both the forward and backward pass of the GRU to produce a 2-D tensor of size (batch_size x hidden_size*2). This tensor is the output of our allele processing layer and is fed as input into the output layer.

(III) *Output Layer*
The 2D tensor hidden unit outputs from both the peptide processing and allele processing layers are fed into the output layer. The tensors are passed together through two fully connected linear layers as (W_1 * peptide_hidden_tensor + W_2 * allele_hidden_tensor) with ReLU activation and 200 hidden units for both linear layers. The third and final layer contains a single output unit which is passed through a sigmoid activation function to produce a probability (between 0 and 1) of our input peptide binding to our input allele.

3. Model Training and Performance Assessment

From the original dataset of 229758 examples, we randomized the data and split it such that 80% was set to the training set and 20% was set to the test set. Following the implementation from MHCSeqNet, we trained our model for 200 epochs using the Adam optimization algorithm with a learning rate of 0.001. We used a BCELoss to evaluate the loss for our output binding affinity predictions against a binary 0 or 1 target classification, corresponding to either positive (1) or negative (0) binding affinity.
Accuracy was calculated based on a 0.5 threshold (absolute values of the difference between predicted binding probability and target less than zero were marked as correct predictions). Additionally, the area under the receiver operating curve (ROC AUC) was obtained to assess test accuracy.

Pytorch version 1.2.0 was used (version 1.2.0 or lower are compatible with the pretrained embedding model from Bepler et al.) with CUDA 9.2. The models were trained on Google CoLab Pro using the 'Tesla P100-PCIE-16GB' GPU.

3. Results

We trained our models using structure-based sequence embeddings for both peptides and class I MHC alleles. After 200 epochs of training, our train set attained an average epoch accuracy of 98.37% with an average epoch BCE loss of 0.045. The trained model was then evaluated on the test set and attained a 91.95% test accuracy with a 0.014 test loss. The true positive and false positive rates were also computed and the area under the receiver operating curve (ROC AUC) was evaluated to be 0.97 for the test set. The ROC curve is shown below.

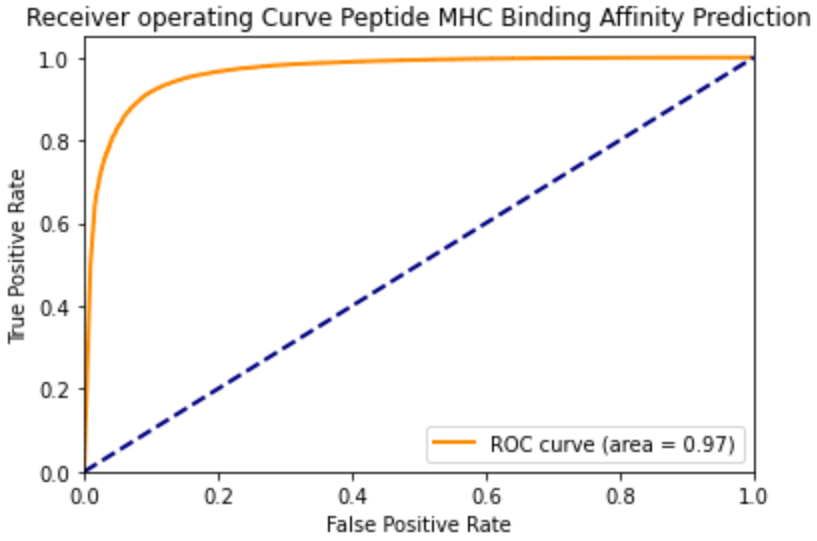

Figure 1: ROC curve for test set of peptide-MHC allele binding affinity predictor. AUC = 0.97.

In addition to overall model performance, we decided to evaluate our model's predicted binding affinities across different peptide lengths. Peptide lengths in the dataset range from 8 to 15 amino acid polymers and the fraction of data values for each polymer are as follows:

`{8-mer: 0.032, 9-mer: 0.597, 10-mer: 0.165, 11-mer: 0.077, 12-mer: 0.035, 13-mer: 0.026, 14-mer: 0.021, 15-mer: 0.047}`

An important observation is that more than half of our dataset consists of 9-mer peptides, with significantly fewer peptides of other lengths. Additionally, as shown in the table below, 9-mers appear to have the largest variation in target binding affinities, with an average target binding affinity of 0.6, while binding affinity data for other peptide lengths appear to mostly be high affinity binding. Nonetheless, as shown in the table below, our attains extremely precise average binding affinity predictions for all peptide lengths.

Evaluation of test set performance by peptide length:

Peptide (polymer) Length	Average Target Binding Affinity	Average Predicted Binding Affinity
8-mer	0.9501	0.9475
9-mer	0.6057	0.6037
10-mer	0.7991	0.7890
11-mer	0.9731	0.9683
12-mer	0.9971	0.9979
13-mer	0.9975	0.9952
14-mer	0.9901	0.9811
15-mer	0.8999	0.8369

Table 1: Breakdown of average target binding affinities and the corresponding average predicted binding affinities by different length peptides. Peptide lengths from our dataset range from 8-15 amino acids.

4. **Application of Peptide-MHC Allele Binding Prediction Model to SARS-CoV-2 Peptides**

4.1 Data Processing

As the novel coronavirus (SARS-CoV-2) continues to wreak havoc across the globe, we decided to apply our binding affinity model to its peptides. While the entire proteome has not yet been accurately sequenced, Uniprot has publicly released 14 of the 28 confirmed SARS-CoV-2 proteins. From these 14 protein sequences, we obtained the peptides by using the NetChop neural network to predict cleavage probabilities at each amino acid. We selected peptides of length 8-15 spanning between cleavage sites with probability 0.7 or higher. Among these peptides, we further narrowed them down by selecting only the ones that immediately followed and preceded other candidate peptides – here, “immediately” meaning by a single amino acid.

It is important to note that we are not certain whether the roughly 5000 peptide sequences we selected from the coronavirus proteome are its actual peptides. Firstly, the cleavage probabilities are just estimates from a neural net. Secondly, many of the generated peptides overlap with at least one other. Amino acids can only belong to one peptide, so peptide sequences that occupy the same region of a protein are mutually exclusive. However, it is likely that in each of these overlapping cases, one of the peptides is correct.

MHC Class I alleles are known to “generally present peptides of eight to ten amino acids” (Burrows, et al.). To account for this, we evaluated our model on a subset of the generated

coronavirus peptides with a much higher proportion of 8-10-mers. This subset randomly selected 625 peptides: 100 8-mers, 250 9-mers, 150 10-mers, and 25 of sequences of length 11 to 15.

We ran each of these peptides through our network paired with one of 30 MHC alleles. Of these alleles, 10 possessed the HLA-A gene, 10 possessed HLA-B, and 10 possessed HLA-C. The chosen alleles in each HLA gene class were the ones most frequently used in the IEDB database.

4.2 Model Evaluation on Coronavirus

Among these peptide-MHC interactions, our model predicted an average binding affinity of 0.71. Most of the actual binding affinity values, however, were close to either 0 or 1. 21% of them were less than 0.1, and 63% were over 0.9; the median value was 0.997. That our model classified an overwhelming majority of peptide-MHC pairs as either very likely or very unlikely to bind suggests a high degree of confidence with its predictions. This is an encouraging result. Moreover, the abundance of high binding affinities confirms our expectation that the human immune system would recognize foreign antigens as such.

Analyzing the results yielded some interesting findings. Firstly, we broke down the results by MHC gene. The general trend we noticed was that on average, coronavirus peptides have highest binding affinity to HLA-C genes, and lowest binding affinity to HLA-A genes. The mean binding affinity was 0.55 for HLA-A genes, 0.71 for HLA-B, and 0.89 for HLA-C. This discrepancy is reinforced looking at the percentage of binding affinities over 0.9: only 45% for HLA-A, but 62% for HLA-B and 84% for HLA-C. While the biological significance of these results may require further investigation, it is worthwhile noting that HLA-A, HLA-B, and HLA-C alleles contain genetic variants (polymorphisms) in their alpha1 and alpha2 domains which are known to play a role in peptide binding. These genetic variants have been found to influence peptide binding specificity (10). Since we encoded the alpha helix residues 140-179 and 50-84, coming from the alpha1 and alpha2 domains in our MHC allele embeddings, it is possible that our model is picking up on differences between the sequences of HLA-A, HLA-B, and HLA-C alpha domains and is thus capable of predicting allele binding specificity. While the results require further experimental study, our finding may suggest that coronavirus peptides tend to bind more specifically to HLA-C alleles over other MHC class I alleles.

Secondly, we broke down binding affinities by protein. To do so, we examined which coronavirus proteins each peptide could be found in. The protein with the highest mean binding affinity, 0.83, was the Nucleoprotein (P0DTC9); second to this was the Envelope Small Membrane Protein (P0DTC4). The mean binding affinities of the two largest proteins that have been sequenced - Replicase polyprotein 1ab (P0DTD1) and Replicase polyprotein 1a (P0DTC1) - were close to the overall average, at 0.71 and 0.70 respectively. So was the mean binding affinity of the Spike glycoprotein, which many researchers consider a viable target for a vaccine. As before, its peptides bind with lowest affinity on average to HLA-A alleles and highest to HLA-C. While these results again require further investigation as to their biological significance,

the differences in average protein binding probabilities may serve as an early framework for screening potential peptide vaccine candidates.

Lastly, we analyzed binding affinities by peptide length. This showed another interesting result: our model predicted significantly higher binding affinities to longer proteins. This is surprising because MHC is empirically shown to more frequently bind to shorter peptides.

SARS-CoV-2 Peptide Length	Average Predicted Binding Affinity	Percentage with binding affinities over 0.9
8-mer	0.82	76%
9-mer	0.57	45%
10-mer	0.70	61%
11-mer	0.89	86%
12-mer	0.84	90%
13-mer	0.98	96%
14-mer	0.98	97%
15-mer	0.94	93%
All	0.71	63%

Table 2: Breakdown of average predicted binding affinity and the percentage of binding affinity probability predictions greater than 0.9 for different length peptides. Peptide lengths range from 8-15 amino acids.

SARS-CoV-2 Peptide Length	Average Predicted Binding Affinity	Percentage with binding affinities over 0.9
HLA-A	0.55	45%
HLA-B	0.71	62%
HLA-C	0.89	84%
All	0.71	63%

Table 3: Breakdown of average predicted binding affinity and the percentage of binding affinity probability predictions greater than 0.9 for MHC alleles with different HLA genes.

5. Conclusion

Inspired by the results of *MHCSeqNet*, we sought to build a peptide-MHC allele predictor model. However, unlike *MHCSeqNet* we implemented a structurally-aware sequence embedding based on the work of *Bepler et al*. Thus, our model is able to take into account structural information about the interaction of peptides with MHC alleles.

Using this method, we were able to obtain 98.37% training accuracy and 91.92% testing accuracy with an AUC metric of 0.97. Although these results are impressive in and of themselves, model performance may further be improved with additional hyperparameter tuning and additional data processing such as removing sequences which contain gaps.

Lastly, we applied our binding affinity prediction model to the peptides of SARS-CoV2 and observed differences in binding affinity predictions across different alleles as well as for different proteins of the virus. While the proper interpretation of these results requires further biological investigation, our results provide interesting starting points for further research into the interaction between the novel coronavirus and the host immune system as well as a potential framework for screening for potential coronavirus vaccine candidates.

Acknowledgements
In order to develop our model, we used the trained full SSA model from Bepler et al. as well as the general architectural structure of MHCSeqNet.
We referred to the following code available on GitHub:
https://github.com/tbepler/protein-sequence-embedding-iclr2019
https://github.com/cmb-chula/MHCSeqNet
We obtained the coronavirus protein sequences from Uniprot:
https://covid-19.uniprot.org/uniprotkb?query=*

References

(1) Janeway CA Jr, Travers P, Walport M, et al. Immunobiology: The Immune System in Health and Disease. 5th edition. New York: Garland Science; 2001. Antigen recognition by T cells. Available from: https://www.ncbi.nlm.nih.gov/books/NBK27098/

(2) Engels B., et al. “Relapse or eradication of cancer is predicted by peptide-major histocompatibility complex affinity.” Cancer Cell. 2013;23(4);516-26.

(3) Bepler, Tristan, and Berger, Bonnie. “Learning Protein Sequence Embeddings Using Information from Structure.” International Conference on Learning Representations, 2019, doi:arXiv:1902.08661.

(4) Burrows, Scott R., et al. “Have We Cut Ourselves Too Short in Mapping CTL Epitopes?” Trends in Immunology, vol. 27, no. 1, 2006, pp. 11–16., doi:10.1016/j.it.2005.11.001.

(5) Kuksa, Pavel P., et al. “High-Order Neural Networks and Kernel Methods for Peptide-MHC Binding Prediction.” Bioinformatics, 2015, doi:10.1093/bioinformatics/btv371.

(6) O’Donnell, Timothy J. et al. “MHCFlurry: Open-Source Class I MHC Binding Affinity Prediction.” Cell System, 2018. 7, 129-132.

(7) Mikolov, Tomas. “Efficient Estimation of Word Representations in Vector Space.” 2013, doi:arXiv:1301.3781v3.

(8) Asgari, Ehsaneddin A., et al. “Continuous Distributed Representations of Biological Sequences for Deep Proteomics and Genomics.” PLoS ONE 10(11), 2015, doi:10.1371/journal.pone.0141287

(9) Phloyphisut, Poomarin, et al. “MHCSeqNet: a Deep Neural Network Model for Universal MHC Binding Prediction.” BMC Bioinformatics, vol. 20, no. 1, 2019, doi:10.1186/s12859-019-2892-4.

(10) “HLA-C Gene - Genetics Home Reference - NIH.” *U.S. National Library of Medicine*, National Institutes of Health, ghr.nlm.nih.gov/gene/HLA-C.

5. Additional Figures

Additional figures 1 and 2 (including captions) are obtained directly from *Bepler et al., 2019.* (3). Additional figure 3 is obtained from Phloyphisut et al., 2019 (9).

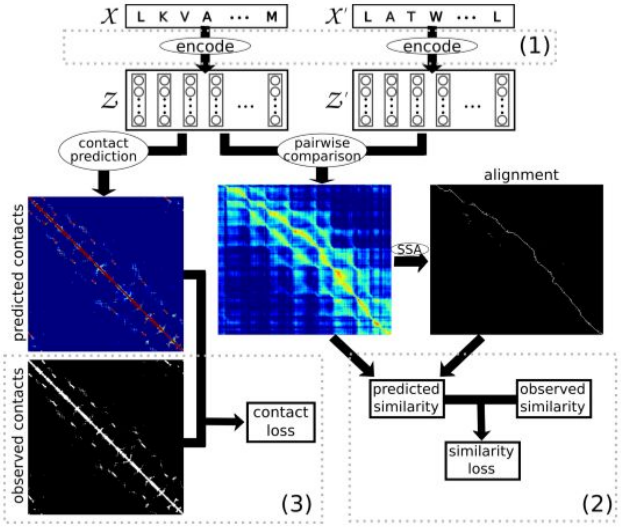

Figure 1: Diagram of the learning framework. **(1)** Amino acid sequences are transformed into sequences of vector embeddings by the encoder model. **(2)** The similarity prediction module takes pairs of proteins represented by their sequences of vector embeddings and predicts their shared SCOP level. Sequences are first aligned based on L1 distance between their vector embeddings using SSA. From the alignment, a similarity score is calculated and related to shared SCOP levels by ordinal regression. **(3)** The contact prediction module uses the sequence of vector embeddings to predict contacts between amino acid positions within each protein. The contact loss is calculated by comparing these predictions with contacts observed in the 3D structure of the protein. Error signal from both tasks is used to fit the parameters of the encoder.

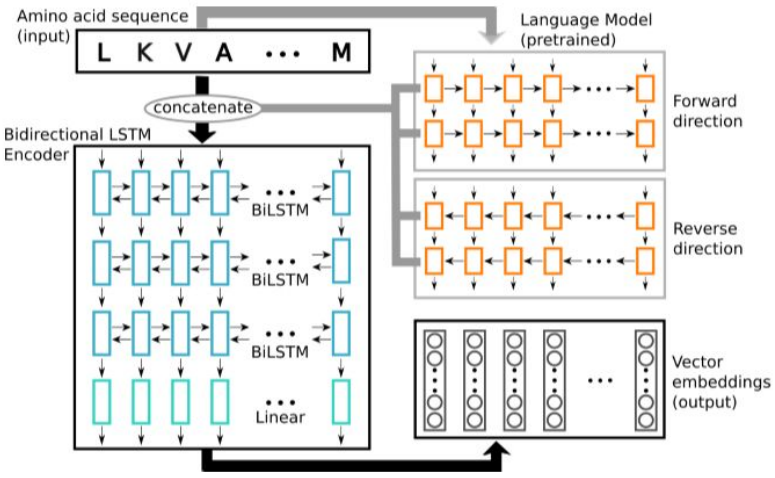

Figure 2: Illustration of the embedding model. The amino acid sequence is first passed through the pretrained language model in both forward and reverse directions. The hidden states at each position of both directions of the language model are concatenated together with a one hot representation of the amino acids and passed as input to the encoder. The final vector representations of each position of the amino acid sequence are given by a linear transformation of the outputs of the final bidirectional LSTM layer.

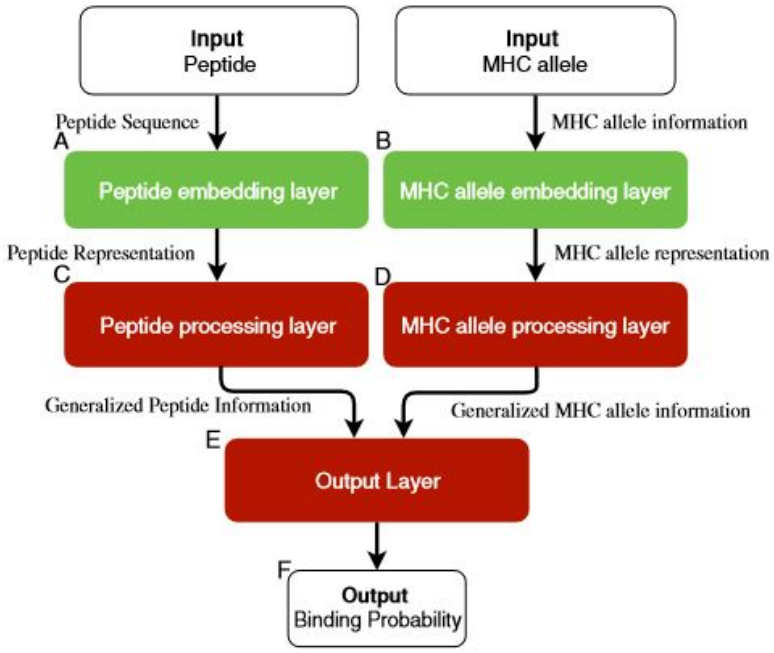

Figure 3: General architectural overview of MHCSeqNet, containing peptide and allele embedding layers, peptide and allele processing layers (GRU for peptide and either GRU or feedforward for allele), and a 3-layer feedforward output layer which outputs a predicted binding affinity probability for a given peptide and allele.