Interpreting BERT architecture predictions for peptide presentation by MHC class I proteins

Anonymous Author(s)
Affiliation
Address
email

Abstract

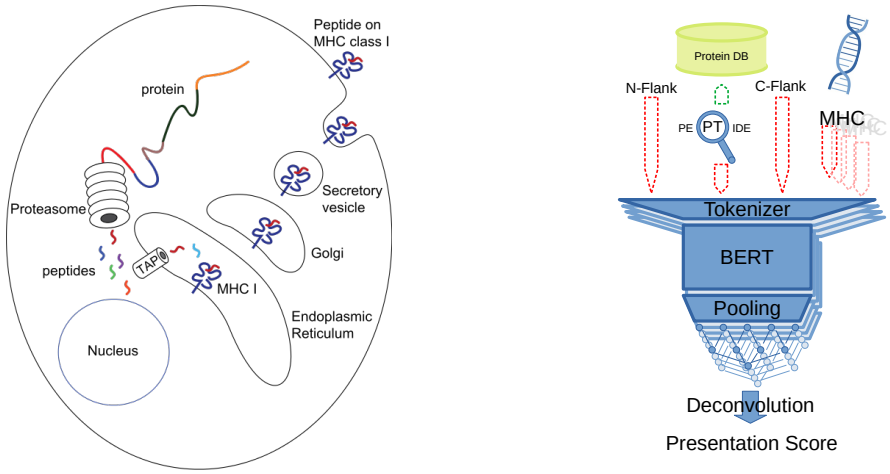
The major histocompatibility complex (MHC) class-I pathway supports the detection of cancer and viruses by the immune system. It presents parts of proteins (peptides) from inside a cell on its membrane surface enabling visiting immune cells that detect non-self peptides to terminate the cell. The ability to predict whether a peptide will get presented on MHC Class I molecules helps in designing vaccines so they can activate the immune system to destroy the invading disease protein. We designed a prediction model using a BERT-based architecture (ImmunoBERT) that takes as input a peptide and its surrounding regions (N and C-terminals) along with a set of MHC class I (MHC-I) molecules. We present a novel application of well known interpretability techniques, SHAP and LIME, to this domain and we use these results along with 3D structure visualizations and amino acid frequencies to understand and identify the most influential parts of the input amino acid sequences contributing to the output. In particular, we find that amino acids close to the peptides' N- and C-terminals are highly relevant. Additionally, some positions within the MHC proteins (in particular in the A, B and F pockets) are often assigned a high importance ranking - which confirms biological studies and the distances in the structure visualizations.

1 Introduction

The immune system defends us from a broad range of threats, some of which are expressed from inside the body's own cells. For example cancer is a disease of the genome, arising from aberrations that accumulate over many years. Also, viruses utilize the cell's gene expression system for their own reproduction and spreading. Cytotoxic T-lymphocytes (CTL), a special kind of T-cells, can detect affected cells and terminate them. For them to 'look inside' cells, a system revolving around MHC proteins - also called Human Leukocyte Antigen (HLA) in humans - has evolved [1]. We focus on MHC-I proteins and their antigen presentation pathway which is active in the nucleated cells of the human body [2]. Proteins present in these cells are constantly being fragmented into smaller pieces (peptides) by proteasomes (see Figure 1a). These then bind to MHC proteins forming peptide:MHC protein complexes (pMHCs) and are eventually presented to the outside world on the cell's surface. These pMHC are antigens for the T-cell receptors (TCRs). [3]

An infection by a virus or a cancer causing mutation, can both result in the production of proteins that would not be present in a healthy cell. Eventually this leads to the presentation of neo-antigens (non-self pMHC) to the outside world [4, 5]. Dependent on whether the CTL consider the peptides presented to them as self or non-self, they decide to terminate the cell. Peptide-based vaccines are a tool that can be used to strengthen the immune response. Identifying which non-self peptides will most likely be presented by a cancerous cell or virus and elicit an immune response (immunogenic peptides [6]) is an important component in their development [7]. This process can be divided into two phases – (1) Identification of peptides that are likely to be presented by the MHC protein

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) MHC-I pathway [8]

(b) ImmunoBERT for peptide presentation on MHC-I

Figure 1: MHC-I pathway and our prediction model for peptide presentation

which is the focus of this paper, (2) Predicting activation of the immune system by the presented peptides (outside the paper's scope). Deep learning models have been used for Phase 1 that identifies presented peptides on MHC molecules. Cheng et al. used a bi-directional transformer architecture to predict peptide presentation by a related molecule - MHC class II (MHC-II). We adapted this work for the MHC-I presentation prediction and find our model, ImmunoBERT, provides comparable performance to popular state of the art (SOTA) models, NetMHCpan [10] and MHCflurry [11] (see Appendix Subsection A.7). ImmunoBERT, as seen in Figure 1b, takes a peptide, its surrounding regions (N-flank and C-flank) and MHC-I molecules as inputs and provides as output a score that represents the likelihood of the peptide being presented on MHC-I molecules.

In this paper, we focus on interpreting the results from our deep learning model. We apply Local Interpretable Model-agnostic Explanations (LIME) [12] and SHapley Additive exPlanations (SHAP) [13] to find the parts of the peptide, MHC protein and surrounding flanks of the peptide, that are particularly relevant for presentation. We corroborate the results of LIME and SHAP using visualizations that present 3D peptide-MHC-II molecule interactions and a motif (short, recurring patterns [14]) analysis that confirms that our model has indeed learnt MHC allele dependent peptide presentations.

Section 2 below introduces the necessary biological concepts, while Section 3 presents the current SOTA approaches. In Section 4, we introduce the datasets and our ImmunoBERT model. Finally, we present model interpretation and visualization in Section 5.

2 Background

Proteins play a pivotal role in the human body. They consist of amino acids (AAs) chained together via peptide bonds into a polypeptide or peptide for short [15]. There are 20 naturally occurring amino acids and each contains a specific side chain that gives it unique properties to interact with the environment [16, 15]. The end of the chain with an exposed amino group is called N-terminus, the end with an exposed carboxyl group is called C-terminus. Each of them can be represented by a letter. Once joined together, we refer to the single AAs as residues. Short residue chains are referred to as peptides, while longer ones are referred to as polypeptides. The full-length polypeptide encoded by a gene in the genome is called a protein. Proteins typically adopt a 3D molecule structure due to the chemical properties of their AAs.

Proteins perform diverse tasks like breaking up nutrition, muscle movement and sustaining cell structure. Owing to their wide range of functionality, and for protection against foreign invading proteins from viruses and cancers, controlling which proteins are present in a cell is important. The MHC-I pathway achieves this (see Figure 1a) [17] with the following steps:

1. Proteasomes constantly fragment the cell's internal proteins into peptides (mostly 8-12 AAs long [18]). For example, part of the AA sequence of the protein Albumin is MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIAFAQYLQQCP.. and the proteasomes could fragment this into peptides – "VTFISLLFL", "VAHRFKDLG" and "FAQYLQQ". Each of these peptides are associated with a N-flank and a C-flank. For example, the peptide "VAHRFKDLG" would have a 15 AA N-flank of "AYSRGVFRRDAHKSE" and a 15 AA C-flank of "EENFKALVLIAFAQY".
2. Transporter associated with antigen processing (TAP) proteins transport these peptides into the endoplasmic reticulum (ER) where they bind to MHC-I proteins, forming pMHC with the peptides.
3. The pMHCs gets transported to the cell membrane, where the MHC protein acts as a pedestal for the peptide and presents it to the extracellular environment
4. Finally, A CTL with a fitting TCR could bind a presented neo-antigen. This might trigger an immune reaction. CTLs do not strongly bind to the body's own (self) peptides but only non-self ones.

We focus on steps 1-3 in this paper that form Phase 1 of identifying non-self peptides by the immune system. There are many different HLA alleles [15, 3] in the human population with three main loci coding for MHC-I proteins: HLA-A, HLA-B and HLA-C [2], with every human expressing up to six different HLA proteins. As the HLA alleles can have different binding properties there is a large variety in immunopeptidomes (entirety of all presented peptides) across humanity. Currently, more than 4,064 HLA-A, 4,962 HLA-B and 3,831 HLA-C proteins are known [19]. Each can bind roughly 1,000 to 10,000 different peptides [20].

Predicting the immunopeptidome of a particular individual is challenging owing to two reasons. First, any of the six different MHC-I alleles present in a cell might be responsible for a peptide observation in an eluted ligand (EL) experiment (see Appendix Subsection A.1). So the observations need to be deconvoluted (assigned to a HLA allele). Second, the high throughput eluted ligand assays only produce positive examples resulting in a highly imbalanced dataset that requires the creation of artificial negative ones (decoys). This is exacerbated by the fact that available labelled data accounts for a very small fraction of the peptides that can/cannot be presented. Absence of ground truth, variance among individuals, limited labeled data makes prediction modeling in the field of immunopeptidomics extremely challenging.

3 Related Work

Below, we take a brief look at the two most popular state of the art prediction models for peptide presentation on MHC-I molecules.

NetMHCpan: The most commonly used model today is NetMHCpan. It has a long history and is currently in version 4.1 which is an ensemble of 50 single hidden layer feed forward neural networks [10]. The MHC allele is input into the model as a pseudo sequence consisting of only 34 AAs. These were identified by [21] as being particularly close to the presented peptide and relevant for peptide binding.

To deconvolute multi-allele (MA) data, NetMHCpan uses the NNAlign_MA [22] framework. First, only single-allele (SA) data (the observation can be unambigiously linked to a single MHC protein) is used to train a classifier (takes as input a peptide and a single MHC allele). In the deconvolution step, each observation that could be caused by multiple MHC alleles, is deconvolved separately. To do so, the classifier trained in the previous step is used to predict the likelihood of each potential peptide:MHC protein combination independently. The MHC allele showing the highest scaled prediction is chosen and used as the MHC protein responsible for the observation until the next deconvolution step. In case of a negative example, a MHC allele is picked at random.

MHCflurry 2.0 [11] explicitly models the process of MHC binding separately from the others (e.g. proteasomal cleavage). This results in a natural integration of binding affinity (BA) and EL data (see Appendix Subsection A.1). There are three sub-models. First, MHCflurry BA models the process of the peptide binding to a MHC protein. Second, MHCflurry AP models the remaining antigen processing steps, like proteasomal cleavage and TAP transportation. Finally, MHCflurry PS combines the output of those two models to predict peptide presentation. O'Donnell, Rubinsteyn, and Laserson [11] benchmarked their performance on held-out MS data against NetMHCpan 4.0 and MixMHCpred

2.0.2. They found their model had better performance (with regards to their chosen metric - positive predictive value). We use this benchmark dataset for comparing our model, ImmunoBERT, against SOTA models, NetMHCpan and MHCflurry 2.0 (see Appendix Subsection A.7).

To our knowledge, there is no existing work on interpreting deep learning models for peptide presentation with MHC molecules. However, several interpretation techniques for deep learning models have been recently developed in the field of computer vision - popular ones include LIME [12] and SHAP [13]. LIME produces local (for a particular example) explanations, treating the model to be explained as a black-box (model-agnostic). Given a particular input, LIME samples the neighborhood of this input and creates a linear model to approximate the model's local behavior. In comparison, SHAP values are based on the idea of Shapley values, that attribute the difference between the average prediction over the dataset and the example's prediction fairly to the various features. Lundberg and Lee [13] developed the SHAP package for the efficient approximation of SHAP values under the assumption that the model's output restricted to a subset of the features is given by the expected model prediction conditioned on this subset. We apply these two techniques to the problem of peptide presentation and combine it with visualizations and anlysis accessible to biologists and clinicians.

4 Method

In this section, we present the peptide and MHC-I data used to train and validate ImmunoBERT, the model architecture and interpretation techniques. Source code for our model and interpretation can be found at `https://anonymous.4open.science/r/ImmunoBERT-E6DF`.

4.1 Data

We combined data from two sources. The first one consists of a collection of peptides from EL assays mapped to the GRCh38 *Homo sapiens* reference genome and proteins within the Ensembl v94 database. The source of this data is studies included in the PRoteomics IDEntifications Database (PRIDE) [23]. We removed samples without linked HLA proteins or where the peptide is not present in the human proteome (neo-antigens). The second data source is the HLA Ligand Atlas [24]. This includes tissue and HLA allele specific ligands from EL experiments. In contrast to the first data source, the HLA Ligand Atlas maps peptides to the Uniprot proteome, some of which map to GRCh38. Similar to other studies, we consider peptides of sequence lengths between 7 and 15 amino acids [**odonnell_mhcflurry_202**, 10].

A sample represents the experiment carried out on a particular cell-line/individual to measure the presented peptides. It is, therefore, linked with up to 6 different HLA alleles and the observations (also referred to as *hits*) of peptides during the experiment. Each peptide was mapped to proteins in the human genome. In total we had 293k SA and 1,666k MA observations of 430k unique peptides. These were observed in 469 samples linked with 109 different MHC alleles.

Decoy generation: Negative example (decoy) generation is particularly important due to the imbalanced nature of the dataset. We follow a similar procedure to the MHCflurry benchmark dataset[11]. A decoy is associated with a single hit. To match the observations' length distribution, the decoy peptides have the same length as their associated hit. To generate a decoy peptide we randomly select a position within all proteins of the hit's sample as the start of the decoy peptide. Implicitly we take the absence of a peptide's observation as evidence for it actually not being presented. We considered using 19 or 99 decoys per hit in our hyper-parameter search.

Data splits: Splitting the data into train, test and validation set is not trivial, as we assessed generalization along 2 dimensions - to unseen MHC alleles and unseen proteins. Also, each observation can be associated with up to six MHC alleles from which at least one is responsible for the presentation. There are also many homologues (areas having a common ancestor) in the human genome and, ideally, a group of homologues would not span different splits. Our methodology for splitting the data can be found in Appendix Subsection A.2. We get one training set, two validation sets and two test sets. The data partition can also be found in the Appendix.

4.2 Model Architecture and Training

We use the pre-trained Tasks Assessing Protein Embeddings (TAPE) transformer as backbone for our model (a BERT architecture adapted to amino acid sequences). As input we provide our model with the peptide's AA sequence. If uniquely identified, we also input the peptide's context which consists of (1) up to 15 AAs that occur to the *left* of the peptide in its source protein sequence known as *N-flank*, and (2) up to 15 AAs that occur to the *right* referred to as *C-flank*. Finally, the model also receives the MHC pseudo sequence as defined by NetMHCpan (see Subsection 3).

Off the shelf, the TAPE transformer supports only a single token type id. To make it easy to distinguish between the various input parts, we use a novel representation of the input and extend the TAPE model's token type embedding matrix to four different token types - one for the N-flank, peptide, C-flank and MHC protein. The resulting embedding vectors are fed through the TAPE encoder (12 self attention layers with 12 heads each). In contrast to BERTMHC [9] that then uses average pooling with the embedded vectors, we explored three options as part of the hyper-parameter search: averaging, attention layer or the classification token's vector. This was done to consider the meaning of the peptide, context and MHC sequence. After comparison of the three options, we chose the classification token's vector that was best performing.

The structure of our model's head is similar to the one used by BERTMHC [9]. It is also a multi layer perceptron (MLP) consisting of two fully connected layers with a hidden dimension of 512. There is a single output neuron, with sigmoid activation for the presentation score. Our model's training procedure is inspired by the NNAlign_MA framework and general results from Multi Instance Learning (MIL). In the first training epoch we only use SA data. This is followed by a deconvolution phase. During this, we deconvolve each MA observation by at first predicting the presentation score for each potentially responsible allele and then selecting the one with the highest score as the relevant allele. This means, the MA example is treated (forward-propagation and backward-propagation) as if coming from the relevant allele until the next deconvolution phase (happens after each epoch). The decoy's relevant MHC allele is the same as the observation.

For training we use standard binary cross entropy as loss function. We trained the full network (including encoder and embedding layers) using the ADAM optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$ [25]). We chose the initial learning rate as part of the hyper-parameter search.

4.2.1 Interpretation

To check, whether our model has learned relationships grounded in biology, we employ LIME [12] and SHAP [13]. We restrict our analysis to 9-mer peptides and SA data. As we interpret test set data, the analysis demonstrates our model's ability to generalize to unseen MHC proteins. We additionally use Sequence Motifs and 3D visualization of the MHC and peptide structures to better understand the model results and the interpretations from LIME and SHAP.

LIME analysis of all features: We first interpret the model output by assessing the importance of all AA features in the input peptide, context and HLA allele sequences. For this, we use the LIME framework. This is in general faster than SHAP - in particular when handling numerous features (in our case 73 AA positions). We adapted the text version of LIME and implement the deactivation of a feature by setting the input mask token to zero. We use the standard cosine distance metric (in binary space) between the original example and the sampled examples. For each test set HLA allele, we selected a random set of 500 observations with each of these observations accompanied by a decoy bringing the total to 1000 examples in the test set that we aim to interpret. Each example output gets explained by sampling from 2000 feature combinations. Figures 2, 5 and 16 then show in each bar the proportion of examples with a given importance-ranking for an AA at a certain position. If for example the bright red bar (1st) of peptide position 9 showed 0.5, this means that 50% of samples in the test set had peptide position 9 as the most important feature.

SHAP analysis of peptide positions: Finally, we examine the average contribution of peptide AAs using SHAP. Similar to the LIME setup, the interpretation of each HLA allele uses 500 9-mer single allele hits and 500 decoys. We sample 250 sequences as background distribution. The nine position features in the peptide would result in a maximum of 512 feature subsets. We carry out the Kernel SHAP analysis using a sample of 64 of these. For the whole process we ignore the flanks. In Figures 4b, 7b and **??** we plot the average SHAP value for each AA at each peptide position.

Sequence Motifs: We visualize for each HLA allele, the frequency of various amino acids at presented peptide positions [26] and contrast this with the feature importance rankings generated by LIME and SHAP. We use two different motifs to understand the results from ImmunoBERT -

1. *Model Motif:* we generate 100,000 random 9-mer peptides from the human proteome (from Uniprot and Ensembl proteins, as well as their context). Then we predict for each of those examples the presentation score for the HLA protein concerned. We select the ones with a presentation score > 0.5 and use them to create the HLA allele dependent *model motif* (using the logomaker [27] package). Our approach for creating model motifs is similar to Wu et al. [28] who use the 2% highest scoring peptides for model motif.

2. *Data Motif:* we take the data from our test set and use all of the 9-mer peptides presented by the HLA allele to create it.

The data motif shows the frequency of AAs at presented peptide positions (independent of our model) based on existing labelled data on peptide presentation within a sample. The model motif, on the other hand, shows the frequency of AAs at presented peptide positions predicted by our model - sampled for peptides across the genome. We do not expect both motifs to be the same but if the test set for any given HLA allele were a representative sample from the human proteome, we expect there will be overlap between the data motif and the model motif.

In a motif logo, for each peptide position a stack of AA letters is displayed. The size of each letter is proportional to the AA's frequency at this position. More frequent AAs can be found on top. Each stack is then scaled with the position's information content (IC), resulting in a bit representation [26]. The lower the position's entropy, the higher the IC and, so, the logo. We do not display positions with IC < 0.5 to avoid distraction. AA with similar chemistry are coloured the same.

PyMOL visualizations: PyMOL [29], a cross-platform molecular graphics tool, has been widely used for three-dimensional (3D) visualization of proteins, nucleic acids, small molecules, electron densities, surfaces, and trajectories. The 3D visualizations presented in this paper are based on the MHC 3D structures in [30]. We colored them with their mean importance ranking - AA that were not used as features are blue, in contrast, the highest important AA are colored red. Visualizing the importance of various amino-acids on a structure of peptide bound MHC can help to shed light on how importance correlates with physical interactions known to be important.

5 Results

In this chapter, we present the results from applying the interpretability techniques described in 4.2.1 to the ImmunoBERT model, limited to peptide presentation by two test set HLA alleles. There are many more test HLA alleles in our data, but we are unable to fit all their analyses and visualizations within the defined page limit. Figures for all the remaining test HLA alleles can be found in the Appendix. It is worth noting that to improve readability all SHAP values were multiplied by 100.

Visualizing the importance of various amino-acids on a 3D structure of peptide bound MHC can help to shed light on how importance correlates with physical interactions known to be important. The Figures 3 and 6 demonstrate that regions near the anchor residues on the MHC molecule are important both within the peptide and the MHC molecule. This re-affirms known biology of antigen binding to MHC and indicates that LIME and SHAP provide reasonable results.

5.1 HLA-A*33:01

LIME Analysis and 3D structure visualization: Figure 2 shows for each input position, the proportion of examples in which this position had a certain value range of importance ranking (see Subsection 4.2.1). The 9-mer peptide's AAs tend to be much more highly ranked than its flanking regions or the AA in the HLA protein. Peptide position 9 is ranked first in roughly half of the examples. The HLA pseudo sequence positions display a high variance in their ranking distributions. Some positions tend to be particularly highly ranked - like 63, 73, 77, 97, 116 and 171. HLA positions 63, 171 are located in the HLA's A and 77, 116 are located in its F pocket [31] which both are supposed to house a peptide terminus [32]. This might explain what we see in the motifs of Figure 4a. A peptide C-terminus arginine (R) is very common and also the N-terminus shows preference for certain AAs.

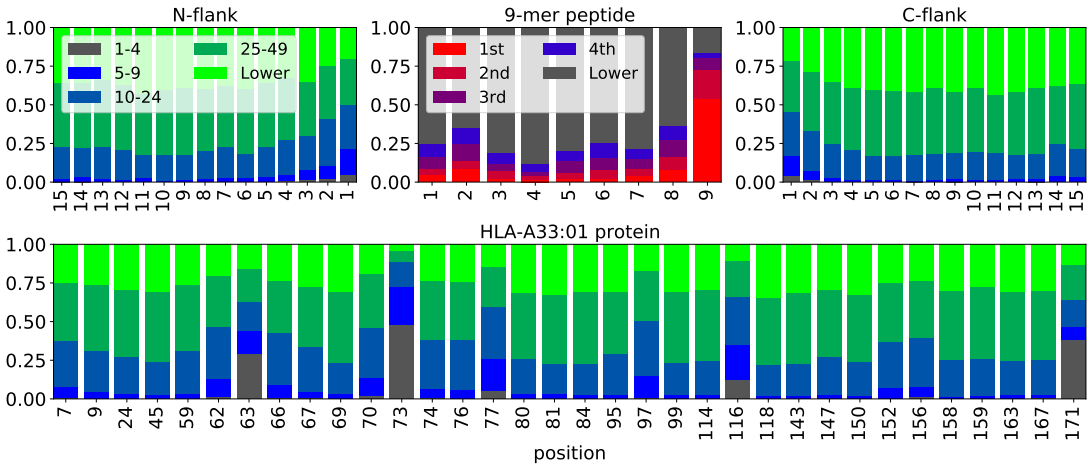
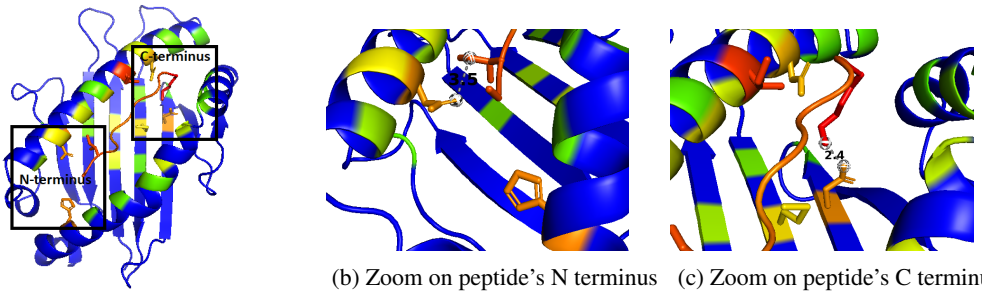


Figure 2: LIME feature importance rank distribution for HLA-A*33:01



(a) Overall perspective

(b) Zoom on peptide's N terminus

(c) Zoom on peptide's C terminus

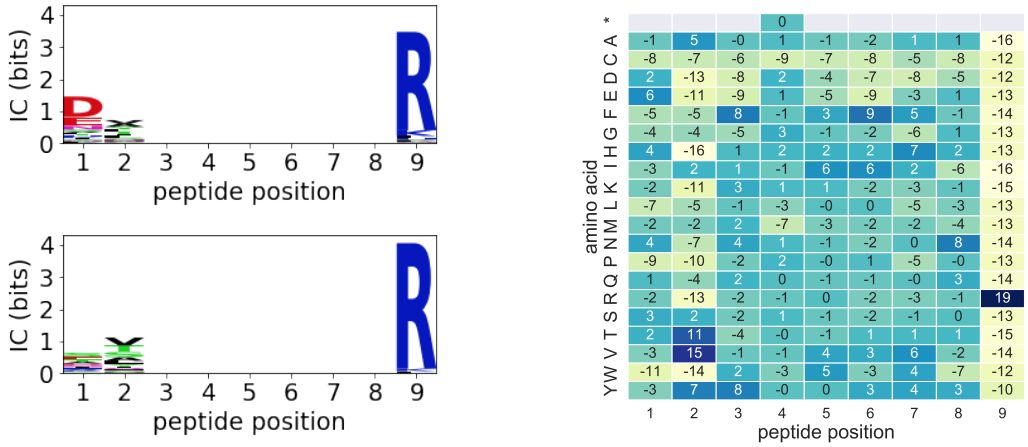
Figure 3: PyMOL visualizations of the HLA-A*33:01protein and the peptide

The significance of HLA positions 73 (in pocket C) and 63 (in pocket B) is less clear. However, in Figure 3 we display the residues of HLA positions 63, 73, 77, 97, 116 and 171 as well as peptide positions 1, 2 and 9. We see, that they can all be found close to the peptide's termini - which aligns with the terminis' high importance values assigned by LIME. We also observe, that the distance between HLA position 116 and peptide position 9 is only 2.4 Angstrom in Figure 6c and the distance between HLA position 63 and peptide position 2 is only 3.5 Angstrom in Figure 6b. The visualizations clearly show that these high importance features are physically close to each other in 3D space helping us understand how the peptides are presented on MHC-I molecules.

The peptide flanks are the the least important in Figure 2. This is similar to observations in [11] where they found that including the peptide flanks resulted in a small but consistent improvement in prediction. Within the flanks, our model attributes most importance to positions closest to the 9-mer peptide.

Sequence Motifs: Figure 4a shows the data and model motifs for HLA-A*33:01. We remind the reader that data motif at the top of Figure 4a is generated from 100K random 9-mer peptides from the human proteome that are presented (have a presentation score > 0.5) with the given HLA allele. Model motif at the bottom of Figure 4a is generated from 9-mer peptides in our test set that are presented with the given HLA allele. We find both motifs show a high frequency for R in position 9. AAs frequencies in positions 1 and 2 do not match exactly. This might be because they were generated from different background distributions - while the model's motif is based on peptides samples from the whole human genome, the data motif is only based on the proteins actually expressed in the samples of the test HLA allele.

SHAP Values: With regards to SHAP values, Figure 4b left side shows the mean SHAP value of each AA at each peptide position. We see the strong positive mean contribution of R at peptide position 9, making its high frequency in Figures 4a plausible. It also shows high values for V and T at position 2 which are also enriched in the model motif.



(a) Data(top) & model motif

(b) Mean SHAP values

Figure 4: Motifs and SHAP values for peptide samples presentation on HLA-A*33:01

5.2 HLA-B*54:01

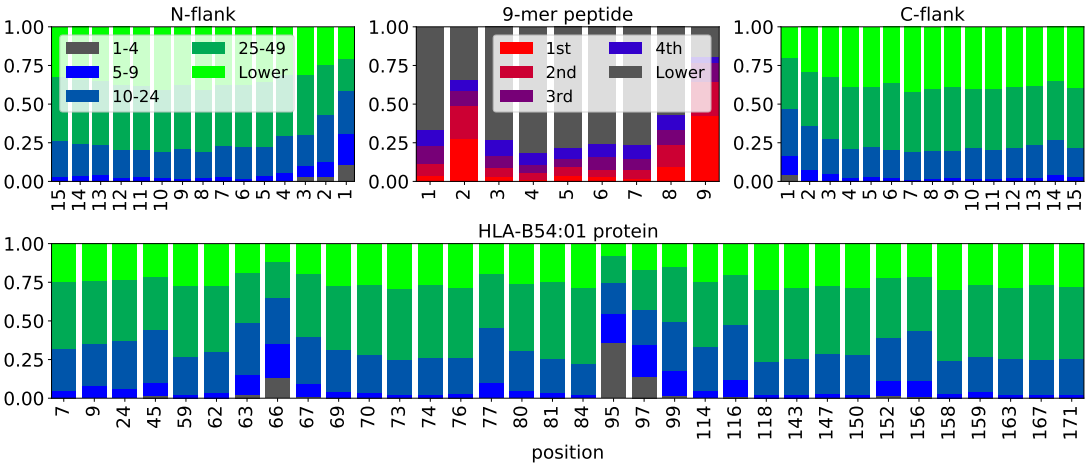


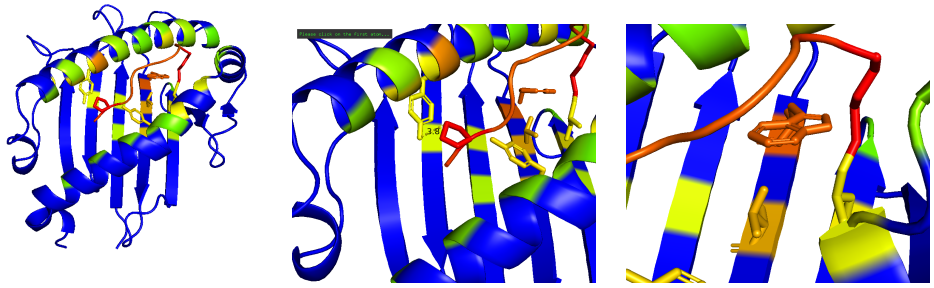
Figure 5: LIME feature importance rank distribution for HLA-B*54:01

Feature importance ranking generated by LIME in Figure 5 for peptide presentation with HLA-B*54:01is similar to that observed in Figure 2. We find again the 9-mer peptide is the most important element followed by the HLA allele. Peptide flanks are the least important. We find two positions in the 9-mer peptide samples to be particularly important for presentation on the HLA-B*54:01, namely positions 2 and 9. This corresponds to what we see in the motifs in Figure 7a and the SHAP values in Figure 7b.

With regards to the HLA protein, its most important position is 95. This is located in the F pocket (houses peptide terminus) [31]. This makes sense given the high SHAP values of peptide position 9. Its second most important position is 66 in pocket B. Given the enrichment of peptide position 2 by P and A we speculate that pocket B houses this peptide position. A similar finding was reported in [33].

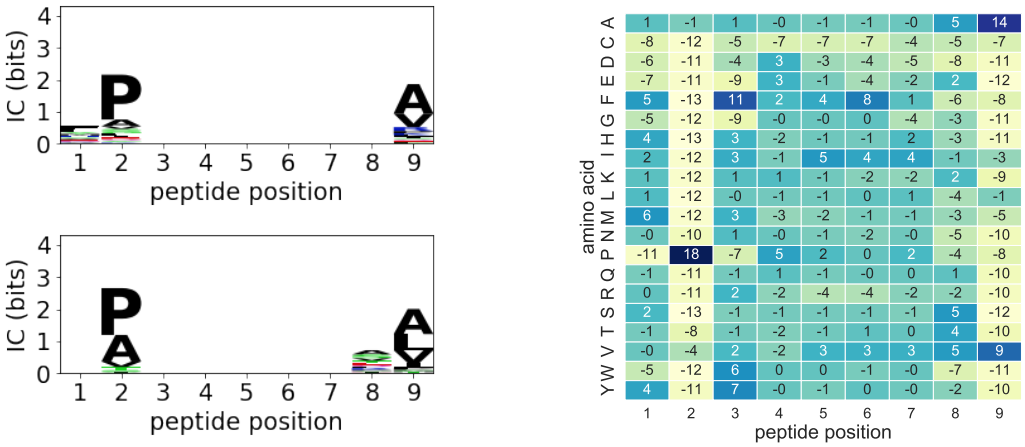
6 Conclusion

We applied interpretability techniques LIME and SHAP to find that our model learned biologically meaningful importance rankings and feature contributions. We confirmed the interpretations using



(a) Overall perspective (b) Zoom on peptide's N terminus (c) Zoom on peptide's C terminus

Figure 6: PyMOL visualizations of the HLA-B*54:01protein and the peptide



(a) Data (top) & model motif (b) Mean SHAP values

Figure 7: Motifs and SHAP values for peptide samples presentation on HLA-B*54:01

3D structure visualizations and sequence motifs of peptides and MHC molecules. We found across HLA alleles, high importance for peptide presentation was given to AAs near the N- and C-termini of the peptide and varying MHC positions in the A, B and F pockets. In contrast, the peptide flanks showed less importance, which explains why [11] found that including them only results in a small but consistent model improvement. The motifs we found using our model followed broadly those observed in the data. As these analysis were all carried out on held out MHC proteins, it also demonstrates the generalization ability of our model.

We have only scratched the surface in attempting to understand peptide presentation with MHC-I molecules by interpreting the results from deep learning models. Visualizations, sequence motifs and interpretability techniques like LIME go hand in hand in helping both computer scientists and biologists understand the application of deep learning models for peptide presentation and the underlying biology. Given the interdisciplinary nature of this field, it is important that future deep learning models that provide accurate predictions in this field are accompanied with explanations and visualisations accessible to biologists and clinicians.

7 Ethics considerations

We expect our research to contribute to an improved understanding of the MHC-I pathway which will enable better customized therapies also for patients with less common MHC-I alleles. The datasets we worked with did not include any sensitive personally identifiable information. It, however, does not represent the diversity of the global human population. This is exactly the reason, why it is important to develop models that can extrapolate to unseen MHC proteins.

Acronyms

AA amino acid. 2, 3, 5–7, 9, 16

AP average precision. 15–17

AUC area-under-the-curve. 15–17

BA binding affinity. 3, 14

CTL Cytotoxic T-lymphocytes. 1, 3

EL eluted ligand. 3, 4, 14, 16

ER endoplasmic reticulum. 3

HLA Human Leukocyte Antigen. 1, 3–8, 14, 16, 17

IC information content. 6

LIME Local Interpretable Model-agnostic Explanations. 1, 2, 4, 5, 7, 8, 16, 18–22

MA multi-allele. 3–5

MHC major histocompatibility complex. 1–6, 9, 14, 15

MHC-I MHC class I. 1–3, 9, 13

MHC-II MHC class II. 2

MIL Multi Instance Learning. 5

MLP multi layer perceptron. 5

pMHC peptide:MHC protein complex. 1, 3

PR precision-recall. 17

PRIDE PRoteomics IDEntifications Database. 4

ROC receiver-operating-curve. 15–17

SA single-allele. 3–5, 15

SHAP SHapley Additive exPlanations. 1, 2, 4–9, 16, 18–22

SOTA state of the art. 2, 4

TAP transporter associated with antigen processing. 3

TAPE Tasks Assessing Protein Embeddings. 5

TCR T-cell receptor. 1, 3

References

[1] Encyclopaedia Britannica. *Major histocompatibility complex*. July 2021. URL: https://www.britannica.com/science/major-histocompatibility-complex.

[2] Y. M. Mosaad. "Clinical Role of Human Leukocyte Antigen in Health and Disease". In: *Scandinavian Journal of Immunology* 82.4 (2015), pp. 283–306. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/sji.12329 (visited on 07/13/2021).

[3] Kenneth Murphy and Casey Weaver. *Janeway's Immunobiology*. 9th. Garland Science, 2017.

[4] Ton N. Schumacher and Robert D. Schreiber. "Neoantigens in cancer immunotherapy". In: *Science* 348.6230 (Apr. 2015). Publisher: American Association for the Advancement of Science Section: Review, pp. 69–74. URL: https://science.sciencemag.org/content/348/6230/69 (visited on 06/15/2021).

[5] Zheying Zhang et al. "Neoantigen: A New Breakthrough in Tumor Immunotherapy". In: *Frontiers in Immunology* 0 (2021). Publisher: Frontiers. URL: https://www.frontiersin.org/articles/10.3389/fimmu.2021.672356/full (visited on 08/11/2021).

[6] Guangyuan Li et al. "DeepImmuno: deep learning-empowered prediction and generation of immunogenic peptides for T-cell immunity". In: *Briefings in Bioinformatics* (May 2021). URL: https://doi.org/10.1093/bib/bbab160 (visited on 06/09/2021).

[7] Joseph D. Comber and Ramila Philip. "MHC class I antigen presentation and implications for developing a new generation of therapeutic vaccines". In: *Therapeutic Advances in Vaccines* 2.3 (May 2014). Publisher: SAGE Publications Ltd STM, pp. 77–89. URL: https://doi.org/10.1177/2051013614525375 (visited on 04/21/2021).

[8] Wikipedia. *Major histocompatibility complex*. Page Version ID: 1020006475. Apr. 2021. URL: https://en.wikipedia.org/w/index.php?title=Major_histocompatibility_complex&oldid=1020006475 (visited on 05/04/2021).

[9] Jun Cheng et al. "BERTMHC: Improves MHC-peptide class II interaction prediction with transformer and multiple instance learning". In: *bioRxiv* (Nov. 2020). Publisher: Cold Spring Harbor Laboratory Section: New Results, p. 2020.11.24.396101. URL: https://www.biorxiv.org/content/10.1101/2020.11.24.396101v1 (visited on 04/05/2021).

[10] Birkir Reynisson et al. "NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data". In: *Nucleic Acids Research* 48.W1 (July 2020), W449–W454. URL: https://doi.org/10.1093/nar/gkaa379 (visited on 04/05/2021).

[11] Timothy J. O'Donnell, Alex Rubinsteyn, and Uri Laserson. "MHCflurry 2.0: Improved Pan-Allele Prediction of MHC Class I-Presented Peptides by Incorporating Antigen Processing". In: *Cell Systems* 11.1 (July 2020). Publisher: Cell Press, 42–48.e7. URL: https://www.sciencedirect.com/science/article/pii/S2405471220302398 (visited on 04/05/2021).

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 1135–1144. URL: https://doi.org/10.1145/2939672.2939778 (visited on 08/15/2021).

[13] Scott M. Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 4768–4777. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (visited on 06/18/2021).

[14] Patrik D'haeseleer. "What are DNA sequence motifs?" In: *Nature Biotechnology* 24.4 (Apr. 2006). Bandiera_abtest: a Cg_type: Nature Research Journals Number: 4 Primary_atype: Reviews Publisher: Nature Publishing Group, pp. 423–425. URL: https://www.nature.com/articles/nbt0406-423 (visited on 07/26/2021).

[15] Thomas D. Pollard et al. *Cell Biology*. 3rd ed. Elsevier Health Sciences, Nov. 2016.

[16] Jonathan Crowe and Tony Bradshaw. *Chemistry for the Biosciences - The essential concepts*. 3rd. Oxford University Press, 2014.

[17] Kenneth L. Rock, Eric Reits, and Jacques Neefjes. "Present Yourself! By MHC Class I and MHC Class II Molecules". In: *Trends in Immunology* 37.11 (Nov. 2016), pp. 724–737. URL: https://www.sciencedirect.com/science/article/pii/S1471490616301004 (visited on 04/05/2021).

[18] Morten Nielsen et al. "Immunoinformatics: Predicting Peptide–MHC Binding". In: *Annual Review of Biomedical Data Science* 3.1 (2020), pp. 191–215. URL: https://doi.org/10.1146/annurev-biodatasci-021920-100259 (visited on 04/08/2021).

[19] EBI. *Immuno Polymorphism Database - Statistics*. 2021. URL: https://www.ebi.ac.uk/ipd/imgt/hla/stats.html (visited on 04/05/2021).

[20] Jennifer G. Abelin et al. "Mass Spectrometry Profiling of HLA-Associated Peptidomes in Mono-allelic Cells Enables More Accurate Epitope Prediction". In: *Immunity* 46.2 (Feb. 2017). Publisher: Elsevier, pp. 315–326. URL: https://www.cell.com/immunity/abstract/S1074-7613(17)30042-0 (visited on 04/22/2021).

[21] Morten Nielsen et al. "NetMHCpan, a Method for Quantitative Predictions of Peptide Binding to Any HLA-A and -B Locus Protein of Known Sequence". In: *PLOS ONE* 2.8 (Aug. 2007). Publisher: Public Library of Science, e796. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0000796 (visited on 05/08/2021).

[22] Bruno Alvarez et al. "NNAlign_MA; MHC Peptidome Deconvolution for Accurate MHC Binding Motif Characterization and Improved T-cell Epitope Predictions". In: *Molecular & Cellular Proteomics* 18.12 (Dec. 2019). Publisher: Elsevier, pp. 2459–2477. URL: https://www.mcponline.org/article/S1535-9476(20)31649-2/abstract (visited on 04/05/2021).

[23] Yasset Perez-Riverol et al. "The PRIDE database and related tools and resources in 2019: improving support for quantification data". In: *Nucleic Acids Research* 47 (Jan. 2019), pp. D442–D450. URL: https://doi.org/10.1093/nar/gky1106 (visited on 02/19/2021).

[24] Ana Marcu et al. "The HLA Ligand Atlas - A resource of natural HLA ligands presented on benign tissues". In: *bioRxiv* (July 2020). Publisher: Cold Spring Harbor Laboratory Section: New Results. URL: https://www.biorxiv.org/content/10.1101/778944v2 (visited on 04/20/2021).

[25] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv:1412.6980 [cs]* (Jan. 2017). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980 (visited on 10/22/2020).

[26] Thomas D. Schneider and R.Michael Stephens. "Sequence logos: a new way to display consensus sequences". In: *Nucleic Acids Research* (Oct. 1990). URL: https://doi.org/10.1093/nar/18.20.6097 (visited on 07/26/2021).

[27] Ammar Tareen and Justin B. Kinney. "Logomaker: Beautiful sequence logos in python". In: *Bioinformatics* (May 2019). Publisher: Cold Spring Harbor Laboratory Section: New Results. URL: https://www.biorxiv.org/content/10.1101/635029v1 (visited on 08/01/2021).

[28] Jingcheng Wu et al. "DeepHLApan: A Deep Learning Approach for Neoantigen Prediction Considering Both HLA-Peptide Binding and Immunogenicity". In: *Frontiers in Immunology* 10 (2019). URL: https://www.frontiersin.org/articles/10.3389/fimmu.2019.02559/full (visited on 03/14/2021).

[29] Schrödinger. *The PyMOL Molecular Graphics System, Version 2.5*. July 2021. URL: https://pymol.org/2/ (visited on 07/22/2021).

[30] Deylane Menezes Teles e Oliveira et al. "pHLA3D: An online database of predicted three-dimensional structures of HLA molecules". In: *Human Immunology* 80.10 (Oct. 2019), pp. 834–841. URL: https://www.sciencedirect.com/science/article/pii/S0198885919302770 (visited on 05/24/2021).

[31] Hanneke W. M. van Deutekom and Can Keşmir. "Zooming into the binding groove of HLA molecules: which positions and which substitutions change peptide binding most?" In: *Immunogenetics* 67.8 (2015), pp. 425–436. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4498290/ (visited on 07/10/2021).

[32] Anette Stryhn et al. "Longer peptide can be accommodated in the MHC class I binding site by a protrusion mechanism". In: *European Journal of Immunology* 30.11 (2000), pp. 3089–3099. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/1521-4141%28200011%2930%3A11%3C3089%3A%3AAID-IMMU3089%3E3.0.CO%3B2-5 (visited on 07/10/2021).

[33] Jun Liu and George F. Gao. "Major Histocompatibility Complex: Interaction with Peptides". In: *Encyclopedia of Life Sciences*. Wiley-Blackwell, 2011. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470015902.a0000922.pub2 (visited on 08/15/2021).

[34] D. F. Hunt et al. "Characterization of peptides bound to the class I MHC molecule HLA-A2.1 by mass spectrometry". In: *Science* 255.5049 (Mar. 1992). Publisher: American Association for the Advancement of Science Section: Reports, pp. 1261–1263. URL: https://science.sciencemag.org/content/255/5049/1261 (visited on 07/19/2021).

[35] Ensembl. *BioMart - Paralogue Table*. June 2021. URL: http://www.ensembl.org/biomart/martview/48db0485487a9c26d139bc5d7cdda420?VIRTUALSCHEMANAME=default&ATTRIBUTES=hsapiens_gene_ensembl.default.homologs.ensembl_gene_id%7Chsapiens_gene_ensembl.default.homologs.ensembl_transcript_id%7Chsapiens_gene_ensembl.default.homologs.ensembl_peptide_id%7Chsapiens_gene_ensembl.default.homologs.hsapiens_paralog_ensembl_

gene%7Chsapiens_gene_ensembl.default.homologs.hsapiens_paralog_associated_gene_name%7Chsapiens_gene_ensembl.default.homologs.hsapiens_paralog_ensembl_peptide&FILTERS=&VISIBLEPANEL=resultspanel.

[36] G. M. Weiss and F. Provost. "Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction". In: *Journal of Artificial Intelligence Research* 19 (Oct. 2003), pp. 315–354. URL: https://jair.org/index.php/jair/article/view/10346 (visited on 08/15/2021).

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section .
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We described that the work was only done for a small subset of all MHC-I proteins
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] see Section 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] https://anonymous.4open.science/r/ImmunoBERT-E6DF, full dataset not yet published
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see Appendix Subsections A.2 and A.3
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] https://anonymous.4open.science/r/ImmunoBERT-E6DF
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] see Section 7

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Experimental datasources

As the most restrictive step in epitope presentation is MHC binding, measuring the BA between a particular MHC protein and a peptide *in vitro*, can give us some insight into how likely it is that a particular peptide will be presented to the extracellular environment. Early presentation predictors were, therefore, only trained on this BA data. Its biggest disadvantage is, that it is costly to carry out the experiments that then only generate little data [22].

In the modern high throughput EL approach, the whole immunopeptidome of the cell is harvested and then the peptides are identified using mass spectrometry [34]. This identifies many peptides at once. However, it is typically not possible to measure which of the up to six different MHC alleles presented which eluted peptide. Monoclonal cell lines, that only express a single MHC allele are a potential way around this limitation. There also exist algorithms to identify which allele is responsible for a peptide's presentation. This step is called "deconvolution" in the literature. However, it also often relies on data that can be unambiguously ascribed to a single allele to kickstart it.

Another disadvantage of the EL approach is, that it does not generate definitive negative examples. It only reports about the presence of a peptide at the cell's surface. It cannot assert the absence of a peptide from the individuals immunopeptidome. For example a peptide present in the human genome, might only not be presented by the cell because its protein is not being expressed by the particular cell. Despite it's drawbacks, the high quantity of data generated by this approach will make it the backbone of our examinations.

A.2 Data split

MHC allele dimension: As each individual normally has at least one working copy of each HLA gene (A/B/C), it is not possible to hold out a full gene **AR:what is holding out a full gene?**. So, we hold out observations on the HLA group level (e.g. HLA-A*01, ...). First we count how many observations belong to each HLA group (from a donor/cell with at least one HLA gene being in the group). We find that the groups are highly unequally represented in the dataset. To find at least 5 groups for each set, we perform the following steps until a satisfactory split is found.

First, we randomly assign HLA groups (and the linked examples) to the validation set until its target number of examples is reached - overriding the standard training set assignment. Then we randomly assign allele groups to the test set until its target number of examples is reached - overriding any earlier assignment. If we assigned too few or too many we repeat. This process ensures that no validation or test MHC group enters the training phase and no test MHC group enters the validation or training phases.

Protein dimension: Following this, we split off another validation and test set from the remaining training set. This second split is, however, based on an observation's mapped proteins not its linked MHC alleles. Due to homology we cannot just split the dataset based on the protein names. There are different approaches to deal with this. We use the python networkx package, which allows to build and explore graph structures. With it we use the Ensembl BioMart paralogue table [35] to link related genes as well as proteins to their respective gene. We then randomly assign disconnected sub-graphs to the various splits until our target values are reached.

Applying the above, we obtain the partition of our data in Table 1.

SPLIT	TRAIN	VAL-PROT	TEST-PROT	VAL-MHC	TEST-MHC
TOTAL OBSERVATIONS	1,408K (71.8%)	70K (3.6%)	71K (3.6%)	204K (10.4%)	206K (10.5%)
SINGLE ALLELE	206K (10.5%)	10K (0.5%)	11K (0.6%)	24K (1.2%)	43K (2.2%)
MULTIPLE ALLELE	1,202K (61.3%)	60K (3.1%)	60K (3.1%)	181K (9.2%)	164K (8.3%)

Table 1: Observations per dataset split

A.3 Hyperparameter search and training

We searched parameters along 3 dimensions: pooling mechanism, decoys per hit and learning rate. We considered using the <cls> tokens embedding, averaging and a classical attention mechanism as pooling mechanism. Further, we compare using datasets enriched by 19 and 99 decoys per observation. Eventually, we tried using 1e-04, 1e-05 and 1e-06 as initial learning rates. For the hyperparameter search, each model was trained on 10% of the SA data for 64,599 steps (one epoch for the 99 decoys per observation datasets and five epochs for the 19 decoys per observation datasets - so both of them have seen the same observations at least once).

We evaluated each model on 10% of the MHC-validation and protein-validation set SA data (using 99 decoys per hit). Appendix Table 3 shows the result for the hyper-parameter search using a learning rate of 1e-05 and Appendix Table 4 the results for a learning rate of 1e-06. Using a learning rate of 1e-04 would most of the time not result in any detected hits.

Table 3 shows various performance metrics (best column values in red) on our two validation sets for the 6 models described above at two points during their training. Different initializations might deliver different results, as mentioned above we didn't use the full dataset and the training will not have converged after 64,559 steps yet. However, given our limited computing resources and the time it takes to train transformers, we had to base our decision on these numbers. Dependent on the chosen metric, one or the other pooling mechanism and one or the other hits-to-decoys ratio looks best. In general, the unbalanced dataset (99 decoys per hit) at first (step 12,911) leads to quite poor classifiers in terms of receiver-operating-curve (ROC)-area-under-the-curve (AUC) and average precision (AP). However, after a full epoch they are able to make up most of this. In fact, as found by [36], models trained on an unbalanced dataset close to the actual data distribution (99 decoys per observation) show better accuracy. Actually, the models using only 19 decoys per observation have a worse accuracy than just always predicting negative. However, accuracy is not informative and reliable when dealing with highly imbalanced data (e.g. when the majority to minority class ratio is 999:1 a classifier always predicting the majority class will have 99.9% accuracy). The models performing best on ROC-AUC and AP all were trained using 19 decoys-per-hit (Appendix Table 3). So we will use this. Using the classification token as input to the head had the best performance (in red) in 5 cases while the attention mechanism only had the best performance in 3 cases and the averaging in 2 cases. We will, therefore, train our final model using the classification token's output as input to the model's head. Due to time reasons and little improvement we stopped after epoch 5 and use this as our final model (Table 2).

AFTER		VAL-MHC			VAL-PROTEIN		
EPOCHS	STEPS	AP	ROC AUC	ACC°	AP	ROC AUC	ACC°
1	128494	0.571	0.966	0.989	0.667	0.985	0.988
2	1008417	0.646	0.976	0.992	0.762	0.993	0.993
3	1888340	0.671	0.978	0.993	0.767	0.993	0.994
4	2768263	0.673	0.978	0.992	0.768	0.993	0.993
5	3648186	0.683	0.979	0.992	0.765	0.993	0.994

Table 2: Performance comparison during training (°... accuracy)

POOLING	DECOYS	AFTER X STEPS	VAL-MHC ROC AUC	VAL-MHC AP	VAL-MHC ACC°	VAL-PROTEIN ROC AUC	VAL-PROTEIN AP	VAL-PROTEIN ACC°
CLS	19	12911	0.938	0.319	0.986	0.956	0.447	0.987
		64559	0.949	0.394	0.982	0.964	0.552	0.983
	99	12911	0.823	0.062	0.990	0.917	0.232	0.990
		64559	0.945	0.349	0.991	0.960	0.492	0.992
ATTN	19	12911	0.936	0.298	0.973	0.957	0.447	0.982
		64559	0.941	0.410	0.983	0.964	0.535	0.982
	99	12911	0.792	0.044	0.990	0.884	0.184	0.990
		64559	0.938	0.315	0.990	0.958	0.496	0.992
AVG	19	12911	0.926	0.252	0.982	0.954	0.417	0.985
		64559	0.949	0.369	0.977	0.960	0.506	0.980
	99	12911	0.689	0.026	0.990	0.835	0.145	0.990
		64559	0.924	0.287	0.990	0.956	0.471	0.992

Table 3: Performance comparison for a learning rate of 1e-05 (°... accuracy)

POOLING	DECOYS	AFTER X STEPS	VAL-MHC ROC AUC	VAL-MHC AP	VAL-MHC ACC°	VAL-PROTEIN ROC AUC	VAL-PROTEIN AP	VAL-PROTEIN ACC°
CLS	19	12911	0.660	0.020	0.990	0.771	0.061	0.990
		64559	0.902	0.209	0.974	0.943	0.356	0.975
	99	12911	0.584	0.013	0.990	0.689	0.033	0.990
		64559	0.721	0.027	0.990	0.835	0.119	0.990
ATTN	19	12911	0.649	0.018	0.990	0.774	0.073	0.990
		64559	0.902	0.210	0.975	0.937	0.349	0.975
	99	12911	0.594	0.014	0.990	0.694	0.034	0.990
		64559	0.731	0.029	0.990	0.829	0.132	0.990
AVG	19	12911	0.657	0.020	0.990	0.776	0.081	0.990
		64559	0.892	0.202	0.969	0.937	0.349	0.970
	99	12911	0.601	0.016	0.990	0.700	0.037	0.990
		64559	0.733	0.031	0.990	0.840	0.134	0.990

Table 4: Performance comparison for a learning rate of 1e-06 (°... accuracy)

A.4 Evaluation and Benchmarking

To shed light on what our model has learnt and to assess its quality, we performed:
- Evaluation on the test sets
- Comparison of our model to MHCflurry and NetMHCpan
- Model interpretation by LIME analysis of peptide, flanks and pseudo sequence feature importances, using motifs and by SHAP analysis of peptide AAs contributions (see)

A.5 Evaluation on the test set

Table 5 shows the test set performance of our selected final model (epoch 5 in Table 2). We see that the values are not very different.

AFTER EPOCHS	AFTER STEPS	TEST-MHC AP	TEST-MHC ROC AUC	TEST-MHC ACC°	TEST-PROTEIN AP	TEST-PROTEIN ROC AUC	TEST-PROTEIN ACC°
5	3648186	0.704	0.981	0.993	0.755	0.992	0.993

Table 5: Performance on the test set (°... accuracy)

A.6 Comparison to MHCflurry and NetMHCpan

The MULTIALLELIC benchmark dataset of MHCflurry consists of 9,158,100 examples. Each has a peptide, N-flank (15 AA), C-flank (15 AA), up to six HLA alleles as well as the predictions of NetMHCpan, MixMHCpred and MHCflurry for the example. [11] generated this dataset from 11 studies using EL data. For each hit they randomly generated 99 decoys. A more detailed description and the full dataset is available in [11, Supplement Data S1]. We run two evaluations on this - one on the whole dataset (9,158,100 examples) and one for which we removed examples of peptides that

were already part of our training dataset (2,781,898 examples). For these we predict our model's presentation score, calculate performance metrics and plot precision-recall (PR) curves for MHCflurry, NetMHCpan and our model (ImmunoBERT).

A.7 Benchmarking

Unluckily, there are no generally agreed upon standard benchmarking datasets available in our domain. However, [11] have curated a benchmark dataset. We ran the below analysis on the full dataset as well as on one in which we exclude peptides from our training set. We calculated the AP as well as the ROC-AUC for MHCflurry, NetMHCpan and ImmunoBERT. We also plotted the PR-curves for them (Table 6). The models were trained on different datasets. So any judgement about the advantageousness of the architectures is not valid. However, the comparison is useful to compare the practical predictive power of the models.

For both datasets, our model shows an AP in between MHCflurry and NetMHCpan. In particular, it does well for thresholds corresponding to intermediate recall levels. However, it achieves less ROC-AUC than the others, possibly caused by the sharp drop in performance for higher recall values. The performance on the reduced set is far worse for all models. So, also the other two models might have already seen similar peptides as were removed during their training. As we explicitly only removed ours, this skews the reduced dataset against our model.

We decided to generate decoys once and show the model the same decoys in each epoch. This was done to ensure reproducibility of results. In contrast, NetMHCpan and MHCflurry resample decoys each epoch [11]. In hindsight, this might be a better design choice and might have led to later convergence during training of our model.

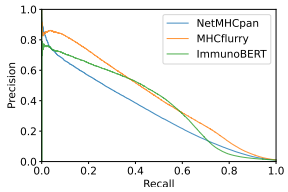
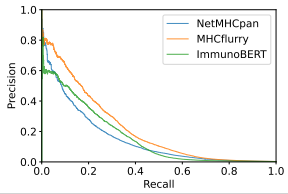
	Model / Metric	AP	ROC AUC	PR-Curve
Full Dataset	NetMHCpan	0.327	0.916	
	MHCflurry	0.427	0.938	
	ImmunoBERT	0.383	0.893	
Reduced Dataset	NetMHCpan	0.151	0.873	
	MHCflurry	0.215	0.890	
	ImmunoBERT	0.163	0.831	

Table 6: Performance comparison on benchmark dataset

A.8 Interpretation

On the following pages, the charts for the remaining HLA alleles that were note selected for detailed discussion can be found.

A.9 HLA-A*33:03

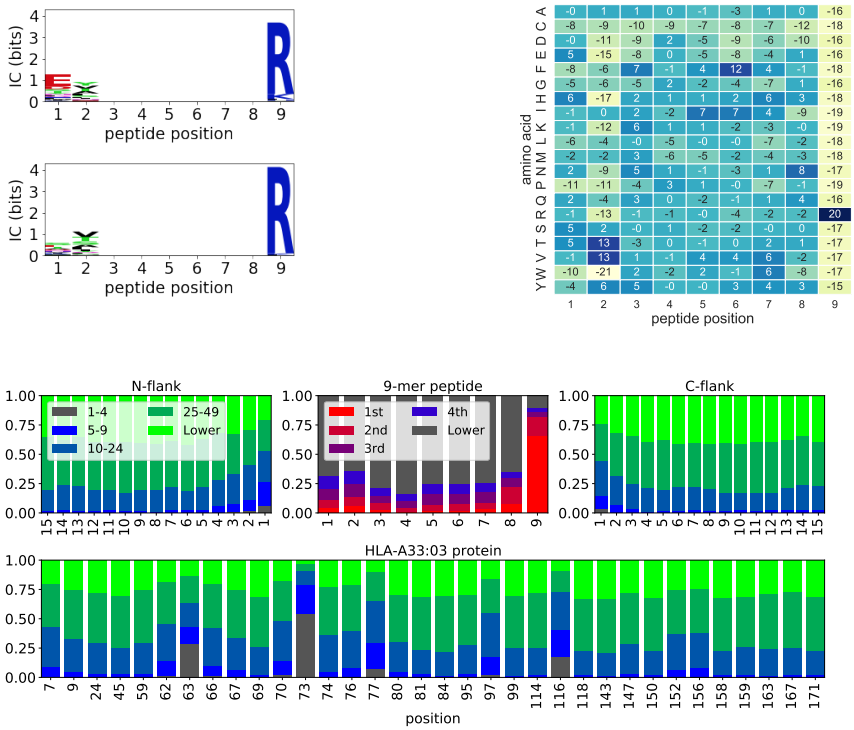


Figure 8: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.10 HLA-A*36:01

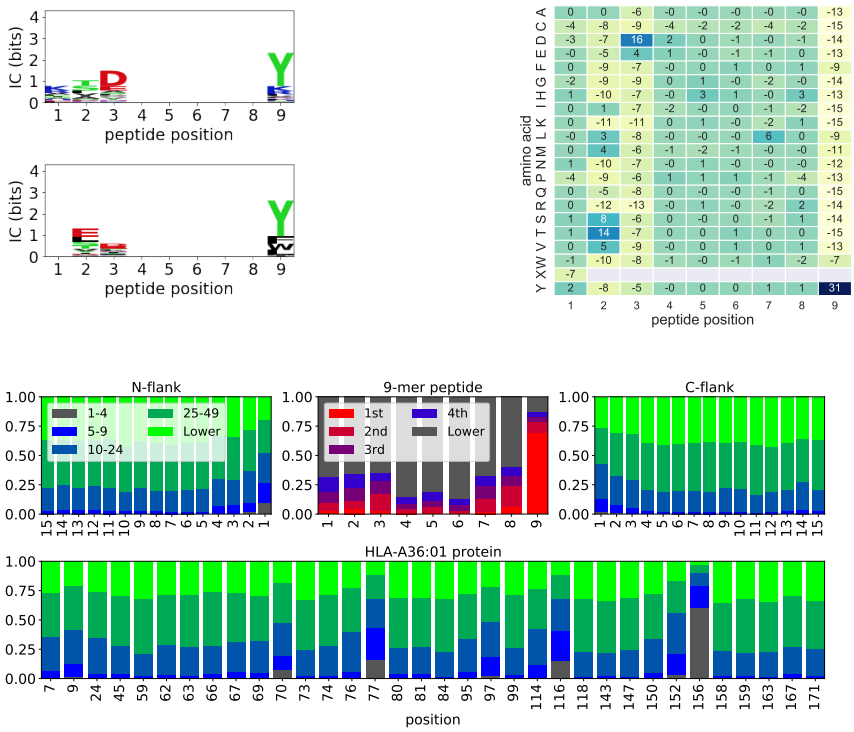


Figure 9: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.11 HLA-A*74:01

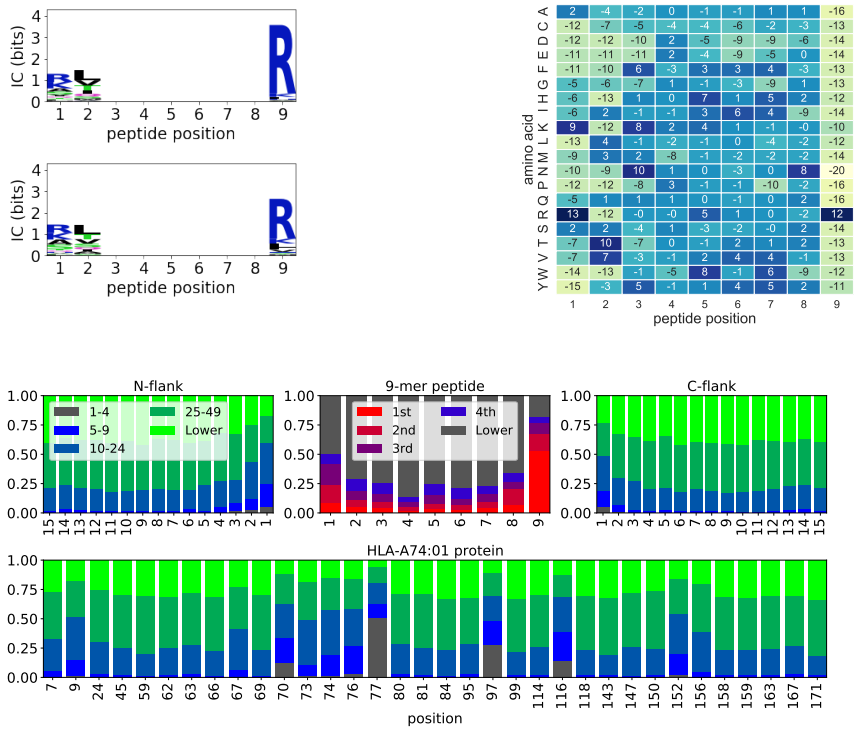


Figure 10: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.12 HLA-B*37:01

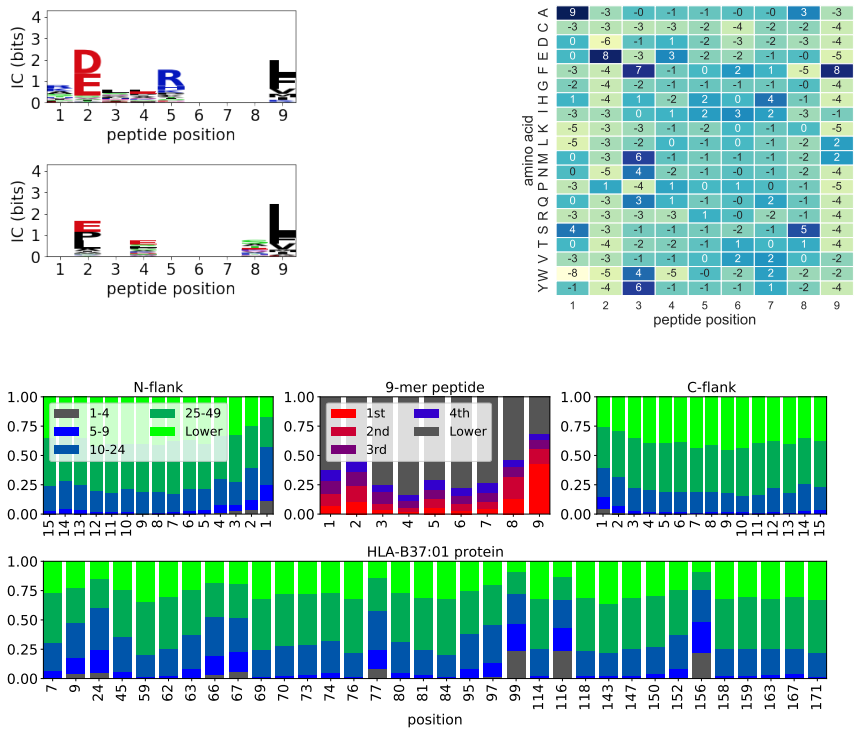


Figure 11: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.13 HLA-B*46:01

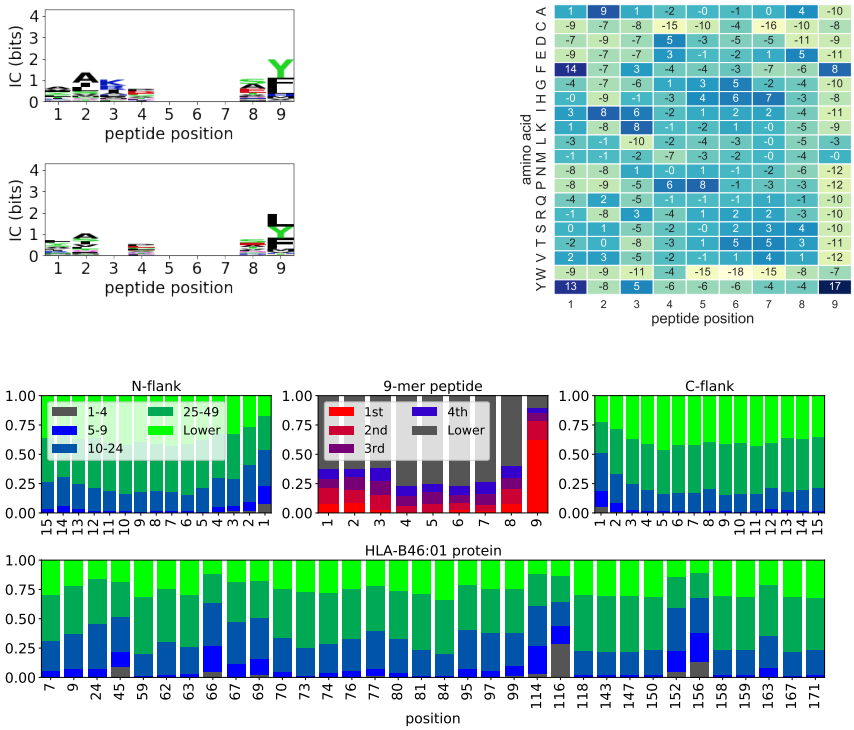


Figure 12: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.14 HLA-B*58:01

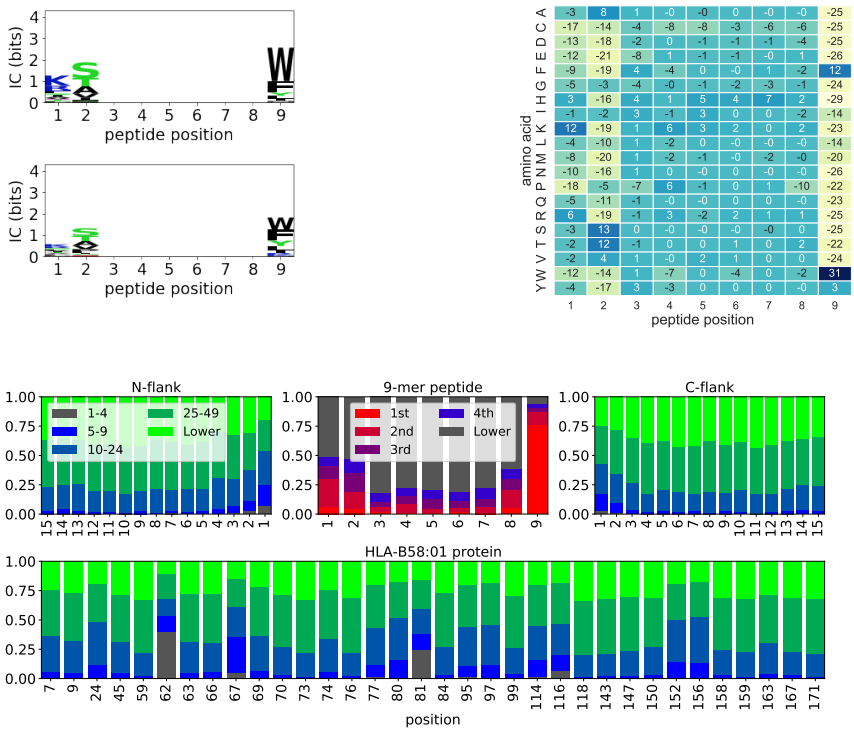


Figure 13: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.15 HLA-B*58:02

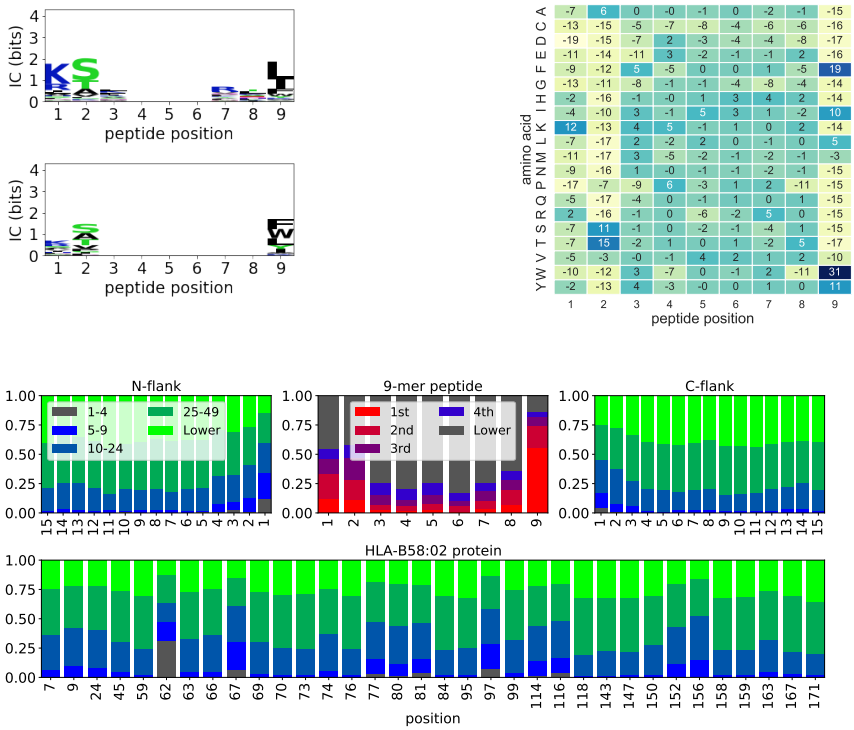


Figure 14: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.16 HLA-C*15:02

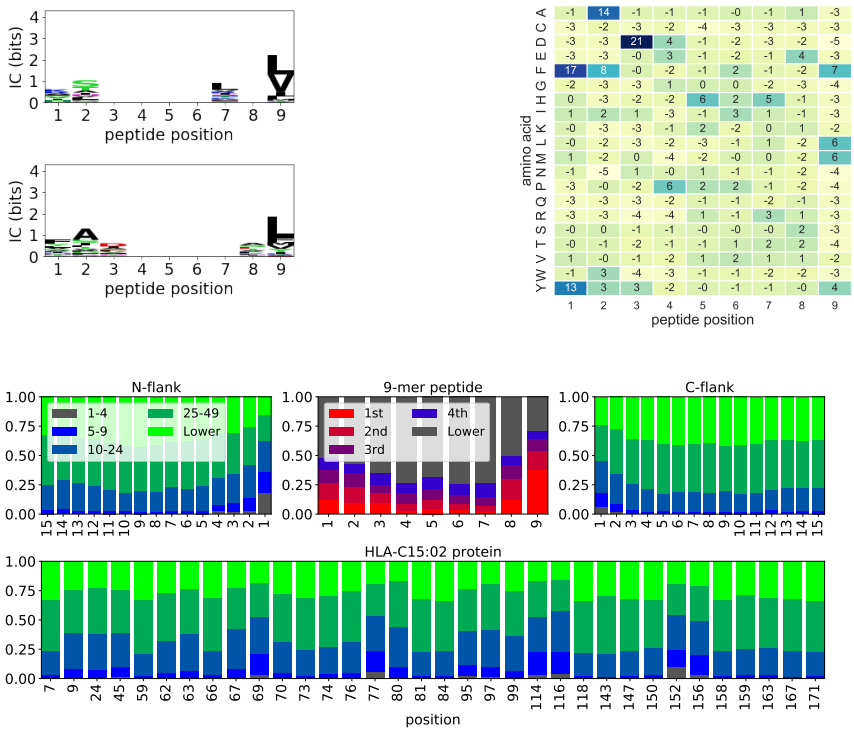


Figure 15: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.17 HLA-C*01:02

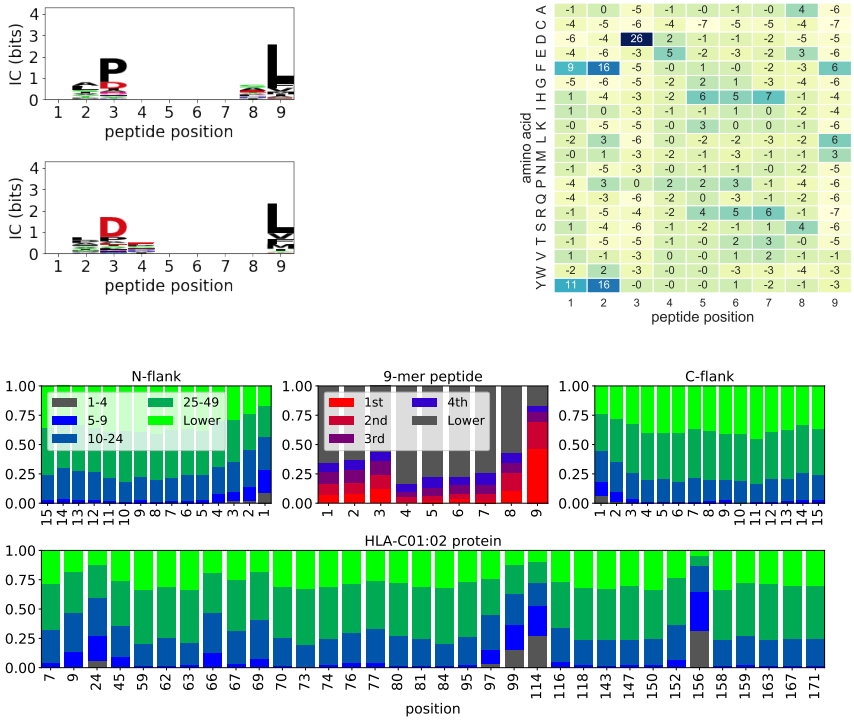


Figure 16: Motifs (top left), mean SHAP values (top right) and LIME feature ranks

A.18 HLA-C*17:01

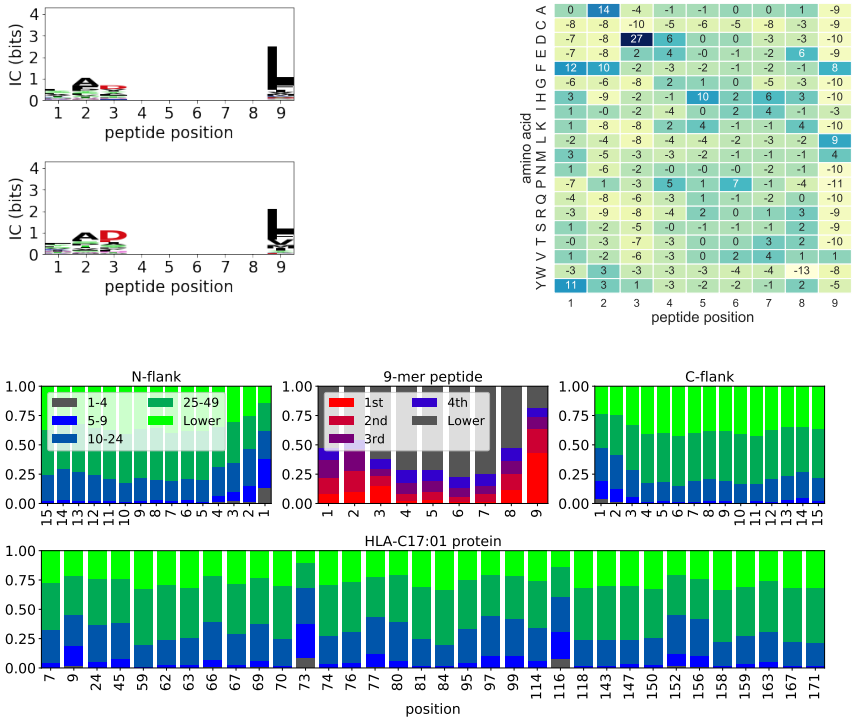


Figure 17: Motifs (top left), mean SHAP values (top right) and LIME feature ranks