# Improving generalizability for MHC-binding peptide predictions through structure-based geometric deep learning

Dario F. Marzella[†], Giulia Crocioni[†], Tadija Radusinović, Daniil Lepikhov, Heleen Severin, Dani L. Bodor, Daniel T. Rademaker, ChiaYu Lin, Sonja Georgievska, Amy L. Kessler, Pablo Lopez-Tarifa, Sonja Buschow, Erik Bekkers, Li C. Xue
[†]these authors contributed equally

## Abstract

Understanding the peptide presentation mechanism of Major Histocompatibility Complex (MHC) is crucial to study the recognition of pathogens, the treatment of autoimmune diseases and in developing cancer immunotherapies. To comprehend these mechanisms and design effective therapies at affordable costs, it is fundamental to be able to precisely predict *in silico* which peptides can be bound by each MHC. Many computational approaches have been developed to identify MHC-binding peptides based on their sequences, however, they present biases and their performance on less-studied MHC alleles with limited experimental binding data is severely restricted by their sequence-based (SeqB) nature. We reason that structure-based (StrB) methods could learn the physico-chemical and geometrical rules of peptide-MHC (pMHC) complexes conserved among the alleles, thus generalizing better. We designed three supervised StrB geometric deep learning (GDL) methods, showing superior generalization across all StrB methods over two SeqB methods when the test data is distant from the training data. To further explore data efficiency, we designed a novel self-supervised GDL approach trained only on ~1K x-ray structures, 3D-SSL, which predicts binding affinities without seeing any during training. Finally, we showcase our StrB method on a case study from the HPV vaccine design to show how StrB methods are robust to biases in the binding data. These findings highlight the potentials of structure-based methods in efficiently utilizing limited data, enhancing generalizability. This study bears important implications on data hungry fields, for example, the long-standing T cell receptor (TCR) specificity challenge.

## Introduction

The Major Histocompatibility Complex class I (MHC-I) guides the immune response by presenting peptides to CD8+ T lymphocytes. With over 14,000 human MHC-I proteins[1] encoded by HLA-A, HLA-B, and HLA-C genes, the theoretical 20^9 9-residue peptides create impractical *in vitro* testing scenarios. Since the high-affinity interaction with MHC is the most selective requirement for peptide presentation[2], predicting peptides bound to specific MHC molecules is pivotal for strategic in vitro testing decisions. Accurate pMHC binding predictors are essential across diverse immunology fields[3,4], including autoimmunity, transplant rejection, pathogen vaccine design, and cancer immunotherapy.

Much effort has been devoted to design predictors for pMHC binding[5]. Most state-of-the-art (SOTA) predictors are sequence-based machine learning methods[6–9] trained on a large amount of experimental binding data for pMHC (~600K for MHC-I[7] and >480K for MHC-II[6] ). Sequence-based (SeqB) methods take amino acid sequences as input to predict binding affinity (BA) or Eluted Ligand (EL) rank. In addition to this, NetMHCpan4.1 uses a structure-derived concept like the MHC pseudosequence[6], which includes only the MHC residues observed to interact with the peptide in X-ray structures. MHCflurry 2.0 adds to the MHC-I binding prediction an antigen processing model,

forming a final MHC presentation predictor that integrates both the scores. MHCflurry 2.0 has been shown to outperform NetMHCpan 4.0 and MixMHCpred 2.0.2 on held-out mass spectrometry experiments, presenting itself as the best available predictor for MHC-I peptide presentation[7]. These methods perform well on 214 well-studied alleles with hundreds of thousands of mass spectrometry data and have made important contributions to clinical treatments[10,11].

While these approaches have shown considerable advancement and contributed to numerous clinical trials, they are not without limitations. A notable constraint stems from their reliance on extensive data, posing challenges for the thousands of less-explored HLA alleles with limited available binding information. It's important to note that these methods often ensure a consistent distribution of alleles, peptides, and associated labels across the training and testing datasets. Some efforts are made to minimize redundancies in test datasets by removing identical peptides or employing independent test sets, but the redundancies between the independent test sets and training set are not removed. However, these practices often overlook the nuanced similarities between test and training cases. Moreover, these methods typically train and test on a subset of MHC alleles, which is minimal if compared to the broader spectrum of approximately 20,000 human alleles. Consequently, predictions for underrepresented alleles in these sets may exhibit lower accuracy[12], akin to an out-of-distribution (OOD) issue in machine learning, where real-case scenarios differ significantly from training data, posing challenges in generalization[13–15].

Even if SeqB methods have become the standard for peptide-MHC BA prediction, structure-based (StrB) methods have the potential to provide better predictions[16]. In fact, a StrB method, relying on local, rotation- and translation-invariant geometrical and physico-chemical properties of the complexes, could apply these learned rules to any peptide-MHC combination, independently of allele or sequence restrictions. In other words, by shifting the features domain from sequences to structural features, we could circumvent the OOD issue posed by real-world datasets.

StrB methods have been used for classifying peptides as binders vs non-binders[16,17]. An effective approach is 3pHLA[17], for which the authors generated 77,000 3D models generated with APE-Gen. They then used Rosetta[18] to extrapolate energy scores at each of the peptide's positions and provided these scores as a vector to train a random-forest predictor. However, one drawback of this method is that using an external software to summarize all the structural features in a single energy score is likely causing the important structural signal (such as clashes, local favorable interactions, and long-distance interactions) to get lost. It has been shown that summary-based energy scoring functions alone are insufficient in distinguishing near-native complexes from modeling artifacts. This suggests that an AI approach capable of directly accessing structural features could use them much more effectively than a classical energy function does.

Geometric Deep Learning (GDL) is a state-of-the-art Deep Learning field in handling protein structures [19–23], and it exploits multiple types of networks that can incorporate the geometrical information included in the data. 3D Convolutional Neural Networks (3D-CNNs) analyze structures on a 3D grid, capturing local geometric relationships effectively. While they maintain translation equivariance, they're sensitive to input rotation, which may be problematic without a common orientation. Despite this, 3D-CNNs perform well on structural datasets[22,24]. In (bio-)chemistry, Graph Neural Networks (GNNs) treat structures as connected nodes (atoms, residues) on a graph, updating information iteratively. Making GNNs structure-aware involves using distances between nodes, creating an invariant network. Equivariant GNNs (EGNN) enhance structural discernment by propagating vector features during message passing[25]. EGNN's simplicity and effectiveness in roto-

transaltion equivatiant (RT-equivariant) applications make it widely used for predicting molecule properties and generating realistic small molecules.

In this work, we evaluate whether GDL StrB methods can have a more robust generalization than SeqB methods for pMHC binding predictions. We first designed an allele-clustered BA dataset to test the generalization power of a network to unseen pMHC cases. We argue that the performance of the allele-clustered configuration is more indicative of the OOD generalization power and the robustness of an pMHC BA predictor compared to the shuffled configuration, in which the distribution of the training and test data should be in-distribution by design. We trained three different GDL StrB networks and two SeqB networks and tested them on our held-out alleles test set, comparing these performances with a shuffled data performance to evaluate the accuracy drop when generalizing to unseen cases. This experiment showed that StrB GDL predictors generalize better than SeqB predictors even with minimal features. In addition, being inspired by the success of the self-supervised learning (SSL) in natural language processing, we also implemented a novel self-supervised GDL method, which is trained on purely X-ray pMHC structures and without exposure to any binding affinity data. Our 3D-SSL method predicts the amino acid identity of randomly masked residues based on their neighboring residues. Our results show that our 3D-SSL method, trained on 1245 amount of data, is competitive with supervised approaches which are trained on ~90K binding data, demonstrating that geometric rules learned from high-quality structures generalize to similar binding prediction problems. Finally, we have tested one of our models on a study-case, showing how StrB methods can ignore hurtful biases present in the training data.

## Results

**Data separation to evaluate generalization**

We set out to test the generalization power of StrB over SeqB approaches to unseen data through two different data splitting configurations of the same binding affinity dataset[7]. We compared a stratified randomly shuffled dataset with an allele-clustered one. The allele-clustered data configuration was designed to test the applicability of the models to a test set composed of unseen data, the distribution of which might deviate significantly from the training data (e.g. patients with rare alleles).

In the first configuration (shuffled), we randomly split the data in training, validation and test sets, using target stratification to ensure identical target distribution in all data sets (Fig 1B). This configuration is similar to what is usually done in the existing literature. In the second experiment (allele-clustered), we clustered the data based on the MHC pseudosequence[26] using a PAM30-based hierarchical clustering (see Methods) for each gene (i.e., HLA-A, -B and -C). Then for each gene we selected the most distant clusters based on the MHC pseudosequence's dendrogram (Fig. 1A, B and C) so that ~10% of each gene's data (10% for HLA-A, 11% for -B and 12% for -C) are selected as a test set. We shuffled and split the remaining data into training and validation sets in a target-stratified manner. This design allows the networks to sample a large number of cases from every HLA gene, but keeps hidden distant groups of alleles, simulating the real-case scenario in which the network must perform on an allele it never or it hardly saw.

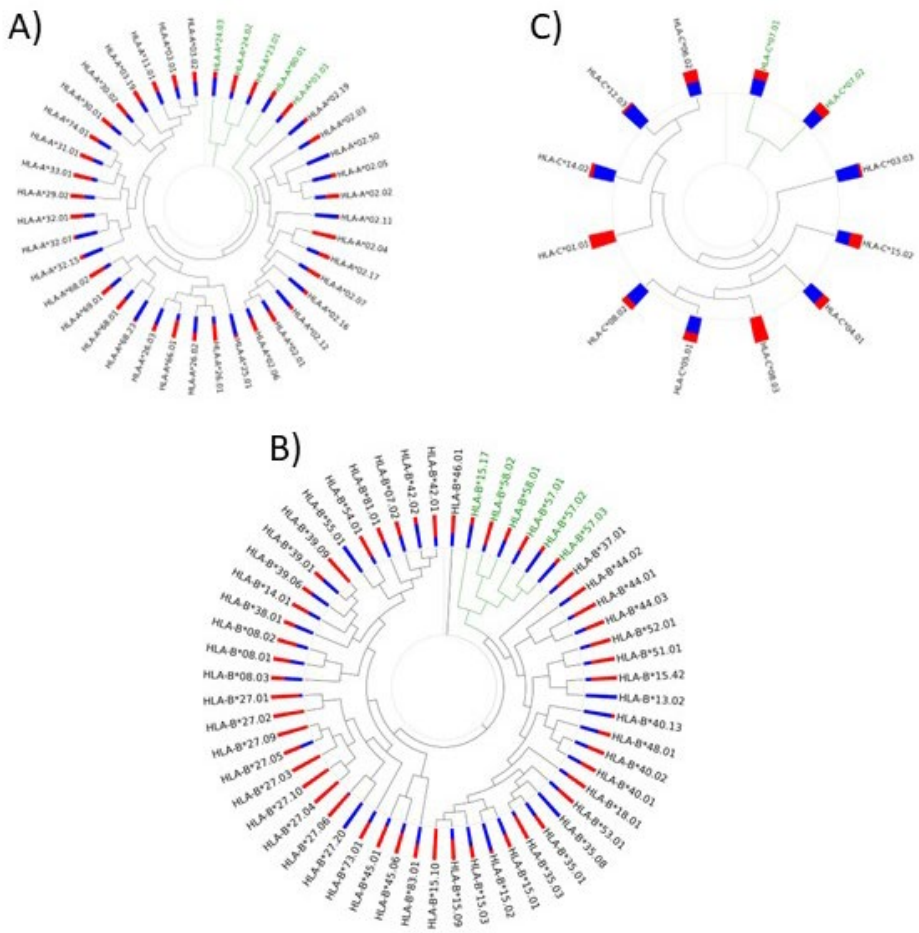


**Figure 1. Clustered data overview**
Hierarchical clustering of the pseudosequence's from the dataset's genes (A for HLA-A, B for -B and C for -C). Black labels and clade lines represent training data, while green labels and clade lines represent test data. The blue/red bar by each label represents the binder/non-binder ratio for the gene.

## StrB deep learning predictors show more robust performance than SeqB

For StrB predictors we used architectures with various E(3)-equivariance properties and with various types of physico-chemical features. We have developed an atom-level 3D-CNN, a residue-level GNN using different structural and energy-related features (Tables 2 and 3, respectively) and a residue-level EGNN using as input features only the residues identities and their XYZ coordinates (Table 4). The 3D-CNN encodes the input into a grid and maps the atoms and their features to the grid's voxels, losing the exact atom location but maintaining a description of the energy landscape (Fig. 2A). The GNN, on the other hand, represents each residue as a graph node and turns each inter-residue interaction within 15 Angstrom into an edge, assigning subsequently to each node and edge a series of features, listed in Table 2. Finally, the EGNN, similarly to GNN, represents a pMHC structure as a graph but it only uses residue identity and relative position as (node) features.

Most SOTA is not available to be retrained to be directly compared with. To mimic SOTA, we used two SeqB methods: we developed a simple Multi-Layer Perceptron (MLP) to represent a baseline sequence-based approach and we retrained MHCflurry2.0, being a state-of-the-art architecture.

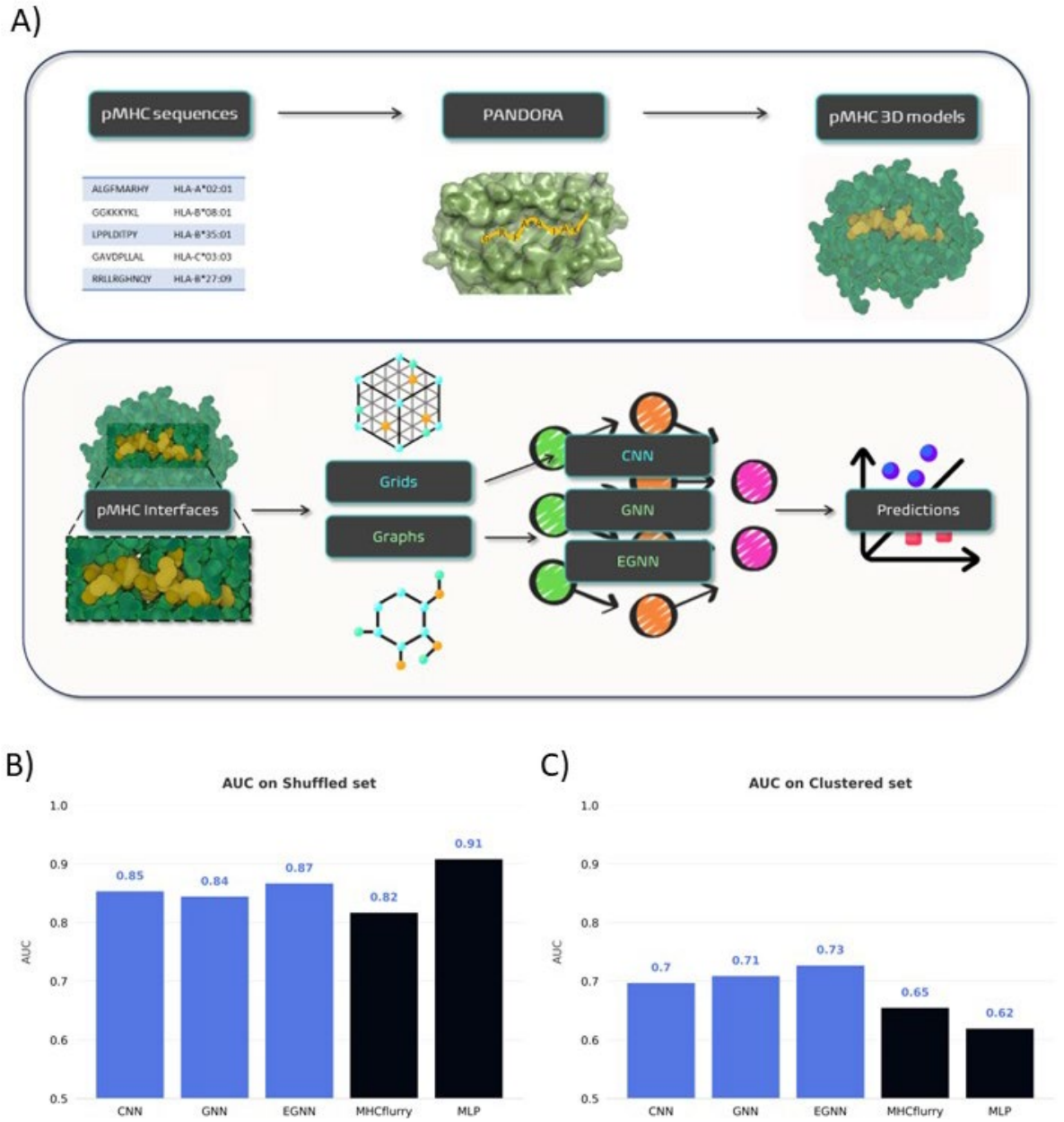


**Figure 2. Supervised StrB methods overvierw and performances**

A) StrB methods overview. In the top panel, 3D-models of the pMHC complexes are generated with PANDORA[27]. Overview of the pipeline employed for processing pMHC complexes and training the supervised structure-based networks. The process involves: 1) Identifying interface atoms and residues, 2) converting each pMHC interface into volumetric grids and graphs enriched with geometrical and physicochemical information 3) Using these representations to train the networks: volumetric grids for the CNN, richer graphs the GNN and simpler graphs for the EGNN 4) Evaluate the prediction scores for BA. B) StrB and SeqB methods AUC on the shuffled dataset. C) StrB and SeqB methods AUC on the clustered-allele dataset.

For the shuffled data configuration, the results are mainly comparable (Figure 2B), with the MLP and our EGNN showing the best AUCs (0.91 and 0.87, respectively). The shuffled data configuration resembles the commonly used scenario in the existing literature, where the test set features distributions do not necessarily deviate from the training set ones. In these scenarios, SeqB methods

that learn the features distribution can provide relatively good predictions, as shown in Figure 2B. StrB performances suggest their potential in reaching performance comparable to the current literature.

The predictors' performance in the allele-clustered configuration (Fig. 2C) shows a major drop in predictive power for the MLP and a consistent drop for both our StrB methods and MHCflurry 2.0, resulting in a considerably better AUC for the StrB methods. Although the results confirm MHCflurry 2.0 predictor's ability to generalize better than a simple MLP, they also show that our StrB methods are able to generalize to unseen allele sequences better than the SeqB methods used. The StrB predictors showed good performances in all configurations, having an area under the curve (AUC) score as high as 0.87 in the shuffled configuration, and as high as 0.73 in the allele-clustered one.

**3D-SSL yields good generalization with minimal training data**

A

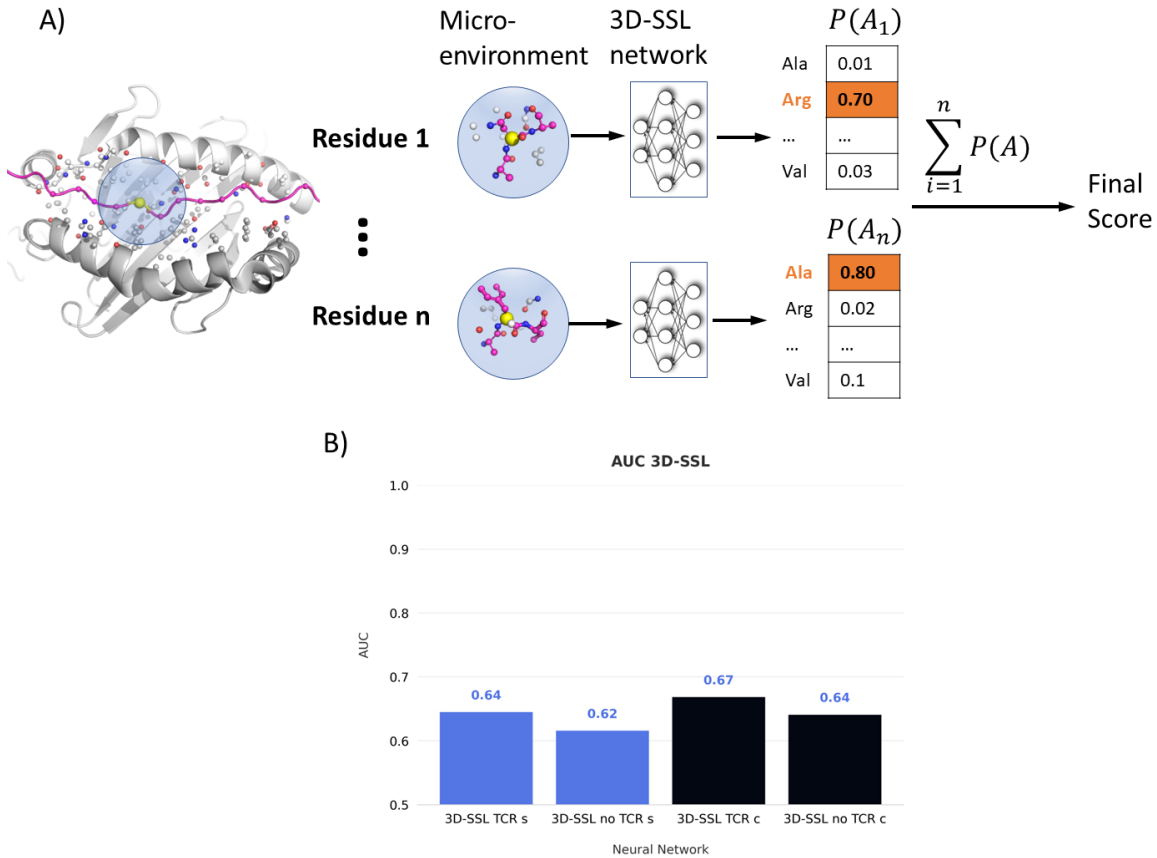


**Figure 3. Self-Supervised approach (3D-SSL) overview and performances**
A) 3D-SSL takes as input the microenvironment of each of the peptide's residues. For each microenvironment, it returns a probability distribution of the residues, giving higher scores to the residues that have higher chance to fit in such microenvironment. Finally, the probability of each of the actual peptide's residues is extracted and summed together to obtain a final score, which we used to predict the binding affinity. B) 3D-SSL method's performances when trained with and without peptide-TCR structures. Blue bars represent the performance on the shuffled test set and black bars represent the performance on the allele-clustered test set. There are no allele overlaps between the 3D-SSL training set and the clustered-allele test set.

Unlike supervised binding affinity prediction, where a network is trained to directly predict the binding label, our SSL method is trained on masked residue prediction (MRP). At training time, a

percentage of the residues in the pMHC complex is masked (i.e., set to a dedicated 'unknown' token). An EGNN is trained to predict the residue types of the masked nodes. This task is analogous to masked language models such as BERT and GPT3, which have had great success in the field of natural language processing[28,29]. In essence, MRP involves the network learning which residues are likely to be found in a given environment, effectively learning a multi-body statistical potential.

We trained our SSL on 514 pMHC X-ray structures from propedia[30], masking each residue of the peptide and making the network predict its identity. We used X-ray structures to allow the network to learn accurate potentials, avoiding possible biases coming from 3D-modeling. Once trained, the network is able to return a score for the likelihood of each residue to be found at each position of the peptide, similarly to a Boltzmann: high scores would be given to residues fitting coherently at each position, and low scores would be given to non-fitting residues. The final score is a sum of individual peptide residue contributions to the binding energy (see details in Methods). We then proceeded to test this network on our shuffled and allele-clustered test sets, by using the final score as a label for the binding affinity (Figure 3B). The SSL network, although being trained on slightly more than a thousand structures and not ever being exposed to the binding affinity label, could reach an AUC of 0.64, lower than the supervised StrB models but competitive with SeqB approaches, which are trained on 100 178 binding affinity data. This result shows the potential of SSL models in this field.

Interestingly, we found that adding x-ray peptide-TCR (T-Cell Receptor) structures to the pMHC training set slightly improved the binding affinity prediction performance, from AUC of 0.64 to AUC of 0.66 (Figure 3B). The improved results with peptide-TCR data indicate that adding competing binding environments could contribute to the prediction of the binding affinity between peptide and MHC.

**A study case: Vaccine for chronic Hepatitis B virus (HBV) infections**

We tested whether our StrB models are prone to the same biases as SeqB methods. A particular issue that we wanted to investigate of SOTA SeqB methods is being unable to identify binders that have a C-terminal (C-term) Glycine, possibly because the proteasome tends to never degrade a glycine C-term bond, leading to an underrepresentation of C-term glycines in the training sets of [6,7]. Although these peptides are indeed rarely produced by the proteasome, they can be produced by dendritic cells (DCs) during the process of cross-presentation by the action of lysosomal proteases[31]. We selected a study case from Kessler *et al.,* (2023)[32], where two cross-presented HLA-I peptides were reproducibly identified in HLA eluates from HLA-A2:01/ HLA-A11:01 matched DCs through liquid chromatography tandem mass spectrometry (LC-MS/MS), with the only difference being an extra C-terminal glycine in the second peptide (HLYSHPIIL and HLYSHPIILG, respectively). HLA-A2:01 binding could be confirmed for both peptides by an in vitro binding assay, yet NetMHCpan4.1 and MHCflurry2.0 both predicted only HLYSHPIIL to be an HLA-A*02:01 binder (Table 1). On the contrary, when we modeled both the peptides with PANDORA[27] and used our GNN to predict HLA binding, both peptides got predicted as HLA-A*02:01 binders. As apparent from the 3D-models (Fig 4), the C-term glycine is overhanging without causing major unfavorable interactions, clash nor torsion angle violation. The peptide HLYSHPIIL can form a weak salt bridge with MHC LYS146 using its C-term carboxyl group. Adding the glycine causes a bigger entropic effect (i.e. the glycine has to reduce its entropy to fit in the pocket), which can be balanced by the interactions of the glycine's carboxyl group, taking another weak salt bridge with LYS146 and an extra Hydrogen bond with THR80. Overall, structural modeling adds important insights into how the second peptide could bind in HLA-A*02:01 just as well as the first one, to which SeqB methods are blind as they solely rely on what they can infer from the sequence data seen during training.

**Table 1: NetMHCpan4.1, MHCflurry2.0 and our GNN prediction on HBV test peptides**

| | HLYSHPIIL | HLYSHPIILG | Thresholds |
|---|---|---|---|
| NetMHCpan4.1b | 0.056 | 2.981 | SB < 0.5; WB < 2.0 |
| MHCflurry2.0 | 36.485 nM | 5636.515 nM | SB < 500 nM; WB < 5000 nM |
| DeepRank2 GNN | 0.706 | 0.634 | SB > 0.51 |

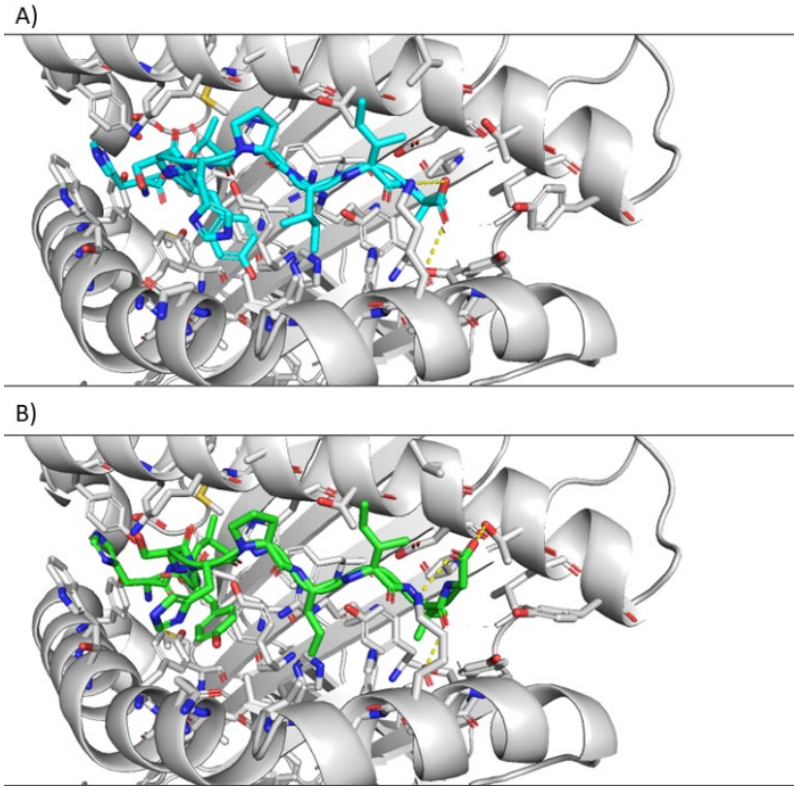


**Figure 4. Models of the two HBV virus case-study peptides in complex with HLA-A*02:01.**
The peptides HLYSHPIIL (panel A) and HLYSHPIILG (panel B) are shown in cyan and green, respectively. Expected polar interactions and hydrogen bonds are shown as yellow dashed lines.

## Discussion

In this study, we investigated the generalizability of StrB approaches for pMHC BA predictions. We trained three separate supervised StrB GDL BA predictors (a 3D-CNN, a GNN and an EGNN) using 3D models of pMHC complexes, and we showed that StrB networks have greater generalization power than SeqB networks when tested on unseen MHC alleles (0.73 vs 0.65, respectively). We have also trained an SSL StrB model on only 1245 X-ray structures and showed how it can exploit the information it learned to infer binding affinity from 3D models. Finally, we have compared one of our StrB models with two SOTA predictors on a challenging study case, showing how our method is more robust to biases from the training data.

We have tested very different StrB architectures, characterized by different data resolution, features, and equivariance properties, and have achieved very similar performances among all of them. Particularly interesting is the EGNN result, as this network's only virtual difference with the SeqB methods is the integration of the residues relative position and the way the network handles it. This

indicates that the greater generalization capacity of StrB methods depends on the geometrical information added with the data in the 3D models and the capacity of GDL to incorporate them in the learning, rather than on one specific set of features or network architecture.

We have also tested generalization to unseen peptides using different clustering methods (Gibbscluster and BLOSUM-based, see Suppl. Mat.). Results showed no clear preference for SeqB or StrB methods, possibly due to poor separation in clustering or the inherent ability of both methods to generalize to neoantigens. This aligns with expectations, as our training data has rich peptide variability (29132 unique peptides) but limited allele diversity (112 alleles), making clustering more impactful for allele variability than peptide variability.

The presented HBV test case, although not representing an extensive comparison of SeqB and StrB models biases, exemplifies how StrB methods better capture the biochemical environment of the pMHC complex, possibly being less influenced by non-biochemically-relevant biases in the data. It is important to notice, though, that some of these biases are determined by the antigen processing and presentation pathways. The absence of C-terminal glycines in MHC-I-presented peptides is to be expected in non-APC cells, as the proteasome has a very low chance of cleaving after a glycine[33]. Although an antigen preprocessing predictor, like the one MHCflurry2.0 uses, should exclude peptides unlikely to be generated by the proteasome, a binding predictor only considering peptide-MHC binding should still recognize them as binders if presented with them. Overall, the StrB GNN robustness might prove effective against other issues caused by the biases that SeqB models inherit from the data, like the low accuracy on predicting MHC binding of peptides containing cysteines, caused by underrepresentation of cysteines in the MS-derived immunopeptidomes used for training.

We note that the performance of retrained MHCflurry 2.0 is lower in our study than previously reported[7]. This can be caused by having used the BA predictor only, while the final score of MHCflurry 2.0 reported in [7] is provided by both the BA and the antigen processing (AP) predictors scores. The AP predictor was not included because we focused on developing predictors for MHC allele-dependent effects only (BA prediction), and not allele-independent effects (AP prediction). Second, MHCflurry 2.0 explorable hyperparameters during learning were designed on a much larger dataset, and not optimized for the restricted one we used. Lastly, we made a deliberate choice to omit the pre-training step of MHCflurry 2.0, in which the network is initially trained on a substantial volume of synthetic data (several hundred million pMHCs) labeled with MHCflurry 1.2.0 and subsequently fine-tuned on the actual data[7]. We opted against this step due to concerns that utilizing randomly generated peptides pre-labeled by another model might inadvertently introduce data leakage from the initial model to the subsequent one, potentially compromising the intended separation of our data.

While our presented methodology has already yielded promising results, there are some aspects that can be further improved. First, our structure-based approach is currently dependent on the availability of structural templates for the PANDORA software tool[27] (i.e., 864 MHC-I templates for the database version used for this study). In addition, the peptides' anchoring is predicted using netMHCpan 4.1, which could cause a mild data leakage through the 3D models and reduce performances in cases where the anchor was incorrectly predicted. Finally, given the higher dimensionality of the dataset used for our structure-based approaches, SeqB methods are currently more computationally efficient and can better adapt to large-scale rough screening. As for the StrB approaches we implemented, the GNN architecture yielded comparable results to CNN in terms of performance metrics, but has clear computational efficiency advantages: coarser grain resolution (residual vs. atomic), faster data

mapping and featurization (3.2±0.1 s vs. 6.2 s of computation on a CPU core for one pMHC complex), and lower storage requirements (744±17 KB vs. 5.6 MB of storage for one processed pMHC complex), thus representing the best choice for larger datasets at the production stage.

This study could expand by incorporating full pMHC binding data, including mass spectrometry, to enhance StrB methods, similar to SeqB methods in [34]. It's applicable to pMHC-II complexes, with StrB methods potentially handling peptide length variability better. Structural data from pMHC complexes can improve TCR/pMHC interaction modeling, addressing TCR specificity challenges. Integrating supervised methods with our self-supervised learning (SSL) and expanding StrB methods' information source to the Protein Data Bank or AlphaFold2 models via transfer learning are potential next steps.

We have showcased the potential inherent in GDL, StrB approaches to exhibit greater generalizability than SeqB approaches and to suffer less from biases in the data that are not connected to the pMHC binding. We've also shown how self-supervised learning can transfer information from reliable sources like X-ray structures to data-hungry fields. This has big implications for areas with limited training data, like the long-standing challenges in understanding pMHC immunogenicity and TCR specificity.

## Methods

**Binding affinity data collection**

BA data was collected from[7] through MHCflurry 2.0 ```download-fetch``` command. A total of 100 178 pMHC binding affinity measurements were collected, considering quantitative experiments only. We also filtered out data points considered not reliable because of known measurement inequalities. The final dataset consists of 100,178 data points each containing a peptide sequence, an MHC allele name, and their measured BA, using a threshold of 500 nM to separate binders from non-binders, resulting in 44,102 binders and 56,076 non-binders from 110 HLA alleles (Table S1).

**Data clustering**

In the shuffled data configuration, we split the data into training (90%), validation (10%) and test (15%) sets after shuffling, stratifying them on the binary target to make sure the positive and negative cases are equally represented in the different sets.

For the allele pseudosequence clustering, allele pseudosequences were collected from [7]. As explained in [7,26], these pseudosequences only containing the residues at the MHC positions that have been shown in X-ray structures to interact with the peptide, which makes it so every pseudosequence has the same length and no alignment is needed, as every position correspond to the same physical position on the MHC structure. These pseudosequence were separated per gene and scored against each other using a BLOSUM62 scoring matrix, and the so-obtained evolutionary distances were used to build a hierarchical clustering using the cluster.hierarchy module from SciPy[35] for each HLA gene. From these three clusterings (one per gene), we selected the most distant clusters until reaching the closest possible value to 10% of the data (10% for HLA-A, 11% for HLA-B and 12% for HLA-C) to form the test set. All the other data points were left as a train/validation set. The peptide sequence hierarchical clustering and Gibbs clustering are described in Suppl. Methods.

**3D modeling with PANDORA**

The BA dataset was processed with PANDORA (v2.0beta2) software[27] to generate 3D models of the pMHC complexes for these binding affinity data. For every case, PANDORA uses NetMHCpan4.1[6] to predict the binding core, thus assigning the anchor residues to restraint. The 3D models were generated with PANDORA's default mode (i.e., 20 3D models per case, no restraints flexibility added), with the exception that only the G-domain of the MHC is modeled, while the rest of the chain and the beta-2 microglobulin are ignored. The top molpdf scored model was selected as the best model and used for each further step.

**3D-CNN**

*Alignment of pMHC models.* 3D-CNNs are translation invariant but not rotation equivariant. For this reason, we decided to align all the pMHC 3D models to prevent any rotation issue and avoid resorting to data augmentation. In order to generate consistent 3D grids for each 3D model, all the selected 3D models were aligned using GradPose[36]. The principal component analysis (PCA) of the peptides coordinates was then calculated and the X, Y and Z axis of each PDB file were translated to match with the three principal components, to make sure the axis of the grid would sit along the axis of biggest variation in the peptides' backbone.

*Grid featurization.* The PDB files of the aligned pMHC models were then processed with DeepRank software to generate 3D grids as described in [22]. Each grid was set to have a shape of 35x30x30 Å and a resolution of 1 Å. Only residues of each chain within 8.5Å from the other chain are considered "interface" residues and kept for feature calculation. Features when then generated for each atom position and mapped to the grid points with a Gaussian mapping. In addition to DeepRank's default features, we added an Energy of Desolvation feature calculated as in [37], as the difference in the solvation energy between the bound and unbound complex (the unbound complex is obtained by simply translating apart peptide and MHC). We have also replaced DeepRank's default PSSM feature used for sequence embedding with the lighter skip-gram residue embedding taken from[38] (see Table 2 for the feature list). The CNN Network architecture is provided in the Supplementary Methods.
Table 1. Input features to the CNN architecture. Full features details on built-in deeprank features can be found on https://deeprank.readthedocs.io/en/latest/deeprank.features.html.

**Table 2. Input features to the CNN architecture.**

| **Feature & Shape & Type** | **Description** |
|---|---|
| SkipGram<br>[6, 1], float | Skip-Gram embedding for residue type |
| Edesolv<br>[1], float | Desolvation energy, calculated as in [37] |
| Anchor<br>[1], bool | Label to indicate if the residue is an anchor residue (peptide side) or a pocket residue (MHC side) |
| RCD_apolar-apolar<br>[1], float | Residue Contact Density for apolar-apolar interactions calculated as in [39] |
| RCD_apolar-charged<br>[1], float | Residue Contact Density for apolar-charged interactions calculated as in [39] |
| RCD_charged-charged<br>[1], float | Residue Contact Density for charged-charged interactions calculated as in [39] |

| | |
|---|---|
| RCD_polar-apolar<br>[1], float | Residue Contact Density forpolar-apolar interactions calculated as in [39] |
| RCD_polar-charged<br>[1], float | Residue Contact Density for polar-charged interactions calculated as in [39] |
| RCD_polar-polar<br>[1], float | Residue Contact Density for polar-polar interactions calculated as in [39] |
| RCD_total<br>[1], float | Residue Contact Density for total interactions calculated as in [39] |
| bsa<br>[1], float | Buried Surface Area |
| charge<br>[1], float | Charge based on the OPLS force field implemented in HADDOCK[37] |
| coulomb<br>[1], float | Electrostatics based on the OPLS force field implemented in HADDOCK |
| vdwaals<br>[1], float | Van Der Waals energy based on the OPLS force field implemented in HADDOCK |

**GNN**

*Graph featurization.* The PDB files were converted into protein-protein interfaces (PPI) in the form of residue-level graphs using DeepRank2 package[40] . The latter is a Python package inherited from our previous package DeepRank and that offers a complete framework to learn PPI interface patterns in an end-to-end fashion using GNNs.

For a given PDB file representing a pMHC complex, we defined the interface graph as all the residues involved in intermolecular contacts, i.e. the residues of a given chain having a heavy atom within an 15 Å distance cutoff of any heavy atom from the other chain. These contact residues formed the nodes of the graph. Edges were defined between two contact residues from distinct chains presenting a minimal atomic distance smaller than 15 Å. Node and edge residue-level features used are summarized in Table 1. The GNN Network architecture is provided in the Supplementary Methods.

**Table 3. Input features to the GNN architecture.** Full features details can be found on https://deeprank2.readthedocs.io/en/latest/features.html

| **Feature & Shape & Type** | **Description** |
|---|---|
| NODE FEATURES | |
| res_type<br>[21, 1], bool | One-hot representation of the node's residue (20 amino acids + unknown). |

| | |
|---|---|
| res_size<br>[1], int | The number of non-hydrogen atoms in the side chain. |
| res_mass<br>[1], float | The average residue mass in Da. |
| res_charge<br>[1], int | The charge of the residue in fully protonated state in Coulomb. |
| res_pI<br>[1], float | The isoelectric point, i.e., the pH at which the molecule has no net electric charge. |
| polarity<br>[4, 1], bool | One-hot representation of the polarity of the amino acid: NONPOLAR, POLAR, NEGATIVE, POSITIVE. |
| hb_donors<br>[1], int | The number of hydrogen bond donor atoms in the residue. |
| hb_acceptors<br>[1], int | The number of hydrogen bond acceptor atoms in the residue. |
| res_depth<br>[1], float | The average distance in Å of the residue to the closest molecule of bulk water, computed via BioPython. |
| hse<br>[3, 1], float | Half sphere exposure, which indicates how buried an amino acid residue is in the biomolecule, computed via BioPython. |
| sasa<br>[1], float | Solvent-accessible surface area, in $Å^2$, computed via FreeSASA. |
| bsa<br>[1], float | Buried surface area, which represents the area of the complex interface, in $Å^2$, computed via FreeSASA. |
| irc_total<br>[1], int | The number of residues on the other chain that are within a cutoff distance of 5.5 Å. |
| irc_negative_negative<br>irc_negative_positive,<br>irc_nonpolar_negative,<br>irc_nonpolar_nonpolar,<br>irc_nonpolar_polar,<br>irc_nonpolar_positive,<br>irc_polar_negative,<br>irc_polar_polar,<br>irc_polar_positive,<br>irc_positive_positive<br>[1], int | As above, but for specific residue polarity pairings. |
| EDGE FEATURES | |
| same_chain<br>[1], bool | Boolean indicating whether the edge connects nodes belonging to the same chain (1) or separate chains (0). |
| covalent<br>[1], bool | Boolean indicating whether nodes are covalently bound (1) or not (0). Covalency is not directly assessed, but any edge with a maximum distance of 2.1 Å is considered covalent. |
| electrostatic<br>[1], float | Electrostatic potential (also known as Coulomb potential) between two nodes, calculated using interatomic distances and charges of each atom. |

| | |
|---|---|
| vanderwaals<br>[1], float | Van der Waals potential (also known as Lennard-Jones potential) between two nodes, calculated using interatomic distances and a list of atoms with Van der Waals parameters. |
| distance<br>[1], float | Interatomic distance between atoms in Å, computed from the xyz atomic coordinates taken from the PDB file. |

**EGNN**

*Graph featurization.* The PDB files were converted into protein-protein interfaces (PPI) in the form of residue-level graphs. For a given PDB file representing a pMHC complex, we defined the interface graph as all the residues whose CA atoms fall within a 10 Å radius of peptide residue CAs. The only features tracked for residues are the residue type, coordinates, and a binary label representing whether the residue comes from the peptide or the MHC. Edges were defined using a k-nearest neighbors (KNN) procedure, where a node is connected to 10 of its closest neighbors. A binary edge attribute distinguishes between cross-chain edges and inter-chain edges. Node and edge features summarized in Table X. The EGNN Network architecture is provided in the Supplementary Methods.

*Training data for the SSL case.* By overlapping PDB IDs containing MHCI from ProtCID with the protein-peptide database Propedia[30], we extracted 1245 Propeida entries. These entries are composed of 514 peptide-MHC and 731 TCR$\alpha$-peptide or TCR$\beta$-peptide structures, separated out from the original PDB structures. We consider two SSL datasets from these entries. First, a reduced dataset of 517 only peptide-MHC structures, and all the extracted entries for the second one. This corresponds to two SSL training scenarios, one where the network sees only pMHC structures, and another where it also sees TCR$\alpha/\beta$-peptide structures.

**Table 4. Input features to the EGNN architecture.**

| **Feature & Shape & Type** | **Description** |
|---|---|
| NODE FEATURES | |
| res_type<br>[1], int | Index of node residue type. |
| pos<br>[3], float | The xyz coordinates of the residue C-alpha. |
| entity<br>[1], bool | Boolean indicating whether the node is the peptide (1) or the protein (0). Used for pooling. |
| EDGE FEATURES | |
| same_chain<br>[1], bool | Boolean indicating whether the edge connects nodes belonging to the same chain (1) or separate chains (0). |

**MHCflurry retraining**
Code from MHCflurry 2.0 GitHub repository was used to build a combined ensemble made of several models. For each experiment (random, peptide clustered and allele clustered), the only parameters modified from the MHCflurry 2.0 script used to generate the model were the data input and output paths. The original script is available at https://github.com/openvax/mhcflurry/blob/master/downloads-generation/models_class1_pan/GENERATE.sh.

**MLP**
The naive MLP consists of a simple architecture built on top of PyTorch 2.13 made of a BatchNorm1d() input layer followed by 3 hidden layers of 512 neurons each. BatchNorm1d() layer is fit for an optimal data representation no matter the minibatch data distribution. Each hidden layer has a dropout rate of 50% meaning that during training forward pass 50% of random neurons have an output set to 0, forcing the fit architecture to update its weight on different data patterns each time[41]. The optimizer used was Adam and the loss function was a binary cross entropy.

**HBV case study**
Immunopeptidomic data was obtained by LC-MS/MS as described in [32]. In short, monocyte-derived dendritic cells (moDCs) from 8 healthy donors were pulsed with 12 prototype hepatitis B virus (HBV) synthetic long peptides (SLPs) in combination with a maturation stimulus for 22 hours. HLA-I-peptide complexes were immunoprecipitated and subjected to data-dependent acquisition LC-MS/MS analysis. Data was analyzed in PEAKS, FDR5. Obtained spectra from cross-presented, SLP-derived peptides were manually validated by two MS experts. For the Table 1, NetMHCpan 4.1b from the web server[6] and MHCflurry2.0 locally installed[7] where used to predict the peptides' binding to HLA-A*02:01 with default options.

## Data & Code Availability

Training, validation and test sets will be rendered available upon journal acceptance as .csv files in zenodo. The code for data generation, featurization and training will be made openly public at https://github.com/DeepRank/3D-Vac.

## Authors Contribution

Conceived the study: LX, DFM
Prepared the data: DFM, DL, HS
3D-CNN: DFM, HS
GNN: GC, CL
EGNN: TR
SSL: TR
MHCflurry and MLP: DL, DFM
HBV data collection: ALK, SB
Project supervision and coordination: LX, EB
Manuscript writing and editing: DFM, GC, LX, SG, PL, DB, DL, ALK, SB, EB

All authors read and contributed to the manuscript. Giulia and Dario agree that the order of their respective names may be changed for personal pursuits to best suit their own interests.

## Acknowledgments

This project was supported by the Netherlands eScience Center under grant number NLESC.OEC.2021.008. This work was also supported by the NVIDIA Academic Award. We thank SurfSara for their generous GPU and CPU supercomputing resources. We sincerely thank the funding through the AI-For-Health program at Radboudumc for H. Severin and D. Lepikov. We also thank Coos Baakman, the main developer of the processing part of the DeepRank2 package, and Sven van der Burg, who helped with the refactoring and the maintenance of the package in the first phase of the project. We thank the MHCflurry 2.0 author Timothy O'Donnell for the kind indications in how to retrain their method.

# References

1. Robinson, J. *et al.* IPD-IMGT/HLA Database. *Nucleic Acids Res.* **48**, D948–D955 (2020).
2. Garstka, M. A. *et al.* The first step of peptide selection in antigen presentation by MHC class I molecules. *Proc. Natl. Acad. Sci.* **112**, 1505–1510 (2015).
3. Gfeller, D. & Bassani-Sternberg, M. Predicting Antigen Presentation-What Could We Learn From a Million Peptides? *Front. Immunol.* **9**, 1716 (2018).
4. Sidney, J., Peters, B. & Sette, A. Epitope prediction and identification- adaptive T cell responses in humans. *Semin. Immunol.* **50**, 101418 (2020).
5. Peters, B. *et al.* A Community Resource Benchmarking Predictions of Peptide Binding to MHC-I Molecules. *PLOS Comput. Biol.* **2**, e65 (2006).
6. Reynisson, B., Alvarez, B., Paul, S., Peters, B. & Nielsen, M. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Res.* **48**, W449–W454 (2020).
7. O'Donnell, T. J., Rubinsteyn, A. & Laserson, U. MHCflurry 2.0: Improved Pan-Allele Prediction of MHC Class I-Presented Peptides by Incorporating Antigen Processing. *Cell Syst.* **11**, 42-48.e7 (2020).
8. Gfeller, D. *et al.* Improved predictions of antigen presentation and TCR recognition with MixMHCpred2.2 and PRIME2.0 reveal potent SARS-CoV-2 CD8+ T-cell epitopes. *Cell Syst.* **14**, 72-83.e5 (2023).

9. Wang, F. *et al.* MHCRoBERTa: pan-specific peptide–MHC class I binding prediction through transfer learning with label-agnostic protein sequences. *Brief. Bioinform.* **23**, bbab595 (2022).
10. Sahin, U. *et al.* Personalized RNA mutanome vaccines mobilize poly-specific therapeutic immunity against cancer. *Nature* **547**, 222–226 (2017).
11. Ott, P. A. *et al.* An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature* **547**, 217–221 (2017).
12. Nguyen, A., Nellore, A. & Thompson, R. F. Discordant results among major histocompatibility complex binding affinity prediction tools. Preprint at https://doi.org/10.12688/f1000research.132538.1 (2023).
13. *Dataset shift in machine learning*. (MIT Press, 2009).
14. Yao, H. *et al.* Improving Out-of-Distribution Robustness via Selective Augmentation. in *Proceedings of the 39th International Conference on Machine Learning* 25407–25437 (PMLR, 2022).
15. Krueger, D. *et al.* Out-of-Distribution Generalization via Risk Extrapolation (REx). in *Proceedings of the 38th International Conference on Machine Learning* 5815–5826 (PMLR, 2021).
16. Antunes, D. A., Abella, J. R., Devaurs, D., Rigo, M. M. & Kavraki, L. E. Structure-based Methods for Binding Mode and Binding Affinity Prediction for Peptide-MHC Complexes. *Curr. Top. Med. Chem.* **18**, 2239–2255 (2019).
17. Conev, A., Devaurs, D., Rigo, M. M., Antunes, D. A. & Kavraki, L. E. 3pHLA-score improves structure-based peptide-HLA binding affinity prediction. *Sci. Rep.* **12**, 10749 (2022).
18. Alford, R. F. *et al.* The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).
19. Gainza, P. *et al.* Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nat. Methods* **17**, 184–192 (2020).
20. Gainza, P. *et al.* De novo design of protein interactions with learned surface fingerprints. *Nature* **617**, 176–184 (2023).

21. Wang, X., Flannery, S. T. & Kihara, D. Protein Docking Model Evaluation by Graph Neural Networks. *Front. Mol. Biosci.* **8**, 647915 (2021).
22. Renaud, N. *et al.* DeepRank: a deep learning framework for data mining 3D protein-protein interfaces. *Nat. Commun.* **12**, 7068 (2021).
23. Tubiana, J., Schneidman-Duhovny, D. & Wolfson, H. J. ScanNet: an interpretable geometric deep learning model for structure-based protein binding site prediction. *Nat. Methods* **19**, 730–739 (2022).
24. Wang, X., Terashi, G., Christoffer, C. W., Zhu, M. & Kihara, D. Protein docking model evaluation by 3D deep convolutional neural networks. *Bioinforma. Oxf. Engl.* **36**, 2113–2118 (2020).
25. Satorras, V. G., Hoogeboom, E. & Welling, M. E(n) Equivariant Graph Neural Networks. Preprint at https://doi.org/10.48550/arXiv.2102.09844 (2022).
26. Hoof, I. *et al.* NetMHCpan, a method for MHC class I binding prediction beyond humans. *Immunogenetics* **61**, 1–13 (2009).
27. PANDORA v2.0: Benchmarking peptide-MHC II models and software improvements | bioRxiv. https://www.biorxiv.org/content/10.1101/2023.07.20.549892v1.
28. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprint at https://doi.org/10.48550/arXiv.1810.04805 (2019).
29. Brown, T. B. *et al.* Language Models are Few-Shot Learners. Preprint at https://doi.org/10.48550/arXiv.2005.14165 (2020).
30. Martins, P. M. *et al.* Propedia: a database for protein–peptide identification based on a hybrid clustering algorithm. *BMC Bioinformatics* **22**, 1 (2021).
31. Biniossek, M. L., Nägler, D. K., Becker-Pauly, C. & Schilling, O. Proteomic Identification of Protease Cleavage Sites Characterizes Prime and Non-prime Specificity of Cysteine Cathepsins B, L, and S. *J. Proteome Res.* **10**, 5363–5373 (2011).
32. Kessler, A. L. & Buschow, S. Manuscript in preparation. (2023).

33. Winter, M. B. *et al.* Immunoproteasome functions explained by divergence in cleavage specificity and regulation. *eLife* **6**, e27364.

34. Alvarez, B. *et al.* NNAlign_MA; MHC Peptidome Deconvolution for Accurate MHC Binding Motif Characterization and Improved T-cell Epitope Predictions. *Mol. Cell. Proteomics* **18**, 2459–2477 (2019).

35. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).

36. Rademaker, D. T., van Geemen, K. J. & Xue, L. C. GradPose: a very fast and memory-efficient gradient descent-based tool for superimposing millions of protein structures from computational simulations. *Bioinformatics* **39**, btad444 (2023).

37. Dominguez, C., Boelens, R. & Bonvin, A. M. J. J. HADDOCK: A Protein−Protein Docking Approach Based on Biochemical or Biophysical Information. *J. Am. Chem. Soc.* **125**, 1731–1737 (2003).

38. Phloyphisut, P., Pornputtapong, N., Sriswasdi, S. & Chuangsuwanich, E. MHCSeqNet: a deep neural network model for universal MHC binding prediction. *BMC Bioinformatics* **20**, 270 (2019).

39. Vangone, A. & Bonvin, A. M. Contacts-based prediction of binding affinity in protein–protein complexes. *eLife* **4**, e07454 (2015).

40. [PRE REVIEW]: DeepRank2: Mining 3D Protein Structures with Geometric Deep Learning · Issue #5878 · openjournals/joss-reviews. *GitHub* https://github.com/openjournals/joss-reviews/issues/5878.

41. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).