PMGen: From Peptide-MHC Prediction to Neoantigen Generation

Amir H. Asgary, Amirreza Aleyassin, Jonas A. Mehl,
Burkhard Ludewig, Michele Mishto, Juliane Liepe,
Johannes Soeding

Abstract

Major histocompatibility complexes (MHCs) present peptides to T cells, playing a central role in adaptive immunity against cancer and infections. Accurate structural modeling of peptide–MHC (pMHC) complexes is essential for developing personalized immunotherapies. Existing tools often focus on either MHC class I or II, support limited peptide lengths, and address isolated tasks such as pMHC binding prediction or structure modeling. We introduce Peptide MHC Generator (PMGen), an integrated framework for predicting, modeling, and generating peptides across both MHC classes and a broad range of peptide lengths. PMGen enhances AlphaFold predictions by incorporating anchor residue information into structural module. PMGen achieves high structural accuracy (mean peptide core RMSD: 0.54 Å for MHC-I, 0.33 Å for MHC-II), improving over the state of the art for both MHC classes. PMGen can also generates diverse, high-affinity peptides with minimal structural deviation and accurately models the impact of single-point mutations in benchmark neoantigen–wild-type pairs. PMGen is a versatile tool for advancing personalized immunotherapy and is freely available at `https://github.com/soedinglab/PMGen`.

1 Introduction

Major Histocompatibility Complexes (MHCs) are central to the adaptive immune system's ability to detect and respond to pathogens and cancer. They are divided into two classes: MHC class I (MHC-I), which presents intracellular peptides to $CD8^+$ T cells, and MHC class II (MHC-II), which presents extracellular peptides to $CD4^+$ T cells. When a peptide–MHC complex (pMHC) is recognized by a matching T cell receptor (TCR), it triggers a signaling cascade that leads to a targeted immune response [1].

Antigenic peptides are vital for orchestrating immune activity, and their generation and presentation pathways are tightly regulated to ensure specificity and ideally prevent autoimmunity. This regulation involves peptide generation from its antigen, transport and selective peptide binding to MHC molecules, activation of TCRs, and modulation by regulatory T cells (Tregs). When this balance is disrupted, such as when self-peptides are misidentified as foreign, autoimmune diseases can result [2].

Designing peptides that bind effectively to MHC molecules has direct applications for immunotherapy and vaccine development. By identifying or engineering antigenic peptides with high binding affinity to specific MHC alleles, we could develop personalized treatments for cancer and infectious diseases. Immune checkpoint therapies, for example, have shown remarkable success in certain cancer patients by leveraging MHC-I-restricted neoantigen responses mediated by $CD8^+$ T cells [3–9] and $CD4^+$ T cell activation is at the cutting edge of targeted anti-cancer immunotherapies [10–12].

Tumors naturally present mutated peptides – defined as neoepitopes if they can trigger an immune response - on MHC-I molecules. However, not all neoepitopes are effective targets. For instance, some are poorly processed by the proteasome (that is the main protease responsible for MHC-I-restricted peptide generation), others bind weakly to MHC-I, and some fail to elicit CD8+ T cell responses due to suboptimal presentation. A well-known example is the melanoma-associated gp100 epitope $GP100_{209\text{-}217}$ and its engineered variant T210M, which was designed to enhance binding to HLA-A*02:01 and used as a vaccine candidate due to its higher pMHC binding affinity compared to the wild-type peptide [13].

In such cases, redesigning neoepitopes in-silico can improve their immunogenicity while preserving structural similarity to the native tumor-presented peptide. Vaccination with a modified neoepitope that binds more strongly to MHC-I can prime naïve $CD8^+$ T cells that can also recognize the native, weaker-binding tumor-specific antigenic peptide. This approach mirrors the concept of mimotopes, which are engineered peptides that mimic pathogen antigens and elicit cross-reactive immune responses.

Beyond vaccines, engineered peptides can be used to design Fab fragments or antibodies that specifically recognize neoantigen–MHC complexes. These can be incorporated into CAR-T cells or bispecific T cell engagers (BiTEs) to direct immune responses toward cancer cells [14, 15]. Recent advances in de novo pMHC binder design [16] and reinforcement learning-based frameworks like PepPPO [17] have expanded the toolkit for MHC-I-targeted therapies. However, most current models are limited to pMHC-I and do not generalize well to pMHC-II, highlighting the need for more versatile approaches.

Structurally, both MHC-I and MHC-II molecules share a similar architecture. MHC-I consists of a single heavy $\alpha-$chain forming the peptide-binding cleft, while MHC-II is composed of paired α and $\beta-$chains. The cleft is formed by a $\beta-$sheet base flanked by two $\alpha-$helices, with the peptide nestled between them [18]. Upon binding, the peptide adopts a conformation with a tightly interacting core region and flanking regions that protrude outward [19, 20]. MHC-II accommodates longer flanking regions than MHC-I, whose core region often protrudes from the groove center (Figure S1). Typically, MHC-I binds peptides 8–15 amino acids in length, while MHC-II binds longer peptides ranging from 11 to 25 amino acids [19–21].

Peptide binding is driven by anchor residues—specific amino acids that interact tightly with MHC pockets and stabilize the core region. Classical anchor positions include P2 and PΩ (C-terminal) for MHC-I [22, 23], and P1, P4, P6, and P9 for MHC-II [24]. However, anchor positions can vary depending on peptide length and MHC allele, necessitating accurate prediction for effective modeling.

Modeling pMHC complexes is particularly challenging when binding affinity data is scarce or when peptides are long and structurally diverse [25]. MHC-II modeling is more complex due to its dual-chain architecture and multiple anchor residues. For MHC-I, the primary challenge lies in predicting the shape of the peptide's core region. Studies have shown that accurate anchor prediction significantly improves modeling accuracy. For example, Marzella et al. [26] demonstrated that restricting modeling to experimentally validated anchor positions reduced the average peptide-core RMSD by 1.46 Å. Moreover, mutations at anchor residues can dramatically alter peptide conformation and T cell recognition, as confirmed by crystallographic analyses [27].

AlphaFold (AF) [28], the leading tool for protein structure prediction, can accurately fold MHC molecules but struggles to dock peptides correctly [29]. To address this, several methods have been developed to incorporate anchor information into AlphaFold or fine-tune it for pMHC modeling. These include Tfold [30], which predicts anchor positions using a neural network and modifies AlphaFold's MSA and template inputs; MHC-Fine [31] and AFfine [32], which fine-tune AlphaFold on pMHC binder/non-binder classification tasks; and PANDORA [26], a fast homology modeling tool that uses anchor positions—either manually defined or predicted by NetMHCpan—for structure generation. Most of these tools support both MHC classes, though some are limited in the range of

peptide lengths they can model.

Most pMHC datasets contain sequence-level data rather than structural information. Several neural network-based models have been trained on these datasets to predict pMHC binding affinity and, indirectly, anchor preferences. For MHC-I, notable models include CapsNet-MHC [33], DeepAttentionPan [34], DeepMHCI [35], NetMHCcons [36], MHCFlurry [37], and a convolutional ranking approach [38]. For MHC-II, fewer models exist, such as Pep2Vec [39] and NetMHCpan [40], the latter being the most widely adopted. NetMHCpan is trained on both measured pMHC binding affinity and mass spectrometry-based immunopeptidome data [41], enabling broader allele coverage. While these tools perform well on MHC alleles included in their training sets, they often fail to generalize to rare or novel MHC alleles, limiting their utility in personalized medicine.

To address these limitations, we introduce Peptide MHC Generator (PMGen; `https://github.com/soedinglab/PMGen`), a unified pipeline for pMHC modeling and peptide design across both MHC classes. PMGen integrates anchor-guided AlphaFold modeling with structure-aware peptide generation. In benchmark evaluations, PMGen outperformed all other structure predictors. PMGen provides a versatile platform for immunotherapy design, neoantigen modeling, and structural immunology, bridging the gap between sequence-based prediction and structure-based optimization.

2 Results

PMGen Pipeline

PMGen consists of three core components: (1) the anchor feeding module, (2) structure prediction, and (3) peptide generation and binder selection (Figure 1). The anchor feeding module takes peptide and MHC sequences as input, predicts anchor residues with NetMHCpan, and provides these as constraints to AlphaFold2. Alternatively, anchors can be specified by the user or selected based on the highest pLDDT score. We implemented two alternative approaches for incorporating anchor information into AlphaFold: Initial Guess (IG), which initializes AlphaFold2's structure module with peptide anchor position-averaged coordinates derived from templates with aligned anchors, and Template Engineering (TE), which performs a single round of anchor-constrained homology modeling using PANDORA, where the generated homology models are supplied to AlphaFold's template module in place of real protein structures (Methods). After structure prediction, PMGen employs ProteinMPNN to sample alternative peptides that align with the predicted backbone, either in a single step or iteratively. The sampled peptides can then be selected based on their predicted binding affinity to the MHC (Figure 1B). PMGen is capable of running multiple AlphaFold model parameters in a single execution. However, for benchmarking purposes, we specifically used `model_2_ptm` due to its superior performance in pMHC structure folding (Figure S2).

Modeling Comparative Studies

In order to assess PMGen's modeling performance, we compared it with five state-of-the-art modeling tools: PANDORA [26], Tfold [30], AlphaFold Multimer [42], AFfine [32], and MHC-Fine [31]. In addition, benchmarked PMGen in its default mode IG (PMGen), with Template Engineering (PMGen+TE), and without anchor prediction, where the best structure was selected based on the highest pLDDT score (PMGen+pLDDT). For a fair comparison, we tested only on structures published after the the cut-off date of 2018-04-30 to exclude any structures used in AlphaFold2 training. Since AlphaFold3 [28] and the latest version of AF-multimer have a more recent training date cut-off than AlphaFold2, we had to use an older version (v2.2) for benchmarking.

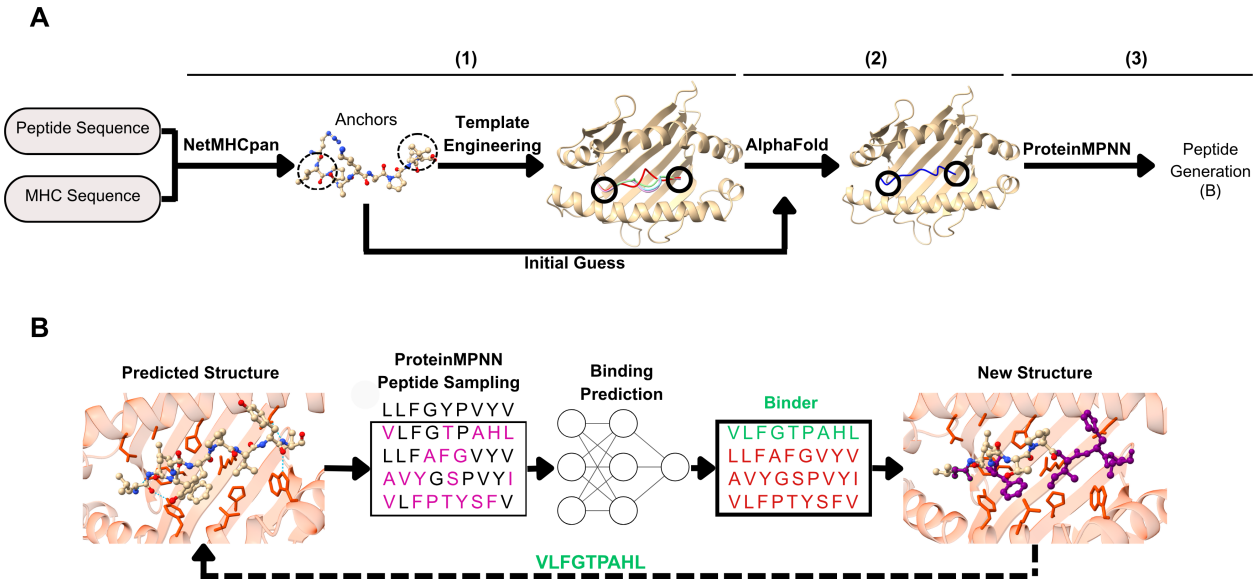

Figure 1: **Anchor-guided modeling and structure-based peptide generation in PMGen.** (A) For a given peptide–MHC pair, PMGen predicts anchor positions using NetMHCpan. Initial structural coordinates (Initial Guess) or Engineered templates (with peptides from different templates shown in color) are used to input anchor position information to AlphaFold (1). AlphaFold performs anchor aware structure prediction (2). The resulting backbone serves as input for ProteinMPNN to design alternative peptides (3). (B) Multiple peptide sequences are sampled using ProteinMPNN, and their binding affinities are predicted. The top-scoring peptide is then used for another round of structure prediction. This process can be repeated iteratively. Dashed lines represent the iterative loop, aiming to converge toward a high-affinity binder with an optimized sequence.

Across both MHC classes, PMGen significantly outperformed all other methods, achieving a peptide-Cα-RMSD (pRMSD) of 0.54 $\pm$ 0.87 Å (median $\pm$ stdev) on MHC-I and 0.33 $\pm$ 1.37 Å on MHC-II (Figure 2A). PMGen+pLDDT ranked as the second-best method, with median pRMSDs of 0.57 $\pm$ 1.12 Å on MHC-I and 0.41 $\pm$ 0.48 Å on MHC-II. Although PMGen+TE was less accurate than the initial guess mode, it still outperformed PANDORA, AFfine, and AlphaFold Multimer, and achieved performance comparable to Tfold, with pRMSDs of 1.04 $\pm$ 0.95 Å on MHC-I and 0.49 $\pm$ 1.29 Å on MHC-II.

Pairwise pRMSD comparisons between PMGen and AlphaFold-Multimer v2.2 demonstrate that incorporating anchor information into AlphaFold substantially improves structure prediction in nearly all cases (Figure 2B, left). Comparisons among PMGen variants show that the Initial Guess mode outperforms Template Engineering in more than two-thirds of cases across both MHC classes (Figure 2B, middle). Interestingly, exhaustively testing all anchor combinations in initial guess mode and selecting the best structure based on the highest pLDDT score (PMGen+pLDDT) performs almost as well as direct anchor prediction, particularly for MHC-I (Figure 2B, right). Additional pairwise comparisons further confirm PMGen's superior performance, with improvements observed in more than two-thirds of cases against all other methods across both MHC classes (Supplementary Figure S3).

Importance of Anchor Prediction in Structure Prediction

We evaluated the impact of accurate anchor residue prediction on structure prediction and assessed how closely the final predicted anchor positions matched those predicted by NetMHCpan. A previous study [27] computationally characterized anchor positioning profiles for each MHC allele, showing that peptide residues with lowest solvent-accessible surface area (SASA) strongly correlate with high anchor scores. Guided by this insight, we identified anchor positions separately for ground-truth and predicted structures by selecting residues with minimal normalized

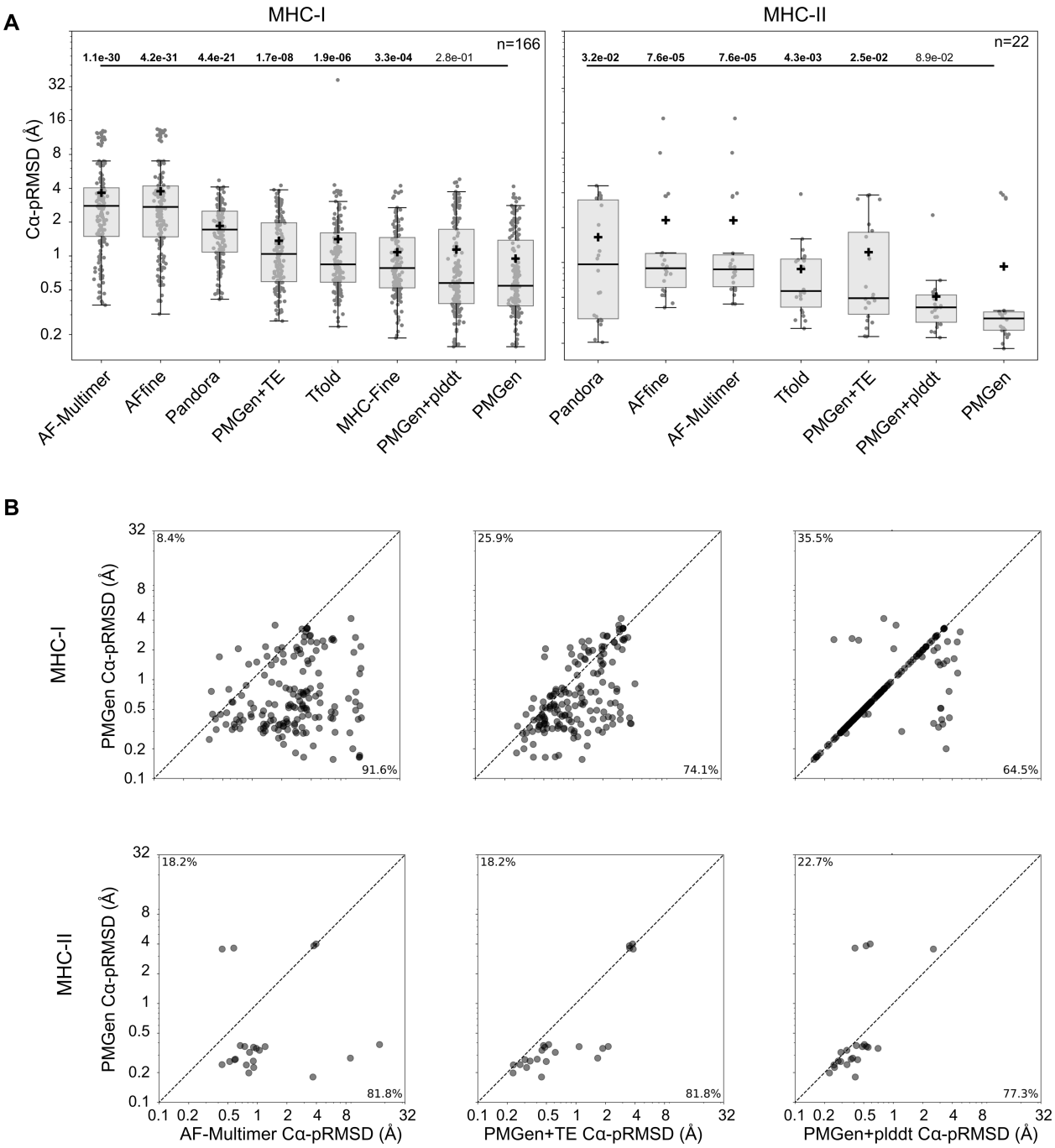

Figure 2: **Comparative Benchmarking.** (A) Modeling performance of PMGen and other methods measured by RMSD over peptide C_α atoms. Left: MHC-I; right: MHC-II. Medians and means are indicated by '−' and '+', respectively. P-values from Wilcoxon signed-rank tests comparing each method with PMGen are shown above. Significant P-values ($P < 0.05$) are shown in bold. (B) Pairwise comparisons of peptide C_α RMSD between PMGen and other methods. From left to right: AF-Multimer v2.2, PMGen+TE, and PMGen+pLDDT. Top row: MHC-I; bottom row: MHC-II.

SASA values and applying positional-distance-based criteria (see Methods). We then compared:

1. NetMHCpan-predicted anchors *vs.* structurally defined anchors from predicted structures,
2. NetMHCpan-predicted anchors *vs.* structurally defined anchors from ground-truth structures, and
3. Structurally defined anchors from predicted structures *vs.* structurally defined anchors from ground-truth structures.

Comparisons were performed using the PDB structures employed for modeling benchmark analyses. PMGen and PMGen+TE, where anchor positions were defined based on NetMHCpan predictions, showed an overall high agreement with NetMHCpan in the positioning of all four anchors in MHC-II. As expected, PMGen+plddt positioned anchors slightly differently from NetMHCpan. For MHC-II, the reported values were identical across all anchors (a1, a2, a3, and a4), since the inter-anchor spacing was defined to be uniform for MHC-II. Consequently, if one anchor was mismatched, all subsequent anchors were also affected (Figure 3A-right). For MHC-I, all PMGen variants showed lower agreement with NetMHCpan at the first anchor position, while the second anchor exhibited near-complete concordance (Figure 3A-left).

We observed that 57 pMHC-I and 5 pMHC-II PDB structures showed at least one anchor mismatch in at least one PMGen variant when compared to the ground-truth anchor positions. In only five cases did PMGen incorrectly position anchors that were correctly predicted by NetMHCpan, whereas in the majority of cases PMGen successfully corrected anchors mispredicted by NetMHCpan. Notably, for MHC-I, when at least one anchor (typically anchor 2) was correctly predicted by NetMHCpan, PMGen was generally able to identify and position this anchor correctly. This trend was also observed for PMGen+TE; however, in certain instances, even when both anchors were correctly predicted by NetMHCpan, PMGen+TE misplaced one of them, while PMGen (IG) followed the NetMHCpan predictions accurately. PMGen+plddt correctly positioned anchors in 24 cases where NetMHCpan failed to predict at least one anchor correctly, whereas in 14 cases it misplaced anchors (Figure 4B).

We further observed that correction of wrongly predicted anchor positions by PMGen was associated with an improvement in pRMSD (Figure 4C-i). In contrast, the reduced performance of PMGen+TE appeared to arise not only from mispositioning of incorrectly predicted anchors (Figure 4C-ii) but also from errors in the core conformation introduced by the engineered templates (Figure 4C-iii).

For MHC-II, neither PMGen nor PMGen+TE was able to correct anchors that were incorrectly predicted by NetMHCpan. In contrast, PMGen+plddt successfully identified the correct anchor positions in several cases where the other PMGen variants completely mispredicted them (Figure 4B–D).

Backbone- and MHC Sequence-Conditioned Peptide Generation

We integrated ProteinMPNN [43] into the PMGen pipeline to enable structure-aware peptide generation for specific target MHC molecules. Peptides were generated based on PMGen's predicted pMHC structures, the corresponding MHC sequences, and fixed (non-variable) peptide sequences. The sampled peptides were subsequently ranked according to their predicted binding affinities and eluted ligand percentile ranks. The objective was to introduce mutations into the original peptide to produce variants with comparable affinity and similar structural properties.

We employed the same PDB dataset used in the structure prediction benchmark. PMGen predictions were categorized into 51 low-quality (pRMSD > 1.0) and 55 high-quality (pRMSD < 0.6) structures. We performed three in-silico screens, mutating windows of a single amino acid (screen 1), of two (screen 2) and of three

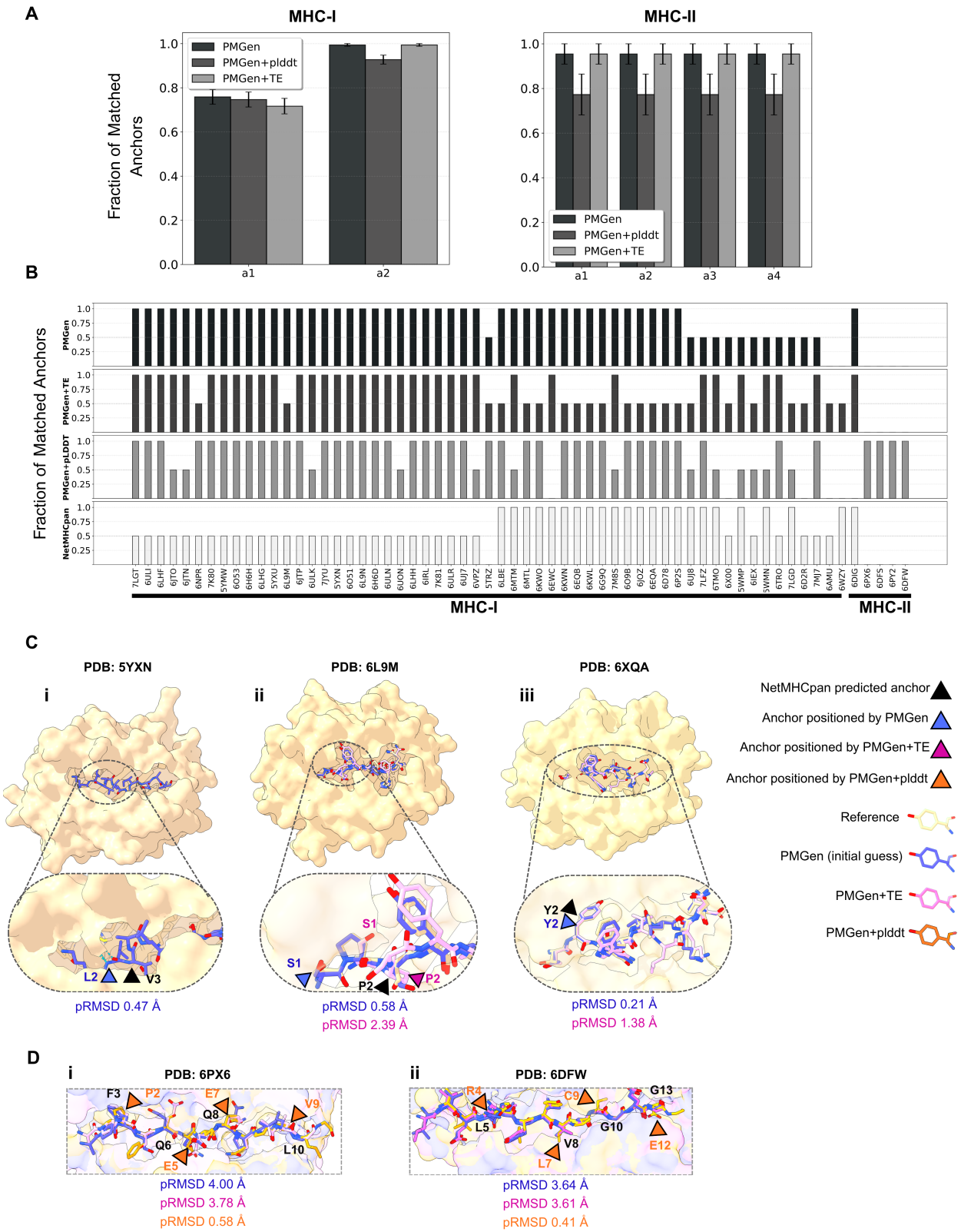

Figure 3: **Anchor residue positioning analysis.** (A) Fraction of anchors positioned by different versions of PMGen that match the anchor positions predicted by NetMHCpan. (B) Structures (PDBs) in which at least one version of PMGen positioned an anchor differently from at least one anchor predicted by NetMHCpan. The Y-axis shows the fraction of correctly positioned/predicted anchors relative to ground-truth structures. (C) Examples of PMGen structure predictions where: **i.** NetMHCpan predicted the wrong anchor, but PMGen (IG) positioned it correctly; **ii.** PMGen+TE agreed with the incorrectly predicted anchors from NetMHCpan, while PMGen (IG) positioned the anchor correctly; **iii.** PMGen+TE, PMGen, and NetMHCpan all correctly defined the anchor, but the core region in PMGen+TE was wrongly predicted due to the influence of engineered templates. (D) PMGen+plddt correctly identified the anchors using only pLDDT scores, whereas PMGen (IG) and PMGen+TE misplaced the anchors due to incorrect predictions from NetMHCpan. pRMSD values and amino acid identities are shown for PMGen (blue), PMGen+TE (pink), PMGen+plddt (orange), the reference structure (tan), and NetMHCpan (black).

consecutive amino acids (screen 3) . For each screen, we generated ten peptide variants with ProteinMPNN and selected the top three with the lowest eluted ligand percentile ranks, predicted by NetMHCpan. Additionally, all possible 20^k random mutations for screen k were generated, and their eluted ligand percentile ranks were predicted as controls.

We then conducted cumulative enrichment analyses to assess the enrichment of PMGen-sampled peptides in terms of affinity and pRMSD relative to the random mutation baseline. The results demonstrated that peptides generated by the PMGen pipeline were consistently enriched (higher AUC) among the lowest percentile ranks of the predicted affinity distribution (Figure 4A). Furthermore, when predicting the structures of a subset of random mutations and comparing their pRMSD values to those of PMGen-generated peptides, we observed that for screens 2 and 3 (involving more mutations), the selected peptides exhibited lower pRMSD values than their randomly mutated counterparts (Figure 4B).

Comparing AUC values across all PDBs revealed that high-quality structural predictions generally resulted in slightly higher affinity enrichment (Figure 4C). This difference was even more pronounced in terms of structural deviation (pRMSD) (Figure 4D). The median pRMSD values of the selected peptides for each PDB indicated that, across both MHC classes, peptides generated from high-quality structures deviated less from the original predicted conformations than those derived from low-quality structures (Figure 4E).

Collectively, these findings indicate that improved structural prediction accuracy enhances the generation of structure-aware peptide variants within the PMGen framework.

PMGen Evaluation on a Neoantigen/Wild-Type Pair Test Case

PMGen is designed for peptide generation and accurate structure prediction. To evaluate its performance, we tested its ability to (i) predict the structure of a wild-type antigen and (ii) model the structural impact of a single-point mutation. We used the test case from the TFold paper: wild-type antigen KLSHQ**P**VLL bound to HLA-A2 (PDB: 8TBW) and its neoantigen variant KLSHQ**L**VLL (PDB: 8U9G). PMGen was run in initial guess mode for both cases, and the predicted models were compared to the experimental structures.

For the wild-type complex, PMGen achieved a pRMSD of 0.8 Å, which was lower than the 1.20 Å reported by TFold (Figure 5A). The predicted peptide orientation and anchor positioning closely resembled the X-ray structure (Figure 5A-i). Residues K1, L2, S3, L8, and L9 were closely aligned with the experimental structure, while P6 and V7 showed a shift in $C\alpha$ position (Figure 5A-ii). H4 and Q5 displayed minor side-chain rotamer differences, with backbone alignment largely preserved.

For the neoantigen (P6→L6), PMGen achieved a pRMSD of 0.56 Å (Figure 5B-i). $C\alpha$ positions were closely aligned with the reference structure. The side chain of the mutated L6 in the experimental structure is oriented outward and slightly toward a MHC α-helix. PMGen predicted a similar orientation, though with a slight inward tilt. Similar to wild-type, PMGen predicted the rotamer of H4 and Q5 slightly different from the true structure (Figure 5B-ii).

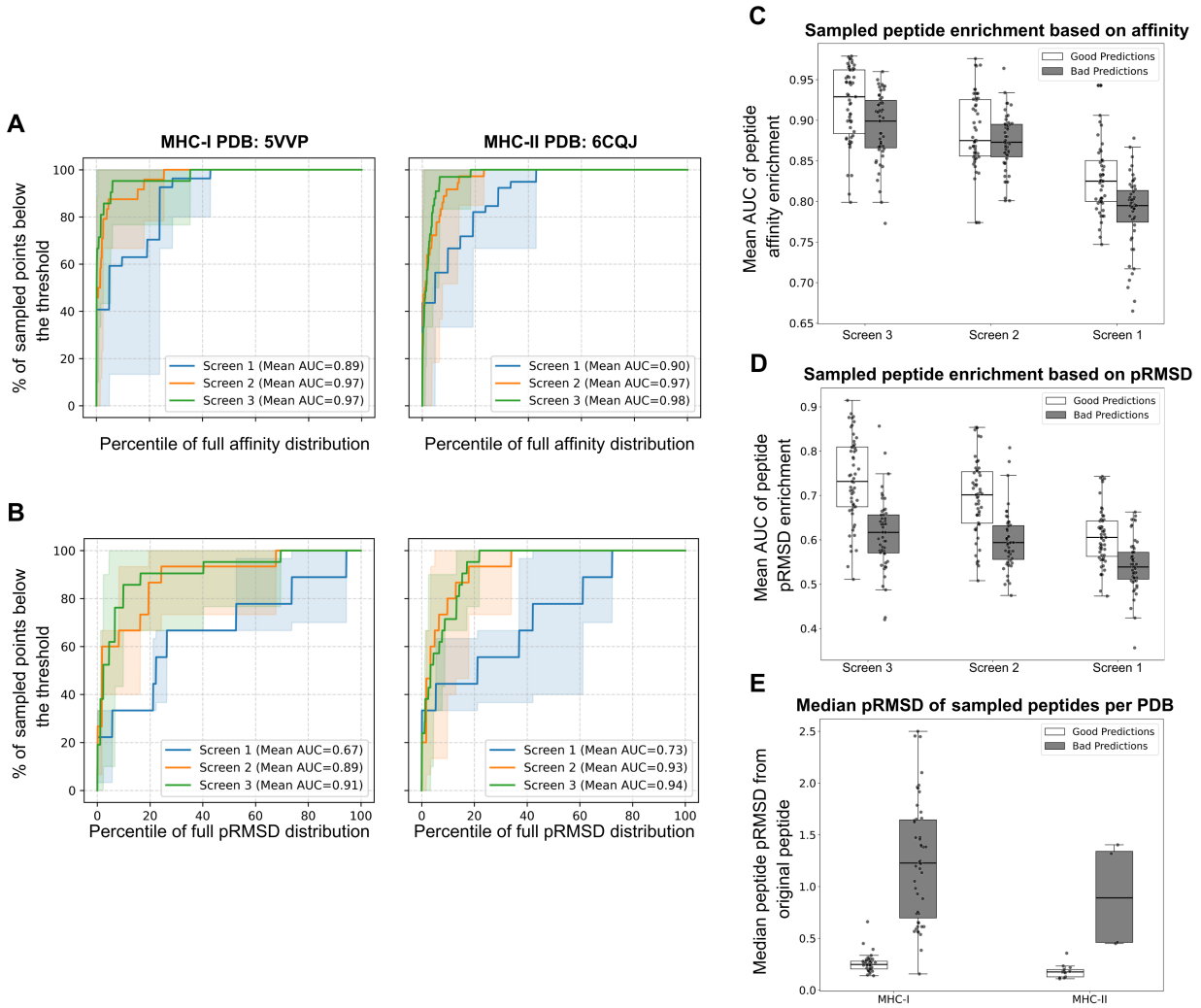

Figure 4: **Peptide generation analysis conditioned on backbone and MHC sequence.** (A) Cumulative enrichment analysis of binding affinity for PMGen-generated peptides compared to randomly mutated peptides. Screen 1, Screen 2, and Screen 3 correspond to sliding windows with one, two, and three point mutations, respectively. A higher AUC indicates greater enrichment among peptides with lower eluted ligand percentile ranks. (B) Cumulative enrichment analysis of structural similarity (pRMSD) for PMGen-generated peptides compared to randomly mutated peptides. pRMSD values are computed with respect to the original predicted structures. Higher AUC values indicate greater enrichment in peptides with lower structural deviation. (C) Comparison of affinity enrichment AUCs between high-quality (good, $n = 55$) and low-quality (bad, $n = 51$) predicted PDB structures. (D) Comparison of pRMSD enrichment AUCs between good and bad structural predictions. (E) Median pRMSD values of generated peptides across all screens, where each point represents the median deviation of generated peptides from the original predicted structure for a single PDB.

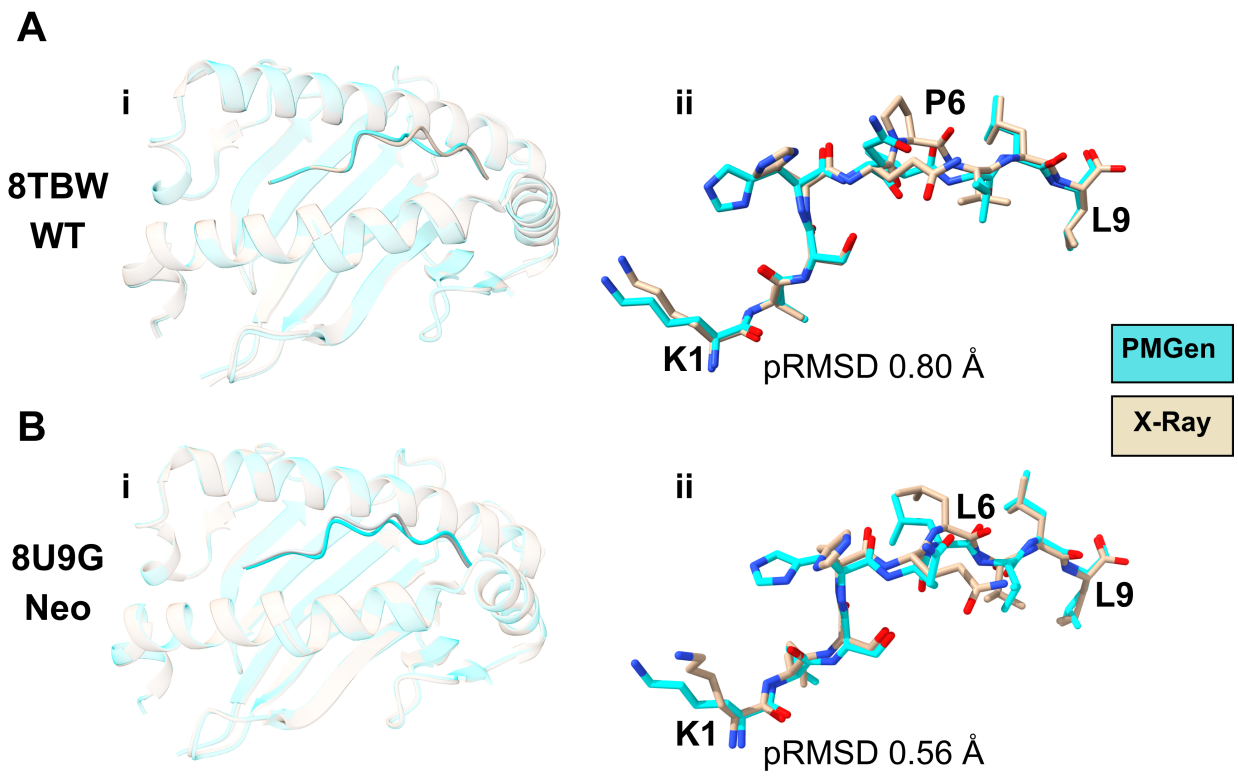

Figure 5: **Wild-type antigen vs. neoantigen modeling with PMGen** (A) Wild-type antigen KLSHQPVLL: (Left) Predicted structure (cyan) aligned with X-ray structure (amber). (Right) Peptide alignment shows $C\alpha$ atoms of almost all residues match, except P6. (B) Neoantigen KLSHQLVLL: (Left) Predicted structure (cyan) aligned with X-ray structure (amber). (Right) PMGen correctly predicts the side-chain of L6 pointing outwards.

3 Methods

pMHC PDB Data Processing

We collected PDB files containing bound-state peptide–MHC structures from the IMGT [44] and Protein Data Bank [45] databases. Structures with a reported resolution above 3.5Å were excluded. All PDB files were cleaned and renumbered, and non-amino acid molecules were removed using the Biopython PDB module [46].

For structural alignment, each chain of the collected PDB structures was aligned against the corresponding chain of our selected MHC reference structures: 4H25 for MHC-II and 4U6Y for MHC-I. Immunoglobulin-like domains ($\beta 2$ and $\alpha 2$ for MHC-II; $\alpha 3$ for MHC-I) were manually removed from the reference structures, leaving only the peptide-binding domains. Structural alignment was then performed exclusively on these peptide-binding domains using TMalign [47], after which all non-aligned MHC regions were discarded.

In cases where the PDB file contained additional chains (e.g., from T-cell receptors), only the peptide and MHC chains were retained. The peptide chain was identified as the chain with an amino acid length ≤ 25 and the smallest euclidean distance to the center of mass (COM) of the MHC binding domain.

This preprocessing resulted in 964 structures in total, comprising 116 MHC-II and 848 MHC-I complexes. We subsequently excluded 40 structures due to peptide length shorter than 8 amino acids or failure of at least one benchmarking tool to process them (Table S1). For our benchmarking, we considered only structures deposited after the AlphaFold 2.2 training cut-off date of 30 April 2018, resulting in 22 pMHC-II and 166 pMHC-I structures. The rest of the data (pre-cut-off) was considered as discovery data to find the best AlphaFold model parameters (based on overall pLDDT) to be used in PMGen.

Template Engineering

PMGen accepts a peptide and MHC sequence as input, predicts the corresponding pMHC structure and binding affinity, and optionally generates peptides. The workflow begins by identifying potential anchor positions on the peptide. These positions are either predicted by NetMHCpan or specified by the user. If the query peptide sequence is long and multiple anchor positions are predicted, the user can choose to retain the top k anchors ranked by NetMHCpan's EL and BA scores (see Methods 3). In this study, we used only the top-ranked anchor for benchmarking.

To identify the MHC allele, PMGen aligns the query MHC sequence against a locally stored list of MHC sequences with available structures and determines the closest match based on sequence similarity. Alternatively, the allele can be directly specified by the user. If the identified allele is included in NetMHCpan's accepted allele list, PMGen uses it to predict anchor positions. Otherwise, PMGen selects the most similar accepted allele (based on alignment score) solely for anchor position prediction, without substituting the query sequence.

The anchor positions, along with the query MHC and peptide sequences, are then passed to a modified version of the PANDORA pipeline [26] (see Methods 3). PANDORA performs a BLAST search against the local pMHC database and selects the top homologous structure. It then identifies the MHC anchor-binding pocket, and uses the coordinates of the peptide's anchor residues as spatial constraints for homology modeling. Anchor-constrained homology modeling is then carried out using MODELLER [48].

We refer to these anchor-constrained pMHC structures generated by PANDORA as *engineered templates*. Multiple engineered templates can be generated per query and are ranked according to MODELLER's `molpdf` score. While the peptide geometry within the MHC binding groove (anchor positions) is consistent, variation is observed in the

peptide's core and flanking regions among engineered templates. By default, PMGen generates four engineered templates for each query, which are then used as input templates for AlphaFold.

AlphaFold Initial Guess Implementation

To reduce the PMGen modeling time, we implemented an approach inspired by the AlphaFold Initial Guess method [49], which skips the template engineering step. We refer to this variant as PMGen Initial Guess (PMGen+IG).

In PMGen+IG, peptide anchor positions are either predicted (see Methods 3) or specified by the user. A BLAST search is then performed against the local pMHC PDB database (see Methods 3). The k structures with the highest alignment scores are selected as candidate templates.

An alignment file is generated containing paired MHC–peptide alignments between the query and each selected template. For the peptide sequences, anchor positions are aligned first between the query and the template; the remaining amino acids are aligned afterwards. This procedure ensures that anchor positions are transferred accurately to AlphaFold.

The aligned templates are subsequently provided as input to AlphaFold. The 3D coordinates of peptide anchor residues and all MHC residues are supplied directly to AlphaFold's structural module. In contrast, the coordinates of the remaining (core) peptide residues are explicitly set to $(0,0,0)$, allowing AlphaFold to model these positions flexibly from initial zero coordinates. These input coordinates are incorporated during the first structure-prediction step of the AlphaFold module, prior to its standard recycling iterations for prediction refinement.

AlphaFold Prediction

To predict final structures, we used the Python implementation of AlphaFold [32]. This implementation also provides a fine-tuned variant, AFfine, trained specifically for the pMHC classification task. Both implementations are based on the AlphaFold2 architecture and adapted for multi-chain prediction.

Multi-chain functionality is enabled by shifting the residue indices of each chain by 200. This prevents AlphaFold from interpreting separate chains as a continuous polypeptide. For each prediction, the query sequence was aligned to the engineered templates (see Methods 3) and supplied to AlphaFold together with only the single query sequence, without multiple sequence alignments.

The number of recycling iterations in AlphaFold was kept at the default value of three. For benchmarking, we used AlphaFold's `model_2_ptm` as well as the AFfine parameter set.

Structure Prediction Benchmarking Setting

For benchmarking with AlphaFold-Multimer 2.2, we used a local installation of ColabFold v1.5.5 [50]. Multiple sequence alignments and template searches were performed against the UniRef30 and PDB70 databases, respectively. Predictions were generated using the five default AlphaFold models, and only the top-ranked structure reported by AlphaFold was used in the benchmarking analysis.

For AFfine predictions, template PDB files were obtained from ColabFold search results. MSAs were omitted to maintain consistency with the original AFfine pipeline [32]. For PMGen+AFfine, we used engineered templates generated in PMGen pipeline. In both cases, we employed AFfine's fine-tuned version of `model_2_ptm_ft`, which was trained on the classification task of peptide–MHC binding.

Tfold [30] was executed with default settings, using its integrated SeqNN model for anchor prediction. As Tfold outputs multiple ranked structures, only the top-ranked structure generated by the tool was considered for benchmarking.

PANDORA [26] predictions were performed with default settings, using the built-in PANDORA BLAST and PDB databases for homology search and template selection. To automate anchor position assignment, we enabled the NetMHCpan option [41].

MHC-Fine [31] was executed locally with default settings. MSA generation involved jackhmmer queries to its dedicated database for MHC complexes.

For PMGen, AFfine, PMGen+AFfine, and PANDORA benchmarking, we applied a leave-one-out strategy during template engineering. Specifically, templates with the same PDB ID as the target structure were excluded during homology modeling to avoid potential bias. The same strategy was used in PMGen+IG and PMGen+IG8 during template search, ensuring that no method had prior exposure to the query protein structure during prediction.

For Multimer 2.2, Tfold, and MHC-Fine, the template search was performed internally by the respective tools, which do not implement a leave-one-out benchmarking protocol. Therefore, for these methods we relied solely on the training structure cut-off date to prevent information leakage, and identical template sequences were not explicitly excluded.

All tools were benchmarked on both MHC-I and MHC-II, except for MHC-Fine, which only supports MHC-I. To ensure fairness, deviations from the default tool settings were minimized. For AlphaFold-based methods, parameters with a training cut-off of 2018 were used by default. All tools were benchmarked on PDB structures released after this training cut-off.

Defining Anchor Residues from Structure

To identify anchor residues in each predicted pMHC structure, we used the solvent accessible surface area (SASA) as a selection criterion. SASA has been shown to be a reliable structural indicator for identifying peptide anchor positions [27]. We calculated the SASA for each residue as the total solvent-accessible surface area of its atoms, using the FreeSASA package [51].

Anchor positions were determined directly from the cleaned structures rather than from sequence conservation, ensuring that the definition was minimally affected by sequence identity. Our method selects residues that (i) have low SASA values and (ii) satisfy a minimum separation in sequence index (difference between their positions in the peptide sequence).

Let P be the set of all peptide residue positions in a given peptide–MHC complex. We define F as the set of sets S of all feasible anchor position combinations:

$$F = \left\{ S \subseteq P \,\middle|\, |S| = m, \forall p \in S: \quad SASA_p < 50, \quad \text{and} \quad \max(S) - \min(S) > n \right\}$$

The selected anchor residues A are then given by:

$$A = \arg\min_{S \in F} \sum_{p \in S} SASA_p \tag{1}$$

$$\text{with} \quad \begin{cases} m = 2, \quad n = 6, & \text{if MHC-I} \\ m = 4, \quad n = 2, & \text{if MHC-II} \end{cases}$$

Here, m denotes the total number of anchor residues to be selected, and n is the minimum sequence index distance allowed between two consecutive anchors. The residues in A are therefore those that satisfy the SASA and spacing constraints while minimizing the total SASA sum.

Evaluation of Modeling Performance

To evaluate the accuracy of the predicted pMHC structures, we first aligned the MHC chain(s) of each predicted structure to the corresponding chain(s) in the ground-truth structure. The alignment was performed using only MHC chains so that the resulting superposition was dominated by MHC structure. For MHC-II structures, both chains were treated as a single concatenated chain during alignment to preserve their relative conformation.

The rotation matrix obtained from the MHC alignment was then applied to the peptide coordinates in the predicted structure, thereby placing the peptide in the same reference frame as the ground-truth complex.

The peptide core region was defined as the segment between the first and last anchor residues, where anchors were identified separately for both the ground-truth and predicted structures according to Eq. (1). As our primary structural accuracy metric, we computed the mean RMSD of the Cα atoms within the peptide core region.

To assess the accuracy of anchor positioning, we compared the index positions of each corresponding anchor in the prediction and reference structures (e.g., anchor 1 index in ground-truth versus anchor 1 index in prediction). The fraction of matched anchors between NetMHCpan predicted anchors and predicted or ground-truth structures was calculated with the same formula. If the indices matched exactly, the anchor was counted as correct. The fraction of matched anchors was calculated as:

$$\text{Fraction of Matched Anchors} = \frac{\text{Number of matched anchors}}{\text{Total number of anchors}} \tag{2}$$

ProteinMPNN

We used ProteinMPNN [43] to sample new peptide binders for a given pMHC backbone structure. In ProteinMPNN, the encoder receives the structure as input rather than the sequence. The decoder can generate sequences either by conditioning on both the structure and existing sequence information (conditional mode) or by using only the encoded backbone (unconditional mode). We applied the conditional mode in this study, as the backbone of MHCs is highly conserved and does not provide sufficient information for peptide sampling across different alleles.

In conditional mode, we distinguish two types of amino acid positions in the structure:

- **Unfixed (designable or variable)**: Amino acid positions whose identities are not provided to the model; the decoder samples new residues for these sites.
- **Fixed (non-designable or conserved)**: Amino acid positions whose identities are provided to the model as conditioning information (along with the backbone) and are therefore not altered during sequence generation.

In PMGen, the MHC sequence is always fixed while the peptide sequence is sampled. By default, only the peptide's anchor positions are fixed. The fraction of randomly fixed non-anchor peptide positions can also be adjusted by the user to control the level of design flexibility.

NetMHCpan

In the current pipeline, we used NetMHCpan 4.1 [41] for MHC-I and NetMHCIIpan 4.3 [52] for MHC-II. NetMHCpan is applied twice during the PMGen workflow: (i) for anchor prediction prior to PANDORA homology modeling, and (ii) after ProteinMPNN peptide generation to rank candidate peptides and select binders based on their predicted binding affinity and EL score (see Methods 3).

Since NetMHCpan requires an HLA allele and peptide sequence as input, PMGen first searches for compatible alleles using a sequence alignment against known HLA alleles. Once a matching allele is identified, PMGen checks whether it is supported by NetMHCpan. If not, the most similar NetMHCpan-accepted allele is selected based on sequence similarity.

The PMGen pipeline supports two modes of anchor prediction:

1. **Multiple-anchor prediction**: NetMHCpan is run across varying peptide lengths (if the peptide is longer than 8 amino acids). Predicted anchor positions are ranked by EL score and, secondarily, by predicted binding affinity. The top k anchors (where k is user-defined) are then selected. For each selected anchor, PMGen predicts a distinct peptide–MHC structure. The best AlphaFold-predicted structure is chosen based on its average interface pLDDT score.
2. **Single most-reliable anchor**: Only the highest-ranked anchor predicted by NetMHCpan is used, and a single structure is predicted for the peptide–MHC pair.

In the benchmarking experiments reported in this paper, we applied only the second mode (single-anchor prediction).

Peptide Generation Analysis

The PDB structures predicted by PMGen were categorized into two groups based on their structural accuracy: high-quality predictions ($n = 55$, pRMSD < 0.6) and low-quality predictions ($n = 51$, pRMSD > 1.0). Three mutation screens were designed, in which one (screen 1), two (screen 2), or three (screen 3) amino acid positions were treated as variable, while the remaining residues were fixed. The variable residues were defined as contiguous linear segments within the peptide sequence.

For each structure and each screen, ten peptide variants were sampled. Their eluted ligand percentile ranks were predicted using NetMHCpan, and the top three peptides with the lowest ranks were selected. In parallel, for each mutation window, all possible random peptide variants were generated and their eluted ligand ranks were also predicted. To limit the computational cost of structure prediction, a random subset of $n = 9 \times n_{\text{mutations}}$ random peptides was selected per mutation window for each PDB in every screen. For this subset, as well as for the sampled peptides, 3D structures were predicted using PMGen, and the pRMSD values were computed relative to the original predicted structures.

Finally, for each mutation window, enrichment analyses were performed comparing the eluted ligand percentile ranks and pRMSD distributions of sampled peptides against those of random peptides. The area under the curve (AUC) was computed for each mutation window and subsequently averaged across all windows corresponding to each pMHC structure.

Main Changes in PANDORA Pipeline

To enable PANDORA to operate robustly in the best-anchor-predicted mode (see Methods 3), we modified its built-in NetMHCpan prediction logic. In the original implementation, only the best-aligned allele was used for anchor prediction. In our modified version, the top 20 aligned alleles are selected and ranked by alignment score. NetMHCpan is then executed iteratively, starting from the highest-ranked MHC sequence. If the NetMHCpan prediction is successful (i.e., the allele is present in the NetMHCpan's `allelelist.txt` file), the loop terminates and that prediction is used to determine the anchor positions.

This modification improves anchor assignment for MHC queries with low sequence similarity to known alleles, as it allows the algorithm to fall back to progressively less similar alleles until a valid prediction is found.

Additionally, we made minor changes to the `Modeling_Functions.py`, `PMHC.py`, and main `PANDORA.py` scripts to:

- fix bugs affecting NetMHCpan execution,
- improve output parsing for our pipeline, and
- save predicted anchors and alignment results as `.json` files.

Further details on these modifications are available on our GitHub repository:
`AFfine_PANDORA_modifications.txt`.

Details on Modeling Software and Prediction Time

PMGen integrates multiple external tools into a single automated pipeline, including NetMHCpan, PANDORA, AlphaFold, and ProteinMPNN. NetMHCpan can be omitted if anchor positions are provided by the user. The core PMGen pipeline is implemented primarily in Python 3.9 and Bash.

PMGen does not use multiple sequence alignments (MSAs) for AlphaFold predictions, as omitting MSAs substantially reduces inference prediction time. Furthermore, modifications to the AlphaFold2 pipeline [32] reduce PMGen's software requirements compared to the original AlphaFold2 implementation. PMGen can be executed on both CPU and GPU hardware. Within the pipeline, PANDORA and the Template Engineering step run exclusively on CPU, whereas AlphaFold and ProteinMPNN can run on either CPU or GPU.

Speed Using default settings, generation of four engineered templates via Template Engineering takes approximately 20 seconds on a single CPU core. A single AlphaFold prediction in PMGen for an average-sized pMHC complex takes approximately 9 seconds on an NVIDIA A100 GPU and 180 seconds on CPU. For the Initial Guess mode, no template engineering step is performed, and therefore its runtime is not included in this estimate. ProteinMPNN sequence generation for 10 peptides on the same GPU requires roughly 3 seconds.

The reported runtimes do not include the model compilation times for AlphaFold and ProteinMPNN, which are typically around 30 seconds.

Parallelization and Additional Tools. PMGen has built-in support for parallelized runs across multiple inputs. It can also optionally execute the BioEmu tool after pMHC structure prediction (for tasks such as further structural refinement or binding energy estimation).

4 Discussion

PMGen is a versatile framework for pMHC structure prediction and structure-guided peptide design, integrating anchor residue information into AlphaFold2[53] via two modes: initial guess and template engineering. Unlike existing tools such as MHC-Fine [31], which are limited to MHC-I and short peptides, PMGen supports both MHC classes and a broad range of peptide lengths. Benchmarking against state-of-the-art methods [26, 30, 31, 42, 54] demonstrated PMGen's superior structure prediction accuracy,0.54 Å for MHC-I and 0.33 Å for MHC-II (Figure 2A). The exclusion of overlapping templates and AlphaFold training structures during evaluation ensured unbiased performance comparisons, establishing PMGen as the top-performing pMHC structure prediction algorithm despite potential validation set overlap in competing tools.

We showed that AlphaFold can utilize the anchor information provided by both IG and TE methods without the need to change or fine-tune its parameters (Figure 2B). This is consistent with previous studies reporting that AlphaFold can effectively use contact maps and distance constraints to improve modeling [55, 56]. Moreover, other studies have shown that using multiple predicted structures as templates or initial guesses [49] can enhance structure prediction and increase the success rate in protein design [49, 57].

We observed that the initial guess strategy can generally result in better structures than the template engineering approach (Figure 2B, middle). This can be explained by the fact that when provided with multiple globally similar templates with slightly different peptide conformations, AlphaFold tends to converge on an average peptide conformation among them with higher confidence. Consequently, the final structure quality is strongly dependent on the quality of the engineered templates, which imposes a limitation. In contrast, the initial guess mode does not face this restriction, as it provides only a rough estimate of anchor coordinates relative to the MHC coordinates, thereby allowing AlphaFold to search a broader conformational space guided by the initial coordinates. This can be observed in cases where the anchor or core region were wrongly positioned by PMGen+TE while PMGen (IG) correctly predicted the structures (Figure 3C).

PMGen generally relies on the provided predicted anchors (Figure 3A). However, in many cases—particularly when up to half of the predicted anchors are incorrect—PMGen is still capable of recovering the correct anchor positions (Figure 3B). This observation suggests an inherent denoising capability of AlphaFold when processing perturbed partial conformations, consistent with previous findings [58]. Employing different PMGen modes can therefore be advantageous, as each mode produces slightly distinct structural predictions, increasing the likelihood that at least one of them corresponds to the correct conformation (Figure 3B).

Structures with correctly positioned anchors generally exhibited the highest pLDDT scores (Figure 2B, right). Consistent with this observation, a previous study also reported a negative correlation between pLDDT and pRMSD in pMHC complexes [30]. Interestingly, for four challenging MHC-II cases (6PX6, 6DFS, 6PY2, and 6DFW), where none of the other methods could identify the correct core window, relying on pLDDT (PMGen+plddt) instead of the predicted anchors successfully located and positioned the correct core (Figure 3B–D). These findings suggest that, since PMGen can leverage pLDDT to identify correct anchors, reliance on purely sequence-based anchor predictors may be reduced. Moreover, structure-based anchor-guided models, when complemented with PMGen, could further enhance prediction accuracy.

To enable structure-aware peptide generation, we integrated ProteinMPNN into PMGen, building on its proven success in T cell-specific neoantigen engineering and pMHC-I binder design [16, 59–61]. Generated peptides were ranked by NetMHCpan-predicted binding affinity and eluted ligand percentile rank, revealing consistent enrichment for high-affinity variants (Figure 4A). Critically, both affinity enrichment (Figure 4C) and structural fidelity—measured by pRMSD (Figure 4B,D,E)—correlate with initial structural prediction quality. These findings

confirm that PMGen can generate high-affinity, structurally faithful peptide variants, though performance depends critically on initial modeling accuracy. Notably, the mutation screening analysis presented here provides a practical metric for assessing prediction quality, as low pRMSD enrichment typically indicates suboptimal structural predictions.

NetMHCpan [41, 52] was selected for anchor prediction in PMGen due to its broad allele coverage and high accuracy [25, 62, 63]. Although newer interpretable models [39, 64] may offer additional benefits, NetMHCpan's training on immunopeptidome data [65] makes it well-suited for predicting peptide presentation. However, pMHC binding affinity alone does not reliably predict immunogenicity [66], highlighting the need for integrative approaches that combine structural modeling with experimental validation or reinforcement learning [67, 68]. PMGen addresses limitations in current neoantigen prediction tools, which often rely on sequence-based features or low-resolution structural models [69–72]. By providing high-quality 3D pMHC structures and enabling structure-aware peptide generation, PMGen facilitates the rational design of immunogenic neoantigens. This capability is particularly valuable for cancer vaccines, where combining MHC-I and MHC-II epitopes enhances therapeutic efficacy [73]. Recent work has demonstrated that incorporating 3D peptide-MHC-II coordinates can successfully yield experimentally validated neoantigens [74], underscoring the importance of structural information in neoantigen design. PMGen's dual-class support and structural precision extend this paradigm, positioning it as a versatile tool for expanding the repertoire of clinically actionable neoantigens. We validated this capability in a benchmark case involving SNX24-derived wild-type and neoantigen peptides (PDB: 8TBW, 8U9G), where PMGen accurately modeled both complexes and captured the mutation-induced conformational changes critical for altered T-cell recognition (Figure 5).

PMGen could contributes to cancer immunotherapy by enabling the rational design of neoantigens with enhanced pMHC binding affinity and preserved structural features. Enhanced mimotopes—engineered peptides that mimic native pMHC conformations—have shown promise in overcoming T-cell tolerance and improving immunogenicity [74–76]. Their efficacy has been demonstrated in preclinical models and early-phase clinical trials [77–81], and similar strategies have shown therapeutic potential in autoimmune diseases [82].

In conclusion, PMGen is a robust and flexible pipeline for pMHC modeling and neoantigen design. Freely available at `https://github.com/soedinglab/PMGen`, PMGen is well-suited for incorporation into experimental pipelines for cancer immunotherapy and autoimmune disease research. Future work should focus on validating PMGen-generated neoantigens in vitro and in vivo, improving interpretability, and integrating with advanced generative models to accelerate the development of next-generation immunotherapies.

References

[1] Camila RR Barbosa et al. "Mechanistic diversity in MHC class I antigen recognition". In: Biochemical Journal 478.24 (2021), pp. 4187–4202.

[2] Manuel Rojas et al. "Antigen-specific T cells and autoimmunity". In: Journal of Autoimmunity 148 (2024), p. 103303.

[3] Joseph S Dolina et al. "Linked CD4+/CD8+ T cell neoantigen vaccination overcomes immune checkpoint blockade resistance and enables tumor regression". In: The Journal of clinical investigation 133.17 (2023).

[4] Hirokazu Matsushita et al. "Cancer exome analysis reveals a T-cell-dependent mechanism of cancer immunoediting". In: Nature 482.7385 (2012), pp. 400–404.

[5] Thomas Wieder et al. "Immune checkpoint blockade therapy". In: Journal of Allergy and Clinical Immunology 142.5 (2018), pp. 1403–1414.

[6] E. Strønen et al. "Targeting of cancer neoantigens with donor-derived T cell receptor repertoires". In: Science 352 (2016), pp. 1337–1341.

[7] A. Snyder et al. "Genetic basis for clinical response to CTLA-4 blockade in melanoma". In: New England Journal of Medicine 371 (2014), pp. 2189–2199.

[8] M. M. Gubin et al. "Checkpoint blockade cancer immunotherapy targets tumour-specific mutant antigens". In: Nature 515 (2014), pp. 577–581.

[9] P. F. Robbins et al. "Mining exomic sequencing data to identify mutated antigens recognized by adoptively transferred tumor-reactive T cells". In: Nature Medicine 19 (2013), pp. 747–752.

[10] S. Kreiter et al. "Mutant MHC class II epitopes drive therapeutic immune responses to cancer". In: Nature 520 (2015), pp. 692–696.

[11] P. A. Ott et al. "An immunogenic personal neoantigen vaccine for patients with melanoma". In: Nature 547 (2017), pp. 217–221.

[12] E. Alspach, D. M. Lussier, A. P. Miceli, et al. "MHC-II neoantigens shape tumour immunity and response to immunotherapy". In: Nature 574 (2019), pp. 696–701. DOI: 10.1038/s41586-019-1671-8.

[13] Ralf Leonhardt et al. "TAP-independent presentation of the melanoma vaccine candidate epitope gp100209-217 requires no gp100 sequences outside the core peptide and is sensitive to cytosolic TPP2 degradation (100.2)". In: The Journal of Immunology 186.1_Supplement (2011), pp. 100–2.

[14] M. Yarmarkovich, Q. F. Marshall, J. M. Warrington, et al. "Targeting of intracellular oncoproteins with peptide-centric CARs". In: Nature 623 (2023), pp. 820–827.

[15] E. H.-C. Hsiue, K. M. Wright, J. Douglass, et al. "Targeting a neoantigen derived from a common TP53 mutation". In: Science 371 (2021).

[16] Bingxu Liu et al. "Design of high specificity binders for peptide-MHC-I complexes". In: bioRxiv (2024), pp. 2024–11.

[17] Ziqi Chen et al. "Binding peptide generation for MHC Class I proteins with deep reinforcement learning". In: Bioinformatics 39.2 (2023), btad055.

[18] Marek Wieczorek et al. "Major histocompatibility complex (MHC) class I and MHC class II proteins: conformational plasticity in antigen presentation". In: Frontiers in immunology 8 (2017), p. 292.

[19] Kirsten Falk et al. "Allele-specific motifs revealed by sequencing of self-peptides eluted from MHC molecules". In: Nature 351.6324 (1991), pp. 290–296.

[20] Donald F Hunt et al. "Characterization of peptides bound to the class I MHC molecule HLA-A2. 1 by mass spectrometry". In: Science 255.5049 (1992), pp. 1261–1263.

[21] Siranush Sarkizova et al. "A large peptidome dataset improves HLA class I epitope prediction across most of the human population". In: Nature biotechnology 38.2 (2020), pp. 199–209.

[22] Martin Zacharias and Sebastian Springer. "Conformational flexibility of the MHC class I α1-α2 domain in peptide bound and free states: a molecular dynamics simulation study". In: Biophysical journal 87.4 (2004), pp. 2203–2214.

[23] Marlene Bouvier and Don C Wiley. "Importance of peptide amino and carboxyl termini to the stability of MHC class I molecules". In: Science 265.5170 (1994), pp. 398–402.

[24] Roman M Chicz et al. "Predominant naturally processed peptides bound to HLA-DR1 are derived from MHC-related molecules and are heterogeneous in size". In: Nature 358.6389 (1992), pp. 764–768.

[25] Soobon Ko et al. "Benchmarking Sequence-Based and AlphaFold-Based Methods for pMHC-II Binding Core Prediction: Distinct Strengths and Consensus Approaches". In: bioRxiv (2024), pp. 2024–10.

[26] Dario F Marzella et al. "PANDORA: a fast, anchor-restrained modelling protocol for peptide: MHC complexes". In: Frontiers in Immunology 13 (2022), p. 878762.

[27] Huiming Xia et al. "Computational prediction of MHC anchor locations guides neoantigen identification and prioritization". In: Science immunology 8.82 (2023), eabg2200.

[28] Josh Abramson et al. "Accurate structure prediction of biomolecular interactions with AlphaFold 3". In: Nature 630.8016 (2024), pp. 493–500.

[29] Ameya Harmalkar, Sergey Lyskov, and Jeffrey J Gray. "Reliable protein–protein docking with AlphaFold, Rosetta, and replica exchange". In: Elife 13 (2025), RP94029.

[30] Victor Mikhaylov et al. "Accurate modeling of peptide-MHC structures with AlphaFold". In: Structure 32.2 (2024), pp. 228–241.

[31] Ernest Glukhov et al. "MHC-Fine: Fine-tuned AlphaFold for precise MHC-peptide complex prediction". In: Biophysical Journal (2024).

[32] Amir Motmaen et al. "Peptide-binding specificity prediction using fine-tuned protein structure prediction networks". In: Proceedings of the National Academy of Sciences 120.9 (2023), e2216697120.

[33] Mahmood Kalemati, Saeid Darvishi, and Somayyeh Koohi. "CapsNet-MHC predicts peptide-MHC class I binding based on capsule neural networks". In: Communications Biology 6.1 (2023), p. 492.

[34] Jing Jin et al. "Deep learning pan-specific model for interpretable MHC-I peptide binding prediction with improved attention mechanism". In: Proteins: Structure, Function, and Bioinformatics 89.7 (2021), pp. 866–883.

[35] Wei Qu et al. "DeepMHCI: an anchor position-aware deep interaction model for accurate MHC-I peptide binding affinity prediction". In: Bioinformatics 39.9 (2023), btad551.

[36] Edita Karosiene et al. "NetMHCcons: a consensus method for the major histocompatibility complex class I predictions". In: Immunogenetics 64 (2012), pp. 177–186.

[37] Timothy J O'Donnell et al. "MHCflurry: open-source class I MHC binding affinity prediction". In: Cell systems 7.1 (2018), pp. 129–132.

[38] Ziqi Chen, Martin Renqiang Min, and Xia Ning. "Ranking-based convolutional neural network models for peptide-MHC class I binding prediction". In: Frontiers in molecular biosciences 8 (2021), p. 634836.

[39] William John Thrift et al. "Pep2Vec: An Interpretable Model for Peptide-MHC Presentation Prediction and Contaminant Identification in Ligandome Datasets". In: bioRxiv (2024), pp. 2024–10.

[40] Ilka Hoof et al. "NetMHCpan, a method for MHC class I binding prediction beyond humans". In: Immunogenetics 61 (2009), pp. 1–13.

[41] 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 (2020), W449–W454.

[42] Richard Evans et al. "Protein complex prediction with AlphaFold-Multimer". In: biorxiv (2021), pp. 2021–10.

[43] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: Science 378.6615 (2022), pp. 49–56.

[44] IMGT. IMGT®, the international ImMunoGeneTics information system®. Laboratoire d'ImmunoGénétique Moléculaire. 2025. URL: https://www.imgt.org (visited on 01/20/2025).

[45] Research Collaboratory for Structural Bioinformatics. RCSB Protein Data Bank. RCSB PDB. 2025. URL: https://www.rcsb.org/ (visited on 01/20/2025).

[46] Peter JA Cock et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: Bioinformatics 25.11 (2009), pp. 1422–1423.

[47] Yang Zhang and Jeffrey Skolnick. "TM-align: a protein structure alignment algorithm based on the TM-score". In: Nucleic acids research 33.7 (2005), pp. 2302–2309.

[48] Benjamin Webb and Andrej Sali. "Comparative protein structure modeling using MODELLER". In: Current protocols in bioinformatics 54.1 (2016), pp. 5–6.

[49] Nathaniel R Bennett et al. "Improving de novo protein binder design with deep learning". In: Nature Communications 14.1 (2023), p. 2625.

[50] Milot Mirdita et al. "ColabFold: making protein folding accessible to all". In: Nature methods 19.6 (2022), pp. 679–682.

[51] Simon Mitternacht. "FreeSASA: An open source C library for solvent accessible surface area calculations". In: F1000Research 5 (2016).

[52] NetMHCIIpan-4.3 Server. Predicts peptide binding to HLA class II molecules using Artificial Neural Networks (ANNs). DTU Health Tech, Department of Health Technology. URL: https://services.healthtech.dtu.dk/services/NetMHCIIpan-4.3/ (visited on 03/23/2025). For publication of results, please cite the relevant publication (see website).

[53] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: nature 596.7873 (2021), pp. 583–589.

[54] Farzaneh M Parizi et al. "PANDORA v2. 0: Benchmarking peptide-MHC II models and software improvements". In: Frontiers in Immunology 14 (2023), p. 1285899.

[55] Kolja Stahl et al. "Modelling protein complexes with crosslinking mass spectrometry and deep learning". In: Nature communications 15.1 (2024), p. 7866.

[56] Kolja Stahl et al. "Protein structure prediction with in-cell photo-crosslinking mass spectrometry and deep learning". In: Nature Biotechnology 41.12 (2023), pp. 1810–1819.

[57] Philip Bradley. "Structure-based prediction of T cell receptor: peptide-MHC interactions". In: Elife 12 (2023), e82813.

[58] Jannik Adrian Gut and Thomas Lemmin. "Dissecting AlphaFold2's Capabilities with Limited Sequence Information". In: Bioinformatics Advances (2024), vbae187.

[59] Kiera H Sumida et al. "Improving protein expression, stability, and function with ProteinMPNN". In: Journal of the American Chemical Society 146.3 (2024), pp. 2054–2061.

[60] Robbert J de Haas et al. "Rapid and automated design of two-component protein nanomaterials using ProteinMPNN". In: Proceedings of the National Academy of Sciences 121.13 (2024), e2314646121.

[61] Kristoffer Haurum Johansen et al. "De novo designed pMHC binders facilitate T cell induced killing of cancer cells". In: bioRxiv (2024), pp. 2024–11.

[62] Yaqing Yang et al. "MHCII-peptide presentation: an assessment of the state-of-the-art prediction methods". In: Frontiers in immunology 15 (2024), p. 1293706.

[63] Thomas Karl Atkins et al. "Evaluating NetMHCpan performance on non-European HLA alleles not present in training data". In: Frontiers in immunology 14 (2024), p. 1288105.

[64] Piyush Borole and Ajitha Rajan. "MHCRule-Rule-based Model for Interpretable MHC Class I Binding Prediction". In: Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing. 2025, pp. 176–185.

[65] Jonas Birkelund Nilsson et al. "NetMHCpan-4.2: improved prediction of CD8+ epitopes by use of transfer learning and structural features". In: Frontiers in Immunology 16 (2025), p. 1616113.

[66] Vicente Machaca et al. "Transformers meets neoantigen detection: a systematic literature review". In: Journal of Integrative Bioinformatics 21.2 (2024), p. 20230043.

[67] Yicheng Lin et al. "Integrating Reinforcement Learning and Monte Carlo Tree Search for enhanced neoantigen vaccine design". In: Briefings in Bioinformatics 25.3 (2024), bbae247.

[68] Qirui Deng et al. "RLpMIEC: High-Affinity Peptide Generation Targeting Major Histocompatibility Complex-I Guided and Interpreted by Interaction Spectrum-Navigated Reinforcement Learning". In: Journal of Chemical Information and Modeling 64.16 (2024), pp. 6432–6449.

[69] Guangshuai Wang et al. "TLimmuno2: predicting MHC class II antigen immunogenicity through transfer learning". In: Briefings in Bioinformatics 24.3 (2023), bbad116.

[70] Jihyun Kim et al. "Strategies to overcome hurdles in cancer immunotherapy". In: Biomaterials Research 28 (2024), p. 0080.

[71] Jeong Yeon Kim et al. "DeepNeo: a webserver for predicting immunogenic neoantigens". In: Nucleic acids research 51.W1 (2023), W134–W140.

[72] Julien Racle et al. "Machine learning predictions of MHC-II specificities reveal alternative binding mode of class II epitopes". In: Immunity 56.6 (2023), pp. 1359–1375.

[73] Malte Roerden and Stefani Spranger. "Cancer immune evasion, immunoediting and intratumour heterogeneity". In: Nature Reviews Immunology 25.5 (2025), pp. 353–369.

[74] Jiayi Yuan et al. "EpiMII: Integrating Structure and Graph Neural Networks for MHC-II Epitope and Neoantigen Design". In: bioRxiv (2025), pp. 2025–04.

[75] Guanqiao Zhang et al. "Multi-strategies embedded framework for neoantigen vaccine maturation". In: bioRxiv (2024), pp. 2024–08.

[76] Qinglu Zhong et al. "A free energy perturbation-assisted machine learning strategy for mimotope screening in neoantigen-based vaccine design". In: Briefings in Bioinformatics 26.4 (2025), bbaf254.

[77] Jonathan D Buhrman and Jill E Slansky. "Improving T cell responses to modified peptides in tumor vaccines". In: Immunologic research 55.1 (2013), pp. 34–47.

[78] Shiqi Zhou et al. "Identification of Enhanced Vaccine Mimotopes for the p15E Murine Cancer Antigen". In: Cancer Research Communications 4.4 (2024), pp. 958–969.

[79] Xuedan He et al. "An in vivo screen to identify short peptide mimotopes with enhanced antitumor immunogenicity". In: Cancer immunology research 10.3 (2022), pp. 314–326.

[80] Fernando Martínez-Cortés et al. "Tumor antigen-unbiased variable epitope library contains mimotopes with antitumor effect in a mouse model of breast cancer". In: Molecular Immunology 157 (2023), pp. 91–100.

[81] Antoni Ribas et al. "Intra–lymph node prime-boost vaccination against melan A and tyrosinase for the treatment of metastatic melanoma: results of a phase 1 clinical trial". In: Clinical Cancer Research 17.9 (2011), pp. 2987–2996.

[82] Yi Song et al. "A mutagenesis study of autoantigen optimization for potential T1D vaccine design". In: Proceedings of the National Academy of Sciences 120.16 (2023), e2214430120.

[83] Fabian Sievers et al. "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega". In: Molecular systems biology 7.1 (2011), p. 539.

5 Abbreviations

Major Histocompatibility Complex: MHC; Major Histocompatibility Complex Class I: MHC-I; Major Histocompatibility Complex Class II: MHC-II; Cluster of Differentiation 8 Positive T-cells: CD8+ T-cells; Cluster of Differentiation 4 Positive T-cells: CD4+ T-cells; T-Cell Receptor: TCR; Regulatory T-cells: T-regs; Immunoglobulin: Ig; AF: AlphaFold; AlphaFold fine tuned: AFfine; Multiple Sequence Alignment: MSA; Amino Acid: AA; Center Of Mass: COM; Binding Affinity: BA; Bispecific T cell Engagers: BiTEs; AlphaFold Initial Guess: IG; All Variable: AV; Half Variable: HV;

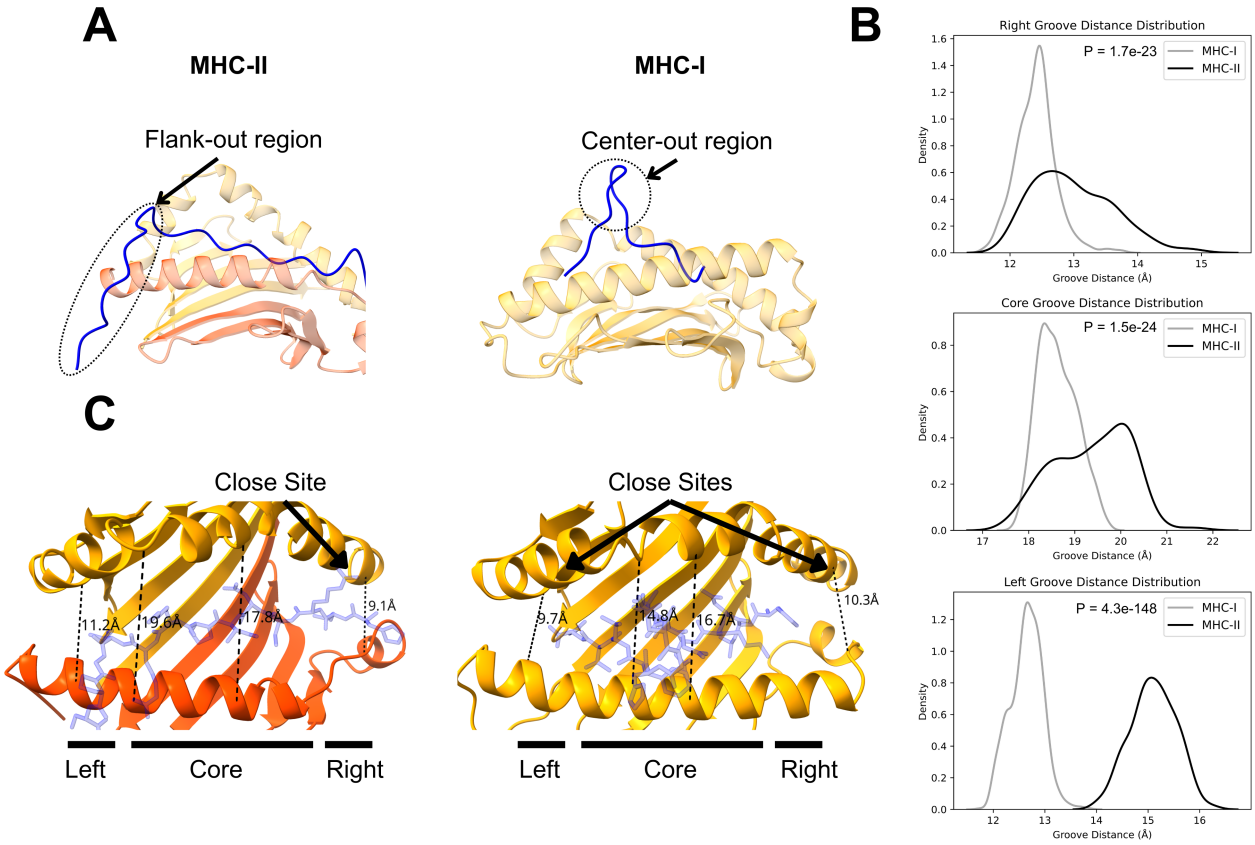

Figure S1: **Distinct binding properties of MHC-I and MHC-II.** (A) MHC-I tends to form a core-out conformation, whereas MHC-II adopts a flank-out conformation. (B) The binding groove of MHC-I has a smaller distance at both the right and left sites, while a larger central groove distance indicates the peptide's tendency to bulge outward. In contrast, MHC-II shows partial overlap with MHC-I in the right and core groove regions but exhibits complete separation in the left region due to its flank-out property. (C) Illustration of the binding groove regions in MHC structures: right (peptide entry site), core, and left (peptide exit site).

6 Supplementary Information

Binding Groove Distance Distribution We selected the longest sequences among the 964 processed structures (PDB ID: 4U6Y for MHC-I, 4H25 for MHC-II). We then manually defined the right, core, and left regions on the α structures. The right region was identified as the set of amino acids $\{4,5,6\}$ following the nearest MHC$_\alpha$ residue to the first tightly bound peptide amino acid (with a C_α distance < 4.5 Å). The left region was defined using the same rule but on the opposite side, primarily where long peptides protrude from MHC-II. The core region was considered a broader segment between the right and left regions, consisting of 10 amino acids. For each region, the nearest partner amino acids on the opposing MHC$_\alpha$ secondary structure were identified based on their sequence position and the middle amino acid's Euclidean distance. These distances were averaged for each side. For the remaining structures, we determined the right, core, and left regions by aligning their amino acids to those of the selected structures (Figure S1). Alignments were performed using Clustal Omega multiple sequence alignments (MSAs) [83].

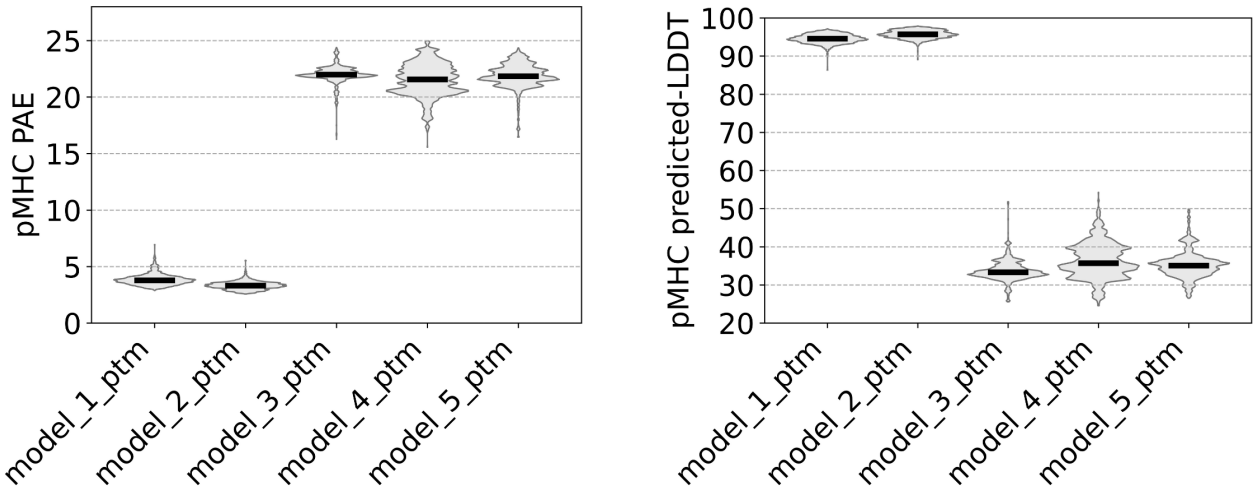

Figure S2: **AlphaFold2 performance.** The performance of different models in peptide-MHC structure prediction is visualized on the discovery set using a combination of different numbers of templates (1 to 6).

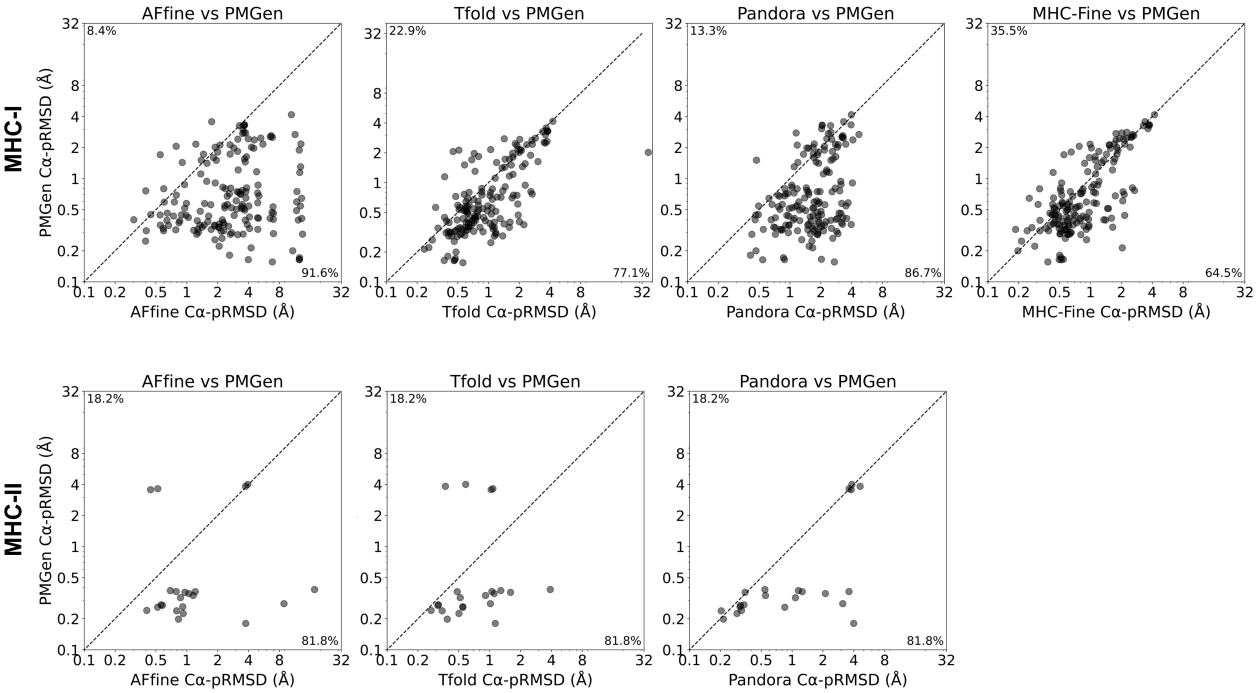

Figure S3: **Pairwise comparision between PMGen and other methods.**

Table S1: Excluded PDBs for benchmarking

3VFU	1ED3	4NO0	3VFT	3VFO	3VXU	2ICW	7MJ8	5DDH	4QRP
4JRY	6RP9	4JQX	1ZHL	6AT5	6LF9	3VFR	4JRX	4U6Y	4ZUT
4ZUV	1ZHK	3W0W	4NO2	3VFP	4ZUW	3VFW	3VFM	3KXF	3VFV
1XH3	6LF8	3BW9	6NF7	2AK4	1R5I	1ZT1	3VFN	3VFS	2OJE