RankMHC: Learning to rank class-I peptide-MHC structural models

Romanos Fasoulis[1], Georgios Paliouras[2], Lydia E. Kavraki[1,3*],

1 Department of Computer Science, Rice University, Houston, TX, United States
2 Institute of Informatics and Telecommunications, NCSR Demokritos, Athens, Greece
3 Ken Kennedy Institute, Rice University, Houston, TX, United States

* kavraki@rice.edu

Abstract

The binding of peptides to class-I Major Histocompability Complex (MHC) receptors and their subsequent recognition downstream by T-cell Receptors are crucial processes for our bodies to be able to fight various diseases. Thus, the identification of peptide antigens that can elicit an immune response is of immense importance for developing successful therapies for bacterial and viral infections, even cancer. Recently, studies have demonstrated the importance of peptide-MHC (pMHC) structural analysis, with pMHC structural modeling methods gradually becoming more popular in peptide antigen identification workflows. Most of the pMHC structural modeling tools provide an ensemble of candidate peptide poses in the MHC-I cleft, each associated with a score stemming from a scoring function, with the top scoring pose assumed to be the better representative of the ensemble. However, identifying the binding mode, that is, the peptide pose from the ensemble that is closer to a hypothetical native structure, is not trivial, and oftentimes the peptide poses characterized as best by a protein-ligand scoring function are not the ones that are the closest to the native structure. In this work, we framed the peptide binding pose identification problem as a Learning-to-Rank (LTR) problem. We developed RankMHC, an LTR-based pMHC binding mode identification function, which is specifically trained in predicting the most accurate ranking of an ensemble of pMHC conformations. RankMHC outperforms classical peptide-ligand scoring functions, as well as previous Machine Learning (ML)-based binding pose predictors. We further demonstrate that RankMHC can be potentially used in many pMHC structural modeling tools that use different structural modeling protocols. RankMHC is publicly available at [insert github link here].

Introduction

Identifying which peptide antigens bind to class-I Major Histocompability Complexes (MHCs) and further elicit an immune response after being presented to T-cell receptors has been a longstanding problem in computational immunology [1, 2]. In recent years, great strides have been made in the task of peptide antigen identification, facilitated by the development of high-throughput mass-spectrometry experiments [3, 4], which generate big datasets of amino acid sequences of MHC-bound peptides. These sequence-based datasets have in turn been used by Machine Learning (ML)-based tools, which are trained to predict pMHC binding [5, 6]. However, it is already known that there is an inherent structural component that plays a crucial role in the peptide-MHC (pMHC) interaction [7–9]. Many experimental studies have illustrated structural effects and properties in the pMHC binding interface, such as important intermolecular bonds, or the resulting solvent accessible surface, that are certain determinants of stronger MHC binding, better stability, or T-cell recognition [10–15]. Protein structure databases such as the Protein Data Bank (PDB) [16], or the IMGT-3D database [17, 18] are ever-increasing in size, with more and more experimentally determined structures being uploaded on a daily basis [19, 20]. This, along with recent successes in peptide-ligand docking tools [21, 22] and protein structural modeling methods such as Alphafold [23–26], have created an immediate need for developing successful and accurate pMHC structural modeling methodologies.

The field of pMHC structural modeling has started booming recently, following the increase in available pMHC structures in the PDB [16, 19], with multiple pMHC modeling tools and methodologies appearing in the last few years [27]. Even though pMHC structural modeling approaches are quite varied in regards to the methodologies that are being employed, most tools fall under the paradigm of sampling and scoring [28]. Specifically, given a pMHC as an input, many peptide binding poses are being generated as solutions in the output, and they are subsequently ranked with a scoring function that prioritizes energetically-feasible conformations. The binding mode identification and ranking of different generated ligand poses in a receptor is a very well-studied problem [29] and scoring functions are the most common way to rank an ensemble of ligand conformations [30]. Typically, scoring functions are split in three categories: (a) physics-based scoring functions that involve force fields [31], solvation models [32], and quantum mechanics [33], (b) empirical scoring functions that use a linear combination of hydrogen bond, hydrophobicity, and potential steric clashes information to determine the energy of a protein-ligand conformation [34], and (c) knowledge-based scoring functions that employ statistical potentials [35]. Most of the pMHC structural modeling tools in the literature employ such types of protein-ligand scoring functions. Specifically, Docktope [36], a web-based tool that uses molecular docking/energy minimization approach to pMHC structural modeling, is using Autodock Vina [37], a molecular docking tool with an empirical scoring function, in order to dock, refine, and score pMHC models. Similarly, APE-Gen [38], as well as its successor APE-Gen2.0 [39], which are two pMHC modeling tools that use a rapid pMHC structural modeling protocol, also employ empirical scoring functions such as Vina [37] and Vinardo [40] for refinement and scoring of the resulting pMHC models. PANDORA, a homology modeling-based pMHC modeling tool [41], employs the MODELLER [42] objective scoring function, molPDF, for ranking the different peptide loops that were refined by MODELLER. Lastly, pMHC structural modeling efforts that are based on the Rosetta modeling suite [43] have used different Rosetta-based scoring functions for evaluation, such as the Score12 [44], talaris2014 [45] and the newest ref2015 scoring function [46].

During the last few years, a new category of ML-based protein-ligand scoring functions has emerged [47]. Such ML-based scoring functions depend on large training datasets of protein-ligand structures and utilize non-linearity, in regards to both activation functions and feature associations, to improve protein-ligand scoring and ranking. ML-based approaches have found much success in molecular docking [48–50], as well as in the related tasks of protein-ligand binding mode identification [51–53], protein-ligand binding affinity prediction [54–56] and virtual screening applications [57–59]. In regards to ML-based scoring functions for pMHC structural models, there has been a plethora of pMHC specific scoring functions designed for predicting pMHC binding affinity [60, 61], performing pMHC virtual screening [62], even for protein deimmunization [63]. Focusing on the peptide binding mode identification task, GradDock [64] was one of the first pMHC structural modeling tools that incorporated a pMHC-specific pose ranking module. The pose ranking function is based on a linear programming approach, where the weights are optimized on the following main condition: all the generated pMHC structural models must have a higher overall energy than the ground-truth native structure. However, GradDock performs only the aforementioned native structure to pMHC model comparison, and does not perform pMHC model to pMHC model comparisons, not using the relative pMHC structural model ranking information during the optimization process. Keller et al. [65] have also created a pMHC-specific linear scoring function. They achieve this by optimizing Rosetta-derived energy terms on the Ligand Root Mean Squared Deviation (LRMSD) labels stemming from the distance between the peptide conformations from a pMHC structural model and the ground truth crystal structure. Their model of choice was a linearSVR, which exhibited better performance over other linear and non-linear regression models that were tested [65]. However, their resulting pMHC scoring function is only limited to the HLA-A*02:01 MHC allele, and only to nonamer peptides. The same linearSVR paradigm was also followed by Gupta et al. [66], one key difference being that instead of using the LRMSD, the authors use the D-score to construct their labels, a metric based on the ψ and ϕ dihedral angles [67]. While their model has shown to generalize to non-A*02:01 alleles, it is still limited to nonamer peptides.

In this work, we present a new pMHC-specific binding mode identification scoring function, RankMHC, trained to identify which pMHC binding pose from a pMHC conformational ensemble is the closest to the native structure. Inspired by the Learning-to-Rank (LTR) literature [68], we formulate the pMHC binding mode prediction identification task as a LTR problem. We demonstrate that, by using an LTR formulation, we obtain increased accuracy and better performance in regards to binding mode identification. Even though LTR has been previously employed in different structural protein-ligand related tasks such as virtual screening [69, 70] or prediction of allosteric sites [71], to our knowledge, this is the first work that applies LTR to pMHC structures specifically. RankMHC outperforms classical protein-ligand scoring functions, as well as pMHC-specific scoring functions, on different dataset splits, on unseen MHC alleles, and can accurately rank peptides of different lengths. We further show that RankMHC exhibits good generalization capabilities, and can accurately rank pMHC structural models stemming from different pMHC structural modeling tools. RankMHC is open source, and publicly available at [insert github link].

Materials and methods

pMHC crystal structure dataset

Experimental pMHC structures were downloaded from the PDB [16]. We follow the same process of pMHC structure collection as previously reported [39, 41]. Specifically, we removed from consideration any pMHC crystal structures that (i) resulted in parsing errors, (ii) exhibited missing residues on the peptide, and (iii) contained foreign molecules other than the bound peptide close to the MHC binding cleft. This filtering resulted in 566 pMHC structures that were used for structural modeling downstream. We refer to this crystal structure dataset as Q in the sections that will follow.

Modeling pMHC structures with APE-Gen2.0

For each pMHC structure we collected, we performed cross-docking for that particular pMHC pair using APE-Gen2.0 [39]. We assumed the MHC to be rigid during the modeling process, as this previously resulted in better pMHC structures that were closer to the native structure [39]. We used Vinardo [40] to locally optimize and score the peptide conformations, and we also opted in using the optional openMM [72] energy minimization step, as this has proved to result in more feasible and "protein-like" pMHC structures [39]. Lastly, we set the maximum number of pMHC conformations that APE-Gen2.0 would generate to be 100, even though, in practice, some of the pMHC conformations are filtered out mainly to anchor constraint violations. As a result of the pMHC structural modeling process, for each pMHC native structure $q_i \in Q$, we have a set of pMHC conformations of number L_i - the subscript i in L referring to the fact that for each q_i, we might a have a different number of pMHC conformations -, each one being closer or further away in regards to closeness to the native structure.

We noticed that some pMHC conformations generated by APE-Gen2.0 were very close to each other in regards to LRMSD. This, from a structural point of view, results in redundancy. From a ML point of view, this would result in very similar data points, with very similar features, and very similar LRMSD labels. To assess whether such redundancy - or lack thereof - would affect our results downstream, we created different instances of the structure dataset Q: one with redundant structures included, and one with redundant structures excluded. To automatically identify redundant pMHC structures, we developed and applied the following protocol: for each PDB code found in our crystal structure database, we (a) calculated the maximum per-residue peptide LRMSD for each pair of pMHC models, ending up in a distance matrix of LRMSD values reflecting maximum per-residue LRMSD residues for all combinations of pMHC pairs, (b) used this distance matrix to cluster the pMHC models using the HDBSCAN algorithm [73], and (c) for each cluster, we randomly selected a representative conformation. With the above process, we filter out conformations that are grouped in the same clusters, removing redundancy in the process.

Learning to Rank approaches for ranking pMHC conformations

Inspired by the LTR literature, we frame the pMHC binding mode identification problem as an LTR problem. Specifically, assume a training dataset comprising $|Q|$ pMHC instances in total. Each pMHC instance i comprises a native structure $q_i \in Q$ and L_i pMHC structural models corresponding to q_i. Each structural model q_{ij} - with $j = 1, ..., L_i$ - is defined by a set of features x_{ij} and a label y_{ij} that denotes its closeness to a native structure. Assume here that a smaller value y_{ij} is associated with a better model that is closer to the native structure (which is true for LRMSD values), but the same holds for the opposite case, without loss of generality.

The goal of LTR here is to learn a ranking function f so that, given a pMHC native structure q_i and a set of features x_{ij} derived from pMHC modeled conformations, the function f would provide a ranking prediction $\widehat{y_{ij}}$ for each conformation:

$$f(x_{ij}) = \widehat{y_{ij}} \tag{1}$$

Predictions stemming from the ranking function f should accurately rank the pMHC models in regards to closeness to the native structure q_i. During inference and testing, when the native structure q_i is not known, the ranking function f is expected to accurately rank pMHC models to a hypothetical native structure.

The way the ranking function f is learned is by specifying a loss function L, which itself defines the training objective. The LTR literature is rich with different ways and approaches for training the ranking function f, but they can all be categorized in the following three approaches:

- *Pointwise approach*: In the pointwise approach, the ranking problem is typically converted to a surrogate regression/classification problem. Emphasizing on ranking pMHC models, a pointwise approach would try to predict closeness from model q_{ij} to the native structure q_i directly using a regression-based loss. The most widely used loss function for regression objectives is the Mean Squared Error loss:

$$L_{MSE}(\widehat{y_{ij}}, y_{ij}; q_i) = \frac{1}{L_i} \sum_{j=1}^{L_i} (\widehat{y_{ij}} - y_{ij})^2 \tag{2}$$

The pointwise approach, even though it can be used as a proxy for ranking, optimizes on predicting the label y_{ij} itself, which is not the actual goal of accurately ranking pMHC conformations. Additionally, pointwise approaches do not exploit information or optimize in regards to the relationship between different objects, for example, the relative ranking order of pMHC models. It is also worth noting here that some of the previous ML-based approaches for pMHC model scoring and ranking, namely, the works by Keller et al. [65] and Gupta et al. [66] are pointwise approaches, in that, their ranking functions are trained to directly predict LRMSD/D-scores to the native structure.

- *Pairwise approach*: In contrast to pointwise approaches, pairwise approaches compare and contrast objects j and j' directly – in our case, pMHC models – by exploiting their relative closeness to the native structure:

  $$L_{Pairwise}(\widehat{y_{ij}}, \widehat{y_{ij'}}, y_{ij}, y_{ij'}; q_i) = \sum_{j=1}^{L_i} \sum_{j'=1}^{L_i} \mathbb{I}_{[y_{ij} < y_{ij'}]} \phi(\widehat{y_{ij}} - \widehat{y_{ij'}}) \tag{3}$$

  where $\mathbb{I}_{[y_{ij} < y_{ij'}]}$ is a indicator function that equals to 1 when pMHC model j is closer to the native structure than j' (or zero otherwise), and ϕ is a function designed to penalize the model when the opposite holds true for the predictions, that is, $\widehat{y_{ij}} > \widehat{y_{ij'}}$. There are many choices in regards to the selection of ϕ, with different pairwise LTR models using different ϕ functions. For example, RankSVM [74] employs the Hinge loss, whereas RankNet employs the Logistic loss [75]. Specifically, RankNet's loss function would be:

  $$L_{RankNet}(\widehat{y_{ij}}, y_{ij}; q_i) = \sum_{j=1}^{L_i} \sum_{j'=1}^{L_i} \mathbb{I}_{[y_{ij} < y_{ij'}]} \log(1 + e^{-\sigma(\widehat{y_{ij'}} - \widehat{y_{ij}})}) \tag{4}$$

  where σ is a parameter that determines the sigmoid function.

  It is worth noting here that the pMHC-specific ranking function developed by GradDock [64] employs a variation of a pairwise approach. GradDock's objective function is a pairwise ranking function, where the objective is that the predicted energy of a pMHC model must be equal or greater than the one from the native structure. Still, information in regards to the relationship of different pMHC models, namely, their relative rankings in the list, are not taken into account.

  While an improvement over the pointwise approaches, pairwise approaches do not solve all the issues related to LTR. For example, the ranking accuracy of the higher scored pMHC conformations is of greater interest that the ranking accuracy of the lower scored pMHC conformations, and a pairwise approach would equally penalize inaccuracies in both cases. Studies have shown however that pairwise approaches can oftentimes be the best performing approaches, showcasing that the objective function selection is data-dependant and problem-dependant [76].

- *Listwise approach*: Listwise approaches define loss functions that operate on the whole list of objects simultaneously, instead of using successive pairwise comparisons. There are many listwise approaches in the literature, such as ListNet [77], ListMLE [78], and SoftRank [79], among others [80], with each approach employing different algorithms and loss functions to optimize on the whole ranked list.

LambdaRank [81], a variation of RankNet, although a pairwise approach at heart, modifies RankNet so that it is list-aware, approximating this way listwise ranking functions [68, 82]. It does so by dynamically scaling RankNet's training loss with list-aware ranking metrics:

$$L_{LambdaRank}(\widehat{y_{ij}}, y_{ij}; q_i) = \sum_{j=1}^{L_i}\sum_{j'=1}^{L_i} \mathbb{I}_{[y_{ij}<y_{ij'}]} \log(1 + e^{-\sigma(\widehat{y_{ij'}} - \widehat{y_{ij}})})|\Delta Z_{jj'}| \tag{5}$$

where $|\Delta Z_{jj'}|$ is the scaling factor. The scaling factor denotes the difference between scores of the ranking metric of choice if pMHC models j and j' were to be swapped in the ranked list. In practice, the typical ranking metric of choice for scaling in LambdaRank is the Normalized Discounted Cumulative Gain (NDCG) (see definition of NDCG below), as such, $|\Delta Z_{jj'}| = |\Delta NDCG(j, j')|$ ($|\Delta Z_{jj'}| = 1$ in RankNet, where no such scaling is applied).

RankMHC

RankMHC is a ML-based model, exhibiting a LTR-inspired architecture, and is trained and designed to rank pMHC conformations by their closeness to a native structure. An illustration of RankMHC is presented in Figure 1. As an input, RankMHC receives a geometrical ensemble of peptide conformations that are bound in the MHC binding cleft. Through the featurization of the aforementioned conformations, and the ranking module that we have developed, RankMHC is able to provide a score for each conformation, and a ranking of the ensemble. The conformation that has the best score is classified as the identified binding mode. In the following two sections, we will present an in-depth description of the two main components of RankMHC: the featurization module, and the ranking module.

Featurization module

A necessary step in the RankMHC workflow is the featurization of our pMHC modeled structures, in order to provide an vectorized input to the ML architecture. We used the Rosetta modeling suite [43] to extract energy-based terms for each pMHC model. Similar to previous works on pMHC-specific binding mode predictors [65, 66] and structure-based binding affinity predictors in the literature [61], we extracted per-amino acid energy terms for each amino acid in the peptide. These energy terms are the features to be given as an input to the ranking module of RankMHC. The energy function that was used in calculating and extracting the per-amino acid energy terms is the *ref2015* scoring function, which is the default, and the latest scoring function in the Rosetta modeling suite [43]. Additionally, inspired by GradDock [64], which extracts a larger set of Rosetta-based energy terms, we also choose to extract an expanded set of energy terms that are not used in *ref2015* by default. Finally, in addition to the Rosetta-derived features, we also used NACCESS 2.1.1 [83] to extract the solvent accessible surface area (SASA), as well as the relative surface area (RSA) per residue. The default parameters of NACCESS, as well as a standard 1.4 Å radius probe for SASA/RSA calculation were used, and in general, we used protocols as previously described in [39] for the calculation of SASA and RSA. Taking the full set of features

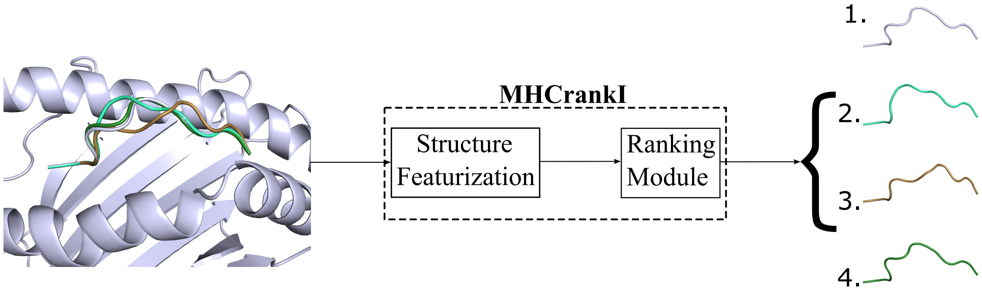

Fig 1. The RankMHC workflow. RankMHC receives as an input an ensemble of bound peptide conformations in the binding cleft, and provides a ranking in the output, with the top conformation being the most favorable one.

into account, while removing features that exhibit the same value for each conformation in the ensemble, the end result is a total of 776 features per structural model. The list of energy terms (either global or per residue features), with the accompanying description for each energy term, are shown in Table **S1**.

It is important to underline here that one of the goals of RankMHC is to expand ML-based binding mode prediction on arbitrary peptide lengths, as previous ML-based approaches in the task of pMHC binding mode identification - with the exeption of GradDock [64] - focused solely on nonamer peptides [65, 66]. As RankMHC uses an ML architecture [84] which expects input features of a fixed length (see the ranking module section below for more details on the architecture of RankMHC), we need a transformation that will convert peptides of different lengths to fixed-length vector inputs. Given the fact that nonamers are the most prevalent class of peptides that bind to class-I MHCs [85], we transform each peptide to a nonamer.

To convert each peptide to a nonamer, we perform a custom average pooling process. An illustration of this process is found in Figure **S1**. The proposed pooling works as follows: first, we perform a structural alignment of the peptide in question with a canonical nonamer pMHC template deposited in the APE-Gen2.0 template database [39]. The template selection follows protocols described in APE-Gen2.0; namely, we choose a nonamer template from the same MHC allele if possible, and as close to the peptide in question from a sequence identity perspective [39]. After the nonamer template choice and the structural alignment is performed, we match the residues of the peptide in question to the nonamer peptide distance-wise (see Figure **S1**). More specifically, we view this process as an assignment problem. We are given a set of amino acids A from the peptide in question and a set of amino acids from the nonamer template T, along with a weight function $C : A \times T \Rightarrow R$ reflecting the Euclidean distance between two amino acid pairs. We are then trying to find a bijection $f : A \Rightarrow T$ such as the total cost function:

$$C_{total} = \sum_{a \in A} C_{a,f(a)}$$

is minimized. As added constraints to the assignment problem, each amino acid $a \in A$ is matched to exactly one amino acid $f(a) \in T$ (in order to avoid multiple matches), and conversely, each amino acid $t \in T$ is matched to at least one amino acid $a \in A$. The assignment problem can also be seen as a integer linear program. Assuming a distance matrix C (corresponding to the bijection as define above), and a bipartite adjancency matrix X, where each variable $x_{a,t}$ assumes either 0 or 1, it follows that the assignment problem can be formulated as:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(a,t)\in A\times T} C_{a,t}\, x_{a,t} \\
\text{subject to} \quad & \sum_{a\in A} x_{a,t} = 1, \;\; t \in T \\
& \sum_{t\in T} x_{a,t} \geq 1, \;\; a \in A \\
& 0 \leq x_{a,t} \leq 1, \;\; (a,t) \in A \times T \\
& x_{a,t} \in \mathbb{Z}, \qquad (a,t) \in A \times T
\end{aligned}$$

To solve the linear integer problem for each peptide instance, we are using the SCIP solver [86] as provided in ortools [87]. As a result, we would have something as depicted in the top of Figure **S1**. Certain amino acids would be grouped together in nine bins, corresponding to a specific residue from the nonamer template. To finally then convert the peptide in question to a nonamer peptide, we perform an average pooling operation on the features of the amino acid groups in each of the nine bins, which will result in nine distinct feature groups. These groups are then concatenated to form the final feature vector.

We wanted to assess the effectiveness of our proposed custom average pooling operation, especially in comparison to other approaches in the literature. Amino acid sequence-based binding affinity predictors either identify the nonamer binding core of the peptide, converting all peptides to 9-mers in the process [5], employ a form of padding with a neutral amino acid 'X' to bring every peptide to the same length [88], or use a deep learning architecture that allows for peptide/feature inputs of variable length [89]. To assess the effectiveness of our custom average pooling operation, we employed neutral amino acid 'X' padding as previously described and used in MHCFlurry [88]. For the neutral amino acid, we set all per residue energies to 0. Such padding does convert the peptide to the same length, but it is worth noting that the feature space ends up being larger due to the padding, which makes the model more difficult and computationally intensive to train in the process.

Finally, as an additional ablation study, we wanted to assess the performance of feature-based variants of the baseline RankMHC feature set. More specifically, we created three additional models: (a) one that considers solely the *ref2015* terms as previously used [61, 65, 66] and not the expanded feature set as seen in Table **S1** (b) one that also includes energy terms from selected residues of the MHC - previously defined as the MHC pseudosequence -, as done in state-of-the-art pMHC binding affinity prediction models [5, 6], and (c) one that considers, instead of per amino-acid energy terms, the energy terms stemming from the intramolecular interactions between the peptide and the MHC directly. For this last case, to find the set of intramolecular interactions that are occurring between the peptide and the MHC, we calculated the set of intramolecular interactions from our crystal structure database. Namely, following the work in [90], for each pMHC crystal structure with a nonamer peptide, we calculated all the intramolecular interactions between the peptide and the MHC residues that are within 4.0 Å of each other. We filtered out the intramolecular interactions that are happening rarely, namely, we deleted the interactions that are happening less than 10% of the time in the crystal structures. The resulting contact map largely resembles the one found in the study of Nielsen et al. [90], and as such, we used the energy terms stemming from these interactions to featurize our pMHC structures.

Ranking module

As previously mentioned, RankMHC is based on LTR methodologies. More specifically, we use the LambdaMART algorithm [91] as the backbone of RankMHC. LambdaMART combines RankNet/LambdaRank [81] (see also above), with MART [92], a gradient boosting framework that uses regression trees and employs gradient descent during training. Even though different metrics can be used for training LambdaMART as scaling factors $|\Delta Z|$, we opted in either using $|\Delta Z_{jj'}| = 1, \ \forall (j, j')$ for training purely on pairwise comparisons, or the NDCG metric ($|\Delta Z_{jj'}| = |\Delta NDCG(j, j')|$) for listwise awareness, the latter being on par with the original LambdaMART publication [91]. Finally we used the XGBoost framework and package [84] for RankMHC, as it offers an efficient implementation of LambdaMART.

To train RankMHC, we used the previously mentioned pMHC database Q. We follow a nested Cross Validation (CV) approach as previously described [93]. Specifically, for the model evaluation step, we partition the dataset into train/test splits using a 6-fold CV. For the hyperparameter tuning/model selection step, we further partition the 6 train portions into train/validation splits in an internal 5-fold CV. The XGBoost hyperparameters that were chosen for optimization, as well as the hyperparameter values that were tested can be found in Table **S2**. After choosing the optimal hyperparameters that lead to the best performance on the validation sets, the models are evaluated on the left-out test sets, leading this way to an unbiased evaluation. As the external CV is a 6-fold CV, the final model is an ensemble of 6 different XGBoost instances having trained on different subsets of Q.

Based on the factor that determines the external and internal splits, one can test different aspects of the generalizability of a model. For testing different generalizability aspects of RankMHC, we opted in using different nested CV schemes. The number of partitions and splits are kept the same for all different nested CV schemes. The proposed nested CV schemes are as follows:

- *Leave-K-PDBs-Out Cross Validation (LKPO-CV)*: The LKPO-CV split refers to random external/internal splits of the dataset, making sure that no PDB code is found in both the train and test splits at the same time. This corresponds to the most realistic scenario of testing RankMHC to MHC alleles and peptide lengths that are previously seen in the training dataset.

- *Leave-K-Alleles-Out Cross Validation (LKAO-CV)*: Here, we perform the external/internal CV splits so that no MHC alleles from the test sets are found in the external/nested training sets. This corresponds to a scenario where RankMHC encounters a previously unseen MHC allele when trying to identify the correct peptide binding mode. This particular scenario is intentionally pursued, as previous studies have demonstrated loss of generalization for unseen MHC alleles [65].

- *Leave-One-Length-Out Cross Validation (LKPO-CV)*: Most previous ML-based approaches for pMHC binding mode identification focus specifically on nonamer peptides [65, 66], as they are the most abundant in the pMHC repertoire [85], and the most abundant in regards to public datasets. RankMHC on the other hand is designed for peptides of arbitrary length. Therefore, to test the generalization capabilities of RankMHC in regards to peptide lengths that have not been encountered during training, we perform the nested CV validation so that, for each split, the external test datasets contain one of the 6 available lengths (from 8 to 13 residues, and the main motivation behind the proposed external 6-fold CV), while the external training splits contain all the rest. The same paradigm applies also to the internal 5-fold CV split.

Modeling pMHC structures with PANDORA

We sought to also prove that the trained RankMHC model can also be used, not just on APE-Gen2.0-generated pMHC models, but also, on generated models stemming from a different tool. To this end, we modeled the same collected set of pMHC structures Q with PANDORA [41]. We used the defaults parameters of PANDORA during modeling, with the maximum set of generated conformations to be 20 in total. We excluded pMHC pairs that PANDORA could not model due to allele name mismatches. In total, 405 pMHC structures were modeled using PANDORA, with a resulting 20 pMHC models for each pMHC structure.

We subsequently used RankMHC to score and rank the 20 models for each pMHC structure. It is worth noting that, to avoid data leakage, we used the RankMHC model instance from the LKPO experiment that had not seen the pMHC pair to be scored previously during training. As such, our scoring of PANDORA pMHC models with RankMHC is unbiased.

Evaluation metrics

To assess the effectiveness of RankMHC over different scoring functions that have been employed in pMHC binding mode identification, we consider several evaluation metrics, each one emphasizing different performance characteristics. Specifically, the intersection of molecular docking and scoring and information retrieval methodologies that characterize RankMHC prompted us to combine different evaluation metrics from the two fields. Below we describe these evaluation metrics in more detail, emphasizing on their different characteristics and goals:

Top-1 Ligand Root Mean Squared Deviation (LRMSD@1)

Evaluating pMHC structural models using the LRMSD to the native pMHC structure is commonplace in the pMHC structural modeling literature [36, 38, 41]. As such, we also adopt the LRMSD formulation to evaluate how close to the native crystal structure is the top-ranked pMHC conformation as ranked by the scoring function at hand. Specifically, the LRMSD is defined as follows:

$$\text{LRMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i}$$

where N denotes the number of atoms in the peptide bound to the MHC cleft, and d_i is the 3D Euclidean distance between an atom from the peptide found in the pMHC structural model and the same exact atom found in the native structure. Here, we consider the full-atom LRMSD, including the atoms found in the side-chains of the peptide residues.

Most pMHC structural modeling tools emphasize the top scoring conformation as the representative and most accurate peptide binding mode. As such, we consider the LRMSD to the native structure only from the top-scoring conformation (@1). Finally, as in our crystal structure dataset we have many pMHC structures q_i, the average LRMSD@1 of the whole dataset is calculated, in order to assess the accuracy and efficacy of a scoring function.

Mean Reciprocal Rank (MRR)

The reciprocal rank denotes the *inverse* rank of the best pMHC model of the ensemble - best here denoting the closest model to the native structure q_i in terms of LMRSD - in the ranked pMHC conformation list. As an example, if the scoring function of choice places the actual best conformation as second best, then it's reciprocal rank would be 1/2. The best value for the reciprocal rank is 1 (where the best scored conformation is the actual best), with $\frac{1}{L_i}$ being the worst reciprocal rank value when the best conformation receives the worst score.

The Mean Reciprocal Rank (MRR) is then calculated by simple aggregation through the crystal structure dataset Q and calculation of the mean:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{6}$$

Top-1 Precision (P@1)

The Top-1 Precision (P@1) is an indicator function, returning 1 when the top ranked pMHC conformation is the actual best, and 0 otherwise. As such, the average top-1 precision is a number between 0 and 1 denoting the likelihood of the best pMHC conformation being identified by the scoring function of choice.

Spearman's correlation co-efficient (ρ)

So far, the three previous evaluation measures that were introduced are operating on binary levels of relevance, meaning that they consider only the top ranked conformation to be relevant. However, for some cases, information about the correctness of whole ranked list can also be relevant.

The Spearman's correlation co-efficient (ρ) measures the monotonicity of the relationship between two variables, defined as the Pearson's correlation coefficient between ranks [94]:

$$\rho = \frac{cov(R(X), R(Y))}{\sigma R(X) \sigma R(Y)} \tag{7}$$

where $R(X), R(Y)$ are the ranks of variables X, Y, cov denotes the covariance between the ranks, and σ denotes the standard deviation of the ranks. Specific to ranking pMHC conformations, ρ can be employed to assess whether the scoring function is able to properly rank the whole ensemble of conformations.

As before, we are calculating the average ρ stemming from the different ρ's we are obtaining for different $q_i \in Q$. However, considering the average of correlations has proven to lead to underestimation [95]. Therefore, we apply the following correction $G(\rho)$ as found in Olkin and Pratt [96]:

$$\frac{G(\rho)}{\rho} = 1 + \frac{1-\rho^2}{2(n-3)}, n > 4 \tag{8}$$

where n equals to the sample size. Even though the transformation refers to Pearson correlation coefficients, treating ρ's as Pearson correlation coefficients before transformations is considered robust [97]. Finally, as the aforementioned transformation has a sample size requirement, we omit q_i datapoints for calculation of ρ when $L_i \leq 4$.

Normalized Discounted Cumulative Gain (NDCG)

The ρ metric assesses the correctness of the ranking taking into account all the pMHC conformations from the ensemble. However, many application, such as virtual screening, consider the top-k results of a ranked list of different targets [98]. In such tasks, the weight falls mostly on the top-k ranking conformations. pMHC conformational scoring and ranking is no different, as emphasis is mostly given on the top scoring conformations for downstream tasks. The Discounted Cumulative Gain (DCG) [99] is a metric that pays more attention to the ranking efficacy of higher ranked objects by introducing a logarithmic factor that reduces relevance according to an object's rank:

$$DCG = \sum_{j=1}^{L_i} \frac{rel_j}{\log(j+1)} \tag{9}$$

where rel_j is the graded relevance of the object at position j. The Normalized Discounted Cumulative Gain (NDCG) is simply a normalized version of DCG:

$$NDCG = \frac{DCG}{maxDCG} \tag{10}$$

where the denominator is the ideal, maximum DCG that one can get by perfectly ranking a set of objects by their relevance.

Normally, relevance values in ranking tasks are integer values ranging from 1 to 5, with higher relevance being better [68]. For the task of pMHC binding mode identification, we use the full-atom LRMSD values directly as relevance values. As LRMSD values closer to zero are better, we reverse the sign of LRMSDs to negative so that higher LRMSD is better. However, negatives relevance labels end in NDCG being unbounded. For this reason, we apply the following normalization as proposed in [100]:

$$NDCG = \frac{DCG - minDCG}{maxDCG - minDCG} \tag{11}$$

This normalization of NDCG works with negative labels, and results in a bounded, 0 to 1 NDCG value, without sacrificing its statistical power. Additionally, as before, as we have many $q_i \in Q$ pMHC instances, the average NDCG is considered in order to compare scoring functions.

Results

RankMHC outperforms other scoring functions on pMHC pose ranking

The workflow of RankMHC can be seen in Figure 1. As an input to RankMHC, an ensemble of pMHC conformations are given. The conformations are subsequently transformed into fixed-length feature vectors comprising global and per-residue energy terms. These features pass through the ranking module of RankMHC, which, based on the feature content, ranks the conformations in regards to LRMSD-based closeness to a hypothetical native structure. To achieve this, RankMHC was trained on a set of pMHC crystal structures and pMHC structural models created using APE-Gen2.0, a rapid pMHC structural modeling tool [39]. As an output, RankMHC provides a ranking of the peptide conformational ensemble, with the top conformation assumed to be the correct peptide binding mode from the provided ensemble.

We trained and evaluated the performance of RankMHC using different nested CV schemes that exhibit different generalization aspects of RankMHC: (A) LKPO-CV that reflects the performance of RankMHC when the MHC allele type and the peptide length has been previously seen in the training data, (B) LKAO-CV, that reflects the performance of RankMHC when an MHC allele is encountered that has not been previously seen during training, and (C) LOLO-CV that reflects the performance of RankMHC for a peptide length that has not been previously seen during training. We benchmarked the performance of RankMHC in these different nested CV schemes with different types of scoring functions that have been previously used for ranking ensembles

Table 1. Benchmark of different scoring functions on pMHC binding mode identification. Methods are categorized into three parts: pMHC specific scoring functions that operate on auxiliary tasks that are correlated to the task of binding mode identification, general protein-ligand scoring functions that are not pMHC-specific, and pMHC-specific binding mode identification functions. Higher ρ, MRR, P@1 and NDCG, as well as lower LRMSD@1, denote better performance. The best performing method is depicted in bold, while the second best performing method is underlined.

	LKPO-CV					**LKAO-CV**					**LOLO-CV**				
Methods	ρ	MRR	P@1	NDCG	LRMSD@1	ρ	MRR	P@1	NDCG	LRMSD@1	ρ	MRR	P@1	NDCG	LRMSD@1
Abella et al.	-0.049	0.402	0.254	0.422	2.440	-0.049	0.402	0.254	0.422	2.440	-0.049	0.402	0.254	0.422	2.440
3pHLA	0.010	0.474	0.187	0.510	2.563	0.010	0.474	0.187	0.510	2.563	0.010	0.474	0.187	0.510	2.563
ref2015	0.394	0.604	0.512	0.756	1.930	0.394	0.604	0.512	0.756	1.930	0.394	0.604	0.512	0.756	1.930
molPDF	0.095	0.498	0.251	0.580	2.428	0.095	0.498	0.251	0.580	2.428	0.095	0.498	0.251	0.580	2.428
vina	0.429	0.634	0.519	0.797	1.910	0.429	0.634	0.519	0.797	1.910	0.429	**0.634**	0.519	**0.797**	1.910
vinardo	0.426	0.622	0.516	0.776	1.945	0.426	0.622	0.516	0.776	1.945	0.426	0.622	0.516	0.776	1.945
GradDock	0.277	0.565	0.327	0.690	2.276	0.277	0.565	0.327	0.690	2.276	0.277	0.565	0.327	0.690	2.276
LinearSVR	0.473	0.641	0.516	0.791	1.904	0.465	0.632	0.516	0.786	1.915	0.448	**0.634**	0.514	0.782	1.926
RankMHC	**0.552**	**0.679**	**0.555**	**0.817**	**1.860**	**0.494**	**0.663**	**0.571**	**0.807**	**1.888**	**0.483**	0.630	**0.539**	0.795	**1.893**

of pMHC conformations: the ref2015 scoring function stems from the Rosetta modeling suite [43], and has been previously used in ranking Rosetta-based pMHC structural models [46]; MODELLER's [42] objective function, molPDF, is the scoring function used in PANDORA [41], a homology-based pMHC structural modeling tool; APE-Gen2.0 uses vina [37] and vinardo [40] to rank the generated ensemble of pMHC structural models [39]. We also benchmarked RankMHC with other ML-based pMHC ranking functions, namely, the one offered by GradDock [64], as well as the linearSVR ranking functions developed by Keller et al. [65] and Gupta et al. [66]. For the later ones, as the training datasets were substantially different, we opted in developing our own linearSVR, which was on the same dataset and following the same protocols as RankMHC. Specifically, the 6-fold nested CV protocol was used to tune the regularization hyperparameter C, as described in Keller et al. [65]. We found a large agreement in the optimal values of C with the Keller et al. study [65] (data not shown). Finally, we also include in our benchmark the structure-based pMHC binding affinity predictor 3pHLA [61], as well as the structure-based random forest model for pMHC virtual screening developed by Abella et. al [62]. This was done in order to assess how ML-based scoring functions, that have not been explicitly trained in the task of pMHC binding mode identification but in surrogate and correlated tasks of pMHC binding/pMHC affinity prediction, can perform in the task of pMHC binding mode identification.

The results of the full benchmark can be seen in Table 1. The first observation is that ML-based approaches, such as 3pHLA [61] and the random forest by Abella et al. [62], which are not explicitly trained in the task of pMHC binding mode identification, but on a related task, are not effective in ranking pMHC models. This is evident by the very close to zero ρ. As far as classical, protein-ligand scoring functions are concerned, vina, used in APE-Gen2.0, is the most efficient in ranking pMHC structural models. Surprisingly, exluding molPDF, all general protein-ligand scoring functions surpass GradDock, which is a pMHC-specific ranking function. The LinearSVR scoring function outperforms general protein-ligand scoring functions on most metrics in the LKPO-CV, consistent with findings in [65, 66]. However, when looking at LKAO-CV and LOLO-CV performances, the LinearSVR regressor loses a fair amount of generalization capability. This is also something that was observed in [65] when the authors applied their LinearSVR on alleles other than the HLA-A*02:01. In contrast to other approaches, RankMHC retains the top performance in almost all evaluation metrics and all nested CV schemes. This highlights the efficacy of the LTR training regime, showcasing that

LTR is a much more natural fit for ranking-specific tasks.

It is worth noting that the results shown in Table 1 do not include redundant pMHC structures as these were filtered out during evaluation (see Methods for more information). Results including redundant structures can be seen in Table **S3**. Even though P@1 and MRR are significantly reduced for all methods, ρ, NDCG and LMRSD@1 all remain similar to the non-redundant test set. We hypothesize that this happens because many of the methods identify a binding mode which is very close to the best one in regards to LRMSD, but not the actual best, due to the existence of redundant structures. However, the relative comparisons between methods remain the same, with RankMHC retaining the top performance throughout different metrics and nested CV schemes.

Ablation/Interpretability studies

Given the success of RankMHC on the main benchmark as seen in Table 1, we sought to answer what are the determining features and design choices that contributed to such performance. To this end, we performed specific ablation studies that demonstrate the increases/decreases in the performance of RankMHC given the existence/absense of specific features/design choices. The full results of the ablation studies can be seen in Figure 2A.

To begin with, in regards to the training set content, we wanted to test whether introducing/excluding redundancy affects the performance of RankMHC. Removing redundant structures from the dataset (denoted as *RankMHC - redundant structures* in Figure 2A) reduces performance of RankMHC in all metrics. Even though data augmentation has proven to be beneficial in the field of ML, overloading the dataset with augmented data that exhibit a specific pattern could always lead to a ML model overfitting a particular pattern, hampering performance and accuracy as a result. However, RankMHC handles the augmented pMHC models well, without loss of generalization. Additionally, instead of the proposed average pooling approach that reduces peptides of different lengths to nonamer peptides (see Methods), we wanted to see whether feature padding with the neutral amino acid 'X' as previously proposed [88] could result in better performance (see Methods on how the padding is performed). When introducing padding, RankMHC exhibits slightly worse performance in three of the five employed metrics (see Figure 2A). Padding in this case results in a bigger feature space and a longer training process that does not necessarily lead to better performance downstream.

We also employed different feature sets to assess whether RankMHC benefits in the existence/absence of certain features. First, we created an instance of RankMHC where the features that are used are only the ones that stem from the ref2015 scoring function from the Rosetta suite, excluding the additional features as seen in Table S1 (denoted as *RankMHC - additional features* in Figure 2A). This is consistent with other works in the literature that have developed pMHC-specific binding mode identification functions [65, 66]. This however resulted in subpar performance across all five metrics (see Figure 2A), demonstrating that the augmented set of Rosetta-derived features (see Table S1 and [64]) contributes to the good performance of RankMHC. Secondly, in addition to the per peptide position feature set, we also include features from the MHC residues, specifically, the residues derived from the previously defined MHC pseudosequence [5] (denoted as *RankMHC + MHC features* in Figure 2A). This resulted in worse performance in three out of five metrics, meaning that MHC residue features are not crucial to the performance of RankMHC. Finally, instead of per peptide residue or per MHC residue features, we attempted to introduce the pairwise features (denoted as *RankMHC + pairwise features* in Figure 2A) stemming directly from the pairwise interactions between the peptide residues and the MHC residues in the peptide-MHC

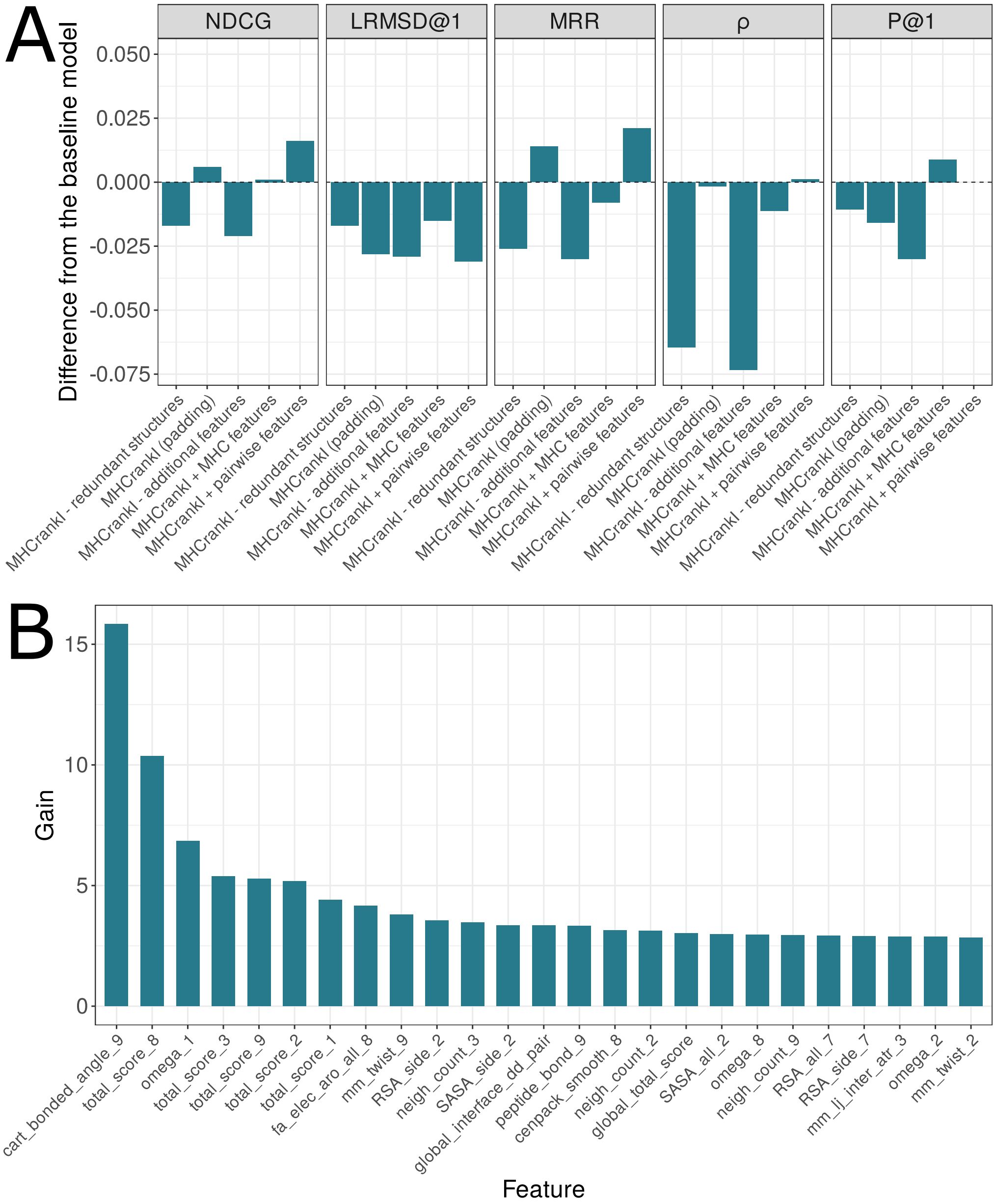

Fig 2. Ablation/Interpretability studies. (A) Comparison of the baseline RankMHC model to other RankMHC instances across five different metrics. The x-axis enumerates the different RankMHC instances, while the y-axis shows the difference between the baseline RankMHC performance and the performance of a particular RankMHC instance. Any bars extending below the $y = 0$ dashed line means that the baseline RankMHC is better, and vice versa. **(B)** Feature Importance for difference features employed by RankMHC. The x-axis denotes different features (including the particular residue position, or the keyword global if it realtes to the entire pMHC complex). Feature names are directly taken from the Rosetta suite [43]. The y-axis denotes the importance gain derived from calculating by taking each feature's contribution for each tree in RankMHC. Only the twenty-five most important features are shown.

binding cleft (see Methods). The resulting feature set exhibits better performance than the baseline RankMHC in three metrics. As such, if a user is interested in employing the top-k pMHC conformations from the RankMHC ranking, using the pairwise features might result in a more accurate top-k ranking, given the better NDCG and MRR scores when compared to the baseline RankMHC. However, interestingly enough, pairwise features also exhibit the worst LRMSD@1 performance out of all the models that we created for the ablation study (see Figure 2A), making the model not particularly suitable if one is interested in the top-1 conformation only. All of the aforementioned RankMHC instances will be provided to the user and can be found in the RankMHC repo [insert repo here]. Depending on the metric that the user is interested in, they can choose which model is the most appropriate for their downstream tasks.

We also experimented with altering the loss function of our XGBoost models. Specifically, we used a pointwise loss function that attempts to directly predict the LRMSD to the native structure, the pairwise approach of RankMHC, and a pairwise, but list-aware loss function, akin to LambdaRank [81]. This way, we tried to cover all three facets of LTR (see Methods for a detailed description for each of the three approaches). Results are shown in Supplementary Table 4. Adopting a list-aware loss function does not result in better performance, with the pairwise approach exhibiting the better results. While initially this seems counter-intuitive (as list-aware approaches use additional information from the whole ranked list), such results have been demonstrated before [76], hinting that the choice of the loss function in LTR is highly dependant on the downstream task, as well as the dataset that is used. More, what is also interesting is that the pointwise XGBoost is outperformed by classical scoring functions on the LKAO-CV and LOLO-CV, even by the LinearSVR, which is the linear version of pointwise RankMHC. We hypothesize that this is due to the model overfitting to LRMSD values, instead of focusing on the ranking task. Pairwise and listwise approaches are able to partly circumvent this issue, by not emphasizing in accurately predicting LRMSD values, but the relative ranking of pMHC model pairs or lists.

We wanted to also inspect which features RankMHC deems as important for the task of pMHC binding mode identification. The feature importance (gain) for the twenty-five most important features can be seen in Figure 2B. To begin with, we observe that, for many features, attention is given to peptide residues that have been previously identified as anchor positions (positions 2 and 9), or residues that are near such anchor positions (position 1, 3 and 8). Features stemming from the middle portion of the peptide (such as RSA values in position 7) are present in the top features, but only slightly. This is surprising, as, for the majority of the pMHC models, anchor positions have mostly fixed geometries in the pMHC system, and are generally easier to predict (see Supplementary Figure 2). Investigating further, we trained a version of RankMHC where features stemming from positions 1, 2 and 9 are omitted. We choose to omit these positions in particular, as the mean RMSD to the native structure for these positions was less than 1.5 Å (see Supplementary Figure 2). Results show that omitting such features affects negatively the performance of the model (see Supplementary Table 5). We additionally trained a version of RankMHC where we included only the anchor positions. On the average case, we also saw diminished performance. Comparing and contrasting results using the different proposed metrics, we hypothesize that the anchor positions are needed as a first layer to filter out bad anchor placements, as well as to bias the set of geometries that should be expected, while the middle positions seem to further refine the whole ranking, which shows in the comparatively increased ρ (see Supplementary Table 5). As such, the baseline RankMHC which includes all positions has the best performance on average. Additionally, the per-position ref2015 score (denoted as *total_score* in Figure 2B) proves to be a very important feature for many anchor and non-anchor peptide residues. We suspect that the linear feature weighting as developed in ref2015 is already robust, particularly for the pMHC system, and provides a good basis for further fine-tuning the per-position weights. This is also proven by Rosetta's widespread use in the pMHC literature [15, 45, 61, 65, 66].

RankMHC performance on different pMHC tools

All training and evaluation for RankMHC has been specifically on pMHC models generated by APE-Gen2.0. However, training on APE-Gen2.0-based models does not guarantee good generalization on pMHC models derived from different pMHC structural modeling methods. As such, in order for RankMHC to be widely adopted, it is imperative that it is tested on different pMHC structural modeling tools. We sought to evaluate the performance of RankMHC on modeled structures generated by PANDORA, a homology modeling-based method [41]. We made sure to not include PANDORA generated pMHC models in our training datasets, to ensure that RankMHC has been exposed to only APE-Gen2.0-generated pMHC models. We were also careful to avoid data leakage, so that a specific instance of the XGBoost ensemble of RankMHC does not predict previously seen PDB codes (see Methods for more information in regards to the datasets used). For this benchmark, we focused on specifically comparing against classical protein-ligand scoring functions, including molPDF, the default scoring function of PANDORA [41].

LKPO-CV performance can be seen in Figure 3. The first row depicts the results for the five evaluation metrics as describes in the Methods. RankMHC exhibits slightly better ρ on average than other methods. In other metrics however, it is outperformed by the simpler ref2015 scoring function from the Rosetta modeling suite, and surprisingly, exhibiting the worst MRR out of all the scoring functions. We hypothesize that this is due to the nature of the PANDORA-generated pMHC models. More specifically, PANDORA uses MODELLER for both homology modeling and peptide loop refinement. This is fundamentally different to how APE-Gen2.0 generates the structures. Specifically, APE-Gen2.0-generated pMHC models include hydrogen placement in the peptide, while PANDORA-generated structures do not. Moreover, APE-Gen2.0, as a final step to the modeling process, uses openMM [101] for a relaxation step. This has proven to result in better MolProbity scores [102] when compared to PANDORA-generated pMHC models [39]. The fact that PANDORA exhibits worse MolProbity scores might be a result of steric clashes, Ramachandran outliers or unfavorable side-chain rotamers [102]. As RankMHC has seen only pMHC models that have been relaxed through the APE-Gen2.0 protocol, we assume introducing a pMHC model with no hydrogen content and no relaxation will result in an out-of-distribution feature set that reflects this.

To test this, we performed a post-processing step on the PANDORA-generated pMHC models. We performed hydrogen atom addition through PDBFixer [101], and perform an energy minimization step with protocols as previously described in [39]. Re-scoring the energy minimized structures with the same scoring functions results in RankMHC outperforming all other scoring functions in the benchmark (see bottom row of Figure 3). Interestingly enough, the performance of other scoring functions is not better, and is in fact slightly reduced when compared to the top-row experiments. This effect however has also been previously reported in previous protein-ligand docking benchmark studies [103].

Discussion

Identifying the binding mode of a ligand that is bound to a receptor is a very well-studied problem, and one of the main goals of molecular docking [29]. The

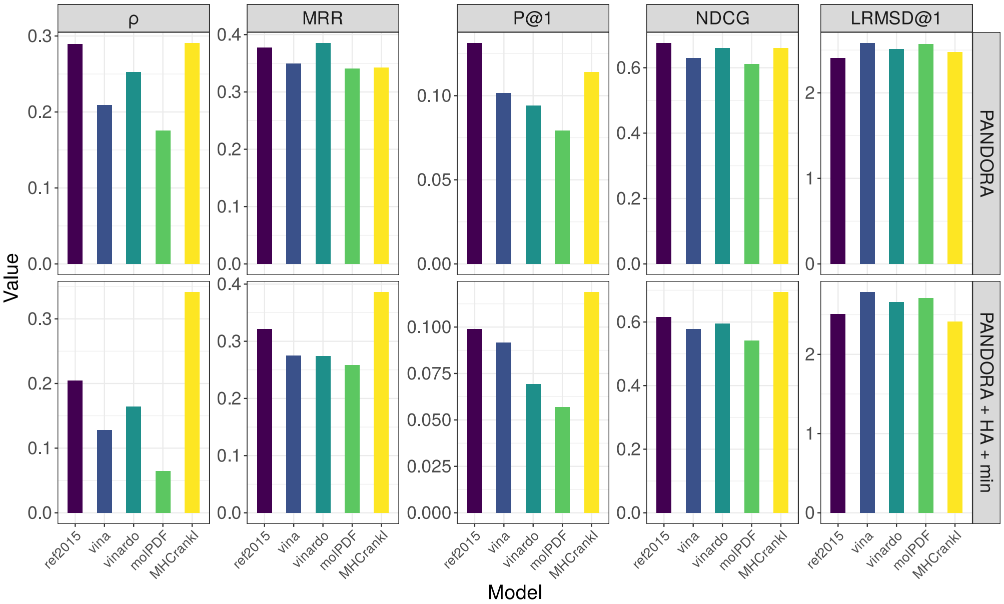

Fig 3. LKPO-CV RankMHC performance on PANDORA-generated pMHC models. The x-axis for each plot depicts different scoring functions, while the y-axis depicts the value for each metric (shown as the title of each subplot). The first row depicts the results of scoring functions on the vanilla PANDORA pMHC models. The second row depicts the results of PANDORA pMHC models with added hydrogens that have also underwent an energy minimization process. Higher ρ, MRR, P@1 and NDCG, as well as lower LRMSD@1, denote better performance.

literature is rich with general protein-ligand scoring functions, which can be applied to many biological systems. The reason for this is that these scoring functions comprise of well-founded theories and principles related to biophysics. However, such scoring functions do not inherently focus on characteristics that define a particular biological domain, and as such, cannot utilize, without modifications, specific domain knowledge that is crucial in identifying a protein ligand binding mode or predict the binding affinity of a ligand to a protein. ML-based scoring functions excel on exactly these two things: (i) specificity to the desired domain of application (or, in other words, specificity to particular biological systems) [56], and (ii) the ability to not be constrained to a classical scoring function's predetermined functional form, to which many biological systems do not necessarily conform to [47]. The last two points are quite important, especially for protein-ligand interactions which are typically unconventional, or exhibit specific domain knowledge that can be utilized downstream, such as the pMHC system. Specifically, the anchor placement of the peptide residues in the MHC cleft is typically known, and the possible search space of peptide conformations is biased towards certain geometries. As such, there is a clear motivation on utilizing ML to advance pMHC binding mode identification.

This evidence is not new to the pMHC literature, with many approaches utilizing ML for better performance in the task of pMHC binding mode identification [64–66]. However, such approaches focus on direct, pointwise predictions of measures that define the distance of a pMHC model to a native structure, or they do not take into account information about the contrastive aspect of the problem, that is, that a pMHC model is closer to a native structure than another. Turning into the LTR literature that is very rich with methodologies that relate to such problems, such as RankNet [75], LambdaRank [81] and LambdaMART [91], we created RankMHC, a new pMHC-specific binding mode identification function. RankMHC outperforms both classical peptide-ligand scoring functions, as well as other ML-based approaches in the literature (see Table 1). Results demonstrate that RankMHC can take advantage of both the domain specificity of the pMHC system, as well as taking advantage of the pairwise, contrastive learning approach to focus on the crux of the problem, which is the relative ranking of pMHC models, rather than directly predicting LRMSD values, which is a much harder problem. Using a pairwise learning approach, we instead focus on the relative differences between conformations, instead of absolute LRMSD values, circumventing this problem altogether. Such an approach helps in improving performance and with potential overfitting (see **Supplementary Table 4**).

Another improvement of RankMHC over other approaches in the literature is that it is generalizable to multiple alleles and peptide lengths, contrary to other approaches that mostly focus on the HLA-A*02:01 allele and nonamer peptides [65, 66]. In regards to peptide length, RankMHC achieves generalization by using a novel average pooling operation that converts any peptide to a nonamer with structural awareness (see **Supplementary Figure 1**). As a result, RankMHC can adequately generalize to unseen peptide lengths (see Table 1 and the LOLO-CV results), and showing slightly better performance than using simple padding (see Figure 2). This demonstrates that RankMHC can be as effective performance-wise with a reduced feature space that is smaller than a padded one, while being more efficient during training. We are curious to see whether such structure-aware padding can be helpful in other systems that perform srtucture-based pMHC binding affinity prediction [61], or even help purely sequence-based methods that either rely on padding [88] or completely pruning amino acids out of the peptide sequence [5]. We will also consider different ways of combining residues - average pooling being the simplest approach - and adopt more sophisticated pooling methodologies in the future, such as max pooling, or even pooling that is attention-based [104] and, in general, a pooling operation that can be learned in a task-specific manner.

It is imperative however that we acknowledge some limitations of RankMHC. In regards to RankMHC being used by modeling tools other than APE-Gen2.0, RankMHC demonstrates the best results when compared to other scoring functions on a dataset of PANDORA-generated pMHC models (see bottom half of Figure 3). However, this is only true when a specific post-processing step to the pMHC models is applied, namely, the addition of hydrogen atoms and the energy minimization of the whole pMHC complex. Indeed, when RankMHC is used on the vanilla PANDORA-generated pMHC models, performance is subpar (see top half of Figure 3). We hypothesize that this is due to the different characteristics of PANDORA-generated pMHC models. Specifically, PANDORA-generated models exhibit lower MolProbity scores than APE-Gen2.0-generated models [39], meaning that the PANDORA-generated models might contain steric clashes, Ramachandran outliers or unfavorable side-chain rotamers [102]. This, in turn, can radically change the feature content of such pMHC models so much that they could be considered as out-of-distribution data points for RankMHC, which explains the subpar performance. Explicitly providing pMHC models with added hydrogens that have also underwent an energy minimization process, solving potential steric clashes and other issues in the process, can be a catalyst for good RankMHC performance. As a future work, to avoid potential out-of-distribution issues, we plan to expand the training dataset of RankMHC, using pMHC models generated by different pMHC structural modeling that employ different methodologies. As such methodologies search the peptide conformational space in different ways, the geometries across different pMHC tools may vary, and such variety in the training data can be beneficial for RankMHC performance.

It also needs to be underlined that RankMHC is - apart from a domain-specific function - a task-specific function, namely, one that is specialized in pMHC binding mode identification. It is not designed for auxiliary but correlated tasks such as pMHC binding affinity prediction or virtual screening applications. The seminal work by Ashtawy and Mahapatra expands more on this topic, showing through benchmarks that task-specific, but even more general scoring functions cannot generalize equally well to different tasks [105]. In our experiments, we also observed similar behaviour with 3pHLA [61] and the random forest model by Abella et al. [62] not being able to properly rank peptide conformations. However, the work by Ashtawy and Mahapatra also proposes as a solution a multi-task approach, where there is one system that shares information for three different tasks - in the paper, the authors discuss molecular docking, virtual screening and binding mode identification - which are correlated with one another [105]. This notion of multi-task learning has been adopted by other ML-based scoring functions, such as GNINA, which is trained on both protein-ligand binding affinity labels, as well as labels related to the appropriate binding mode [50, 106]. Therefore, future work will also focus on combining knowledge from RankMHC and our previously developed structural pMHC tools for binding affinity prediction [61] and virtual screening [62]. We will consider building a pMHC multi-task system that shares structural information between different tasks in order to improve performance, as the tasks of pMHC binding affinity prediction, binding mode identification and virtual screening are, to an extend, correlated.

Supporting information

S1 Fig. Structure-aware average pooling. Example of how a 13-mer (PDB code: *2AK4*) is converted into a nonamer, using a canonical nonamer template (PDB code: *1DUZ*). The result of areas to be pooled is shown at the top.

S2 Fig. Per-position LRMSD distribution The x-axis depicts the nine different peptide residue positions. On the y-axis, the mean + standard deviation LRMSD for each position is shown.

S1 Table. Full list of features used in RankMHC.

S2 Table. Set of hyperparameters that all XGBoost-LambdaMART models were optimized on. The best performing method is depicted in bold, while the second best performing method is underlined.

S3 Table. Benchmark of different scoring functions on pMHC binding mode identification (redundant structures included). The best performing method is depicted in bold, while the second best performing method is underlined.

S4 Table. Benchmark of different LTR-based loss functions (vina/LinearSVR are included as reference). The best performing method is depicted in bold, while the second best performing method is underlined.

S5 Table. Contributions of peptide residues 1, 2 and 9 to the performance of RankMHC (vina is included as reference). The best performing method is depicted in bold, while the second best performing method is underlined.

Acknowledgments

R.F. and L.E.K. are supported in part by NIH U01CA258512 and by Rice University funds.

References

1. Peters B, Nielsen M, Sette A. T Cell Epitope Predictions. Annual Review of Immunology. 2020;38(1):123–145. doi:10.1146/annurev-immunol-082119-124838.
2. Schaap-Johansen AL, Vujović M, Borch A, Hadrup SR, Marcatili P. T Cell Epitope Prediction and Its Application to Immunotherapy. Frontiers in Immunology. 2021;12. doi:10.3389/fimmu.2021.712488.
3. Bassani-Sternberg M. In: Mass Spectrometry Based Immunopeptidomics for the Discovery of Cancer Neoantigens. Springer New York; 2018. p. 209–221. Available from: `http://dx.doi.org/10.1007/978-1-4939-7537-2_14`.
4. Pan K, Chiu Y, Huang E, Chen M, Wang J, Lai I, et al. Mass spectrometric identification of immunogenic SARS-CoV-2 epitopes and cognate TCRs. Proceedings of the National Academy of Sciences. 2021;118(46). doi:10.1073/pnas.2111815118.
5. 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 Research. 2020;48(W1):W449–W454. doi:10.1093/nar/gkaa379.
6. O'Donnell TJ, Rubinsteyn A, Laserson U. MHCflurry 2.0: Improved Pan-Allele Prediction of MHC Class I-Presented Peptides by Incorporating Antigen Processing. Cell Systems. 2020;11(1):42–48.e7. doi:10.1016/j.cels.2020.06.010.
7. Madden D. The antigenic identity of peptide-MHC complexes: A comparison of the conformations of five viral peptides presented by HLA-A2. Cell. 1993;75(4):693–708. doi:10.1016/0092-8674(93)90490-h.
8. Engelhard VH. Structure of peptides associated with MHC class I molecules. Current Opinion in Immunology. 1994;6(1):13–23. doi:10.1016/0952-7915(94)90028-0.
9. Madden DR. The Three-Dimensional Structure of Peptide-MHC Complexes. Annual Review of Immunology. 1995;13(1):587–622. doi:10.1146/annurev.iy.13.040195.003103.
10. Hellman LM, Foley KC, Singh NK, Alonso JA, Riley TP, Devlin JR, et al. Improving T Cell Receptor On-Target Specificity via Structure-Guided Design. Molecular Therapy. 2019;27(2):300–313. doi:10.1016/j.ymthe.2018.12.010.
11. Devlin JR, Alonso JA, Ayres CM, Keller GLJ, Bobisse S, Vander Kooi CW, et al. Structural dissimilarity from self drives neoepitope escape from immune tolerance. Nature Chemical Biology. 2020;16(11):1269–1276. doi:10.1038/s41589-020-0610-1.
12. Poole A, Karuppiah V, Hartt A, Haidar JN, Moureau S, Dobrzycki T, et al. Therapeutic high affinity T cell receptor targeting a KRASG12D cancer neoantigen. Nature Communications. 2022;13(1). doi:10.1038/s41467-022-32811-1.

13. Hopkins JR, MacLachlan BJ, Harper S, Sewell AK, Cole DK. Unconventional modes of peptide–HLA-I presentation change the rules of TCR engagement. Discovery Immunology. 2022;1(1). doi:10.1093/discim/kyac001.

14. Singh NK, Alonso JA, Devlin JR, Keller GLJ, Gray GI, Chiranjivi AK, et al. A class-mismatched TCR bypasses MHC restriction via an unorthodox but fully functional binding geometry. Nature Communications. 2022;13(1). doi:10.1038/s41467-022-34896-0.

15. Custodio JM, Ayres CM, Rosales TJ, Brambley CA, Arbuiso AG, Landau LM, et al. Structural and physical features that distinguish tumor-controlling from inactive cancer neoepitopes. Proceedings of the National Academy of Sciences. 2023;120(51). doi:10.1073/pnas.2312057120.

16. Berman HM. The Protein Data Bank. Nucleic Acids Research. 2000;28(1):235–242. doi:10.1093/nar/28.1.235.

17. Kaas Q. IMGT/3Dstructure-DB and IMGT/StructuralQuery, a database and a tool for immunoglobulin, T cell receptor and MHC structural data. Nucleic Acids Research. 2004;32(90001):208D – 210. doi:10.1093/nar/gkh042.

18. Ehrenmann F, Kaas Q, Lefranc MP. IMGT/3Dstructure-DB and IMGT/DomainGapAlign: a database and a tool for immunoglobulins or antibodies, T cell receptors, MHC, IgSF and MhcSF. Nucleic Acids Research. 2009;38:D301–D307. doi:10.1093/nar/gkp946.

19. Markosian C, Di Costanzo L, Sekharan M, Shao C, Burley SK, Zardecki C. Analysis of impact metrics for the Protein Data Bank. Scientific Data. 2018;5(1). doi:10.1038/sdata.2018.212.

20. Goodsell DS, Zardecki C, Di Costanzo L, Duarte JM, Hudson BP, Persikova I, et al. RCSB Protein Data Bank: Enabling biomedical research and drug discovery. Protein Science. 2019;29(1):52–65. doi:10.1002/pro.3730.

21. Ciemny M, Kurcinski M, Kamel K, Kolinski A, Alam N, Schueler-Furman O, et al. Protein–peptide docking: opportunities and challenges. Drug Discovery Today. 2018;23(8):1530–1537. doi:10.1016/j.drudis.2018.05.006.

22. Agrawal P, Singh H, Srivastava HK, Singh S, Kishore G, Raghava GPS. Benchmarking of different molecular docking methods for protein-peptide docking. BMC Bioinformatics. 2019;19(S13). doi:10.1186/s12859-018-2449-y.

23. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583–589. doi:10.1038/s41586-021-03819-2.

24. Skolnick J, Gao M, Zhou H, Singh S. AlphaFold 2: Why It Works and Its Implications for Understanding the Relationships of Protein Sequence, Structure, and Function. Journal of Chemical Information and Modeling. 2021;61(10):4827–4831. doi:10.1021/acs.jcim.1c01114.

25. Varadi M, Bertoni D, Magana P, Paramval U, Pidruchna I, Radhakrishnan M, et al. AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences. Nucleic Acids Research. 2023;doi:10.1093/nar/gkad1011.

26. Abramson J, Adler J, Dunger J, Evans R, Green T, Pritzel A, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature. 2024;doi:10.1038/s41586-024-07487-w.

27. Perez MAS, Cuendet MA, Röhrig UF, Michielin O, Zoete V. In: Structural Prediction of Peptide–MHC Binding Modes. Springer US; 2022. p. 245–282. Available from: http://dx.doi.org/10.1007/978-1-0716-1855-4_13.

28. Antunes DA, Abella JR, Devaurs D, Rigo MM, Kavraki LE. Structure-based Methods for Binding Mode and Binding Affinity Prediction for Peptide-MHC Complexes. Current Topics in Medicinal Chemistry. 2019;18(26):2239–2255. doi:10.2174/1568026619666181224101744.

29. Guedes IA, de Magalhães CS, Dardenne LE. Receptor–ligand molecular docking. Biophysical Reviews. 2013;6(1):75–87. doi:10.1007/s12551-013-0130-2.

30. Li J, Fu A, Zhang L. An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking. Interdisciplinary Sciences: Computational Life Sciences. 2019;11(2):320–328. doi:10.1007/s12539-019-00327-w.

31. Meng EC, Shoichet BK, Kuntz ID. Automated docking with grid-based energy evaluation. Journal of Computational Chemistry. 1992;13(4):505–524. doi:10.1002/jcc.540130412.

32. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML. Comparison of simple potential functions for simulating liquid water. The Journal of Chemical Physics. 1983;79(2):926–935. doi:10.1063/1.445869.

33. Raha K, Peters MB, Wang B, Yu N, Wollacott AM, Westerhoff LM, et al. The role of quantum mechanics in structure-based drug design. Drug Discovery Today. 2007;12(17–18):725–731. doi:10.1016/j.drudis.2007.07.006.

34. Murray CW, Auton TR, Eldridge MD. Journal of Computer-Aided Molecular Design. 1998;12(5):503–519. doi:10.1023/a:1008040323669.

35. Gohlke H, Hendlich M, Klebe G. Knowledge-based scoring function to predict protein-ligand interactions. Journal of Molecular Biology. 2000;295(2):337–356. doi:10.1006/jmbi.1999.3371.

36. Menegatti Rigo M, Amaral Antunes D, Vaz de Freitas M, Fabiano de Almeida Mendes M, Meira L, Sinigaglia M, et al. DockTope: a Web-based tool for automated pMHC-I modelling. Scientific Reports. 2015;5(1). doi:10.1038/srep18413.

37. Trott O, Olson AJ. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of Computational Chemistry. 2009;31(2):455–461. doi:10.1002/jcc.21334.

38. Abella J, Antunes D, Clementi C, Kavraki L. APE-Gen: A Fast Method for Generating Ensembles of Bound Peptide-MHC Conformations. Molecules. 2019;24(5):881. doi:10.3390/molecules24050881.

39. Fasoulis R, Rigo MM, Lizée G, Antunes DA, Kavraki LE. APE-Gen2.0: Expanding Rapid Class I Peptide–Major Histocompatibility Complex Modeling to Post-Translational Modifications and Noncanonical Peptide Geometries. Journal of Chemical Information and Modeling. 2024;doi:10.1021/acs.jcim.3c01667.

40. Quiroga R, Villarreal MA. Vinardo: A Scoring Function Based on Autodock Vina Improves Scoring, Docking, and Virtual Screening. PLOS ONE. 2016;11(5):e0155183. doi:10.1371/journal.pone.0155183.

41. Marzella DF, Parizi FM, Tilborg Dv, Renaud N, Sybrandi D, Buzatu R, et al. PANDORA: A Fast, Anchor-Restrained Modelling Protocol for Peptide: MHC Complexes. Frontiers in Immunology. 2022;13. doi:10.3389/fimmu.2022.878762.

42. Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen M, et al. Comparative Protein Structure Modeling Using Modeller. Current Protocols in Bioinformatics. 2006;15(1). doi:10.1002/0471250953.bi0506s15.

43. Alford RF, Leaver-Fay A, Jeliazkov JR, O'Meara MJ, DiMaio FP, Park H, et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. Journal of Chemical Theory and Computation. 2017;13(6):3031–3048. doi:10.1021/acs.jctc.7b00125.

44. Liu T, Pan X, Chao L, Tan W, Qu S, Yang L, et al. Subangstrom Accuracy in pHLA-I Modeling by Rosetta FlexPepDock Refinement Protocol. Journal of Chemical Information and Modeling. 2014;54(8):2233–2242. doi:10.1021/ci500393h.

45. Riley TP, Keller GLJ, Smith AR, Davancaze LM, Arbuiso AG, Devlin JR, et al. Structure Based Prediction of Neoantigen Immunogenicity. Frontiers in Immunology. 2019;10. doi:10.3389/fimmu.2019.02047.

46. Aranha MP, Spooner C, Demerdash O, Czejdo B, Smith JC, Mitchell JC. Prediction of peptide binding to MHC using machine learning with sequence and structure-based feature sets. Biochimica et Biophysica Acta (BBA) - General Subjects. 2020;1864(4):129535. doi:10.1016/j.bbagen.2020.129535.

47. Ain QU, Aleksandrova A, Roessler FD, Ballester PJ. Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. WIREs Computational Molecular Science. 2015;5(6):405–424. doi:10.1002/wcms.1225.

48. Stärk H, Ganea O, Pattanaik L, Barzilay R, Jaakkola T. Equibind: Geometric deep learning for drug binding structure prediction. In: International Conference on Machine Learning. PMLR; 2022. p. 20503–20521.

49. Corso G, Stärk H, Jing B, Barzilay R, Jaakkola T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. In: International Conference on Learning Representations (ICLR); 2023.

50. McNutt AT, Francoeur P, Aggarwal R, Masuda T, Meli R, Ragoza M, et al. GNINA 1.0: molecular docking with deep learning. Journal of Cheminformatics. 2021;13(1). doi:10.1186/s13321-021-00522-2.

51. Ashtawy HM, Mahapatra NR. In: Molecular Docking for Drug Discovery: Machine-Learning Approaches for Native Pose Prediction of Protein-Ligand Complexes. Springer International Publishing; 2014. p. 15–32. Available from: `http://dx.doi.org/10.1007/978-3-319-09042-9_2`.

52. Ashtawy HM, Mahapatra NR. Machine-learning scoring functions for identifying native poses of ligands docked to known and novel proteins. BMC Bioinformatics. 2015;16(S6). doi:10.1186/1471-2105-16-s6-s3.

53. Shim H, Kim H, Allen JE, Wulff H. Pose Classification Using Three-Dimensional Atomic Structure-Based Neural Networks Applied to Ion Channel–Ligand Docking. Journal of Chemical Information and Modeling. 2022;62(10):2301–2315. doi:10.1021/acs.jcim.1c01510.

54. Wang Z, Zheng L, Liu Y, Qu Y, Li YQ, Zhao M, et al. OnionNet-2: A Convolutional Neural Network Model for Predicting Protein-Ligand Binding Affinity Based on Residue-Atom Contacting Shells. Frontiers in Chemistry. 2021;9. doi:10.3389/fchem.2021.753002.

55. Wang Y, Wu S, Duan Y, Huang Y. A point cloud-based deep learning strategy for protein–ligand binding affinity prediction. Briefings in Bioinformatics. 2021;23(1). doi:10.1093/bib/bbab474.

56. Meli R, Morris GM, Biggin PC. Scoring Functions for Protein-Ligand Binding Affinity Prediction Using Structure-based Deep Learning: A Review. Frontiers in Bioinformatics. 2022;2. doi:10.3389/fbinf.2022.885983.

57. Wójcikowski M, Ballester PJ, Siedlecki P. Performance of machine-learning scoring functions in structure-based virtual screening. Scientific Reports. 2017;7(1). doi:10.1038/srep46710.

58. Zhang X, Shen C, Jiang D, Zhang J, Ye Q, Xu L, et al. TB-IECS: an accurate machine learning-based scoring function for virtual screening. Journal of Cheminformatics. 2023;15(1). doi:10.1186/s13321-023-00731-x.

59. McGibbon M, Money-Kyrle S, Blay V, Houston DR. SCORCH: Improving structure-based virtual screening with machine learning classifiers, data augmentation, and uncertainty estimation. Journal of Advanced Research. 2023;46:135–147. doi:10.1016/j.jare.2022.07.001.

60. Logean A, Sette A, Rognan D. Customized versus universal scoring functions. Bioorganic 'I&' Medicinal Chemistry Letters. 2001;11(5):675–679. doi:10.1016/s0960-894x(01)00021-x.

61. Conev A, Devaurs D, Rigo MM, Antunes DA, Kavraki LE. 3pHLA-score improves structure-based peptide-HLA binding affinity prediction. Scientific Reports. 2022;12(1). doi:10.1038/s41598-022-14526-x.

62. Abella JR, Antunes DA, Clementi C, Kavraki LE. Large-Scale Structure-Based Prediction of Stable Peptide Binding to Class I HLAs Using Random Forests. Frontiers in Immunology. 2020;11. doi:10.3389/fimmu.2020.01583.

63. Yachnin BJ, Mulligan VK, Khare SD, Bailey-Kellogg C. MHCEpitopeEnergy, a Flexible Rosetta-Based Biotherapeutic Deimmunization Platform. Journal of Chemical Information and Modeling. 2021;61(5):2368–2382. doi:10.1021/acs.jcim.1c00056.

64. Kyeong HH, Choi Y, Kim HS. GradDock: rapid simulation and tailored ranking functions for peptide-MHC Class I docking. Bioinformatics. 2017;34(3):469–476. doi:10.1093/bioinformatics/btx589.

65. Keller GLJ, Weiss LI, Baker BM. Physicochemical Heuristics for Identifying High Fidelity, Near-Native Structural Models of Peptide/MHC Complexes. Frontiers in Immunology. 2022;13. doi:10.3389/fimmu.2022.887759.

66. Gupta S, Nerli S, Kutti Kandy S, Mersky GL, Sgourakis NG. HLA3DB: comprehensive annotation of peptide/HLA complexes enables blind structure prediction of T cell epitopes. Nature Communications. 2023;14(1). doi:10.1038/s41467-023-42163-z.

67. North B, Lehmann A, Dunbrack RL. A New Clustering of Antibody CDR Loop Conformations. Journal of Molecular Biology. 2011;406(2):228–256. doi:10.1016/j.jmb.2010.10.030.

68. Burges C. From RankNet to LambdaRank toLambdaMART: An Overview; 2010.

69. Zhang W, Ji L, Chen Y, Tang K, Wang H, Zhu R, et al. When drug discovery meets web search: Learning to Rank for ligand-based virtual screening. Journal of Cheminformatics. 2015;7(1). doi:10.1186/s13321-015-0052-z.

70. Yuan Q, Gao J, Wu D, Zhang S, Mamitsuka H, Zhu S. DrugE-Rank: improving drug–target interaction prediction of new candidate drugs or targets by ensemble learning to rank. Bioinformatics. 2016;32(12):i18–i27. doi:10.1093/bioinformatics/btw244.

71. Tian H, Xiao S, Jiang X, Tao P. PASSerRank: Prediction of allosteric sites with learning to rank. Journal of Computational Chemistry. 2023;44(28):2223–2229. doi:10.1002/jcc.27193.

72. Eastman P, Swails J, Chodera JD, McGibbon RT, Zhao Y, Beauchamp KA, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. PLoS Comput Biol. 2017;13(7):e1005659. doi:10.1371/journal.pcbi.1005659.

73. Campello RJGB, Moulavi D, Sander J. In: Density-Based Clustering Based on Hierarchical Density Estimates. Springer Berlin Heidelberg; 2013. p. 160–172. Available from: `http://dx.doi.org/10.1007/978-3-642-37456-2_14`.

74. Herbrich R, Graepel T, Obermayer K. In: Large Margin Rank Boundaries for Ordinal Regression. The MIT Press; 2000. p. 115–132. Available from: `http://dx.doi.org/10.7551/mitpress/1113.003.0010`.

75. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, et al. Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on Machine learning - ICML '05. ICML '05. ACM Press; 2005.Available from: `http://dx.doi.org/10.1145/1102351.1102363`.

76. Qomariyah NN, Kazakov D, Fajar AN. Predicting User Preferences with XGBoost Learning to Rank Method. In: 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). IEEE; 2020.Available from: `http://dx.doi.org/10.1109/ISRITI51436.2020.9315494`.

77. Cao Z, Qin T, Liu TY, Tsai MF, Li H. Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th international conference on Machine learning. ICML '07 'I&' ILP '07. ACM; 2007.Available from: `http://dx.doi.org/10.1145/1273496.1273513`.

78. Xia F, Liu TY, Wang J, Zhang W, Li H. Listwise approach to learning to rank: theory and algorithm. In: Proceedings of the 25th international conference on Machine learning - ICML '08. ICML '08. ACM Press; 2008.Available from: `http://dx.doi.org/10.1145/1390156.1390306`.

79. Taylor M, Guiver J, Robertson S, Minka T. SoftRank: optimizing non-smooth rank metrics. In: Proceedings of the international conference on Web search and web data mining - WSDM '08. WSDM '08. ACM Press; 2008.Available from: http://dx.doi.org/10.1145/1341531.1341544.

80. LI H. A Short Introduction to Learning to Rank. IEICE Transactions on Information and Systems. 2011;E94-D(10):1854–1862. doi:10.1587/transinf.e94.d.1854.

81. Burges C, Ragno R, Le Q. Learning to Rank with Nonsmooth Cost Functions. In: Schölkopf B, Platt J, Hoffman T, editors. Advances in Neural Information Processing Systems. vol. 19. MIT Press; 2006.Available from: https://proceedings.neurips.cc/paper_files/paper/2006/file/af44c4c56f385c43f2529f9b1b018f6a-Paper.pdf.

82. Wang X, Li C, Golbandi N, Bendersky M, Najork M. The LambdaLoss Framework for Ranking Metric Optimization. In: Proceedings of The 27th ACM International Conference on Information and Knowledge Management (CIKM '18); 2018. p. 1313–1322.

83. Hubbard S, Thornton J. NACCESS; 1993. Computer Program, Department of Biochemistry and Molecular Biology, University College London.

84. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. ACM; 2016.Available from: http://dx.doi.org/10.1145/2939672.2939785.

85. Gfeller D, Guillaume P, Michaux J, Pak HS, Daniel RT, Racle J, et al. The Length Distribution and Multiple Specificity of Naturally Presented HLA-I Ligands. The Journal of Immunology. 2018;201(12):3705–3716. doi:10.4049/jimmunol.1800914.

86. Bolusani S, Besançon M, Bestuzheva K, Chmiela A, Dionísio J, Donkiewicz T, et al. The SCIP Optimization Suite 9.0. Optimization Online; 2024. Available from: https://optimization-online.org/2024/02/the-scip-optimization-suite-9-0/.

87. Perron L, Didier F. CP-SAT;. Available from: https://developers.google.com/optimization/cp/cp_solver/.

88. O'Donnell TJ, Rubinsteyn A, Bonsack M, Riemer AB, Laserson U, Hammerbacher J. MHCflurry: Open-Source Class I MHC Binding Affinity Prediction. Cell Systems. 2018;7(1):129–132.e4. doi:10.1016/j.cels.2018.05.014.

89. Venkatesh G, Grover A, Srinivasaraghavan G, Rao S. MHCAttnNet: predicting MHC-peptide bindings for MHC alleles classes I and II using an attention-based deep neural model. Bioinformatics. 2020;36(1):i399–i406. doi:10.1093/bioinformatics/btaa479.

90. Nielsen M, Lundegaard C, Blicher T, Lamberth K, Harndahl M, Justesen S, et al. NetMHCpan, a Method for Quantitative Predictions of Peptide Binding to Any HLA-A and -B Locus Protein of Known Sequence. PLoS ONE. 2007;2(8):e796. doi:10.1371/journal.pone.0000796.

91. Wu Q, Burges CJC, Svore KM, Gao J. Adapting boosting for information retrieval measures. Information Retrieval. 2009;13(3):254–270. doi:10.1007/s10791-009-9112-1.

92. Friedman JH. Greedy function approximation: A gradient boosting machine. The Annals of Statistics. 2001;29(5). doi:10.1214/aos/1013203451.

93. Errica F, Podda M, Bacciu D, Micheli A. A fair comparison of graph neural networks for graph classification. In: Proceedings of the 8th International Conference on Learning Representations (ICLR); 2020.

94. Myers JL, Well AD, Lorch Jr RF. Research Design and Statistical Analysis. Routledge; 2013. Available from: `http://dx.doi.org/10.4324/9780203726631`.

95. Silver NC, Dunlap WP. Averaging correlation coefficients: Should Fisher's z transformation be used? Journal of Applied Psychology. 1987;72(1):146–148. doi:10.1037/0021-9010.72.1.146.

96. Olkin I, Pratt JW. Unbiased Estimation of Certain Correlation Coefficients. The Annals of Mathematical Statistics. 1958;29(1):201–211. doi:10.1214/aoms/1177706717.

97. Myers L, Sirois MJ. Spearman Correlation Coefficients, Differences between; 2005. Available from: `http://dx.doi.org/10.1002/0471667196.ess5050.pub2`.

98. Fromer JC, Graff DE, Coley CW. Pareto optimization to accelerate multi-objective virtual screening. Digital Discovery. 2024;3(3):467–481. doi:10.1039/d3dd00227f.

99. Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems. 2002;20(4):422–446. doi:10.1145/582415.582418.

100. Gienapp L, Fröbe M, Hagen M, Potthast M. The Impact of Negative Relevance Judgments on NDCG. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. CIKM '20. ACM; 2020.Available from: `http://dx.doi.org/10.1145/3340531.3412123`.

101. Eastman P, Swails J, Chodera JD, McGibbon RT, Zhao Y, Beauchamp KA, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. PLoS Comput Biol. 2017;13(7):e1005659. doi:10.1371/journal.pcbi.1005659.

102. Davis IW, Leaver-Fay A, Chen VB, Block JN, Kapral GJ, Wang X, et al. MolProbity: all-atom contacts and structure validation for proteins and nucleic acids. Nucleic Acids Res. 2007;35(Web Server):W375–W383. doi:10.1093/nar/gkm216.

103. Buttenschoen M, Morris GM, Deane CM. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. Chemical Science. 2024;15(9):3130–3139. doi:10.1039/d3sc04185a.

104. Lee J, Lee I, Kang J. Self-Attention Graph Pooling. In: Proceedings of the 36th International Conference on Machine Learning; 2019.

105. Ashtawy HM, Mahapatra NR. Task-Specific Scoring Functions for Predicting Ligand Binding Poses and Affinity and for Screening Enrichment. Journal of Chemical Information and Modeling. 2017;58(1):119–133. doi:10.1021/acs.jcim.7b00309.

106. Francoeur PG, Masuda T, Sunseri J, Jia A, Iovanisci RB, Snyder I, et al. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. Journal of Chemical Information and Modeling. 2020;60(9):4200–4215. doi:10.1021/acs.jcim.0c00411.