PLOS ONE

RESEARCH ARTICLE

Assessing the generalization capabilities of TCR binding predictors via peptide distance analysis

Leonardo V. Castorina[1,2*], Filippo Grazioli[2], Pierre Machart[2], Anja Mösch[2], Federico Errica[2]

1 School of Informatics, University of Edinburgh, Edinburgh, United Kingdom, **2** NEC Laboratories Europe, Heidelberg, Germany

* leonardo.castorina@ed.ac.uk

OPEN ACCESS

Citation: Castorina LV, Grazioli F, Machart P, Mösch A, Errica F (2025) Assessing the generalization capabilities of TCR binding predictors via peptide distance analysis. PLoS One 20(5): e0324011. https://doi.org/10.1371/journal.pone.0324011

Editor: Amit Kumar, University of Cagliari, ITALY

Received: March 20, 2024

Accepted: April 19, 2025

Published: May 20, 2025

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0324011

Data availability statement: All relevant data for this study are publicly available from the

Abstract

Understanding the interaction between T Cell Receptors (TCRs) and peptide-bound Major Histocompatibility Complexes (pMHCs) is crucial for comprehending immune responses and developing targeted immunotherapies. While recent machine learning (ML) models show remarkable success in predicting TCR-pMHC binding within training data, these models often fail to generalize to peptides outside their training distributions, raising concerns about their applicability in therapeutic settings. Understanding and improving the generalization of these models is therefore critical to ensure real-world applications. To address this issue, we evaluate the effect of the distance between training and testing peptide distributions on ML model empirical risk assessments, using sequence-based and 3D structure-based distance metrics. In our analysis we use several state-of-the-art models for TCR-peptide binding prediction: Attentive Variational Information Bottleneck (AVIB), NetTCR-2.0 and -2.2, and ERGO II (pre-trained autoencoder) and ERGO II (LSTM). In this work, we introduce a novel approach for assessing the generalization capabilities of TCR binding predictors: the Distance Split (DS) algorithm. The DS algorithm controls the distance between training and testing peptides based on both sequence and structure, allowing for a more nuanced evaluation of model performance. We show that lower 3D shape similarity between training and test peptides is associated with a harder out-of-distribution task definition, which is more interesting when measuring the ability to generalize to unseen peptides. However, we observe the opposite effect when splitting using sequence-based similarity. These findings highlight the importance of using a distance-based splitting approach to benchmark models. This could then be used to estimate a confidence score on predictions on novel and unseen peptides, based on how different they are from the training ones. Additionally, our results may hint that employing 3D shape to complement sequence information could improve the accuracy of TCR-pMHC binding predictors.

GitHub repository (https://github.com/nec-research/peptide-distance-analysis).

Funding: Funding for this project was provided by NEC Labs Europe GmbH.

Competing interests: Funding for this project was provided by NEC Labs Europe GmbH.

Introduction

The immune system has evolved to recognize different pathogens, such as viruses, which enter cells and exploit their resources for replication. The immune response depends on how well the system can distinguish between healthy and infected/aberrant cells. It does so by leveraging specific molecules on the cell surface called Major Histocompatibility Complexes (MHC). Class I or class II MHCs are presented depending on the cell type [1–3]. An antigenic peptide is displayed by the MHC forming a peptide-bound MHC (pMHC). These peptides are derived from the proteasome, a complex of protease enzymes in cells which break down proteins into short peptides [4].

In infected cells, the presented peptides may be derived from viral proteins, whereas in healthy cells, peptides are derived from housekeeping proteins [5]. T cells can recognize viral peptides in the pMHC by binding it with the T Cell Receptors (TCRs), leading to an immune response [4]. TCRs are able to recognize many different peptides via diverse sequences at the variable regions of their α and β chains [6]. TCR binding to the pMHC primarily occurs at the Complementarity-Determining Regions (CDRs). The peptide recognition is primarily mediated by the CDR3. The CDR3-α is derived from the alleles of the V and J genes; the D gene, in addition to the V and J, is involved in shaping the sequence and structure of the CDR3-β, [7,8]. These alleles can be recombined extensively, ensuring a high TCR repertoire diversity and allowing for a broad T cell-based immune response [9]. The exposure of a naive T cell to an antigen leads to its activation and to the development of a memory T cell population with the same TCR. This allows for a long-lasting immune response [10,11]. A visual overview of the TCR-pMHC complex is presented in the S1 File.

Immunotherapies targeting cancer or viral infections use binding specificity to activate T cells, enhancing the immune system response. Binding specificity ensures that T cell binding occurs uniquely on the target, to kill say cancerous or infected cells, and avoid healthy cells. Research in immunotherapy covers two main categories: adoptive T cell therapy and T cell inducing vaccines. In adoptive T cell therapy, cancer patients receive specific T cells targeting and destroying the tumour [12,13]. T cell inducing vaccines leverage antigenic peptides or other classes of antigens to trigger specific T cell development against pathogens, such as viruses [14,15].

Immunotherapy design requires an in-depth understanding of the biochemical interactions between TCRs and pMHCs to accurately stimulate the immune system. The development of computational models that predict whether TCRs and pMHCs interact would allow for a faster *in silico* screening of sequences and consequent design of TCR sequences that bind specific target pMHCs. Recent advancements in machine learning (ML) lead to the development of TCR binding predictors [16–22]. The data input of these models consists of tuples of short amino acid sequences such as (peptide, CDR3-β) or (peptide, CDR3-β, CDR3-α, MHC). The task is usually formalized as a binary classification between binding or non-binding pairs.

Several computational studies on TCR binding prediction employ data from the Immune Epitope Database (IEDB) [23], VDJdb [24] and McPAS-TCR [25]. These databases mainly contain CDR3-β sequences, but often lack information on CDR3-α. Additionally, public TCR-pMHC interaction datasets are often limited in diversity and size. They also present sequence bias towards commonly studied viruses or binding/non-binding pairs bias from experiment setups. Furthermore, laboratory methods that validate the TCR-pMHC interactions, such as surface plasmon resonance, titration calorimetry and fluorescence anisotropy, are usually resource-intensive and time-consuming, hindering the creation of larger datasets [26].

Generally, machine learning (ML) TCR binding predictors achieve high test performance when evaluated on test sets originating from the same source as the training set. However, various studies [27–29] have shown that these methods exhibit weak cross-dataset generalization, meaning, models performance is significantly lower when training and test samples come from different distributions. Furthermore, [27] investigated the effect of different splitting techniques on the TChard dataset, which combines samples from IEDB, VDJdb and McPAS-TCR. They also investigated the effect of considering negative samples from wet lab assays and from random shuffling of the positive tuples. They observed that, when using a "vanilla" Random Split (RS) with negative samples from wet lab assays, ML-based models achieve an estimated area under the receiver operator characteristic (AUROC) larger than 95% on the test set. In this setting, the sets of CDR3 sequences in positive and negative samples are disjoint. This allows the models to memorize whether a CDR3 sequence was observed in either the negative or positive samples at training time. [27] then showed that a simple countermeasure to this problem consists in creating negative samples by randomly shuffling the positive tuples. In this setting, when the RS is employed, models achieve an AUROC score between 70% and 80%. Nevertheless, using the RS, peptides and CDR3 sequences may appear in both the training and test sets, leading to inflated estimates of the real-world model generalization capabilities.

Hence, to approximately test model performance for unseen peptides, they propose an alternative splitting method, named Hard Split (HS). The HS ensures that test peptides are never observed at training time. Therefore, the peptide sequences in the training, validation and test sets are unique to their respective sets. When using the HS and negative samples are obtained via random shuffling, the model performs barely better than random guesses, with an AUROC smaller than 55%.

To ensure real-world applicability, the HS, as shown by [27], aims at evaluating models' generalization abilities under the toughest setup possible. In fact, new peptides arising from new pathogens may differ significantly from those used at training time. ML models can only be safely employed for broad real-world applications if they show *sufficient* generalization to unseen sequences.

In practice, however, RS and HS represent two extremes of a wider spectrum of data splitting options. Assessing other splitting options can provide further insights on the robustness of ML models.

Our work is driven by the following research question: can we estimate the performance of ML models given an unseen test peptide and its distance from the training ones? We hypothesize that, as the difference in distributions between training and test sets increases, the prediction task becomes gradually harder. Knowing how well ML predictors generalize to an increasing shift in distributions between training and test sets would provide a tangible metric to measure when it is appropriate and to which extent we can employ these methods in the real world.

For this reason, in this work we introduce a new dataset splitting algorithm called *Distance Split (DS)*. Analogously to the HS, peptides placed in the training set are absent from the test set and vice versa. However, given a distance metric, we control the distances between sequences in the training and in the test sets. We increase such distance to test performance on peptides that are "further apart" from the ones seen by the model at training time, making the task increasingly harder.

Previous efforts have predominantly relied on sequence-based metrics, such as Levenshtein distance and BLOSUM substitution matrices, to measure similarity between peptides [30–32]. However, sequence metrics may not fully capture the structural aspects of TCR-peptide interactions, which are essential for accurate binding predictions. The integration of 3D structural information, such as Root Mean Square Deviation (RMSD) between peptides, could offer a more comprehensive approach to evaluating model performance on unseen peptides. Therefore, we explore different distance metrics, i.e., sequence metrics such as Levenshtein and BLOSUM [33], as well as shape metrics such as RMSD [34], using the predicted structures of peptides. As opposed to binning the HS based on distance, the DS allows for control over the median distance of the peptides in the test and validation.

We show that the increased peptide distance between the training and test split directly correlates with the models performances. In particular, increased RMSD (shape) distance between training and test peptides leads to decreased performance. Surprisingly, for BLOSUM (sequence) distance we see the inverse relationship. In real-world scenarios, when predictions on new unseen viral epitopes are required, we believe that leveraging RMSD between the 3D structures of the training and test peptides could serve as a valuable indicator of model reliability, with higher RMSD implying increased prediction uncertainty. Conversely, the inverse relationship observed with BLOSUM distance suggests that incorporating sequence-diverse training data may actually improve model generalization to novel peptides. Thus, a combination of structural and sequence-based metrics could provide a balanced approach, enhancing both the robustness and reliability of TCR-pMHC binding predictions.

Materials and methods

Dataset creation

With the goal of vaccine development against viruses, we select a viral subset of the VDJdb dataset, focusing on human host and MHC class I [35]. We omit MHC class II due to the small number of available samples. The dataset includes peptides from SARS-CoV-2, Influenza A, Human Immunodeficiency Virus (HIV), Hepatitis C Virus (HCV) and Cytomegalovirus (CMV). We discard data points which do not include both the CDR3-β and the peptide. The dataset contains 52 unique MHC A and 1 MHC B alleles, 16,504 unique CDR3-α sequences, 28,831 unique CDR3-β sequences and 757 unique peptides, for a total of 34,415 binding samples. To generate non-binding (i.e., negative) samples, we randomly shuffle the available (peptide, CDR3-β) pairs. This process is commonly employed in the literature [19,36] and leverages the hypothesis that random pairs will most likely not bind. To create a balanced dataset, we randomly generate 36,641 samples of non-binding combinations of CDR3-β and peptide sequences, to increase the total number of data points to 65,946. In this study, the data consists of pairs of (peptide, CDR3-β) and a binding/non-binding label.

Obtaining 3D structures of peptides. The VDJdb dataset contains information on the primary structure of proteins, i.e., the sequence of amino acids. In reality, these sequences exist as 3D shapes and as part of a complex [35]. However, public datasets like VDJDB lack the 3D shapes of peptides. We use ESMFold, a light-weight sequence-to-shape Language Model [37] and its online API[1] to generate 3D structures for the peptide sequences. For structures that gave errors, we use OmegaFold [38], a similar sequence-to-shape Language Model. We use the shapes to calculate the 3D distance between peptides, as described in the following section.

The distance split algorithm

Given a distance metric, the goal of the Distance Split (DS) algorithm is to create data splits of the available (peptide, CDR3-β) samples, enforcing a specified median peptide distance between training and test (and validation).

[1] https://esmatlas.com/resources?action=fold.

As distance metrics, we use Levenshtein, BLOSUM and RMSD. The Levenshtein distance is a string edit distance measuring the minimum number of edits required to change one peptide sequence into another [39]. The BLOSUM distance is a substitution matrix-based metric that quantifies the evolutionary similarity between two peptide sequences by comparing their amino acid global alignments [40]. The RMSD distance is a 3D structural metric that measures the average deviation between the atomic positions of two superimposed peptide structures, providing an estimate of their conformational similarity. We calculated the RMSD shape distance using PyMol [34]. PyMol aligns any pair of peptide 3D structures computes the RMSD between Cα of the two structures. The Cα atoms are the backbone carbon atoms in each amino acid, which are commonly used in RMSD calculations to provide a simplified yet accurate representation of the overall peptide structure. We focus on Cα atoms to avoid over-penalizing minor deviations in side chains, as protein folding models can generate physically impossible conformations that would otherwise lead to inflated RMSD values [41].

For each metric (Levenshtein, BLOSUM, RMSD), we calculate the pairwise distances between all 757 peptides, resulting in three distance matrices: $\mathbf{M}_{Levenshtein}$, $\mathbf{M}_{BLOSUM}$ $\mathbf{M}_{RMSD}$ $\in \mathbb{R}^{757\times757}$. For each distance matrix, we then calculate the row-wise median $\mathbf{m}_{med} \in \mathbb{R}^{757}$. Each value of $\mathbf{m}_{med}$ corresponds to the median distance between a given peptide and all other peptides. We then select three bin ranges over the cumulative distribution of the median distances of $\mathbf{m}_{med}$, defined by a lower and an upper bound (b_l, b_u) distance, i.e., (0,33), (33,66) and (66,100). Given the median vector $\mathbf{m}_{med}$, for each interval, we compute the distances corresponding to the lower and upper percentiles, i.e., d_l and d_u, for b_l and b_u, respectively. We then filter out all the peptides whose median distance falls outside the interval of interest. The sets of peptides that fall inside this percentile interval are then sampled for test and validation. For each training-validation-test split, we use a 90-5-5 ratio. The DS algorithm iteratively samples the validation and test sets based on this ratio and the total available data. To guide this process, we calculate the expected number of data points for each split N_{train}, N_{test}, $N_{validation}$, which we refer to as the test and validation budget.

We select the test samples by iteratively sampling peptides from the specified percentile interval. In each iteration, we randomly choose a peptide from this set, and move all corresponding (peptide, CDR3-β) pairs to the test set. This process continues until the target number of test samples (test budget) is reached. Peptides that were not selected are used to form the validation set, following the same iterative procedure. Then, the remaining peptides are assigned to the training set.

Additionally, we enforce a minimum and maximum count for peptides to be included in the validation and test sets. This is meant to constrain the peptide diversity in the set. In this work, we use a minimum count of 5 maximum count of 5,000. The complete algorithm for the DS is available in 1.

Using the DS, as well as the RS and HS, we generate training, test and validation splits of the available data points for all distance metrics, $\mathbf{M}_{Levenshtein}$, $\mathbf{M}_{BLOSUM}$ and $\mathbf{M}_{RMSD}$. As shown in S4 Fig, the average number of unique peptides is roughly the same across splits.

TCR-peptide interaction prediction models

We select 5 state-of-the-art models for TCR-peptide interaction prediction: Attentive Variational Information Bottleneck (AVIB) [27], NetTCR-2.0 [36], NetTCR-2.2 [42], ERGO II (with pre-trained TCR autoencoder), and ERGO II-LSTM [16] (see details in S1 File, Sect 2).

Algorithm 1. Distance Split (DS) algorithm based on peptide distances within specified percentile bounds.

```
Input: Dataset of peptide-TCR pairs $\mathcal{D} = \{(pep_i, TCR_i)\}_{i=1}^{N}$;
Peptide distance metric (e.g., BLOSUM or RMSD);
Training split ratio s (fraction of data for training);
Lower and upper percentile bounds $(b_l, b_u)$ over the distance distribution;
Leeway factor l: flexibility when the exact sample budget cannot be met. If
the number of available samples falls below the target by up to $l \times N$, the
algorithm will accept the smaller set.
Output: Training, validation, and test sets.
1) Compute unique peptides and counts:
• Extract the set of unique peptides $\mathcal{P}$ from $\mathcal{D}$.
• $\forall$ $pep \in \mathcal{P}$, count the number of (pep,TCR) pairs and store in CountMap(pep).
2) Calculate peptide distances:
• Compute the distance matrix $\mathbf{M}_{\texttt{dist}}$ between peptides using the chosen method.
• Calculate the median distance $\mathbf{m}_{\texttt{med}}$ for each peptide from $\mathbf{M}_{\texttt{dist}}$.
3) Select peptides within percentile bounds:
• Determine $d_l$ and $d_u$ distances corresponding to percentiles $b_l$ and $b_u$ from $\mathbf{m}_{\texttt{med}}$.
• Select peptides $\mathcal{P}^* = \{pep \in \mathcal{P} \mid \mathbf{m}_{\texttt{med}}(pep) \in [d_l, d_u]\}$.
4) Determine split sizes:
• Calculate total sample count: $N_{\texttt{total}} = \sum_{pep \in \mathcal{P}} CountMap(pep)$.
• Compute training sample count: $N_{\texttt{train}} = s \times N_{\texttt{total}}$.
• Remaining samples: $N_{\texttt{remaining}} = N_{\texttt{total}} - N_{\texttt{train}}$.
• Set test and validation counts: $N_{\texttt{test}} = N_{\texttt{validation}} = N_{\texttt{remaining}}/2$.
5) Initialize sets:
• $\mathcal{P}_{\texttt{train}} \leftarrow \mathcal{P}$, $\mathcal{P}_{\texttt{test}} \leftarrow \{\}$, $\mathcal{P}_{\texttt{val}} \leftarrow \{\}$.
• Counters: $t_{\texttt{test}} \leftarrow 0$, $t_{\texttt{val}} \leftarrow 0$.
6) Sample peptides for test and validation sets:
for each set in {test, validation} do
    while counter $t < N \times (1\text{–}l)$ do
        Randomly sample pep from $\mathcal{P}^*$ not already selected.
        if $t + CountMap(pep) \leq (1 + l) \times N$ then
            Add pep to the current set.
            Remove pep from $\mathcal{P}_{\texttt{train}}$ and $\mathcal{P}^*$.
            Update counter: $t \leftarrow t + CountMap(pep)$.
        end
    end
end
7) Construct final training, validation and test datasets:
• $\mathcal{D}_{train} = \{(pep, TCR)_i\}$, where $pep \in \mathcal{P}_{\texttt{train}}$; $\mathcal{D}_{val} = \{(pep, TCR)_i\}$, where $pep \in \mathcal{P}_{\texttt{val}}$;
  $\mathcal{D}_{test} = \{(pep, TCR)_i\}$, where $pep \in \mathcal{P}_{\texttt{test}}$.
```

With the exception of ERGO, all models encode the TCRβ and peptide sequences using the BLOSUM substitution matrix [40].

In AVIB, the TCRβ and Peptide encoders estimate Gaussian posteriors over a latent space. The various posteriors are then combined using an attention mechanism to estimate a single multi-sequence posterior distribution, which is then used to estimate the binding probability [27]. In NetTCR-2.0, the TCRβ and peptide encoders are encoded using several convolutional and global max pooling layers. The encoding is then concatenated and passed through two fully connected layers [36]. NetTCR-2.2 introduces a dropout layer before the fully connected layers, ReLU over Sigmoid, and 64 units rather than 32 for the fully connected layer [42]. In the ERGO II model, the TCRβ is encoded using a pre-trained autoencoder. ERGO II-LSTM encodes the peptide using an LSTM encoder. The encodings are then passed to two fully connected layers [16].

Results

We compute the pairwise distance between all peptides using Levenshtein, BLOSUM and RMSD distance for the viral VDJdb dataset. Descriptive statistics of the metrics are summarized in Table 1 and the median visualized in Fig 1. The peptides count distribution is heavily skewed towards a count of less than 10 instances, with several outliers, as shown by the high standard deviation. The median RMSD has a bimodal distribution with peaks at ≈ 0.0 and ≈ 0.5 Å. The median BLOSUM is normally distributed with most values ranging from 50 to 100. Finally, the median Levenshtein distance is ≈ 8, and shows very little distribution of values.

Then, we calculate the correlation between these distances in Fig 2. Levenshtein and BLOSUM distances are very significantly correlated (Spearman ρ: 0.32, p-value: 0), as expected since BLOSUM distance uses a sequence alignment. However, additional physico-chemical information is implicitly accounted for in BLOSUM, which considers evolutionary substitution patterns of amino acids [40]. This inclusion captures the likelihood of biologically relevant substitutions based on factors like size, charge, and hydrophobicity, which are not reflected in simple sequence-based metrics like Levenshtein distance. We observe no correlation between RMSD and sequence-based metrics.

For our experiments, we use five state-of-the-art deep learning models for TCR-peptide interaction prediction: NetTCR-2.0 [36], NetTCR-2.2 [42], AVIB [19], ERGO II and ERGO II-LSTM [16]. We evaluate how increasing the distance between the training and test set affects their performance. We train and test all models with RS, HS as baselines and and DS. For the

Table 1. Descriptive statistics of peptide counts, as well as Levenshtein, BLOSUM, and RMSD distances for the viral VDJdb dataset. Levenshtein and BLOSUM are sequence distance metrics. RMSD refers to the average 3D distance between peptides.

Metric	Average	Median	Stdev
Peptide (counts)	45	2	576
Levenshtein	8.29	8.25	0.60
BLOSUM	85.04	71.10	28.97
RMSD (Å)	0.46	0.36	0.27

https://doi.org/10.1371/journal.pone.0324011.t001

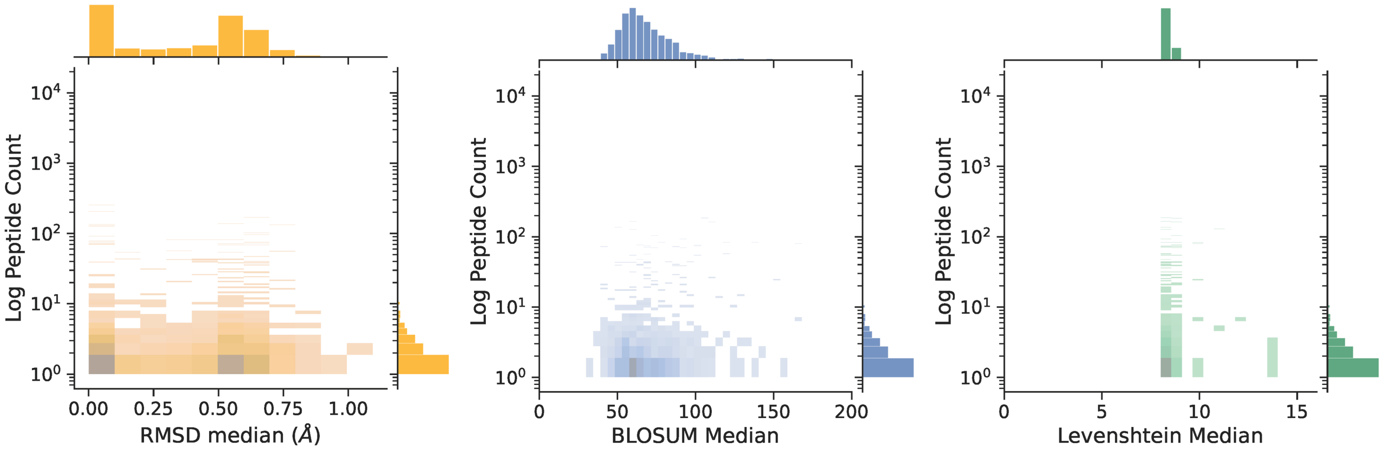

Fig 1. Distribution of peptide count against median distance of the peptide against all other peptides. Distances are RMSD (3D shape), BLOSUM (sequence), and Levenshtein (sequence), respectively.

https://doi.org/10.1371/journal.pone.0324011.g001

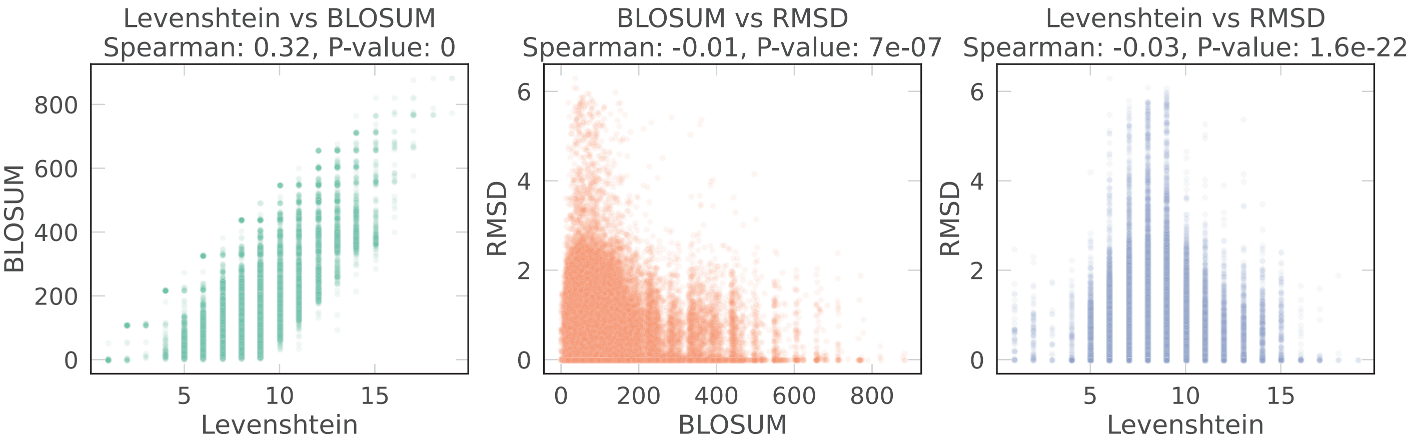

Fig 2. Correlations between peptides distance metrics in the Viral VDJDB dataset. Levenshtein and BLOSUM are sequence-based distances, while RMSD is a shape-based distance. There is a positive correlation between sequence distances and no correlation between shape and sequence distances.

https://doi.org/10.1371/journal.pone.0324011.g002

DS, we use lower and upper bound pairs over the cumulative distribution of the median distances: (0, 33), (33, 66) and (66, 100). We repeat each split with 5 different random seeds and the average results and 95% confidence intervals are shown in Fig 3. More detailed results with Levenshtein DS, overall and per-peptide metrics are available in S1 File Sect 2.

As shown in Fig 3, AVIB and NetTCR-2.2 perform similarly in the sequence and shape DS splits, followed by NetTCR-2.0 and both ERGO II models. As expected, all models perform best in the RS, as peptides in the training set also appear in the test set. On the other hand, performance on the HS is significantly lower compared to the RS.

Regarding the DS splits, as the median distance between the training and test sets increases, model performance worsens in RMSD-based DS splits, as shown by significant Spearman correlations ($p < 0.05$) for NetTCR-2.2, NetTCR-2.0, and AVIB, all within a 95% confidence interval. In contrast, we observe the opposite trend for BLOSUM-based DS splits, where models perform better as the median BLOSUM distance between the training and test sets increases, with all p-values below 0.05. In general, the highest performance in the DS splits is slightly higher in the RMSD (0,33) bin, than in the BLOSUM (66,100) bin, except for ERGO II (LSTM). There is generally no significant effect on the performance trend when using Levenshtein-based distance split, except for ERGO II (LSTM) which shows a similar trend to BLOSUM (see S14 Fig).

Discussion

The TCR-peptide/pMHC interaction prediction problem presents evaluation challenges [27]. When data is randomly split, models can achieve over-optimistic test performance, as test sequences can be observed at training time.

As a solution, [27] proposed the Hard Split (HS), which consists in randomly sampling test peptides and moving all instances of that peptide to the test set. The HS ensures that test peptides are never seen at training time, thus simulating what happens in the real world, for example, when new peptides arise from new pathogens.

This raises the question of whether it is possible to infer how well a model will generalize to an unseen peptide, by knowing how different the novel peptide is from those seen at training time.

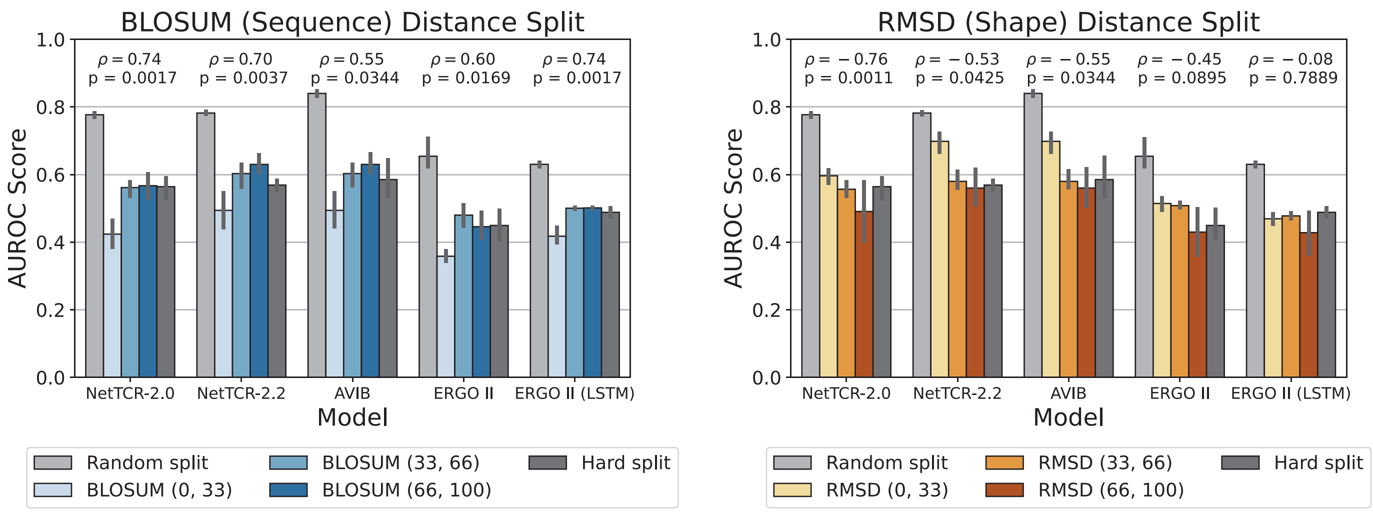

Fig 3. AUROC Scores for TCR's CDR3β-peptide binding prediction for models trained and tested using various splitting techniques. The commonly used *Random Split* allows test peptides to be observed also during training. In the *Hard Split*, peptides are exclusively allocated to either the training or test set. In the *Distance Split*, we enforce specific median distances between the training and test peptides, using either sequence distance (BLOSUM) or shape distance (RMSD). The training-test distance between peptides is controlled by selecting three percentile intervals over the cumulative median peptide-peptide distance distribution: (0,33), (33,66) and (66,100). Mean and 95% confidence intervals are calculated over 5 repeated experiments with different random seeds. More detailed results are available in S1 File Sect 2, with Levenshtein DS (S15 Fig), overall and per-peptide metrics (S1 File Sects 2.2.2 and 2.2.3).

https://doi.org/10.1371/journal.pone.0324011.g003

We divide the peptides into different training and test sets using a distance-based algorithm using a sequence metric (BLOSUM) and a shape metric (RMSD). To obtain the RMSD, we generate the 3D structures of the antigenic peptides using ESMFold and OmegaFold [37, 38]. We then use PyMol to align and calculate the Root Mean Squared Distance (RMSD) between the Cα atoms in the protein backbone [34]. We use Cα RMSD, as opposed to all-atoms RMSD, as it is computationally less intensive and produces similar results [43]. Additionally, predicted 3D structures can (and often will) have artifacts such as physically-unlikely positioning of the amino acid side chains, which could unfairly increase the all-atom RMSD [41].

To simulate real-world scenarios involving unseen peptides in a controlled manner, we propose the *Distance Split* (DS) approach (see Algorithm 1). Using the DS, we chose each test split to guarantee a given median distance between peptides in the training and test peptides. By using the median, some peptides with properties similar to those in the training set may still be included in the test set, thereby allowing for a degree of overlap while maintaining sufficient variability. Nevertheless, when employing the DS, analogously to the HS, test and validation peptides cannot appear at training time. For more stringent conditions, the minimum distance could be used with the DS to ensure that all test peptides are distinctly different from those in the training set.

Specifically, the test peptides are chosen by sampling from a subset of peptides, whose median distance from the other peptides is between a lower and an upper bound. In our experiments, we consider the following intervals over the cumulative median distance distribution: (0,33), (33,66) and (66,100).

As shown in Fig 2, we observe no correlation between RMSD and BLOSUM. Several factors could explain this. The 3D structure, as opposed to the amino acid sequence, is more relevant to describe protein-protein interactions. For example, certain angles in structures could make specific areas interact more than others. Related to this, some small differences in sequence metrics may result in larger structural differences, for instance, if two oppositely charged amino acids are placed next to each other. The 3D structure and RMSD metric can therefore provide a stronger signal for binding prediction models and may be used for testing models performance in out-of-distribution settings, or for the development of new models. Interestingly, we observe that models' performance is slightly lower for DS in the ranges (33,66) and (66,100) compared to that of the HS. The HS ensures that both test and training peptides are *unique*. However, even with this uniqueness, test and training peptides might still possess a certain degree of similarity. This similarity provides the model with additional chemical information about the peptide. In contrast, DS enforces a threshold distance between the training and test sets, thereby controlling the difficulty.

Obtaining a 3D structure of a protein often requires lengthy experiments to crystallize it, which may take days, months, or years [44]. With the advent of sequence-to-shape models like AlphaFold [45], OmegaFold [38] and ESMFold [37], this lengthy process can often be reduced to minutes or hours and makes it possible to obtain a fairly accurate 3D shapes of proteins. Structural validation in the lab of these models is currently underway, with some researchers reporting partial success, even when solving for previously unknown structures [46]. Successful applications of sequence-to-shape models in protein research include vaccine design [47], binding affinity ranking [48], protein sequence design [49,50] and benchmarking [51].

The structures produced by sequence-to-shape models like the ones mentioned above, may be biased towards the training sequences in the Protein Data Bank (PDB), which is composed of larger structures compared to peptides, which are made up of just a few amino acids in length [52]. Nevertheless, in this study, we use the shape as a representation of the real shape and assume that differences between two shapes would be consistent. This analysis could be extended to the full VDJdb dataset to test how the findings transfer to non-viral peptides.

In the structure-based DS, as the median RMSD distance between training and test peptide is increased, the model performance generally worsens. On the other hand, we observe the opposite effect in sequence-based DS, where increasing the BLOSUM sequence between training-test leads to better generalization performance (See Fig 3).

The immune system has evolved to recognize multitudes of pathogens, some of which may have never existed. The TCR-pMHC interaction is therefore degenerate, meaning that many TCRs can recognize the same peptide and that a TCR can recognize millions of peptides, even if the sequences are not identical [53]. Additionally, TCRs may bind different peptides using different binding modes [54]. However, as shown in Fig 2, we find no correlation between sequence and shape distance metrics *for the peptide*. Coupled with the results of Fig 3, this may imply that TCRs bind conformationally similar viral peptides even if the sequences are different. This is supported by the fact that the peptide binding pocket of MHCs have anchor positions, like the B pocket, which can accommodate different secondary anchor amino acids without altering the overall peptide conformation [55,56]. This flexibility allows TCRs to recognize structurally similar peptides despite sequence variations, suggesting that structural rather than purely sequence-based features drive TCR recognition efficiency in some cases [55,56]. Additionally, viruses, like SARS-CoV-2, evolve by changing their sequences to escape immune detection while keeping the structural features needed to interact with host cells. For example, mutations in the spike protein allow SARS-CoV-2 to evade the immune system but still bind effectively to the ACE2 receptor, ensuring the virus remains functional [57,58]. This suggests that TCRs may bind structurally similar peptides even when their sequences differ, which could explain why models trained on sequence-diverse data (high BLOSUM distance) show improved generalization. In the BLOSUM (0,33) split, where training peptides share higher sequence similarity, models may overfit to sequence-based patterns, potentially memorizing sequence motifs rather than learning underlying structural determinants of binding, leading to lower test performance. This may explain why models trained on more sequence-diverse peptides (33,66) and (66,100) generalize better, as they are forced to rely on broader structural and contextual features.

Additionally, in the field of protein design, benchmarking has revealed that even with a sequence similarity below 50%, the predicted 3D structures look very similar [51]. These results suggest that BLOSUM-based sequence augmentation, aimed to generate similar binding peptides for a given TCR, may reduce generalization. In contrast, a shape-based augmentation, for example by using protein sequence design models to generate new peptides, may enhance generalization to unseen peptides [49,59]. Additionally, given that the peptide sequence and shape are not correlated, including the 3D shape may increase the overall performance, as shown by recent papers [60–63] However, as TCR and MHC structures are generally similar, it may be worthwhile to limit the 3D information to the regions that vary in shape, like the peptide, the CDRs, and the binding pocket to reduce the model complexity and maintain performance [62]. Future work can explore the differences in structural distance between models such as AlphaFold, ESMFold and OmegaFold in the context of peptides, to further evaluate biases and divergences of structures generated by different sequence-to-shape models.

A short-coming of the BLOSUM distance for the peptide is that changes in the anchor amino acids should be weighted more than changes outside the anchors. Future work could explore a more peptide-centric metric of distance based on anchor positions. Similarly, a short-coming of the RMSD is that due to differences in peptide lengths, it is not possible to normalize the distance by the length of the peptide. In our pairwise comparison, however, most (about 87%)[2] of the peptide distances were same-length comparison. Furthermore, the RMSD is computed on predicted structures, which may contain modeling artifacts. These artifacts could introduce noise in structure-based splits, potentially leading to misleading structure-based generalization patterns. Future work could explore the RMSD of the core amino acids of the peptide and using length-normalized RMSD like RMSD_{100} [64].

Additionally, datasets with highly clustered peptides may present disproportionately low median distances, leading to unintended effects where the test set consists largely of local clusters rather than globally diverse peptides. Conversely, outliers with high median distances will always be placed in test or validation, making these sets more challenging than intended. We mitigate this by enforcing minimum and maximum peptide counts. Future work could explore clustering-based analysis of the peptide datasets or clustering to produce balanced peptide representation in training-test splits.

Despite the limitations of both sequence- and structure-based metrics, the DS algorithm is designed to be metric-agnostic, meaning it can be applied to any distance function that quantifies peptide similarity (see S1 File Sect 2.2.4 for results obtained using the Min operation instead of Mean in the DS algorithm). However, the choice of distance metric significantly impacts the nature of the training-test split and, consequently, model generalization. Future work could explore more biologically or immunologically relevant distances, such as a BLOSUM matrix derived from TCR-pMHC interaction data or alternative physico-chemical distance metrics [65].

Sequence-to-shape models could also be used to predict full TCR complexes for binding prediction. However, current methods for structural complex modeling are computationally expensive. While they may be comparably faster than molecular dynamics simulations,

[2] 248,630 matching length pairs comparisons out of 285,090 using 757 peptides

it is currently unfeasible to test all possible combinations of TCRs, peptides and pMHCs for the binding prediction task. Future work in lightweight representations for proteins, such as fragments, could help bridge this gap, allowing for faster and more efficient incorporation of structural features in TCR binding prediction [66].

Conclusion

Our results suggest that the use of 3D shapes in the context of TCR-pMHC interaction prediction could help reduce the uncertainty about the generalization capabilities of ML models to unseen sequences. Given enough computational resources, 3D shapes could be predicted for the whole TCR structure, as well as for the MHCs presenting the peptide. This would enable the design of models that, given the individual structures of each input sequence, will predict more accurate binding interactions.

Supporting information

S1 File. Supplementary Tables and Figures. Supplementary figures and tables referenced in the main text, including TCR-pMHC complex diagrams, additional model results, and extended correlation tables. See also Supplementary Sects 2.1–2.3 for details.
(PDF)

Author contributions

Conceptualization: Leonardo V. Castorina, Filippo Grazioli, Pierre Machart, Anja Mösch, Federico Errica.

Investigation: Leonardo V. Castorina, Filippo Grazioli, Pierre Machart, Federico Errica.

Methodology: Leonardo V. Castorina, Filippo Grazioli, Pierre Machart, Federico Errica.

Project administration: Leonardo V. Castorina, Filippo Grazioli, Pierre Machart.

Resources: Leonardo V. Castorina, Filippo Grazioli.

Software: Leonardo V. Castorina, Filippo Grazioli.

Supervision: Filippo Grazioli, Pierre Machart, Anja Mösch, Federico Errica.

Validation: Leonardo V. Castorina, Filippo Grazioli.

Visualization: Leonardo V. Castorina, Filippo Grazioli.

Writing – original draft: Leonardo V. Castorina, Filippo Grazioli, Pierre Machart, Anja Mösch, Federico Errica.

Writing – review & editing: Leonardo V. Castorina, Filippo Grazioli, Pierre Machart, Anja Mösch, Federico Errica.

References

1. Alberts B, Johnson A, Lewis J, Morgan D, Raff M, Roberts K. Molecular biology of the cell. WW Norton & Company. 2017.
2. Rowen L, Koop BF, Hood L. The complete 685-kilobase DNA sequence of the human beta T cell receptor locus. Science. 1996;272(5269):1755–62. https://doi.org/10.1126/science.272.5269.1755 PMID: 8650574
3. Glanville J, Huang H, Nau A, Hatton O, Wagar LE, Rubelt F, et al. Identifying specificity groups in the T cell receptor repertoire. Nature. 2017;547(7661):94–8. https://doi.org/10.1038/nature22976 PMID: 28636589

4. Hewitt EW. The MHC class I antigen presentation pathway: strategies for viral immune evasion. Immunology. 2003;110(2):163–9. https://doi.org/10.1046/j.1365-2567.2003.01738.x PMID: 14511229
5. Eisenlohr LC, Huang L, Golovina TN. Rethinking peptide supply to MHC class I molecules. Nat Rev Immunol. 2007;7(5):403–10. https://doi.org/10.1038/nri2077 PMID: 17457346
6. Smith-Garvin JE, Koretzky GA, Jordan MS. T cell activation. Annu Rev Immunol. 2009;27:591–619. https://doi.org/10.1146/annurev.immunol.021908.132706 PMID: 19132916
7. Feng D, Bond CJ, Ely LK, Maynard J, Garcia KC. Structural evidence for a germline-encoded T cell receptor-major histocompatibility complex interaction "codon". Nat Immunol. 2007;8(9):975–83. https://doi.org/10.1038/ni1502 PMID: 17694060
8. Rossjohn J, Gras S, Miles JJ, Turner SJ, Godfrey DI, McCluskey J. T cell antigen receptor recognition of antigen-presenting molecules. Annu Rev Immunol. 2015;33:169–200. https://doi.org/10.1146/annurev-immunol-032414-112334 PMID: 25493333
9. Qi Q, Liu Y, Cheng Y, Glanville J, Zhang D, Lee J-Y, et al. Diversity and clonal selection in the human T-cell repertoire. Proc Natl Acad Sci U S A. 2014;111(36):13139–44. https://doi.org/10.1073/pnas.1409155111 PMID: 25157137
10. Jameson SC, Masopust D. Understanding subset diversity in T cell memory. Immunity. 2018;48(2):214–26. https://doi.org/10.1016/j.immuni.2018.02.010 PMID: 29466754
11. Omilusik KD, Goldrath AW. Remembering to remember: T cell memory maintenance and plasticity. Curr Opin Immunol. 2019;58:89–97. https://doi.org/10.1016/j.coi.2019.04.009 PMID: 31170601
12. He Q, Jiang X, Zhou X, Weng J. Targeting cancers through TCR-peptide/MHC interactions. J Hematol Oncol. 2019;12(1):139. https://doi.org/10.1186/s13045-019-0812-8 PMID: 31852498
13. Waldman AD, Fritz JM, Lenardo MJ. A guide to cancer immunotherapy: from T cell basic science to clinical practice. Nat Rev Immunol. 2020;20(11):651–68. https://doi.org/10.1038/s41577-020-0306-5 PMID: 32433532
14. Gilbert SC. T-cell-inducing vaccines - what's the future. Immunology. 2012;135(1):19–26. https://doi.org/10.1111/j.1365-2567.2011.03517.x PMID: 22044118
15. Heslop HE, Leen AM. T-cell therapy for viral infections. Hematol Am Soc Hematol Educ Program. 2013;2013:342–7. https://doi.org/10.1182/asheducation-2013.1.342 PMID: 24319202
16. Springer I, Tickotsky N, Louzoun Y. Contribution of T cell receptor alpha and beta CDR3, MHC typing, V and J genes to peptide binding prediction. Front Immunol. 2021;12:664514. https://doi.org/10.3389/fimmu.2021.664514 PMID: 33981311
17. Montemurro A, Schuster V, Povlsen HR, Bentzen AK, Jurtz V, Chronister WD, et al. NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. Commun Biol. 2021;4(1):1060. https://doi.org/10.1038/s42003-021-02610-3 PMID: 34508155
18. Weber A, Born J, Rodriguez Martínez M. TITAN: T-cell receptor specificity prediction with bimodal attention networks. Bioinformatics. 2021;37(Suppl_1):i237–44. https://doi.org/10.1093/bioinformatics/btab294 PMID: 34252922
19. Grazioli F, Machart P, Mösch A, Li K, Castorina LV, Pfeifer N, et al. Attentive variational information bottleneck for TCR-peptide interaction prediction. Bioinformatics. 2023;39(1):btac820. https://doi.org/10.1093/bioinformatics/btac820 PMID: 36571499
20. Springer I, Besser H, Tickotsky-Moskovitz N, Dvorkin S, Louzoun Y. Prediction of specific TCR-peptide binding from large dictionaries of TCR-peptide pairs. Front Immunol. 2020;11:1803. https://doi.org/10.3389/fimmu.2020.01803 PMID: 32983088
21. Koyama K, Hashimoto K, Nagao C, Mizuguchi K. Attention network for predicting T-cell receptor-peptide binding can associate attention with interpretable protein structural properties. Front Bioinform. 2023;3:1274599. https://doi.org/10.3389/fbinf.2023.1274599 PMID: 38170146
22. Springer I, Tickotsky N, Louzoun Y. Contribution of T cell receptor alpha and beta CDR3, MHC typing, v and j genes to peptide binding prediction. Front Immunol. 2021;12:664514. https://doi.org/10.3389/fimmu.2021.664514 PMID: 33981311
23. Vita R, Mahajan S, Overton JA, Dhanda SK, Martini S, Cantrell JR, et al. The Immune Epitope Database (IEDB): 2018 update. Nucleic Acids Res. 2019;47(D1):D339–43. https://doi.org/10.1093/nar/gky1006 PMID: 30357391
24. Bagaev DV, Vroomans RMA, Samir J, Stervbo U, Rius C, Dolton G, et al. VDJdb in 2019: database extension, new analysis infrastructure and a T-cell receptor motif compendium. Nucleic Acids Res. 2020;48(D1):D1057–62. https://doi.org/10.1093/nar/gkz874 PMID: 31588507
25. Tickotsky N, Sagiv T, Prilusky J, Shifrut E, Friedman N. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. Bioinformatics. 2017;33(18):2924–9. https://doi.org/10.1093/bioinformatics/btx286 PMID: 28481982

26. Piepenbrink KH, Gloor BE, Armstrong KM, Baker BM. Methods for quantifying T cell receptor binding affinities and thermodynamics. Methods Enzymol. 2009;466:359–81. https://doi.org/10.1016/S0076-6879(09)66015-8 PMID: 21609868

27. Grazioli F, Mösch A, Machart P, Li K, Alqassem I, O'Donnell TJ, et al. On TCR binding predictors failing to generalize to unseen peptides. Front Immunol. 2022;13:1014256. https://doi.org/10.3389/fimmu.2022.1014256 PMID: 36341448

28. Deng L, Ly C, Abdollahi S, Zhao Y, Prinz I, Bonn S. Performance comparison of TCR-pMHC prediction tools reveals a strong data dependency. Front Immunol. 2023;14:1128326. https://doi.org/10.3389/fimmu.2023.1128326 PMID: 37143667

29. Mahajan S, Yan Z, Jespersen MC, Jensen KK, Marcatili P, Nielsen M, et al. Benchmark datasets of immune receptor-epitope structural complexes. BMC Bioinformatics. 2019;20(1):490. https://doi.org/10.1186/s12859-019-3109-6 PMID: 31601176

30. Korpela D, Jokinen E, Dumitrescu A, Huuhtanen J, Mustjoki S, Lähdesmäki H. EPIC-TRACE: predicting TCR binding to unseen epitopes using attention and contextualized embeddings. Bioinformatics. 2023;39(12):btad743. https://doi.org/10.1093/bioinformatics/btad743 PMID: 38070156

31. Moris P, De Pauw J, Postovskaya A, Gielis S, De Neuter N, Bittremieux W, et al. Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. Brief Bioinform. 2021;22(4):bbaa318. https://doi.org/10.1093/bib/bbaa318 PMID: 33346826

32. Croce G, Bobisse S, Moreno DL, Schmidt J, Guillame P, Harari A, et al. Deep learning predictions of TCR-epitope interactions reveal epitope-specific chains in dual alpha T cells. Nat Commun. 2024;15(1):3211. https://doi.org/10.1038/s41467-024-47461-8 PMID: 38615042

33. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422–3. https://doi.org/10.1093/bioinformatics/btp163 PMID: 19304878

34. Schrödinger LLC. The PyMOL molecular graphics system, version 1.8. 2015.

35. Shugay M, Bagaev DV, Zvyagin IV, Vroomans RM, Crawford JC, Dolton G, et al. VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. Nucleic Acids Res. 2018;46(D1):D419–27. https://doi.org/10.1093/nar/gkx760 PMID: 28977646

36. Jurtz V, Jessen L, Bentzen A, Jespersen M, Mahajan S, Vita R. NetTCR: sequence-based prediction of TCR binding to peptide-MHC complexes using convolutional neural networks. BioRxiv. 2018:433706.

37. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, et al. Evolutionary-scale prediction of atomic level protein structure with a language model. Cold Spring Harbor Laboratory. 2022. https://doi.org/10.1101/2022.07.20.500902

38. Wu R, Ding F, Wang R, Shen R, Zhang X, Luo S, et al. High-resolutionde novostructure prediction from primary sequence. Cold Spring Harbor Laboratory. 2022. https://doi.org/10.1101/2022.07.21.500999

39. Wagner RA, Fischer MJ. The string-to-string correction problem. J ACM. 1974;21(1):168–73. https://doi.org/10.1145/321796.321811

40. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A. 1992;89(22):10915–9. https://doi.org/10.1073/pnas.89.22.10915 PMID: 1438297

41. Moore PB, Hendrickson WA, Henderson R, Brunger AT. The protein-folding problem: not yet solved. Science. 2022;375(6580):507. https://doi.org/10.1126/science.abn9422 PMID: 35113705

42. Jensen MF, Nielsen M. NetTCR 2.2 - improved TCR specificity predictions by combining pan- and peptide-specific training strategies, loss-scaling and integration of sequence similarity. eLife Sciences Publications, Ltd. 2024. https://doi.org/10.7554/elife.93934.2

43. Mechelke M, Habeck M. Robust probabilistic superposition and comparison of protein structures. BMC Bioinformatics. 2010;11:363. https://doi.org/10.1186/1471-2105-11-363 PMID: 20594332

44. Ballone A, Lau RA, Zweipfenning FPA, Ottmann C. A new soaking procedure for X-ray crystallographic structural determination of protein-peptide complexes. Acta Crystallogr F Struct Biol Commun. 2020;76(Pt 10):501–7. https://doi.org/10.1107/S2053230X2001122X PMID: 33006579

45. 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):583–9. https://doi.org/10.1038/s41586-021-03819-2 PMID: 34265844

46. van Breugel M, Rosa E Silva I, Andreeva A. Structural validation and assessment of AlphaFold2 predictions for centrosomal and centriolar proteins and their complexes. Commun Biol. 2022;5(1):312. https://doi.org/10.1038/s42003-022-03269-0 PMID: 35383272

47. Goswami A, Kumar SM, Ullah S, Gore MM. De novodesign of anti-variant COVID-19 Vaccine. Cold Spring Harbor Laboratory. 2022. https://doi.org/10.1101/2022.10.20.513049

48. Chang L, Perez A. Ranking peptide binders by affinity with AlphaFold**. Angewandte Chemie. 2023;135(7). https://doi.org/10.1002/ange.202213362

49. Castorina LV, Ünal SM, Subr K, Wood CW. TIMED-Design: flexible and accessible protein sequence design with convolutional neural networks. Protein Eng Des Sel. 2024;37:gzae002. https://doi.org/10.1093/protein/gzae002 PMID: 38288671

50. Watson JL, Juergens D, Bennett NR, Trippe BL, Yim J, Eisenach HE, et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. Cold Spring Harbor Laboratory. 2022. https://doi.org/10.1101/2022.12.09.519842

51. Castorina LV, Petrenas R, Subr K, Wood CW. PDBench: evaluating computational methods for protein-sequence design. Bioinformatics. 2023;39(1):btad027. https://doi.org/10.1093/bioinformatics/btad027 PMID: 36637198

52. Berman H, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. Nat Struct Biol. 2003;10(12):980. https://doi.org/10.1038/nsb1203-980 PMID: 14634627

53. Lee CH, Salio M, Napolitani G, Ogg G, Simmons A, Koohy H. Predicting cross-reactivity and antigen specificity of T cell receptors. Front Immunol. 2020;11:565096. https://doi.org/10.3389/fimmu.2020.565096 PMID: 33193332

54. Coles CH, Mulvaney RM, Malla S, Walker A, Smith KJ, Lloyd A, et al. TCRs with distinct specificity profiles use different binding modes to engage an identical peptide-HLA complex. J Immunol. 2020;204(7):1943–53. https://doi.org/10.4049/jimmunol.1900915 PMID: 32102902

55. Kersh GJ, Miley MJ, Nelson CA, Grakoui A, Horvath S, Donermeyer DL, et al. Structural and functional consequences of altering a peptide MHC anchor residue. J Immunol. 2001;166(5):3345–54. https://doi.org/10.4049/jimmunol.166.5.3345 PMID: 11207290

56. Nguyen AT, Szeto C, Gras S. The pockets guide to HLA class I molecules. Biochem Soc Trans. 2021;49(5):2319–31. https://doi.org/10.1042/BST20210410 PMID: 34581761

57. Barton MI, MacGowan SA, Kutuzov MA, Dushek O, Barton GJ, van der Merwe PA. Effects of common mutations in the SARS-CoV-2 Spike RBD and its ligand, the human ACE2 receptor on binding affinity and kinetics. Elife. 2021;10:e70658. https://doi.org/10.7554/eLife.70658 PMID: 34435953

58. Xue S, Han Y, Wu F, Wang Q. Mutations in the SARS-CoV-2 spike receptor binding domain and their delicate balance between ACE2 affinity and antibody evasion. Protein Cell. 2024;15(6):403–18. https://doi.org/10.1093/procel/pwae007 PMID: 38442025

59. Qi Y, Zhang JZH. DenseCPD: improving the accuracy of neural-network-based computational protein sequence design with DenseNet. J Chem Inf Model. 2020;60(3):1245–52. https://doi.org/10.1021/acs.jcim.0c00043 PMID: 32126171

60. Ji H, Wang X-X, Zhang Q, Zhang C, Zhang H-M. Predicting TCR sequences for unseen antigen epitopes using structural and sequence features. Brief Bioinform. 2024;25(3):bbae210. https://doi.org/10.1093/bib/bbae210 PMID: 38711371

61. Li F, Qian X, Zhu X, Lai X, Zhang X, Wang J. TCRcost: a deep learning model utilizing TCR 3D structure for enhanced of TCR-peptide binding. Front Genet. 2024;15:1346784. https://doi.org/10.3389/fgene.2024.1346784 PMID: 39415981

62. Bradley P. Structure-based prediction of T cell receptor:peptide-MHC interactions. Elife. 2023;12:e82813. https://doi.org/10.7554/eLife.82813 PMID: 36661395

63. Pham M-DN, Su CT-T, Nguyen T-N, Nguyen H-N, Nguyen DDA, Giang H, et al. epiTCR-KDA: knowledge distillation model on dihedral angles for TCR-peptide prediction. Cold Spring Harbor Laboratory. 2024. https://doi.org/10.1101/2024.05.18.594806

64. Carugo O. Statistical validation of the root-mean-square-distance, a measure of protein structural proximity. Protein Eng Des Sel. 2007;20(1):33–7. https://doi.org/10.1093/protein/gzl051 PMID: 17218333

65. Postovskaya A, Vercauteren K, Meysman P, Laukens K. tcrBLOSUM: an amino acid substitution matrix for sensitive alignment of distant epitope-specific TCRs. Brief Bioinform. 2024;26(1):bbae602. https://doi.org/10.1093/bib/bbae602 PMID: 39576224

66. Castorina LV, Wood CW, Subr K. From atoms to fragments: a coarse representation for functional and efficient protein design. Cold Spring Harbor Laboratory. 2025. https://doi.org/10.1101/2025.03.19.644162