scientific reports

Check for updates

OPEN

Assessing data size requirements for training generalizable sequence-based TCR specificity models via pan-allelic MHC-I point-mutation ligandome evaluation

Antoine Delaunay[1,3]✉, Miles McGibbon[1,3]✉, Bachir Djermani[1], Nikolai Gorbushin[1], Sergio Chaves García-Mascaraque[1], Isaac Rayment[1], Ilya Kizhvatov[2], Cécile Petit[2], Maren Lang[2], Karim Beguir[1], Ugur Sahin[2], Liviu Copoiu[1], Nicolas Lopez Carranza[1] & Andrey Tovchigrechko[2]✉

Rapid identification of T cell receptors (TCRs) that specifically bind patient-unique neoepitopes is a critical challenge for personalized TCR-based therapies in oncology. Due to enormous diversity of both TCR and neoepitope repertoires, a machine learning predictor of TCR-pMHC specificity for personalized therapy must generalize to TCRs and epitopes not seen in the training data. We estimate the necessary size of such training data. We first confirm that published models fail to generalize beyond a single-residue dissimilarity to the epitope training set distribution. We then impute the point-mutation ligandome across the 34 most prevalent human MHC alleles and represent it as a graph based on our established dissimilarity cutoff. By finding the dominating set of this graph, we estimate that between one and 100 million epitopes are required to train a generalizable sequence-based TCR specificity prediction model—1000 times the size of current public data.

Keywords TCR specificity prediction, Oncology, Personalized T cell therapy, Sample size estimation

The adaptive immune system is a critical part of the immune response to infections and tumor cells. It contains a diverse set of T lymphocytes, each having surface receptors targeting specific antigens. This study focuses on antigens presented to cytotoxic T cells, specifically tumor neoantigens; novel mutant peptides derived from normal self tissues via somatic mutations during tumorigenesis and tumor progression. For antigen presentation, a mutated protein within the cell undergoes cleavage by the proteasome into peptides. Subsequently, these peptides translocate to the endoplasmic reticulum, where they can bind to the class-I Major Histocompatibility Complex (MHC) to form a peptide-MHC protein complex (pMHC)[1] (Fig. 1a–c). The resulting pMHC complex may eventually migrate to the cell surface, where it becomes accessible for recognition by a cytotoxic T cell receptor (TCR). The binding of the TCR to the pMHC is a necessary but not sufficient condition for T cell activation, which is also dependent on other biological factors such as mediator proteins[1]. Class-I MHC proteins are composed of two chains, alpha and beta. The alpha chain forms a binding groove resembling a pocket to which the peptide is anchored, leaving middle residues exposed for TCR recognition[2]. In contrast, the beta(2)-microglobulin chain is not in direct interaction with the peptide. The peptide sequence is typically 8-12 amino acids long for class-I MHCs. The TCR consists of two chains with highly variable regions known as Complementarity-Determining Region (CDR) loops. In particular, the third CDR of the β-chain (CDR3β) loop of the TCR is more critical than other CDR loops in the interaction with the peptide and in determining the specificity of the TCR-pMHC binding[3] (Fig 1c). Its length and amino acid sequence are highly variable, which reflects the high diversity of T cell receptors selected in the thymus[4].

Accurate prediction of TCR-pMHC specificity would be highly beneficial for designing personalized T cell receptor-based cancer immunotherapies[5,6] without requiring a lengthy experimental screening process. Such

[1]InstaDeep Ltd, 5 Merchant Sq, London, UK. [2]BioNTech SE, An der Goldgrube 12, Mainz, Germany. [3]Antoine Delaunay and Miles McGibbon contributed equally to this work. ✉email: a.delaunay@instadeep.com; m.mcgibbon@instadeep.com; andreyto@gmail.com

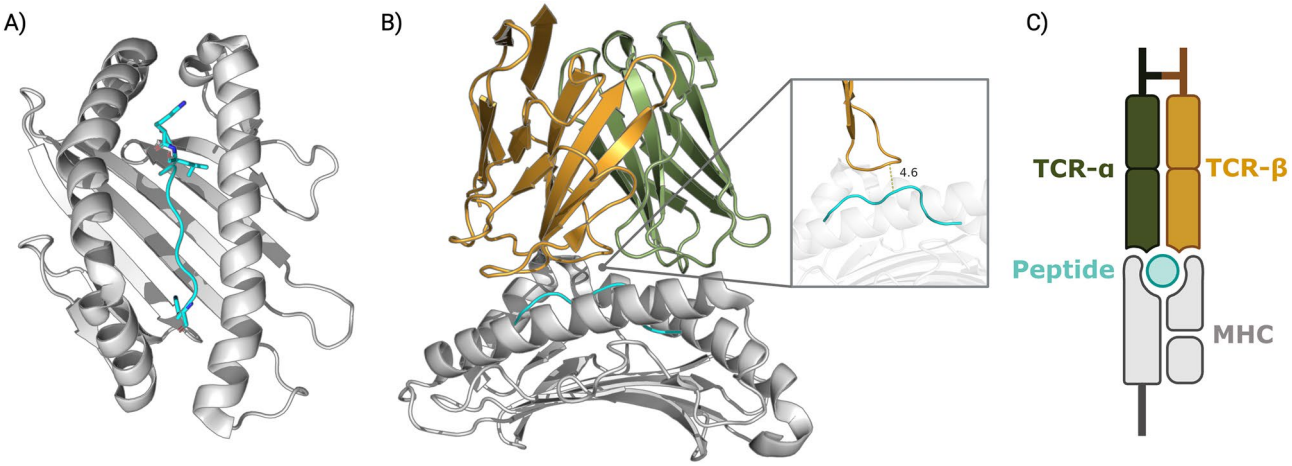


Fig. 1. Structural basis of TCRpMHC specificity demonstrated with PDB structure 7phr[38]. (**A**) Peptide (cyan) binding in the MHC-I binding groove (grey) is driven by N and C terminal anchor residues (sticks). (**B**) Presented peptide (cyan) recognition by TCR variable alpha (green) and beta (orange) domains is primarily modulated by CDR3β loop (zoomed: CDR3B in orange, backbone distance in angstroms). (**C**) Cartoon schematic of pMHC recognition (cyan/grey) recognition by TCR (orange/green). Macromolecular renderings were produced with PyMOL https://pymol.org/https://pymol.org/. Cartoon schematic was created with Biorender https://www.biorender.com/https://www.biorender.com/.

personalized therapy can cover a very broad patient population. Unlike models which rely on the observed phenotype of T cells in the patient to identify likely tumor-reactive TCR clones[5], prediction of specificity based on TCR sequence would not be affected by confounding factors such as prior treatment history and differences between cancer indications. Each patient has a largely unique set of TCRs generated through the process of random recombination during thymic development. Moreover, the neoantigens exposed on tumor cells are mostly unique to each individual[7], with each neoantigen presented by MHC molecules which also differ between patients. Thus, a useful model that predicts TCR-pMHC binding specificity for personalized TCR-based treatment should generalize to previously unseen peptide-MHC complexes and TCRs. Machine learning (ML) models learn patterns from the training data of known examples in order to make predictions for new data. Deep learning (DL) models have had notable success in recent years to predict protein interactions by learning statistical patterns from large datasets of biological sequences. NetMHCPan 4.1[8] and MHCFlurry[9] are robust predictors of peptide-MHC binding. Recently, several models including ERGO[10], TULIP[11], ATM-TCR[12], PiTE[13] and TCR-H[14] have tried to leverage datasets of reactive TCR-pMHC sequences as training data to predict TCR-pMHC binding specificity using DL models. However, although efforts have been made to collect more TCR-pMHC binding data, the most widely used sequence-based databases such as IEDB[15], VDJdb[16] and McPasTCR[17] exhibit a low number of unique peptides compared to the number of TCRs, with a very uneven distribution of TCRs across peptides, and they lack the experimental examples of nonbinding pairs. This could introduce a bias into both training and evaluation procedures and lead to an overestimation of generalization capabilities on unseen data.

In this study, we benchmarked several representative TCR-pMHC binding specificity models for their predictive accuracy on novel peptide epitopes and TCRs using a carefully constructed dataset to eliminate possible causes of inflated performance. Our findings revealed a universal shortcoming in the ability of these models to accurately predict TCR-binding specificity for previously unseen peptides, which was also found for earlier models in the previous benchmarking efforts[11,18]. From this stage of the analysis, we derived the maximum sequence dissimilarity between evaluation and training peptides at which the performance remains acceptable. Based on this result, we aimed to determine how many training data samples would be needed to cover the entire space of potential human tumor neoantigens primarily derived from single nucleotide variants (SNVs) in the coding sequence and presented on MHC class-I molecules. We focus the estimation on SNVs for computational tractability and because SNVs in the protein-coding regions comprise the majority of candidate targets identified by the current methods which discover personalized tumor neoantigens. To achieve this, we formulated the problem as a graph where each vertex represents a peptide sequence in the space of tumor neoantigens and each edge connects peptides within the specified edit distance from each other. Leveraging graph theory, we identified a dominating set, defined as a subset of vertices, ensuring that every vertex in the graph is either within the set or adjacent to it. The calculated dominating set represents the minimum set of peptides required for training a sequence-based ML model with acceptable performance on the unseen epitopes. The overall design of our sample size estimation is illustrated on (Supplementary Fig. 1)

To the best of our knowledge, this is the first attempt at estimating the sufficient training data size for this important and intensely studied prediction problem. This estimation represents the main novel contribution of our work.

Results

Specificity prediction models fail to generalize to unseen epitopes

Here, we assessed the performance of several published models which aim to predict whether a given TCR-pMHC pair will bind. We began by comparing their average performance on peptides unseen during training, using the dataset split described in Section "TCR-pMHC binding data"). Figure 2a and b showcase the area under the receiver operating characteristic curve (ROC-AUC) and area under the precision recall curve (PR-AUC) metrics on unseen peptides. All models demonstrated poor generalization capabilities when presented with unseen peptides, performing close to a random classifier with a ROC-AUC of 0.5. The uniformity of performance variance among models indicates that the model architectures were likely not the cause of poor performance. Rather, the deficiency in training data availability appeared to be the predominant factor. Note that this performance obtained with the proper unbiasing of both data and the evaluation procedure (see Methods) is in agreement with the results reported by the STAPLER[19] team on their proprietary experimental test dataset that was generated independently of the training data.

In light of these results, we refined this initial analysis by stratifying the results of the unseen peptide test case based on the minimum peptide edit distance to the training set. All models demonstrated a failure to generalize effectively, as evidenced by their ROC-AUC hovering around 0.5, regardless of the edit distance, indicative of performance akin to random classifiers (Fig 2c). Specifically, there was a discernibly weak signal for ATM-TCR

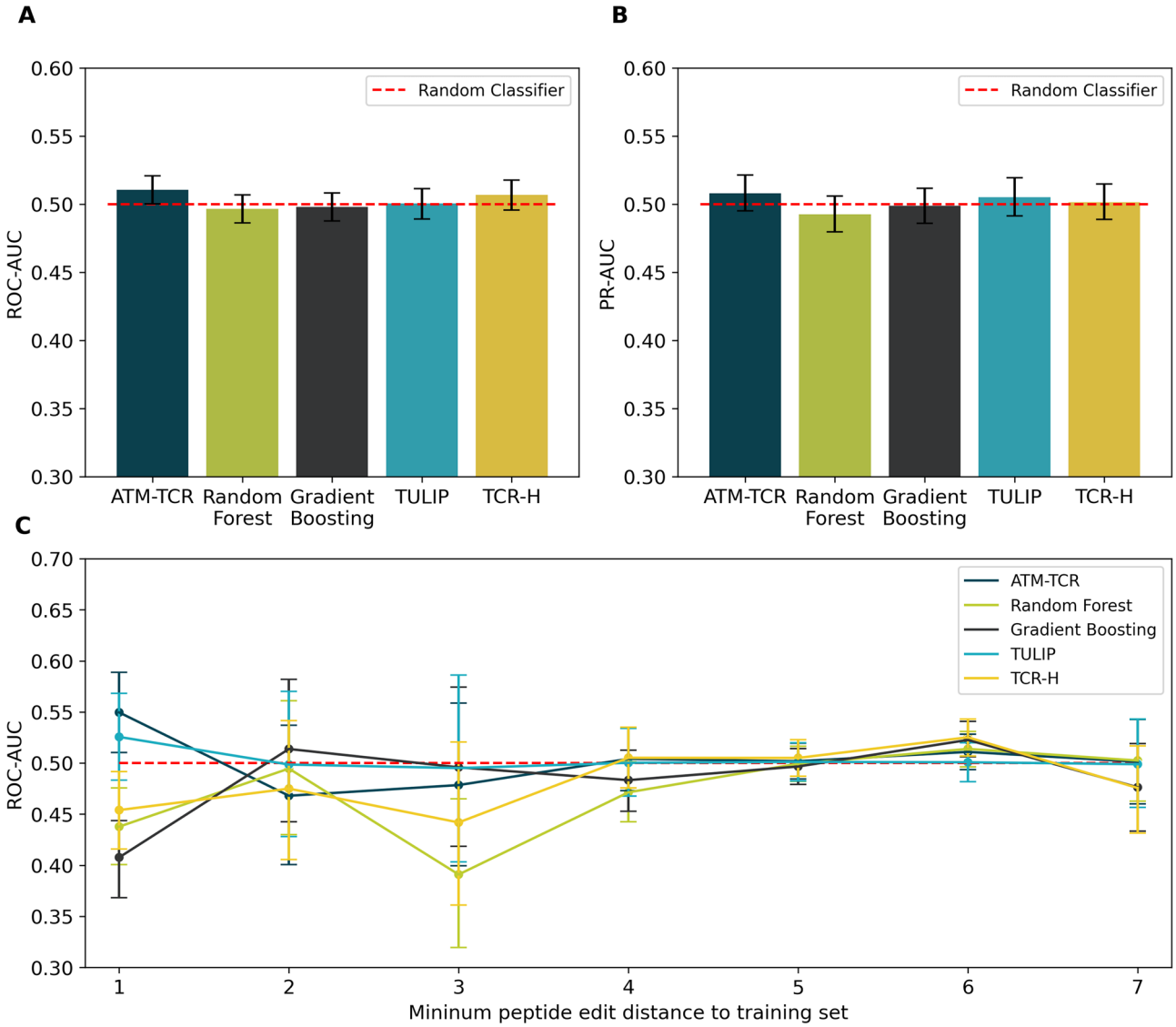


Fig. 2. Model generalization abilities for TCRpMHC classification. (**A**) Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) of evaluated models on unseen test peptides. (**B**) Precision-Recall AUC (PR-AUC) of evaluated models on unseen test peptides. (**C**) Model performance stratified by the test peptide minimum edit distance to the training set. Points show weighted ROC–AUC with vertical error bars indicating 90% confidence intervals from peptide-cluster bootstrap computed as described in the text.

at an edit distance of 1, as judged by the lower range of a 90% confidence interval (CI) located above the 0.5 threshold.

We obtained similar results from an alternative 5-fold cross-validation experiment (referred to as CV split in the following text) derived from the entire TULIP dataset (Supplementary Fig. 2). This experiment confirmed that our results were not specific to the original way in which the TULIP dataset was split into training and testing sets. One exception was the results from the Gradient Boosting model, that showed slightly above-random lower CI at edit distance 1 in the CV split experiment, while showing <0.45 upper CI in the TULIP split experiment at the same edit distance. Considering that this model showed a random-level performance at distance bins above 1, it is likely that its generalization at edit distance 1 in the CV split is an overfitting in which this model exploited hidden correlations in the training or testing data.

As a complement to our main analysis stratified by peptide edit distance, we assessed how the number of training TCRs paired with the closest peptide in the training set affects model performance. For each test example, we identified the nearest training peptide by minimal edit distance and binned it by the number of associated TCRs from the positive examples, with bins 1–10, 11–100 and > 100. ROC-AUC was then computed across test examples in each bin for all five models. These bins were applied to both TULIP split and CV split to obtain comparable graphical representations for both experiments (Supplementary Figs. 3 and 4), although there were very few peptides populating the 1–10 bin in the TULIP split due to its construction procedure. If we focus our attention on bins with at least 25 test peptides to avoid conclusions being driven by just a few peptides, we can see that there is a discernible positive, although weak, association between performance and training TCR count only at edit distance 1. Even at that distance, it is not entirely consistent. For example, the performance drops for ATM-TCR in the TULIP split. The nonmonotonic trends along the Y axis across neighboring edit distance bins likely reflect hidden correlations in the data and the fact that data compiled in the public databases was generated to answer biological questions specific to respective studies rather than for unbiased training of specificity prediction models.

We also observed that TULIP performance (Fig 2c) notably deviated from that reported in the original publication[11]. To better understand the performance gap between our evaluation and the original TULIP publication, we performed a series of diagnostic tests using the pretrained TULIP checkpoint and their original test set, with both model weights and data downloaded from the original GitHub repository (https://github.com/barthelemymp/TULIP-TCR/blob/main/data/UnseenPeptides.zip). The results are shown in Supplementary Fig. 6.

The first source of the discrepancy in the reported performance lies in the differences between evaluation criteria. The original TULIP publication evaluated ROC-AUC per peptide and subsequently averaged ROC-AUC across peptides. Their approach circumvented the need to estimate the marginal probability of observing a given epitope peptide inherent in their Bayesian formalism. In contrast, our study addresses a more challenging and practically relevant task: predicting binding of a given $\mathrm{CDR3}\beta$ against any candidate peptide. In the case of the TULIP model, this amounted to assigning an equal marginal probability to each candidate epitope peptide and using TULIP-generated scores for classification because their publication did not provide a method for estimating this marginal probability. Therefore, in Supplementary Fig. 6, we show performance for our across-epitopes and TULIP's within-epitope ROC-AUC aggregation approaches. We consistently observe a drop in performance for the across-epitopes metric.

When evaluating the model on the TULIP unmodified test dataset and using their within-epitope ROC-AUC aggregation method, we closely reproduced ROC-AUC metrics reported in the TULIP publication. However, we noticed that all negative test examples in the original TULIP dataset had the sequence of the $\mathrm{CDR3}\alpha$ chain present, while the positive test examples were distributed similarly to the training dataset in terms of the $\mathrm{CDR3}\alpha$ chain availability. We performed a diagnostic resampling experiment by shuffling $\mathrm{CDR3}\alpha$ sequences between those test examples which had non-missing $\mathrm{CDR3}\alpha$ entries. We observed a negligible impact on performance, indicating that the model has learned to predict TCR-pMHC specificity from the missingness status of the α chain information rather than the actual sequence.

Finally, while none of the negative examples in the TULIP test set had TCR $\mathrm{CDR3}\beta$ chains found in the training set, 16% of the positive testing examples showed overlap with the training set, pointing to a possibility that a model might have exploited this imbalance as a label leak. After we applied our curated version of their test set in which the construction of the negative examples was rebalanced to eliminate such biases, the performance dropped toward the random classifier level. In summary, the results shown in Supplementary Fig. 6 support the validity of our metrics shown for TULIP on our main Fig. 2c and explain the difference with the original TULIP publication by either a difference in evaluation metric or by subtle label leaks inherent in the construction of the original negative test dataset in the TULIP publication.

Overall, only ATM-TCR demonstrated better than random performance, and only on epitopes a maximum of edit distance 1, or possibly 2, away from those seen during training. Based on this result, we then estimated the number of training datapoints required to effectively cover the entire human MHC-I SNV ligandome at edit distances 1, 2 and 3, and achieve a comparable ROC-AUC in real-world applications to newly sequenced tumors.

Estimates based on experimentally determined ligandomes exceed one million peptides required to train a generalizable model of TCR specificity

To provide lower bound estimates for the amount of training data required for a generalizable TCRpMHC specificity model, expanded experimental ligandome dominating set sizes at edit distances 1, 2 and 3 were calculated for 34 alleles (see Section "Experimental ligandome estimations"). These 34 alleles were chosen to maximize population coverage (see Section "MHC-I Allele Selection").

Large inter-allele variation was seen for dominating sets at all edit distance thresholds (Fig. 3). Dominating sets at edit distance 1 showed the largest variation; A*02:01's dominating set was estimated at over 100,000

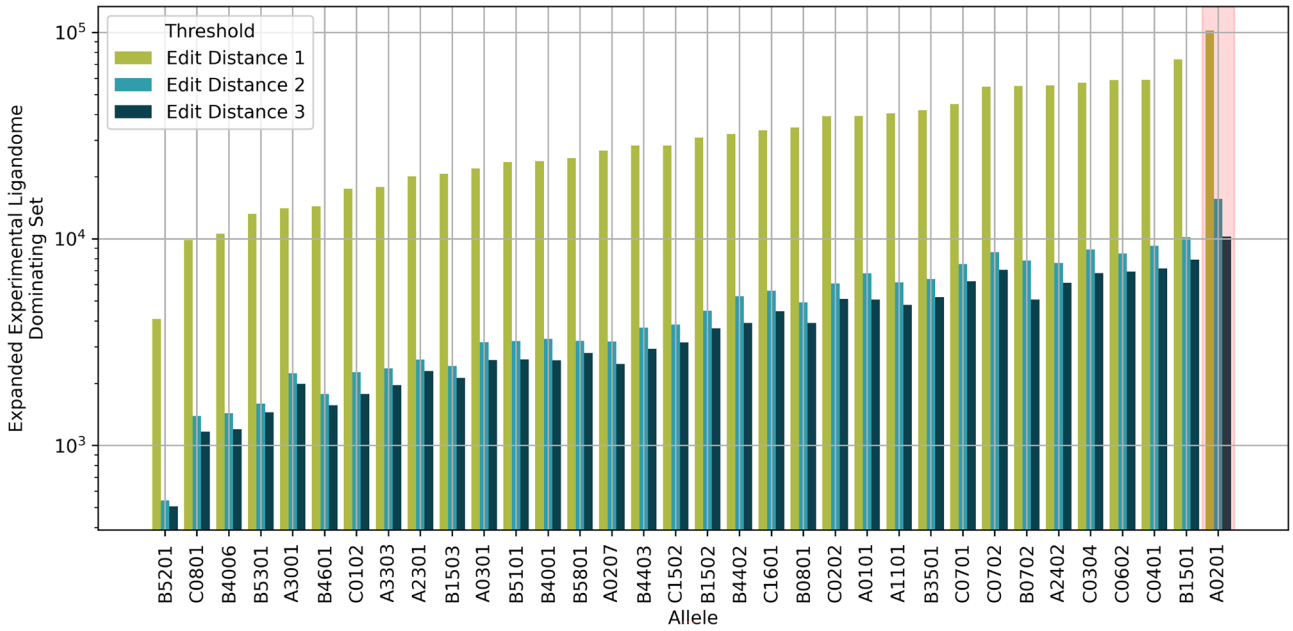


Fig. 3. Expanded experimental ligandome dominating set sizes across edit distance thresholds 1–3 for 34 representative MHC-I alleles. Allele A*02:01 is highlighted in red.

Edit distance threshold	Size	Parent ligandome
1	1,174,573	Expanded experimental
2	172,000	
3	135,064	
1	110,392,479	Union
2	21,277,040	
3	3,035,244	

Table 1. Dominating set sizes for expanded experimental and union ligandomes.

unique 9-mers, whilst B*52:01's dominating set estimation was fewer than 3,000 unique peptides. Dominating sets solved at edit distance 1 were an order of magnitude larger than those solved at edit distances 2 and 3.

Expanded experimental ligandome dominating sets for all 34 alleles were combined into a pan-allelic dominating set (Table 1). These values represent a lower, experimentally supported bound of the minimum number of unique peptides required to train a classifier capable of generalizing across a given edit distance threshold. For example, if a given model architecture can generalize to peptides of edit distance 3 or below from the training set, only 135,000 peptides are required for training. Again, we considered a peptide-allele pair as a unique data point; identical peptides present in dominating sets for multiple alleles were all counted as separate instances due to the importance of TCR-MHC interactions.

As these estimates were based on experimental data for each allele, we next investigated the effect of available public data quantity on expanded experimental ligandome size estimates. As NetMHCPan 4.1 is a machine learning based predictor, we assessed correlations between the amount of experimental data in NetMHCPan 4.1's training set and our expanded experimental ligandome estimates (Supplementary Fig. 7a). We observed a strong, highly significant positive correlation between these two variables.

In addition to potential NetMHCPan 4.1 bias, the initial number of experimental data points available directly influenced the size of the experimental synthetic mutanome, and therefore could affect final expanded experimental ligandome sizes. When examining correlations between these two variables, we again found that experimental ligandome size showed a strong, highly significant positive correlation with expanded experimental ligandome sizes for a given allele (Supplementary Fig. 7b). The observed correlation between amount of experimental data and expanded experimental ligandome size was both stronger and more significant than the correlation between NetMHCPan 4.1 training data and expanded experimental ligandome size.

In the context of the results in (1.1), this suggests the lower bound of experimental data points required to train a useful ATM-TCR style model is over 1,174,000. However, investigating possible biases revealed significant correlations with the amount of available experimental data.

Estimates based on theoretical ligandomes predicted from the human proteome indicate more than a hundred million peptides required for training data

Due to possible bias in (1.2), we next calculated upper bound estimates for ligandome sizes based on the entire human proteome for all 34 chosen alleles (see Section "Predicted ligandome estimations"). Union ligandome sizes were an order of magnitude larger than expanded experimental ligandome estimates, ranging from $1.3 * 10^7$ to $3.5 * 10^7$ per allele (Supplementary Fig. 8a).

Unlike experimental estimates where A*02:01 had the largest ligandome, C*06:02 had the largest estimated union ligandome. Indeed, no significant correlation was seen between experimental ligandome sizes and union ligandome sizes (Supplementary Fig. 8b). To further investigate possible drivers of the difference in union ligandome sizes between alleles, we again compared final estimates to the amount of NetMHCPan 4.1 training data and found no significant correlation (Supplementary Fig. 8a).

The brute force calculation of an all-to-all edit distance matrix for a given set of peptides is computationally intensive, with $O(n^2)$ time and compute complexity. While this is possible for smaller sets, it is unfeasible for million scale peptide pools even with modern compute capabilities. Despite our employment of a heuristic method for edit distance calculation, we were unable to calculate edit distance matrices for entire union ligandomes. Hence, we employed a subsampling technique to estimate final dominating set sizes (see Section "Union ligandome subsampling"; Fig. 4a). Estimated union ligandome dominating set sizes for A*02:01 were, again, an order of magnitude larger than the expanded experimental ligandome dominating sets (Fig. 4a). Using parameters obtained from A*02:01 sampling, the dominating set sizes were estimated for all 34 alleles (Fig. 4b).

Finally, we took the sum of the union ligandome estimated dominating set sizes for all 34 alleles (Table 1). These represent a theoretical upper bound for the amount of data required to cover the ligandomes of all 34 considered alleles at each given edit distance threshold. Notably, both the upper bound calculated here and the

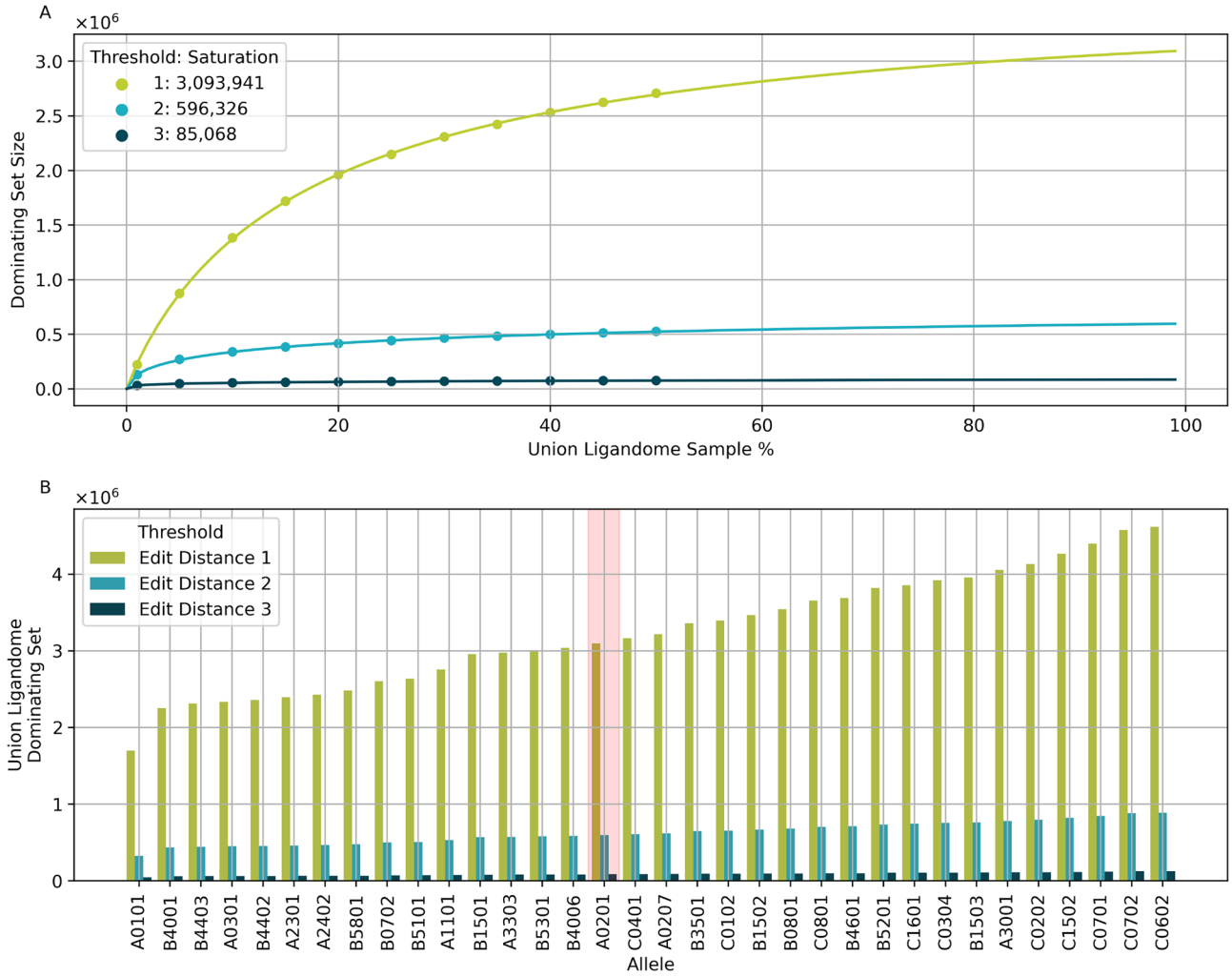


Fig. 4. Estimation of dominating set sizes for union ligandomes. (**A**) Dominating set sizes for A*02:01 union ligandome using random subsamples up to 50%; displayed saturation values at 100% are estimated from fitted curves. (**B**) Estimated dominating set sizes for union ligandomes of 34 considered alleles. Allele A*02:01 is highlighted in red.

lower bound calculated in (1.2) exceed the amount of training data currently available in public datasets by three orders of magnitude.

Discussion

Our benchmarking of published and baseline ML models confirms the importance of taking into account similarities between training and testing datasets at the TCR and epitope level, as well as considering other correlations which can confound performance evaluation. All evaluated models exhibited random or near-random performance on unseen epitopes, with large variance in AUC metrics across folds. Our results are in agreement with a recently published open source competition assessing TCR specificity prediction on unseen epitopes[20]. The cross-validation studies performed here demonstrate the ability of some architectures to generalize to epitopes highly similar to those seen during training, but not those that are dissimilar; ATM-TCR shows better-than-random performance on epitopes up to edit distance 1, and possibly 2, away from training set examples.

Our datasets constructed from public databases show much higher $\mathrm{CDR3}\beta$ diversity than epitope diversity. When we consider previously published better-than-random results on TCR-based splits (confirmed in our benchmarking and not reported here), we conclude that the limited number of unique peptide epitopes available for training constitutes the primary bottleneck for generalization of the sequence-based TCR-pMHC specificity prediction models.

A useful TCR-pMHC specificity prediction model for personalized TCR-based immunotherapy design must be able to generalize to any possible neoantigen presented on a given patient's MHC-I. Here, we use a representative set of 34 MHC-I alleles and an MHC-I binding predictor (NetMHCPan 4.1) to identify upper and lower estimates for the size of SNV-derived non-self ligandomes. Our lower, experimentally supported estimates suggest that 1,174,573, 172,000, and 135,064 unique epitopes are required to train a specificity model that can generalize to unseen epitopes at an edit distance of 1, 2, and 3, respectively. However, we observe strong correlations between expanded experimental ligandome sizes and the amount of public experimental data for a given allele. This suggests that our estimates for less studied alleles with few experimental data points may be large underestimations. Indeed, the theoretical 9-mer space is greater than $5*10^{11}$, making it unlikely any MHC-I allele will ever be exhaustively experimentally screened. Hence, we thoroughly investigated our union ligandomes for their reliability as upper bound estimates.

Interestingly, we see orders of magnitude differences in union ligandome sizes for different MHC-I alleles. The estimates for union ligandome dominating set sizes at various edit distance thresholds show no significant correlation with NetMHCPan training data or publicly available training data. Therefore, the correlation between NetMHCPan training data volume and experimental ligandome size is likely driven by separate correlations between both of these variables and the volume of public peptide-MHC data available for a given allele. Indeed, union ligandome size differences may reflect true variation in the number of peptides each MHC-I can bind. These findings are consistent with other work noting the differences in magnitude between MHC-I ligandomes[21]. Hence, our produced union ligandomes are likely reasonable upper bound estimations, suggesting that 110,392,479, 21,277,040, and 3,035,244 unique epitopes are required to train a sequence-based specificity model that can generalize to unseen epitopes at an edit distance of 1, 2, and 3, respectively. To our knowledge, this is the first exhaustive estimation of a pan-allelic neoantigen ligandome. These estimations required significant engineering and computing infrastructure; we share the results with the scientific community hoping that they will help focus further efforts in both data generation and modeling.

We note the following limitations in the work presented here. Our compilation of the available public TCR-pMHC binding data contains 1,407 unique epitopes, representing just 1.04% and 0.05% of the lower and upper estimates for data required to train a model that can generalize at edit distance 3, respectively. These estimates do not consider the possibility of "emergent" model behavior, in which a model with the generalization distance of 1 would start generalizing at progressively longer edit distances as more and more data is added to the training set. The presence of such emergent behavior could be tested from the learning curves[22] by evaluating the generalization edit distance as a function of dataset size upon iterative subsampling, approximating with analytical form such as a power-law function and then extrapolating to future larger training sizes. As we have shown, the performance on the full set of currently available data is close to a random baseline with wide confidence intervals. That means we can only sample the very beginning of the learning curve, and extrapolation beyond the current data size will have extreme uncertainty making such an estimation non-informative. A theoretical alternative to resampling of real training data could be the generation of synthetic ground truth data. For example, TULIP has the capability to generate novel sequences of TCR binders given a target epitope. We could use TULIP or another generative model to emit synthetic binders for a subset of our simulated non-self ligandome, and then use this dataset as ground truth data to build a learning curve for ATM-TCR and other specificity prediction models as discussed above. However, we would only evaluate the capacity of a model such as ATM-TCR to learn the distribution produced by another model, the generator. It is reasonable to assume that this distribution will not capture the full biological and experimental complexity of real TCR-pMHC interactions. For example, if TULIP was able to recapitulate the diversity matching a real TCR-pMHC distribution, it would have shown generalization in specificity prediction on real test data, which in our benchmarking was not observed. As such, any scaling trends derived from synthetic data will not serve as proof of behavior on future real data, which is the object of our present study.

Furthermore, due to the size of union ligandomes, and the computational complexity of solving the minimum dominating set problem, we employed several heuristic methods in producing these estimates. The SACHICA algorithm used for edit distance clustering has been validated as highly accurate for detecting similar short strings with fixed alphabets, and is unlikely to introduce errors which affect the conclusions of this work[23]. However, the fast greedy algorithm employed for dominating set solving may return up to $2n$ nodes for a

minimum dominating set of size n[24]. Additionally, NetMHCPan 4.1 is employed here as predictive model for MHC-I binding, with a median positive predictive value reported in the literature of 0.83[8]. Hence, in a worst case scenario these union ligandomes are an overestimation of their true values by more than a factor of two. Yet, the lack of experimental data, bias in experimentally supported estimates, and computational intractablity of brute force approaches to clustering and dominating set solving means more accurate estimations are currently out of reach.

Additionally, the performance of evaluated models on epitope based splits shows large variation. This may be due to the large imbalances in the number of peptides available when stratifying by peptide similarity, differences in experimental methods used to obtain these TCR-pMHC binding pairs, and hidden correlations due to non-random selection of targets in the experiments which generated the available data. Indeed, similarly to the MHC-I binding data, public TCR-pMHC binding data is dominated by common alleles such as A*02:01 and peptides of clinical interest such as melanoma and cancer-testis antigens. Hence, while we can confidently show lack of generalization to unseen peptides, it is difficult to reliably conclude at what edit distance and associated TCR counts these models could generalize because of the lack of clearly observed trends.

This work evaluated only supervised and generative ML models which are trained by learning patterns of amino acid sequences directly from the available data of TCR-pMHC binding examples, an approach advanced across numerous publications over the last few years with some representative methods benchmarked in our work[11,12,14]. Although the alternative approaches were outside of the scope of our study, we expect that the path to building a generalizable TCR specificity prediction model against unseen epitopes will lie in building a foundational model trained to predict general protein interactions on a much larger, already available corpus of data and possibly fine-tuning it on TCR-pMHC interaction data. In particular, models which aim to predict structures and/or physical ensembles of TCR-pMHC complexes and use estimated structural quality metrics or thermodynamic averages to score the binding strength look very promising[25], although their current performance in predicting TCR specificity is likely limited by the insufficient accuracy of structure prediction algorithms on immune receptor complexes.

Overall, it is clear from this work that despite the application of increasingly complex architectures to the TCR-pMHC specificity problem, insufficient epitope data prevents the production of a useful sequence-based model via both supervised and unsupervised learning with acceptable performance on unseen epitopes. Existing datasets cover less than 1% of the lowest estimate produced here for required data, without considering the clustering of this current data around disease-linked targets. While there will be a large cost associated with the production of this essential wet-lab data, it is becoming comparable to the computational cost spent on training complex machine learning architectures on this problem. Overall, we suggest that the main efforts in this space should be directed toward experimental generation of sufficiently large and unbiased training data to improve coverage of the TCR-pMHC binding space.

Methods

Model generalization capabilities

TCR-pMHC binding data

Data collection and curation. In our study, we utilize the dataset compiled by TULIP authors[11], which consists of binding pairs of T cell receptors (TCRs) and peptide-major histocompatibility complex class-I (pMHC-I) molecules from publicly accessible repositories, including VDJdb[16], McPAS-TCR[17], and IEDB[15]. Data acquisition was completed as of 2024-02-19. We ensured the dataset does not contain any redundant samples or non-standard amino acids. Given all models are subsequently trained with CDR3β and peptide sequences, we keep only CDR3β information from the TCR.

An initial observation revealed a significantly greater diversity in CDR3β sequences compared to epitope sequences with 125,901 and 1,558 unique sequences respectively.

Dataset split. Our objective was to evaluate the performance of our model on unseen peptides for potential application in a personalized TCR therapy setting. To achieve this, we initially divided the positive CDR3β -peptide pairs into training and testing sets. Our approach follows that of TULIP[11], consisting of allocating pairs to the training set where the peptide has more than 20 samples, while the testing set comprises pairs with fewer than 20 samples. This division was motivated by two factors: firstly, to mitigate the influence of over-represented peptides during evaluation; and secondly, to ensure a sufficient number of samples per peptide in the training set to ensure meaningful learning by the model. To further eliminate potential label leaks through a shared TCR sequence, we filtered out from the testing set all samples of TCR CDR3β if they were found in the training set. This approach goes beyond similar filtering reported in the TULIP publication where it was applied only to TCRs paired with both a training peptide and a testing peptide when those peptides were closely related with an edit distance of 1.

Negative sampling. Negative (non-binder) examples of TCR-pMHC reactivities are essential for evaluating predictive performance of all ML models. Such examples are additionally required for training supervised ML models, although not generative models which would require only the positive examples to learn the distribution of binders. Ideal data would include the experimentally confirmed non-binder TCR-pMHC pairs sampled from the same TCR and pMHC distributions as the positive examples. Such experimental negative examples are very rare in publicly available data. Some publications on TCR-pMHC specificity modeling use internally generated undisclosed datasets[19]. Most publications impute the negative examples by randomly pairing TCRs and pMHCs from the positive examples[10,12,13]. This approach relies on the assumption that randomly sampled pairs are highly unlikely to exhibit binding due to the theoretical diversity of TCRs and antigens in addition to the high specificity of TCR-pMHC recognition. In our approach, for each positive pair, we perform random sampling to select a CDR3β sequence that does not belong to the set of CDR3β sequences known to bind to the epitope

associated with that specific positive pair. We maintain a 1:1 positive/negative ratio between binding and non-binding CDR3β-peptide pairs.

While this technique is commonly employed for CDR3β-peptide specificity prediction models, its implementation can vary, potentially introducing biases and inflating performance metrics. To avoid introducing any form of bias and ensuring the consistency of the data, we sampled CDR3β sequences from the training and testing set independently. This deliberate approach prevents the introduction of any data leakage from training into testing set. Alternative methodologies exist in the literature; in studies like ATM-TCR[12], negatives are generated prior to data splitting, which can introduce biases. Another approach involves pairing negatives using a separate dataset of healthy donors, as done in PiTE[13]. This method also carries its own bias, as the model may inadvertently learn from which dataset the CDR3β originates, rather than solely focusing on discerning the CDR3β pairing status, potentially leading to overly optimistic results. Further discussion on the biases introduced by the alternative dataset construction strategies in specific publications can be found in TChard[18] and TULIP[11] publications. Similarly, recent work such as TCR-H[14] evaluates models on a dataset sourced from the TChard benchmark[18] using assay-derived negative examples (referred to as "NA"). As noted in the TChard publication[18], this setup results in completely disjoint positive and negative TCR repertoires, enabling models to achieve high performance by relying on the identity of the TCR sequence alone, without considering its pairing with a specific peptide. This leads to an inflated estimate of generalization performance and does not reflect the actual challenge of TCR-pMHC specificity prediction.

Fivefold cross-validation experiment. To rule out the possibility that our results on a modified TULIP split were split-specific, we additionally conducted a separate 5-fold cross-validation experiment. Positive examples from the entire TULIP dataset were split randomly into 5 folds with equal counts of unique and non-overlapping peptides. Then, in a loop over the folds, each fold was treated as a testing split and the remaining 4 folds combined - as a training split. The same filtering of the shared TCRs and imputation of the negative pairs was applied to each testing / training pair of splits, followed by training the models on the training splits and evaluation of the combined performance on the testing splits as described below.

Predictive models

In our benchmark, we evaluate five models, prioritizing diverse architectural approaches over exhaustive benchmarking. This selection process allows for a comprehensive exploration of available architectures. In particular, we retained ATM-TCR[12], TCR-H[14] and TULIP[11], as well as two standard machine learning baselines (XGBoost and Random Forest).

ATM-TCR. The ATM-TCR model[12] predicts binding affinity between TCR and epitope sequences using multi-head self-attention mechanisms to capture contextual information within sequences and understanding of amino acid interactions. Each residue is represented as a one-hot vector and processed through embedding layers. The encoded sequences then pass through corresponding encoders with multi-head self-attention, and the model's decoder computes binding affinity by combining encoded representations through linear layers and activation functions. Our preliminary training revealed that the model's predicted probabilities exhibited poor calibration, predominantly clustering around 0.99 for unseen peptides. Consequently, following Mukhoti et al.[26] we opted to retrain the model employing focal loss, resulting in significantly improved calibration of outcomes. We note that our improvement in the score calibration has not resulted in any deterioration of the predictive performance metrics in our comparative benchmarking against the original implementation on the same dataset.

Random forest. A Random Forest is an ensemble machine learning method that combines multiple decision trees to make predictions[27]. It works by constructing a multitude of decision trees during training and then averaging or voting on their outputs during prediction. Each tree in the forest is trained on a random subset of the data, making the model robust and capable of handling complex datasets. In this study, we employ BLOSUM50 embeddings[28] as input features, representing each CDR3β and peptide amino acid. They capture similarity, conservation, functional characteristics, evolutionary relationships, and sequence patterns of amino acids. A similar approach involving a Random Forest model was previously employed in a TCR-pMHC classification task in TCREX[29], wherein a distinct TCR-pMHC classifier is trained for each peptide.

TCR-H. TCR-H[14] is a Support Vector Machine classifier that leverages a rich set of sequence-derived physicochemical descriptors calculated from both CDR3β and epitope sequences. These include amino acid composition (e.g., counts of hydrophobic, acidic, and basic residues), global sequence properties (such as molecular weight, isoelectric point, and net charge), and pairwise/dipeptide features that capture residue co-occurrence patterns. To reduce redundancy and improve interpretability, features exhibiting high inter-correlation (Pearson > 0.8) were pruned, resulting in a compact yet informative subset. Post?hoc SHAP analysis revealed that attributes like average hydrophobicity, charge distribution, and specific dipeptide frequencies contributed most to binding prediction, enhancing both model explainability and biological insight.

TULIP. TULIP[11] is another transformer-based model using three encoders and decoders to focus on key interacting regions within epitopes, MHCs, and TCR chains. Contrary to most TCR specificity models, TULIP is trained in an unsupervised way as a generative model, as it does not explicitly predict a binary binding variable but is trained to maximize the likelihood of observing a given component of a TCR-peptide-MHC tuple from the positive examples (e.g. the peptide) conditioned on observing the other two components (e.g. TCR and MHC allele). This approach circumvents the need to create artificial non-binding pairs for training but not for evaluation.

XGBoost. XGBoost is a gradient boosting algorithm used for supervised learning tasks such as classification and regression[30]. It builds an ensemble of decision trees sequentially, with each tree focusing on the errors of the previous one. XGBoost optimizes performance by employing techniques like gradient boosting and regularization. XGBoost has become a popular choice in various machine learning competitions and real-world

applications due to its significant predictive performance and flexibility. Our XGBoost model employs the same input features as the Random Forest model.

Training and hyper-parameters. We train ATM-TCR, TCR-H and TULIP using the hyper-parameters and training settings provided in their official implementations[11,12,14]. For the Random Forest we use 40 estimators without extensive fine-tuning. Similarly, for XGBoost models, we use 100 estimators and a learning rate of 0.1.

Predictive metrics and their confidence intervals

We computed the Receiver Operating Characteristic (ROC) curve and its corresponding Area Under the Curve (AUC) value for each model providing an overall assessment of their performance on unseen peptides across folds. To refine our analysis, we stratified the results based on the similarity of peptides to the training set. More precisely, we calculated the edit distance between each test peptide and its nearest counterpart in the training set. We then report the ROC-AUC stratified per minimum dissimilarity (edit distance) between each testing peptide and peptides in the training dataset. We also report the Precision-Recall Area Under the Curve (PR-AUC), which measures the area under the precision-recall curve to evaluate the performance of binary classification models.

To quantify uncertainty while respecting dependence among test examples that share a peptide, we estimated confidence intervals (CIs) using a balanced peptide–cluster bootstrap. In each of 5000 replicates, we (i) sampled each peptide once (without replacement), (ii) resampled its TCR pairs (both positive and negative pairs as one pool) with replacement matching the original count of TCRs for that peptide, and (iii) pooled resampled examples from all peptides and computed ROC–AUC (binned by edit distance and count of training set TCRs as described above). To limit dominance by peptides with many test TCRs, we used capped per–peptide weights in the AUC: for peptide p with n_p test pairs, each example received

$$w_{ip} = \begin{cases} 1, & n_p \le 20, \\ 20/n_p, & n_p > 20, \end{cases} \quad \text{so that} \quad \sum_{i \in p} w_{ip} = \min(n_p, 20).$$

In the case of the 5-fold cross-validation experiment, the resampling procedure described above was used to generate 5000 ROC-AUC values per test dataset in each of the 5 folds. Then, 5000 new replicates were generated by randomly sampling with replacement a single ROC-AUC value from each of the folds, resulting in 5000 tuples of 5 values each. These 5 values were averaged using as weight the count of resampled examples per fold which were used to compute each value. Percentile intervals were taken over the final 5000 bootstrapped ROC-AUC values to compute 90% CIs. The motivation of using this nested bootstrap approach is that the alternative of pooling the resampled examples across folds and computing ROC-AUC would have resulted in comparing scores from models trained on different, although overlapping, subsets of data. Fold-specific scores can be on different scales and calibrations. ROC–AUC is invariant to a global monotone transform, but not to piecewise transforms that differ by fold, so cross-fold pooling can distort the cross-peptide ranking and bias the estimate.

Our within–peptide resampling combined with a weighted ROC–AUC is motivated by a positive association between the number of test TCRs per peptide and apparent performance. Before applying resampling/weighting, we computed a diagnostic 2D histogram for each model with the *x*–axis as edit distance to the nearest training peptide and the *y*–axis as the *test* count of positive TCRs per peptide (note that this differs from the aggregation used in the main results, where counts come from the *training* pairs). In an evaluation where test examples are exchangeable and independent, one would not expect a *systematic* increase in ROC–AUC with the test TCR count per peptide because prediction is performed for each test example independently of other test examples. Instead, for edit-distance bins where the overall across-peptide ROC–AUC exceeded 0.5, peptides with > 10 test TCRs showed higher ROC–AUC and higher lower confidence bounds than peptides with [0, 10] test TCRs, while contributing disproportionately more examples despite fewer peptides (Supplementary Fig. 5; ATM-TCR and TCR-H shown as representative examples). Treating peptides as the primary sampling units and capping their influence at high TCR counts mitigates this size-driven inflation of pooled performance while preserving cross-peptide classification errors. In our balanced design (negatives imputed 1:1 per peptide), 10 positive TCRs correspond to approximately 20 total TCR–peptide pairs per peptide, which is the threshold at which the per-peptide cap is applied.

As an additional check, we also implemented the *pigeonhole* (row∧column) bootstrap for crossed peptide× TCR effects[31]. On our extremely sparse, highly unbalanced layout (a small number of peptides account for roughly half of positives), this two–way resampling produced markedly conservative intervals—so wide that lower bounds overlapped the 0.5 random baseline in all edit–distance bins, consistent with the conservative behavior of pigeonhole bootstrap for such data layouts as pointed out in the original publication. Given that the dominant dependence in our data is peptide–level, we report ROC-AUC estimation with capped weights and the peptide–cluster bootstrap CIs for the uncertainty estimate.

The reader can experiment with different estimation and bootstrapping approaches by varying parameters at the top of the `bootstrap.ipynb` Jupyter notebook file included in our companion GitHub repository along with the necessary input data.

Non-self ligandome estimations

To estimate the lower bound for each allele's set of potential class-I human tumor neoantigens, we aimed to generate all point mutations of normal proteome peptides which are presented by the MHC alleles dominant in the population. Such SNV-derived mutations comprise the majority of targets which can be identified with currently available approaches to personalized tumor epitope discovery. We then added known viral and tumor origin epitopes presented by these alleles. A generalizable TCR-pMHC specificity prediction model would

have to work on peptide targets from this set, which we call a "non-self ligandome". The non-self ligandome estimations were performed in an allele-wise manner via two distinct methods. Expanded experimental ligandomes for each selected allele were based on existing experimental binding data and the local experimental mutanome (See Section "Experimental ligandome estimations"). Predicted ligandomes were calculated based on exhaustive synthetic point mutations introduced into the human proteome and augmented with verified tumor neoantigens with no associated MHC binding data (See Section "Predicted ligandome estimations"). For each allele, we further calculated a set of unique peptides from both predicted ligandomes and expanded experimental ligandomes termed the union ligandome (See Section "Predicted ligandome estimations"). A full overview of the workflow for the construction of the various ligandomes produced in this study is available in the supplementary information (Supplementary Fig. 9).

MHC-I allele selection

As ligandome estimates were performed allele-wise, we aimed to select a representative set of MHC-I alleles to ensure sufficient coverage of loci A, B and C in the world population. To obtain this representative set, HLA allele frequencies for 21 ethnic populations were obtained from Gragert et al.[32]. To better facilitate general population coverage estimates, ethnic groups were further subset to six broad populations: Caucasian ('CAU'), African American ('AFA'), Hispanic ('HIS'), Asian & Pacific Islands ('API'), Northern Chinese ('NCHI'), and Asian Indian ('AINDI'). For each included population, the five most common alleles at each locus A, B and C were selected as members of the representative set. Coverage was estimated as a fraction of the individuals within a given ethnic group with at least one of the alleles found in their diploid genotype. Duplicate alleles between categories were removed, resulting in 34 representative MHC-I alleles present in between 56% and 99% of individuals in a given population (Supplementary Fig. 10). We note that there is a possible alternative definition of coverage requiring all alleles in the MHC-I genotype of an individual to be contained in the chosen set of alleles. That population coverage is much lower from the same set of alleles and not discussed further in this work.

Experimental ligandome estimations

To obtain all available experimentally verified binding peptides for a given MHC allele, we queried multiple publicly available immunopeptidomic databases as of 2023-08-09. All experimental binders from the HLA Ligand Atlas were retrieved from the 2020.12 release[33]. The Immune Epitope Database (IEDB) was queried for all human pMHC binding assays involving healthy, tumor or viral origin peptides[15]. Peptide-MHC binding data was additionally retrieved from the NetMHCPan 4.1 training data[8]. This data consists of pMHC interaction data derived from both binding affinity assays and mass spectrometry experiments. While this dataset in its entirety contains over 13M data points, many entries are from multi-allelic assays and lack annotation with a specific MHC I allele. Hence, we kept only monoallelic data for our analysis. Due to the scarcity of experimental pMHC binding data for viral origin peptides in IEDB, we further supplemented our experimentally verified pMHC assays with viral origin peptides from VDJdb[16]. Lastly, we include all peptides labelled as binders from multiple in-house mass spectrometry assays [Internal proteomics dataset]. From these collected data, all unique 9-mer peptides were extracted; we subsequently refer to this unique subset as the experimental ligandome throughout (Supplementary Table 1).

To verify that peptides labelled as normal self were indeed of human origin, we performed a BLAST[34] search of all peptides not explicitly labelled as tumor or viral in origin against the UniProt human reference proteome[35]. All peptides labelled as normal self without probable BLAST matches were excluded from all subsequent analysis. For this task, we defined a probable match as a sequence within one edit distance of any 9-mer present in the human reference proteome.

As the subject of this article is the estimation of the TCR-reactive mutanome, we then synthesized a 'local experimental mutanome'. To produce this local experimental mutanome, we enumerated all possible 9-mer peptides within edit distance 1 of any normal self origin (or tumor origin) peptide in the experimental ligandome. In this way, we created a set of peptides which are both very close to experimentally verified binders for a given allele, and most likely to arise via single-residue mutations to healthy or existing tumor tissues. The latter was treated this way under an assumption that the vast majority of MHC ligands identified or predicted from the tumor tissue are still normal (non-mutated) peptides. For a given allele, the local experimental mutanome was passed through NetMHCPan 4.1 and all peptides predicted as strong binders kept. Lastly, all predicted binders from the local mutanome or peptides from the experimental ligandome with exact 9-mer matches to the human reference proteome were removed. All remaining peptides from the experimental mutanome and predicted binders from the local experimental mutanome were combined into a final 'expanded experimental ligandome'. This subset contains only non-normal-self peptides and viral peptides, and aims to represent all putative TCR-reactive peptides binding a given MHC allele with some experimental evidence. We treat this expanded experimental ligandome as a lower bound estimate for the number of non-normal-self peptides a given MHC allele may bind.

Predicted ligandome estimations

For each allele in our representative set, we aimed to estimate the upper bound of all non-normal-self point mutation peptides they may bind. To achieve this, we created an 'exhaustive human 9-mer mutanome' from the UniProt reference human proteome[35]. All unique 9-mer peptides (n= 10.2M) were identified using a 9-mer sliding window approach across all 20,408 genes in the reference proteome. All unique single residue mutations were enumerated from these 9-mers, excluding mutations which resulted in the creation of another existing normal self peptide. Hence, the exhaustive human 9-mer mutanome contains all non-normal-self 9-mer peptides which may arise from a single residue mutation in any position in the human proteome (n= 1.73B).

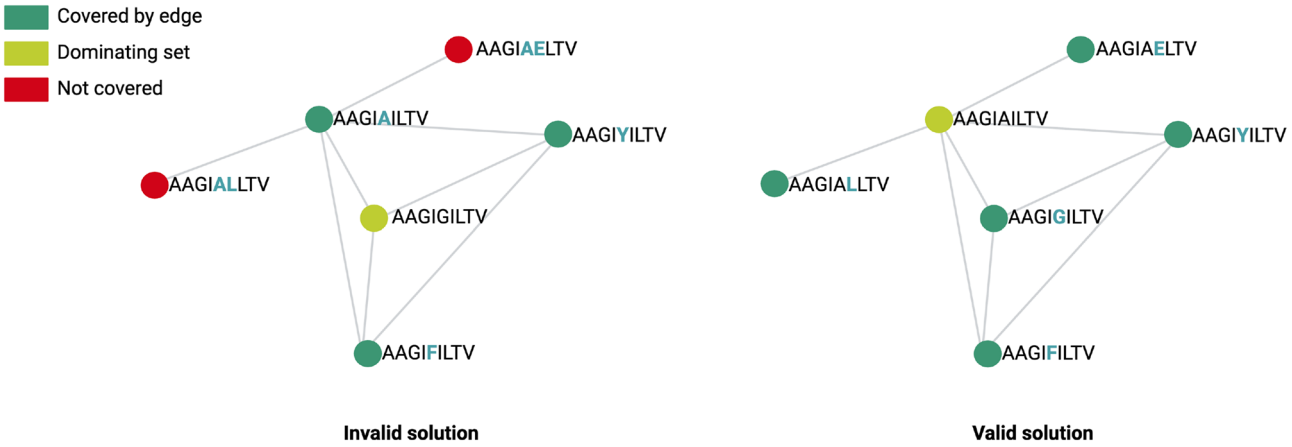


Fig. 5. Example of valid and invalid dominating sets for a peptide graph with edges drawn at edit distance one. The left hand graph is an invalid solution at edit distance one (AAGIALLTV and AAGIAELTV are not covered by AAGIGILTV at edit distance one), but would be valid at edit distance two. The right hand graph shows a valid optimal solution (all peptides are covered by AAGIAILTV at edit distance one); this single peptide (AAGIAILTV) would therefore be sufficient epitope training data for a model to generalize to the entire graph at edit distance one.

Additionally, we obtained all entries from TSNAdb 2.0, a database of experimentally verified human tumor neoantigens from tumor sample sequence data[36]. As this database contained peptides of varying length, we excluded all peptides with length <9. All peptides of length 9 are included unmodified, whilst peptides of length >9 were split into all possible linear 9-mer binding cores using a sliding window approach. For example, an 11-mer peptide would be present as three unique 9-mers consisting of residues 0:9, residues 1:10, and residues 2:11. All TSNAdb 2.0 peptides with exact matches to the UniProt human reference proteome were excluded, ensuring TSNAdb data represented known non-normal-self human tumor neoantigens.

All remaining peptides from both the exhaustive human 9-mer proteome and TSNAdb 2.0 were passed through NetMHCPan 4.1 and strong binders kept. Hence, these binders aim to represent all possible mutant 9-mer peptides a given MHC allele may bind; we refer to these predicted binders throughout as an allele's 'predicted ligandome'. For a given MHC allele, these predicted binders were then combined with all peptides from that allele's expanded experimental ligandome. The unique subset of this combination represents an exhaustive upper bound for the allele's ligandome, and is subsequently referred to as the 'union ligandome' throughout.

Dominating set estimations

For any given set of peptides, we use the 'dominating set' as a coverage measure of the peptide space. In graph based approaches, the dominating set is defined as a subset of nodes (or vertices - V) such that every node in the graph is either a member of this dominating set, or adjacent (connected by an edge - E) to a node which is a member of this dominating set[37].

Let $G = (V, E) \in \mathcal{V} \times \mathcal{E}$ be an undirected graph. A set $D \subset V$ is a dominating set if for all $(u, v) \in V^2$:

$$\begin{cases} u \in D \\ \text{or} \\ (u, v) \in E \cap (D^c \times D) \end{cases} \tag{1}$$

The size of this set represents the training sample size required for generalization. Let us suppose that a TCR-pMHC binding model can generalize to peptides 1 edit distance from any training examples, but not further. Figure 5 shows how a valid dominating set consisting of a single peptide can be used as sufficient training data to train a model that can generalize to predict TCR specificity for any peptide from the entire graph. By constructing these peptide graphs and solving the minimum dominating sets for a given allele's ligandome, we therefore estimate the minimum number of peptides required to train a model that can generalize across the entire considered ligandome at a given edit distance.

To enable graph construction, all pairwise edit distances for considered peptides were determined heuristically using the SACHICA algorithm[23]. Peptide graphs for a given ligandome were constructed using the NetworkX Python package[24]. Edges between peptides were considered at edit distance thresholds 1, 2 and 3. As determining the true minimum dominating set is an NP-complete problem, dominating sets for constructed graphs were obtained heuristically using a modified version of the NetworkX package's *min_weighted_dominating_set* function. Calculated dominating sets were verified as true dominating sets using the NetworkX *is_dominating_set* function. To reduce the computational complexity of greedily solving the dominating set for a given graph, we first removed isolated nodes before dominating set calculations were undertaken. Here, isolated nodes were peptides where no other peptide in the considered ligandome was within the edit distance threshold; these peptides are manually added to final dominating sets after their calculation.

To determine overall pan-allelic dominating set sizes, we take the sum (not the union) of all union ligandome sizes across all 34 alleles; identical peptides present in the union ligandomes for two different MHC alleles are considered as separate data points due to the importance of TCR-MHC interactions in driving the specific TCR recognition of the epitopes.

Union ligandome subsampling

Despite the use of heuristics, determining valid dominating sets for union ligandomes of millions of peptides was computationally intractable. Therefore, to estimate union ligandome dominating set sizes, we randomly subsampled up to 50% of the union ligandome peptides from A*02:01 and solved the dominating sets at edit distance thresholds 1, 2 and 3 (Fig. 4a). From this subsampling, we fitted an asymptotic curve to both interpolate and extrapolate the dominating set size estimation as necessary for the other alleles. We assume parameters obtained for A*02:01 broadly apply to all considered alleles and hence estimate dominating set sizes for the remaining 34 alleles using this approach.

Data availability

We cannot directly distribute the public datasets used in this study due to licensing restrictions for the third-party redistribution of the original or derived data from some of the data providers. Therefore, our code repository https://github.com/instadeepai/tcr-dataset-size-estimation provides detailed instructions and scripts for obtain ing and transforming all necessary external datasets to reproduce our analysis. To help other researchers save computational resources, we share via Zenodo the 'exhaustive human 9-mer mutanome' component generated from the UniProt reference human proteome as described in section "Predicted ligandome estimations" of the Methods with associated NetMHCPan scores https://doi.org/10.5281/zenodo.15126598. The additional propri etary "Internal proteomics dataset" that we used to augment the input set for non-self ligandome computation is not publicly released. As a test of reproducibility, our code repository shows results of applying our non-self ligandome estimation code to just the publicly available datasets as referenced in the code. These results demonstrate comparable total dominating set sizes with the sizes reported in this manuscript for a more complete set of inputs augmented by our unreleased proprietary proteomics data.

Code availablity

The code supporting the conclusions of this study is available on GitHub at https://github.com/instadeepai/tcr-dataset-size-estimation. The repository contains the code, environment, training and inference scripts to train and evaluate the machine-learning based models, as well as the ligandome computation workflow. Notebooks are provided to generate the figures.

Received: 9 June 2025; Accepted: 28 October 2025
Published online: 27 November 2025

References

1. Murphy, K. & Weaver, C. *Janeway's Immunobiology* (Garland Science, New York, 2016).
2. Wilson, I. A. & Fremont, D. H. Structural analysis of MHC class I molecules with bound peptide antigens. *Semin. Immunol.* **5**, 75–80 (1993).
3. Krogsgaard, M. & Davis, M. M. How T cells "see" antigen. *Nat. Immunol.* **6**, 239–245 (2005).
4. Davis, M. M. & Bjorkman, P. J. T-cell antigen receptor genes and T-cell recognition. *Nature* **334**, 395–402 (1988).
5. Levy, P. L. & Gros, A. Fast track to personalized TCR T cell therapies. *Cancer Cell* **40**, 447–449 (2022).
6. Hudson, D., Fernandes, R. A., Basham, M., Ogg, G. & Koohy, H. Can we predict T cell specificity with digital biology and machine learning?. *Nat. Rev. Immunol.* **23**, 511–521 (2023).
7. Sahin, U. & Türeci, Ö. Personalized vaccines for cancer immunotherapy. *Science* **359**, 1355–1360 (2018).
8. Reynisson, B., Alvarez, B., Paul, S., Peters, B. & Nielsen, M. NetMHCPan-4.1 and NetMHCIIPan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Res.* **48**, W449–W454 (2020).
9. O'Donnell, T. J., Rubinsteyn, A. & Laserson, U. MHCflurry 2.0: Improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. *Cell Syst.* **11**, 42–48 (2020).
10. 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.* **11**, 1803 (2020).
11. Meynard-Piganeau, B., Feinauer, C., Weigt, M., Walczak, A. M. & Mora, T. TULIP: A transformer-based unsupervised language model for interacting peptides and T cell receptors that generalizes to unseen epitopes. *Proc. Natl. Acad. Sci.* **121**, e2316401121 (2024).
12. Cai, M., Bang, S., Zhang, P. & Lee, H. ATM-TCR: TCR-epitope binding affinity prediction using a multi-head self-attention model. *Front. Immunol.* **13**, 893247 (2022).
13. Zhang, P., Bang, S. & Lee, H. PiTE: TCR-epitope binding affinity prediction pipeline using transformer-based sequence encoder. *Biocomputing* 347–358 (2022).
14. Demerdash, O. N. et al. TCR-H: Explainable machine learning prediction of t-cell receptor epitope binding on unseen datasets. *Front. Immunol.* **15**, 1426173 (2024).
15. Vita, R. et al. The immune epitope database (IEDB): 2018 update. *Nucleic Acids Res.* **47**, D339–D343 (2019).
16. Shugay, M. et al. VDJdb: A curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Res.* **46**, D419–D427 (2018).
17. Tickotsky, N., Sagiv, T., Prilusky, J., Shifrut, E. & Friedman, N. McPAS-TCR: A manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics* **33**, 2924–2929 (2017).
18. Grazioli, F. et al. On TCR binding predictors failing to generalize to unseen peptides. *Front. Immunol.* **13**, 1014256 (2022).
19. Kwee, B. P. *et al.* STAPLER: Efficient learning of TCR-peptide specificity prediction from full-length TCR-peptide data. *BioRxiv* 2023–04 (2023).
20. Nielsen, M. et al. Lessons learned from the IMMREP23 TCR-epitope prediction challenge. *ImmunoInformatics* **16**, 100045 (2024).
21. Paul, S. et al. HLA class I alleles are associated with peptide-binding repertoires of different size, affinity, and immunogenicity. *J. Immunol.* **191**, 5831–5839 (2013).

22. Kaplan, J. *et al.* Scaling laws for neural language models (2020).
23. Uno, T. Multi-sorting algorithm for finding pairs of similar short substrings from large-scale string data. *Knowl. Inf. Syst.* **25**, 229–251 (2010).
24. Hagberg, A. A., Schult, D. A. & Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. In: *Proceedings of the 7th Python in Science Conference* 11 – 15 (2008).
25. Bradley, P. Structure-based prediction of T cell receptor: Peptide-MHC interactions. *Elife* **12**, e82813 (2023).
26. Mukhoti, J. et al. Calibrating deep neural networks using focal loss. *Adv. Neural. Inf. Process. Syst.* **33**, 15288–15299 (2020).
27. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
28. Mount, D. W. Using BLOSUM in sequence alignments. *CSH Protoc.* **6** (2008).
29. Gielis, S. et al. Detection of enriched T cell epitope specificity in full T cell receptor sequence repertoires. *Front. Immunol.* **10**, 2820 (2019).
30. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
31. Owen, A. B. The pigeonhole bootstrap. *Ann. Appl. Stat.* **1**, 368–411 (2007).
32. Gragert, L., Madbouly, A., Freeman, J. & Maiers, M. Six-locus high resolution HLA haplotype frequencies derived from mixed-resolution DNA typing for the entire US donor registry. *Hum. Immunol.* **74**, 1313–1320 (2013).
33. Marcu, A. et al. HLA ligand atlas: A benign reference of HLA-presented peptides to improve T-cell-based cancer immunotherapy. *J. Immunother. Cancer* **9**, e002071 (2021).
34. Camacho, C. et al. BLAST+: Architecture and applications. *BMC Bioinform.* **10**, 1–9 (2009).
35. The UniProt Consortium. UniProt: The universal protein knowledgebase. *Nucleic Acids Res.* **45**, D158–D169 (2017).
36. Wu, J. *et al.* TSNAdb v2.0: The updated version of tumor-specific neoantigen database. *Genomics Proteomics Bioinform.* S1672–0229(22)00128–0 (2022).
37. Casado, A., Bermudo, S., López-Sánchez, A. D. & Sánchez-Oro, J. An iterated greedy algorithm for finding the minimum dominating set in graphs. *Math. Comput. Simul.* **207**, 41–58 (2023).
38. Sušac, L. *et al.* Structure of a fully assembled tumor-specific T cell receptor ligated by pMHC. *Cell* **185**, 3201–3213 (2022).

Acknowledgements

We thank Victor Mikhaylov (BioNTech SE) for correctly identifying the dominating set problem. We thank Michael Rooney (BioNTech SE) for discussions and advice about the ligandome datasets. We thank Florian Bock (BioNTech SE) for providing editorial assistance.

Author contributions

A. D., M.M., B.D., N.G, S.C.G-M, I.K., L.C. and A.T. contributed to the methodology, conceptualization, writing the code, and running the experiments. I.R. contributed the high-performance computational workflow engineering. A.D., M.M., B.D., L.C., and A.T. wrote the manuscript. B.D, L.C. and A.T. were responsible for the project administration and technical supervision. C.P., M.L., K.B., N.L.C. and U.S. contributed to the project conceptualization and administration. All authors reviewed the manuscript. A.D. and M.M. contributed equally to this work.

Funding

This research was funded by BioNTech SE

Declarations

Competing interests

A. Delaunay, M. McGibbon, N. Gorbushin, S. Chaves Garcà-Mascaraque, I. Rayment, K. Beguir, L. Copoiu and N. Lopez Carranza are employees of InstaDeep, a subsidiary of the BioNTech group. B. Djermani was an employee of InstaDeep during his work on this study. I. Kizhvatov, C. Petit, M. Lang are employees of BioNTech. U. Sahin is the CEO of BioNTech. A. Tovchigrechko was an employee of BioNTech during his work on this study.This study was funded by BioNTech.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-26454-7.

Correspondence and requests for materials should be addressed to A.D., M.M. or A.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025