SwiftTCR: Efficient computational docking protocol of TCRpMHC-I complexes using restricted rotation matrices

Farzaneh M. Parizi[1†], Yannick J. M. Aarts [1†], Sanna Eerden[1], Gayatri Ramakrishnan[1], and Li C Xue[1*]
[1]Medical BioSciences, Radboud University Medical Center, Nijmegen, Netherlands
[†]These authors equally contributed to the work

Abstract

The T cell's ability to discern self and non-self depends on its T cell receptor (TCR), which recognizes peptides presented by MHC molecules. Understanding this TCR-peptide-MHC (TCRpMHC) interaction is important for cancer immunotherapy design, tissue transplantation, pathogen identification, and autoimmune disease treatments. Understanding the intricacies of TCR recognition, encapsulated in TCRpMHC structures, remains challenging due to the immense diversity of TCRs (>10^8/individual), rendering experimental determination and general-purpose computational docking impractical. Addressing this gap, we've developed a rapid integrative modeling protocol leveraging unique docking patterns in TCRpMHC complexes. Built upon PIPER, our pipeline significantly cuts down FFT rotation sets, exploiting the consistent polarized docking angle of TCRs at pMHC. Additionally, our ultra-fast structure superimposition tool, GradPose, accelerates clustering. It models a case in minutes, showcasing a 25 times speedup compared to ClusPro/PIPER. On a benchmark set of 38 TCRpMHC-I complexes, our protocol outperforms the state-of-the-art docking tools in model quality. This protocol can potentially provide structural information to TCR repertoires targeting specific peptides. Its computational efficiency can also enrich existing pMHC-specific single-cell sequencing TCR data, facilitating the development of structure-based deep learning (DL) algorithms. These insights are essential for understanding T cell recognition and specificity, advancing the development of therapeutic interventions.

Keywords: T cell receptor (TCR), peptide-MHC (pMHC), TCR binding angle, restricted docking, integrative modelling, computational immunology

1. Introduction

T cells discern between self and non-self antigens through their T cell receptors' (TCRs) recognition of peptide antigens presented by major histocompatibility complexes (MHCs). Endogenous peptides expressed by MHC class I (MHC-I) can activate CD8+ cytotoxic T cells, and exogenous antigens expressed by MHC class II (MHC-II) can activate CD4+ T cells. Understanding the interaction between TCR-peptide-MHC (TCRpMHC) has broad applications in immunology, autoimmune diseases[1], infectious disease treatment [2] and cancer immunotherapy [3].

The fundamental question of how T cells recognize antigens within a variety of peptide antigens with high specificity has been the subject of considerable research effort [4], [5], [6]. This inquiry has revealed an evolving understanding that T cell activation relies on the simultaneous recognition of foreign peptide antigens and self MHC molecules, with a focus on reconciling selection and germline-encoded models [1]. Structural investigations into T cell surveillance have yielded valuable insights into this inquiry. Notably, all (with a few exceptions [7]) existing experimentally determined TCRpMHC structures demonstrate that TCRs maintain consistent polarity over pMHC (**Fig. 1**), which influences TCR specificity and binding affinity [8], and is also shown to impact T cell activation and signaling in mice and humans. For instance, TCRs with reversed polarity in a viral epitope-specific T cell repertoire were unable to support TCR signaling due to Lck mislocalization in the presence of the CD8+ coreceptor [8]. Structural investigation has also contributed to tumor-associated or tumor-specific peptide target identification [9] and the optimization of cancer immunotherapies [10] and also holds the promise to reduce the toxicity of TCR therapies, which is caused by TCR cross-reactivity to self-peptides.

The TCR structure is a heterodimer composed of α and β chains (or γ and δ chains in γδ T cells). Each chain consists of a variable (V) domain and a constant (C) domain (**Fig. 1A**). The V domains participate in antigen recognition, whereas the C domains assist in the TCR's anchoring to the T cell's surface. Each variable domain of TCR contains three complementarity-determining region (CDR) loops (CDR1-3). The TCR heterodimer docks diagonally with the long axis of the MHC peptide-binding groove**.** CDR1-2 loops primarily interact with MHC, whereas CDR3 interacts with the peptide, contributing uniquely to TCR specificity (**Fig. 1B, C**). Structures provide atomistic details of the TCRpMHC interaction, as well as insights into interaction energies and physicochemical properties. With increasing advances in artificial intelligence (AI) approaches, structure-based TCR specificity predictions may hold the promise to decipher the long-elusive TCR recognition code. The integration of structural information into AI methods has been shown to enhance the development of generalizable predictive approaches for MHC-binding peptide predictions and drug design [11]. However, the available experimental TCRpMHC structures (~350 experimental structures in PDB [12]/ IMGT [13]) are not enough to train a powerful AI predictor, necessitating the development of computational modeling methods that encompass the immense diversity of TCR, MHC, and peptide sequences.

A few computational tools have been developed for 3D modeling of TCR-pMHC complexes, based on docking, AI, or homology modeling. Docking predicts complex structures from

individual protein structures by generating tens of thousands of docking conformations and selecting the ones with lowest energies (the hallmark of native conformations). Flexible docking allows conformational changes for potentially better model quality but requires much more computational time, while rigid body docking maintains fixed conformations for faster modeling. A recent benchmark study [14] assessed the performance of four docking platforms —ClusPro [15], ZDOCK [16], and HADDOCK [17]— in modeling TCRpMHC interactions. This study, utilizing an expanded benchmark set of 44 TCRpMHC docking cases, revealed that HADDOCK demonstrated superior performance. Further, ClusPro performed effectively in terms of model quality and computing time. ClusPro uses PIPER [18] for sampling and a closed source clustering method. AI-based tools like Immunebuilder [19] and TCRmodel2 [20] predict TCR structures, with TCRmodel2 additionally capable of modeling entire TCRpMHC complexes. Another AI-based docking pipeline [21] uses hybrid structural templates for sampling docking modes, similar to homology modeling. Homology modeling tools, exemplified by PANDORA [22] and APE-Gen [23], generate pMHC complex structures and LYRA [24] for modelling TCR structures. Additionally, ImmuneScape [26] is designed to model the entire TCRpMHC structure.

While AI-based tools have potential, their performance on novel TCRpMHCs remains uncertain. Nonetheless, standalone TCR and pMHC modeling tools show promise [19], [20], [22], [23]. However, for TCRpMHC modeling, by leveraging the wealth of information known for TCR binding to pMHC, it is possible to incorporate these into 3D modeling for more efficient and accurate modeling.

Exploiting the docking polarity of TCRs over pMHC (**Fig. 1C**), we developed swiftTCR an integrative framework building on top of PIPER software by only sampling on the feasible conformational space (**Fig. 2**). Tailored for TCRpMHC modeling, our framework reduces rotation sets of Fast Fourier Transform (FFT) from 202,491 to 4,329 rotations, aligning with the canonical binding orientation of TCR and pMHCs. This allows more sampling in the feasible conformation space with improved modelling results. With several other innovations (see Methods/Results), our protocol is 25 times faster than ClusPro, suiting for large-scale TCRpMHC modelling. Benchmarking on 38 experimentally resolved TCRpMHC-I complexes revealed that our method competes with state-of-the-art docking tools in terms of model quality and speed. With this acceleration, we envision our software delivering large-scale 3D models of TCRpMHC conformations closely aligned with experimental structure orientations, facilitating the design of more generalizable deep learning (DL) algorithms for TCR specificity predictors, unrestricted by species or allele types.

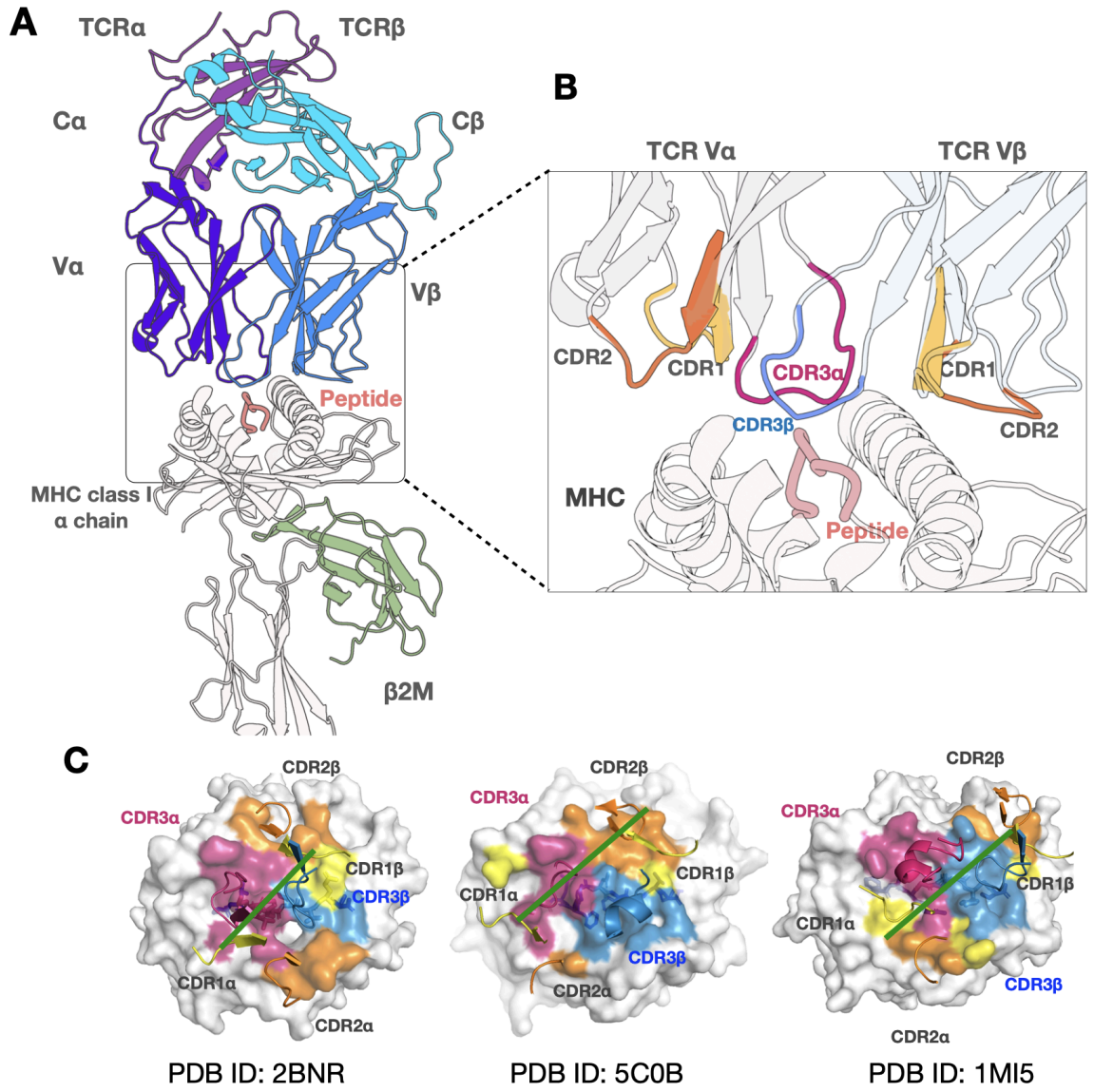


Figure 1. Structural visualization of TCR binding to pMHC. A) General TCRpMHC overview. The peptide (pink) binds inside the binding groove of the MHC molecule (white) and is presented on the cell surface to be visible to T cells (shades of blue and purple). The MHC molecule consists of two domains: alpha chain and beta-2 microglobulin (β2m) (green). The TCR, comprising alpha and beta chains with constant domains (Cα, purple; Cβ, blue) binding to T cell's surface and variable domains (Vα, dark purple; Vβ, cyan), interacting with pMHC. The experimental structure (PDB ID: 2BNR) is illustrated in the figure. **B**) The TCR has three CDR loops, CDR1 and CDR2 typically interact with the MHC's conserved alpha helices to stabilize the interaction, CDR3 plays a major role by recognizing the peptide antigen presented by the MHC as self or non-self. Binding interface of TCRpMHC complex, where TCR Vα (light purple) positions over MHC-I α2 helix (white) with its CDR3α (magenta) above the N-terminal half of the peptide (salmon). TCR Vβ positions over MHC-I α1 helix, with its CDR3β (blue) extending over the C-terminal portion of the peptide. Images made by Protein Imager [25] for an experimental structure (PDB ID: 2BNR). **C)** The colors indicate the docking footprints of the TCR CDR loops (CDR1: yellow, CDR2: orange, CDR3α: magenta, CDR3β: skyblue) on their cognate pMHC complex surface (white).The green line on the surface shows the docking angle, representing the polarity of TCR over pMHC for three different experimental structures,.

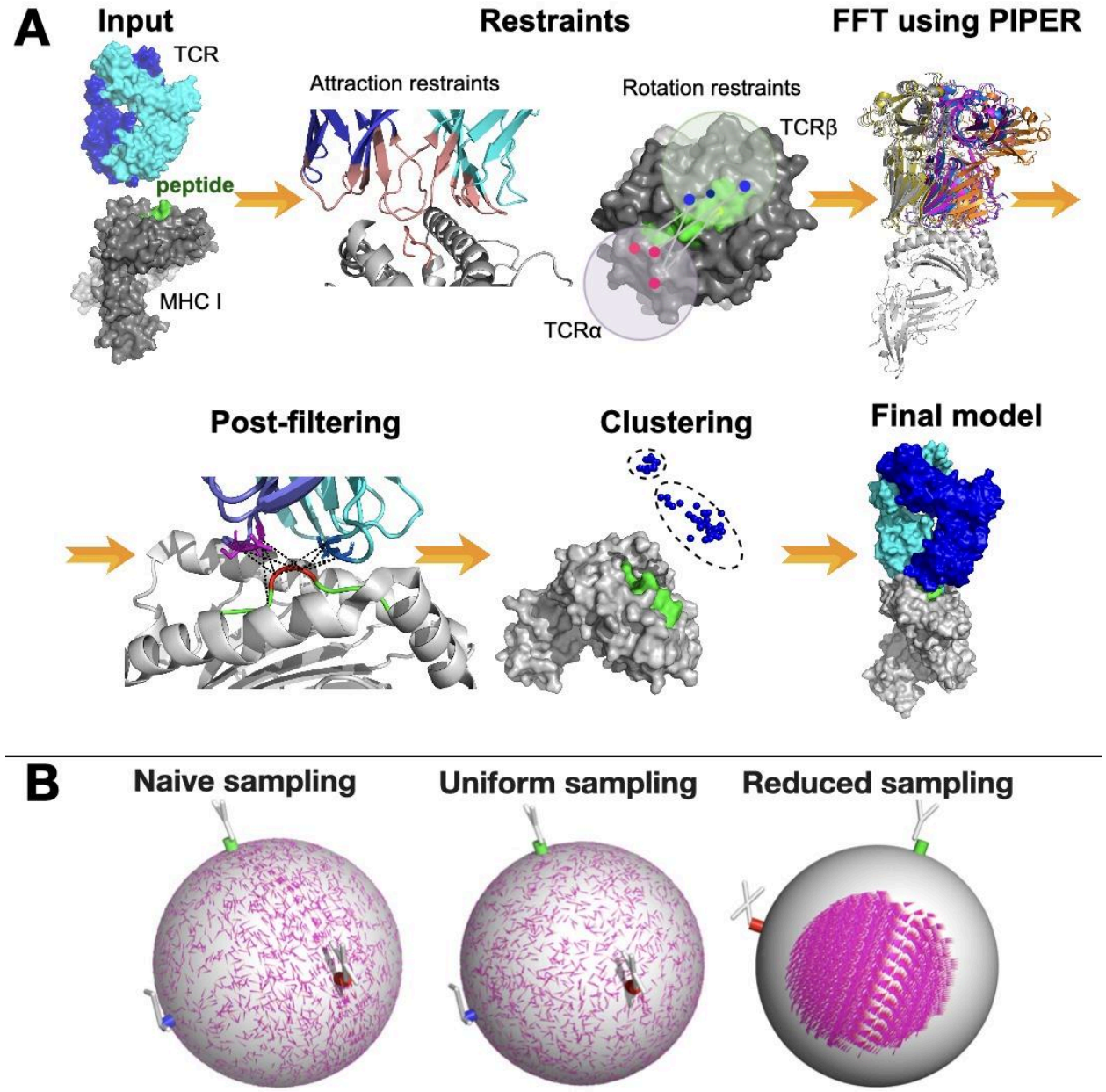


Figure 2. Schematic overview of swiftTCR docking pipeline for modelling TCRpMHC complexes. A) Given the conformation of TCR and pMHC structures, the pipeline generates TCRpMHC structures. Initial orientation of pMHC and TCR to a reference plane, followed by docking using PIPER with a customized rotation set for consistent TCR docking angles to pMHC and specified attraction to peptide and CDR3 residues. The generated models undergo post-filtering to ensure CDR3 loop binding to the peptide. Subsequently, clustering of the models is performed using our fast pairwise i-RMSD calculation, and the top-ranked models are selected to identify near-native complexes. **B)** Sampled Rotations on a 3D Globe. Naive sampling: is generated by taking even step sizes on the three Euler angles, leading to uneven distribution (more dense sampling along the X-Y plane) and oversight of critical regions. Uniform Rotation Sampling using quaternions: Rotation matrices are generated uniformly across a 3D sphere. Reduced sampling: Rotation matrices generated in B are filtered based on crossing angles and incident angles observed in the experimental TCRpMHC structures. For easy visualization, only a random sample of 5000 rotation matrices are shown in A and B. The reduced sampling contains in total 4,044 rotations, all of which are shown here.

2. Methods

We established a pipeline for TCRpMHC modeling using PIPER docking software. In our docking approach, we impose two key restrictions: 1) limiting the buried surface area to TCR CDR loops and peptide regions, and 2) constraining rotational space to ensure consistent polarized docking angles of TCRs at pMHC. This enhances both the speed and accuracy of the docking process. Following TCR and pMHC docking, we refine the generated models by filtering out those where CDR3 does not bind to exposed peptide residues (discarding unrealistic TCRpMHC orientations). In clustering, we leverage GradPose [27], our efficient structure superimposition tool, which accelerates the process by swiftly calculating pairwise distances. **Fig. 2A** depicts the general pipeline architecture used in this work.

2.1 Benchmark Dataset

To evaluate swiftTCR, we used a TCR benchmark set [14] which includes 44 cases of experimental TCRpMHC class I and II complexes, where for each of the bound TCRpMHC structures the corresponding unbound TCR and unbound pMHC structures are available. The structures were IMGT [13] numbered. In this proof-of-concept study, we have restricted our analyses to MHC class I alone, which amounts to a total of 38 cases of Human (36) and mouse (2) alleles.

2.2 SwiftTCR docking procedure

2.2.1 Docking prerequisite

To maintain consistency in the docking process, uniform chain names and residue numbering were used. The pMHC structures are named as chain A (MHC class I α1), B (MHC class I β2 Microglobulin), and C (peptide). Similarly, the TCR α and β chains are denoted as chains D and E, respectively. IMGT numbering is used in our experiments in this study. Different numberings are also accepted, but the users need to manually set the residue numbers for the post-filtering restraints (**Table 1**), the boundary of TCR variable domains in ”initial placement” (**Fig. 4B**) and the definition of CDR1-3 region in “definition of attractive residues” (see Table in **Fig. 4A**). It is expected that the input TCR has at least complete variable domains, as we use the variable domains to determine the initial docking positions (section “Orient the starting position of pMHC and TCR to reference plane”) and to calculate the crossing angle and incident angles for the reduced rotation matrices.

2.2.2 Defining crossing angle and incident angle

Taking into account the TCR docking polarity, our algorithm conducts sampling within the favorable binding angle ranges, for speed and better model quality. The binding angle is characterized by two parameters: the crossing angle and the incident angle [28] (**Fig. 3**).

The crossing angle measures the docking polarity or the twist of the TCR over the pMHC, which is the angle between the TCR docking vector (centroid of conserved disulfide bonds) and the vector of MHC binding groove (**Fig. 3B**). Specifically, we applied PCA (principal component analysis)

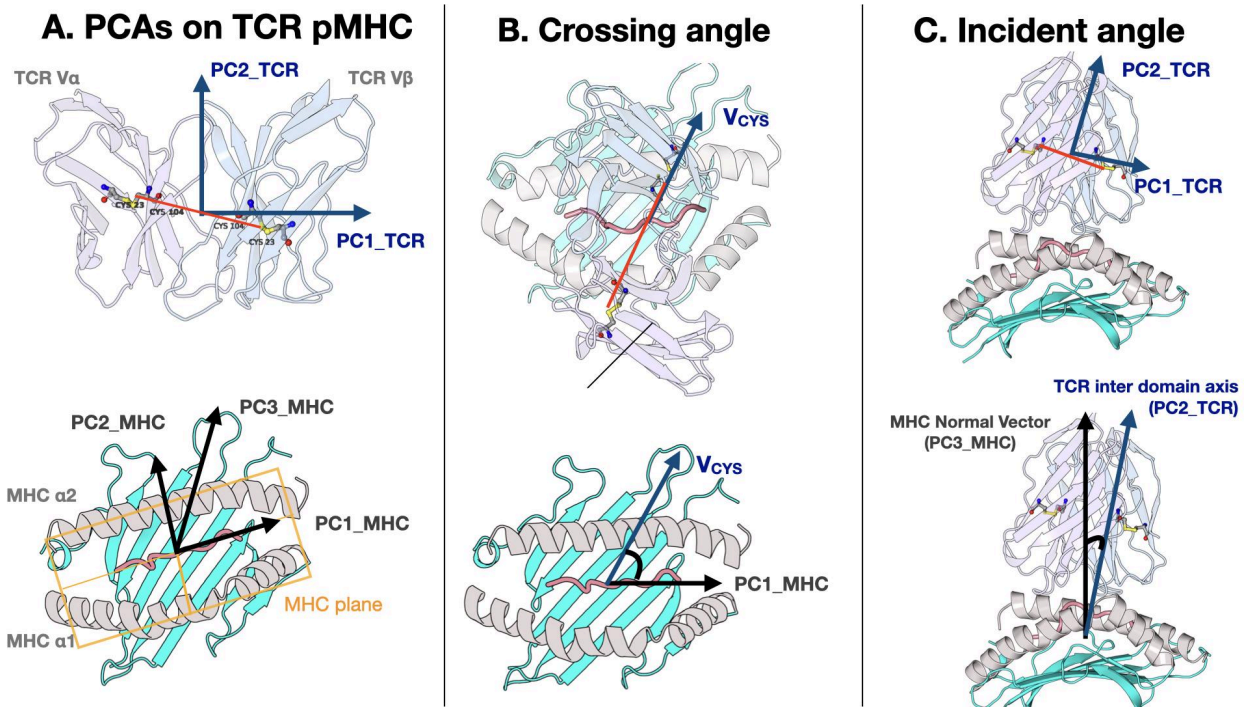


Figure 3. The definition of crossing angle and incident angle. A) PCA applied on TCR variable domain. To take into account the TCR docking polarity, PC1_TCR always points from TCR Vα to TCR Vβ. PC2_TCR: TCR interdomain axis of rotation also points from TCR variable domain to TCR constant domain. More defined, V_cys: the TCR docking vector which is the centroid of the four conserved cysteines of TCR variable domains (shown by the red lines crossing these residues) also pointing from TCR Vα to TCR Vβ. The MHC plane is defined by PCA analysis on the MHC helices. PC1_MHC: The long axis of the MHC binding groove, pointing from the N-ter of the peptide to the C-ter of the peptide. PC2_MHC: The width axis of the MHC binding groove, pointing to the MHC α1 helix. PC1_mhc and PC2_mhc define the MHC plane. PC3_MHC: The norm vector of the MHC plane, pointing to TCR. B) The definition of crossing angle, which measures the angle between the TCR docking vector (V_cys) and the long axis of the MHC binding groove (PC1_MHC). Thus, a reversed polarity will have a negative crossing angle. C) The definition of incident angle, which measures the tiltness of TCR over the MHC plan. Incident angle is the angle between the TCR interdomain axis of rotation and the MHC normal vector (PC3_MHC). The experimental PDB structure used is 2BNR.

on the MHC α1 and α2 helices (IMGT number: 50-86 and 140-176) to derive the first principal component (PC1_mhc, pointing from N-ter of the peptide to the C-ter) as the binding groove vector. (**Fig. 3A**) To calculate the TCR docking vector, we take four conserved cysteines of the TCR (IMGT number: 23, 104 for both chains), and use the vector pointing from the cysteine geometric center on chain D to the one on chain E as the TCR docking vector (V_cys). The angle between PC1_mhc and V_cys is the crossing angle (see details in Results).

The incident angle measures the tilt of the TCR variable domain over the pMHC, which is the angle between the normal vector of the MHC peptide binding groove plane (PC3_mhc, pointing outside the MHC and towards the TCR) and the TCR interdomain rotation axis (PC2_tcr, pointing from the TCR variable domains to the constant domains) (**Fig. 3C**). To get the TCR interdomain rotation axis, we conducted PCA on the variable domain of TCR, and PC2_tcr

which is the TCR interdomain rotation axis (**Fig. 3A**). We did not include TCR constant domains in this calculation, because the majority of single-cell sequencing data do not contain TCR constant domain [29], which are actively accumulating large amount of valuable TCR specificity data that could greatly benefit from structural enrichments from our SwiftTCR docking.

2.3. Definition of restraints

2.3.1 Defining reduced docking rotation matrices for favorable docking angles

We first generated 202,491 rotation matrices uniformly sampled (using a step size of 6 degrees) and then retained only those rotation matrices that can generate favorable docking angles for the TCRpMHC complex, i.e., $crossing\ angle \in [15\ ,\ 90]\ degrees$ and $incident\ angle \in [0, 35]\ degrees$, resulting in 4,044 valid rotation matrices (see **Fig. 5** for the angles for 177 x-ray TCRpMHC structures as of 06/06/2023).

Generating uniformly distributed rotations is an important topic in robotics for sampling-based motion planning [30], [31]. Similar to ClusPro, we use cluster sizes to rank our docked models assuming that the low-energy near-native conformations tend to form highly populated clusters. Therefore, uniform sampling is crucial to our docking method to sample uniformly distributed docking orientations in 3D space for over-sampling or under sampling in certain areas to avoid biases in our clustering step. According to Euler's rotation theorem, any orientation can be described by three successive rotations about the three axes (x, y, z-axis). For example, rotating around the x-axis for θ degree, then around y-axis for ϕ degree and then around z-axis for η degree. However, evenly sampling of three Euler angles (θ, ϕ, η) result in unevenly distributed rotations [31] (naive sampling, **Fig. 2B**). We generated uniformly distributed rotations using the algorithm proposed by Shoemake [31](**Algorithm 1, Fig. 2B**). Uniform sampling ensures uniform distribution across the entire space, enhancing precision, coverage and avoiding a bias towards specific orientations

Once we generate rotation matrices that uniformly cover the 3D sphere, we filter these rotation matrices based on crossing angles and incident angles. For this we take a reference structure 2bnr to calculate docking angles after each rotation. We keep the pMHC structure fixed and use the generated rotation matrices to rotate TCR and calculate the crossing angle and incident angle for the rotated TCRs. We used a range of [0, 35] degrees for the incident angle and a range of [15, 90] degrees for the crossing angle (these docking angle ranges are calculated on 177 experimental TCRpMHC structures in the PDB, **Fig. 5**). **Fig. 2B** shows the reduced rotation set. This results in 4,329 rotation matrices that only include relevant rotations based on prior knowledge of existing TCRpMHC complexes in the PDB. The final rotation matrices are written as a rotation file (.prm), which is used as input for PIPER. PIPER uses FFT to find the best translation for the provided rotations.

2.3.2. Defining attractive restraints for binding residues

We further enhance the docking accuracy by restricting the conformational search space with additional attraction terms targeting interface residues. These constraints guide the docking process towards selecting the appropriate pose. We added attraction to all peptide residues and all

CDR residues (**Fig. 4A**). The objective is to ensure that during sampling, conformations where the peptide is not in close proximity to TCRs are assigned a relatively unfavorable energy score.

Algorithm 1: uniformly distributed rotation matrices (quaternions)

---

def uniform_sampling (n_samples = 60):

$s = linspace(0, 1, \text{n_samples})$

$\sigma_1 = \sqrt{1 - s}$

$\sigma_2 = \sqrt{s}$

$t_1 = linspace(0, 1, \text{n_samples})$

$t_2 = linspace(0, 1, \text{n_samples})$

$\theta_1 = 2\pi * t_1$

$\theta_2 = 2\pi * t_2$

 # Define quaternion: [w, x, y, z]

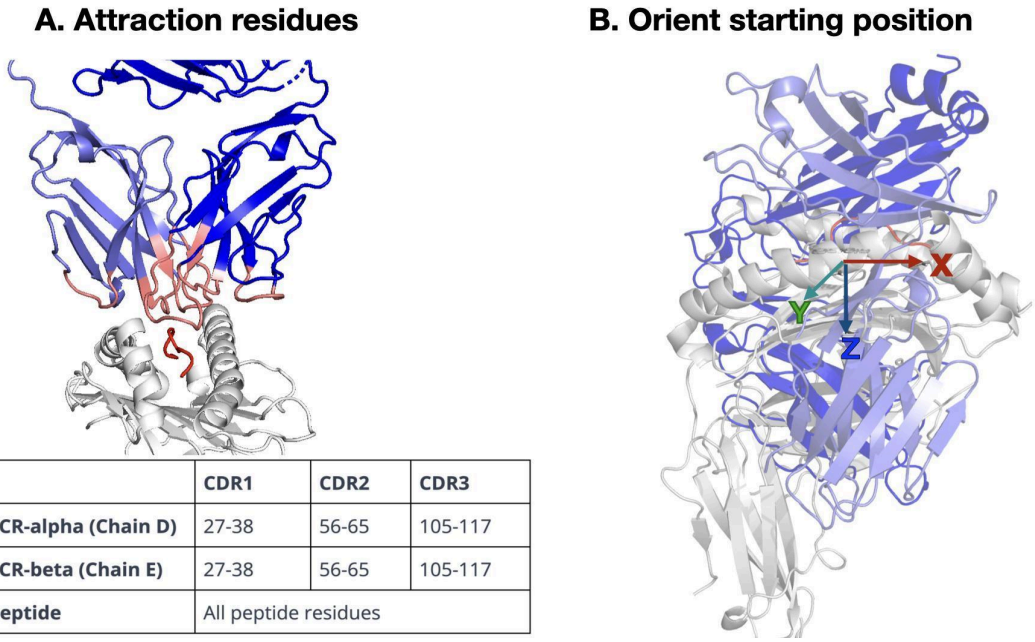
$w = cos(\theta_2) * \sigma_2$

$x = sin(\theta_1) * \sigma_1$

$y = cos(\theta_1) * \sigma_1$

$z = sing(\theta_2) * \sigma_2$

return (w,x,y,z)



	CDR1	CDR2	CDR3
TCR-alpha (Chain D)	27-38	56-65	105-117
TCR-beta (Chain E)	27-38	56-65	105-117
peptide	All peptide residues		

Figure 4. Preparatory insights to swiftTCR. A) Attractive Residues highlighted by pink marking key residues in the TCR (purple) binding interface to the peptide bound to MHC (white), guiding docking around this interface. Attractive Residues Table lists peptide residues and corresponding TCR CDR1-3 residues for binding, following IMGT numbering. B) Initial Docking positions Illustrates pMHC and TCR structures' initial orientation, centered on their principal components at the Cartesian coordinate origin, exemplified by PDB ID 1AO7.

We add these attraction residues by modifying the PDB files to include Attraction in column 55-60 of ATOM records of PDB files (0= no attraction, 1= attraction), which were then used as input for PIPER.

2.4 Docking

2.4.1 Preparation for electrostatic calculations

We used PDB2PQR [32] for the preparation of PDB files (`prepare.py` provided by PIPER). This tool converts PDB files of biomolecular structures into PQR files, essential for electrostatics calculations. Notably, PDB2PQR optimizes the hydrogen bonding network by strategically adding hydrogens to maximize bond strength.

2.4.2. Orient the starting position of pMHC and TCR to reference plane

Before docking, we need to first align the TCR and pMHC structures to be docked, to the same reference positions that are used to filter relevant rotation matrices in order to use the reduced rotation matrices properly. For this, we superimpose each query TCR and pMHC onto the reference structure 2bnr. Only TCR variable domain is used for the superimposition, as we do not want to assume the query TCR structure has constant domains. This approach ensures consistent positioning of TCR and pMHC complexes, facilitating accurate analysis of docking experiments' rotations. **Fig. 4B** illustrates one initial docking position.

Table 1. Post-filtering residues used. IMGT numbered.

	TCR Alpha chain	TCR Beta chain
TCR_{tip1}	**109**	**109**
TCR_{tip2}	**110**	**110**
$peptide_{middle\ residues}$	**4-5**	

Table 2. Post-filtering restraints (user-adjustable)

$Group\ 1$	$Group\ 2$
$1Å \leq distance(TCR\alpha_{tip1}, any\ peptide_{middle}) \leq 10Å$ **or** $1Å \leq distance(TCR\alpha_{tip2}, any\ peptide_{middle}) \leq 10Å$	$1Å \leq distance(TCR\beta_{tip1}, any\ peptide_{middle}) \leq 10Å$ **or** $1Å \leq distance(TCR\beta_{tip2}, any\ peptide_{middle}) \leq 10Å$

2.4.3 Docking with reduced rotation matrices and attractive restraints

Considering the TCR as the ligand and the pMHC as the receptor we run PIPER docking with the restricted rotational space and specified attraction to peptide and CDR3 residues. Following the reduction of our rotation set the PIPER sampling generated 4,329 ligand (TCR) orientations, exploring diverse molecular orientations. Generated docking positions are scored based on their energy levels using PIPER.

2.4.4 Post filtering

After generating docking models, we systematically remove those failing to comply with specified distance constraints.

We calculated statistics of the distance between the CDR3 tips and peptides from the benchmark set (**Suppl. Fig. S1**) and used these statistics to conduct post-filtering. Specifically, unbound TCR and pMHC structures were aligned with their respective bound reference structures. Subsequently, the distances between exposed peptide residues (IMGT numbers: 4-5, chain C) to the T cell and the tip residues of the CDR3 (IMGT numbers: 109, 110, chain D/E) loop residues for both TCR chains in the unbound structures were calculated (**Table 1**). The minimum and maximum distances for each benchmark case were then recorded. Utilizing the ClusPro restraint set generator, distance restraints were generated based on these criteria as satisfying this rule: $Group\ 1\ \&\ Group\ 2$ (**Table 2**).

2.4.5 Clustering

Following post-filtering we retain up to top 1000 docked models with the lowest energy scores. We cluster the complexes with a greedy clustering algorithm with pairwise Interface Root Mean Square Deviation (i-RMSD) as a distance metric the same as ClusPro but with differing thresholds.

Specifically, we calculate all pairwise i-RMSD for all top 1000 models. Interface residues are defined as TCR residues having any atom within 10 Å of any pMHC atom.

The clustering is performed on the generated matrix of pairwise i-RMSD values for the retained docked conformations. Hierarchical clustering is employed to rank conformations based on their adherence to the specified threshold. Utilizing greedy clustering, our method identifies the ligand position with the highest number of "neighbors" within the defined distance as the cluster center. All members within the threshold are added to this cluster and removed from the dataset. This process repeats until no more clusters can be formed or until reaching a maximum of 100 clusters, as predetermined. While ClusPro utilizes a fixed threshold of 9Å RMSD radius for greedy clustering, we opt for a more flexible approach with a threshold of different cutoffs (C-alpha RMSD).

Clustering of a large number of protein complexes can be time consuming. We adopted our GradPose, an ultrafast structure alignment software, to implement our clustering step. We were able to speed up the process.

2.5 Evaluations

2.5.1 Ligand Root Mean Squared Deviation (L-RMSD), interface RMSD (i-RMSD) and fraction of native contacts (f_{nat})

We evaluated model qualities in terms of standard CAPRI metrics as L-RMSD, i-RMSD, and f_{nat} using our PDB2SQL software [33]. RMSD values were computed based on backbone atoms (Cα, N, C, O), and interface residues were defined as those within a 10 Å distance of atoms on the other molecule.

For L-RMSD, pMHCs of the modeled TCRpMHC complex and the experimentally determined complex were superimposed to compute the RMSD between their TCR conformations. Similarly, I-RMSD was calculated by superimposing the interfaces of the modeled TCRpMHC complex and the experimentally determined complex, followed by assessing the deviation of interface residues. f_{nat} evaluates contact similarity between the modeled TCRpMHC and the experimental complex, while $f_{non\text{-}nat}$ quantifies the proportion of contacts in the modeled TCRpMHC that are absent in the experimental structure.

The variability in modeling interfaces across benchmark docking cases was evaluated in the docking benchmark using i-RMSD and $f_{non\text{-}nat}$, and categorized into Rigid, Medium, and Difficult. This assessment is particularly relevant, especially for flexible surfaces prone to significant conformational changes between bound and unbound states.

2.5.2 Success Rate

Following CAPRI [34], the model qualities are categorized based on how closely they resemble the experimental structure, classified as incorrect, acceptable, medium, or high-quality. The Success Rate is determined by calculating the percentage of cases where at least one model of acceptable, medium, or high-quality appears among the top M ranked models:

$$success\ rate\ (M) = \frac{num\ of\ successful\ cases\ when\ top\ M\ models\ are\ evaluated}{N}$$

Where N is the total number of cases (N = 38 in this study), and a case is counted as a success if at least one hit model (satisfying defined quality criteria[35]) appears among the top M ranked models.

3. Results

3.1 The TCR docking polarity in experimental structures

We downloaded 177 experimentally determined TCRpMHC structures from IMGT [13]/PDB (as of 06/06/2023) and calculated their crossing angles and incident angles. All, except for a few exceptions, the crossing angles fall in [0, 90] degrees, and the incident angles fall within [0, 30] degrees, demonstrating the docking polarity of TCRs (**Fig. 5**).

3.2 SwiftTCR generates good-quality models in minutes

Our swiftTCR method generates three-dimensional structures of TCRpMHC complexes. It tailors docking by using advanced techniques integrated with PIPER software, effectively narrowing the conformational space to identify energetically favorable binding conformations between TCRs and pMHCs.

To assess its performance, we evaluated our method using the TCRpMHC benchmark dataset, which comprises 38 experimentally determined TCRpMHC-I structures [14]. We compared each experimental structure to the swiftTCR generated TCRpMHC conformation, given its corresponding unbound pMHC and TCR structures. Model qualities were categorized into high, medium, and acceptable quality groups according to CAPRI criteria for protein-protein docking quality evaluation [35].

The success of producing models with acceptable, medium, and high-quality from the top-ranked models for every modelling case is shown in **Fig. 6A**. While ClusPro achieved competitive performance, swiftTCR demonstrated better performance. Examining swiftTCR's performance, all the top 10 ranked models demonstrate acceptable quality (100% success rate). Among the top 50 models, swiftTCR produces medium to high-quality results in well over half of the cases.

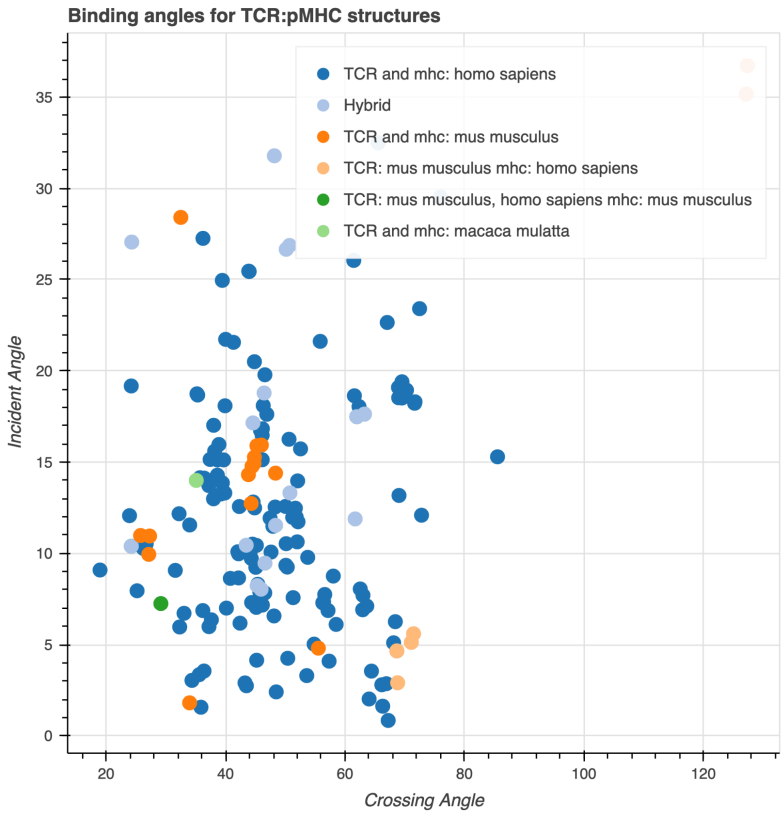


Figure 5. The incident and crossing angles observed in experimental TCRpMHC structures. 177 experimental TCRpMHC structures in the IMGT/PDB (as of 06/06/2023) are used.

The success rate of swiftTCR in reproducing benchmark PDBs was assessed (**Fig. 7B**). Success rate, a widely adopted metric for evaluating the scoring of protein-protein docking tools, is defined as the percentage of cases where at least one acceptable, medium, or high-quality model is among the top N-ranked models. SwiftTCR using the default clustering threshold of 3 Å achieved a notable success rate, with over 75% of the top models and 90% of the top 5 models classified as acceptable quality models, meeting stringent criteria (fnat > 0.1, i-RMSD < 4 Å, L-RMSD < 10 Å). Achieving high quality models in the top 20 models. Different thresholds yield different outcomes (**Suppl. Fig. S2, Fig. 7**): a higher cluster threshold prioritises in the top 20 acceptable quality models, while lower thresholds prioritize better model quality within the top 5 or even the top 1.

Visualizing examples of varying model quality (acceptable, medium, high) models among the top models generated by SwiftTCR, demonstrates that the binding mode of the model for TCR and pMHC closely resembles experimental data (**Fig. 6B**).

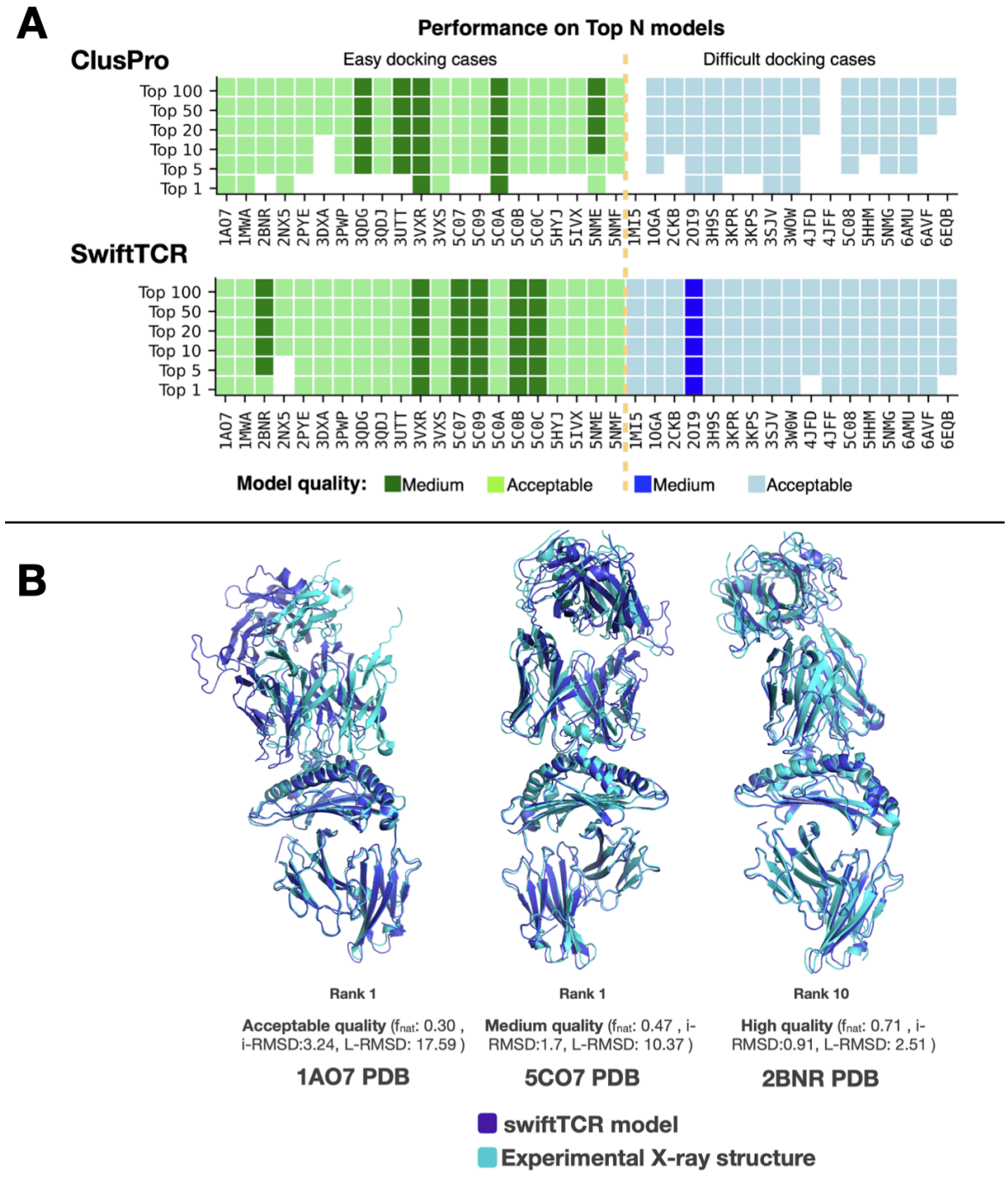


Figure 6. Comparison of swiftTCR and Cluspro performance in modeling TCRpMHC interactions on a benchmark set. A) The set comprises 38 Experimental TCRpMHC-I Structures [14]. A) Quality of models (high, medium, and acceptable) generated by ClusPro. B) Quality of models generated by swiftTCR with a 3 Å clustering threshold. Model quality was assessed based on L-RMSD, i-RMSD, and f_{nat} metrics, following CAPRI criteria. B) Example of a 3D model generated by SwiftTCR (using the default clustering threshold of 3 Å) shown in purple, compared to experimental structures in green. Models are presented for PDB ID: 1AO7 (acceptable quality), PDB ID: 5C07 (medium quality), and PDB ID: 2BNR (high quality). Cases indicated by * are mouse alleles.

3.4 Model Quality Comparison: Our TCRpMHC Modeling Pipeline competes with State-of-the-Art Methods

In the latest benchmark study on TCRpMHC docking tools [14], the expanded TCR benchmark docking set was identified and used to evaluate the performance of four docking platforms: ClusPro [15], LightDock [36] , ZDOCK [16], and HADDOCK [17].
Scenario 1, focusing on simplicity with only CDR loop and peptide residue information, exhibited higher docking success rates compared to scenarios involving additional MHC residue information. Our swiftTCR aligns with Scenario 1, justifying its selection for a fair comparison with other benchmarked docking methods.
ClusPro emerged as the most consistent performer in the benchmark study, utilizing PIPER, an FFT-based docking algorithm, and a clustering step to select native-like interactions. Our pipeline follows a similar approach, employing PIPER and incorporating novel techniques for efficient TCR and pMHC binding. Additionally, our clustering method introduces a swift approach for distance calculations.
While ClusPro demonstrated consistency in success plots for each docking case, our method, SwiftTCR, outperformed ClusPro by generating a greater number of medium and high-quality models **(Fig. 6)**.

3.5 Accelerating TCRpMHC Modeling: Our Approach as the Fastest Physics-Based Method

The computational time for model generation depends on the number of rotation matrices used. By default, PIPER employs 70,000 rotations, taking 27 minutes on a server with 30 CPU cores to dock one case without pairwise distance restraints. Because we focus on sampling around the valid interfaces, we are able to do much more dense sampling (step size of 6 degrees), resulting in 202,491 unique rotations for the 3D globe, which is reduced to 4,329 rotations after applying the docking angle and incident angle filters. This allows swiftTCR to generate models for one TCRpMHC modeling case in minutes on one Intel(R) Xeon(R) Gold 6142 CPU @ 2.60GHz.

The state-of-the-art docking tools like the HADDOCK, Zdock and ClusPro [15] servers provide user-friendly interfaces but often demand several hours to complete the docking process for a single TCR and pMHC case. Consequently, their computational runtimes remain prohibitive for large-scale modeling tasks involving TCRpMHC complexes, particularly for AI applications like structural TCR specificity predictions.

3.6 Clustering and Scoring

Similar to ClusPro, after generating 1000 low-energy models from the docking process, we perform clustering based on the RMSD radius of docked models. Clusters are formed using a greedy clustering approach as follows: First, the model with the most neighbors within a specified RMSD radius is taken as cluster center. This model represents the first cluster and it and all members are removed from the remaining set. This process is repeated until no more clusters can be formed or the maximum number of clusters is reached.

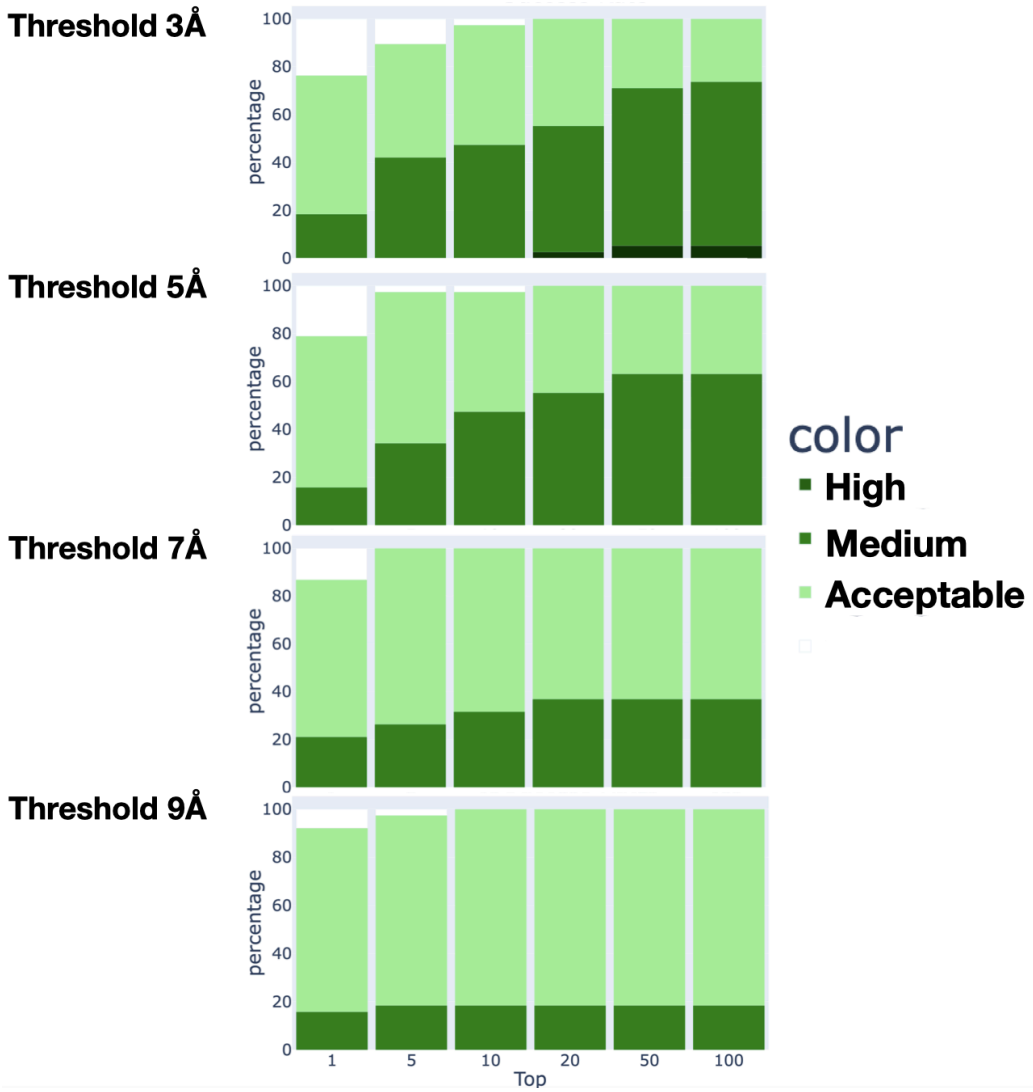


Figure 7. Evaluation of swiftTCR performance using different clustering thresholds. The success rates with different clustering thresholds in generating top-ranked models on the benchmark set, representing cluster thresholds 3,5,7,9 (Å) respectively.

ClusPro uses a clustering cutoff with a 9 Å RMSD. This approach aims to identify the native site, presuming it has the highest frequency of occurrences. They designate the ligand position surrounded by the most "neighbors" within a 9 Å radius, as a cluster center, with its neighboring positions becoming cluster members. Members are then excluded in the search for additional cluster centers till no more clusters can be formed or the maximum set of clusters is reached. SwiftTCR uses a smaller default clustering cutoff: 3 Å. While ClusPro uses a 9 Å threshold to explore the entire conformational landscape, our specific sampling space, focused on TCR docking polarity to pMHC, leads to significantly fewer clusters when using the same threshold. By employing a lower threshold in our approach, we generated more clusters, thereby enhancing diversity. This flexible threshold allowed for the identification of structurally similar docked conformations within individual clusters.

Finally, models are ranked based on cluster size. The largest cluster is ranked top. When two clusters have the same cluster size, the cluster centers are ranked based on energy from PIPER. The cluster center is selected as the representative of the cluster.

In our experimentation with different thresholds on the benchmark cases, we evaluated the success rates to assess the impact of employing various thresholds on the quality of selected top-ranked models. (**Suppl. Fig. S2, Suppl. Fig. S4**). The threshold of 3 Å showcased superior performance in generating more medium to high-quality models. Considering medium and high-quality models, the threshold of 3 Å performs better among the top 10 ranked models.

4. Discussion

Three-dimensional TCRpMHC structures are invaluable for deciphering the long-elusive TCR recognition code. To this end, we have developed SwiftTCR, a fast and reliable integrative rigid-body docking software for modeling TCRpMHC complexes. Its unique features include uniform sampling for evenly distributed docking orientations, reduced rotation sets for TCRpMHC interactions, and a modular Python design. Our protocol achieves rapid modeling speeds (~5 min on 30 CPUs), outperforming state-of-the-art docking tools in model quality and computational efficiency (25 times faster).

SwiftTCR is built upon PIPER [18] for sampling and energy calculation for TCRpMHC modeling. SwiftTCR utilizes a dual criterion for selecting top models: prioritizing energy first (preferring lower values), followed by cluster sizes (preferring larger clusters, and reporting the cluster center as the representative model) to efficiently group similar models, aiding in the selection of near-native TCRpMHC orientations. The idea of using cluster size to rank the models was originally used in ClusPro [15]. The rationale is that docking software typically samples more models in the lowest energy funnel. The results from our analysis with different clustering cutoffs confirms the effectiveness of cluster sizes in ranking models (see **Fig. 7**, **Suppl. Fig. S2**).

While SwiftTCR is able to generate medium to high-quality models for most cases (20 out of 38 cases, using default threshold), they are not necessarily ranked top, indicating the need for improving our scoring scheme. Several improvements could be made. A more refined energy function could increase accuracy. Implementing a multi-step clustering approach, starting with large cutoffs to select the top 1-2 clusters and refining with smaller cutoffs, may also help. Additionally, including coevolution information between TCR and MHC could improve both the sampling step, enabling the generation of more medium-high quality models, and the scoring function.

Artificial intelligence (AI) based modelling approaches like AlphaFold-X [39, 40] have achieved impressive results in modelling structures of single-chain proteins and protein complexes. In contrast to data-driven AI-based methods (e.g., AlphaFold [39, 40]), SwiftTCR employs a physics-based docking method. While AlphaFold-X relies on evolution information from multiple sequence alignments (MSA), its performance is limited for antibodies, peptides and other proteins lacking large MSA data. Moreover, AI-based methods are essentially

statistics-based methods, meaning they potentially are prone to biases towards specific structural templates or alleles existing in the PDB (e.g., the comparison between AlphaFold2 and our physics-based method PANDORA in [22]). Physics-based approaches, on the other hand, are energy-guided modeling, potentially extending their utility to novel TCRs, peptides, and MHCs.
In our analysis, we found no significant correlation between model performance and CDR3 lengths (**Suppl. Fig. S3**). However, evidence suggests a relationship between performance and CDR2 conformational changes in specific cases (PDB ID: 2NX5, 4JFD, 6EQB, **Suppl. Fig. S5**). Further research is needed to understand the factors influencing TCRpMHC binding, particularly considering the flexibility of CDR loops.

Given SwiftTCR's rigid-body docking nature, additional flexible refinement is necessary to capture the dynamic conformational changes of TCR loops upon binding to pMHC. Specifically, techniques like loop repacking are needed to remove clashes, and enhance interaction energy.
SwiftTCR requires at least the TCR's variable domain to begin modeling, regardless of the input's constant domain presence. Regarding the MHC side, this study currently focuses on MHC class I. However, it can be readily expanded to MHC-II by specifying rotation sets suitable for TCRpMHC-II structures (ongoing work).

SwiftTCR's rapid modeling of TCRpMHC complexes offers insights into TCR chain pairing from bulk sequencing and functional TCR identification. It may also provide a missing solution for TCR specificity predictions, where most existing methods are sequence-based without prediction power to unseen peptides [41]. SwiftTCR offers a fast way to model TCRpMHC structures of interest, whose energy signature and docking orientations may provide signals of TCR specificity. Further, we expect enrichment of TCR sequencing data with 3D structures from SwiftTCR can provide valuable training data for structure-based deep learning tools, advancing capabilities in immunological research and therapeutic design.

Data and Code availability

We will make the data available on Zenodo with DOIs upon the acceptance of the manuscript. And the docking code will be made available on GitHub.

Acknowledgment

FM is supported by the Kika grant (grant number 454). LX is supported by Hanarth Fond. GR was supported by Europees Fonds voor Regionale Ontwikkeling (EFRO) (R0005582). We would like to express our gratitude to the ClusPro team and George Jones for generously sharing the PIPER code and their assistance, which greatly facilitated the implementation of restraints in our work. We also thank Kevin J van Geemen and Daniel T Rademaker, developers of the GradPose software, for making the clustering step fast.

Authors Contribution

Conceived the study: LX, YA, GR and FP. Project supervision and coordination: LX. Manuscript writing: FP and LX. Manuscript editing: YA, LX, GR and SE. Software development: YA. Benchmark evaluation: YA, FP, SE. Figures: FP, YA, and LX.

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

References

[1] S. Zhong *et al.*, "T-cell receptor affinity and avidity defines antitumor response and autoimmunity in T-cell immunotherapy," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 17, pp. 6973–6978, Apr. 2013, doi: 10.1073/pnas.1221609110.

[2] G. Dolton *et al.*, "Emergence of immune escape at dominant SARS-CoV-2 killer T cell epitope," *Cell*, vol. 185, no. 16, pp. 2936-2951.e19, Aug. 2022, doi: 10.1016/j.cell.2022.07.002.

[3] A. D. Waldman, J. M. Fritz, and M. J. Lenardo, "A guide to cancer immunotherapy: from T cell basic science to clinical practice," *Nat. Rev. Immunol.*, vol. 20, no. 11, Art. no. 11, Nov. 2020, doi: 10.1038/s41577-020-0306-5.

[4] N. K. Jerne, "The somatic generation of immune recognition," *Eur. J. Immunol.*, vol. 1, no. 1, pp. 1–9, Jan. 1971, doi: 10.1002/eji.1830010102.

[5] J. P. Scott-Browne, J. White, J. W. Kappler, L. Gapin, and P. Marrack, "Germline-encoded amino acids in the αβ T-cell receptor control thymic selection," *Nature*, vol. 458, no. 7241, pp. 1043–1046, Apr. 2009, doi: 10.1038/nature07812.

[6] F. Van Laethem, A. N. Tikhonova, and A. Singer, "MHC restriction is imposed on a diverse T cell receptor repertoire by CD4 and CD8 co-receptors during thymic selection," *Trends Immunol.*, vol. 33, no. 9, pp. 437–441, Sep. 2012, doi: 10.1016/j.it.2012.05.006.

[7] S. Gras *et al.*, "Reversed T Cell Receptor Docking on a Major Histocompatibility Class I Complex Limits Involvement in the Immune Response," *Immunity*, vol. 45, no. 4, pp. 749–760, Oct. 2016, doi: 10.1016/j.immuni.2016.09.007.

[8] P. Zareie *et al.*, "Canonical T cell receptor docking on peptide–MHC is essential for T cell signaling," *Science*, vol. 372, no. 6546, p. eabe9124, Jun. 2021, doi: 10.1126/science.abe9124.

[9] D. A. Antunes, J. R. Abella, D. Devaurs, M. M. Rigo, and L. E. Kavraki, "Structure-based Methods for Binding Mode and Binding Affinity Prediction for Peptide-MHC Complexes," *Curr. Top. Med. Chem.*, vol. 18, no. 26, pp. 2239–2255, Jan. 2019, doi: 10.2174/1568026619666181224101744.

[10] K. Saotome *et al.*, "Structural analysis of cancer-relevant TCR-CD3 and peptide-MHC complexes by cryoEM," *Nat. Commun.*, vol. 14, no. 1, Art. no. 1, Apr. 2023, doi: 10.1038/s41467-023-37532-7.

[11] D. F. Marzella *et al.*, "Improving generalizability for MHC-binding peptide predictions through structure-based geometric deep learning." bioRxiv, p. 2023.12.04.569776, Dec. 05, 2023. doi: 10.1101/2023.12.04.569776.

[12] H. M. Berman, "The Protein Data Bank," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, Jan. 2000, doi: 10.1093/nar/28.1.235.

[13] F. Ehrenmann, Q. Kaas, and M.-P. Lefranc, "IMGT/3Dstructure-DB and IMGT/DomainGapAlign: a database and a tool for immunoglobulins or antibodies, T cell receptors, MHC, IgSF and MhcSF," *Nucleic Acids Res.*, vol. 38, no. Database issue, pp. D301–D307, Jan. 2010, doi: 10.1093/nar/gkp946.

[14] T. Peacock and B. Chain, “Information-Driven Docking for TCR-pMHC Complex Prediction,” *Front. Immunol.*, vol. 12, 2021, Accessed: Mar. 13, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fimmu.2021.686127

[15] D. Kozakov *et al.*, “The ClusPro web server for protein–protein docking,” *Nat. Protoc.*, vol. 12, no. 2, pp. 255–278, Feb. 2017, doi: 10.1038/nprot.2016.169.

[16] B. G. Pierce, K. Wiehe, H. Hwang, B.-H. Kim, T. Vreven, and Z. Weng, “ZDOCK server: interactive docking prediction of protein–protein complexes and symmetric multimers,” *Bioinformatics*, vol. 30, no. 12, pp. 1771–1773, Jun. 2014, doi: 10.1093/bioinformatics/btu097.

[17] S. J. de Vries, M. van Dijk, and A. M. J. J. Bonvin, “The HADDOCK web server for data-driven biomolecular docking,” *Nat. Protoc.*, vol. 5, no. 5, pp. 883–897, May 2010, doi: 10.1038/nprot.2010.32.

[18] D. Kozakov, R. Brenke, S. R. Comeau, and S. Vajda, “PIPER: An FFT-based protein docking program with pairwise potentials,” *Proteins Struct. Funct. Bioinforma.*, vol. 65, no. 2, pp. 392–406, Aug. 2006, doi: 10.1002/prot.21117.

[19] B. Abanades, W. K. Wong, F. Boyles, G. Georges, A. Bujotzek, and C. M. Deane, “ImmuneBuilder: Deep-Learning models for predicting the structures of immune proteins,” *Commun. Biol.*, vol. 6, no. 1, Art. no. 1, May 2023, doi: 10.1038/s42003-023-04927-7.

[20] R. Yin, H. V. Ribeiro-Filho, V. Lin, R. Gowthaman, M. Cheung, and B. G. Pierce, “TCRmodel2: high-resolution modeling of T cell receptor recognition using deep learning,” *Nucleic Acids Res.*, vol. 51, no. W1, pp. W569–W576, Jul. 2023, doi: 10.1093/nar/gkad356.

[21] P. Bradley, “Structure-based prediction of T cell receptor:peptide-MHC interactions,” *eLife*, vol. 12, p. e82813, Jan. 2023, doi: 10.7554/eLife.82813.

[22] F. M. Parizi, D. F. Marzella, G. Ramakrishnan, P. A. C. ‘t Hoen, M. H. Karimi-Jafari, and L. C. Xue, “PANDORA v2.0: Benchmarking peptide-MHC II models and software improvements,” *Front. Immunol.*, vol. 14, 2023, Accessed: Dec. 09, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fimmu.2023.1285899

[23] R. Fasoulis, M. M. Rigo, G. Lizée, D. A. Antunes, and L. E. Kavraki, “APE-Gen2.0: Expanding Rapid Class I Peptide–Major Histocompatibility Complex Modeling to Post-Translational Modifications and Noncanonical Peptide Geometries,” *J. Chem. Inf. Model.*, vol. 64, no. 5, pp. 1730–1750, Mar. 2024, doi: 10.1021/acs.jcim.3c01667.

[24] M. S. Klausen, M. V. Anderson, M. C. Jespersen, M. Nielsen, and P. Marcatili, “LYRA, a webserver for lymphocyte receptor structural modeling,” *Nucleic Acids Res.*, vol. 43, no. W1, pp. W349–W355, Jul. 2015, doi: 10.1093/nar/gkv535.

[25] G. Tomasello, I. Armenia, and G. Molla, “The Protein Imager: a full-featured online molecular viewer interface with server-side HQ-rendering capabilities,” *Bioinformatics*, vol. 36, no. 9, pp. 2909–2911, May 2020, doi: 10.1093/bioinformatics/btaa009.

[26] “Structural Modeling of Lymphocyte Receptors and Their Antigens | SpringerLink.” Accessed: Aug. 03, 2023. [Online]. Available: https://link.springer.com/protocol/10.1007/978-1-4939-9728-2_17

[27] D. T. Rademaker, K. J. Van Geemen, and L. C. Xue, “GradPose: a very fast and memory-efficient gradient descent-based tool for superimposing millions of protein structures from computational simulations,” *Bioinformatics*, vol. 39, no. 8, p. btad444, Aug. 2023, doi: 10.1093/bioinformatics/btad444.

[28] M. G. Rudolph, R. L. Stanfield, and I. A. Wilson, “HOW TCRS BIND MHCS, PEPTIDES, AND CORECEPTORS,” *Annu. Rev. Immunol.*, vol. 24, no. 1, pp. 419–466, Apr. 2006, doi: 10.1146/annurev.immunol.23.021704.115658.

[29] D. V. Bagaev *et al.*, "VDJdb in 2019: database extension, new analysis infrastructure and a T-cell receptor motif compendium," *Nucleic Acids Res.*, vol. 48, no. D1, pp. D1057–D1062, Jan. 2020, doi: 10.1093/nar/gkz874.

[30] J. J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, New Orleans, LA, USA: IEEE, 2004, pp. 3993-3998 Vol.4. doi: 10.1109/ROBOT.2004.1308895.

[31] K. Shoemake, "UNIFORM RANDOM ROTATIONS," in *Graphics Gems III (IBM Version)*, Elsevier, 1992, pp. 124–132. doi: 10.1016/B978-0-08-050755-2.50036-1.

[32] T. J. Dolinsky *et al.*, "PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations," *Nucleic Acids Res.*, vol. 35, no. Web Server, pp. W522–W525, May 2007, doi: 10.1093/nar/gkm276.

[33] N. Renaud and C. Geng, "The pdb2sql Python Package: Parsing, Manipulation and Analysis of PDB Files Using SQL Queries," *J. Open Source Softw.*, vol. 5, no. 49, p. 2077, May 2020, doi: 10.21105/joss.02077.

[34] M. F. Lensink, R. Méndez, and S. J. Wodak, "Docking and scoring protein complexes: CAPRI 3rd Edition," *Proteins Struct. Funct. Bioinforma.*, vol. 69, no. 4, pp. 704–718, Oct. 2007, doi: 10.1002/prot.21804.

[35] M. F. Lensink and S. J. Wodak, "Docking and scoring protein interactions: CAPRI 2009," *Proteins Struct. Funct. Bioinforma.*, vol. 78, no. 15, pp. 3073–3084, 2010, doi: 10.1002/prot.22818.

[36] B. Jiménez-García, J. Roel-Touris, M. Romero-Durana, M. Vidal, D. Jiménez-González, and J. Fernández-Recio, "LightDock: a new multi-scale approach to protein–protein docking," *Bioinformatics*, vol. 34, no. 1, pp. 49–55, Jan. 2018, doi: 10.1093/bioinformatics/btx555.

[37] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, "GROMACS: Fast, flexible, and free," *J. Comput. Chem.*, vol. 26, no. 16, pp. 1701–1718, Dec. 2005, doi: 10.1002/jcc.20291.

[38] Crocioni *et al.*, "Deeprank2." Zenodo, Nov. 17, 2023. doi: 10.5281/ZENODO.8305000.

[39] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, et al., "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[40] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, et al., "Accurate structure prediction of biomolecular interactions with AlphaFold 3," *Nature*, pp. 1–3, 2024.

[41] F. Grazioli, A. Mösch, P. Machart, K. Li, I. Alqassem, T. J. O'Donnell, and M. R. Min, "On TCR binding predictors failing to generalize to unseen peptides," *Frontiers in Immunology*, vol. 13, Art. no. 1014256, 2022.