Improved prediction methods for understanding the TCR-peptide-MHC interaction

Munk, Kamilla Kjærgaard

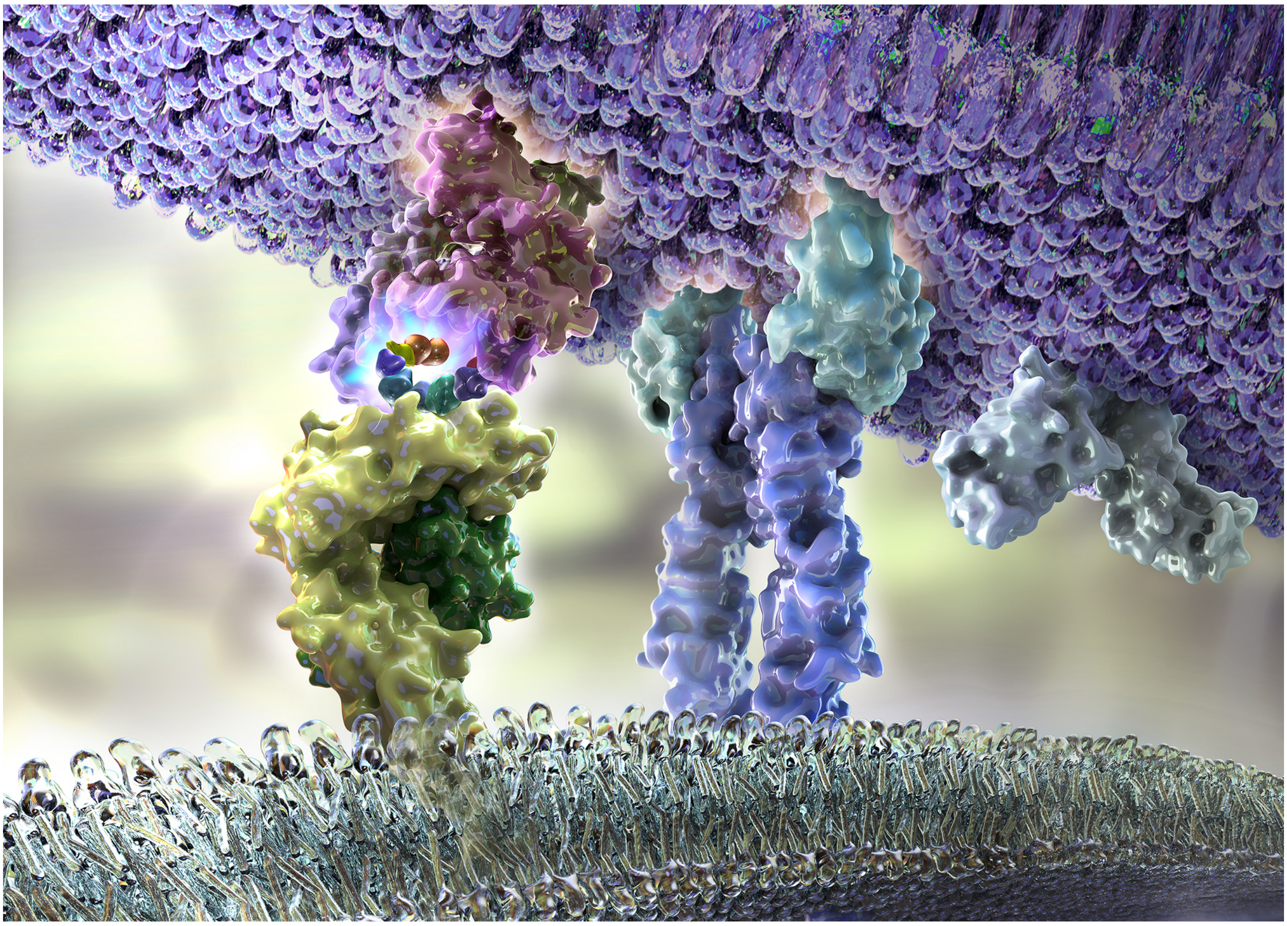
Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Munk, K. K. (2019). *Improved prediction methods for understanding the TCR-peptide-MHC interaction*. DTU Health Technology.

Improved prediction methods for understanding the TCR-peptide-MHC interaction

PhD thesis



Kamilla Kjærgaard Munk
August 2019

DTU Health Tech
Department of Health Technology

Front page art:
© 2018 Audra Geras
www.audrageras.com

DTU Bioinformatics
Department of Bio and Health Informatics
Technical University of Denmark

Preface

The work presented in this thesis has been performed at the Department of Health Technology, in the Immunoinformatics and Machine Learning group at the Technical University of Denmark under the supervision of professor Morten Nielsen and associate professor Paolo Marcatili. The work was carried out between July 2016 and August 2019.

The work was supported by the Technical University of Denmark and the federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN272201200010C.

The thesis consists of three parts: An introduction explaining essential concepts for understanding the scope of the thesis, four research manuscripts and an epilogue discussing the impact and perspectives of each manuscript.

Kamilla K Munk

Kamilla Kjærgaard Munk

Kongens Lyngby, August 2019

Research papers

Research included in this thesis:

PAPER I

Improved methods for predicting peptide binding affinity to MHC class II molecules

Kamilla K. Jensen, Massimo Andreatta, Paolo Marcatili, Søren Buus, Jason A. Greenbaum, Zhen Yan, Alessandro Sette, Bjoern Peters & Morten Nielsen

Published in: Immunology, Volume: 154, Issue: 3, Pages: 394-406, Year: 2018

PAPER II

TCRpMHCmodels: Structural modeling of TCR-pMHC class I complexes

Kamilla K. Jensen, Vasileios Rantos, Emma Jappe, Tobias H. Olesen, Martin C. Jespersen, Vanessa I. Jurtz, Leon E. Jessen, Esteban Lanzarotti, Swapnil Mahajan, Bjoern Peters, Morten Nielsen & Paolo Marcatili

Submitted to: Scientific Reports, Year: February 2019

PAPER III

T cell receptor fingerprinting enables in-depth characterization of the interactions governing recognition of peptide-MHC complexes

Amalie K. Bentzen, Lina Such, Kamilla K. Jensen, Andrea M. Marquard, Leon E. Jessen, Natalie J. Miller, Candice D. Church, Rikke Lyngaa, David M. Koelle, Jürgen C. Becker, Carsten Linnemann, Ton N. M. Schumacher, Paolo Marcatili, Paul Nghiem, Morten Nielsen & Sine R. Hadrup

Published in: Nature Biotechnology, Volume: 36, Issue: 12, Pages: 1191-1196, Year: 2018

PAPER IV

Structural modeling of lymphocyte receptor loops using Generative Adversarial Networks

Kamilla K. Munk, Morten Nielsen & Paolo Marcatili

Data from an ongoing proof-of-concept

Research not included in this thesis:

Book chapter

Modeling of Antibody and T Cell Receptor Structures

Kamilla K. Jensen, Anna Chailyan, Davide Cirillo, Anna Tramontano and Paolo Marcatili

Published in: Encyclopedia of Biophysics, Year: 2017

Genome-wide association and HLA fine-mapping studies identify risk loci and genetic pathways underlying allergic rhinitis

Johannes Waage, Marie Standl, John A. Curtin, Leon E. Jessen, Jonathan Thorsen, Chao Tian, Nathan Schoettler, The 23andMe Research Team, AAGC collaborators, Carlos Flores, Abdel Abdellaoui, Tarunveer S. Ahluwalia, Alexessander C. Alves, Andre F. S. Amaral, Josep M. Antó, Andreas Arnold, Amalia Barreto-Luis, Hansjörg Baurecht, Catharina E. M. van Beijsterveldt, Eugene R. Bleecker, Sílvia Bonàs-Guarch, Dorret I. Boomsma, Susanne Brix, Supinda Bunyavanich, Esteban G. Burchard, Zhanghua Chen, Ivan Curjuric, Adnan Custovic, Herman T. den Dekker, Shyamali C. Dharmage, Julia Dmitrieva, Liesbeth Duijts, Markus J. Ege, W. James Gauderman, Michel Georges, Christian Gieger, Frank Gilliland, Raquel Granell, Hongsheng Gui, Torben Hansen, Joachim Heinrich, John Henderson, Natalia Hernandez-Pacheco, Patrick Holt, Medea Imboden, Vincent W. V. Jaddoe, Marjo-Riitta Jarvelin, Deborah L. Jarvis, Kamilla K. Jensen, Ingileif Jónsdóttir, Michael Kabesch, Jaakko Kaprio, Ashish Kumar, Young-Ae Lee, Albert M. Levin, Xingnan Li, Fabian Lorenzo-Diaz, Erik Melén, Josep M. Mercader, Deborah A. Meyers, Rachel Myers, Dan L. Nicolae, Ellen A. Nohr, Teemu Palviainen, Lavinia Paternoster, Craig E. Pennell, Göran Pershagen, Maria Pino-Yanes, Nicole M. Probst-Hensch, Franz Rüschendorf, Angela Simpson, Kari Stefansson, Jordi Sunyer, Gardar Sveinbjornsson, Elisabeth Thiering, Philip J. Thompson, Maties Torrent, David Torrents, Joyce Y. Tung, Carol A. Wang, Stephan Weidinger, Scott Weiss, Gonneke Willemsen, L. Keoki Williams, Carole Ober, David A. Hinds, Manuel A. Ferreira, Hans Bisgaard, David P. Strachan & Klaus Bønnelykke

Published in: Nature Genetics, Volume: 50, Issue: 8, Pages: 1072-1080, Year: 2018

NetTCR: sequence-based prediction of TCR binding to peptide-MHC complexes using convolutional neural networks

Vanessa I. Jurtz, Leon E. Jessen, Amalie K. Bentzen, Martin C. Jespersen, Swapnil Mahajan, Randi Vita, Kamilla K. Jensen, Paolo Marcatili, Sine R. Hadrup, Bjoern Peters & Morten Nielsen

Published in: bioRxiv, doi.org/10.1101/433706, Year: 2018

NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning
Michael S. Klausen, Martin C. Jespersen, Henrik Nielsen, Kamilla K. Jensen, Vanessa I. Jurtz, Casper K. Sønderby, Morten O. A. Sommer, Ole Winther, Morten Nielsen, Bent Petersen & Paolo Marcatili
Published in: Proteins: Structure, Function, and Bioinformatics, Volume: 87, Issue: 6, Pages: 520-527, Year: 2019

Benchmark datasets of immune receptor-epitope structural complexes.
Swapnil Mahajan, Zhen Yan, Martin Closter Jespersen, Kamilla K. Jensen, Paolo Marcatili, Morten Nielsen, Alessandro Sette & Bjoern Peters
Manuscript in revision - BMC Bioinformatics

Abstract

The last decades have seen a rapid increase in our understanding of the immune system, but there are still many unsolved problems. Solving some of these could be invaluable for future advances in drug development and cancer immunotherapy. This thesis introduces methods for understanding an important interaction in the adaptive immune system.

One of the key events in the adaptive immune system is the interaction between T-cell expressed receptors (TCRs) and peptides bound to major histocompatibility complexes (pMHCs). If the TCR recognizes a pMHC, the T-cell is activated and the peptide driving this activation is called a T-cell epitope. Predicting T-cell epitopes has been a long standing challenge within the field of immunoinformatics. There are two strategies to solve the problem. One is to use the protein sequences, and the other is to use the structures. Data on protein structures is usually quite limited, so developing reliable tools that use just the sequences is of great interest to the field.

A commonly used measure for identifying T-cell epitopes is the pMHC binding strength, as this quantity can be used to limit the number of potential peptide candidates.

In the first project of this thesis, we develop an improved method for predicting such peptide-MHC binding strengths by training a neural network on an extended dataset of peptide binding affinities. Further, we show that the updated methods have superior performance when used for detecting T-cell epitopes.

However, not all MHC presented peptides are immunogenic. So in order to truly understand what makes a peptide immunogenic we need to understand the interaction between TCRs and pMHCs. One way to do this is to build structural models of the TCR-pMHC complex and use these structures to predict the TCR-pMHC binding strength.

In the second project, we develop an automated tool for building such structural models of the TCR-pMHC complex using only the amino acid sequence as input. The tool utilizes comparative modeling techniques and generates accurate models within minutes.

In the third project, we investigate the TCR recognition of pMHCs using an experimental technique which measures the relative binding affinity between TCRs and pMHC variants. The relative binding affinities can be translated into TCR motifs, named TCR fingerprints, and these can be used to identify which peptides can be cross-recognized by the TCR. Structural modeling is used in this project to investigate how the TCR recognition is affected by conformational changes in the peptides.

In the fourth and last project, we present preliminary results on improving structural models of TCRs by using state-of-the-art machine learning techniques to generate the peptide-binding loops.

Collectively, the four projects of the thesis provide improved methods for predicting T-cell epitopes and for structural modeling of the TCR-pMHC complex. We hope that these methods can increase our understanding of T-cell immunogenicity and serve as a foundation for developing improved methods for rational T-cell epitope predictions.

Dansk resumé

I de seneste årtier har vores forståelse af immunsystemet udviklet sig drastisk, men der er stadig mange uløste problemer. Hvis vi løser nogle af disse, kan det have uvurderlige konsekvenser for fremtidens udviklingen af ny medicin og immunterapi til kræftbehandling. I denne afhandling introduceres metoder til at undersøge en af de vigtigste interaktioner i det adaptive immunforsvar.

En af de mest betydningsfulde begivenheder i det adaptive immunforsvar er interaktionen mellem en T-cellereceptorer (TCR) og et peptid bundet til et MHC molekyle (major histocompatibiliy complexes), forkortet pMHC. Hvis en TCR genkender et pMHC, aktiveres T-cellen, og det peptid der driver denne aktivering, kaldes en T-celle epitop. En af de store udfordringer i immunoinformatik er at forudsige T-celle epitoper, og der er her to mulige strategier til at gøre dette. Den første er at bruge proteinsekvensen, og den anden er at bruge strukturen. Mængden af data på proteinstrukturer er forholdsvis begrænset, så der er derfor stor interesse for at udvikle pålidelige metoder, der kun gør brug af sekvenserne.

Når man skal identificere T-celleepitoper, bruger man ofte den såkaldte pMHC bindingsstyrke, til at begrænse antallet af potentielle peptider.

I det første projekt udvikler vi en forbedret metode til at forudsige disse peptid-MHC bindingsstyrker, ved at træne et neuralt netværk på et udvidet datasæt af bindingsaffiniteter for forskellige peptider. Vi viser her, at den opdaterede metode både er bedre til at forudsige bindingsaffiniteter, samt bedre til at forudsige T-celleepitoper.

Som nævnt ovenfor er det ikke alle MHC præsenterede peptider der er immunogene. Så for virkelig at forstå, hvad der gør et peptid immunogent, er vi nødt til at forstå interaktionen mellem TCR'er og pMHC'er. En måde at gøre dette på er at bygge strukturelle modeller af TCR-pMHC-komplekset, og bruge disse strukturer til at forudsige TCR-pMHC-bindingsstyrken.

I det andet projekt udvikler vi et automatiseret værktøj til at bygge sådanne strukturelle modeller af TCR-pMHC-komplekset ved kun at bruge aminosyresekvensen som input. Værktøjet bruger komparative modelleringsteknikker og genererer nøjagtige modeller inden for få minutter.

I det tredje projekt undersøger vi hvordan en TCR genkender pMHC'er ved hjælp af en eksperimentel teknik, som måler den relative bindingsaffinitet mellem en TCR og pMHC varianter. De relative bindingsaffiniteter kan oversættes til TCR-motiver, som kan bruges til at identificere hvilke peptider, der kan krydsgenkendes af en TCR. Vi bruger her strukturel modellering til at undersøge, hvordan TCR genkendelse af pMHC'er påvirkes af konformationelle ændringer i peptiderne.

I det fjerde og sidste projekt viser vi de foreløbige resultater fra en metode der kan forbedre den strukturelle modellering af TCR'er. Denne metode bruger avancerede machine learning teknikker til at forudsige strukturen af den del af TCR’en der er ansvarlig for bindingen til pMHC’erne.

Samlet set omhandler de fire projekter i afhandlingen forbedrede metoder til at forudsige T-celleepitoper og til at lave strukturelle modeller af TCR-pMHC-komplekset. Vi håber, at disse metoder kan øge vores forståelse af interaktionen mellem TCR-pMHC-komplekset og fungere som et fundament for at udvikle forbedrede metoder til at forudsigelse T-celleepitoper.

Acknowledgments

I would like to start by thanking Morten Nielsen and Paolo Marcatili for their supervision throughout my PhD – you are both truly amazing people to work with. Thanks to both of you for all our scientific discussions, for the guidance and support you have given me throughout the years. You have both been there to help and encourage me when things were difficult. I am truly grateful for all that I have learned from both of you and hope to continue the collaboration in the future.

Next, I would like to thank all my former and present colleagues for providing a great working environment with room for all types of scientific discussion. I have been very fortunate to have had such good colleagues.

Lastly, I want to give a special thanks to the people at the Institute of Biotechnological Research at the University of San Martin, for making sure that I had an amazing stay in Argentina and to the people from the Section of Experimental and Translational Immunology for a great collaboration.

Abbreviations

ACC	Accuracy
AFND	Allele Frequency Net Database
APC	antigen presenting cells
AUC	area under the ROC curve
β2m	β2-microglobulin
BCR	B-cell receptor
CAPRI	Critical Assessment of PRedicted Interactions
CASP	Critical Assessment of protein Structure Prediction
CD4	cluster of differentiation 4
CD8	cluster of differentiation 8
CDR	complementary determining region
CNNs	convolutional neural networks
FPR	false positive rate
GAN	Generative Adversarial Network
HLA	human leukocyte antigen
IC50	concentration required to achieve 50% inhibition
IEDB	Immune Epitope Database
LSTM	long short-term memory
MCC	Matthew Correlation Coefficient
MHC	major histocompatibility complex
MS	mass spectrometry
PCC	Pearson Correlation Coefficient
PDB	Protein Data Bank
PFR	peptide flanking region
pMHC	Peptide-MHC complex
RMSD	root mean square deviation
RNN	Recurrent neural networks
ROC	receiver operating characteristic
SGD	stochastic gradient descent
TCR	T-cell receptor
TM-score	template modeling score
TPR	true positive rate
WGAN	Wasserstein GAN
WGAN-GP	WGAN with gradient penalty

Contents

Scope of thesis

The main focus of the adaptive immune system is to keep the host healthy by eliminating pathogenic infections and malfunctioning cells [1]. One of the key players in the adaptive immune system is the T-cell. These cells monitor the health of the host using T-cell receptors (TCRs) that interact with peptides presented by major histocompatibility complexes (MHCs) found on the surface of antigen-presenting cells. If a specific T-cell recognizes a peptide presented by an MHC, the T-cell can become activated. T-cell activation results in T-cell proliferation, which in turn eliminates the threat of the pathogen-infected or malfunctioning cell [2]. Thus, some of the main challenges for our understanding of cellular immunity and T-cell activation are to understand which peptides are presented by the MHC, and to understand the interplay between TCRs and peptide-MHC (pMHC) complexes. Gaining a better understanding of these key events in the adaptive immune system could lead to the development of advanced T-cell based immunotherapies and rational vaccines [3].

The activation of T-cells are primarily driven by two processes: Peptide presentation on MHC molecules and T-cell receptor recognition of these MHC-bound peptides [4, 5]. Peptides with the ability to activate an effective T-cell response are called T-cell epitopes.

Today T-cell epitopes are mainly found using experimental methods, but these are both time-consuming and expensive. The development of less time-consuming, more cost effective and more reliable tools for predicting T-cell epitopes would therefore be of considerable interest to the industry and in research.

The overall aim of this PhD thesis was therefore to develop methods for improved prediction of T-cell epitopes and to enhance our understanding of the molecular interactions found between MHC-presented peptides and TCRs.

The thesis is structured in the following way:

Chapter 1 covers the background of the biology together with a description of existing methods with a discussion of the key advantages and disadvantages for each.

Chapter 2 introduces the first scientific paper of this thesis. The main aim of the project behind this paper was to develop improved methods for predicting peptide-MHC binding for MHC class II molecules. This was done by updating NetMHCII [6] and NetMHCIIpan [7], using a new data set obtained from the Immune Epitope Database (IEDB). The paper shows that training with this new data set improved the performance for the peptide-MHC binding predictions for both NetMHCII and NetMHCIIpan.

Chapter 3 introduces the second scientific paper. The main aim of the project behind this paper was to develop an automated computational tool for structural modeling of the TCR-pMHC complex using only the amino acid sequence as input. The resulting tool, named TCRpMHCmodels, utilizes comparative modeling to generate accurate TCR-pMHC models within a few minutes.

Chapter 4 introduces the third scientific paper. The main aim of the project behind this paper was to investigate the TCR recognition of pMHCs using an experimental technique developed by Amalie K. Bentzen. My main contribution to this scientific paper was visualization of the experimental results, generation of sequence motifs, and an investigation of the TCR-pMHC interaction using structural models.

Chapter 5 presents an ongoing project of applying a Generative Adversarial Network (GAN) architecture [8] to predict antigen-binding loops in T-cell receptors and B-cell receptors. Preliminary results from this project indicate that the GAN is capable of learning structural features from these loops, but the accuracy of the generated loops needs to be improved.

Chapter 6 provides a summary of the PhD thesis and reflects on all four projects and provides future perspectives.

Chapter 1: Introduction

Immune system

The immune system is a host's defense mechanism against threats such as pathogenic invasions from bacteria, viruses and malfunctioning cells. The immune system can broadly be divided into two subsystems - the innate and the adaptive immune system [1]. The innate immune system is fast and non-specific, while the adaptive immune system is slower but highly specific. This thesis focuses on the adaptive immune system.

In the adaptive immune system, there are two major types of cells: T-cells, and B-cells. The purpose of T-cells is to activate other immune cells and eliminate infected cells, whereas B-cells are responsible for producing antibodies with the ability to bind pathogens and flag them for destruction [9]. Both B-cells and T-cells have the ability to develop long-term protection, whereby the immune system can quickly and efficiently respond to re-exposure of the same threat.

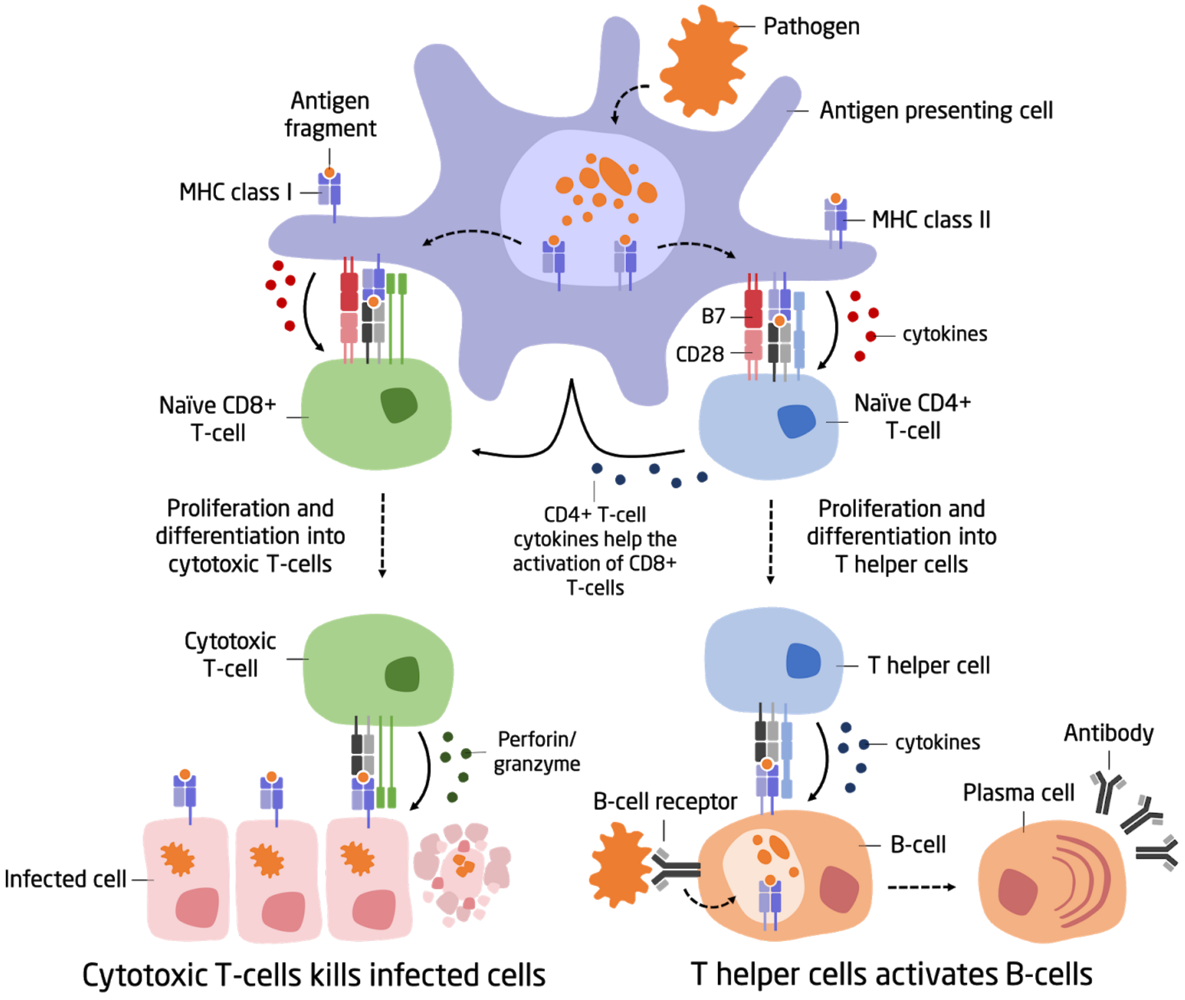

Pathogen
Antigen presenting cell
Antigen fragment
MHC class I
MHC class II
B7
CD28
cytokines
Naïve CD8+ T-cell
Naïve CD4+ T-cell
Proliferation and differentiation into cytotoxic T-cells
CD4+ T-cell cytokines help the activation of CD8+ T-cells
Proliferation and differentiation into T helper cells
Cytotoxic T-cell
Perforin/ granzyme
T helper cell
cytokines
Antibody
Plasma cell
B-cell receptor
B-cell
Infected cell
Cytotoxic T-cells kills infected cells
T helper cells activates B-cells

Figure 1: Schematic representation of the activation of the adaptive immune response. See text for further explanation.

The main function of the adaptive immune system is to recognize antigens and generate specific immune responses toward these. An antigen is a foreign macromolecule that reacts with and stimulates B-cells and T-cells.

Antigens are presented to the immune system by major histocompatibility complexes (MHC). There are two major classes of MHC molecules: MHC class I and MHC class II [10]. MHC class I molecules are found on all nucleated cells where they present fragments of antigens from pathogens or malfunctioning cells. These antigenic fragments can then be recognized by T-cells from the adaptive immune system. MHC class II molecules are primarily found on specialized antigen-presenting cells (APCs) where they present antigen fragments from the extracellular space. When the host is attacked by a pathogen the APCs will engulf the pathogen and digest it into antigen fragments, which are loaded onto the MHC molecule. MHC-presented antigens have the ability to activate T-cells and B-cells, which will lead to pathogenic clearance [1]. An illustration of the T-cell and B-cell activation is shown in Figure 1.

T-cell and B-cell activation

T-cells are initially produced in the bone marrow, but once produced, they migrate to the thymus where they mature into naive CD8+ T-cells (cytotoxic T-cells) or naive CD4+ T-cells (T-helper cells), expressing either CD8 or CD4 co-receptors [11]. Depending on the type of co-receptor, the T-cell can be activated when its T-cell receptor (TCR), found on the surface of the T-cell, interacts with antigens bound to MHC molecules of either class I or class II, found on the surface of antigen-presenting cells [1]. Naive CD8+ T-cells are activated when the TCR interacts with an antigen bound to an MHC class I molecule, while the CD4+ T-cells are activated when the TCR interacts with an antigen bound to an MHC class II molecule.

For long-lasting T-cell activation and differentiation, other signals such as co-stimulation and differentiation signals are also required [1]. The co-stimulation signal is responsible for the long-term survival of T-cells and this signal is mainly triggered when a B7 molecule from an antigen-presenting cell interacts with a CD28 molecule from the T-cell [12]. The differentiation signal is mainly driven by cytokines produced by the antigen-presenting cells and depends on the type of cytokines. In the activation process, naive CD4+ T-cells are stimulated to differentiate into different T-helper cells, while naive CD8+ T-cells are stimulated to become cytotoxic T-cells. The function of cytotoxic T-cells is to identify and eliminate pathogen-infected cells, while the function of T-helper cells is the activation of both cytotoxic T-cells and B-cells.

B-cells are also produced in the bone marrow, but unlike T-cells, B-cells remain in the bone marrow where they mature into naive B-cells. The activation of naive B-cell begins when the B-cell receptor (BCR), found on the cell surface of the B-cell, interacts with extracellular proteins from a pathogen. After the initial interaction, the pathogen is engulfed by the B-cell and digested into antigens which are then presented on the cell surface by MHC class I molecules. Activated T-helper cells then register antigens presented by the B-cell, after which the B-cell is activated. Activated B-cells migrate to the lymph nodes where they proliferate into antibody-producing plasma cells, which primary function is to eliminate extracellular pathogens such as pathogenic bacteria and viruses by neutralizing them or marking them for destruction by other immune cells [1].

The main aim of this thesis was to develop prediction methods for improved understanding of the TCR-peptide-MHC interaction and the following sections will therefore give an in-depth description of the individual components taking part in this interaction.

MHC molecules: Structure and diversity

As described earlier the MHC molecules can be divided into two classes called MHC class I and class II (see Figure 2). Both classes are transmembrane molecules and their structures are very conserved both within and between MHC classes [1, 13].

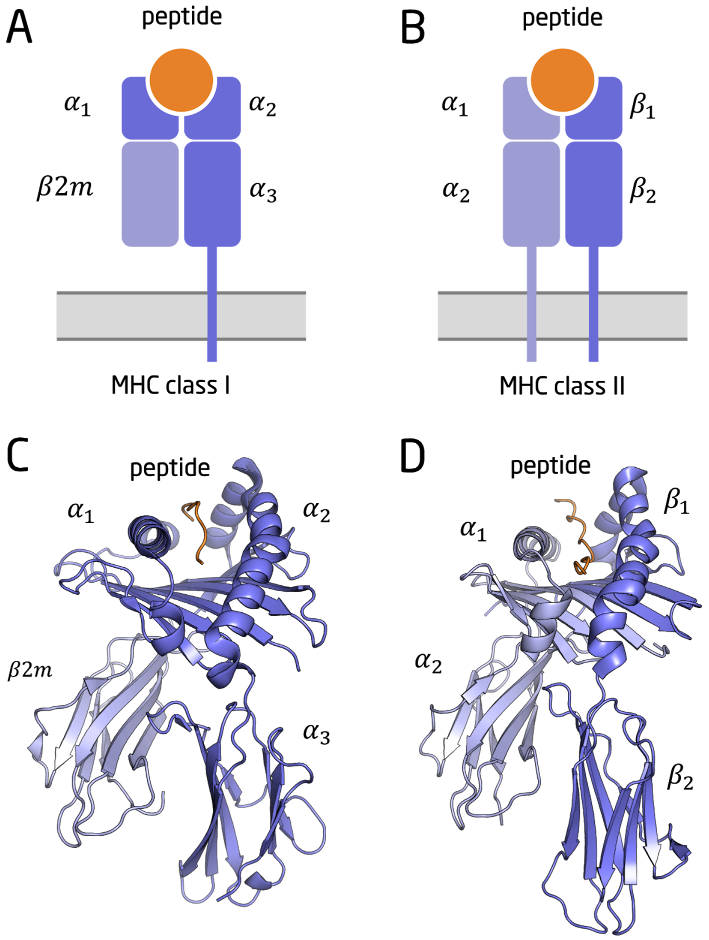


Figure 2: MHC class I and class II molecules. Panel **A)** and **B)** show schematic representations of MHC molecules of class I and class II, respectively. The regions indicated by α and β refer to MHC chains described in the main text. Panel **C)** and **B)** depicts structural representation of the MHC molecules of class I and class II, respectively. The structural representations were made in PyMOL with the use of PDB structures 1OGA and 3C5Z.

The MHC class I molecule is composed of two chains: A membrane-spanning α chain and a β2-microglobulin (β2m) chain. The α chain folds into three domains: α_1, α_2 and α_3. The region between the α_1 and α_2 domains is called the peptide binding groove, and it is here peptides bind to the molecule [1]. For MHC class I molecules the binding groove is relatively narrow at the ends and the length of peptides binding to this MHC class is therefore usually short. The typical peptide length is between 8 and 11 residues long, with peptides of 9 residues being the most abundant [14]. As the peptide length increases, the narrow ends of the binding groove forces the central residues of the peptide up and out of the groove to accommodate the length of the peptide. Peptides presented by the MHC class I molecule will therefore generally assume a centrally bulged conformation, which can then be recognised by the TCR. Peptides binding to MHC class I molecules are selected based on their ability to bind to two specific MHC pockets within the binding groove of the molecule, termed P2 and P9 [15] (See Figure 3). Peptide residues interacting with these positions are termed anchor residues.
The MHC class II molecule is composed of two chains, named the α-chain and β-chain [1]. Both chains are membrane-spanning and they each contain two domains: α_1, α_2, β_1 and β_2. Compared to the MHC class I molecule, the MHC class II molecule has a peptide binding groove which is open at both ends, enabling this MHC class to bind longer peptides. The most common peptide lengths for this class is between 13 and 25 residues, with peptides of 15 residues being the most abundant [16]. The peptide binding specificity of MHC class II molecule is defined by four binding pockets named P1, P4, P6 and P9 [15] (See Figure 3). The part of the peptide interacting with these binding pockets is called the peptide binding core, while the parts of the peptide extending out of the binding groove is called the peptide flanking regions (PFRs).

MHC molecules varies both within each individual person and within the population as a whole. This characteristic ensures a broad immunological protection against any pathogen.

In humans, MHC molecules are encoded in the human leukocyte antigen (HLA) locus [17]. There are three MHC class I α-chain genes, called HLA-A, HLA-B and HLA-C, and there are three pairs of α- and β-chain genes called HLA-DR, HLA-DP and HLA-DQ. Thus, the presence of several different genes of each MHC class ensures that any one individual possesses different MHC molecules.

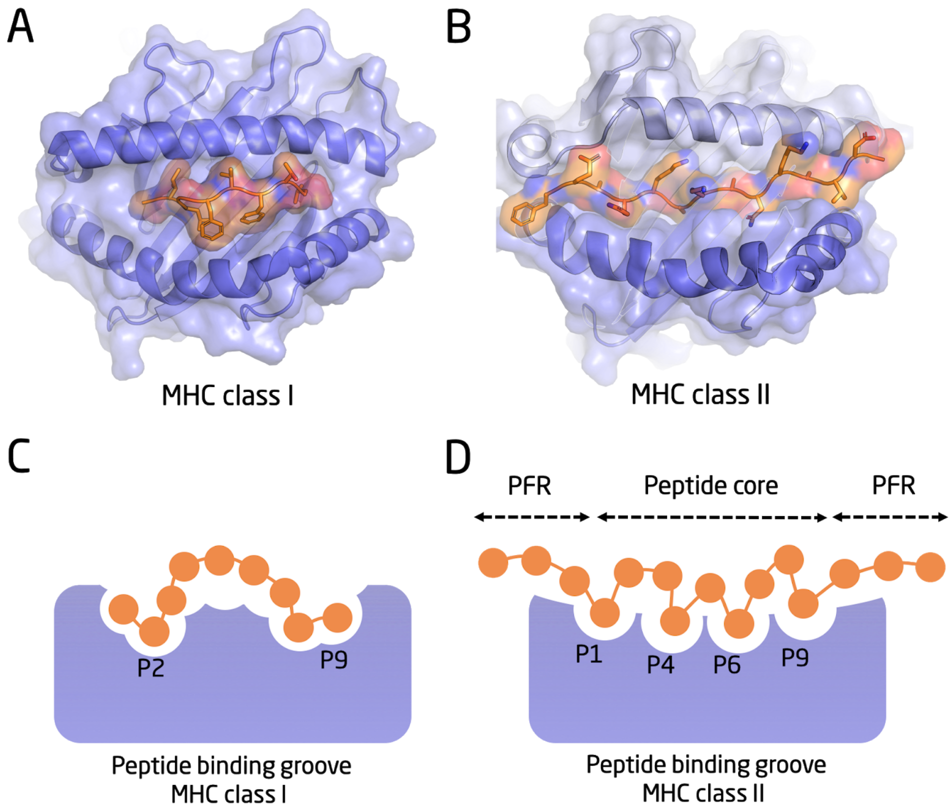


Figure 3: Peptide binding groove of the MHC class I and class II. The part of the peptide that interacts with the MHC is called the peptide core and the rest is called the peptide flanking regions (PFRs). The MHC is shown in purple and the peptide in orange. Circles represent individual amino acids of the peptide. Panel **A)** and **B)** depict the structural representation of the binding groove of an MHC molecule of class I and class II, respectively. **C)** Schematic representation of the peptide binding groove of the MHC class I molecule showing the binding pockets P2 and P9. **D)** Schematic representation of the peptide binding groove of MHC class II molecules showing the binding pockets P1, P4, P6 and P9. Structural representations were made in PyMOL with the use of PDB structures 1OGA and 3C5Z.

Furthermore, each MHC molecule has the ability to present a large amount of highly diverse peptides. This is achieved by having only a few interactions between specific residues in the peptide and residues found in MHC binding pockets. The diversity of the residues at different positions in the peptide can be visualized using sequence logos (see Figure 4). Sequence logos are visual representations of residues essential for the binding of a given peptide to a given MHC molecule. Here, we see that peptide binding to MHC class I molecules have a specific amino acid preference at the 2nd and 9th positions, while MHC class II molecules tend to bind peptides with a specific amino acid preference at the 1st, 4th, 6th and 9th position. Positions which are non-influential on binding have higher diversity and therefore lower information.

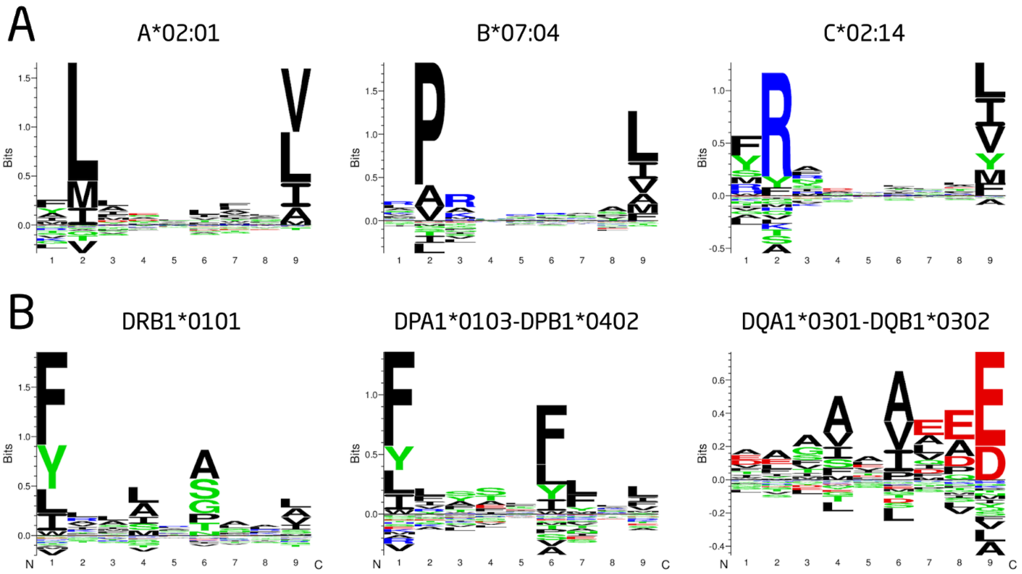


Figure 4: Sequence motifs showing the predicted peptide binding preference for different HLAs. **A)** Peptide binding specificity for three MHC class I molecules. **B)** Peptide binding specificity for three MHC class II molecules. The logos shown in the figure was generated using the motif viewer from NetMHCpan-4.0 [18] and NetMHCIIpan-3.2 [19]. Over-represented amino acids are shown on the positive y-axis and under-represented amino acids on the negative y-axis.

Within the population as a whole, the number of different MHC molecules is even larger as the MHC genes are highly polymorphic [17]. In humans the HLA locus is the most polymorphic region in the human genome, with more than 130,000 HLA variations identified and cataloged in the Allele Frequency Net Database http://www.allelefrequencies.net.

T-cell receptors: Structure and diversity

The T-cell receptor (TCR) is a heterodimeric protein composed of two transmembrane chains with the ability to recognise peptides presented by the MHC molecule. In humans, around 95% of T-cells express TCRs with α- and β-chains, whereas around 5% of T-cells express TCRs with γ- and δ-chains [20]. In this thesis we only worked with the αβ TCRs and the following section will therefore exclusively focus on these.

Each of the TCR chains has a variable and constant region. Located within the variable regions are three complementarity determining regions (CDRs), named CDR1, CDR2 and CDR3 (see Figure 5). The CDRs consist of loops, and they account for the interaction with the pMHC complex.

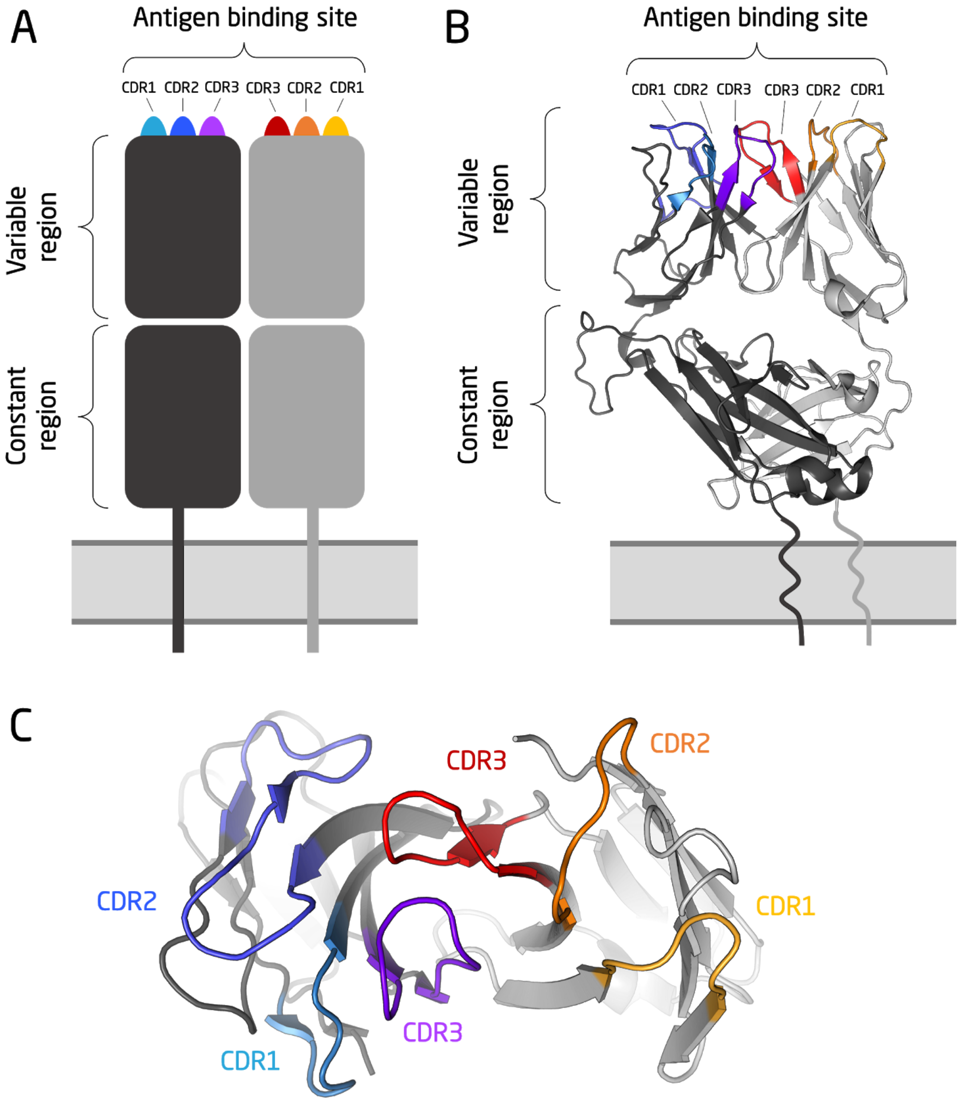


Figure 5: Structure of the T-cell receptor. **A)** Schematic representation of the T-cell receptor. **B)** Side view of the structural representation of the T-cell receptor. **C)** Top view of the structural representation of the T-cell receptor, showing the location of the complementary determining regions (CDRs). Structural representations were made in PyMOL with the use of the PDB structure 1OGA.

During early T-cell development each chain within the TCR is generated through a process known as somatic V(D)J recombination [21, 22]. The result of this process is highly variable CDRs which grant T-cells the ability to recognize and respond to a large variety of antigens and thereby specifically target many different pathogens and malfunctioning cells. The somatic recombination process can theoretically generate more than 10^{15} T-cell variants [23], but only a sizable fraction of these, around 10^{6} to 10^{8}, are expressed at any given time in the human organism [24].

The most variable part of the TCR is the CDR3 loop [22]. This loop is found in the center of the antigen binding site of the TCR and it interacts with the peptide, thus accounting for most of the TCR specificity. CDR1 and CDR2 loops are less variable and these parts of the TCR mostly interact with the MHC [25]. When predicting the TCR-pMHC interaction, it is therefore important to focus on the CDR loops as they have a huge impact on the TCR binding specificity.

TCR-pMHC complexes: Structure and binding

TCR recognition of peptides presented by either MHC class I or class II molecules has been demonstrated in several X-ray crystallographic studies [13]. These show how the TCR binding orientation is similar for TCRs, irrespective of whether they are recognizing peptides presented by the MHC class I and MHC class II molecules (See Figure 6).

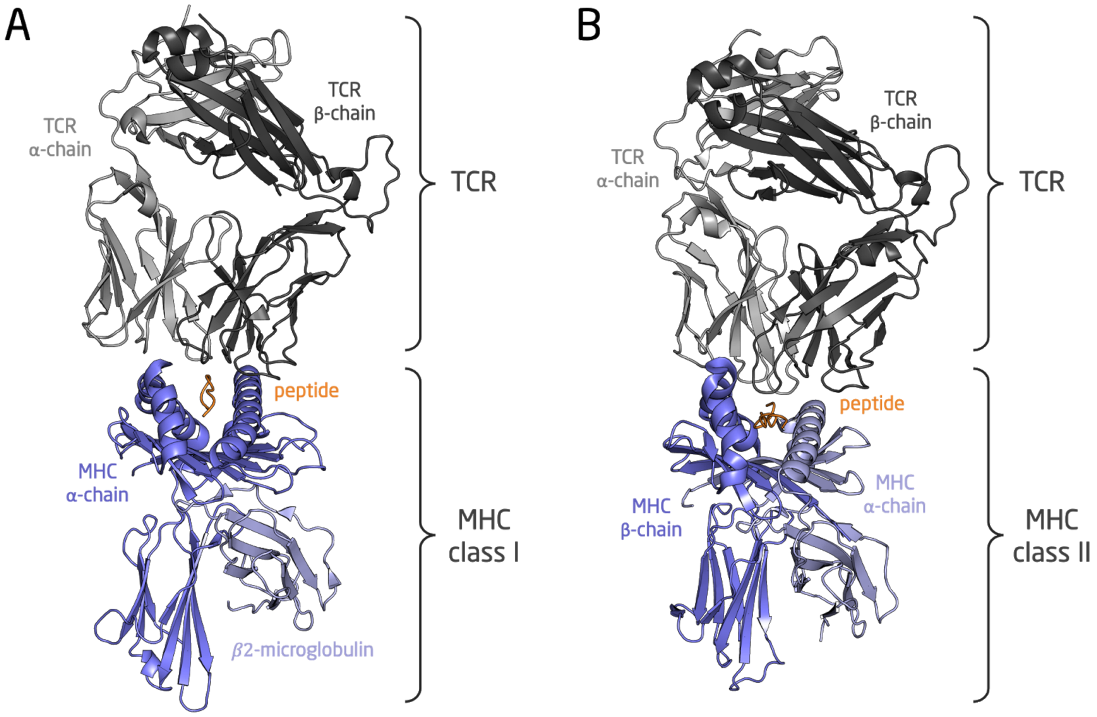


Figure 6: Structural representation of the TCR-pMHC complex for the TCR binding to the peptide presented by either **A)** the MHC class I or **B)** the MHC class II molecule. The structural representations were made in PyMOL with the use of PDB structures 1OGA and 3C5Z.

In both cases, the TCR is oriented approximately directly on top of the pMHC and the variable part of the TCR is in contact with the peptide and the MHC molecule. For a more thorough review of the TCR and MHC interactions see Gruta *et al.* [26] and Rudolph *et al.* [13].

Machine learning

Machine learning algorithms are mathematical models which can be trained to identify non-obvious patterns in a dataset and use these patterns to understand the dataset or to develop predictive models [27]. In the last decades, many different machine learning techniques have been developed and successfully applied to solve complex problems, including image recognition [28, 29] and natural language processing tasks, such as speech recognition [30] and language translation [31]. Many of these machine learning techniques are now being used to solve biological problems within the field of bioinformatics [32, 33].

Dataset preparation

When using machine learning to solve biological problems it is extremely important to remove redundancy within the dataset, encode non-numerical data points and minimize overfitting by splitting the dataset into different partitions. Since these tasks are of such critical importance, we will devote a few short sections to explain them in greater detail.

Data redundancy

A machine learning model usually adapts to the distribution of the training data, and removing redundant data is therefore important to ensure that the model is not trained to overrepresent one type of data.

Methods for eliminating redundant protein sequences in a data set include clustering algorithms such as Hobohm [34], CD-HIT [35], PISCES [36] and UCLUST [37]. These clustering methods use sequence similarity between data points to generate clusters of similar sequences, and a non-redundant dataset can then be made using only a single datapoint from each of the clusters. For example, in project I redundant peptides containing different peptide-MHC binding values were clustered and an average of these values were used in the final dataset for training.

Feature extraction and encoding

For most biological data, additional domain-specific features can be extracted to provide more information for training. Within structural biology, this could be extracting structural features, such as backbone angles and distances between atoms.

After cleaning the dataset and extracting additional features, all non-numerical data points, such as protein sequences, need to be encoded as numbers. Protein sequences are usually encoded using one-hot or BLOSUM encoding. In one-hot encoding, each amino acid is encoded using a vector of 20 bits where the number 1 represents the current amino acid while the remaining bits are zeros. BLOSUM encoded sequences uses the so-called BLOSUM score to encode each amino acid [38].

Overfitting and cross validation

One of the most common mistakes in machine learning is to overestimate the model performance. This usually happens if data points used for testing the model performance have also been used during the training process [39]. In this case, the model could have learned the noise or random fluctuations in the training data instead of general concepts. If this has happened the network is said to be overfit. An overfit model will therefore have a good model performance on the training data, but will fail to make accurate predictions for data points, which have not been part of the training process.The main problem with overfitting is that we cannot know how well a model will perform on new data until we actually test it.

A common approach to solve this problem is to either split the dataset into three independent subsets, called training set, validation set and test set, or to use a technique called cross validation.

Cross validation techniques can be used to assess how the neural network model will generalize to an independent data set [40]. The idea behind cross validation techniques is to train the network model on one subset of the dataset while validating the model performance on another.

One of the most popular implementation of this technique is known as *K*-fold cross validation, where *K* is the number of subsets. $K-1$ of the *K* partitions are used for network training while the *K*'th partition is used to validate the model performance. This process is then repeated *K* times, such that each partition has been used once as a validation set.

Other cross-validation techniques includes leave-p-out, where a number p of observations are left out of the training set and used as the validation set. A special case of this is used in project I, where we wanted to test the networks ability to predict peptide binding of uncharacterized MHC class II molecules. In this case we left out any data points belonging to a specific MHC class. The remaining data was used for training a network, and its performance was evaluated using the data which was left out of the training set.

One of the most commonly used machine learning techniques is called artificial neural networks. In project I of this PhD thesis, artificial neural networks have been used to predict peptide-MHC binding affinities. In this project, the main task was to train the neural networks to learn the underlying mechanism for peptide-MHC binding to MHC class II molecules.

Artificial neural networks

Artificial neural networks are multi-layered networks, capable of learning patterns within a given dataset [41]. The artificial neural network architecture consists of an input layer, one or more hidden layers and an output layer (See Figure 7 A). Each layer consists of nodes connected by weights. Most nodes are fully connected, meaning that there is a link between all the nodes. The only exception is the bias node, which is only connected to the next layer in the network. The output value for each neuron in the network is calculated by a weighted sum of the input values plus the bias, after which this value is reshaped using an activation function (See Figure 7 B). Multiple activation functions exist, but the sigmoid activation function is the most commonly used for artificial neural networks (see Figure 7 C).

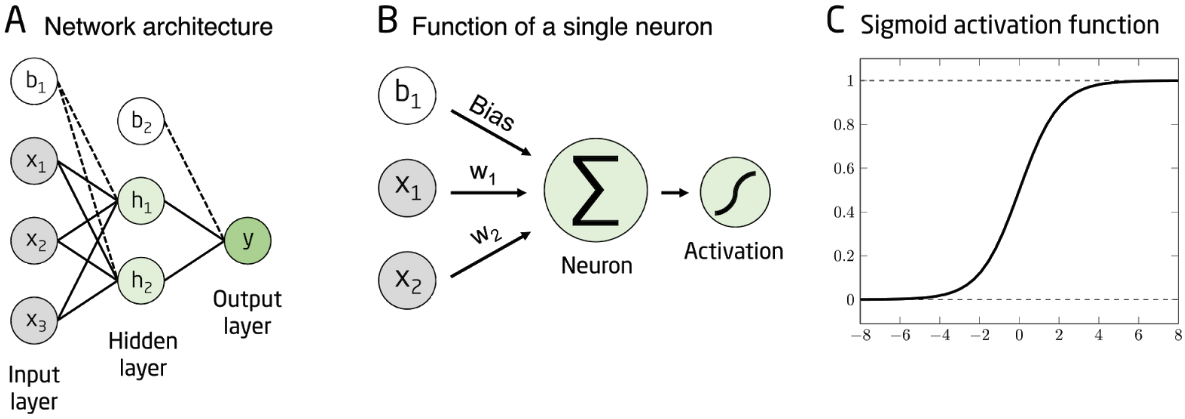


Figure 7: Illustration of a simple neural network and its activation function. **A)** Example of a neural network architecture. **B)** Function of a single neuron. **C)** The sigmoid activation function.

When training the neural network, the main objective is to optimize the network weights, so that the network can predict the correct output value given any input. This training process consists of two steps: i) Forward propagation, which generates a prediction based on an input and ii) backward propagation, which adjusts the weights of the network (see Figure 8).

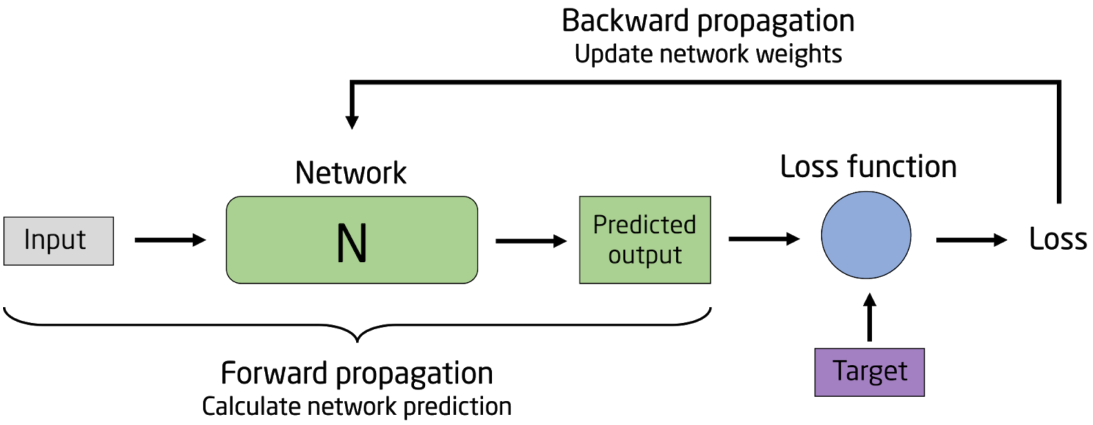


Figure 8: Principle of neural network training. Each iteration of the training consists of two steps: Forward- and backward propagation. In forward propagation, the network predicts an output based on a specific input. The predicted output is then compared to the target and a loss function quantifies the difference between these two. Based on the loss, the backward propagation is performed to adjust the network weights.

In the forward propagation step, a random input variable from the dataset is passed through the network to calculate the corresponding network prediction. After this the backward propagation is performed to adjust the weights of the network based on the error between the predicted output and the true value (called the target). The error between the network prediction and the target is calculated using a loss function and the weights are adjusted using a weight optimization algorithm. In neural networks, the most common loss function is the mean squared error function, and the most common weight optimization method is called stochastic gradient descent (SGD) [42]. The gradient over all network weights is calculated based on the error between the network prediction and the target value, and the weights are then adjusted based on the gradient. In neural networks, gradients are usually multiplied by a learning rate which controls the size of the gradients.

In the last decade, many other network types have been developed including recurrent neural networks (RNNs) [43, 44], convolutional neural networks (CNNs) [45] and generative adversarial networks (GANs) [8]. In project IV we used a GAN to predict the structure of CDR3 loops, and an introduction to this type of network is therefore given in the following section.

Generative adversarial networks

A generative adversarial network (GAN) essentially consists of two networks, a Generator and a Discriminator, trying to outsmart one another [8]. The Generator produces artificial samples from random noise, while the Discriminator determines if a given sample is real or fake (see Figure 9). If the Discriminator makes a correct prediction, the Generator network is updated in order to generate better fake samples which will be able to fool the Discriminator. If the Discriminator prediction is incorrect, the Discriminator network is updated to avoid similar mistakes in the future. This process is performed iteratively until the Generator has learned to consistently generate samples which look real enough to fool the Discriminator.

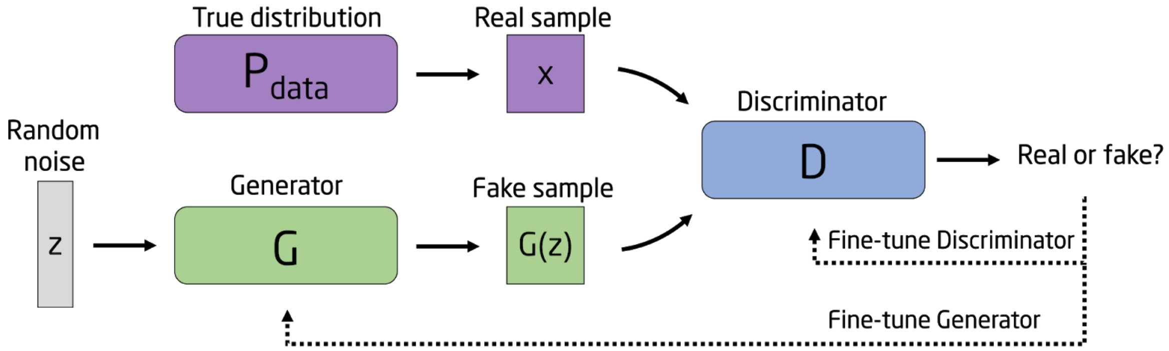


Figure 9: The general principle of conditional generative adversarial networks. The Generator (G) attempts to fool the Discriminator (D) by generating realistic looking samples, while the Discriminator tries to figure out if a given sample is real or fake.

Challenges with GAN and suggested improvements

In the traditional GAN the Generator is trained to produce samples which look real. These samples may belong to different classes. For instance, the GAN may be designed to generate images of handwritten numbers between 0 and 9, in which case the numbers represent different classes. For traditional GANs, it is impossible to control which class the produced samples should belong to. This problem can be solved by using a conditional generative adversarial network (cGAN), where the Generator is trained using an additional input in the form of a class label, which allows the Generator to produce samples belonging to a specific class [46]. The principles from the conditional GANs were used in project IV, where the protein sequence was used as class label to generate loop structures for a specific sequence.

Another problem with traditional GANs, is that the Generator has a tendency to get stuck during training, after which it will start producing samples which are either identical or have limited diversity, regardless of the input. This problem is typically referred to as mode collapse. Furthermore, traditional GANs are also highly sensitive to the choice of network architecture and hyperparameters. This makes it hard to train successful models where the two networks converge.

The last problem with the traditional GANs is that the values from the loss function is not easy to interpret and it does not correlate with the quality of the generated samples. Understanding the progress of training therefore requires to save samples during training, after which the quality is usually determined by visual inspection.

There have been many attempts to solve the aforementioned problems and some of the most successful are called Wasserstein GAN (WGAN) [47] and WGAN with gradient penalty (WGAN-GP) [48].

The main idea behind the WGAN is to implement a new loss function, which has a smoother gradient. Arjovsky *et al.* [47] used the so-called Wasserstein distance to calculate the loss. With this the authors showed that a WGAN should theoretically reduce the risk of mode collapse, while providing meaningful learning curves which can be used for debugging and finding the best hyperparameters. The Wasserstein distance is defined through an optimization over a set of functions constrained to be 1-Lipschitz. Since a function which is 1-Lipschitz has a limited rate of change, this constraint ensures the stability of the WGAN. While it is an unsolved problem to perform the optimization over all 1-Lipschitz functions, there are several ways to implement the constraint approximately. WGANs use weight clipping to restrict the maximum weight value in the network.

Even though WGAN solved many of the original problems of GAN, they are still highly sensitive to the choice of hyperparameters, which makes it hard to optimize them during training and get the network to converge. Gulrajani *et al. [48]* found that this issue is often caused by the weight clipping, and they proposed a new network called WGAN-GP, where the 1-Lipschitz constraint is imposed by penalizing the norm of the gradient instead. This optimization of the loss function dramatically improved the stability of learning, and made it less sensitive to the choice of network architecture and hyperparameters, while still minimizing the problem with mode collapse. Due to this, we chose to use WGAN-GP in project IV.

Performance measures for Machine Learning

There are many different measurements used to estimate the performance of machine learning models and neural networks, such as Accuracy (ACC), Pearson Correlation Coefficient (PCC) and Matthew Correlation Coefficient (MCC).

Area under the ROC curve

One of the most common measurements for model evaluation within the field of bioinformatics is the area under the ROC curve (AUC) [49]. The Receiver Operating Characteristic (ROC) curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR), as seen in Figure 10.

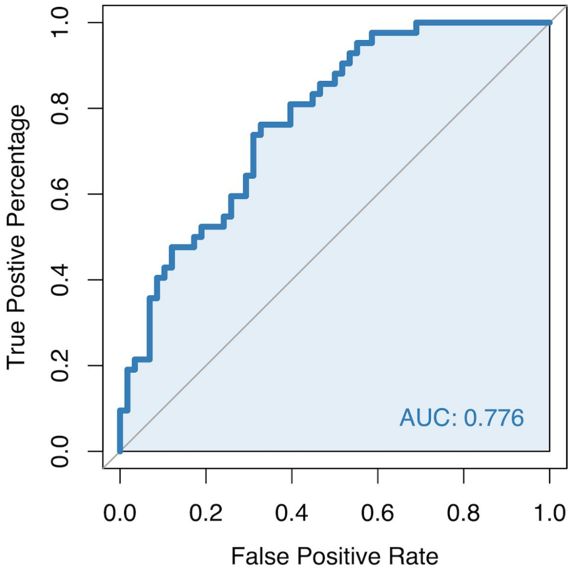


Figure 10: Illustration of the general principle behind calculating the AUC using the ROC curve. The area under the ROC curve is highlighted in light blue and the AUC is shown in the lower right corner of the plot. The diagonal line illustrates a randomly performing model.

After plotting the ROC curve the AUC is calculated as the area under the curve. An AUC of 1 indicates that the model has a perfect performance, where all the predictions are classified correctly, whereas a model with an AUC of 0.5 has a random performance.

Frank score

To investigate a methods ability to predict T-cell epitopes a score known as Frank is usually used [18]. Frank is calculated by first extracting the source protein from which the T-cell epitope was obtained. The sequence from the source protein is then used to generate possible peptides with the same length as the epitope. A machine learning model is used to rank each of these peptides based on their predictions and Frank is then the relative number of predictions with a score higher than the true T-cell epitope. In this way, a perfect prediction would give a Frank score of 0 as the T-cell epitope in this case is ranked at the top of the list, while a Frank score of 0.5 would correspond to a random prediction.

Structural modeling

When the structure of a protein is not known, there are multiple methods for building a structural model using only the information contained within the amino acid sequence. The most successful method for building structural models is template-based modeling, which uses experimentally determined structures as templates for modeling the structure of a target protein. This method is based on the assumption that the protein structure is more conserved than the protein sequence [50, 51].

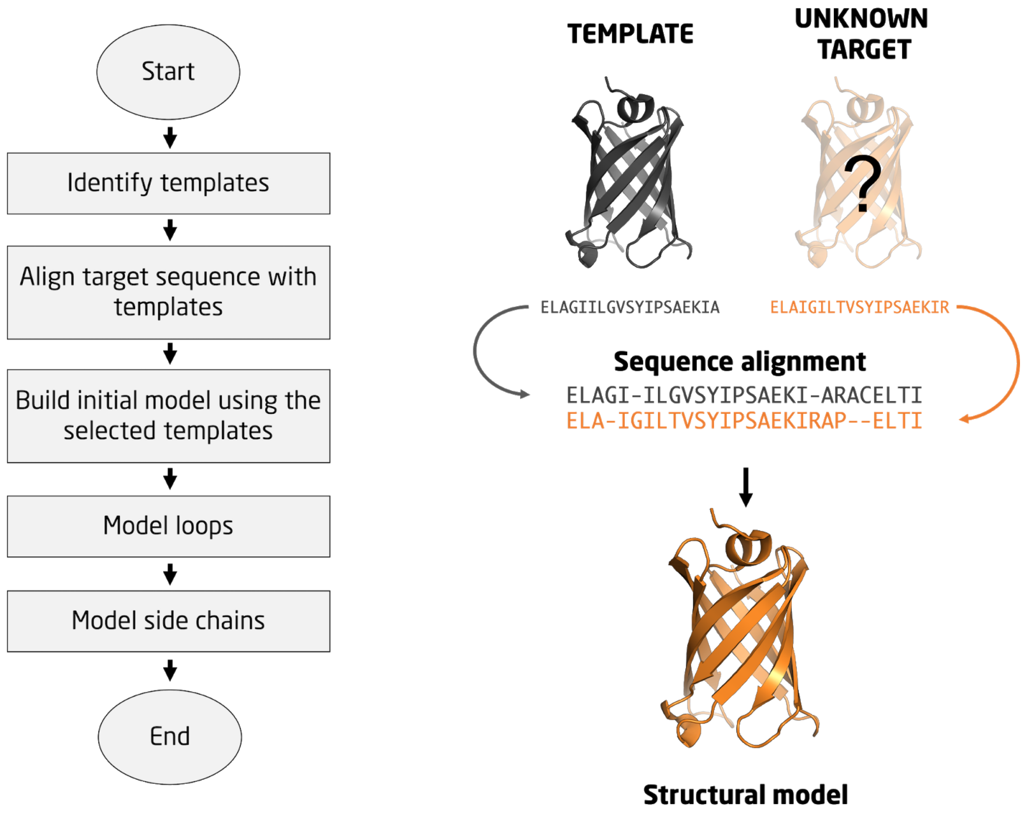


Figure 11: Illustration of the general principles of template-based methods for structural modeling. At first, structural templates of experimentally determined structures of proteins related to the target sequence are identified. Secondly, a target-template alignment is constructed and the structural framework of the model is build using this alignment. Finally, missing regions are generated and the side-chain atoms are added.

Template-based methods have five essential steps: i) Template selection, ii) target-template alignment, iii) model construction, iv) loop modeling and v) side chain modeling [52]. Figure 11 illustrates the general principles of template-based modeling.

In the first step the sequence from the target protein is used to identify experimentally determined structures which can be used as templates. In the second step, the target sequence is aligned to the sequences from the identified templates. In the third step the structural frameworks of the model is built by copying the aligned regions or by satisfying the spatial restraints from the templates. In the final two steps the unaligned loop regions are generated and the side-chain atoms are added.

The most commonly used tool for generating structural models using template-based methods are HHpred [53], SWISS-MODEL [54], I-TASSER [55] and MODELLER [56].

The main problem with template-based modeling methods is that they rely on the existence of good templates. When this condition is not met, alternative methods, such as *ab initio* modeling can be used, but a common limitation of these methods is that they are both time consuming and usually less accurate than template-based methods [52].

The progress of the different template-based and *ab initio* methods is evaluated biennial in the Critical Assessment of protein Structure Prediction (CASP) experiment [57].

Structural modeling performance measurements

The structural similarity between proteins is mostly determined using the root mean square deviation (RMSD) [58] and/or the template modeling score (TM-score) [59].

The RMSD is a measure of the average distance between the atoms of two structurally aligned proteins, and can be calculated using either the C_α atoms, the backbone heavy atoms C, N, O, and C_α or all atoms [58]. The RMSD value of identical structures is zero and as the values increase the two structures become more different. While RMSD is a commonly used measurement for structural similarity, it has two limitations [60]. The first limitation is that the RMSD is dominated by the largest deviation and it might therefore overlook substructural similarities. This can happen if two structures are identical with the exception of a single loop or flexible N- or C-terminus, as this would result in large RMSDs. The second limitation is that the RMSD is dependent on the length of the protein sequence, and it is therefore difficult to compare structures of different lengths. This means that there is no universal threshold which can be used for quality classification.

In comparison to the RMSD, the TM-score is a length-independent metric developed for measuring structural similarity for models generated using template-based methods and the experimentally found structure. The TM-score is calculated using the following formula:

$$\text{TM-score} = \text{Max}\left[\frac{1}{L_N}\sum_{i=1}^{L_T}\frac{1}{1+\left(\frac{d_i}{d_0}\right)^2}\right]$$

where L_N is the number of residues the native structure, L_T is the number of residues in the template structure, d_i is the distance between the i'th pair of aligned residues and $d_0 = 1.24\sqrt[3]{L_N - 15} - 1.8$ is a scale to normalize the TM-score [59].

The TM-score ranges between 0 and 1, where a TM-score of 1 indicates a perfect match between two structures. A TM-score below 0.2 corresponds to randomly choosing an unrelated protein and a TM-score higher than 0.5 implies that the two structures roughly have the same structural fold.

When evaluating the performance of structural models both the RMSD and TM-score is used to measure the structural difference between a model and its naive structure, which is usually determined by X-ray crystallography.

Chapter 2: Paper I

In the last decade, machine learning methods for predicting peptide binding to MHC class II molecules have vastly improved [6, 7, 61–63]. These methods are trained on data obtained from the Immune Epitope Database (IEDB), which constantly collects and catalogs experimental data from peptide-MHC binding studies. As this database gradually increases, methods tained on the latest data set usually have an improved overall performance and an improved MHC coverage.

In this chapter, we present NetMHCII-2.3 and NetMHCIIpan-3.2, two improved methods for predicting peptide-MHC binding affinity for MHC class II molecules. Both methods were trained using the NNAlign algorithm and are based on ensembles of artificial neural networks. The updated methods were trained using a data set of peptide binding affinity measurements from the IEDB with extended MHC coverage and peptide volume compared to earlier data sets. The updated methods show improved performances for predicting peptide binding affinities and for detecting T-cell epitopes, when compared to the previous versions of the methods.

The tools are available at:
http://www.cbs.dtu.dk/services/NetMHCII/
http://www.cbs.dtu.dk/services/NetMHCIIpan/

Immunology
The Journal of cells, molecules, systems and technologies

British Society for immunology

IMMUNOLOGY ORIGINAL ARTICLE

Improved methods for predicting peptide binding affinity to MHC class II molecules

Kamilla Kjærgaard Jensen,[1] Massimo Andreatta,[2] Paolo Marcatili,[1] Søren Buus,[3] Jason A. Greenbaum,[4] Zhen Yan,[4] Alessandro Sette,[5,6] Bjoern Peters[5,6] and Morten Nielsen[1,2]

[1]*Department of Bio and Health Informatics, Technical University of Denmark, Lyngby, Denmark,* [2]*Instituto de Investigaciones Biotecnológicas, Universidad Nacional de San Martín, Buenos Aires, Argentina,* [3]*Department of Immunology and Microbiology, Faculty of Health Sciences, University of Copenhagen, Copenhagen, Denmark,* [4]*Bioinformatics Core Facility, La Jolla Institute for Allergy and Immunology, La Jolla, CA,* [5]*Division of Vaccine Discovery, La Jolla Institute for Allergy and Immunology, La Jolla, CA,* and [6]*Department of Medicine, University of California San Diego, La Jolla, CA, USA*

doi:10.1111/imm.12889
Received 13 November 2017; revised 19 December 2017; accepted 22 December 2017.

Correspondence: Morten Nielsen, Department of Bio and Health Informatics, Technical University of Denmark, DK-2800 Lyngby, Denmark. Email: mniel@bioinformatics.dtu.dk
Senior author: Morten Nielsen

Summary

Major histocompatibility complex class II (MHC-II) molecules are expressed on the surface of professional antigen-presenting cells where they display peptides to T helper cells, which orchestrate the onset and outcome of many host immune responses. Understanding which peptides will be presented by the MHC-II molecule is therefore important for understanding the activation of T helper cells and can be used to identify T-cell epitopes. We here present updated versions of two MHC–II–peptide binding affinity prediction methods, NetMHCII and NetMHCIIpan. These were constructed using an extended data set of quantitative MHC–peptide binding affinity data obtained from the Immune Epitope Database covering HLA-DR, HLA-DQ, HLA-DP and H-2 mouse molecules. We show that training with this extended data set improved the performance for peptide binding predictions for both methods. Both methods are publicly available at www.cbs.dtu.dk/services/NetMHCII-2.3 and www.cbs.dtu.dk/services/NetMHCIIpan-3.2.

Keywords: affinity predictions; immunogenic peptides; MHC binding specificity; peptide–MHC binding; T-cell epitope.

Introduction

Major histocompatibility complex class II (MHC-II) molecules are found on the surface of antigen-presenting cells where they present peptides derived from extracellular proteins to T helper cells.[1] Many peptide–MHC complexes are presented on the surface of antigen-presenting cells, but only peptides recognized by T-cell receptors will trigger an immune response, and are referred to as T-cell epitopes. Identifying T-cell epitopes is important for the general understanding of cellular immunity and the design of peptide-based diagnostics, therapeutics and vaccines.[2] The MHC-II molecule is a heterodimeric glycoprotein that consists of an α-chain and a β-chain. In humans, these two chains are encoded in the human leucocyte antigen (HLA) gene complex in one of three loci called HLA-DR, -DP and -DQ.[3] In mice, the MHC-II chains are encoded in the histocompatibility 2 (H-2) locus. Each locus is comprised of many different allelic variants, which makes the MHC-II molecule highly

Abbreviations: AUC, area under the receiver operating characteristics curve; H-2, histocompatibility 2; HLA, human leucocyte antigen; IEDB, Immune Epitope Database; LOMO, leave-one-molecule-out; MHC-II, MHC class II; MHC-I, MHC class I; MHC, major histocompatibility complex; PFR, peptide flanking regions; UPGMA, unweighted pair group method with arithmetic mean polymorphic.[4] Peptides presented by the MHC-II molecule bind to a binding groove formed by residues of the MHC α- and the β-chains. The peptide binding groove is open at both ends and therefore allows binding of peptides with different lengths.[5] Even though the MHC-II molecule can accommodate peptides of variable lengths the most abundant peptides found in nature are between 13 and 25 residues long.[6] The part of the peptide ligand that primarily interacts with the MHC binding groove is called the peptide binding core and is usually nine amino acids long[7] with anchor residues at positions P1, P4, P6 and P9.[8] The peptide–MHC binding affinity is primarily determined by the amino acid sequence of the peptide binding core. However, it has been shown that peptide flanking regions (PFRs) on either side of the binding core affect peptide–MHC binding and, thereby ultimately also influence the peptide immunogenicity.[7,9]

There are therefore many factors that make it difficult to predict peptide binding affinities to MHC-II molecules, including the polymorphic nature of MHC-II molecules, the variations in peptide length, the influence of the PFRs and the identification of the correct peptide binding core. All these factors complicate the task of predicting peptide binding affinities to MHC-II molecules; most methods therefore still have a low performance compared with MHC class I (MHC-I) peptide binding prediction methods. Earlier work has demonstrated that the prediction performance of both NetMHCII and NetMHCIIpan is dependent on the amount of peptide binding data[10,11] and one would therefore expect the two methods to improve in performance if retrained on an extended peptide binding data set. We have here investigated if this is indeed the case.

Identifying T-cell epitopes is difficult because of the large diversity in potentially binding peptides. However, as peptide-MHC binding is a prerequisite for T-cell immunogenicity, multiple studies have shown that there is a strong correlation between MHC peptide binding strength and peptide immunogenicity.[12–14] It is therefore desirable to have accurate and reliable peptide binding affinity prediction methods that can be used for *in silico* screening peptides with the purpose of identifying T-cell epitopes that match MHC-II molecules in a given host. Given this, many different methods have been developed, including NetMHCII,[15] NetMHCIIpan,[16] TEPITOPE,[17] TEPITOPEpan,[18] PROPRED,[19] RANKPEP[20,21] and SVRMHC.[22] Both NetMHCII[15] and NetMHCIIpan[16] have been shown to be among the best methods for predicting binding affinities to MHC-II molecules.[2,8,23] These two methods are trained using the NNAlign framework[15,24,25] and are based on ensembles of artificial neural networks that are trained on quantitative peptide binding affinity data from the Immune Epitope Database (IEDB).[26] One of the main differences between NetMHCII and NetMHCIIpan is that NetMHCII is a collection of individual networks for each MHC molecule whereas NetMHCIIpan contains a single universal network that can predict peptide binding affinities for all MHC molecules of known protein sequence.

NetMHCII and NetMHCIIpan predict peptide binding affinities to MHC-II molecules covering HLA-DR, HLA-DQ, HLA-DP and H-2 mouse molecules. The main difference between the two methods is that NetMHCII only predicts peptide binding affinities to MHC molecules for which it has been trained, whereas NetMCHIIpan can predict peptide binding affinities to any MHC molecule with a known protein sequence. As mentioned above there is a strong correlation between MHC binding strength and peptide immunogenicity and the two methods have been used extensively as a guide to identify T-cell epitopes that can be used in the design of peptide-based diagnostics, therapeutics and vaccines.

In this paper, we present updated versions of our binding affinity prediction methods, NetMHCII and NetMHCIIpan, trained on an extended data set of > 100 000 quantitative peptide binding measurements from IEDB,[26] covering 36 HLA-DR, 27 HLA-DQ, 9 HLA-DP, as well as 8 mouse MHC-II molecules. We then evaluate the performance of these new versions using a set of large-scale benchmarks to investigate how the extended data set improves the predictive performance of the two methods.

Materials and methods

Data sets

The data set used to generate the new versions of NetMHCII and NetMHCIIpan contains peptide–MHC II binding affinities retrieved from the IEDB (www.iedb.org) in 2016. All data points are experimental IC_{50} binding values, which were log-transformed to fall in the range between 0 and 1 using the relation $1-\log(IC_{50}\ \text{nM})/\log(50\ 000)$ as explained by Nielsen *et al.*[27]. The 2016 data set contains 134 281 data points, covering 36 HLA-DR, 27 HLA-DQ, 9 HLA-DP and 8 H-2 molecules. The data set was split into five groups by clustering the common motif of peptides as described by Nielsen *et al.*[28] and these five groups were used for a five-fold cross-validation. This 2016 data set is publicly available at www.cbs.dtu.dk/suppl/immunology/NetMHCIIpan-3.2. The data set used to develop the previous versions of NetMHCII and NetMHCIIpan is available at www.cbs.dtu.dk/suppl/immunology/NetMHCIIpan-3.0.

A summary of the data included in the 2013 and 2016 data sets is shown in Table 1 and a description of the full 2016 data set is available in the Supplementary material (Table S1).

Network training

The NetMHCII method was implemented as described by Nielsen and Lund[15] and the NetMHCIIpan method was

Table 1. Description of the two MHC class II peptide binding data sets

	Data set 2013	Data set 2016
# Data points	52062	134281
Type of alleles	24 HLA-DR	36 HLA-DR
	6 HLA-DQ	27 HLA-DQ
	5 HLA-DP	9 HLA-DP
	2 H-2	8 H-2

implemented as described by Andreatta *et al.*[16] NetMHCII is an allele-specific method that contains a specific predictor for each MHC molecule in the data set and it can therefore only predict binding affinities for MHC molecules found in the training data, whereas NetMHCIIpan is a pan-specific method that can make predictions for any MHC molecule with a known protein sequence. To achieve its pan-specificity, NetMHCIIpan incorporates information about the MHC-II molecule, using a pseudo sequence consisting of residues that are considered important for peptide binding. This pseudo sequence is constructed using the method described by Karosiene *et al.*[11] and is composed of 34 residues: 15 from the α-chain and 19 from the β-chain. Both methods were trained using a five-fold cross-validation set-up. For each fold, we generate a network ensemble of individual networks trained without early stopping for 500 cycles with 10, 15, 40 and 60 hidden neurons using 10 different initial configurations, generating a total of 40 networks. This was done for each of the five training/test set combinations leading to a total of 200 networks. The peptide and the MHC pseudo sequence were encoded using the BLOSUM50 matrix and the PFR was encoded using the average BLOSUM scores on a maximum window of three amino acids at either end of the binding core.[29] For each peptide core, the input to the neural network therefore consisted of the peptide core ($9 \times 20 = 180$ inputs), the PFRs ($2 \times 20 = 40$ inputs), the peptide length (2 inputs), the length of the C-terminal and N-terminal PFRs ($2 \times 2 = 4$ inputs), resulting in a total of 226 input values for NetMHCII and 906 for NetMHCIIpan (an additional $34 \times 20 = 680$ input values from the pseudo sequence).

Binding core predictions

To improve the binding core predictions, we include the offset correction step to both NetMHCII and NetMHCIIpan. We followed the procedure described by Andreatta *et al.*[16] and we evaluated the performance of this offset correction using the benchmark data set of 51 crystal structures of peptide–MHC-II complexes.

Performance measures

The predictive performance of the different methods was measured using the area under the receiver operating characteristics curve (AUC). To classify peptides into binders and non-binders, a binding threshold of 500 nM was used, classifying all peptides with an IC_{50} binding value < 500 nM as binders. All performance values shown in this paper are averages of the AUC performance per MHC molecule using only molecules with more than 20 peptides and at least four binders.

Leave-one-molecule-out network training

To assess the predictive performance of NetMHCIIpan in the situation where a molecule is not part of the training data, a leave-one-molecule-out (LOMO) approach was applied.

To estimate LOMO performance for MHC molecule X, the NetMHCIIpan networks were trained using the five-fold cross-validation set-up from above. In the LOMO cross-validation set-up all binding data from molecule X were removed from the training sets and all test sets only include binding data from molecule X. This set-up ensures that the method is trained without peptides binding to molecule X and it can therefore be used to evaluate the ability of the method to predict peptide binding of uncharacterized MHC-II molecules.

Nearest neighbour distance calculation

The nearest neighbour distance is estimated from the alignment score of the HLA pseudo sequences using the relation $d = (s(\text{A,B}))/\left(\sqrt{s(\text{A,A}) \cdot s(\text{B,B})}\right)$. In this equation s(A,B) is the BLOSUM50 alignment score between the pseudo sequences for MHC molecules A and B, respectively.[29] Nearest neighbours are found from the subset of molecules characterized with at least 50 data points and at least 10 binders.

Sequence logos

Sequence logos were constructed from the predicted binding cores of the top 1% strongest predicted binders using 200 000 natural random 15-mer peptides and was visualized using Seq2Logo[30] with default settings.

Generation of HLA-II distance trees

The HLA-II distance tree was generated for each of the HLA-DR, -DQ and -DP molecules in our data set using MHCCluster.[31] To make the tree we first predicted the binding affinity for 200 000 natural random 15-mer peptides using the new version of NetMHCIIpan. We then used MHCCluster to find the functional similarity between any two MHC molecules. MHCCluster calculates the similarity between two MHC molecules by correlating the union of the predicted top 10% strongest binding peptides. Using the bootstrap method in

MHCCLUSTER we generated 100 distance matrices and converted these to distance trees using the unweighted pair group method with arithmetic mean clustering. These trees were then combined into a consensus tree and visualized in SPLITSTREE.[32] Sequence logos were constructed as explained above.

T-cell epitope benchmark

A set of MHC-II restricted T-cell epitopes identified by multimer/tetramer staining assays was downloaded from IEDB. Only fully typed restrictions were included; that is, fully typed α- and β-chains for HLA-DQ and HLA-DP, and a fully typed β-chain for HLA-DR (where the α-chain is invariant). Epitopes with non-natural amino acids were excluded. Also, epitopes with identical match to the peptides in the training data were excluded. The source protein sequence for each epitope was identified by mapping the annotated IEDB protein ID to the NCBI protein database. The final validation data set consisted of 1698 epitopes, restricted to 33 distinct MHC-II molecules. For performance evaluation, the epitope source protein was split into overlapping peptides of the length of the epitope, and AUC and Frank values were calculated for each epitope–MHC pair annotating the epitopes as positive and all others as negatives. Here, Frank is the ratio of the number of peptides with a prediction score higher than the positive peptide to the number of peptides contained within the source protein. Hence, the Frank value is 0 if the positive peptide has the highest prediction value of all peptides within the source protein and a value of 0·5 in cases in which an equal number of peptides has a higher and lower prediction value compared with the positive peptide.

Results

Comparing NetMHCII and NetMHCIIpan on a shared evaluation set

Using the data set from 2016, we retrained NetMHCII[15] and NetMHCIIpan[11] using a five-fold cross-validation setup to generate two new versions of these methods, named NetMHCII-2.3 and NetMHCIIpan-3.2. We then investigated how these new versions performed compared with the previous versions, which are NetMHCII-2.2 and NetMHCIIpan-3.1, trained on the 2013 data set. To make the comparison, we used the same fivefold cross-validation set-up and compared peptide data points in common between the 2013 and 2016 data sets. The result from this analysis in shown in Table 2.

The new versions of NetMHCII and NetMHCIIpan improved performance compared with the older versions (Table 2); but the performance gain was not statistically significant ($P > 0{\cdot}1$ in both cases). Another interesting point is that the allele-specific NetMHCII-2.3 obtained a higher average performance than the pan-specific NetMHCIIpan-3.2, but this effect will be discussed later.

Performance of NetMHCIIpan on new data points for common MHC molecules

Using the five-fold cross-validation setup, we then evaluated the performance of the two versions of NetMHCII and NetMHCIIpan using only the subset of new peptides for the MHC molecules common between the old and new data sets. The result of this analysis is shown in Table 3 and it demonstrates a significant gain in predictive performance of the new versions (NetMHCII, $P < 0{\cdot}001$ and NetMHCIIpan, $P < 0{\cdot}0003$, using paired *t*-test). This result underlines the importance of expanding the size of the training data even for previously characterized MHC molecules. [Correction added on 02 April 2018, after first online publication: In the preceding sentence, $P < 0{\cdot}005$ and $P < 0{\cdot}001$ was corrected to $P < 0{\cdot}001$ and $P < 0{\cdot}0003$ respectively.]

Binding core predictions

We evaluated the accuracy for binding core identification of the two updated MHC-II binding prediction methods on the data set of peptide–MHC crystal structures described by Andreatta *et al.*[16] Overall we find that (i) the inclusion of the offset correction has a substantial impact on the accuracy of binding core identification for both methods, and (ii) the overall accuracy of both methods is improved compared with the earlier version. For details see the Supplementary material (Table S2).

Performance of a consensus method

For predicting binding affinities to MHC-I, it has been shown that a simple combination of the predictions from NetMHC[27] and NetMHCpan[10] gives a higher performance than using each method individually.[33] We therefore made a similar combination of the predictions from NetMHCII-2.3 and NetMHCIIpan-3.2 to investigate if the performance could be improved for MHC-II using this consensus approach. In the consensus method, we use an average of the prediction scores (values between 0 and 1) from NetMHCII-2.3 and NetMHCIIpan-3.2 to define the consensus method. The result of this analysis is shown in Fig. 1 and detailed performance values are found in the Supplementary material (Table S3). Figure 1(a) shows that the combination of NetMHCII-2.3 and NetMCHIIpan-3.2 has a significantly improved performance compared with each individual method and Fig. 1(b) shows that NetMHCIIpan-3.2 outperforms NetMHCII-2.3, especially for MHC molecules where only a few peptides are found in the data set.

Table 2. Comparing predictions from the old and the new versions of NetMHCII and NetMHCIIpan trained using a fivefold cross-validation on the set of data points common between the two data sets

Molecule	#Peptides	#Binders	NetMHCII-2.2	NetMHCII-2.3	NetMHCIIpan-3.1	NetMHCIIpan-3.2
DRB1_0101	2754	2635	0·817	**0·822**	0·828	**0·830**
DRB1_0301	1403	379	**0·832**	0·826	0·829	**0·835**
DRB1_0401	1639	695	**0·801**	0·791	**0·804**	0·798
DRB1_0404	542	331	**0·783**	0·768	**0·813**	0·810
DRB1_0405	1438	595	**0·862**	0·860	**0·852**	0·844
DRB1_0701	1619	806	**0·858**	0·857	0·852	**0·857**
DRB1_0802	1310	400	0·757	**0·767**	**0·753**	0·749
DRB1_0901	841	560	0·746	**0·761**	0·777	**0·779**
DRB1_1101	1604	730	0·876	0·876	0·875	**0·876**
DRB1_1302	1351	463	0·811	**0·823**	0·801	**0·810**
DRB1_1501	1601	672	0·818	**0·820**	0·817	**0·831**
DRB3_0101	1266	267	0·835	**0·846**	**0·835**	0·824
DRB4_0101	1329	467	0·840	**0·841**	**0·832**	0·817
DRB5_0101	1606	765	**0·852**	0·847	**0·855**	0·846
H-2-IAb	525	125	0·850	**0·857**	0·849	**0·868**
H-2-IAd	100	24	0·718	**0·809**	0·734	**0·808**
HLA-DPA10103-DPB10401	1075	458	0·957	**0·960**	0·956	**0·961**
HLA-DPA10201-DPB10101	1180	558	0·949	0·949	**0·949**	0·948
HLA-DPA10201-DPB10501	1114	415	**0·957**	0·954	**0·949**	0·948
HLA-DPA10301-DPB10402	1193	498	**0·958**	0·957	**0·957**	0·952
HLA-DQA10101-DQB10501	990	246	0·856	**0·890**	0·834	**0·857**
HLA-DQA10102-DQB10602	1121	503	0·838	**0·901**	0·877	**0·887**
HLA-DQA10301-DQB10302	1461	330	**0·824**	0·820	**0·796**	0·774
HLA-DQA10401-DQB10402	1436	516	0·919	**0·923**	**0·915**	0·903
HLA-DQA10501-DQB10201	1386	477	0·898	**0·901**	**0·886**	0·883
HLA-DQA10501-DQB10301	1274	530	**0·893**	0·873	**0·881**	0·860
Average			0·856	0·863	0·856	0·858

For each MHC molecule, we show the total number of peptides, the number of binders, the AUC performance. The different methods included are the NetMHCII and NetMHCIIpan methods training on the original 2013 data set (versions 2.2 and 3.1), and the versions of the two methods trained on the extended 2016 data set (versions 2.3 and 3.2). The highest performance for NetMHCII and NetMHCIIpan is highlighted in bold.

Performance of NetMHCIIpan for previously uncharacterized MHC molecules

For NetMHCIIpan, we also tested the performance on MHC molecules that were not part of the 2013 data set (see Table 4). As expected, we observed that the new version of NetMHCIIpan had a significant increase in the predictive performance when compared with the previous version of NetMHCIIpan ($P = 3{\cdot}6 \times 10^{-5}$, using a paired *t*-test); this result therefore demonstrates the importance of expanding the allotypic coverage of the training data.

Leave-one-molecule-out performance

The pan-specific method is capable of making predictions for uncharacterized MHC molecules, so to assess the predictive performance of the NetMHCIIpan method in these situations we conducted a LOMO experiment. In the LOMO, the binding data for the MHC molecule in question were excluded from training and the resulting model was then evaluated using only binding data for the MHC molecule in question (for details see the Materials and methods). The LOMO experiment was made for all MHC molecules shared between the 2013 and the 2016 data sets, and the performance was evaluated on peptides shared between the two data sets. The result of this LOMO benchmark is shown in Table 5, together with the pseudo distances of the MHC molecule to each of the two training data sets estimated from the nearest neighbour sequence similarity as described in Materials and methods.

Table 5 shows an increased performance for NetMHCIIpan-3.2-LOMO compared with netMHCIIpan-3.1-LOMO. This gain is in general most pronounced for the MHC molecules that share a decrease in the pseudo sequence distance.

To further investigate this last observation, the LOMO performance evaluation was extended to include all MHC molecules in the 2016 data set. The result from this analysis is shown in Fig. 2 with a scatterplot of the relationship between the distance to the nearest neighbour in the training data set and the LOMO performance. The complete data used to create Fig. 2 can be found in Table S4. The figure shows that the HLA-DQ and the HLA-DP molecules have close nearest neighbours whereas the HLA-DR and H-2 molecules tend to have more distant

Table 3. Comparing predictions from the old (versions 2.2 and 3.1), and the new version (versions 2.3 and 3.2), of NetMHCII and NetMHCpan using the fivefold cross-validation setup and evaluating on the subset of new peptides using only MHC molecules shared between the 2013 and 2016 data sets

Allele	#Peptides	#Binders	NetMHCII-2.2	NetMHCII-2.3	NetMHCIIpan-3.1	NetMHCIIpan-3.2
DRB1_0101	7658	3741	**0·850**	0·815	**0·836**	0·823
DRB1_0301	3949	1078	0·799	**0·813**	0·779	**0·812**
DRB1_0401	4678	2327	0·771	**0·798**	0·770	**0·811**
DRB1_0404	3115	1521	0·710	**0·788**	0·761	**0·810**
DRB1_0405	2524	1059	0·798	**0·828**	0·809	**0·817**
DRB1_0701	4706	2650	0·822	**0·882**	0·825	**0·880**
DRB1_0802	3155	1636	0·797	**0·845**	0·825	**0·853**
DRB1_0901	3477	1604	0·842	**0·844**	0·833	**0·840**
DRB1_1101	4441	1937	0·826	**0·865**	0·820	**0·862**
DRB1_1302	3126	1786	0·853	**0·907**	0·860	**0·907**
DRB1_1501	3249	1435	0·806	**0·840**	0·817	**0·836**
DRB3_0101	3367	1148	0·898	**0·913**	0·898	**0·906**
DRB4_0101	2632	1073	0·796	**0·834**	0·804	**0·822**
DRB5_0101	3519	1665	0·836	**0·851**	0·841	**0·851**
H-2-IAb	1268	306	**0·936**	0·894	**0·919**	0·902
H-2-Iad	674	297	0·762	**0·819**	0·799	**0·820**
HLA-DPA10103-DPB10201	782	140	**0·968**	0·909	**0·954**	0·916
HLA-DPA10103-DPB10401	1650	328	0·887	**0·900**	0·885	**0·898**
HLA-DPA10201-DPB10101	1267	301	0·819	**0·830**	0·828	**0·845**
HLA-DPA10201-DPB10501	1356	298	0·849	**0·858**	0·817	**0·858**
HLA-DPA10301-DPB10402	1448	423	0·839	**0·840**	0·841	**0·844**
HLA-DQA10101-DQB10501	1956	569	**0·930**	**0·930**	**0·922**	0·920
HLA-DQA10102-DQB10602	1626	753	0·856	**0·913**	0·880	**0·902**
HLA-DQA10301-DQB10302	1650	238	0·850	**0·868**	**0·838**	0·832
HLA-DQA10401-DQB10402	1454	412	0·781	**0·858**	0·781	**0·857**
HLA-DQA10501-DQB10201	1511	397	0·831	**0·874**	0·833	**0·871**
HLA-DQA10501-DQB10301	2311	1282	0·909	**0·944**	0·921	**0·943**
Average			0·838	0·861	0·841	0·861

For each MHC molecule, we show the total number of peptides, the number of binders and the AUC performance for the different versions. Highlighted in bold is the highest performance between the two NetMHCII and NetMHCIIpan methods.

[Correction added on 02 April 2018, after first online publication: Table 3 has been updated in this version.]

neighbours. This figure also demonstrates a weak but statistically significant ($P = 0{\cdot}04$ with exact permutation test) correlation between the LOMO performance and the distance to the nearest neighbour in the training data. This is in agreement with earlier findings for both MHC-I and MHC-II molecules[10,11] and shows how the predictive performance of the pan-specific method depends on the distance to the nearest neighbour.

Distance tree for HLA molecules

Having arrived at the final retrained versions of NetMHCIIpan, we next use the MHCCluster method[31] to evaluate the similarities of binding motifs between the HLA molecules included in the 2016 training data. In short, the MHCCluster method estimates the similarity between two MHC molecules using the correlation between predicted binding values for a large set of random natural peptides. The similarity is 1 if the two molecules have a perfect binding specificity overlap and -1 if the two molecules share no specificity overlap (for details see Materials and methods). Comparing the binding pattern similarity between any two HLA class II molecules in the 2016 training data, we constructed the distance tree shown in Fig. 3. This figure confirms the earlier findings by Karosiene *et al.*:[11] (i) the different loci show limited overlap in binding preference, (ii) HLA-DP is less diverse compared with HLA-DQ and HLA-DR, and (iii) the diversity of HLA-DQ can largely be split into three groups; one with preference for negatively charged amino acids towards the C-terminus, one with a preference for positively charged amino acids towards the C-terminus, and one with a preference for small amino acids at the anchor positions.

T-cell epitope benchmark

We next evaluated the predictive performance of the two NetMHCIIpan methods on an IEDB T-cell epitope data set. We queried the IEDB for MHC-II-restricted epitopes

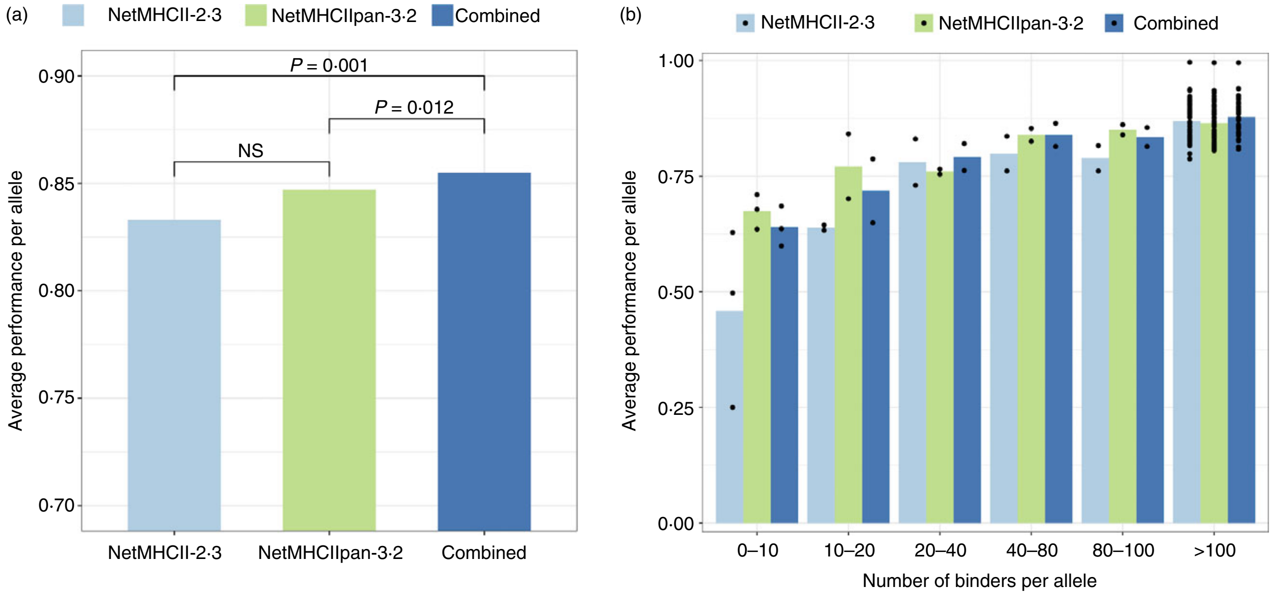


Figure 1. Performance of NetMHCII-2.3 and NetMHCIIpan-3.2 together with the combination method. (a) The average performance per MHC molecule of NetMHCII-2.3, NetMHCIIpan-3.2 and the combination method, including the significance between the methods. *P*-values where found using a paired *t*-test using the predictions per molecule found in Table S3 (see the Supplementary material). (b) The average predictive performance of the MHC molecules in the data set as a function of the number of peptides. [Colour figure can be viewed at wileyonlinelibrary.com]

identified by tetramer/multi-mer staining, which is the reference standard for epitope identification with known MHC restriction. For each epitope–MHC-II pair, we calculated AUC and Frank values for the two NetMHCIIpan methods by predicting binding affinities to the MHC-II restriction element of the epitope for all overlapping peptides with the same length as the epitope in the source protein sequence, annotating the epitope as positive and the remaining peptides as negative. This annotation is very stringent because peptides that share the same ligand binding-core are counted as negatives even though they could be presented by the human MHC molecule; the set-up will therefore most likely underestimate the predictive performance. The details from this analysis are found in Table S5 and the results are summarized in Fig. 4.

The Frank value is 0 if the positive peptide has the highest prediction value of all peptides within the source protein, and a value of 0·5 in cases where an equal number of peptides has a higher and lower prediction value compared with the positive peptide. Figure 4(a) shows that the Frank score for NetMHCIIpan-3.1 is significantly lower than NetMHCIIpan-3.1. It further shows that NetMHCIIpan-3.2 has a median < 0·2 indicating that the positive peptide was found among the top 20% of the peptides from the source protein if sorted on their predicted peptide binding affinity. Figure 4(b) demonstrates a significant improvement in the AUC performance of NetMHCIIpan-3.2 compared with NetMHCIIpan-3.1. We speculate that the gain in predictive performance of NetMHCIIpan-3.2 could be attributed to at least two factors, the inclusion of binding data for additional MHC-II molecules in the training data, and the expansion of the number of data points for MHC-II molecules already included in the old training data. Figure 4(c,d) quantifies that both of these factors indeed contribute to the performance gain. Figure 4(c) shows the performance gain as a function of the change in distance of the query molecule to the nearest neighbour of the training data. From this plot, we see that the gain in predictive performance is related to a decrease in the nearest neighbour distance, and hence directly related to the inclusion of binding data for additional MHC-II molecules in the new data set. Figure 4(d) shows the performance gain as a function of the change in the number of data points between the two data sets used for training. We here only include molecules shared between the two data sets used for training NetMHCIIpan-3.1 and NetMHCIIpan-3.2, as we in the previous analysis demonstrated how the distance to the nearest neighbour influences the performance. Figure 4(d) shows that the gain in performance is correlated to change in the number of data points for the given MHC molecules. This indicates that the performance gain of the new NetMHCIIpan version is also driven by the increase in the number of data points for molecules already included in the 2013 data set. The one data point in Figure 4(c,d) with increased nearest neighbour distance and decreased number of data points corresponds to the HLA-DPA10103-DPB10201 molecule for which faulty data were removed in the 2016 data set.

Table 4. Comparing predictions from the old and the new version of NetMHCIIpan using the fivefold cross-validation setup on the set of MHC molecules found in the 2016 data set but not in the 2013 data set

Molecule	#Peptides	#Binders	NetMHCIIpan-3.1	NetMHCIIpan-3.2
DRB1_0103	42	4	0·664	**0·678**
DRB1_0402	53	19	0·680	**0·701**
DRB1_0403	59	14	0·767	**0·841**
DRB1_0801	937	390	0·839	**0·844**
DRB1_1001	2066	1521	0·907	**0·923**
DRB1_1104	27	5	0·682	**0·791**
DRB1_1301	1034	520	0·727	**0·857**
DRB1_1502	23	7	**0·688**	0·652
DRB1_1602	1699	989	0·827	**0·883**
DRB3_0202	3334	1055	0·789	**0·869**
DRB4_0103	846	525	0·786	**0·841**
H-2-IAk	115	4	0·426	**0·635**
H-2-IAs	190	48	0·438	**0·825**
H-2-IAu	56	22	**0·790**	0·765
H-2-IEd	245	28	0·623	**0·754**
H-2-IEk	68	40	**0·881**	0·853
HLA-DPA10103-DPB10301	1563	575	0·588	**0·902**
HLA-DPA10103-DPB10402	45	9	**0·815**	0·710
HLA-DPA10103-DPB10601	584	282	**0·996**	0·995
HLA-DPA10201-DPB11401	2302	849	0·696	**0·930**
HLA-DQA10102-DQB10501	833	458	0·606	**0·839**
HLA-DQA10102-DQB10502	800	158	0·825	**0·835**
HLA-DQA10103-DQB10603	462	90	0·802	**0·861**
HLA-DQA10104-DQB10503	883	105	0·787	**0·805**
HLA-DQA10201-DQB10202	944	119	0·779	**0·814**
HLA-DQA10201-DQB10301	827	374	0·813	**0·849**
HLA-DQA10201-DQB10303	761	265	0·743	**0·894**
HLA-DQA10201-DQB10402	768	241	0·529	**0·860**
HLA-DQA10301-DQB10301	207	66	0·822	**0·839**
HLA-DQA10303-DQB10402	567	117	0·483	**0·820**
HLA-DQA10501-DQB10302	847	203	0·772	**0·822**
HLA-DQA10501-DQB10303	564	179	0·809	**0·876**
HLA-DQA10501-DQB10402	749	337	0·584	**0·868**
HLA-DQA10601-DQB10402	565	133	0·498	**0·848**
Average			0·719	0·826

For each molecule, we show the total number of peptides, the number of binders and the AUC performance for the two NetMHCIIpan versions. In bold is highlighted the highest performance of the two versions 3.1 and 3.2 of NetMHCIIpan. Highlighted in bold is the highest performance between the two methods.

Discussion

The genomic region encoding the MHC-II molecule is extremely polymorphic comprising several thousand alleles and it is therefore difficult to produce enough experimental data to characterize the peptide binding preference for all existing MHC-II molecules. Because of this, most MHC-II molecules are still only represented with very few or no binding data, limiting the coverage and performance of previous binding affinity prediction methods. We have therefore updated our two binding affinity prediction methods, NetMHCII and NetMHCIIpan using updated and extended data sets. For several large-scale benchmarks, this improved the predictive performance for both methods.

Comparing NetMHCII and NetMHCIIpan

Using the data points shared by the old and updated data sets, we first compared the different versions of NetMHCII and NetMHCIIpan. We showed how the new versions of the methods outperformed the previous versions for both NetMHCII and NetMHCIIpan. We then evaluated the performance of the two versions of the methods using only ‘new’ peptides, for the MHC molecules covered both by the old and the updated data sets. The result of this

Table 5. Comparing LOMO predictions from the old and the new method on the set of data points common between the two data sets

			NetMHCIIpan-3.1-LOMO		NetMHCIIpan-3.2-LOMO	
Allele	#Peptides	#Binders	AUC	Pseudo distance 2013	AUC	Pseudo distance 2016
DRB1_0101	2754	2635	0·742	0·22	**0·768**	0·16
DRB1_0301	1403	379	0·727	0·11	**0·736**	0·14
DRB1_0401	1639	695	0·761	0·04	**0·768**	0·04
DRB1_0404	542	331	**0·775**	0·06	0·774	0·03
DRB1_0405	1438	595	**0·825**	0·04	0·817	0·04
DRB1_0701	1619	806	**0·821**	0·28	**0·821**	0·27
DRB1_0802	1310	400	0·676	0·03	**0·701**	0·03
DRB1_0901	841	560	0·709	0·25	**0·730**	0·25
DRB1_1101	1604	730	0·713	0·06	**0·772**	0·06
DRB1_1302	1351	463	0·652	0·06	**0·663**	0·05
DRB1_1501	1601	672	0·721	0·20	**0·790**	0·13
DRB3_0101	1266	267	0·690	0·12	**0·700**	0·14
DRB4_0101	1329	467	**0·747**	0·27	0·718	0·00
DRB5_0101	1606	765	**0·802**	0·20	0·800	0·20
H-2-IAb	525	125	0·698	0·34	**0·725**	0·34
H-2-IAd	100	24	0·793	0·34	**0·805**	0·34
HLA-DPA10103-DPB10201	5	1	**1·000**	0·06	**1·000**	0·06
HLA-DPA10103-DPB10401	1075	458	0·945	0·06	**0·953**	0·06
HLA-DPA10201-DPB10101	1180	558	**0·938**	0·07	0·933	0·07
HLA-DPA10201-DPB10501	1114	415	0·935	0·07	**0·939**	0·07
HLA-DPA10301-DPB10402	1193	498	0·934	0·09	**0·938**	0·11
HLA-DQA10101-DQB10501	990	246	**0·742**	0·23	0·681	0·02
HLA-DQA10102-DQB10602	1121	503	0·570	0·23	**0·809**	0·07
HLA-DQA10301-DQB10302	1461	330	**0·635**	0·19	0·623	0·09
HLA-DQA10401-DQB10402	1436	516	**0·880**	0·26	0·703	0·02
HLA-DQA10501-DQB10201	1386	477	0·555	0·27	**0·767**	0·07
HLA-DQA10501-DQB10301	1274	530	0·451	0·19	**0·648**	0·06
Average			0·757		0·781	

For each molecule, we show the number of peptides, the number of binders, the AUC performance for the old (3.1) and new (3.2) methods, and the distance to the nearest neighbor for the old and new data set. Nearest neighbors are found from the subset of molecules in the training data characterized with at least 50 data points and at least 10 binders. Highlighted in bold is the highest performance between the two methods.
[Correction added on 02 April 2018, after first online publication: Table 5 has been updated in this version.]

analysis showed that both methods on this data set gained a significant improvement in the predictive performance, supporting the importance of expanding the size of the training data even for MHC molecules already characterized by binding data. When evaluating new peptides one has to keep in mind that MHC binding predictors are often used to select peptides for experimental validation and new data sets may be less diverse than historic data sets generated sampling the entire space of a given set of protein sequences.[34]

The main difference between NetMHCII and NetMHCIIpan is that NetMHCII is an allele-specific method trained separately for each MHC molecule, whereas NetMHCIIpan is a pan-specific method that contains a single ensemble of networks using information from all MHC molecules in the data set. We would therefore expect that the allele-specific method outperforms the pan-specific method for MHC molecules where sufficient data are available to accurately characterize the binding motif, and we would expect the pan-specific method to outperform the allele-specific method when data are scarcer. This is exactly what we observed when we compared the predictive performances of NetMHCII-2.3 and NetMHCpan-3.2. Earlier work has shown a similar result, namely that when allele-specific neural network prediction algorithms rely on a sufficient number of peptide binders to achieve high predictive performances.[33,35] This illustrates how the allele-specific method is preferable only if a large amount of data is available for the MHC molecule in question, but highlights the strength of the pan-specific methods, which can benefit from the data of related MHC molecules to make reliable predictions for MHC molecules with limited data. Because of this difference between the allele-specific and pan-specific methods, we implemented a simple combination of two methods as this has been shown to improve the predictive

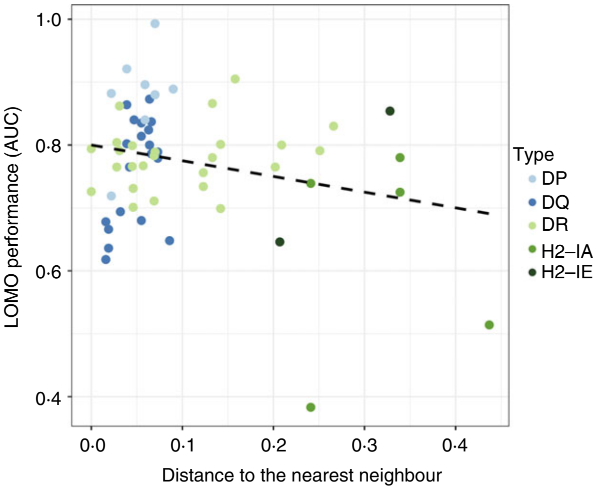


Figure 2. Predictive performance for NetMHCIIpan-3.2 LOMO on the MHC class II molecules from the 2016 data set as a function of distance to the nearest neighbour. Each HLA II isotype and H-2 molecules are displayed in different colours and the dashed line represents the least square fit for the data. [Colour figure can be viewed at wileyonlinelibrary.com]

performance for MHC-I molecules.[33] This analysis showed that NetMHCIIpan-3.2 outperforms NetMHCII-2.3 for MHC molecules, which has been trained with very few peptides, but that a combination of the predictions from the two MHC-II methods still outperformed each individual method.

Leave-one-molecule-out performance for NetMHCIIpan

One of the main powers of the NetMHCIIpan method is that it can predict binding affinities for uncharacterized MHC molecules. To assess the performance of the method in such a task, we constructed a LOMO experiment where we tested the performance of the NetMHCIIpan method for predicting binding affinity for MHC molecules not included in the training data for the method. From this analysis, we could show that the pan-specific method is capable of prediction binding affinity for MHC molecules where no binding affinity data are available and further demonstrate that the predictive performance is dependent on the distance to the nearest neighbour. This last observation indicated that the predictive performance of the NetMHCIIpan method could be further improved by including more uncharacterized MHC molecules into the training data and it is therefore important to generate experimental peptide binding affinity data points in a targeted fashion for MHC molecules not yet characterized.

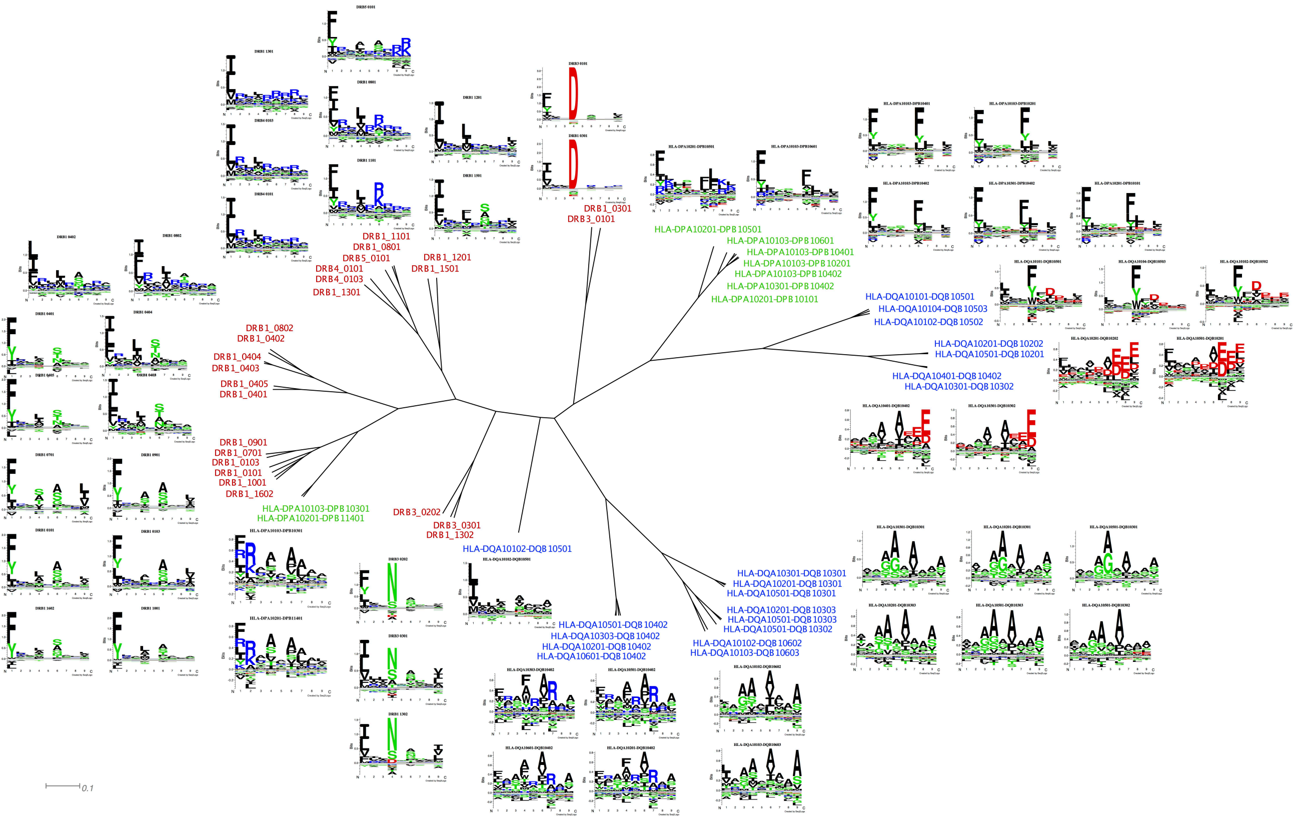


Figure 3. Distance tree for all HLA molecules found in our data set generated using the MHCCluster method. Sequence logos shows the motif of the predicted binding core for each HLA and were generated using Seq2Logo.[30] [Colour figure can be viewed at wileyonlinelibrary.com]

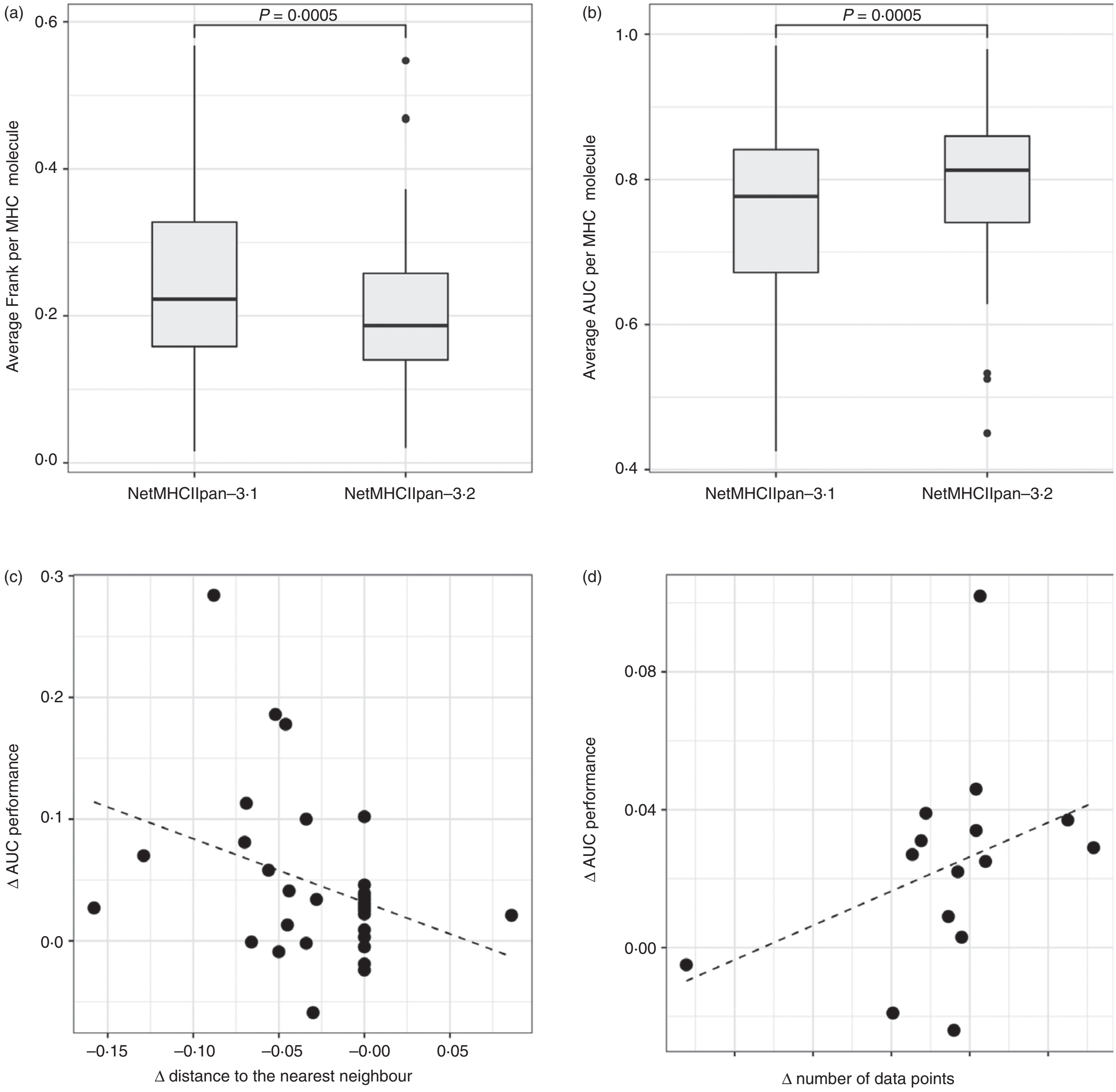


Figure 4. Performance of NetMHCIIpan-3.1 and NetMHCIIpan-3.2 using the T-cell epitope benchmark set. (a) The average Frank performance per MHC molecule for the two versions of NetMHCIIpan. (b) The average AUC performance per MHC molecule for the two versions of NetMHCIIpan. (c) The change in the distance to the nearest neighbour between the two data sets used for training the old and the new versions of NetMHCIIpan as a function of the change in distance to the nearest neighbour. (d) the change in the number of data points between the two data sets used for training NetMHCIIpan-3.1 and NetMHCIIpan-3.2 as a function of the change in the performance, including only MHC molecules where the pseudo sequence did not change between two data sets. The dashed line in the two scatterplots represents the least square fit for the data.

Distance tree for HLA class II molecules

To understand the different groups of HLA class II molecules, we generated a fictional distance tree using NetMHCIIpan-3.2. The groups shown in this distance tree can be used to understand how peptides interact with different MHC molecules and can be used to discriminate between binders and non-binders. The distance tree can also be used to identify T-cell epitopes with similar properties important for the design of epitope-based vaccines. Another aspect that can be observed for the tree is that most MHC molecules have strong anchor positions at P1, P4, P6 and P9, which have also been observed in previous studies.[8]

T-cell epitope benchmark

Accurate predictions of peptide binding affinities to MHC molecules are important for understanding the cell-mediated immune response and for generating better screening methods for cost-effective identification of immunogenic peptides. We therefore wanted to test the predictive performance of the two versions of NetMHCIIpan on a T-cell epitope data set, and doing this we demonstrated how the new version of NetMHCIIpan obtained a significantly improved predictive performance compared with the earlier version. Two main factors explain this performance gain: (i) including data for new MHC-II molecules decreases the distance to the nearest neighbour, (ii) including an increased number of data points allows the method better characterizing the specificity of a given MHC-II molecule.

In conclusion, we believe that NetMHCII and NetMHCIIpan can be used to improve MHC-II binding predictions and reduce experimental costs for immunologists working within the field of epitope-based vaccine design, and to improve our knowledge about the peptide–MHC interaction, a key event in the cellular immune response.

Acknowledgements

This work was supported by Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN272201200010C.

Disclosures

The authors declare having no competing interests.

References

1 Castellino F, Zhong G, Germain RN. Antigen presentation by MHC class II molecules: invariant chain function, protein trafficking, and the molecular basis of diverse determinant capture. *Hum Immunol* 1997; **54**:159–69.

2 Lin HH, Zhang GL, Tongchusak S, Reinherz EL, Brusic V. Evaluation of MHC-II peptide binding prediction servers: applications for vaccine research. *BMC Bioinformatics* 2008; **9**(Suppl. 12):S22.

3 Traherne JA. Human MHC architecture and evolution: implications for disease association studies. *Int J Immunogenet* 2008; **35**:179–92.

4 Nielsen M, Lund O, Buus S, Lundegaard C. MHC Class II epitope predictive algorithms. *Immunology* 2010; **130**:319–28.

5 Brown JH, Jardetzky TS, Gorga JC, Stern LJ, Urban RG, Strominger JL *et al.* Three-dimensional structure of the human Class II histocompatibility antigen HLA-DR1. *J Immunol* 2015; **194**:5–11.

6 Chicz RM, Urban RG, Lane WS, Gorga JC, Stern LJ, Vignali DAA *et al.* Predominant naturally processed peptides bound to HLA-DR1 are derived from MHC-related molecules and are heterogeneous in size. *Nature* 1992; **358**:764–8.

7 Holland CJ, Cole DK, Godkin A. Re-directing CD4$^+$ T cell responses with the flanking residues of MHC class II-bound peptides: the core is not enough. *Front Immunol* 2013; **4**:172.

8 Zhang L, Udaka K, Mamitsuka H, Zhu S. Toward more accurate pan-specific MHC-peptide binding prediction: a review of current methods and tools. *Brief Bioinform* 2012; **13**:350–64.

9 Arnold PY, La Gruta NL, Miller T, Vignali KM, Adams PS, Woodland DL *et al.* The majority of immunogenic epitopes generate CD4$^+$ T cells that are dependent on MHC Class II-bound peptide-flanking residues. *J Immunol* 2002; **169**:739–49.

10 Nielsen M, Lundegaard C, Blicher T, Lamberth K, Harndahl M, Justesen S *et al.* NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. *PLoS ONE* 2007; **2**:e796.

11 Karosiene E, Rasmussen M, Blicher T, Lund O, Buus S, Nielsen M. NetMHCIIpan-3.0, a common pan-specific MHC class II prediction method including all three human MHC class II isotypes, HLA-DR, HLA-DP and HLA-DQ. *Immunogenetics* 2013; **65**:711–24.

12 Iwai LK, Yoshida M, Sidney J, Shikanai-Yasuda MA, Goldberg AC, Juliano MA *et al. In silico* prediction of peptides binding to multiple HLA-DR molecules accurately identifies immunodominant epitopes from gp43 of *Paracoccidioides brasiliensis* frequently recognized in primary peripheral blood mononuclear cell responses from sensitized individuals. *Mol Med* 2003; **9**:209–19.

13 Mustafa AS, Shaban FA. ProPred analysis and experimental evaluation of promiscuous T-cell epitopes of three major secreted antigens of *Mycobacterium tuberculosis*. *Tuberculosis* 2006; **86**:115–24.

14 Al-Attiyah R, Mustafa AS. Computer-assisted prediction of HLA-DR binding and experimental analysis for human promiscuous Th1-cell peptides in the 24 kDa secreted lipoprotein (LppX) of *Mycobacterium tuberculosis*. *Scand J Immunol* 2004; **59**:16–24.

15 Nielsen M, Lund O. NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. *BMC Bioinformatics* 2009; **10**:1471.

16 Andreatta M, Karosiene E, Rasmussen M, Stryhn A, Buus S, Nielsen M. Accurate pan-specific prediction of peptide-MHC class II binding affinity with improved binding core identification. *Immunogenetics* 2015; **67**:641–50.

17 Sturniolo T, Bono E, Ding J, Raddrizzani L, Tuereci O, Sahin U *et al.* Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. *Nat Biotechnol* 1999; **17**:555–61.

18 Zhang L, Chen Y, Wong H, Zhou S, Mamitsuka H. TEPITOPEpan: extending TEPITOPE for peptide binding prediction covering over 700 HLA-DR molecules. *PLoS ONE* 2012; **7**:e30483.

19 Singh H, Raghava GPS. ProPred: prediction of HLA-DR binding sites. *Bioinformatics* 2002; **17**:1236–7.

20 Reche PA, Glutting JP, Reinherz EL. Prediction of MHC class I binding peptides using profile motifs. *Hum Immunol* 2002; **63**:701–9.

21 Reche PA, Glutting JP, Zhang H, Reinherz EL. Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles. *Immunogenetics* 2004; **56**:405–19.

22 Wan J, Liu W, Xu Q, Ren Y, Flower DR, Li T. SVRMHC prediction server for MHC-binding peptides. *BMC Bioinformatics* 2006; **7**:463.

23 Sette A, Peters B, Wang P, Sidney J, Dow C, Mothe B. A systematic assessment of MHC Class II peptide binding predictions and evaluation of a consensus approach. *PLoS Comput Biol* 2008; **4**:e1000048.

24 Nielsen M, Andreatta M. NNAlign: a platform to construct and evaluate artificial neural network models of receptor–ligand interactions. *Nucleic Acids Res* 2017; **45**:344–9.

25 Andreatta M, Schafer-nielsen C, Lund O, Buus S, Nielsen M. NNAlign: a web-based prediction method allowing non- expert end-user discovery of sequence motifs in quantitative peptide data. *PLoS ONE* 2011; **6**:e26781.

26 Vita R, Overton JA, Greenbaum JA, Ponomarenko J, Clark D, Cantrell JR *et al.* The immune epitope database (IEDB) 3.0. *Nucleic Acids Res* 2015; **43**:D405–12.

27 Nielsen M, Lundegaard C, Worning P, Lauemøller SL, Lamberth K, Buus S *et al.* Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Sci* 2003; **12**:1007–17.

28 Nielsen M, Lundegaard C, Lund O. Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. *BMC Bioinformatics* 2007; **8**:238.

29 Nielsen M, Lundegaard C, Blicher T, Peters B, Sette A, Justesen S *et al.* Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan. *PLoS Comput Biol* 2008; **4**:e1000107.

30 Thomsen MCF, Nielsen M. Seq2Logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic Acids Res* 2012; **40**:281–7.

31 Thomsen M, Lundegaard C, Nielsen M. MHCcluster, a method for functional clustering of MHC molecules. *Immunogenetics* 2013; **65**:655–65.

32 Huson DH, Bryant D. Application of phylogenetic networks in evolutionary studies. *Mol Biol Evol* 2006; **23**:254–67.

33 Karosiene E, Lundegaard C, Lund O. NetMHCcons: a consensus method for the major histocompatibility complex class I predictions. *Immunogenetics* 2012; **64**:177–86.

34 Kim Y, Sidney J, Buus S, Sette A, Nielsen M, Peters B. Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions. *BMC Bioinformatics* 2014; **15**:241.

35 Zhang H, Lundegaard C, Nielsen M. Pan-specific MHC class I predictors: a benchmark of HLA class I pan-specific prediction methods. *Bioinformatics* 2009; **25**:83–9.

Supporting Information

Additional Supporting Information may be found in the online version of this article:

Table S1. Description of the full 2016 data set.

Table S2. NetMHCII and NetMHCIIpan predictions of peptide binding cores.

Table S3. Performance for NetMHCII-2.3, NetMHCIIpan-3.2 and the combined method.

Table S4. The performance of the Leave-one-molecule-out (LOMO) benchmark analysis of NetMHCIIpan-3.2 including information about distance to nearest neighbour.

Table S5. The predictive performance for NetMHCIIpan-3.1 and NetMHCIIpan-3.2 on the IEDB T-cell epitope data set.

Chapter 3: Paper II

One of the unsolved problems in the field of immunoinformatics is the prediction of T-cell epitopes. Most existing tools for identifying T-cell epitopes predict the peptide-MHC binding strength and uses this to select potential peptide candidates when searching for T-cell epitopes. However, beyond being presented by the MHC, a peptide also needs to find a matching T-cell in order to become immunogenic. So to truly understand what makes a peptide immunogenic, we need to understand the interaction between TCRs and peptide-MHC complexes. A first step to achieve this, is to build structural models of the TCR-pMHC complex and use these structures to characterize and potentially predict TCR-pMHC binding.

In this chapter, we therefore present TCR-pMHCmodels, an automated tool for modeling the structure of TCR-pMHC complexes.

The tool is available at:
http://www.cbs.dtu.dk/services/TCRpMHCmodels/

TCRpMHCmodels: Structural modelling of TCR-pMHC class I complexes

Kamilla Kjærgaard Jensen[1], Vasileios Rantos[1,2], Emma Jappe[1,7], Tobias Hegelund Olsen[1], Martin Closter Jespersen[1], Vanessa Jurtz[3], Leon Eyrich Jessen[1], Esteban Lanzarotti[4], Swapnil Mahajan[5], Bjoern Peters[5,6], Morten Nielsen[1,4], Paolo Marcatili[1*]

[1] Department of Bio and Health Informatics, Technical University of Denmark, Kgs. Lyngby, Denmark
[2] Centre for Structural Systems Biology (CSSB), DESY and European Molecular Biology Laboratory, Notkestrasse 85, 22607, Hamburg, Germany
[3] Department of Bioinformatics and Data Mining, Novo Nordisk A/S, Måløv 2760, Denmark
[4] Instituto de Investigaciones Biotecnológicas, Universidad Nacional de San Martín, Argentina
[5] Division of Vaccine Discovery, La Jolla Institute for Allergy and Immunology, La Jolla, CA 92037, USA
[6] University of California San Diego, Department of Medicine, La Jolla, CA 92037, USA
[7] Evaxion Biotech, Bredgade 34E, 1260 Copenhagen K, Denmark

Keywords: TCR-pMHC complex, major histocompatibility complex (MHC), T-cell receptor (TCR), comparative modelling, structural modelling

Abstract

The interaction between the class I major histocompatibility complex (MHC), the peptide presented by the MHC and the T-cell receptor (TCR) is a key determinant of the cellular immune response. Here, we present TCRpMHCmodels, a method for accurate structural modelling of the TCR-peptide-MHC (TCR-pMHC) complex. This TCR-pMHC modelling pipeline takes as input the amino acid sequence and generates models of the TCR-pMHC complex, with a median Cα RMSD of 2.31Å. TCRpMHCmodels significantly outperforms TCRFlexDock, a specialised method for docking pMHC and TCR structures.

TCRpMHCmodels is simple to use and the modelling pipeline takes, on average, only two minutes. Thanks to its ease of use and high modelling accuracy, we expect TCRpMHCmodels to provide insights into the underlying mechanisms of TCR and pMHC interactions and aid in the development of advanced T-cell-based immunotherapies and rational design of vaccines. The TCRpMHCmodels tool is available at:
http://www.cbs.dtu.dk/services/TCRpMHCmodels/.

Introduction

As part of the adaptive immune response, T-cells recognise and kill pathogenic or pathogen-infected cells[1,2]. Understanding the mechanisms of such immune responses is therefore important for the development of cancer immunotherapies and rational vaccine design [3–9]. The activation of T-cell immunity is primarily driven by the interaction between peptides presented by major histocompatibility complexes (pMHCs) and T cell receptors (TCRs) [1,10,11]. TCRs are found on the surface of T-cells where they recognise protein fragments, named antigens, when these are presented by the MHC on the cell surface of antigen presenting cells. TCRs consist of two membrane-bound chains, which can be either α and β chains or γ and δ chains [12]. The majority of T-cells expresses αβ-TCRs and these T-cells can be further subdivided into cytotoxic T-cells and T-helper cells [13]. Cytotoxic T-cells interact with the MHC class I molecules and are involved in direct killing of pathogen-infected cells, whereas T-helper cells interact with the MHC class II molecules after which they directly or indirectly activate other immune cells to combat the pathogenic infection [14]. In this work, we focus on modelling the TCR-pMHC complex of αβ-TCRs and MHC class I molecules, as these constitute the majority of the available structural complexes.

The TCR-pMHC complex consists of three components, namely the TCR, the MHC and the MHC-bound peptide [2]. The MHC class I molecule is a heterodimeric glycoprotein that consists of an α chain and a β2-microglobulin chain. The α chain is composed of three globular domains named α1, α2 and α3 which are highly polymorphic, allowing the MHC variants to accommodate a diverse range of peptides of different lengths and compositions [2].

Each of the two chains in the αβ-TCR has a variable (V) and constant (C) domain. Located within the variable domains are three complementarity determining region (CDR) loops and these account for the main interaction with the pMHC [15]. The sequence of the CDR loops are determined by a recombination process which leads to a highly diverse set of T-cells with different TCRs [16]. It is assumed that the recombination process can theoretically generate more than 10^{15} T-cell variants [17], but only a minor fraction of these, 10^{6} to 10^{8}, are actually expressed at any given time in the human organism [15]. Despite the high variability in the CDR loop sequence, it has been shown that most CDRs only adopt a limited number of main chain conformations named canonical structures and that these canonical structures can usually be identified by specific sequence features [18–20].

In the past, numerous sequence- and structure-based tools have been developed to predict and model the structure of and/or the interaction between the peptide and the MHC class I molecule [21–27]. Several structure-based tools for modelling the TCR have likewise been developed in the past [18,28]. In recent years, there has been an increased focus on the TCR-pMHC binding accompanied by the development of tools for predicting the interaction between the pMHC and the TCR [29–32]. In particular, previous work has demonstrated how a simple force-field-based approach can be used to identify the cognate pMHC target of a TCR given the availability of structural models of the TCR-pMHC complex [33]. Additionally, structural models have been used to analyse how mutations in the peptide affect the binding to a specific TCR [34]. While tools to deal with peptide-MHC binding and predicting T-cell epitopes have been developed over the last decade [14–17], limited work has been dedicated to the task of generating accurate

TCR-pMHC models. In order to aid this development, we present a novel framework for automated modelling of TCR-pMHC complexes. The modelling pipeline, named TCRpMHCmodels, utilises the amino acid sequences of the MHC, peptide and TCR α and β chains. In a fully automated manner, the pipeline applies a series of simple comparative modelling steps to construct structural models of the pMHC, the TCR, and, subsequently, the pMHC-TCR complex. The tool does not include any assessment of the binding energy or prediction and ranking of potential T-cell epitopes; however, we believe that the models produced by our tool in combination with refined binding energy models can be used to provide valuable insights into the mechanisms underlying the interaction between TCR and pMHC. Thus, the models can guide the refined prediction of T cell epitopes extending beyond prediction of MHC antigen presentation.

Here, we report the large-scale benchmark evaluation of different modelling strategies, including single- versus multi-template modelling, as well as different similarity measures for optimal template selection, to arrive at the optimal method implemented in TCRpMHCmodels. The performance of TCRpMHCmodels is benchmarked against TCRFlexDock [29], a specialised protein docking method for identifying the correct orientation between the TCR and pMHC structure. Lastly, we test the performance of TCRpMHCmodels on 14 TCR-pMHC structures deposited in the Protein Data Bank (PDB) [35] after the generation of the TCR-pMHC template database.

Results

TCRpMHCmodels is an automated modelling pipeline for generating structural models of a TCR-pMHC complex using only the amino acid sequence as input. This method adopts a template-based modelling approach, generating a structural model of a given protein sequence (target), using one or more experimentally determined structures of related homologous proteins (templates).

The initial steps in the TCRpMHCmodels pipeline involves the modelling of the TCR and the pMHC separately. These two models are then combined when building the final TCR-pMHC complex. The TCR is generated with LYRA [18], using templates from a TCR database (see Method section), while the pMHC is generated with MODELLER [36], using templates from a pMHC database (see Method section). The full TCR-pMHC model is then generated with MODELLER using the TCR and pMHC model as templates together with one or more templates from the TCR-pMHC database (see Method section, Figure 1).

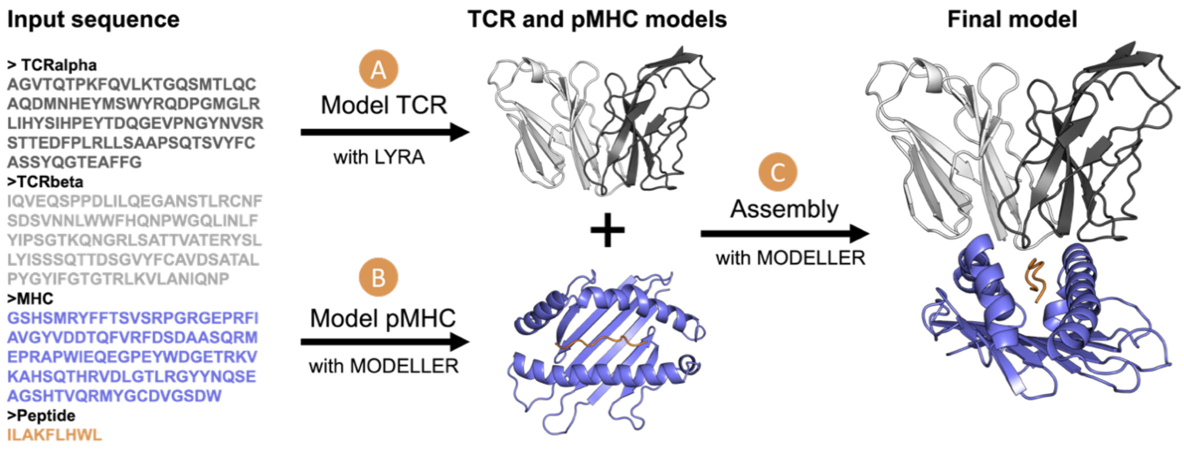


Figure 1: Flowchart of the computational framework for modelling TCR-pMHC complexes, from the input sequence to the final TCR-pMHC model. The MHC molecule is depicted in blue and the peptide in orange, while the two chains of TCR, α and β, are represented in light and dark grey, respectively.

MODELLER is a comparative modelling tool for predicting the three-dimensional structure of proteins [37]. The tool needs an initial alignment of the sequence to be modelled and one or more known structures. Based on the alignment, MODELLER automatically extracts spatial features, such as Cα-Cα distances, hydrogen bonds, and main chain and side chain dihedral angles, and transfers these from the templates to the target. Lastly, the three-dimensional model is obtained by satisfying all spatial restraints as accurately as possible.

LYRA is a tool that can predict the structure of TCRs. The tool starts by selecting the best framework template for each chain in the TCR, after which it uses the canonical structure model to select the best templates for each of the CDRs. The CDRs are then grafted onto the framework templates which is then merged and the side chains are repacked to generate the final TCR model.

To ensure good model quality of the TCR-pMHC complex, we have optimised each of the steps in TCRpMHCmodels. The results from these optimisations are described in the following sections. All RMSDs, unless otherwise specified, are calculated on Cα atoms only.

pMHC model optimisation

The first step in the TCRpMHCmodels method is building a structural model of the pMHC. In order to assess the model quality of this step, we have generated structural models for each structure in the pMHC database using a leave-one-out (LOO) approach and evaluated the quality of the generated models by comparing them to their native structure found in the pMHC database. For the model optimisation process, we imposed four different template-target sequence identity thresholds of 99.9%, 95%, 90% and 80%, selecting only structural templates with a sequence identity below the given threshold (see the Method section). By using different sequence identity thresholds, we thereby generate a more diverse set of structural models with both high and low sequence identity to the template database.

Using the LOO approach with the four sequence identity thresholds, we generated four structural models for each target in the pMHC database. When modelling the pMHC, we also investigated four different template selection methods, which we denote OneWeighted, OneUnweighted, MultiWeighted and MultiUnweighted, to evaluate the effect of using a single or multiple templates as well as using an unweighted or weighted sequence identity score for template selection. The four different template selection methods were further compared with a random baseline. For more details on the template selection methods and the random baseline (see the Method section). The results of this analysis are illustrated in Figure 2 and Supplementary Figure S2.

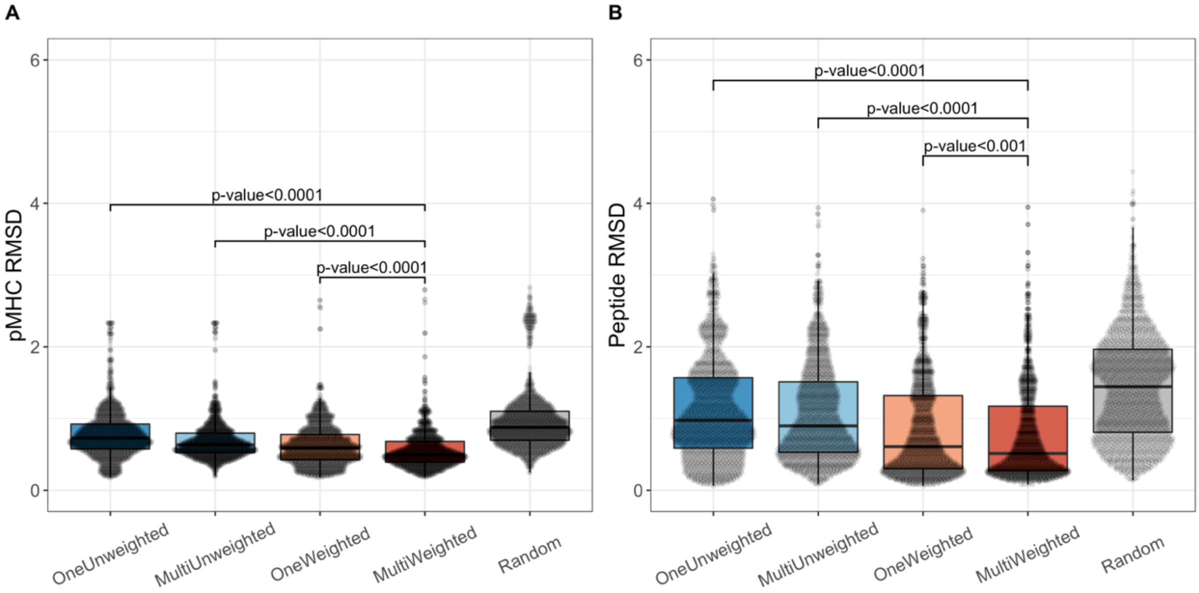


Figure 2: RMSD accuracy for the different template selection methods. **A)** The RMSD for the pMHC complex. **B)** The RMSD for the peptide. For each target in the template database, we generate four models using the four different sequence identity thresholds. The OneUnweighted method uses only a single pMHC template with no weights on the sequence identity. The MultiUnweighted also have no weights on the sequence identity but this method uses multiple templates. The OneWeighted method uses only a single pMHC template and a weighted sequence identity. The MultiWeighted method uses the weighted sequence identity and multiple templates. The four different template selection methods are compared with a random baseline (see the Method section for more details). The p-values were obtained using the Wilcoxon signed-rank test.

From Figure 2 A and B, we observe that the MultiWeighted method performed significantly better than the other methods, both when comparing the RMSD of the pMHC and the peptide. The median RMSD values of the pMHC and the peptide in the MultiWeighted method are 0.54Å and 0.50Å, respectively. For comparison, the median RMSD values of the Random method are 0.88Å and 1.44Å for the pMHC and the peptide, respectively. The improved accuracy of the peptide RMSD shows that the MultiWeighted method is capable of accurately modelling this part of the pMHC complex, which is less conserved and fundamental for the TCR specificity. Similar conclusions were obtained using the TM-score (see Supplementary Figure S1). Due to the improved accuracy, we therefore selected the MultiWeighted method as the default method for building the pMHC in TCRpMHCmodels.

In Figure 3, we display the accuracy of the MultiWeighted method in a more detailed manner, showing how the modelling accuracy of the pMHC and the peptide depends on the sequence identity to the best template, using a Chothia-Lesk plot [38]. From this figure, it is clear that the modelling accuracy for the pMHC complex is in general very high (less than 1Å), even in situations where the best template shares very limited similarity to the target. However, it is also clear that this high accuracy is driven by the very conserved structure of the MHC, and that the picture is very different when focusing only on the peptide.

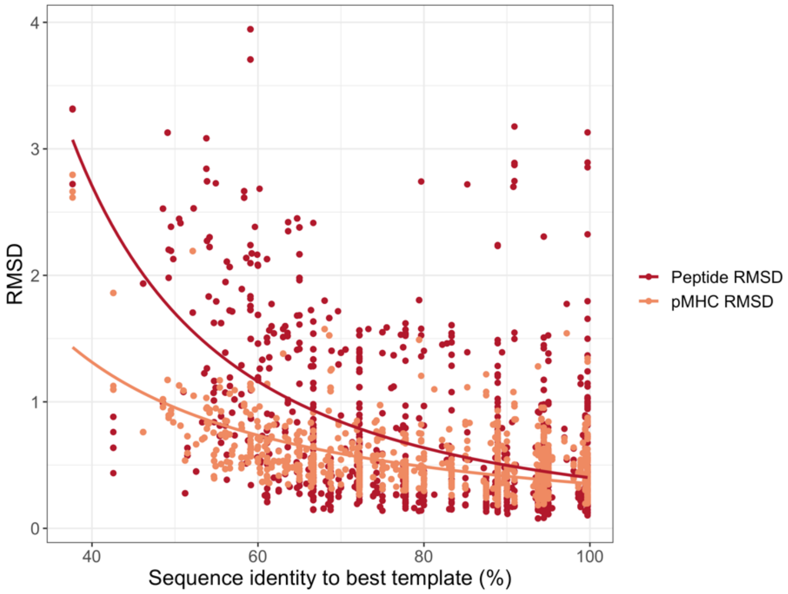


Figure 3: Benchmark results for the pMHC models generated using the MultiWeighted method. Chothia-Lesk plot showing the RMSD accuracy for the pMHC (orange) and the peptide (red) against the sequence identity to the best template.

To further investigate this, we analysed how the pMHC model accuracy depends on the peptide length and the sequence identity. Supplementary Figure S3 demonstrates (as expected) that longer peptides tend to have higher peptide RMSDs. The same tendency is observed when investigating models generated using different sequence identity thresholds for template selection, see Supplementary Figure S4.

TCR model accuracy

The TCR subunit of the TCR-pMHC complex is modelled using LYRA [18]. LYRA is an automated method for modelling TCRs and it generates models of high accuracy with a 1.48Å global RMSD and 2.13Å binding site RMSD. The main advantage of LYRA is that it uses the so-called canonical structure method to select the best templates for the CDR loops. The canonical structures are conserved and limited in conformations of CDR loops that can usually be identified by sequence-based rules. The canonical structure model has been proven valid for both antibodies and TCRs [18–20] and LYRA is the only automated method that uses the canonical structure method for building structural models of the TCR.

TCR-pMHC model optimisation

The final task of TCRpMHCmodels is to find the optimal approach for assembling the TCR and pMHC model to form the TCR-pMHC complex. In our TCR-pMHC modelling pipeline, this is achieved with MODELLER using the TCR and pMHC model as templates together with one or more templates from the TCR-pMHC database.

In order to assess the model quality of this step, we generated structural models for each structure in the TCR-pMHC database using a LOO approach, and we then evaluated the quality of the generated models by comparing them to their native structure found in the TCR-pMHC database.

Using the LOO approach with the four different sequence identity thresholds, we then generated four models for each target in the TCR-pMHC database. When modelling the TCR-pMHC, we investigated three different template selection methods, which we denote OneUnweighted, OneWeighted and MultiWeighted, to evaluate the effect of using a single or multiple templates as well as using an unweighted or weighted sequence identity score for template selection. The results from this analysis are depicted in Figure 4.

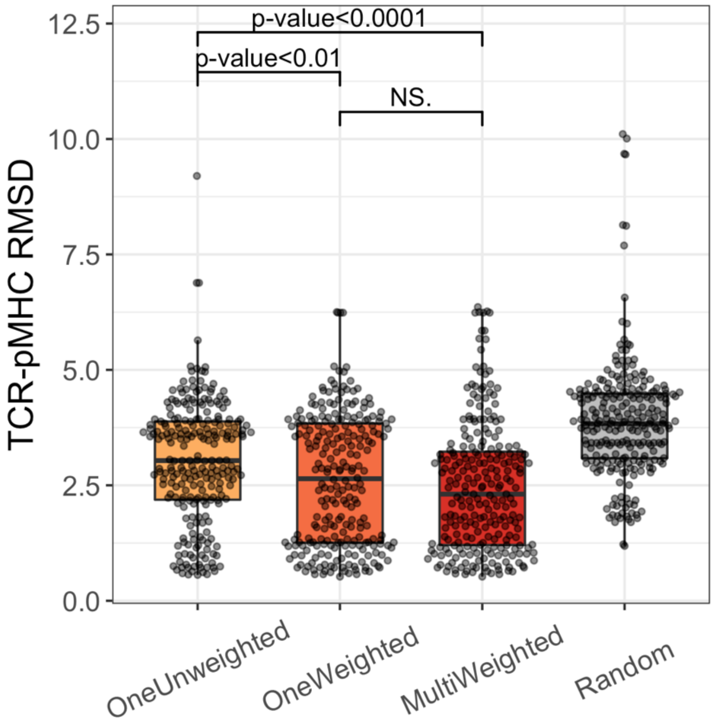


Figure 4: The TCR-pMHC RMSD accuracy for the different template selection methods. For each target in the template database we generated four models using the four different sequence identity thresholds and evaluated the generated models using the RMSD for the TCR-pMHC complex. The OneUnweighted method uses only a single TCR-pMHC template with no weights on the sequence identity. The OneWeighted method uses only a single TCR-pMHC template and a weighted sequence identity. The MultiWeighted method uses the weighted sequence identity and multiple templates. The three different template selection methods are compared with a random baseline shown in grey. The p-values were obtained using the Wilcoxon signed-rank test. The RMSD accuracy of the TCR, the pMHC and the peptide are shown in Supplementary Figure S5.

From Figure 4, we observe that the MultiWeighted method has a lower median than the other methods and we therefore used this method as the default template selection method in the final TCRpMHCmodels pipeline. We also show the TM-scores (see Supplementary Figure S6) and the Chochia-Lesk plot (see Supplementary Figure S7). The MultiWeighted method has a median TCR-pMHC RMSD of 2.31Å which shows that model accuracy for the TCR-pMHC complex is in general very high. Comparing the median TCR-pMHC RMSD from the MultiWeighted method with the median RMSD of the Random method, we see an 86% improvement in the accuracy.

Figure 5 shows the accuracy of the MultiWeighted method in a more detailed manner, by plotting the model accuracy based on the sequence identity to the best template, using a Chothia-Lesk plot.

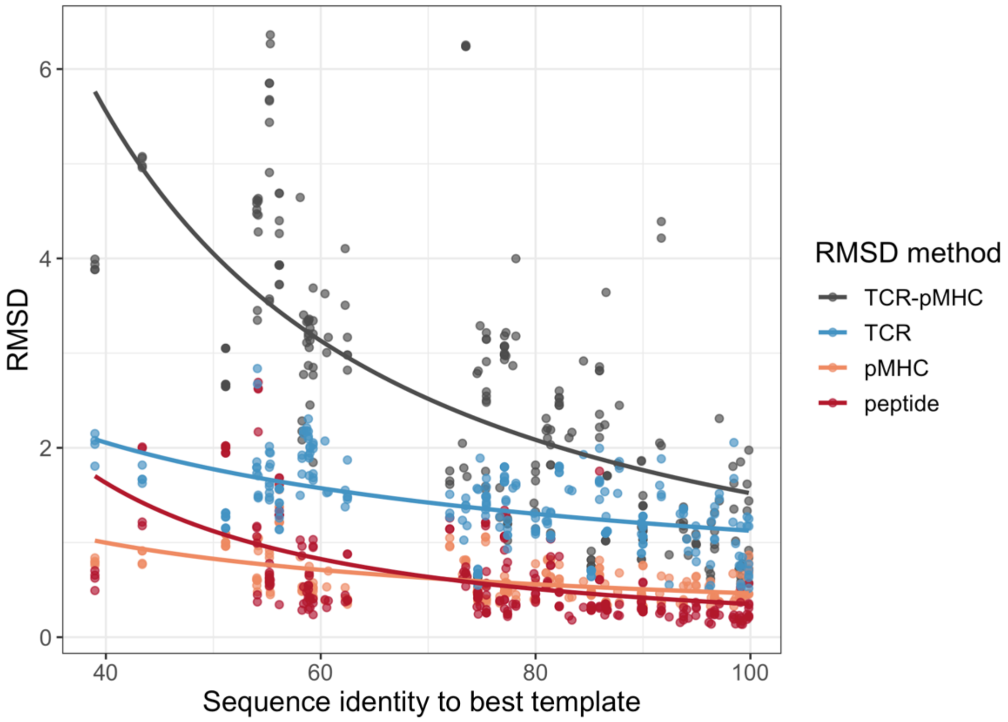


Figure 5: RMSD accuracy for the TCR-pMHC models generated using the MultiWeighted method. Chothia-Lesk plot showing the RMSD accuracy for the TCR-pMHC (grey), the TCR (blue), the pMHC (orange) and the peptide (red) against the sequence identity to the best template.

From Figure 5 we see that the modelling of final TCR-pMHC complexes is much more dependent on the sequence identity to the templates compared to the pMHC and the TCR subunits. This could be explained by the fact that the conformation of the TCR-pMHC is much more variable than the conformation of the TCR or the pMHC alone.

Benchmark against TCRFlexDock

Our optimised TCRpMHCmodels pipeline was benchmarked against the TCRFlexDock method, a specialised protein docking method for finding the correct orientation between the TCR and pMHC to form the final TCR-pMHC complex (see the Method section). The TCRFlexDock protocol applies a set of iterative Monte Carlo moves and side chain packing, combined with refinement of both peptide and CDR loop conformations [29]. The TCRFlexDock docking protocol was run 1000 times to generate 1000 TCR-pMHC models, which were then scored using ZRANK [39] to select the best models.
To compare the quality of the models produced by the two different methods we used both RMSDs and DockQ scores [40]. The result of the benchmark analysis is shown in Figure 6.

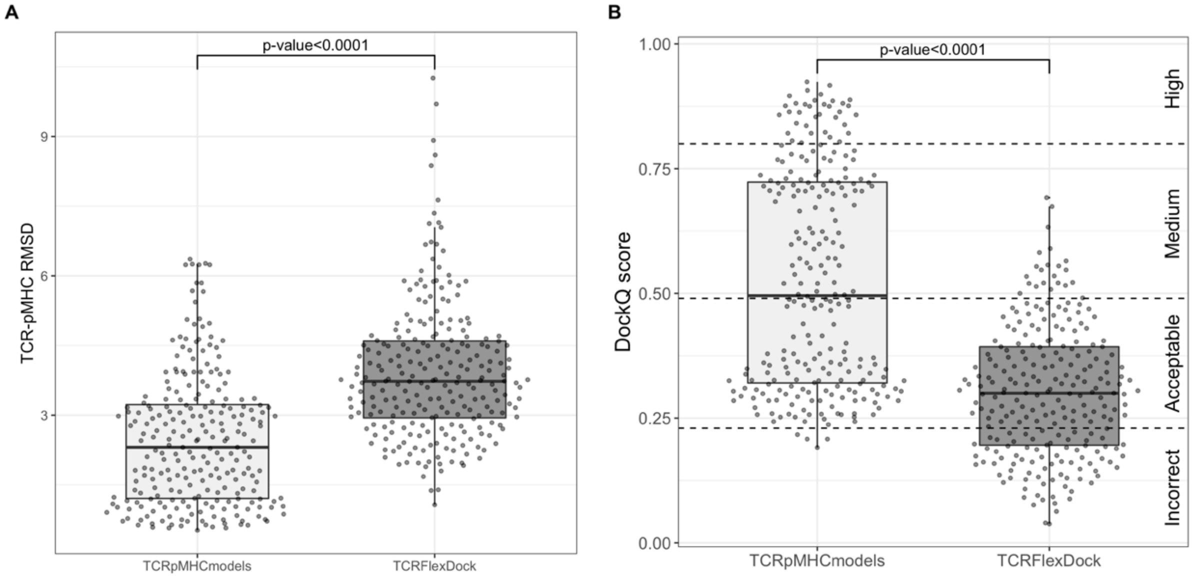


Figure 6: Benchmark analysis of the TCR-pMHC models. **A)** Shows the TCR-pMHC RMSD accuracy between the models produced by TCRpMHCmodels and TCRFlexDock. **B)** Shows the DockQ scores between the models produced by TCRpMHCmodels and TCRFlexDock. The statistical comparison was performed using the Wilcoxon signed-rank test.

We compared the accuracy of the models produced with the TCRpMHCmodels pipeline and the TCRFlexDock protocol, using RMSDs and DockQ scores (see Figure 6). The RMSD is a measure of the average distance between the Cα atoms from the model and the Cα atoms in the native structure solved using X-ray crystallography. This measure, while accounting for the overall accuracy of the model, does not take into account side-chain placement which is critical for identifying molecular interactions and the TCR-pMHC interface as a whole. This is partially accounted for by using the DockQ score, a model quality measure derived by combining Fnat, LRMS, and iRMS, three measures of model quality proposed and standardised by the Critical Assessment of PRedicted Interactions (CAPRI) community [41]. Fnat is the fraction of native and non-native residue-residue contacts in the interface, LRMS is the RMSD of the backbone atoms in the ligand after superimposing only the receptor from the native and non-native structure, and iRMS is the backbone atoms of all interface residues [42]. The DockQ score ranges from 0 to 1 and can be used to assign the quality of a model into the four classes: Incorrect (DockQ score < 0.23), Acceptable (DockQ score ≥ 0.23 and DockQ score < 0.49), Medium (DockQ score ≥ 0.49 and DockQ score < 0.80) and High (DockQ score ≥ 0.80) [40]. From Figure 6, we observe that the models generated with TCRpMHCmodels were significantly more accurate than the models generated using the TCRFlexDock protocol, both in terms of RMSDs and DockQ scores. The median RMSD values of TCRpMHCmodels and TCRFlexDock were 2.31Å and 3.73Å, respectively, while the median DockQ scores were 0.50 and 0.3, respectively. Looking only at the DockQ scores we see that almost all the models produced by the TCRpMHCmodels pipeline had an acceptable, medium or high model quality, whereas the models produced using the TCRFlexDock protocol had an incorrect, medium or acceptable model quality. The model quality measure Fnat, LRMS, and iRMS is shown in Supplementary Figure S8. This indicates that the TCRFlexDock protocol is less accurate at identifying the correct conformation of the TCR-pMHC complex compared to TCRpMHCmodels, even though the TCRFlexDock protocol optimises both the TCR orientation, the CDR loop conformation and the MHC bound peptide conformation during docking.

To better understand the quality of the models generated by TCRpMHCmodels and TCRFlexDock, we made a visual inspection of the models with the highest and lowest quality for each method (see Supplementary Figure S9). We here observe that the models produced by TCRpMHCmodels are better at predicting the interface between the TCR and pMHC compared to TCRFlexDock. The model with the lowest quality from TCRpMHCmodes had a Fnat score of 0.38, indicating that around 38 % of the native residue-residue contacts in the interface where correctly predicted. The model with the lowest quality from TCRFlexDock had Fnat score of 0.02. This low Fnat score would be classified as an incorrectly predicted interface as only around 2% of the native residue-residue contacts in the interface where correctly predicted. Looking at this low-quality model generated by TCRFlexDock, it can be observed that the CDR loops of the TCR is mainly interacting with one of the sides in the MHC molecule instead of interacting with the peptide as would be expected (see Supplementary Figure S9).

We further investigated the CDR loop accuracy between the models generated by TCRFlexDock and TCRpMHCmodels and compared these to the initial TCR model produced by LYRA (see Figure 7). Looking only at the RMSD for the CDR loops, we observe that the models generated with TCRpMHCmodels have a slightly better loop accuracy compared to the initial TCR models. Generating the final TCR-pMHC complex must therefore change the loop conformation of the CDRs to better fit the peptide-MHC, thereby generating CDR loops which are closer to the loops found in the native TCR-pMHC complex. In comparison to TCRpMHCmodels, the CDR loop accuracy of the model generated with TCRFlexDock decreases, both compared to the initial TCR model and the models generated with TCRpMHCmodels.

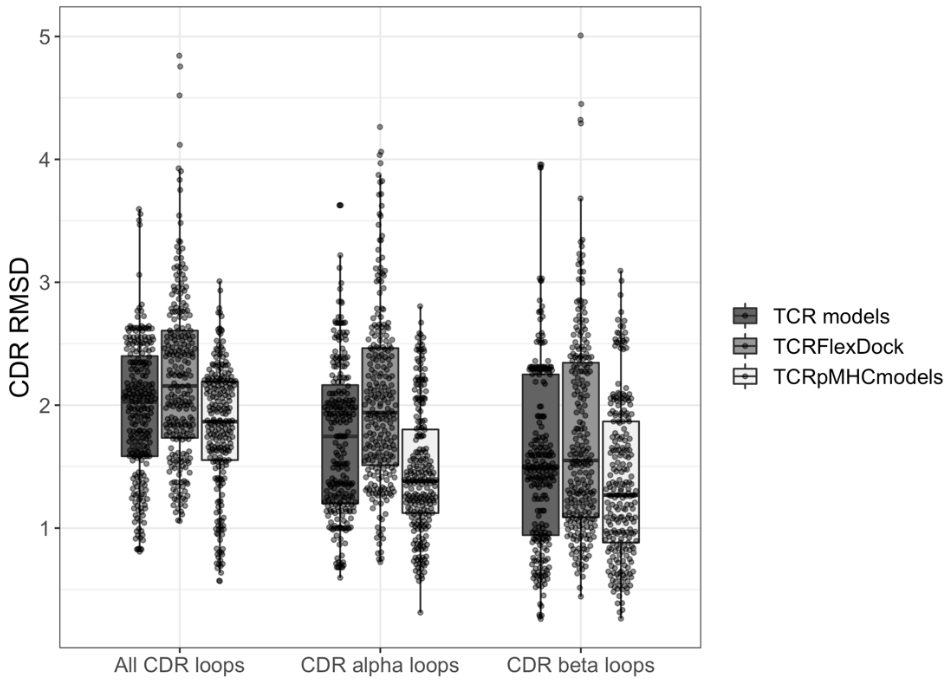


Figure 7: CDR accuracy of the TCR-pMHC models from the benchmark analysis. Shows the CDR RMSD accuracy between the TCR-pMHC models produced by the TCRpMHCmodels pipeline and TCRFlexDock protocol, compared to the initial TCR model produced by the TCR-pMHC pipeline.

Benchmark against new structures

TCRpMHCmodels was benchmarked using 14 TCR-pMHC structures deposited in IEDB after the TCR-pMHC template database was created. Note that 4 additional structures were available, that could not be modelled; two due to lack of available CDR templates and two due to lack of available TCR-pMHC templates with the correct peptide length. For each of the 14 cases, we generated a single model using TCRpMHCmodels. The average RMSDs for the TCR-pMHC, TCR, pMHC and peptide were 3.20Å, 1.81Å, 0.69Å and 0.77Å, respectively. For more details see Supplementary Figure S6 and Supplementary Table S1. This data suggests that TCRpMHCmodels generates accurate models for both the TCR and pMHC complex but is less accurate at predicting the TCR orientation over the pMHC and these predictions should therefore not be over-interpreted. The model accuracy for these new structures is comparable to the results shown in Figure 6, with the exception of one structure (PDB ID: 5TEZ [43]). The 5TEZ complex has a high sequence identity of 81% to the best TCR-pMHC template (PDB ID: 5EUO), but the resulting model has a relatively poor accuracy (TCR-pMHC RMSD = 5.66Å). The 5EUO and 5TEZ are both complexes of TCRs bound to the HLA-A2-restricted Influenza A GIL peptides, and hence share 100% identity to the peptide and the MHC. However, the two TCRs are very different (sequence identity of 37% for the α chain and 52% for the β chain), resulting in the TCR of 5TEZ adopting a non-canonical binding orientation to the pMHC [43]. Modelling the 5TEZ structure is, therefore, a highly challenging case, as there are no good templates found in our template database.

Discussion

Here, we present TCRpMHCmodels, an automated pipeline for building structural models of TCR-pMHC complexes. Using as input only the amino acid sequence of a target TCR-pMHC, TCRpMHCmodels automatically identifies the best structural templates, generates the best target-template sequence alignment and builds a structural model of the target using comparative modelling. The structural models have a high quality and are generated within a computational time of only 2 minutes.

It has been suggested that using multiple templates can increase the model accuracy for comparative models [44], especially when modelling protein complexes with multiple chains [45,46]. Using multiple templates is harder than it appears, since finding the optimal combination of templates is non-trivial [47]. Including all suitable template candidates usually leads to accumulation of noise and wrong templates which decreases the model quality [33,46]. However, each additional template increases the probability of detecting a template with the correct structural conformation. Finding the right balance is therefore very important when using multiple templates.

In this study, we have evaluated different template selection methods, including single versus multi template modelling. Comparing single versus multi template modelling of the TCR-pMHC complex, we found that using multiple templates produced the most accurate models. In our multiple template selection method, we always included the template with highest sequence identity; additional templates were added if they had an identity of less than 95% to any template already selected and their identity to the target was at least 80% of the identity of the best scoring template. By doing this, we ended up with a non-redundant list of templates which were then used for the multi template modelling. This both decreases the number of templates used and increases the chance of selecting structures with the correct conformation.

In the present study, we evaluated the effect of using a weighted sequence identity score by changing the weight of the different chains in the TCR-pMHC complex. We here showed that this weighted identity score achieved the best model accuracy, both when modelling the pMHC and the TCR-pMHC complex.

TCRpMHCmodels first models the pMHC and the TCR separately, after which these are assembled in an additional modelling step to form the full TCR-pMHC complex. The reasoning for modelling the TCR and the pMHC as separate units is that there are more structures of the TCR and the pMHC as separate units, than for the full TCR-pMHC complex. By modelling the TCR and the pMHC separately, we have a larger number of templates which can be used in the comparative modelling step, resulting in more accurate models. This is especially true when modelling the TCR CDR loops and the MHC bound peptide, as these parts are more variable and therefore more difficult to model. In the final modelling step, the two models of the TCR and pMHC are used as templates, together with one or more templates of the full TCR-pMHC complex. Using this additional modelling step is a simple way of assembling the

TCR and pMHC, and we here show that this approach gives more accurate models than using the more traditional docking approach.

We compared TCRpMHCmodels with TCRFlexDock [29] and showed that TCRpMHCmodels significantly outperformed TCRFlexDock, both at predicting the full TCR-pMHC complex, the TCR-pMHC interface and the CDR loop conformations. The two methods use a different approach for modelling the TCR-pMHC complex. TCRFlexDock uses a flexible backbone docking protocol based on RosettaDock [48] to perform TCR-pMHC docking and uses ZRANK [39] to identify the best TCR-pMHC complexes. The TCRFlexDock protocol was optimised using structures of crystallized TCR and pMHC complexes for which a crystallized structure of the full TCR-pMHC complex also existed. After optimising the TCRFlexDock protocol, the authors of TCRFlexDock then show how the protocol can also produce accurate TCR-pMHC complexes using TCR and pMHC models instead of crystallized structures. TCRpMHCmodels on the other hand is based on a comparative modelling approach and no explicit docking is performed. To make our tool accessible we have implemented TCRpMHCmodels into a web server, which is both fast and easy to use. In contrast to this, the TCRFlexDock protocol is not readily implemented or available as a web server and is both time and computationally intensive as it takes on average 130 CPU hours to run the complete protocol on a single complex. Finally, as the authors of the TCRFlexDock method mention, using TCR and pMHC models is more challenging than using crystal structures, so one reason for the relatively low accuracy of TCRFlexDock in this study could be that we have here only used TCR and pMHC models rather than crystal structures as the initial input for the TCRFlexDock protocol.

A key factor in determining the accuracy of TCRpMHCmodels is the availability of templates suitable for the comparative modelling steps. The current implementation of TCRpMHCmodels is limited to model structures where the length of the bound peptide matches the length of the structures in the pMHC and TCR-pMHC template databases. In practice, this limits the application of the current tool to only model structures with 8-11mer peptides bound in the peptide-binding groove. Also, the accuracy of the tool was demonstrated (as is the case for all comparative modelling approaches) to depend strongly on the sequence identity between the target entry and the template used for modelling. Due to the availability of only a relatively small number of known structures, this dependency has the most pronounced effect when it comes to the full TCR-pMHC template database. This was demonstrated in the case of the 5TEZ PDB structure, where TCRpMHCmodels was shown to achieve an unexpected low predictive performance imposed by the lack of a suitable TCR-pMHC template sharing the non-canonical TCR binding orientation of 5TEZ. This problem could potentially be resolved in the future by including new TCR-pMHC structures into our internal template database as soon as these are deposited into IEDB (https://www.iedb.org/).

It has been shown that structural features of the pMHC complex can shape the TCR repertoire, indicating that key features for TCR recognition may come from the combined structure of the pMHC complex [49]. Furthermore, it is known that a given TCR has the potential to recognise different pMHC complexes, in a process known as T-cell cross-reactivity [49]. Understanding T-cell cross-reactivity is very important for TCR-based immunotherapies, as cross-reactive T-cells can cause serious or even fatal side effects [50,51]. Unfortunately, the available structural data for cross-reactive TCRs and pMHCs is not large enough to draw any conclusions on the ability of our tool to model such cases, but as an illustrative example, we have included the modelling of some cross-reactive peptides and TCRs in Supplementary Figures S11 and Supplementary Figures S12. In all cases, the accuracy of the models is similar to or marginally worse than the average accuracy of the tool. Future work regarding integration of structural modelling of the TCR-pMHC interaction interface with refined binding energy models might aid in defining such cross-reactivities and allow the development of corresponding predictive models.

Here, we have shown that TCRpMHCmodels generates accurate structural models of the TCR-pMHC complex and that it outperforms TCRFlexDock, a specialised docking protocol for assembling TCR and pMHC molecules. We believe that this work has generated the foundation for future work within the prediction of TCR-pMHC interactions, and we expect the model performance to increase as more structural and sequence data describing TCR-pMHC interactions becomes available.

Method

Template databases

TCRpMHCmodels applies three structural databases which are used for modelling the pMHC, the TCR, and the complete TCR-pMHC complex, respectively. At each step, one or more templates are selected from each database according to their sequence identity. In the sections below, we describe the generation of these structural template databases.

The pMHC database

The pMHC database included 455 non-redundant pMHC structures. The structures found in the database were identified using the Immune Epitope Database (IEDB) [52–54], with a few additional pMHC structures from the Protein Data Bank (PDB) [35]. Using the sequence from these structures we then generated an in-house Hidden Markov Model (HMM) profile for the MHC class I chain. The in-house HMM was generated using the HMMER software (version 3.1) http://hmmer.org/ with the HMM profile from Pfam [55] called MHC_I.hmm (accession number: PF00129). This HMM profile includes the α1 and α2 domains of the MHC class I family. We first used hmmsearch to identify PDB entries with a sequence that matched the HMM profile obtained from Pfam. To remove false positive "hits", we used an E-value threshold of 10^{-5} and only selected entries with a full sequence bit score larger than 250. This yielded 700 PDB entries. All identified entries were then aligned to the MHC class I HMM profile from Pfam using hmmalign to generate a multiple sequence alignment (MSA). We here included the options --trim to exclude residues at the protein terminals that did not fit the HMM model. By performing a manual analysis of the MHC molecules in the database, we found that some of the entries included uncommon insertions at specific positions. These few insertions were primarily found in chicken and canine and, to include these in the HMM, we therefore constructed an in-house HMM profile matching all the identified entries. This new HMM profile was made using hmmbuild using the --symfrac 0 option. The resulting HMM profile contained 181 positions and included all the uncommon insertions, and it identified the same set of pMHC molecules as the original Pfam profile.

The database was next cleaned up by removing pMHC structures without a peptide or with missing residues in the peptide. This reduced the database to 645 pMHC structures. We then used CD-HIT [56] with a global sequence similarity threshold of 100% to ensure that that the final database only contained unique pMHC structures.

The TCR database

The TCR database was obtained from LYRA [18]. This database consisted of 105 paired TCR chains, two individual α chains and nine individual β chain structures. For more details see [18].

The TCR-pMHC database

The TCR-pMHC database included 61 non-redundant TCR-pMHC structures. These were identified using IEDB [52–54], with a few additional structures from the Protein Data Bank (PDB). The additional TCR-pMHC structures were found by aligning each entry in the PDB database to the in-house MHC class I HMM profile, plus the HMM profile for the TCR α and β chain. We then used PISCES server [57] on all the identified TCR-pMHC structures to exclude redundant entries and to remove structures with a resolution above 3Å. Furthermore, we removed PDB structures with missing residues in the peptide.

An overview of the different databases is shown in Figure 8. From Figure 8 A, we observe that the pMHC database contains the largest amount of structures, followed by the TCR database and lastly the TCR-pMHC database. Figure 8 B and C show the distribution of structures based on the length of the peptide found in the pMHC database and the TCR-pMHC database.

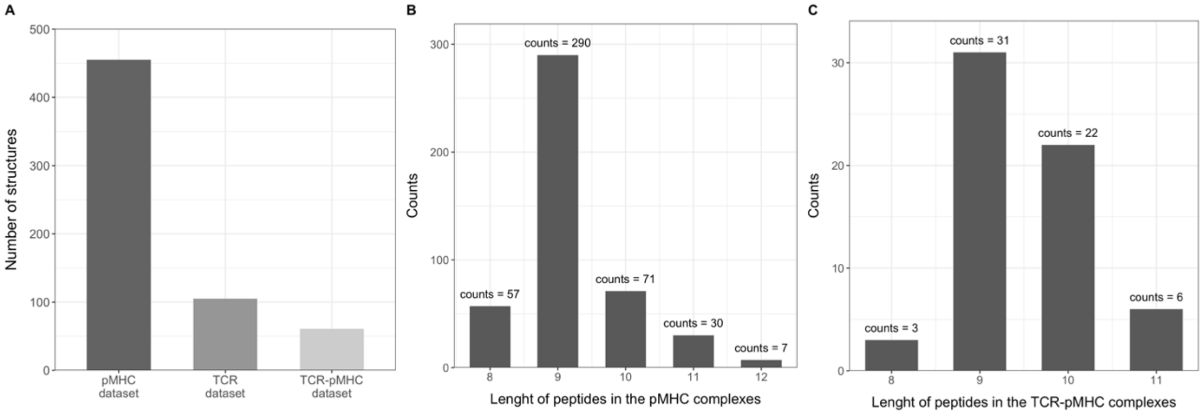


Figure 8: Database visualization. **A)** Number of structures in each template database. **B)** The peptide length distribution of structures in the pMHC database. **C)** The peptide length distribution of structures in the TCR-pMHC database.

Modelling the TCR-pMHC complex

The models produced by the TCRpMHCmodels method were generated using the automodel class from MODELLER v9.18 [36] with default settings. The automodel class takes two inputs: i) one or multiple template structures, and ii) an alignment of the target sequence and the sequence of the selected templates, in PIR format. In TCRpMHCmodels both the template selection and the alignment are generated automatically.

To calculate sequence identities, all sequences were aligned to the most similar HMM profile (either MHC, TCR α or TCR β) before calculating the sequence identity. These alignments were further used in the alignment file, after selecting the best templates. Using the structural templates and the alignment, MODELLER then builds a structural model of the target by optimally satisfying spatial restraints derived from the alignment.

Template selection

Template selection for the pMHC

When modelling the pMHC, we investigated four different template selection methods: i) OneUnweighted, ii) OneWeighted, iii) MultiWeighted and iv) MultiUnweighted. In each of these template selection methods, we only used templates with the same peptide length. Using the in-house MHC class I HMM profile, we aligned the MHC chain of the target to templates found in the pMHC database (see pMHC database for further details about the class I HMM profile). The alignment was generated using hmmalign with the --trim option. After the alignment, we calculated the sequence identity between the target and each of the templates, excluding all insertions.

In the OneUnweighted method, the sequence identity was calculated by summing the identities from the peptide and the MHC alignment, dividing with sum of the peptide and HMM alignment lengths (excluding gaps). The templates were next sorted based on the sequence identity and the template with the highest sequence identity used for modelling.

The MultiUnweighted method uses the same approach to calculate the sequences identity, but instead of selecting only the single template with the highest identity score, this method selects multiple templates for model building. The selection of these multiple templates was done using a Hobohm1-like [58] approach similarly to what we have described earlier [33] by first sorting the templates according to sequence identity as described for method OneUnweighted. Next, the template with the highest sequence identity was selected and considered the best template. This template was always included. Next, looping through the sorted template list, additional templates were included if 1) they had an identity of less than 95% to any template already selected, and 2) their identity to the target was at least 80% of the identity of the top scoring best template.

In the OneWeighted method, the sequence identity measure was calculated by introducing a weight to the peptide and MHC sequence so that they each contributed equally to the sequence identity. Using the weighted sequence identity, we then selected the single template with the highest sequence identity.

In the MultiWeighted method, we used the weighted sequence identity, but selected multiple templates with the algorithm described above for the MultiUnweighted method.

Template selection for the TCR-pMHC

When modelling the TCR-pMHC, we investigated three different template selection methods, named: i) OneUnweighted, ii) OneWeighted and iii) MultiWeighted. These three methods were performed as explained above, the only difference was the calculation of the weighted sequence identity. Here, the weighted sequence identity was calculated by introducing a weight to the peptide, MHC, TCR α and TCR β sequence so that the peptide and the MHC contributed to ⅓ of the sequence identity, while the TCR α and TCR β contributed to ⅙ of the sequence identity, respectively.

Model validation

In order to assess the model accuracy of TCRpMHCmodels in situations where the structure is not known, we performed a leave-one-out (LOO) assessments of all structures in our template database. For each structure in the template database, we removed that structure from the database and built a structural model using the remaining templates. To further increase the model variability in terms of sequence identity of the adopted templates, we furthermore imposed four different template-target sequence identity thresholds of 99.9%, 95%, 90% and 80%, thereby removing any template having an identity higher than the selected threshold. For each structure in the template database, we therefore generated four different structural models imposing the four different sequence identity thresholds. The resulting models were then evaluated by comparing them to their native structure from the template database. This LOO assessments was used to evaluate the performance of both the pMHC and the final TCR-pMHC models.

Random performance

In order to estimate a baseline performance value, we, for each structure in the template database, randomly selected another template from the database using the four different sequence identity thresholds as described above.

TCRFlexDock

Protein-protein docking is a common method for assembling multi-chain proteins [59], [60]. Thus, we compared TCRpMHCmodels with the CDRPep protocol from TCRFlexDock protocol [29]. TCRFlexDock is a specialised protein docking method for predicting the correct orientation between the TCR and pMHC molecules. While docking, the TCRFlexDock CDRPep protocol allows for some flexibility in the CDR loops and the MHC bound peptide. For each TCR-pMHC complex in the TCR-pMHC template database we modelled the TCR and the pMHC, and we then used the TCRFlexDock protocol to assemble these two models. As described in the TCRFlexDock protocol, we created 1000 docking decoys and each decoy was subsequently scored using ZRANK [39]. The decoy with the lowest ZRANK score was selected as the best TCR-pMHC complex from the protocol, and this complex was evaluated by comparing it to the native structure from the template database.

Model performance

To assess the quality of the structural models, we used the root-mean-square deviation (RMSD) between the Cα atoms from the model and the Cα atoms in the native structure from the template database, after making a structural alignment of the model and the template. The structural alignment was made using the Superimposer class from BioPython [61] minimising the distance between the Cα atoms in the model and the structural template before calculating the RMSD. To evaluate the model accuracy of the different parts in TCR-pMHC complex we generated four RMSD values, the TCR-pMHC RMSD, the TCR RMSD, the pMHC RMSD and the peptide RMSD. Each of these RMSD values were defined by calculating the RMSD after structural alignment of the different TCR-pMHC, the TCR, the pMHC and the peptide, respectively.

After superimposing, we used the template modelling score (TM-score), calculated as described by Y. Zhang et. al. [62] between all the Cα atoms. The TM-score is a length-independent metric used for measuring structural similarity between two proteins. The TM-score ranges between 0 and 1, where a TM-score of 1 indicates a perfect match between two structures. A TM-score below 0.2 corresponds to randomly choosing an unrelated protein and a TM-score higher than 0.5 assumes that the two structures roughly have the same structural fold.

To evaluate the model quality of the docking models from the TCRFlexDock protocol, we used the DockQ score calculated with the DockQ tool [40]. The DockQ score is made by combining Fnat, LRMS, and iRMS to a single score ranging between 0 and 1 that can be used to assess the quality of protein docking models.

References

1. Zhang, N. & Bevan, M. J. CD8 + T Cells: Foot Soldiers of the Immune System Introduction to Cytotoxic T Cells. *Immunity* **35,** 161–168 (2011).
2. Rudolph, M. G., Stanfield, R. L. & Wilson, I. A. How TCRs Bind MHCs, Peptides, and Coreceptors. *Annu. Rev. Immunol.* **24,** 419–466 (2006).
3. Restifo, N. P., Dudley, M. E. & Rosenberg, S. A. Adoptive immunotherapy for cancer: Harnessing the T cell response. *Nat. Rev. Immunol.* **12,** 269–281 (2012).
4. Sette, A. & Fikes, J. Epitope-based vaccines: An update on epitope identification, vaccine design and delivery. *Curr. Opin. Immunol.* **15,** 461–470 (2003).
5. Koup, R. A. *et al.* Vaccine Design for CD8 T Lymphocyte. 1–16 (2011). doi:10.1101/cshperspect.a007252
6. Schubert, B., Lund, O. & Nielsen, M. Evaluation of peptide selection approaches for epitope-based vaccine design. *Tissue Antigens* **82,** 243–251 (2013).
7. Morgan, R. A. *et al.* Cancer Regression in Patients After Transfer of Genetically Engineered Lymphocytes. *Science (80-.).* **314,** 126–129 (2006).
8. Rosenberg, S. A., Restifo, N. P., Yang, J. C., Morgan, R. A. & Dudley, M. E. Adoptive cell transfer: A clinical path to effective cancer immunotherapy. *Nat. Rev. Cancer* **8,** 299–308 (2008).
9. Chen, J.-L. *et al.* Structural and kinetic basis for heightened immunogenicity of T cell vaccines. *J. Exp. Med.* **201,** 1243–1255 (2005).
10. Neefjes, J., Jongsma, M. L., Paul, P. & Bakke, O. Towards a systems understanding of MHC class I and MHC class II antigen presentation. *Nat Rev Immunol* **11,** 823–36 (2011).
11. La Gruta, N. L., Gras, S., Daley, S. R., Thomas, P. G. & Rossjohn, J. Understanding the drivers of MHC restriction of T cell receptors. *Nat. Rev. Immunol.* **18,** 467–478 (2018).
12. Davis, M. M. & Bjorkman, P. J. T-cell antigen receptor genes and T-cell recognition. *Nature* **335,** 744 (1988).
13. Marrack, P. & Kappler, J. The T Cell Receptor. *Science (80-.).* **238,** 1073–1079 (1987).
14. Burrell, C. J., Howard, C. R. & Murphy, F. A. in *Fenner and White's Medical Virology* 65–76 (2017). doi:10.1016/B978-0-12-375156-0.00006-0
15. Qi, Q. *et al.* Diversity and clonal selection in the human T-cell repertoire. *Proc. Natl. Acad. Sci.* **111,** 13139–13144 (2014).
16. De Simone, M., Rossetti, G. & Pagani, M. Single Cell T Cell Receptor Sequencing: Techniques and Future Challenges. *Front. Immunol.* **9,** 1638 (2018).
17. Nikolich-Žugich, J., Slifka, M. K. & Messaoudi, I. The many important facets of T-cell repertoire diversity. *Nat. Rev. Immunol.* **4,** 123–132 (2004).
18. Klausen, M. S., Anderson, M. V., Jespersen, M. C., Nielsen, M. & Marcatili, P. LYRA , a webserver for lymphocyte receptor structural modeling. *Nucleic Acids Res.* **43,** W349–W355 (2015).
19. Krawczyk, K., Kelm, S., Kovaltsuk, A., Galson, J. D. & Deane, C. M. Structurally Mapping Antibody Repertoires. *Front. Immunol.* **9,** 1698 (2018).
20. Al-Lazikani, B., Lesk, a M. & Chothia, C. Standard conformations for the canonical structures of immunoglobulins. *J. Mol. Biol.* **273,** 927–948 (1997).
21. Andreatta, M. & Nielsen, M. Gapped sequence alignment using artificial neural networks: Application to the MHC class i system. *Bioinformatics* **32,** 511–517 (2015).
22. Jurtz, V. *et al.* NetMHCpan-4.0: Improved Peptide− MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *J. Immunol.* **199,** 3360–3368 (2017).
23. Kim, Y., Sidney, J., Pinilla, C., Sette, A. & Peters, B. Derivation of an amino acid similarity matrix for peptide:MHC binding and its application as a Bayesian prior. *BMC Bioinformatics* **10,** 1–11 (2009).
24. O'Donnell, T. J. *et al.* MHCflurry: Open-Source Class I MHC Binding Affinity Prediction. *Cell Syst.*

7, 129–132 (2018).

25. Khan, J. M. & Ranganathan, S. PDOCK: A new technique for rapid and accurate docking of peptide ligands to Major Histocompatibility Complexes. *Immunome Res.* **6,** 1–16 (2010).
26. Antes, I., Siu, S. W. I. & Lengauer, T. DynaPred: A structure and sequence based method for the prediction of MHC class I binding peptide sequences and conformations. *Bioinformatics* **22,** 16–24 (2006).
27. Menegatti Rigo, M. *et al.* DockTope: a Web-based tool for automated pMHC-I modelling. *Sci. Rep.* **5,** 18413 (2015).
28. Gowthaman, R. & Pierce, B. G. TCRmodel: High resolution modeling of T cell receptors from sequence. *Nucleic Acids Res.* **46,** W396–W401 (2018).
29. Pierce, B. G. & Weng, Z. A flexible docking approach for prediction of T cell receptor-peptide-MHC complexes. *Protein Sci.* **22,** 35–46 (2013).
30. Liu, I. H., Lo, Y. S. & Yang, J. M. Genome-wide structural modelling of TCR-pMHC interactions. *BMC Genomics* **14,** S5 (2013).
31. Hoffmann, T., Marion, A. & Antes, I. DynaDom: structure-based prediction of T cell receptor inter-domain and T cell receptor-peptide-MHC (class I) association angles. *BMC Struct. Biol.* **17,** 2 (2018).
32. Liu, I. H., Lo, Y. S. & Yang, J. M. PAComplex: A web server to infer peptide antigen families and binding models from TCR-pMHC complexes. *Nucleic Acids Res.* **39,** 254–260 (2011).
33. Lanzarotti, E., Marcatili, P. & Nielsen, M. Identification of the cognate peptide-MHC target of T cell receptors using molecular modeling and force field scoring. *Mol. Immunol.* **94,** 91–97 (2018).
34. Bentzen, A. K. *et al.* T cell receptor fingerprinting enables in-depth characterization of the interactions governing recognition of peptide-MHC complexes. *Nat. Biotechnol.* **36,** 1191–1196 (2018).
35. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28,** 235–242 (2000).
36. Sali, A. & Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234,** 779–815 (1993).
37. Fiser, A. & Andrej, S. MODELLER: generation and refinement of homology-based protein structure models. *Methods Enzymol.* **374,** 461–491 (2003).
38. Chothia, C. & Lesk, A. M. The relation between the divergence of sequence and structure in proteins. *EMBO J.* **5,** 823–6 (1986).
39. Pierce, B. & Weng, Z. ZRANK: reranking protein docking predictions with an optimised energy function. *Korea Obs.* **67,** 1078–1086 (2008).
40. Basu, S. & Wallner, B. DockQ: A quality measure for protein-protein docking models. *PLoS One* **11,** 1–9 (2016).
41. Lensink, M. F. & Wodak, S. J. Docking, scoring, and affinity prediction in CAPRI. *Proteins Struct. Funct. Bioinforma.* **81,** 2082–2095 (2013).
42. Lensink, M. F., Méndez, R. & Wodak, S. J. Docking and scoring protein complexes: CAPRI 3rd Edition. *Proteins Struct. Funct. Genet.* **69,** 704–718 (2007).
43. Yang, X., Chen, G., Weng, N. ping & Mariuzza, R. A. Structural basis for clonal diversity of the human T-cell response to a dominant influenza virus epitope. *J. Biol. Chem.* **292,** 18618–18627 (2017).
44. Larsson, P., Wallner, B., Lindahl, E. & Elofsson, A. Using multiple templates to improve quality of homology models in automated homology modeling. *Protein Sci.* **17,** 990–1002 (2008).
45. Cheng, J. A multi-template combination algorithm for protein comparative modeling. *BMC Struct. Biol.* **8,** 1–13 (2008).
46. Meier, A. & Söding, J. Automatic Prediction of Protein 3D Structures by Probabilistic Multi-template Homology Modeling. *PLoS Comput. Biol.* **11,** 1–20 (2015).
47. Fernandez-Fuentes, N., Rai, B. K., Madrid-Aliste, C. J., Eduardo Fajardo, J. & Fiser, A. Comparative protein structure modeling by combining multiple templates and optimizing sequence-to-structure alignments. *Bioinformatics* **23,** 2558–2565 (2007).

48. Chaudhury, S. *et al.* Benchmarking and analysis of protein docking performance in Rosetta v3.2. *PLoS One* **6,** (2011).

49. Antunes, D. A., Rigo, M. M., Freitas, M. V & Mendes, M. F. A. Interpreting T-Cell Cross-reactivity through Structure: Implications for TCR-Based Cancer Immunotherapy. *Front. Immunol.* **8,** 1210 (2017).

50. Linette, G. P. *et al.* Cardiovascular toxicity and titin cross-reactivity of affinity-enhanced T cells in myeloma and melanoma. *Blood* **122,** 863–871 (2013).

51. Tcr, A. *et al.* Cancer regression and neurologic toxicity following anti-MAGEA3 TCR gene therapy. *J. Immunother.* **36,** 133–151 (2013).

52. Vita, R. *et al.* The immune epitope database (IEDB) 3.0. *Nucleic Acids Res.* **43,** D405–D412 (2015).

53. Ponomarenko, J. *et al.* IEDB-3D: Structural data within the immune epitope database. *Nucleic Acids Res.* **39,** 1164–1170 (2011).

54. Mahajan, S. *et al.* Epitope specific antibodies and T cell receptors in the Immune Epitope Database. *Front. Immunol.* **9,** 2688 (2018).

55. Finn, R. D. *et al.* Pfam: the protein families database. *Nucleic Acids Res.* **42,** D222–D230 (2014).

56. Li, W. & Godzik, A. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22,** 1658–1659 (2006).

57. Wang, G. & Dunbrack, R. L. PISCES: A protein sequence culling server. *Bioinformatics* **19,** 1589–1591 (2003).

58. Hobohm, U., Scharf, M., Schneider, R. & Sander, C. Selection of representative protein data sets. *Protein Sci.* **1,** 409–417 (1992).

59. Smith, G. R. & Sternberg, M. J. E. Prediction of protein-protein interactions by docking methods. *Curr. Opin. Struct. Biol.* **12,** 28–35 (2002).

60. Gray, J. J. High-resolution protein-protein docking. *Curr. Opin. Struct. Biol.* **16,** 183–193 (2006).

61. Cock, P. J. A. *et al.* Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25,** 1422–1423 (2009).

62. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins Struct. Funct. Genet.* **57,** 702–710 (2004).

Acknowledgements

This work was supported in part by the Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN272201200010C.

Author Contributions

K.K.J, V.R, M.N and P.M conceived the study. K.K.J, M.N and P.M wrote the paper. K.K.J, V.R, S.M and B.P created the template databases. K.K.J, V.R and Esteban developed the scripts for the automated modelling tool. K.K.J and L.E.J preparation of figures. E.J, K.K.J and V.J performed the TCRFlexDock analysis. T.H.O performed the DockQ analysis. K.K.J and M.C.J were responsible for web-server development. All authors analysed and interpreted the data. All authors revised the manuscript and approved the final version to be published.

Competing financial interests

The authors declare no competing financial interests.

Chapter 4: Paper III

In this chapter, we present an experimental method for determining the exact molecular interaction points of a TCR with peptides presented in an MHC molecule. The experimental method measures the relative binding affinity between clonal TCRs and different peptide-MHC variants. The peptide-MHC variants are constructed from an identified TCR target, and includes single amino acid variations of this peptide. For each peptide variant individual DNA barcode-labeled MHC multimers are generated, pooled and incubated with the clonal T-cells. After incubation, the DNA barcodes are sequenced and the distribution and relative counts of the barcode reads reflects the TCR-pMHC binding hierarchy. To determine the TCR interaction points, the TCR binding can be translated into a TCR motif (referred to as a TCR fingerprint). The TCR fingerprint illustrates the amino acids essential for the interaction between the TCR and the peptide-MHC complex.

In this chapter, we use this experimental method to investigate different TCR-pMHC interactions. This was done by generating TCR fingerprints of TCRs binding to different peptide-MHC variants and utilizing these to predict and validate cross-recognized peptides from the human proteome. This suggests that the TCR fingerprints presented here, can be used both as a screening tool for understanding the molecular interactions of TCRs and for selecting TCRs intended for adoptive T-cell therapy.

To gain a deeper understanding of the peptide-MHC binding specificity, we also generated structure-based models of the identified pMHC target and used these models to investigate how the different peptide variations affect the peptide-MHC binding.

This project was carried out in a collaboration with Sine Reker Hadrup's group at DTU. The underlying experimental strategy was predominantly developed by Amalie K. Bentzen and Andrea M. Marquard and the experimental results shown in the paper were predominantly performed by Lina Such and Amalie K. Bentzen. My primary contribution to the work was the visualization of the experimental results, generation of sequence motifs, as well as the generation of the structural models used for understanding of the peptide-MHC binding specificity.

TEXT FOR ONLINE VERSION

In this online version of the thesis, paper III is not included but can be obtained from electronic article databases or on request from:

DTU Health Tech Technical
University of Denmark
Kemitorvet, Building 202
2800 Kongens Lyngby
Denmark

healthtech-info@dtu.dk

Online papers can be obtained through the following link:
Paper III: https://www.nature.com/articles/nbt.4303

Chapter 5: Paper IV

Many methods for modeling the structure of BCRs and TCRs exist, but due to the high variability in particular of the CDR3 loop in these receptors, the structural accuracy for modeling this loop is usually low [64]. Traditionally these CDR loops are modeled using either template-based or *ab initio* methods. The main limitation with the template-based methods is that they rely on the limited amount of experimentally determined structures. Given the diversity of the CDR3 loops in terms of structure, sequence and length, these experimentally determined structures cover only a marginal fraction of all possible loop conformations. The biggest limitation of the *ab initio* methods is that they are slow, as they usually have to generate many possible loop conformations, which then need to be ranked to find the best loop. In this project, we wanted to improve the speed and accuracy of the *ab initio* methods, by using a recently developed deep neural network architecture called generative adversarial network (GAN). The main idea is to teach the GAN how to generate accurate and diverse CDR3 loops using the dihedral angles, from which the loop structure can be built.

In this chapter, we present the preliminary results from this project.

Structural modeling of lymphocyte receptor loops using Generative Adversarial Networks

Kamilla Kjærgaard Munk, Morten Nielsen, and Paolo Marcatili
Department of Bio and Health Informatics, Technical University of Denmark, Kgs. Lyngby, Denmark

Abstract

The antigen-binding sites of B-cell receptors (BCRs) and T-cell receptors (TCR) consist of hypervariable loops, known as complementarity-determining regions (CDRs). In particular, the CDR3 loop of the heavy chain for BCRs and the CDR3 loop of the beta chain for TCRs, are the most variable parts of the two receptor systems, both in terms of sequence variability, length, and conformation. Because of this, current loop modeling techniques usually fail at building high-quality structural models for these loops. Here, we use Generative Adversarial Networks (GANs) to generate accurate and diverse structures of the CDR3 loop given only the sequence. The GAN is trained using the backbone dihedral angles from the CDR3 loop structures, and the preliminary results show that it is possible to generate diverse CDR3 loop structures, but that the network still needs to learn essential structural features, such as loop closure.

Introduction

The accurate structural modeling of B-cell receptors (BCRs) and T-cell receptors (TCRs) is fundamental to gaining a detailed insight into the mechanisms underlying immunity, understanding the B- and T-cell receptor specificity towards their cognate targets, and to develop new drugs and therapies [1–4]. The binding sites of both BCRs and TCRs comprises six loops called complementarity determining regions (CDRs). Each of the chains contribute with three loops, named CDR1, CDR2 and CDR3. The CDR3 loop of the heavy chain in BCRs (denoted H3) and the CDR3 loop of the beta chain in TCRs (denoted B3), comprise the antigen-binding site and they have the largest sequence diversity [5, 6]. Previous studies have shown that five of the six CDR loops mainly assume limited structural conformations, named canonical structures, which can be used to predict the structure of these loops [7–9]. However, the H3 and B3 loop defy these standard classification attempts, and the model accuracy for these loops is therefore usually much lower compared to the remaining CDR loops [10]. The main problem when modeling the CDR3 loop arises due to the variability in loop lengths and the high diversity of both the sequence and structure. The traditional way of modeling CDR loops is based on template-based modeling, which uses experimentally determined structures as templates for modeling. The problem with this type of method is that it relies on the limited amount of experimentally known structures for BCR and TCR available in the Protein Data Bank [11]. Given the diversity of structure, sequence and length of the CDR3 loops, these known structures cover only a marginal fraction of all possible loop conformations.

Another way to model the CDR loops is to use *ab initio* methods. These methods are template-free and seek to predict the loop conformation without the use of structural templates. *Ab initio* methods generate possible loop conformations (called decoys) by sampling the conformational space, after which the generated decoys are ranked using energy functions or the size of conformational clusters to select the best decoy. Methods that combine *ab initio* and template-based techniques also exist. Specific tools for predicting both TCR and BCR structures include LYRA [9], for predicting the TCR structures there is TCRmodels [12], and tools for predicting BCR structures include RosettaAntibody [13], ABodyBuilder [14] and PIGS [15, 16]. The currently available tools for predicting the CDR3 loop found in BCR structures are H3Loopred [17] and the H3-specific version of Sphinx [18].

When modeling TCR and BCR structures, it is only the CDR regions that pose a significant challenge, since these vary greatly from case to case. The rest of the molecule, called the framework, is highly conserved. Therefore, when predicting a CDR loop structure it is important to keep in mind that the generated loop has to eventually be connected to the framework. The framework residues at which the loop is attached are termed anchor residues, and if the generated loop cannot be connected to the anchors in a geometrically consistent way, the loop structure has to be adjusted to fit this requirement. This problem is referred to as the loop closure problem and is usually solved with loop closure algorithms, such as cyclic coordinate descent (CCD) [19], random tweak [20] or kinematic closure (KIC) [21].

Algorithms which use an *ab initio* approach have two major limitations. The first limitation is that they are computationally demanding as generating enough decoys to sufficiently sample the conformational space of a single loop takes time. As an implication of this, *ab initio* prediction accuracy usually decreases as the loop length increases. This is because the number of degrees of freedom increases with the loop length.The second problem is that even if the conformational space can be adequately explored and decoys that are close to the loop native conformation can be generated, an accurate energy or scoring function is required to select the best from a multitude of decoys. In recent years, thanks to improvements in the energy functions [22] and by exploiting deep learning techniques [23], consistent improvements in the accuracy of these functions has been observed, however, even with these advances the accuracy remains moderate.

To solve the first limitation of the *ab initio* methods, we here use generative adversarial networks (GANs) to generate diverse and accurate decoys of the CDR3 loop in a fast and efficient way. The GAN architecture was originally proposed by Goodfellow *et al. [24]*, but since then many different GAN implementations and network architectures have been proposed [25–29]. Here, we used the WGAN-GP architecture proposed by Ishaan Gulrajani *et al.* [29] to generate structures of CDR3 loops of BCRs and TCRs from the primary protein sequence.

Method

Dataset

A structural dataset consisting of 330 TCR structures and 2,529 BCR structures was obtained by using TCR and BCR specific Hidden Markov Model (HMM) profiles from LYRA [9] to scan the Protein Data Bank (PDB) [11] and identify the CDR3 sequences. All structures with missing atoms in the CDR3 loop structure were removed, as well as CDR3 structures with uncommon characteristics, such as cis peptide bonds and structures with unconventional loop closure. For the BCR dataset, structures with CDR loops shorter than 3 residues were also removed. To avoid redundancy, the same CDR3 sequence was only allowed to appear multiple times in the dataset if the corresponding loop structures differed. To be precise, we use the TM-score as a measure of structural similarity, and only included identical CDR3 sequences in the dataset if the TM-score between its corresponding structure, and any other in the dataset with the same sequence, was greater than 0.6.

Lastly, we generated an independent benchmark set for model evaluation. For the BCR structures, we used the Rosetta Antibody (RA) benchmark dataset [30] and for the TCR, we used any structure released into the PDB after the 1st of August 2018. To avoid overfitting, all CDR3 structures with sequences identical to any of the CDR3 structures found in the benchmark set were also removed from the final dataset. As a result, we were left with 129 TCR structures and 1339 BCR structures for network development.

Feature Extraction

For each residue in the CDR3 structure, we calculated the backbone dihedral angles phi (φ), psi (ψ), omega (ω), and the distance between the C_α atom of that given residue to the C_α atom of the C- and N-terminal residues. All feature extractions were performed using BioPython [31]. The φ, ψ and ω angles were each encoded as a vector of length 2, where the first element is the sine of the angle and the second element is the cosine. This encoding reduces the effect of the periodicity of the angles [32].

Network architecture

In this study, we decided to use the WGAN-GP method as proposed by Ishaan Gulrajani *et al.* [29], since this method has been shown to dramatically improve the stability of learning, while reducing the risk of mode collapse. The WGAN-GP models were built in Python 3.6 using Pytorch [33]. A GAN network consists of two networks, called the Generator and the Discriminator. The Generator is trained to generate samples, while the Discriminator tries to determine if a given sample is made by the Generator or is a true sample from the dataset. In our case, the Generator has two inputs consisting of the CDR3 loop sequence and some random noise, and for each residue in the input sequence, the Generator outputs five values, consisting of three angles, φ, ψ, ω, and two distances, one to the C-terminal and one to the N-terminal. The input to the Discriminator is either artificially produced by the Generator or true samples from the dataset, while the output is a single value between 1 and 0 determining if a given input is real or fake [24]. During training, the two networks are fine-tuned by adjusting network parameters. The Discriminator is updated to get better at discriminating between real and fake samples, while the Generator is updated to produce real looking samples. A schematic representation of the training process and the specific network architecture of the Generator and the Discriminator can be found in Figure 1.

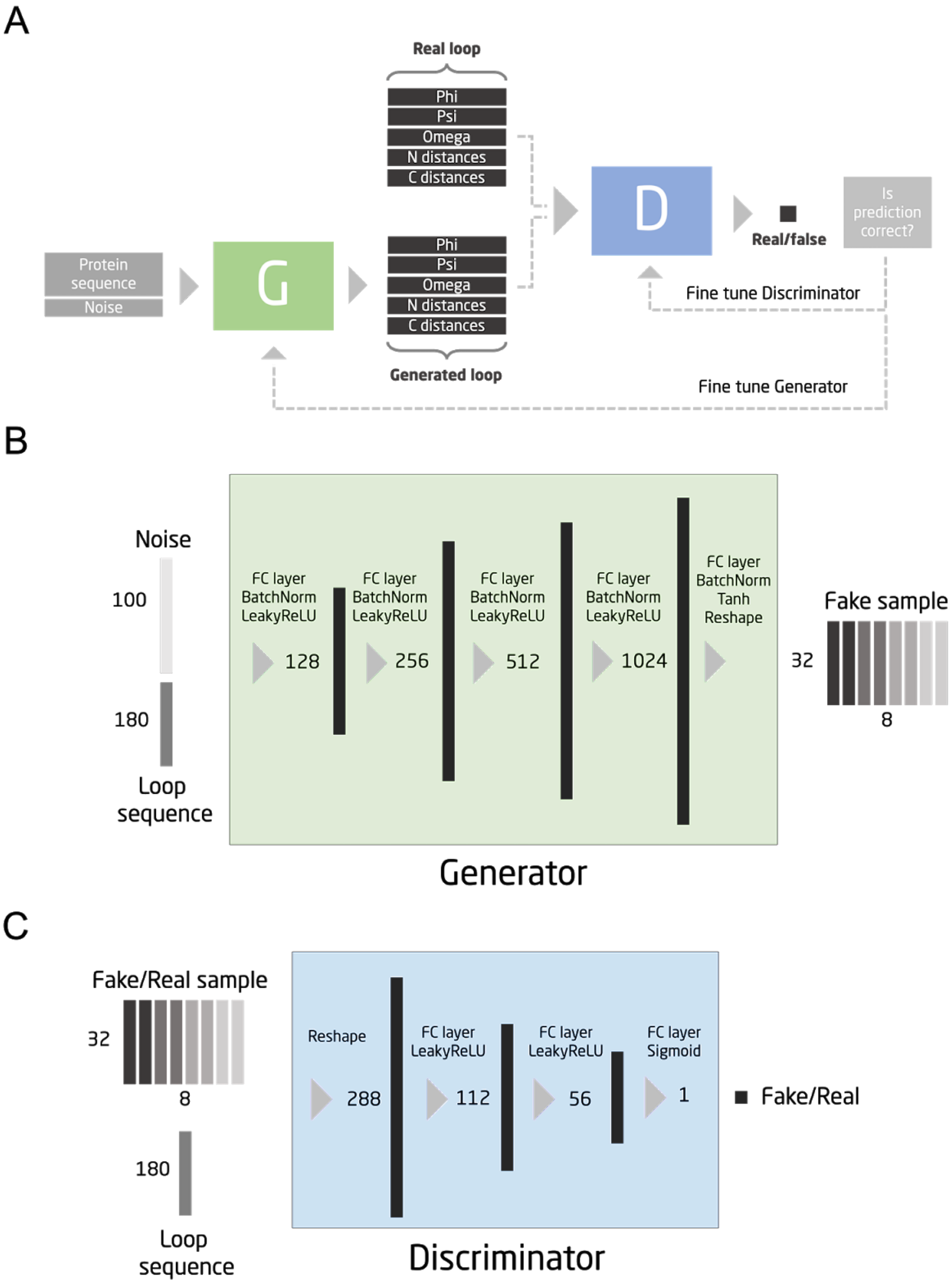


Figure 1: Network training process and architecture for the WGAN-GP model. **A)** The general method of training the WGAN-GP network. **B)** Network architecture for the Generator and Discriminator. Each vertical box in the network represents the different layers and the corresponding number of neurons of that layer.

The input layer of the Generator model consists of a one-hot encoded CDR3 sequence (the maximum length of the CDR3 sequence is 32, the size of the amino acid vocabulary is 20, totaling 180 inputs) plus the noise generated using 100 random numbers drawn from a standard normal distribution, giving a total of 280 inputs. These inputs are then connected to an artificial neural network (ANN) with four fully connected (FC) layers of size 128, 256 and 512, 1024, using the LeakyReLU activation function, and an output layer of size 256 using the Tanh activation function. The output includes the dihedral angles φ, ψ, and ω, for each residue in the CDR3 loop sequence (the maximum length is 32, there are 3 angles, each encoded with sine and cosine, totaling 192 outputs) and the C- and N-terminal C_α distances (the maximum length is 32, and there are two distances, totaling 64 outputs), giving a total of 256 outputs.

The input layer of the Discriminator model consists of either the fake sample (the predicted output from the Generator) or a real sample, plus the one-hot encoded CDR3 sequence (the maximum length is 32, the size of the amino acid vocabulary is 20, totaling 180 inputs), giving a total of 436 inputs. This input is then connected to an ANN with two FC layers of size 112 and 56, using the LeakyReLU activation function and an output layer of size 1 using the Sigmoid activation function. The Discriminator returns a single output value between 0 and 1, which describes the probability of the given input being real. When training GANs, the discriminator seeks to maximize the probability of correctly classifying real and fake samples, while the generator seeks to fool the Discriminator by generating samples that look real.

Network training

Before training, the dataset was randomly divided such that 80% of the dataset was chosen for training, leaving the remaining 20% of the dataset for validation. By doing this we are able to find optimal hyperparameters and test different network architectures. The final network can then be evaluated on the independent benchmark set, leaving the reported performances of this unbiased.

For training, we used batch normalization with mini-batches of size 30 and the individual learning rate of each neuron was optimized using the Adam function [34]. The WGAN-GP model was trained over a series of epochs (full pass over the training set), measuring the model performance using only the first mini-batch from the training and validation set for each epoch (See Figure 3).

Loop building

There are three backbone dihedral angles in the protein chain, φ, ψ and ω, from which φ and ψ can essentially determine the backbone geometry of the protein chain. This is because ω is restricted by the planarity of the peptide bond it is usually fixed around 180° [35]. From the predicted backbone dihedral angles, we used PeptideBuilder [36] to reconstruct the loop structure.

Evaluation

The final WGAN-GP models were tested on the independent TCR and BCR benchmark dataset, which contained only unique CDR3 loop sequences with no overlap to the CDR loops present in the validation data.

We used different metrics to capture the overall quality of the predicted CDR3 loop structures: The mean absolute error (MAE) in degrees for φ and ψ angles, the root mean squared deviation (RMSD) between all backbone atoms and the distance between the C_α atom of the N- and C-terminal residue.

To visualize energetically allowed regions for φ and ψ angles of amino acid residues in the protein structure, we used a Ramachandran plot as described by Ramachandran *et al.* [37].

We also investigated how much the predicted angles differ when the Generator is given the same input sequence but different random noise. We therefore produced 50 different outputs for each loop sequence in the benchmark dataset and calculated the standard deviation of the φ and ψ angles for each residue in each sequence.

It takes around 4 seconds to generate 50 outputs, so it would take less than 2 minutes to generate 1000 possible loop conformations, which is the most common number of decoys for *ab initio* structure prediction methods. For comparison, most of these *ab initio* methods takes hours or even days to generate that many decoys [21, 38, 39].

Results

The WGAN-GP network is trained to generate backbone dihedral angles, φ and ψ, for each residue of the CDR3 sequence, from which the loop structure is built using PeptideBuilder [36].

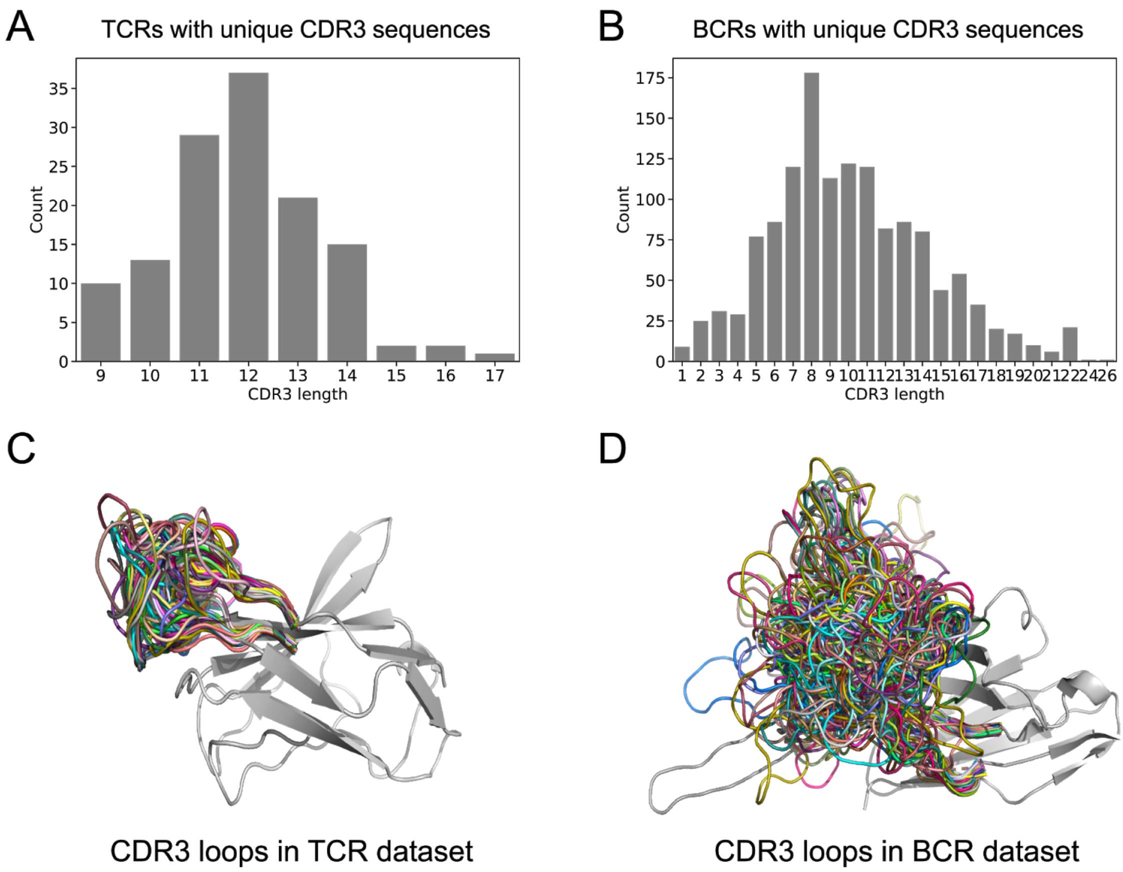


Figure 2: Dataset visualization. Panel **A)** and **B)** show CDR3 loop length distribution of TCR and BCR structures, respectively. Panel **C)** and **D)** show the conformational space of CDR3 loops in TCR and BCR structures, respectively.

The two datasets used for training the WGAN-GP network contain 129 TCR CDR3 structures and 1339 BCR CDR3 structures. Panel A and B in Figure 2 show that CDR3 loops from the BCR dataset are both shorter and longer than CDR3 loops found in the TCR dataset, while panel C and D show the conformational space of CDR3 loops in both TCR and BCR structures. From Figure 2 it is evident that CDR3 loops are diverse both in loop length and structure.

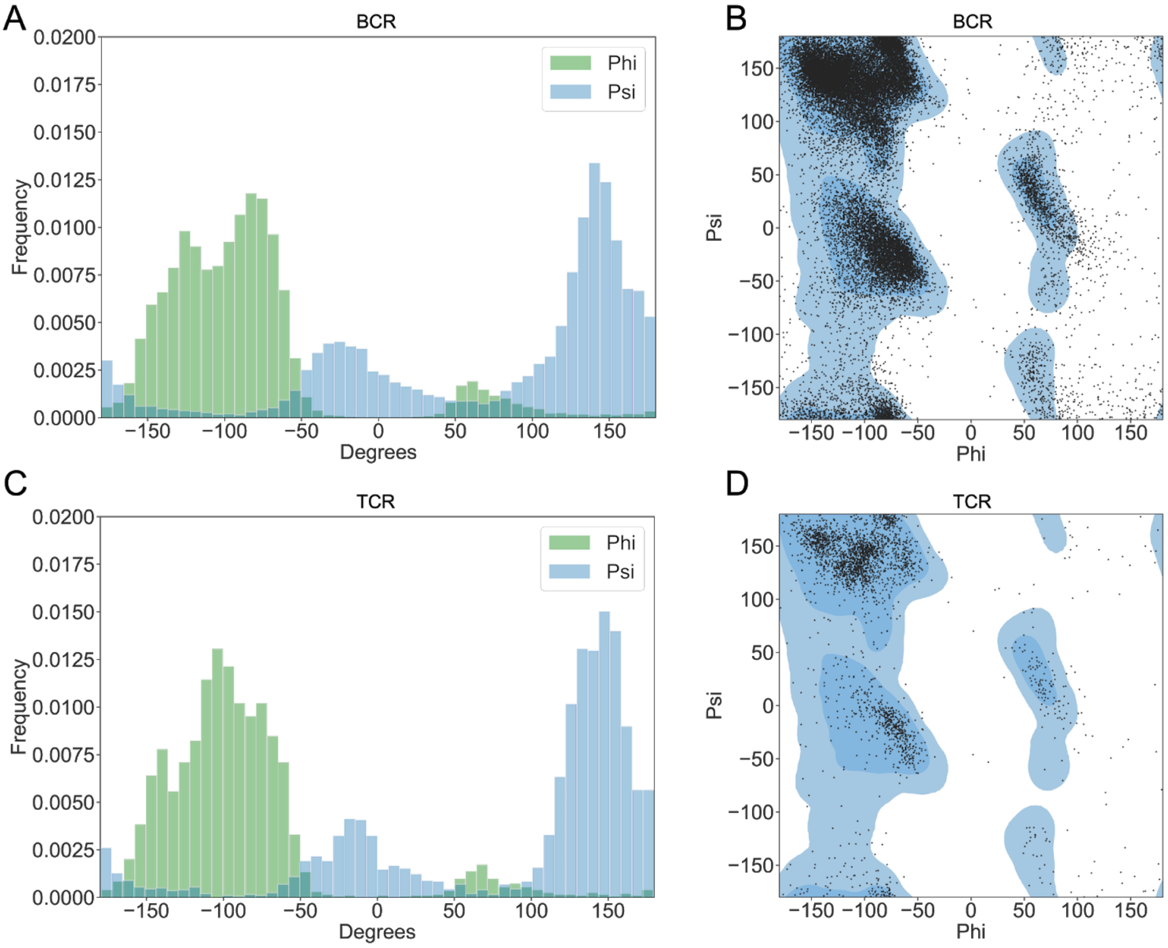


Figure 3: The distribution of the backbone φ (phi) and ψ (psi) angles and the Ramachandran plot for the CDR3 loops from the TCR and BCR dataset. **A)** Distribution of angles in the BCR dataset. **B)** The Ramachandran plot for the BCR dataset. **C)** Distribution angles in the TCR dataset. **D)** The Ramachandran plot for the TCR dataset. In the ramachandran plot dark and light blue represent favoured and allowed regions for the φ and ψ angles, based on the findings by *Lovell et. al. [40]*.

The φ and ψ angles for the TCR and BCR loops used for network training are displayed in Figure 3. From panel A and C, it is observed that the φ and ψ angles in BCR and TCR CDR3 loops follow the same distribution, while φ angles have a major peak around -100° and a small peak around 70°, and ψ angles have a major peak around 140° and a smaller peak around -30°. Furthermore, the ψ angles seem to have a broader distribution than the φ angles. Panel B and C illustrate that most of the φ and ψ angles are found in the favoured regions in the Ramachandran plot, indicating that the backbone angles of the CDR3 loops are similar to those of other proteins. When training the WGAN-GP, we expect that the network will learn this background distribution of backbone angles.

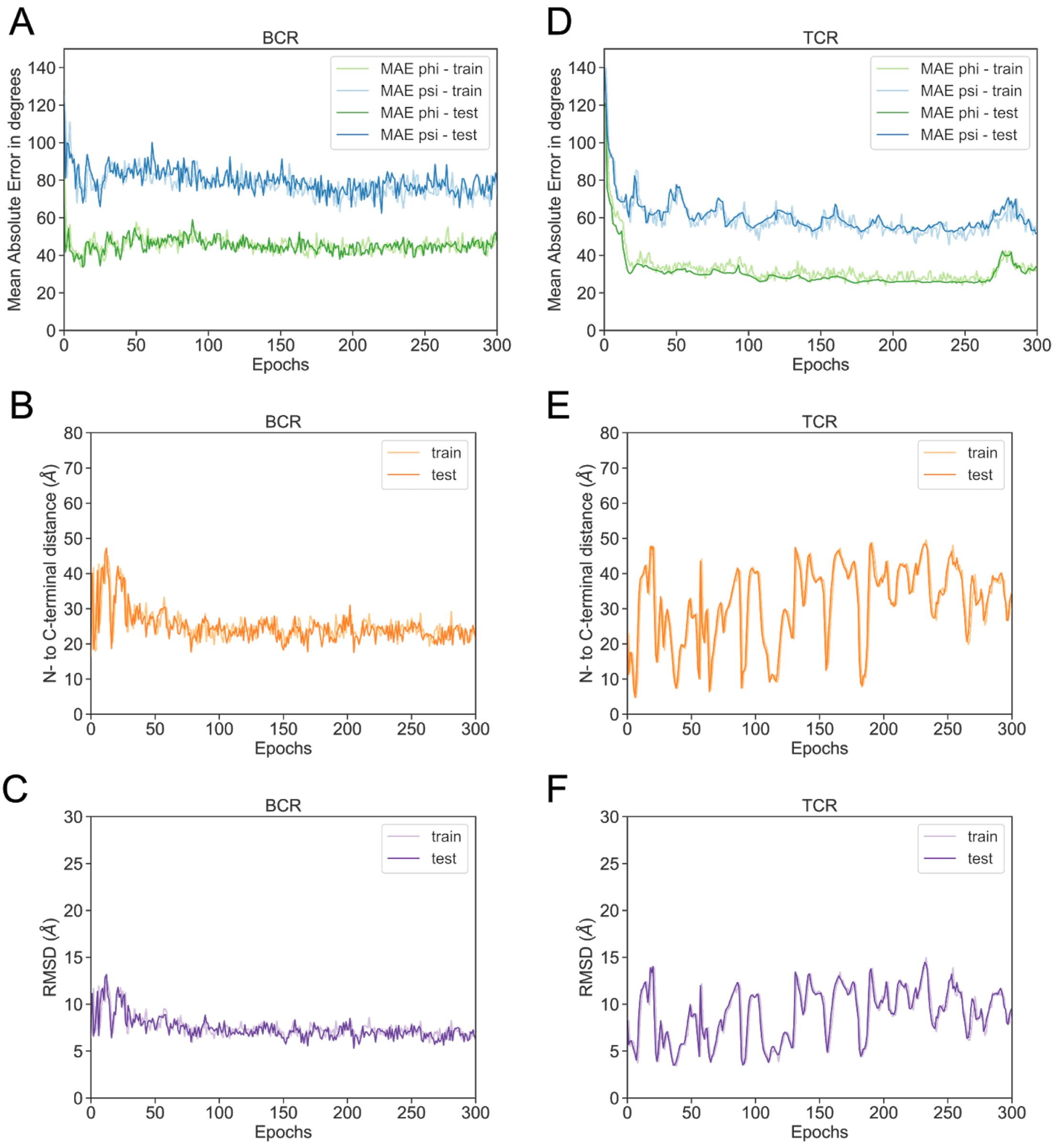


Figure 4: Training and validating CDR3 loop accuracy over 300 epochs for the TCR and BCR dataset. Panel **A)** and **D)** show the MAE in degrees for φ (phi) and ψ (psi) . Panel **B)** and **E)** show the distance between the C_{α} atom of the N- and C-terminal residue in the CDR3 loop. Panel **C)** and **F)** show the RMSD between the predicted CDR3 loop and the true CDR3 loop structure.

For training, the dataset was randomly divided into a training and a validation set, after which the WGAN-GP model was fitted for 300 epochs and evaluated for each epoch using both the training and the validation set (see Figure 4). From Figure 4 panel A and D, it is observed that the MAE for φ is lower than the MAE for ψ. This is expected, as the φ angles have a broader distribution compared to the ψ angle. From Figure 4 panel B and E we see the N- to C-terminal distance is around 25Å and from panel C and F we see that the RMSD is around 7Å. The N- to C-terminal distance in the crystal structures for CDR3 loops is around 9Å so it is clear that the network has not learned to generate loops in with the correct distance between the N- to C-terminal residue, and the RMSD values also indicate that the generated loops are not close the original crystal structure.

When taking a closer look at the two different networks shown in Figure 4, we see that the network trained on BCR loops seems more stable than the network trained on TCR loops. This is most likely because the BCR network is trained using a large data set, compared to the TCR network. Furthermore, we see that RMSD and the loop closure distance is still decreasing around 300 epochs, which could indicate that training the network using a larger number of epochs could improve the results.

Benchmarking

The ability of the Generator network to generate diverse and accurate CDR3 loops was tested using the independent benchmark datasets. This benchmark dataset includes unique CDR3 loops from 50 BCR structures and 11 TCR structures. For each CDR3 loop, we used the Generator network from the trained WGAN-GP to build 50 structures. An example of such 50 predicted CDR3 loops is illustrated in Figure 5 together with the true structure (PDB id 1BQL).

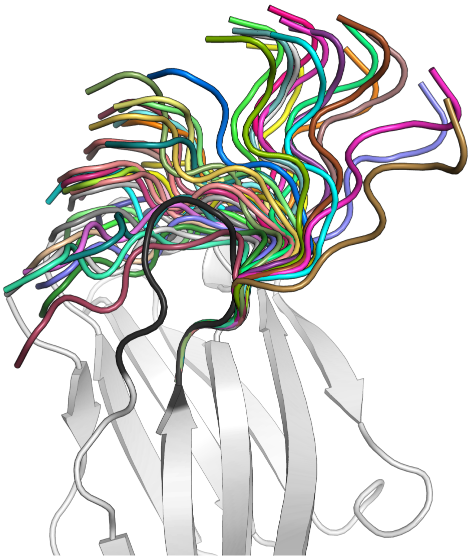


Figure 5: Example of the 50 CDR3 loops made using the Generator from the trained WGAN-GP. In the figure, all predicted CDR3 loops are shown with colors, while the CDR3 loop and the framework of the true structure are shown in black and grey, respectively. All generated structures were aligned using only the first three residues. Structural representations were made in PyMOL.

Figure 5 illustrates one of the major limitations with the current network, which is the loop closure problem. The network can predict φ and ψ angles which are quite diverse, and a majority of the generated structures seem to have understood that the loop should bend back toward the framework, but none of the predicted loops have the correct distance between the N- and C-terminal residues.

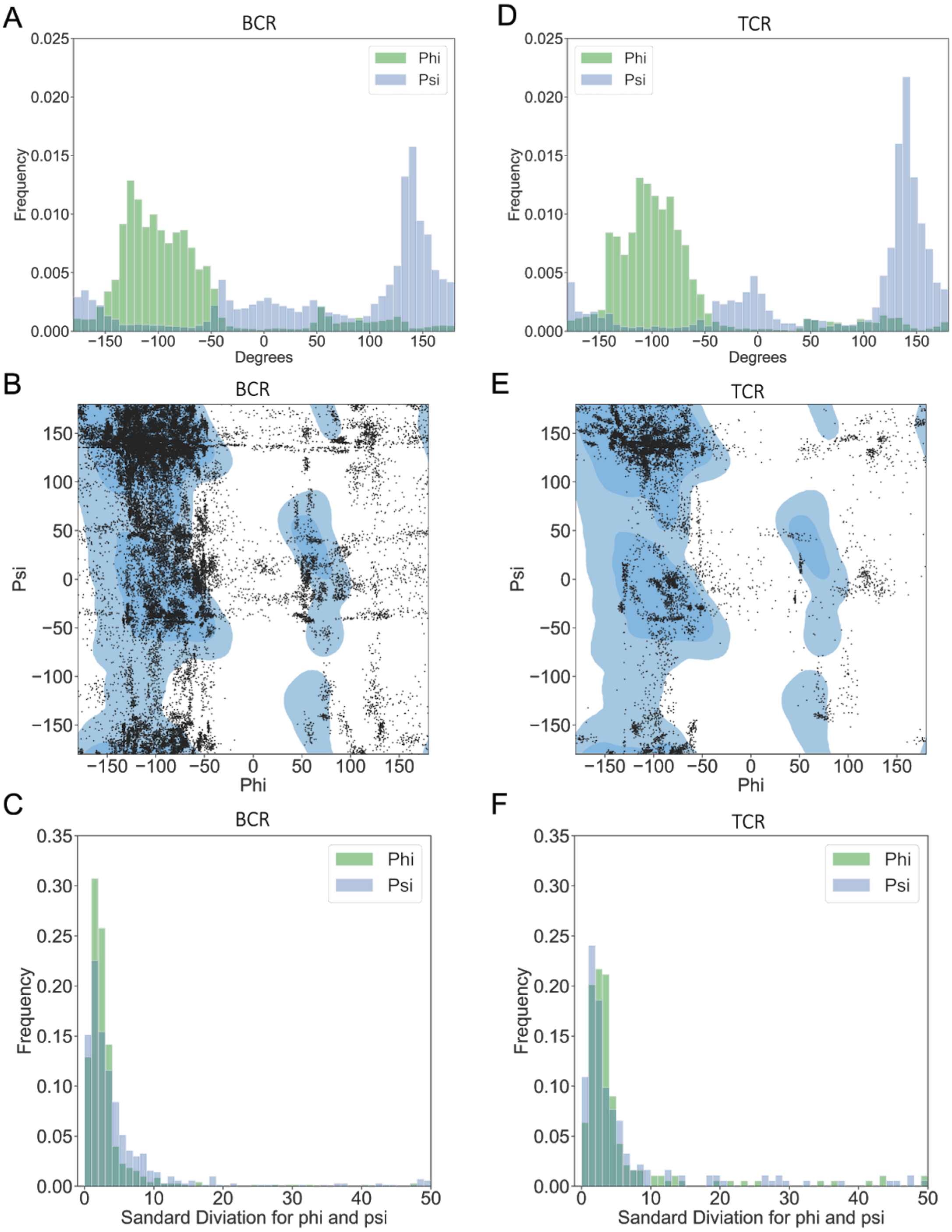


Figure 6: Network performance on the independent benchmark dataset evaluated by generating 50 different predictions for each of the CDR3 loops. **A)** Distribution of angles for the predicted BCR loops. **B)** The Ramachandran plot for the predicted BCR loops. **C)** Distribution of the standard deviation for φ (phi) and ψ (psi) angles for each residue in the BCR loop among 50 predictions. **D)** Distribution angles for the predicted TCR loops. **E)** The Ramachandran plot for the predicted TCR loops. **F)** Distribution of the standard deviation for the φ and ψ angles for each residue in the TCR loop among 50 predictions. In the ramachandran plot dark and light blue represents favoured and allowed regions for the φ and ψ angles, based on the findings by Lovell *et al.* [40].

From Figure 6, it is observed that the predicted angles tend to follow the same background distribution as shown in Figure 3, indicating that the network learned to generate CDR3 loops which follow the same rules as the biological data. From the Ramachandran plot in Figure 6, it is observed that paired φ and ψ predictions for each residue are less correct as many of them are within the unfavoured part of the Ramachandran plot. A common problem when training a GAN is that the Generator has a tendency to always produce the same or very similar output for all inputs - this problem is typically referred to as mode collapse. To test the Generator's ability to generate diverse outputs, we plotted the distribution of standard deviations for φ and ψ angles for each residue in the CDR3 loop among the 50 predictions for each structure in the benchmark dataset. Figure 6 C and F show the distribution of these standard deviations, and we note that each angle tends to deviate with approximately 5° between the 50 different outputs generated for each loop sequence in the benchmark dataset. For longer loops, the predicted structures may therefore vary quite significantly, which is also observed in Figure 5. Furthermore, this indicates that the Generator is capable of producing diverse outputs, meaning that the network does not suffer from mode collapse.

Discussion and perspectives

In this paper, we have described a first attempt to apply GANs to solve one of the main bottlenecks of *ab initio* CDR3 loop modeling for BCR and TCR structures; namely to generate diverse and accurate decoys in a computationally effective manner. The overall conclusion from our work, is that the GAN network is capable of predicting backbone dihedral angles for the CDR3 loop, which follow the background distribution of real structures, but that building the loop using these predicted angles produces structures which do not resemble true CDR3 loop structures. In the following sections, we discuss different issues that explain the limited performance of our current network.

Loop closure problem

One of the problems of the current WGAN-GP network is that the N- and C-terminal of the predicted CDR3 loops do not close. This problem can be solved using loop closure algorithms, but these algorithms are time-consuming and it would therefore be more ideal if the network could learn this feature. One possible solution is to include the loop closure distance in the Generator output. By including this distance in the output, it will be passed on to the Discriminator and it would thereby be included in the network optimization. Another solution is to implement different network architectures, both for the Generator and the Discriminator. Here, it could be interesting to try convolutional neural networks (CNNs) [41] and bidirectional long short-term memory (LSTM) networks [42]. The CNNs uses convolutional filters to detect short local motifs within the sequence, while the bidirectional LSTMs scan the sequence in both directions, extracting spatial dependencies between amino acids. Using these types of networks therefore has the potential to detect sequential motifs important for learning the underlying mechanism of loop closure.

Another approach is to include more training examples, by generating alternative loop conformations using general loop modeling methods such as Loopy [43], PLOP [44], LoopBuilder [45], Rosetta [38], LEAP [39], or Sphinx [18], to increase the number of training examples. These alternative loops could either be included directly into the training dataset or used to pre-train the network before training on the true CDR loop structures. Increasing the number of training examples might by itself improve the accuracy of the network predictions, as the network will be trained on loops covering a larger conformational space.

PeptideBuilder limitations

When reconstructing the loop structure using only the backbone dihedral angles, the resulting loop structure might not be identical to the native loop structure, as tight turns and unstructured loops have small deviations in backbone bond angles which can have a major impact on where in the three-dimensional space downstream secondary structure elements are located [36]. In this project, we used PeptideBuilder to reconstruct the loops and this tool is also affected by this limitation. Especially for longer loops, PeptideBuilder fails to obtain accurate reconstructions without adjusting all backbone bond angles, including planar angles.

Other structural problems

Aside from the loop closure problem, the predicted structures can also have other structural problems such as main-chain and side-chain clashes and we have also observed cases where the resulting structure is twisted. As these features are not biological, we also need some good metrics for evaluating and detecting such cases [46].

Selection problem

Before looking into the selection problem, we need to retrain the Generator network to solve some of the aforementioned problems, but when we have a Generator that can produce accurate and diverse outputs, we need a method for selecting the best loop among a pool of generated structures. Our idea here is to use the Discriminator to solve the problem of selecting the best loops. The Discriminator is trained to distinguish between loops made by the Generator and true loops from the training data, and the Discriminator score might therefore be useful for ranking and selecting the best loops.

When we have a finalized method for both generating and selecting the best CDR3 loops the plan is to compare the accuracy of the generated loops with other state-of-the-art template-based methods, such as LYRA [9], *ab initio* methods, such as KIC [21], and combination methods using both template-based and *ab initio* methods, such as Sphinx [18].

Conclusion

In this work, we have demonstrated that we can train GANs to generate diverse CDR3 loop structures, where the predicted dihedral angles follow the same distribution as real CDR3 structures. But the network still needs to learn how to generate more accurate structures with loop closure.

References

[1] L. A. Clark *et al.*, “Affinity enhancement of an in vivo matured therapeutic antibody using structure-based computational design,” *Protein Science*, vol. 15, no. 5. pp. 949–960, 2006.

[2] R. Diskin *et al.*, “Increasing the potency and breadth of an HIV antibody by using structure-based rational design,” *Science*, vol. 334, no. 6060, pp. 1289–1293, Dec. 2011.

[3] S. M. Lewis *et al.*, “Generation of bispecific IgG antibodies by structure-based design of an orthogonal Fab interface,” *Nat. Biotechnol.*, vol. 32, no. 2, pp. 191–198, Feb. 2014.

[4] N. Liddy *et al.*, “Monoclonal TCR-redirected tumor cell killing,” *Nat. Med.*, vol. 18, no. 6, pp. 980–987, Jun. 2012.

[5] P. Alzari, “Three-Dimensional Structure Of Antibodies,” *Annual Review of Immunology*, vol. 6, no. 1. pp. 555–580, 1988.

[6] R. M. MacCallum, A. C. Martin, and J. M. Thornton, “Antibody-antigen interactions: contact analysis and binding site topography,” *J. Mol. Biol.*, vol. 262, no. 5, pp. 732–745, Oct. 1996.

[7] V. Morea, A. Tramontano, M. Rustici, C. Chothia, and A. M. Lesk, “Conformations of the third hypervariable region in the VH domain of immunoglobulins,” *J. Mol. Biol.*, vol. 275, no. 2, pp. 269–294, Jan. 1998.

[8] B. North, A. Lehmann, and R. L. Dunbrack Jr, “A new clustering of antibody CDR loop conformations,” *J. Mol. Biol.*, vol. 406, no. 2, pp. 228–256, Feb. 2011.

[9] M. S. Klausen, M. V. Anderson, M. C. Jespersen, M. Nielsen, and P. Marcatili, “LYRA, a webserver for lymphocyte receptor structural modeling,” *Nucleic Acids Research*, vol. 43, no. W1. pp. W349–W355, 2015.

[10] A. Teplyakov *et al.*, “Antibody modeling assessment II. Structures and models,” *Proteins*, vol. 82, no. 8, pp. 1563–1582, Aug. 2014.

[11] H. M. Berman *et al.*, “The Protein Data Bank,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, Jan. 2000.

[12] R. Gowthaman and B. G. Pierce, “TCRmodel: high resolution modeling of T cell receptors from sequence,” *Nucleic Acids Res.*, vol. 46, no. W1, pp. W396–W401, Jul. 2018.

[13] A. Sircar, E. T. Kim, and J. J. Gray, “RosettaAntibody: antibody variable region homology modeling server,” *Nucleic Acids Res.*, vol. 37, no. Web Server issue, pp. W474–9, Jul. 2009.

[14] J. Leem, J. Dunbar, G. Georges, J. Shi, and C. M. Deane, “ABodyBuilder: Automated antibody structure prediction with data-driven accuracy estimation,” *MAbs*, vol. 8, no. 7, pp. 1259–1268, Oct. 2016.

[15] P. Marcatili, P. P. Olimpieri, A. Chailyan, and A. Tramontano, “Antibody modeling using the prediction of immunoglobulin structure (PIGS) web server [corrected],” *Nat. Protoc.*, vol. 9, no. 12, pp. 2771–2783, Dec. 2014.

[16] R. Lepore, P. P. Olimpieri, M. A. Messih, and A. Tramontano, “PIGSPro: prediction of immunoGlobulin structures v2,” *Nucleic Acids Res.*, vol. 45, no. W1, pp. W17–W23, Jul. 2017.

[17] M. A. Messih, R. Lepore, P. Marcatili, and A. Tramontano, “Improving the accuracy of the structure prediction of the third hypervariable loop of the heavy chains of antibodies,” *Bioinformatics*, vol. 30, no. 19. pp. 2733–2740, 2014.

[18] C. Marks *et al.*, “Sphinx: merging knowledge-based and ab initio approaches to improve

protein loop prediction," *Bioinformatics*, vol. 33, no. 9, pp. 1346–1353, May 2017.

[19] A. A. Canutescu and R. L. Dunbrack Jr, "Cyclic coordinate descent: A robotics algorithm for protein loop closure," *Protein Sci.*, vol. 12, no. 5, pp. 963–972, May 2003.

[20] P. S. Shenkin, D. L. Yarmush, R. M. Fine, H. J. Wang, and C. Levinthal, "Predicting antibody hypervariable loop conformation. I. Ensembles of random conformations for ringlike structures," *Biopolymers*, vol. 26, no. 12, pp. 2053–2085, Dec. 1987.

[21] D. J. Mandell, E. A. Coutsias, and T. Kortemme, "Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling," *Nat. Methods*, vol. 6, no. 8, pp. 551–552, Aug. 2009.

[22] R. F. Alford *et al.*, "The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design," *J. Chem. Theory Comput.*, vol. 13, no. 6, pp. 3031–3048, Jun. 2017.

[23] K. Uziela, D. Menéndez Hurtado, N. Shu, B. Wallner, and A. Elofsson, "ProQ3D: improved model quality assessments using deep learning," *Bioinformatics*, vol. 33, no. 10, pp. 1578–1580, May 2017.

[24] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., pp. 2672–2680, 2014.

[25] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv [cs.LG]*, 19-Nov-2015.

[26] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv [cs.LG]*, 06-Nov-2014.

[27] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least Squares Generative Adversarial Networks," *arXiv [cs.CV]*, 13-Nov-2016.

[28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 214–223, 2017.

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," *arXiv [cs.LG]*, 31-Mar-2017.

[30] A. Sivasubramanian, A. Sircar, S. Chaudhury, and J. J. Gray, "Toward high-resolution homology modeling of antibody Fv regions and application to antibody-antigen docking," *Proteins*, vol. 74, no. 2, pp. 497–514, Feb. 2009.

[31] P. J. A. Cock *et al.*, "Biopython: freely available Python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, Jun. 2009.

[32] J. Lyons *et al.*, "Predicting backbone Cα angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network," *J. Comput. Chem.*, vol. 35, no. 28, pp. 2040–2046, Oct. 2014.

[33] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[34] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 22-Dec-2014.

[35] R. Improta, L. Vitagliano, and L. Esposito, "Peptide bond distortions from planarity: new insights from quantum mechanical calculations and peptide/protein crystal structures," *PLoS One*, vol. 6, no. 9, p. e24533, Sep. 2011.

[36] M. Z. Tien, D. K. Sydykova, A. G. Meyer, and C. O. Wilke, "PeptideBuilder: A simple Python library to generate model peptides," *PeerJ*, vol. 1, p. e80, May 2013.

[37] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan, "Stereochemistry of polypeptide chain configurations," *J. Mol. Biol.*, vol. 7, pp. 95–99, Jul. 1963.

[38] A. Stein and T. Kortemme, "Improvements to robotics-inspired conformational

sampling in rosetta," *PLoS One*, vol. 8, no. 5, p. e63090, May 2013.

[39] S. Liang, C. Zhang, and Y. Zhou, "LEAP: highly accurate prediction of protein loop conformations by integrating coarse-grained sampling and optimized energy scores with all-atom refinement of backbone and side chains," *J. Comput. Chem.*, vol. 35, no. 4, pp. 335–341, Feb. 2014.

[40] S. C. Lovell *et al.*, "Structure validation by Calpha geometry: phi,psi and Cbeta deviation," *Proteins*, vol. 50, no. 3, pp. 437–450, Feb. 2003.

[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11. pp. 2278–2324, 1998.

[42] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8. pp. 1735–1780, 1997.

[43] Z. Xiang, C. S. Soto, and B. Honig, "Evaluating conformational free energies: the colony energy and its application to the problem of loop prediction," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 11, pp. 7432–7437, May 2002.

[44] M. P. Jacobson *et al.*, "A hierarchical approach to all-atom protein loop prediction," *Proteins*, vol. 55, no. 2, pp. 351–367, May 2004.

[45] C. S. Soto, M. Fasnacht, J. Zhu, L. Forrest, and B. Honig, "Loop modeling: Sampling, filtering, and scoring," *Proteins*, vol. 70, no. 3, pp. 834–843, Feb. 2008.

[46] P. Chys and P. Chacón, "Random Coordinate Descent with Spinor-matrices and Geometric Filters for Efficient Loop Closure," *J. Chem. Theory Comput.*, vol. 9, no. 3, pp. 1821–1829, Mar. 2013.

Chapter 6: Epilogue

The different research projects described in this thesis all concern the development of methods for improving our understanding of the molecular interactions between MHC molecules, peptides and T-cell receptors. The expectation is that these methods can be used in the future to improve predictions of T-cell epitopes and thereby assist the development of advanced T-cell based immunotherapies and rational vaccines [3].

Paper I

In this first project, we updated two in-house methods for predicting peptide binding affinity to MHC class II molecules. The two updated methods, named NetMHCII-2.3 and NetMHCIIpan-3.2, were trained using an expanded dataset from the Immune Epitope Database, enabling prediction of peptide binding to an extended number of MHC II molecules. The paper showed how training with this expanded dataset improved the performance for the peptide binding affinity predictions and for detecting T-cell epitopes for both NetMHCII-2.3 and NetMHCIIpan-3.2, when compared to the previous versions of the methods.

The peptide binding affinity data used in this project is generated using *in vitro* binding assays [65], but other types of data exist including peptide-MHC stability data and mass spectrometry (MS) data from MHC eluted peptides. As peptide-MHC stability is a measure of the half-life of the peptide-MHC complex, an immunological perspective is that such data could carry more relevant information compared to binding affinity. However, including this type of data when predicting the peptide-MHC binding have up to now manifested only marginal improvements [66].

Recent advancements in the field of MS have enabled the development of high-throughput assays, in which a single experiment can identify thousands of MHC eluted peptides (see review by Caron et al. [67]). Since this type of data is generated *in vivo*, it contains essential biological information about the peptide presentation pathway and the length distribution of naturally presented peptides.

When predicting the peptide-MHC binding it has been shown that including *in vivo* generated MS data improves performance for MHC class I molecules [18]. Similar results have been found for MHC class II molecules on a limited dataset [68] and it could therefore be interesting to train on a larger dataset and make an updated version of the NetMHCIIpan webserver.

The current version of NetMHCIIpan is trained using conventional feedforward artificial neural networks, but in recent years many new types of network architectures have been developed, including convolutional neural networks (CNNs) [45] and Recurrent neural networks (RNNs) [43, 44] and using these new network architectures might improve the peptide-MHC binding predictions. Especially, the use of CNNs could be interesting as this type of network allows for inputs of variable lengths. Since peptides and MHC sequences differ in length, this is a promising tool for predicting peptide-MHC binding affinities. CNNs have already been implemented for predicting peptide-MHC binding to MHC class I molecules [69, 70] and it could be interesting to train similar networks for predicting peptide-MHC binding to MHC class II molecules.

Paper II

In this project, we developed an automated tool for building structural models of the TCR-pMHC complex using only the amino acid sequence as input. In the paper we showed how the models produced by our tool have a higher accuracy than models produced using the TCRFlexDock method.

In this work, we focused on modelling the TCR-pMHC complex of αβ-TCRs and MHC class I molecules, as these constitute the majority of the available structural complexes. As more structural data is deposited in the Protein Data Bank, we hope to expand the tool to include TCR-pMHC complexes of MHC class II molecules.

One of the challenging parts in this project was to find a good way to select the best templates, especially when modeling the TCR-pMHC complex. In this project, we used the target-template sequence similarity for selecting the best templates, and we tried different ways of adjusting the contribution of each chain in the TCR-pMHC complex by introducing sequence weights. We showed that using weighted sequence similarity scores achieved the best modeling accuracy.

In the project we also selected multiple templates in the modeling step, as it has been suggested that using multiple templates can increase the model accuracy, especially when modelling protein complexes with multiple chains [71, 72]. But finding the most optimal combination of templates is a non-trivial task, since including too many templates usually leads to accumulation of noise. We therefore generated a method for selecting non-redundant templates, which decreases the number of selected templates, while still increasing the chance of selecting structures with a correct conformation. In the paper, we show that using this way of selecting multiple templates improves the accuracy of the generated models.

Another way to solve the problem of selecting the best templates could be to use machine learning methods trained for this task. The main idea of such an approach could be to train a model to predict the structural distances (eg. using TM-scores) between a target and all its potential templates, after which the template predicted to have the most similar structure to the target would be used to build the structural model. In this case, the input to the model would be the target sequence and the model should learn to predict the structural similarity to all the templates found in the template database. The template predicted to have the best structural similarity would be used for building the final model.

To predict T-cell immunogenicity, we need to understand how the TCR recognizes a specific pMHC. We expect that the structural models of the TCR-pMHC complex will serve as an important component for resolving this, and for characterizing properties of TCR-pMHC binding. This could be done by generating structural models of TCRs known to recognize specific pMHCs (binders) and compare these to structural models of TCRs and pMHCs which do not interact (non-binders). A machine learning model can be trained to predict the TCR-pMHC binding strengths either by using an energy function, as the one developed by Rosetta [73], or by using alternative refined force fields as suggested by Lanzarotti *et al.* [74]. If we can reliably predict the TCR-pMHC binding, we can utilize such a tool to provide valuable insights into the mechanisms underlying the interaction between TCR and pMHC, and ultimately predict T-cell specific epitopes.

Paper III

In this project, we investigated the TCR recognition profile using an experimental technique which measures the relative binding affinity between clonal TCRs and pMHC variants. By utilizing this experimental technique, the TCR binding can be translated into a TCR motif, called the TCR fingerprint, which can be used to predict cross-recognized peptides from the human proteome.

Understanding how TCRs cross-recognize structurally related pMHCs is extremely important within the field of adoptive T-cell therapy, as it can be used to identify self-reactive TCRs. It has been shown that genetically modified TCRs with high affinity for the pMHC can mediate self-reactivity [75–78], which in some cases can cause serious or even fatal side effects [79]. Due to these serious side effects of genetically modified TCRs, it has been proposed that these TCRs should be engineered to optimize the TCR-peptide binding while decreasing the TCR-MHC binding to avoid self-reactive TCRs [80].
Having an experimental technique for describing the molecular interaction points of the TCR can be used to advance the process of developing genetically modified TCRs and to select TCRs intended for adoptive T-cell therapy.

In the past, available data linking TCRs to their target pMHCs has been very limited, but recently more data of this type is being generated using high-throughput sequencing techniques [81–83]. It could be interesting to utilize such data to train both structure-based and sequence-based models to predict the interaction between TCRs and their target pMHCs, hopefully enabling the prediction of T-cell specific epitopes, and identification of immunogenic and non-immunogenic pMHCs.

PAPER IV

In the final project, we investigated the possibility of using generative adversarial networks (GANs), to improve the accuracy of structural models for CDR3 loops found in TCRs and BCRs. We showed that it is possible to train GANs to learn the background distribution of dihedral angles within the CDR3 loop, but the current network architecture still needs further development to learn essential structural features, such as the loop closure.

In recent years, many different GAN methods have been proposed for improving the network stability and performance [46–48, 84, 85]. We decided to use the WGAN-GP method [48] as this method has been shown to dramatically improve the stability of learning, while reducing the risk of mode collapse. Both the Discriminator and Generator network in our current network, are traditional feed forward neural networks, but many other network types exist, and it could be interesting to see if the implementation of other network types could improve the accuracy of the generated CDR3 structures. Within the field of bioinformatics, it has been shown that using convolutional neural networks (CNNs) [45] followed by bidirectional long short-term memory (LSTM) networks [44] is a very powerful architecture for capturing both sequential and structural information within the protein sequence [86–88]. The idea behind this architecture is that the CNNs can detect local motifs in the input sequence, while the bidirectional LSTM can capture long range sequence dependencies [89]. Because of this, these types of networks have the potential to capture the underlying mechanism of loop closure, which makes implementing them a very interesting subject for future research.

There are currently only 330 TCR structures and 2,529 BCR structures in the Protein Data Bank, for which some have identical CDR3 loops. Given the diversity of CDR3 loops, these experimentally determined structures only cover a small fraction of all possible CDR3 loop conformations. The limited amount of data used for training our current network could therefore be expanded by generating alternative loop conformations for each of the loops in our dataset. By increasing the amount of data we would expect an increase in the network performance as the network will be trained on CDR3 loops covering a larger conformational space.

The CDR3 structures generated by the current networks are not very accurate, but with the suggestions mentioned above, we expect to improve the accuracy of the generated CDR3 structures in the future. If the final network is capable of generating accurate structures, it can be used to improve tools like LYRA and TCRmodels for predicting TCR and RosettaAntibody [90], ABodyBuilder [91] and PIGS [92] for BCR structures, as well as the TCRpMHCmodels tool described in project II for predicting the structure of TCR-pMHC complexes.

The potential of GANs is huge and it could be interesting to apply it to solve other biological problems. Using the GAN architecture we could, for example, train a network to generate artificial TCR sequences for specific pMHC complexes. The input to the Generator for such a network would be the randomly generated noise plus a label in the form of the peptide and MHC sequence. Based on the random noise, the Generator should then learn to generate TCR sequences with the potential to recognize specific pMHCs. The Discriminator would be presented with a pMHC as well as either a fake TCR sequence produced by the Generator or a real TCR sequence. The Discriminator should then learn to distinguish between real and fake TCR-pMHC combinations, which will ideally pressure the Generator to produce artificial TCR sequences that recognize the pMHCs. If we can successfully train a network to generate TCR sequences with the potential to recognize specific pMHCs, we could utilize such a network to design TCRs used in adoptive T-cell therapy.

Bibliography

[1] K. P. Murphy and C. Janeway, *Janeway's Immunobiology*. Garland Pub, 2008.

[2] T. W. Mak, M. E. Saunders, and B. D. Jett, *Primer to the Immune Response*. Newnes, 2013.

[3] P. A. Ott *et al.*, "An immunogenic personal neoantigen vaccine for patients with melanoma," *Nature*, vol. 547, no. 7662, pp. 217–221, Jul. 2017.

[4] J. Neefjes, M. L. M. Jongsma, P. Paul, and O. Bakke, "Towards a systems understanding of MHC class I and MHC class II antigen presentation," *Nat. Rev. Immunol.*, vol. 11, no. 12, pp. 823–836, Nov. 2011.

[5] N. Zhang and M. J. Bevan, "CD8(+) T cells: foot soldiers of the immune system," *Immunity*, vol. 35, no. 2, pp. 161–168, Aug. 2011.

[6] M. Nielsen and O. Lund, "NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction," *BMC Bioinformatics*, vol. 10, no. 1, p. 296, Sep. 2009.

[7] M. Andreatta, E. Karosiene, M. Rasmussen, A. Stryhn, S. Buus, and M. Nielsen, "Accurate pan-specific prediction of peptide-MHC class II binding affinity with improved binding core identification," *Immunogenetics*, vol. 67, no. 11–12, pp. 641–650, Nov. 2015.

[8] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[9] T. Boehm, "Design principles of adaptive immune systems," *Nat. Rev. Immunol.*, vol. 11, no. 5, pp. 307–317, May 2011.

[10] M. Wieczorek *et al.*, "Major Histocompatibility Complex (MHC) Class I and MHC Class II Proteins: Conformational Plasticity in Antigen Presentation," *Frontiers in Immunology*, vol. 8. 2017.

[11] A. C. Carpenter and R. Bosselut, "Decision checkpoints in the thymus," *Nat. Immunol.*, vol. 11, no. 8, pp. 666–673, Aug. 2010.

[12] L. Chen and D. B. Flies, "Molecular mechanisms of T cell co-stimulation and co-inhibition," *Nat. Rev. Immunol.*, vol. 13, no. 4, pp. 227–242, Apr. 2013.

[13] M. G. Rudolph, R. L. Stanfield, and I. A. Wilson, "How TCRs bind MHCs, peptides, and coreceptors," *Annu. Rev. Immunol.*, vol. 24, pp. 419–466, 2006.

[14] V. Jurtz, S. Paul, M. Andreatta, P. Marcatili, B. Peters, and M. Nielsen, "NetMHCpan 4.0: Improved peptide-MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data." .

[15] L. Zhang, K. Udaka, H. Mamitsuka, and S. Zhu, "Toward more accurate pan-specific MHC-peptide binding prediction: a review of current methods and tools," *Brief. Bioinform.*, vol. 13, no. 3, pp. 350–364, May 2012.

[16] R. M. Chicz *et al.*, "Predominant naturally processed peptides bound to HLA-DR1 are derived from MHC-related molecules and are heterogeneous in size," *Nature*, vol. 358, no. 6389, pp. 764–768, Aug. 1992.

[17] C. A. Janeway Jr, P. Travers, M. Walport, and M. J. Shlomchik, "The major histocompatibility complex and its functions," in *Immunobiology: The Immune System in Health and Disease. 5th edition*, Garland Science, 2001.

[18] V. Jurtz, S. Paul, M. Andreatta, P. Marcatili, B. Peters, and M. Nielsen,

"NetMHCpan-4.0: Improved Peptide-MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data," *J. Immunol.*, vol. 199, no. 9, pp. 3360–3368, Nov. 2017.

[19] K. K. Jensen *et al.*, "Improved methods for predicting peptide binding affinity to MHC class II molecules," *Immunology*, vol. 154, no. 3. pp. 394–406, 2018.

[20] C. Zou, P. Zhao, Z. Xiao, X. Han, F. Fu, and L. Fu, "γδ T cells in cancer immunotherapy," *Oncotarget*, vol. 8, no. 5, pp. 8900–8909, Jan. 2017.

[21] D. J. Laydon, C. R. M. Bangham, and B. Asquith, "Estimating T-cell repertoire diversity: limitations of classical estimators and a new approach," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 370, no. 1675. p. 20140291, 2015.

[22] M. De Simone, G. Rossetti, and M. Pagani, "Single Cell T Cell Receptor Sequencing: Techniques and Future Challenges," *Front. Immunol.*, vol. 9, p. 1638, Jul. 2018.

[23] J. Nikolich-Zugich, M. K. Slifka, and I. Messaoudi, "The many important facets of T-cell repertoire diversity," *Nat. Rev. Immunol.*, vol. 4, no. 2, pp. 123–132, Feb. 2004.

[24] Q. Qi *et al.*, "Diversity and clonal selection in the human T-cell repertoire," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 36, pp. 13139–13144, Sep. 2014.

[25] K. C. Garcia, K. Christopher Garcia, L. Teyton, and I. A. Wilson, "STRUCTURAL BASIS OF T CELL RECOGNITION," *Annual Review of Immunology*, vol. 17, no. 1. pp. 369–397, 1999.

[26] N. L. L. Gruta, N. L. La Gruta, S. Gras, S. R. Daley, P. G. Thomas, and J. Rossjohn, "Understanding the drivers of MHC restriction of T cell receptors," *Nature Reviews Immunology*, vol. 18, no. 7. pp. 467–478, 2018.

[27] A. Dey, "Machine Learning Algorithms : A Review," 2016.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6. pp. 84–90, 2017.

[29] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," *arXiv [cs.LG]*, 28-Sep-2018.

[30] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016.

[31] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.

[32] P. Larrañaga *et al.*, "Machine learning in bioinformatics," *Brief. Bioinform.*, vol. 7, no. 1, pp. 86–112, Mar. 2006.

[33] V. I. Jurtz *et al.*, "An introduction to deep learning on biological sequence data: examples and solutions," *Bioinformatics*, vol. 33, no. 22, pp. 3685–3690, Nov. 2017.

[34] U. Hobohm, M. Scharf, R. Schneider, and C. Sander, "Selection of representative protein data sets," *Protein Sci.*, vol. 1, no. 3, pp. 409–417, Mar. 1992.

[35] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, "CD-HIT Suite: a web server for clustering and comparing biological sequences," *Bioinformatics*, vol. 26, no. 5, pp. 680–682, Mar. 2010.

[36] G. Wang and R. L. Dunbrack Jr, "PISCES: recent improvements to a PDB sequence culling server," *Nucleic Acids Res.*, vol. 33, no. Web Server issue, pp. W94–8, Jul. 2005.

[37] R. C. Edgar, "Search and clustering orders of magnitude faster than BLAST," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, Oct. 2010.

[38] M. Nielsen *et al.*, “Reliable prediction of T-cell epitopes using neural networks with novel sequence representations,” *Protein Sci.*, vol. 12, no. 5, pp. 1007–1017, May 2003.

[39] D. Chicco, “Ten quick tips for machine learning in computational biology,” *BioData Min.*, vol. 10, p. 35, Dec. 2017.

[40] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, 1995, pp. 1137–1143.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2016.

[42] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *Herbert Robbins Selected Papers*. pp. 102–109, 1985.

[43] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088. pp. 533–536, 1986.

[44] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8. pp. 1735–1780, 1997.

[45] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11. pp. 2278–2324, 1998.

[46] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” *arXiv [cs.LG]*, 06-Nov-2014.

[47] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein Generative Adversarial Networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 214–223.

[48] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved Training of Wasserstein GANs,” in *Advances in Neural Information Processing Systems 30*, pp. 5767–5777, 2017.

[49] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognition*, vol. 30, no. 7. pp. 1145–1159, 1997.

[50] C. Chothia and A. M. Lesk, “The relation between the divergence of sequence and structure in proteins,” *The EMBO Journal*, vol. 5, no. 4. pp. 823–826, 1986.

[51] S. Kaczanowski and P. Zielenkiewicz, “Why similar protein sequences encode similar three-dimensional structures?,” *Theoretical Chemistry Accounts*, vol. 125, no. 3–6. pp. 643–650, 2010.

[52] Y. Zhang, “Progress and challenges in protein structure prediction,” *Current Opinion in Structural Biology*, vol. 18, no. 3. pp. 342–348, 2008.

[53] J. Söding, A. Biegert, and A. N. Lupas, “The HHpred interactive server for protein homology detection and structure prediction,” *Nucleic Acids Res.*, vol. 33, no. Web Server issue, pp. W244–8, Jul. 2005.

[54] A. Waterhouse *et al.*, “SWISS-MODEL: homology modelling of protein structures and complexes,” *Nucleic Acids Res.*, vol. 46, no. W1, pp. W296–W303, Jul. 2018.

[55] W. Zheng, C. Zhang, E. W. Bell, and Y. Zhang, “I-TASSER gateway: A protein structure and function prediction server powered by XSEDE,” *Future Gener. Comput. Syst.*, vol. 99, pp. 73–85, Oct. 2019.

[56] B. Webb and A. Sali, “Comparative Protein Structure Modeling Using MODELLER,” *Curr. Protoc. Protein Sci.*, vol. 86, pp. 2.9.1–2.9.37, Nov. 2016.

[57] J. Moult, J. T. Pedersen, R. Judson, and K. Fidelis, “A large-scale experiment to assess protein structure prediction methods,” *Proteins*, vol. 23, no. 3, pp. ii–v, Nov. 1995.

[58] I. Kufareva and R. Abagyan, “Methods of Protein Structure Comparison,” *Methods in Molecular Biology*. pp. 231–257, 2011.

[59] Y. Zhang and J. Skolnick, “Scoring function for automated assessment of protein

structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 68, no. 4. pp. 1020–1020, 2007.

[60] M. Gao and J. Skolnick, "New benchmark metrics for protein-protein docking methods," *Proteins*, vol. 79, no. 5, pp. 1623–1634, May 2011.

[61] M. Nielsen, C. Lundegaard, and O. Lund, "Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method," *BMC Bioinformatics*, vol. 8, p. 238, Jul. 2007.

[62] M. Nielsen *et al.*, "Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan," *PLoS Comput. Biol.*, vol. 4, no. 7, p. e1000107, Jul. 2008.

[63] E. Karosiene, M. Rasmussen, T. Blicher, O. Lund, S. Buus, and M. Nielsen, "NetMHCIIpan-3.0, a common pan-specific MHC class II prediction method including all three human MHC class II isotypes, HLA-DR, HLA-DP and HLA-DQ," *Immunogenetics*, vol. 65, no. 10, pp. 711–724, Oct. 2013.

[64] A. Teplyakov *et al.*, "Antibody modeling assessment II. Structures and models," *Proteins*, vol. 82, no. 8, pp. 1563–1582, Aug. 2014.

[65] S. Justesen, M. Harndahl, K. Lamberth, L.-L. B. Nielsen, and S. Buus, "Functional recombinant MHC class II molecules and high-throughput peptide-binding assays," *Immunome Res.*, vol. 5, p. 2, May 2009.

[66] K. W. Jørgensen, M. Rasmussen, S. Buus, and M. Nielsen, "NetMHCstab - predicting stability of peptide-MHC-I complexes; impacts for cytotoxic T lymphocyte epitope discovery," *Immunology*, vol. 141, no. 1, pp. 18–26, Jan. 2014.

[67] E. Caron, D. J. Kowalewski, C. Chiek Koh, T. Sturm, H. Schuster, and R. Aebersold, "Analysis of Major Histocompatibility Complex (MHC) Immunopeptidomes Using Mass Spectrometry," *Mol. Cell. Proteomics*, vol. 14, no. 12, pp. 3105–3117, Dec. 2015.

[68] C. Garde *et al.*, "Improved peptide-MHC class II interaction prediction through integration of eluted ligand and peptide affinity data," *Immunogenetics*, vol. 71, no. 7, pp. 445–454, Jul. 2019.

[69] Y. Han and D. Kim, "Deep convolutional neural networks for pan-specific peptide-MHC class I binding prediction," *BMC Bioinformatics*, vol. 18, no. 1. 2017.

[70] Z. Liu, Y. Cui, Z. Xiong, A. Nasiri, A. Zhang, and J. Hu, "DeepSeqPan, a novel deep convolutional neural network model for pan-specific class I HLA-peptide binding affinity prediction." .

[71] J. Cheng, "A multi-template combination algorithm for protein comparative modeling," *BMC Struct. Biol.*, vol. 8, p. 18, Mar. 2008.

[72] A. Meier and J. Söding, "Automatic Prediction of Protein 3D Structures by Probabilistic Multi-template Homology Modeling," *PLoS Comput. Biol.*, vol. 11, no. 10, p. e1004343, Oct. 2015.

[73] R. F. Alford *et al.*, "The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design," *J. Chem. Theory Comput.*, vol. 13, no. 6, pp. 3031–3048, Jun. 2017.

[74] E. Lanzarotti, P. Marcatili, and M. Nielsen, "Identification of the cognate peptide-MHC target of T cell receptors using molecular modeling and force field scoring," *Mol. Immunol.*, vol. 94, pp. 91–97, Feb. 2018.

[75] P. D. Holler, L. K. Chlewicki, and D. M. Kranz, "TCRs with high affinity for foreign pMHC show self-reactivity," *Nat. Immunol.*, vol. 4, no. 1, pp. 55–62, Jan. 2003.

[76] J. D. Stone and D. M. Kranz, "Role of T cell receptor affinity in the efficacy and specificity of adoptive T cell therapies," *Front. Immunol.*, vol. 4, p. 244, Aug. 2013.

[77] M. Hebeisen, S. G. Oberle, D. Presotto, D. E. Speiser, D. Zehn, and N. Rufer, “Molecular insights for optimizing T cell receptor specificity against cancer,” *Front. Immunol.*, vol. 4, p. 154, Jun. 2013.

[78] J. E. Slansky and K. R. Jordan, “The Goldilocks Model for TCR—Too Much Attraction Might Not Be Best for Vaccine Design,” *PLoS Biology*, vol. 8, no. 9. p. e1000482, 2010.

[79] D. A. Antunes *et al.*, “Interpreting T-Cell Cross-reactivity through Structure: Implications for TCR-Based Cancer Immunotherapy,” *Front. Immunol.*, vol. 8, p. 1210, Oct. 2017.

[80] T. P. Riley and B. M. Baker, “The intersection of affinity and specificity in the development and optimization of T cell receptor based therapeutics,” *Semin. Cell Dev. Biol.*, vol. 84, pp. 30–41, Dec. 2018.

[81] S.-Q. Zhang *et al.*, “High-throughput determination of the antigen specificities of T cell receptors in single cells,” *Nat. Biotechnol.*, Nov. 2018.

[82] A. K. Bentzen *et al.*, “Large-scale detection of antigen-specific T cells using peptide-MHC-I multimers labeled with DNA barcodes,” *Nat. Biotechnol.*, vol. 34, no. 10, pp. 1037–1045, Oct. 2016.

[83] M. Klinger *et al.*, “Multiplex Identification of Antigen-Specific T Cell Receptors Using a Combination of Immune Assays and Immune Receptor Sequencing,” *PLoS One*, vol. 10, no. 10, p. e0141561, Oct. 2015.

[84] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *arXiv [cs.LG]*, 19-Nov-2015.

[85] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least Squares Generative Adversarial Networks,” *arXiv [cs.CV]*, 13-Nov-2016.

[86] M. S. Klausen *et al.*, “NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning,” *Proteins*, vol. 87, no. 6, pp. 520–527, Jun. 2019.

[87] S. K. Sønderby, C. K. Sønderby, H. Nielsen, and O. Winther, “Convolutional LSTM Networks for Subcellular Localization of Proteins,” *Algorithms for Computational Biology*. pp. 68–80, 2015.

[88] J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, and O. Winther, “DeepLoc: prediction of protein subcellular localization using deep learning,” *Bioinformatics*, vol. 33, no. 21, pp. 3387–3395, Nov. 2017.

[89] A. Graves, “Supervised Sequence Labelling with Recurrent Neural Networks,” *Studies in Computational Intelligence*. 2012.

[90] B. D. Weitzner *et al.*, “Modeling and docking of antibody structures with Rosetta,” *Nat. Protoc.*, vol. 12, no. 2, pp. 401–416, Feb. 2017.

[91] J. Leem, J. Dunbar, G. Georges, J. Shi, and C. M. Deane, “ABodyBuilder: Automated antibody structure prediction with data-driven accuracy estimation,” *MAbs*, vol. 8, no. 7, pp. 1259–1268, Oct. 2016.

[92] P. Marcatili, P. P. Olimpieri, A. Chailyan, and A. Tramontano, “Antibody modeling using the prediction of immunoglobulin structure (PIGS) web server,” *Nat. Protoc.*, vol. 9, no. 12, pp. 2771–2783, Dec. 2014.

Paper I: Appendix

Supplementary

Suppl Table 1: Description of the full 2016 data set. The #peptides indicates the total number of peptides present for each MHC molecule and the #binders indicate the number of peptides that have a log-transformed IC50 value above 0.5.

Molecules	#peptides	#binders
DRB1_0101	10412	6376
DRB1_0102	8	3
DRB1_0103	42	4
DRB1_0301	5352	1457
DRB1_0302	37	0
DRB1_0401	6317	3022
DRB1_0402	53	19
DRB1_0403	59	14
DRB1_0404	3657	1852
DRB1_0405	3962	1654
DRB1_0406	14	1
DRB1_0411	2	2
DRB1_0701	6325	3456
DRB1_0801	937	390
DRB1_0802	4465	2036
DRB1_0803	8	1
DRB1_0804	3	3
DRB1_0901	4318	2164
DRB1_1001	2066	1521
DRB1_1101	6045	2667
DRB1_1104	27	5
DRB1_1201	2384	759
DRB1_1301	1034	520
DRB1_1302	4477	2249
DRB1_1402	1	0
DRB1_1501	4850	2107
DRB1_1502	23	7
DRB1_1503	1	1
DRB1_1602	1699	989
DRB3_0101	4633	1415
DRB3_0202	3334	1055
DRB3_0301	884	510
DRB4_0101	3961	1540
DRB4_0103	846	525
DRB5_0101	5125	2430
DRB5_0102	2	2
H-2-IAb	1794	431
H-2-IAd	774	321
H-2-IAk	115	4
H-2-IAq	31	0
H-2-IAs	190	48
H-2-IAu	56	22
H-2-IEd	245	28
H-2-IEk	68	40
HLA-DPA10103-DPB10201	787	141
HLA-DPA10103-DPB10301	1563	575
HLA-DPA10103-DPB10401	2725	786
HLA-DPA10103-DPB10402	45	9
HLA-DPA10103-DPB10601	584	282
HLA-DPA10201-DPB10101	2447	859
HLA-DPA10201-DPB10501	2470	713
HLA-DPA10201-DPB11401	2302	849
HLA-DPA10301-DPB10402	2641	921
HLA-DQA10101-DQB10501	2946	815
HLA-DQA10102-DQB10501	833	458
HLA-DQA10102-DQB10502	800	158
HLA-DQA10102-DQB10602	2747	1256
HLA-DQA10102-DQB10604	61	0
HLA-DQA10103-DQB10302	6	0
HLA-DQA10103-DQB10603	462	90
HLA-DQA10104-DQB10503	883	105
HLA-DQA10201-DQB10201	23	0
HLA-DQA10201-DQB10202	944	119
HLA-DQA10201-DQB10301	827	374
HLA-DQA10201-DQB10303	761	265
HLA-DQA10201-DQB10402	768	241
HLA-DQA10301-DQB10201	4	0
HLA-DQA10301-DQB10301	207	66
HLA-DQA10301-DQB10302	3111	568
HLA-DQA10302-DQB10303	6	0
HLA-DQA10302-DQB10401	27	0
HLA-DQA10303-DQB10402	567	117
HLA-DQA10401-DQB10402	2890	928
HLA-DQA10501-DQB10201	2897	874
HLA-DQA10501-DQB10301	3585	1812
HLA-DQA10501-DQB10302	847	203
HLA-DQA10501-DQB10303	564	179
HLA-DQA10501-DQB10402	749	337
HLA-DQA10505-DQB10301	1	0
HLA-DQA10601-DQB10402	565	133
Total:	**134281**	**55883**

Suppl table 2: NetMHCII and NetMHCIIpan predictions of peptide binding cores. The offset correction is used to improve the identification of the right peptide binding core. To evaluate the effect of introducing this offset correction in our two methods, we benchmarked our results using 51 crystal structures of peptide-MHC class II complexes from the PDB database. Core (PDB) is the validated binding register as observed in the PDB crystal structures. Incorrect core predictions are highlighted in grey.

				NetMHCII-2.2 (with offset)	NetMHCII-2.2 (without offset)	NetMHCII-2.3 (with offset)	NetMHCIIpan-3.1 (with offset)	NetMHCIIpan-3.2 (without offset)	NetMHCIIpan-3.2 (with offset)
PDB	Allele	Antigen	Core (PDB)	Predicted core	Predicted core	Predicted core	Predicted core	Predicted core	Predicted core
1T5X	DRB1*01:01	AAYSDQATPLLLSPR	YSDQATPLL	YSDQATPLL	SDQATPLLL	SDQATPLLL	YSDQATPLL	YSDQATPLL	YSDQATPLL
2FSE	DRB1*01:01	AGFKGEQGPKGEPG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG
3L6F	DRB1*01:01	APPAYEKLSAEQSPP	YEKLSAEQS	YEKLSAEQS	YEKLSAEQS	YEKLSAEQS	YEKLSAEQS	YEKLSAEQS	YEKLSAEQS
1KLG	DRB1*01:01	GELIGTLNAAKVPAD	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV
4OV5	DRB1*01:01	GSDARFLRGYHLYA	ARFLRGYHL	ARFLRGYHL	ARFLRGYHL	ARFLRGYHL	ARFLRGYHL	ARFLRGYHL	ARFLRGYHL
3PGD	DRB1*01:01	KMRMATPLLMQALPM	MRMATPLLM	KMRMATPLL	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM
3PDO	DRB1*01:01	KPVSKMRMATPLLMQALPM	MRMATPLLM	KMRMATPLL	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM
4AEN	DRB1*01:01	MPLAQMLLPTAMRMKM	MLLPTAMRM	LAQMLLPTA	LAQMLLPTA	MLLPTAMRM	MLLPTAMRM	MLLPTAMRM	MLLPTAMRM
1SJH	DRB1*01:01	PEVIPMFSALSEG	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS
1SJE	DRB1*01:01	PEVIPMFSALSEGATP	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS
1FYT	DRB1*01:01	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
3QXA	DRB1*01:01	PVSKMRMATPLLMQA	MRMATPLLM	KMRMATPLL	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM
1AQD	DRB1*01:01	VGSDWRFLRGYHQYA	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ
4I5B	DRB1*01:01	VVKQNCLKLATK	VVKQNCLKL	VKQNCLKLA	VVKQNCLKL	VVKQNCLKL	VKQNCLKLA	VKQNCLKLA	VKQNCLKLA
1PYW	DRB1*01:01	XFVKQNAAALX	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL
2IPK	DRB1*01:01	XPKWVKQNTLKLAT	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL
1A6A	DRB1*03:01	PVSKMRMATPLLMQA	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM
4MD4	DRB1*04:01	ATEYRVRVNSAYQDK	YRVRVNSAY	YRVRVNSAY	EYRVRVNSA	YRVRVNSAY	YRVRVNSAY	YRVRVNSAY	YRVRVNSAY
2SEB	DRB1*04:01	AYMRADAAAGGA	MRADAAAGG	YMRADAAAG	YMRADAAAG	YMRADAAAG	YMRADAAAG	YMRADAAAG	YMRADAAAG
4MCZ	DRB1*04:01	GVYATRSSAVRLR	YATRSSAVR	YATRSSAVR	VYATRSSAV	VYATRSSAV	VYATRSSAV	VYATRSSAV	VYATRSSAV
1J8H	DRB1*04:01	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
4MCY	DRB1*04:01	SAVRLRSSVPGVR	VRLRSSVPG	VRLRSSVPG	VRLRSSVPG	VRLRSSVPG	VRLRSSVPG	VRLRSSVPG	VRLRSSVPG
4IS6	DRB1*04:01	WNRQLYPEWTEAQRLD	LYPEWTEAQ	LYPEWTEAQ	LYPEWTEAQ	LYPEWTEAQ	LYPEWTEAQ	LYPEWTEAQ	LYPEWTEAQ
4MDI	DRB1*04:02	SAVRLRSSVPGVR	VRLRSSVPG				VRLRSSVPG	VRLRSSVPG	VRLRSSVPG
4MD5	DRB1*04:04	SAVRLRSSVPGVR	VRLRSSVPG	VRLRSSVPG	AVRLRSSVP	AVRLRSSVP	VRLRSSVPG	VRLRSSVPG	VRLRSSVPG
1BX2	DRB1*15:01	ENPVVHFFKNIVTPR	VHFFKNIVT	VVHFFKNIV	VHFFKNIVT	VHFFKNIVT	VHFFKNIVT	VVHFFKNIV	VHFFKNIVT
1YMM	DRB1*15:01	ENPVVHFFKNIVTPRGGSGGGGG	VHFFKNIVT	VVHFFKNIV	VHFFKNIVT	VHFFKNIVT	VHFFKNIVT	VVHFFKNIV	VHFFKNIVT
2Q6W	DRB3*01:01	AWRSDEALPLGS	WRSDEALPL	WRSDEALPL	WRSDEALPL	WRSDEALPL	WRSDEALPL	WRSDEALPL	WRSDEALPL
4H25	DRB3*03:01	QHIRCNIPKRIGPSKVATLVPR	IRCNIPKRI				IRCNIPKRI	IRCNIPKRI	IRCNIPKRI
4H1L	DRB3*03:01	QHIRCNIPKRISA	IRCNIPKRI				IRCNIPKRI	IRCNIPKRI	IRCNIPKRI
3C5J	DRB3*03:01	QVIILNHPGQISA	IILNHPGQI				IILNHPGQI	IILNHPGQI	IILNHPGQI
4H26	DRB3*03:01	QWIRVNIPKRI	IRVNIPKRI				IRVNIPKRI	IRVNIPKRI	IRVNIPKRI
1H15	DRB5*01:01	GGVYHFVKKHVHES	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH
1FV1	DRB5*01:01	NPVVHFFKNIVTPRTPPPSQ	FKNIVTPRT	FFKNIVTPR	FFKNIVTPR	FFKNIVTPR	FKNIVTPRT	FFKNIVTPR	FKNIVTPRT
1HQR	DRB5*01:01	VHFFKNIVTPRTP	FKNIVTPRT	FFKNIVTPR	FFKNIVTPR	FFKNIVTPR	FKNIVTPRT	FFKNIVTPR	FKNIVTPRT
1ZGL	DRB5*01:01	VHFFKNIVTPRTPGG	FKNIVTPRT	FFKNIVTPR	FFKNIVTPR	FFKNIVTPR	FKNIVTPRT	FFKNIVTPR	FKNIVTPRT
4P23	H-2-IAb	FEAQKAKANKAVD	AQKAKANKA	AQKAKANKA	FEAQKAKAN	FEAQKAKAN	AQKAKANKA	FWIDLFETI	AQKAKANKA
1MUJ	H-2-IAb	PVSKMRMATPLLMQA	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	FWIDLFETI	MRMATPLLM
2IAD	H-2-IAd	HATQGVTAASSHE	TQGVTAASS	HATQGVTAA	TQGVTAASS	TQGVTAASS	TQGVTAASS	YDGKDYIAL	TQGVTAASS
1IAO	H-2-IAd	ISQAVHAAHAEI	SQAVHAAHA	QAVHAAHAE	SQAVHAAHA	SQAVHAAHA	SQAVHAAHA	FHYLPFLPS	SQAVHAAHA
4P4K	DPA1*01:03-DPB1*02:01	QAFWIDLFETIG	FWIDLFETI	FWIDLFETI	FWIDLFETI	FWIDLFETI	FWIDLFETI	KVTVAFNQF	FWIDLFETI
4P57	DPA1*01:03-DPB1*02:01	QAFWIDLFETIGGGSLV	FWIDLFETI	FWIDLFETI	FWIDLFETI	FWIDLFETI	FWIDLFETI	TKVSWAAVG	FWIDLFETI
4P5M	DPA1*01:03-DPB1*02:01	QAYDGKDYIALKG	YDGKDYIAL	YDGKDYIAL	YDGKDYIAL	YDGKDYIAL	YDGKDYIAL	EQPEQPFPQ	YDGKDYIAL
3LQZ	DPA1*01:03-DPB1*02:01	RKFHYLPFLPSTGGS	FHYLPFLPS	FHYLPFLPS	RKFHYLPFL	FHYLPFLPS	FHYLPFLPS	EGSFQPSQE	FHYLPFLPS
3WEX	DPA1*02:01-DPB1*05:01	KVTVAFNQFGGS	KVTVAFNQF	VAFNQFGGS	KVTVAFNQF	XKVTVAFNQ	VAFNQFGGS	EALYLVCGE	KVTVAFNQF
1UVQ	DQA1*01:02-DQB1*06:02	MNLPSTKVSWAAVGGGGSLV	LPSTKVSWA	VSWAAVGGG	VSWAAVGGG	VSWAAVGGG	TKVSWAAVG	PELPYPQPG	TKVSWAAVG
4D8P	DQA1*03:01-DQB1*02:01	PQPEQPEQPFPQP	EQPEQPFPQ				EQPEQPFPQ	LQPFPQPEL	EQPEQPFPQ
4GG6	DQA1*03:01-DQB1*03:02	QQYPSGEGSFQPSQENPQ	EGSFQPSQE	EGSFQPSQE	EGSFQPSQE	EGSFQPSQE	EGSFQPSQE	AQKAKANKA	EGSFQPSQE
1JK8	DQA1*03:03-DQB1*03:02	LVEALYLVCGERGG	EALYLVCGE				EALYLVCGE	MRMATPLLM	EALYLVCGE
4OZG	DQA1*05:05-DQB1*02:01	APQPELPYPQPGS	PQPELPYPQ				PQPELPYPQ	TQGVTAASS	PELPYPQPG
1S9V	DQA1*05:05-DQB1*02:01	LQPFPQPELPY	PFPQPELPY				LQPFPQPEL	SQAVHAAHA	LQPFPQPEL
				27/42	30/42	32/42	45/51	28/51	45/51

Suppl Table 3: Performance for NetMHCII-2.3, NetMHCIIpan-3.2 and the combined method. The combined method is made using a simple average of the prediction scores from NetMHCII-2.3 and NetMHCIIpan-3.2.

Molecule	#peptide	#binders	NetMHCII-2.3	NetMHCIIpan-3.2	Combined
DRB1_0101	10412	6376	0.829	0.832	0.838
DRB1_0103	42	4	0.250	0.678	0.599
DRB1_0301	5352	1457	0.816	0.816	0.826
DRB1_0401	6317	3022	0.798	0.809	0.813
DRB1_0402	53	19	0.633	0.701	0.649
DRB1_0403	59	14	0.644	0.841	0.787
DRB1_0404	3657	1852	0.787	0.812	0.808
DRB1_0405	3962	1654	0.839	0.827	0.846
DRB1_0701	6325	3456	0.877	0.875	0.885
DRB1_0801	937	390	0.834	0.844	0.854
DRB1_0802	4465	2036	0.834	0.834	0.844
DRB1_0901	4318	2164	0.832	0.833	0.843
DRB1_1001	2066	1521	0.912	0.923	0.924
DRB1_1101	6045	2667	0.867	0.864	0.873
DRB1_1201	2384	759	0.891	0.868	0.892
DRB1_1301	1034	520	0.828	0.857	0.856
DRB1_1302	4477	2249	0.889	0.885	0.895
DRB1_1501	4850	2107	0.833	0.834	0.842
DRB1_1602	1699	989	0.879	0.883	0.888
DRB3_0101	4633	1415	0.898	0.888	0.900
DRB3_0202	3334	1055	0.887	0.869	0.886
DRB3_0301	884	510	0.824	0.840	0.845
DRB4_0101	3961	1540	0.837	0.822	0.844
DRB4_0103	846	525	0.839	0.841	0.861
DRB5_0101	5125	2430	0.849	0.849	0.858
H-2-IAb	1794	431	0.884	0.894	0.895
H-2-IAd	774	321	0.819	0.819	0.829
H-2-IAk	115	4	0.628	0.635	0.685
H-2-IAs	190	48	0.761	0.825	0.814
H-2-IAu	56	22	0.830	0.765	0.820
H-2-IEd	245	28	0.730	0.754	0.762
H-2-IEk	68	40	0.836	0.853	0.864
HLA-DPA10103-DPB10201	787	141	0.910	0.917	0.921
HLA-DPA10103-DPB10301	1563	575	0.914	0.902	0.916
HLA-DPA10103-DPB10401	2725	786	0.935	0.935	0.939
HLA-DPA10103-DPB10402	45	9	0.497	0.710	0.636
HLA-DPA10103-DPB10601	584	282	0.996	0.995	0.995
HLA-DPA10201-DPB10101	2447	859	0.903	0.903	0.909
HLA-DPA10201-DPB10501	2470	713	0.914	0.911	0.919
HLA-DPA10201-DPB11401	2302	849	0.937	0.930	0.938
HLA-DPA10301-DPB10402	2641	921	0.906	0.904	0.910
HLA-DQA10101-DQB10501	2946	815	0.917	0.900	0.917
HLA-DQA10102-DQB10501	833	458	0.867	0.839	0.874
HLA-DQA10102-DQB10502	800	158	0.850	0.835	0.859
HLA-DQA10102-DQB10602	2747	1256	0.905	0.890	0.906
HLA-DQA10103-DQB10603	462	90	0.816	0.861	0.855
HLA-DQA10104-DQB10503	883	105	0.844	0.805	0.844
HLA-DQA10201-DQB10202	944	119	0.851	0.814	0.853
HLA-DQA10201-DQB10301	827	374	0.864	0.849	0.871
HLA-DQA10201-DQB10303	761	265	0.887	0.894	0.899
HLA-DQA10201-DQB10402	768	241	0.858	0.860	0.875
HLA-DQA10301-DQB10301	207	66	0.761	0.839	0.814
HLA-DQA10301-DQB10302	3111	568	0.849	0.810	0.842
HLA-DQA10303-DQB10402	567	117	0.836	0.820	0.855
HLA-DQA10401-DQB10402	2890	928	0.894	0.883	0.897
HLA-DQA10501-DQB10201	2897	874	0.889	0.876	0.888
HLA-DQA10501-DQB10301	3585	1812	0.922	0.915	0.924
HLA-DQA10501-DQB10302	847	203	0.831	0.822	0.840
HLA-DQA10501-DQB10303	564	179	0.884	0.876	0.892
HLA-DQA10501-DQB10402	749	337	0.857	0.868	0.876
HLA-DQA10601-DQB10402	565	133	0.845	0.848	0.872
Average			**0.833**	**0.847**	**0.855**

Suppl Table 4: The performance of the Leave-one-molecule-out (LOMO) benchmark analysis of NetMHCIIpan-3.2 including information about distance to nearest neighbor.

Molecule	**#peptides**	**#binders**	**AUC**	**Distance**
HLA-DQA10601-DQB10402	565	133	0.636	0.019
HLA-DQA10501-DQB10402	749	337	0.824	0.063
HLA-DQA10501-DQB10303	564	179	0.864	0.039
HLA-DQA10501-DQB10302	847	203	0.802	0.039
HLA-DQA10501-DQB10301	3585	1812	0.800	0.064
HLA-DQA10501-DQB10201	2897	874	0.779	0.073
HLA-DQA10401-DQB10402	2890	928	0.666	0.019
HLA-DQA10303-DQB10402	567	117	0.680	0.055
HLA-DQA10301-DQB10302	3111	568	0.648	0.086
HLA-DQA10301-DQB10301	207	66	0.814	0.055
HLA-DQA10201-DQB10402	768	241	0.840	0.047
HLA-DQA10201-DQB10303	761	265	0.873	0.064
HLA-DQA10201-DQB10301	827	374	0.835	0.055
HLA-DQA10201-DQB10202	944	119	0.789	0.073
HLA-DQA10104-DQB10503	883	105	0.765	0.042
HLA-DQA10103-DQB10603	462	90	0.837	0.066
HLA-DQA10102-DQB10602	2747	1256	0.786	0.066
HLA-DQA10102-DQB10502	800	158	0.694	0.032
HLA-DQA10102-DQB10501	833	458	0.618	0.016
HLA-DQA10101-DQB10501	2946	815	0.678	0.016
HLA-DPA10301-DPB10402	2641	921	0.889	0.090
HLA-DPA10201-DPB11401	2302	849	0.896	0.059
HLA-DPA10201-DPB10501	2470	713	0.880	0.070
HLA-DPA10201-DPB10101	2447	859	0.880	0.070
HLA-DPA10103-DPB10601	584	282	0.993	0.070
HLA-DPA10103-DPB10402	45	9	0.719	0.022
HLA-DPA10103-DPB10401	2725	786	0.921	0.039
HLA-DPA10103-DPB10301	1563	575	0.840	0.059
HLA-DPA10103-DPB10201	787	141	0.882	0.022
H-2-IEk	68	40	0.854	0.328
H-2-IEd	245	28	0.646	0.207
H-2-IAu	56	22	0.739	0.241
H-2-IAs	190	48	0.514	0.437
H-2-IAk	115	4	0.383	0.241
H-2-IAd	774	321	0.725	0.339
H-2-IAb	1794	431	0.780	0.339
DRB5_0101	5125	2430	0.765	0.202
DRB4_0103	846	525	0.794	0.000
DRB4_0101	3961	1540	0.726	0.000
DRB3_0301	884	510	0.734	0.123
DRB3_0202	3334	1055	0.756	0.123
DRB3_0101	4633	1415	0.801	0.142
DRB1_1602	1699	989	0.866	0.133
DRB1_1501	4850	2107	0.780	0.133
DRB1_1302	4477	2249	0.701	0.046
DRB1_1301	1034	520	0.731	0.046
DRB1_1201	2384	759	0.800	0.209
DRB1_1101	6045	2667	0.767	0.057
DRB1_1001	2066	1521	0.905	0.158
DRB1_0901	4318	2164	0.791	0.251
DRB1_0802	4465	2036	0.765	0.028
DRB1_0801	937	390	0.804	0.028
DRB1_0701	6325	3456	0.830	0.266
DRB1_0405	3962	1654	0.799	0.045
DRB1_0404	3657	1852	0.791	0.031
DRB1_0403	59	14	0.862	0.031
DRB1_0402	53	19	0.789	0.070
DRB1_0401	6317	3022	0.766	0.045
DRB1_0301	5352	1457	0.699	0.142
DRB1_0103	42	4	0.711	0.069
DRB1_0101	10412	6376	0.783	0.069
Average			**0.731**	

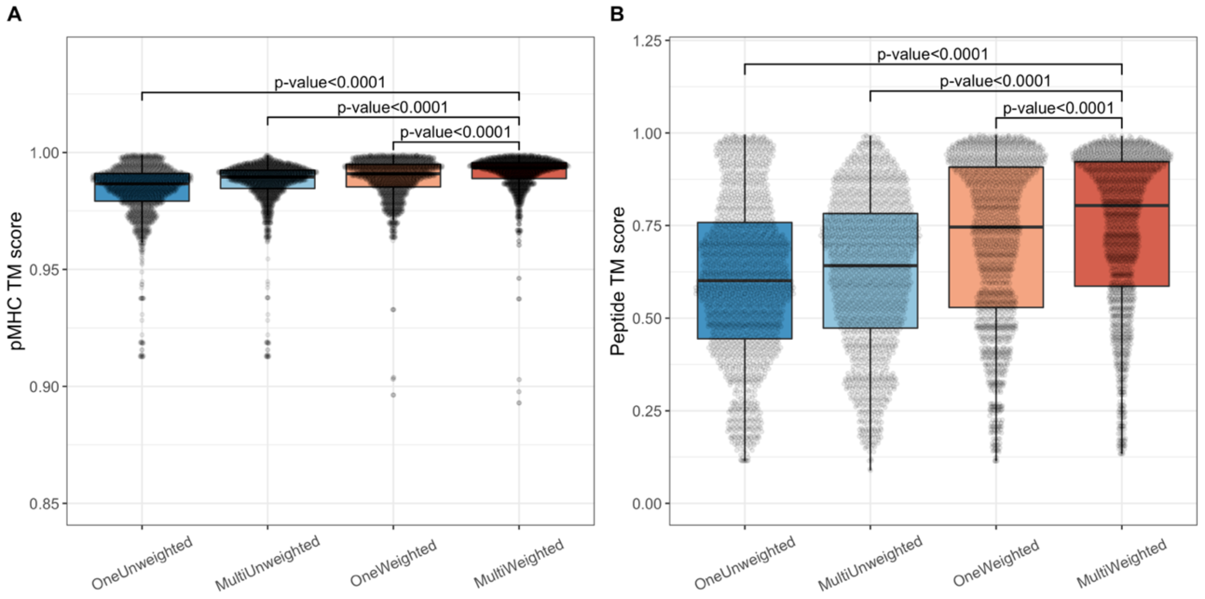
Suppl table 5: The predictive performance for NetMHCIIpan-3.1 and NetMHCIIpan-3.2 on the IEDB T-cell epitope data set. For each epitope in this data set, we calculated AUC and Frank values for the two NetMHCIIpan methods by predicting binding affinities for all overlapping peptides in the source protein sequence with the same length as the epitope, annotating the epitope as positive and the remaining peptides as negatives. For each MHC molecule, we calculated the average AUC performance. Also, shown in the table is the difference in the performance, as well as the difference in the nearest neighbor
distance and the difference in the number of data points in the data sets used for training NetMHCIIpan-3.1 and NetMHCIIpan-3.2.

Molecule	**Number of epitopes per MHC**	**Average Frank (3.1)**	**Average Frank (3.2)**	**Average AUC (3.1)**	**Average AUC (3.2)**	**Δ AUC perfor-mance**	**Nearest neighbor distance (3.1)**	**Nearest neighbor distance (3.2)**	**Δ distance to nearest neighbor**	**Number of data points in 2013 data set**	**Number of data points in 2016 data set**	**Δ number of data points**
DRB1_0101	240	0.19	0.181	0.809	0.818	0.009	0	0	0	7685	10412	2727
DRB1_0102	5	0.14	0.142	0.86	0.858	-0.001	0.066	0	-0.066	0	0	0
DRB1_0103	43	0.319	0.206	0.681	0.794	0.113	0.069	0	-0.069	0	42	42
DRB1_0301	101	0.161	0.14	0.838	0.86	0.022	0	0	0	2505	5352	2847
DRB1_0401	232	0.22	0.195	0.779	0.804	0.025	0	0	0	3116	6317	3201
DRB1_0402	3	0.286	0.206	0.712	0.793	0.081	0.07	0	-0.07	0	53	53
DRB1_0404	146	0.235	0.19	0.764	0.81	0.046	0	0	0	577	3657	3080
DRB1_0405	3	0.055	0.03	0.934	0.964	0.031	0	0	0	1582	3962	2380
DRB1_0701	197	0.208	0.179	0.791	0.821	0.029	0	0	0	1745	6325	4580
DRB1_0801	22	0.273	0.24	0.726	0.76	0.034	0.028	0	-0.028	0	937	937
DRB1_0803	3	0.3	0.305	0.696	0.693	-0.002	0.034	0	-0.034	0	0	0
DRB1_0818	1	0.568	0.469	0.425	0.525	0.1	0.063	0.028	-0.034	0	0	0
DRB1_0901	40	0.301	0.326	0.696	0.672	-0.024	0	0	0	1520	4318	2798
DRB1_1001	10	0.355	0.328	0.644	0.672	0.027	0.158	0	-0.158	0	2066	2066
DRB1_1101	196	0.177	0.14	0.822	0.859	0.037	0	0	0	1794	6045	4251
DRB1_1104	44	0.169	0.156	0.831	0.844	0.013	0.045	0	-0.045	0	0	0
DRB1_1201	2	0.114	0.086	0.887	0.914	0.027	0	0	0	117	2384	2267
DRB1_1301	12	0.423	0.245	0.576	0.754	0.178	0.046	0	-0.046	0	1034	1034
DRB1_1302	3	0.553	0.547	0.447	0.45	0.003	0	0	0	1580	4477	2897
DRB1_1401	20	0.226	0.206	0.773	0.795	0.021	0.029	0.115	0.086	0	0	0
DRB1_1501	122	0.218	0.184	0.781	0.815	0.034	0	0	0	1769	4850	3081
DRB1_1502	16	0.139	0.098	0.861	0.902	0.041	0.044	0	-0.044	0	0	0
DRB1_1503	2	0.241	0.296	0.759	0.7	-0.059	0.03	0	-0.03	0	0	0
DRB3_0101	4	0.17	0.068	0.83	0.932	0.102	0	0	0	1501	4633	3132
DRB3_0202	7	0.432	0.149	0.566	0.85	0.284	0.088	0	-0.088	0	3334	3334
DRB4_0101	3	0.411	0.372	0.589	0.628	0.039	0	0	0	1521	3961	2440
DRB5_0101	120	0.151	0.17	0.849	0.83	-0.019	0	0	0	3106	5125	2019
H-2-IAb	85	0.12	0.129	0.879	0.87	-0.009	0.05	0	-0.05	660	1794	1134
H-2-IAd	11	0.399	0.216	0.599	0.785	0.186	0.052	0	-0.052	379	774	395
HLA-DPA10103-DPB10201	1	0.015	0.02	0.985	0.98	-0.005	0	0	0	1404	787	-617
HLA-DQA10102-DQB10602	2	0.108	0.051	0.891	0.948	0.058	0.056	0	-0.056	1629	2747	1118
HLA-DQA10201-DQB10201	2	0.535	0.467	0.462	0.533	0.07	0.129	0	-0.129	0	0	0
Average		**0.257**	**0.211**	**0.742**	**0.789**							

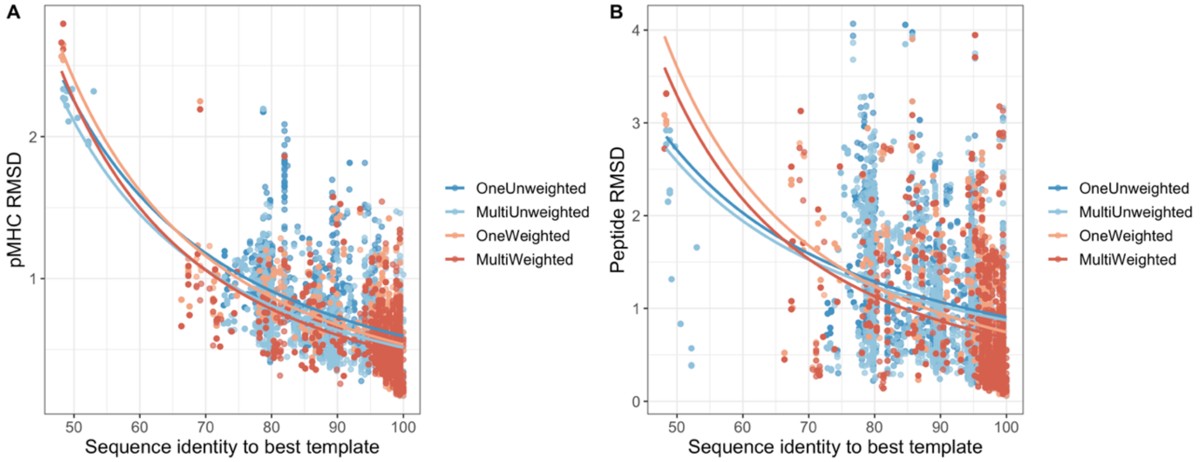
Paper II: Appendix

Additional Information

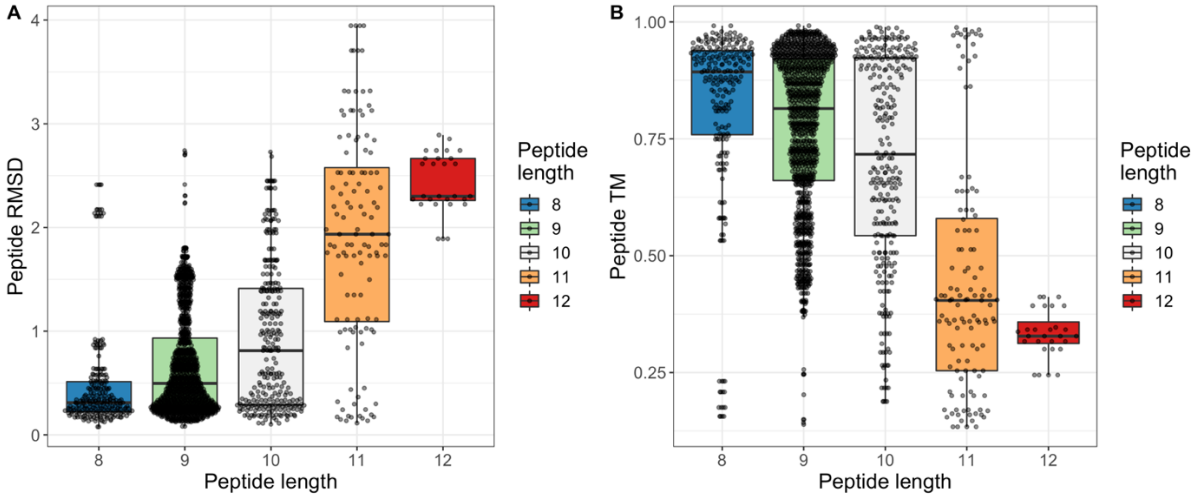
Supplementary figures



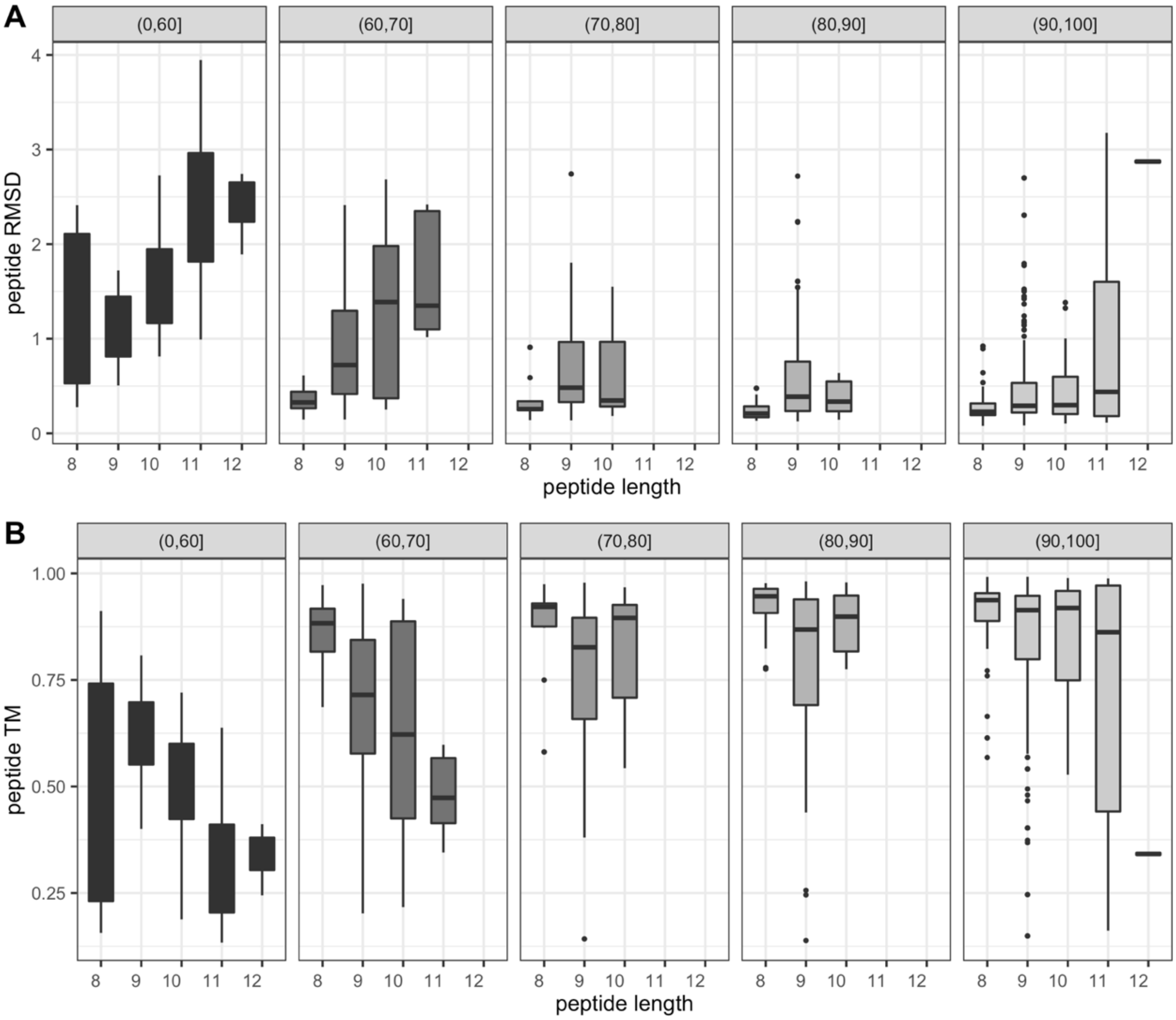
Supplementary Figure S1: TM-score performances for the different template selection methods. **A)** The TM-score performance for the pMHC complex. **B)** The TM-score for the peptide. For each target in the template database we generate four models using the four different sequence identity thresholds. Method OneUnweighted uses only a single template with a weighted sequence identity, while method MultiUnweighted uses multiple templates with a weighted sequence identity. Method OneWeighted used a single template and the weighted sequence identity. MultiWeighted uses multiple templates and the weighted sequence identity. The four different template selection methods are compared with a random baseline (see method for more details). Statistical comparison was performed using the Wilcoxon signed-rank test.



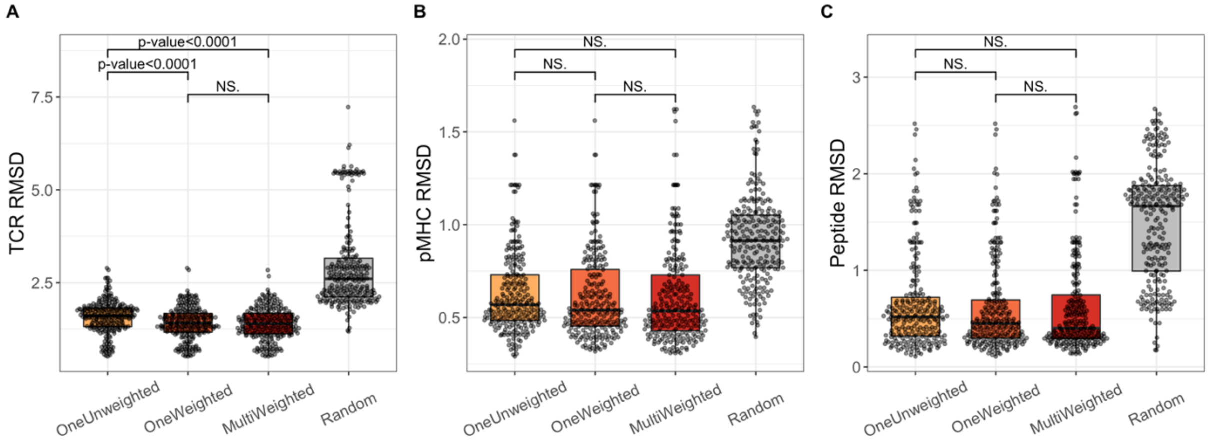
Supplementary Figure S2: Chothia-Lesk plot showing the RMSD performance for the pMHC models generated using the different template selection methods (see Method section). **A)** Shows the RMSD performance for the pMHC complex. **B)** Shows the RMSD performance for the peptide. The sequence identity to the best template is calculated using the unweighted sequence identity.



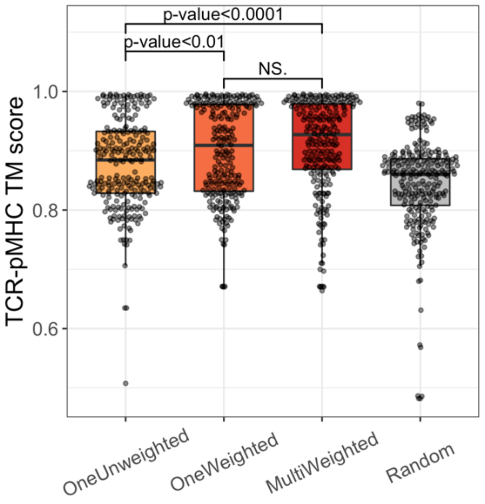
Supplementary Figure S3: A) Peptide RMSD performance for the pMHC models based on peptide length and **B)** the TM-score for the peptide for the pMHC models based on peptide length. Each pMHC model were produced using the MulitWeighted method for template selection.



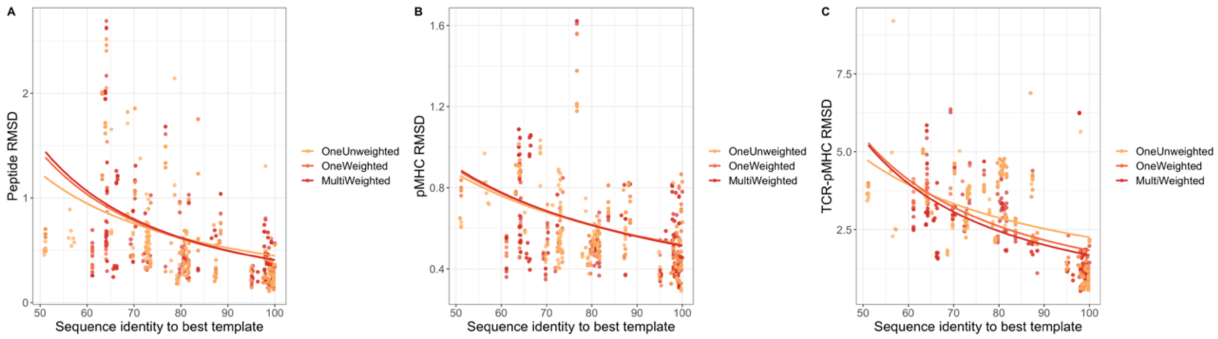
Supplementary Figure S4: A) Peptide RMSD accuracy for each pMHC model based on peptide length binned according to the sequence identity to the best template. **B)** TM-scores for the peptide for each pMHC model based on peptide length binned according to the sequence identity to the best template. THe results shown in these plots are based on the pMHC models produced using the MulitWeighted method for template selection.



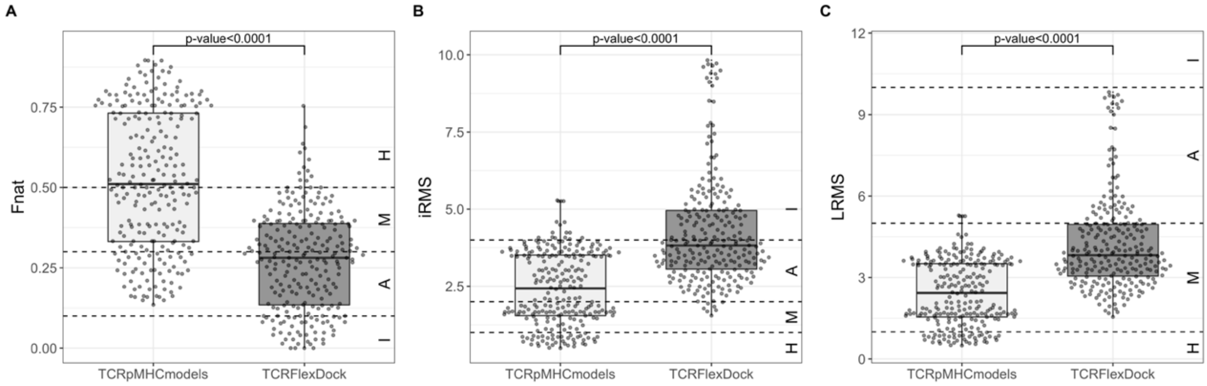
Supplementary Figure S5: The RMSD accuracy for the different template selection methods using **A)** the TCR RMSD, **B)** the pMHC RMSD and **C)** the peptide RMSD. For each target in the template database we generate four models using the four different sequence identity thresholds and evaluate the generated models using the RMSD for the TCR-pMHC complex. The OneUnweighted method uses only a single TCR-pMHC template with no weights on the sequence identity. The OneWeighted method uses only a single TCR-pMHC template and a weighted sequence identity. The MultiWeighted method uses the weighted sequence identity and multiple templates. The three different template selection methods are compared with a random baseline shown in grey. Statistical comparison was performed using the Wilcoxon signed-rank test.



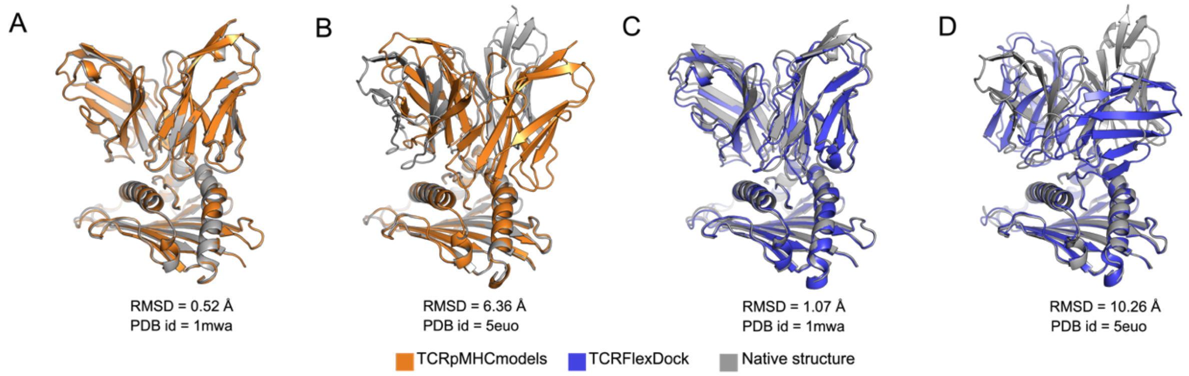
Supplementary Figure S6: The TCR-pMHC TM-score accuracy for the different template selection methods. For each target in the TCR-pMHC template database we generate four models using the four different sequence identity thresholds and evaluate the generated models using the RMSD for the TCR-pMHC complex. The OneUnweighted method uses only a single TCR-pMHC template with no weights on the sequence identity. The OneWeighted method uses only a single TCR-pMHC template and a weighted sequence identity. The MultiWeighted method uses the weighted sequence identity and multiple templates. The three different template selection methods are compared with a random baseline shown in grey. Statistical comparison was performed using the Wilcoxon signed-rank test.



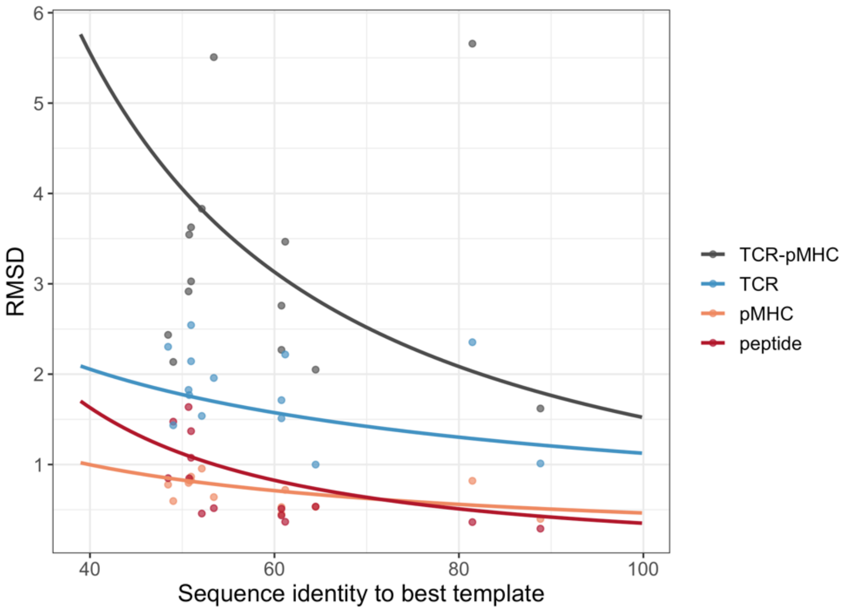
Supplementary Figure S7: Chothia-Lesk plot showing the RMSD accuracy for the TCR-pMHC models generated using the different template selection methods (see method section for more details). **A)** Shows the TCR-pMHC RMSD accuracy. **B)** Shows the pMHC RMSD accuracy. **C)** Shows the peptide RMSD accuracy. The sequence identity to the best template is calculated using the unweighted sequence identity.



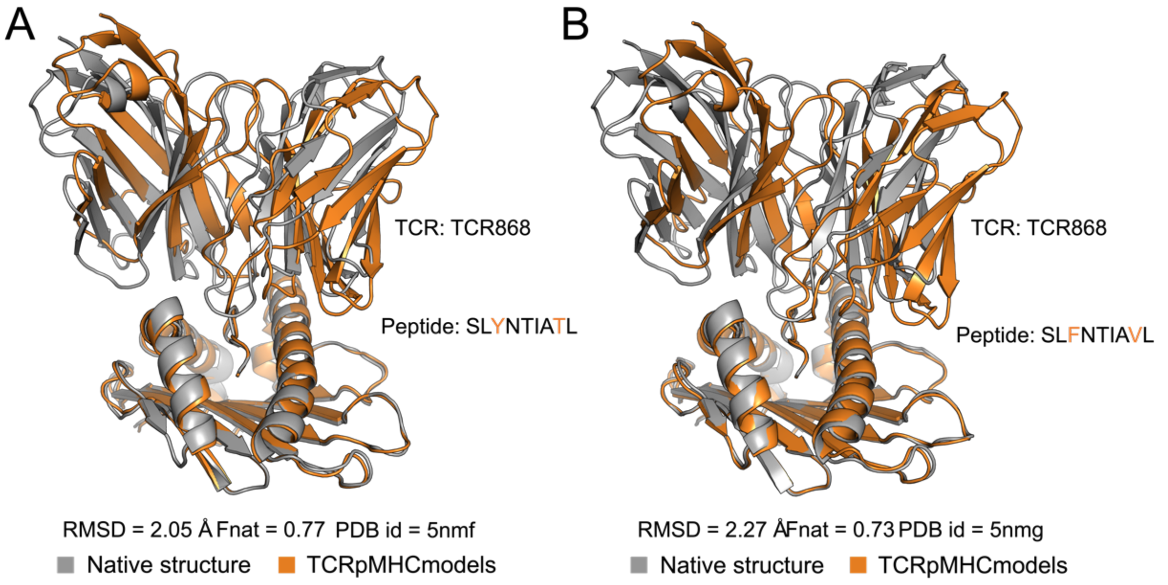
Supplementary Figure S8: Benchmark analysis of the TCR-pMHC models, showing different performance values between the models produced by TCRpMHCmodels and TCRFlexDock. **A)** Shows the Fnat accuracy **B)** Shows the iRMS accuracy and **C)** shows the LRMS accuracy. The statistical comparison was performed using the Wilcoxon signed-rank test and the dashed line indicates the thresholds for the four quality classes: High (H), Medium (M), Acceptable (A) and Incorrect (I) (see Method section).



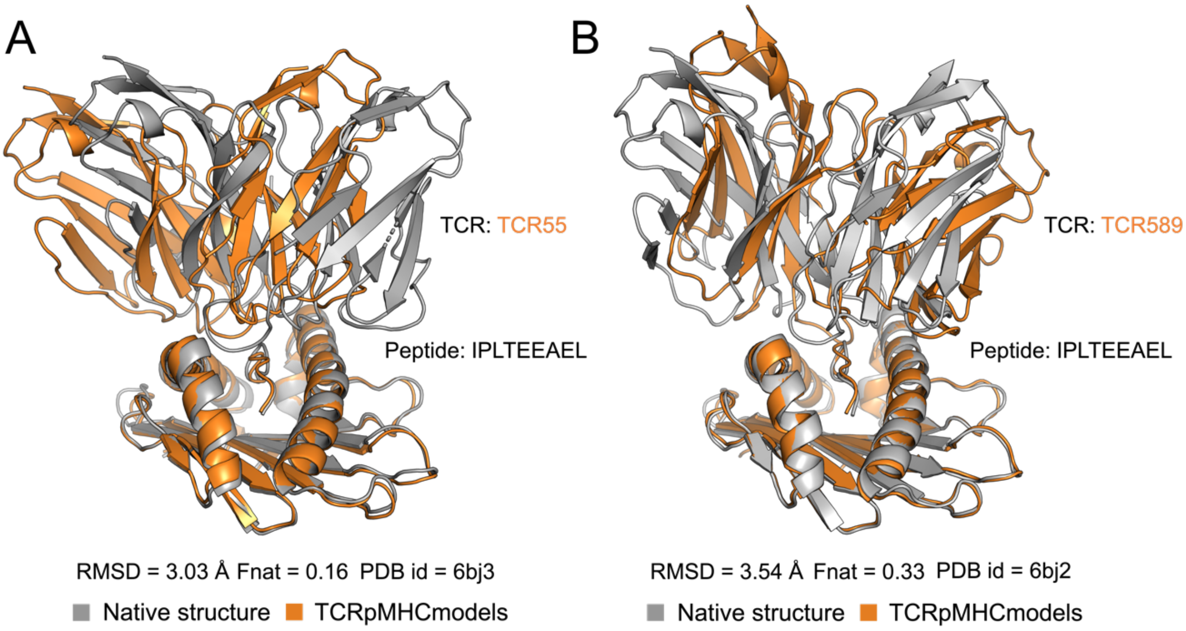
Supplementary Figure S9: Visualisation of high and low quality models generated by TCRpMHCmodels and TCRFlexDock. The orange and blue structures in the figure are models generated by TCRpMHCmodels and TCRFlexDock respectively, and native structures are shown in gray. To get a good view of the TCR orientation to the pMHC we superimposed only the pMHC, but the the RMSDs shown in the figure was calculated by superimposing all the C-alpha atoms in the TCR-pMHC model with all the C-alpha atoms in the native structure after which the RMSD was calculated for all C-alpha atoms. **A)** High quality model generated with TCRpMHCmodels, PDB id: 1mwa, RMSD: 0.52 and Fnat: 0.719. **B)** Low quality model generated with TCRpMHCmodels, PDB id: 5euo, RMSD: 6.36 and Fnat: 0.377. **C)** High quality model generated with TCRFlexDock, PDB id: 1mwa, RMSD: 1.07 and Fnat: 0.484. **D)** Low quality model generated with TCRFlexDock PDB id: 5hho, RMSD: 10.26 and Fnat: 0.020. Structural representations were made in PyMOL.



Supplementary Figure S10: The RMSD accuracy for the TCR-pMHC models generated using TCRpMHCmodels. The TCR-pMHC RMSD (grey), the TCR RMSD (blue), the pMHC RMSD (orange) and the peptide RMSD (read).



Supplementary Figure S11: Visualisation of a case where the same TCR binds different peptides. The native structures are shown in gray, while the models generated with TCRpMHCmodels are shown in orange. **A)** Shows the TCR-pMHC complex with the TCR55, PDB id: 6bj3. **B)** Shows the TCR-pMHC complex with the TCR589, PDB id: 6bj2. To get a good view of the TCR-pMHC models we superimposed only the pMHC, but the RMSDs shown in the figure was calculated by superimposing the C-alpha atoms in the TCR-pMHC model with the C-alpha atoms in the native structure. Structural representations were made in PyMOL.


A
TCR: TCR55
Peptide: IPLTEEAEL
RMSD = 3.03 Å Fnat = 0.16 PDB id = 6bj3
■ Native structure ■ TCRpMHCmodels
B
TCR: TCR589
Peptide: IPLTEEAEL
RMSD = 3.54 Å Fnat = 0.33 PDB id = 6bj2
■ Native structure ■ TCRpMHCmodels

Supplementary Figure S12: Visualisation of a case where the same TCR binds different peptides. The native structures are shown in gray while the models generated with TCRpMHCmodels are shown in orange. **A)** Shows the TCR-pMHC complex with the SLYNTIATL peptide, PDB id: 5nmf. **B)** Shows the TCR-pMHC complex with the SLFNTIAVL peptide, PDB id: 5nmg. To get a good view of the TCR-pMHC models we superimposed only the pMHC, but the RMSDs shown in the figure was calculated by superimposing the C-alpha atoms in the TCR-pMHC model with the C-alpha atoms in the native structure. Structural representations were made in PyMOL.

Supplementary tables

Supplementary table S1: The RMSD accuracy for the TCR-pMHC models generated using TCRpMHCmodels for the 14 TCR-pMHC structures not found in the TCR-pMHC database.

PDBid	TCR-pMHC rmsd	TCR rmsd	pMHC rmsd	Peptide rmsd	Sequence identity of best template
5isz	1.62	1.01	0.40	0.29	88.84
5ivx	5.51	1.96	0.64	0.52	53.43
5jzi	3.47	2.22	0.72	0.37	61.17
5nme	2.76	1.71	0.53	0.51	60.76
5nmf	2.05	1.00	0.53	0.53	64.46
5nmg	2.27	1.51	0.45	0.44	60.76
5tez	5.66	2.35	0.82	0.36	81.46
5wkf	2.14	1.43	0.60	1.47	49.03
5wkh	2.92	1.83	0.80	1.64	50.69
5wlg	3.83	1.54	0.95	0.46	52.12
5xot	3.63	2.14	0.87	1.37	50.96
6bj2	3.54	1.77	0.82	0.85	50.76
6bj3	3.03	2.54	0.82	1.07	50.96
6bj8	2.43	2.30	0.78	0.85	48.47
Mean	**3.20**	**1.81**	**0.69**	**0.77**	**58.85**