Epitope prediction methods

Karosiene, Edita; Nielsen, Morten; Lund, Ole

Publication date:
2013

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

PhD Thesis

Epitope prediction methods

Edita Karosiene

Center for Biological Sequence Analysis
Department of Systems Biology
Technical University of Denmark

August 30, 2013

Preface

This thesis was prepared at the Center for Biological Sequence Analysis (CBS), Department of Systems Biology, Technical University of Denmark (DTU) as a requirement for obtaining the PhD degree. The PhD was funded by the Graduate School of Immunology, Faculty of Health Sciences, University of Copenhagen, by DTU, and by the National Institutes of Health, Department of Health and Human Services, under contract numbers: HHSNN26600400006C (first IEDB contract), HHSN272201200010C (second IEDB contract), and HHSN272200900045C (Complete Analysis of T cell epitopes from Yellow Fever Virus).

The work was carried out at the Center for Biological Sequence Analysis, under the supervision of Associate Professor Morten Nielsen and Professor Ole Lund. Part of the work presented in Chapter 3 was done during my external stay at the Instituto Fundación Leloir in Buenos Aires, Argentina.

Edita Karosiene
August 2013

Contents

Abstract

Major histocompatibility complex (MHC) molecules play a crucial role in adaptive immunity by sampling peptides from self and non-self proteins to be recognised by the immune system. MHC molecules present peptides on cell surfaces for recognition by $CD8^+$ and $CD4^+$ T lymphocytes that can initiate immune responses. Therefore, it is of great importance to be able to identify peptides that bind to MHC molecules, in order to understand the nature of immune responses and discover T cell epitopes useful for designing new vaccines and immunotherapies. MHC molecules in humans, referred to as human leucocyte antigen (HLA) molecules, are encoded by extremely polymorphic genes on chromosome 6. Due to this polymorphism, thousands of different MHC molecules exist, making the experimental identification of peptide–MHC interactions a very costly procedure. This has primed the need for *in silico* peptide–MHC prediction methods, and over the last decade several such methods have been successfully developed and used for epitope discovery purposes.

My PhD project has been dedicated to improve methods for predicting peptide–MHC interactions by developing new strategies for training prediction algorithms based on machine learning techniques.

Several MHC class I binding prediction algorithms have been developed and due to their high accuracy they are used by many immunologists to facilitate the conventional experimental process of epitope discovery. However, the accuracy of these methods depends on data defining the MHC molecule in question, making it difficult for the non-expert end-user to choose the most suitable predictor. The first paper in this thesis presents a new, publicly available, consensus method for MHC class I predictions. The *NetMHCcons* predictor combines three state-of-the-art prediction tools and provides the most accurate predictions for any given MHC molecule.

While the methods for MHC class I binding have reached a very high accuracy and are widely used for immunological research, the case of MHC class II is less clear. The open binding groove of MHC class II molecules and differences in polymorphism among MHC encoding genes makes predictions of peptide binding to MHC class II molecules a complicated problem. We addressed these issues in order to develop the first pan-specific predictor common for all three human class II isotypes, HLA-DR, HLA-DP and HLA-DQ. The second paper introduces the *NetMHCIIpan-3.0* predictor based on artificial neural networks, which is capable of giving binding affinities to any human MHC class II molecule.

Chapter 4 of this thesis gives an overview of bioinformatics tools developed by the Immunological Bioinformatics group at Center for Biological Sequence Analysis. The chapter provides detailed explanations on how to use different methods for T cell epitope discovery research, explaining how input should be given as well as how to interpret the output.

In the last chapter, I present the results of a bioinformatics analysis of epitopes from the yellow fever virus. The analysis demonstrated the absence of

distinct regions of higher epitope density within the virus polyprotein. Also, the density of epitopes among different proteins was demonstrated to mostly depend on protein length and amino acid composition, underlining the importance of identifying peptide–MHC interactions. Furthermore, using yellow fever virus epitopes, we demonstrated the power of the %Rank score when compared with the binding affinity score of MHC prediction methods, suggesting that this score should be considered to be used for selecting potential T cell epitopes.

In summary, this thesis presents methods for prediction of peptides that bind to both MHC class I and class II molecules, which is important for driving immunological research within the field of T cell epitope discovery and for general understanding of the cellular responses.

Dansk resumé

Major histocompatibility complex (MHC) molekyler spiller en afgørende rolle i adaptiv immunitet, ved at udvælge peptider fra egne og ikke-egne proteiner til genkendelse af immunsystemet. MHC molekyler præsenterer peptider på celleoverfladen, som så genkendes af $CD8^+$ og $CD4^+$ T lymfocytter, som er i stand til at initiere et immunrespons. Det er derfor yderst vigtigt at kunne identificere peptider, som binder til MHC molekyler, for derved at kunne forstå arten af immunresponset, samt at opdage T celle epitoper, som kan bruges til at designe nye vacciner og immunterapi. Humane MHC molekyler, humant leukocyt antigen (HLA) molekyler, kodes af ekstremt polymorfe gener på kromosom 6. Grundet denne polymorfi eksisterer der tusindvis af forskellige MHC molekyler, hvilket gør den eksperimentelle identifikation af peptid–MHC interaktioner til en særdeles bekostelig procedure. Af denne årsag er behovet for *in silico* peptid–MHC forudsigelsesmetoder opstået, og over det sidste årti er adskillige sådanne metoder med succes blevet udviklet og anvendt til bestemmelse af epitoper.

Mit ph.d.-projekt har været dedikeret til at forbedre metoder til forudsigelse af peptid–MHC interaktioner, ved at udvikle nye strategier til træning af forudsigelsesalgoritmer baseret på machine learning-teknikker.

Adskillige algoritmer til forudsigelse af MHC klasse I binding er blevet udviklet, og anvendes, grundet deres høje nøjagtighed, af mange immunologer til at lette den konventionelle eksperimentelle proces til bestemmelse af epitoper. Nøjagtigheden af disse metoder afhænger dog af de forudsætninger, som definerer det pågældende MHC molekyle. Dette gør det svært for en ikke-ekspert slutbruger at vælge den bedst egnede forudsigelsesalgoritme. Den første artikel i denne afhandling præsenterer en ny, offentligt tilgængelig konsensusmetode for MHC klasse I forudsigelser. *NetMHCcons* forudsigelsesalgoritmen kombinerer tre state-of-the-art forudsigelsesværktøjer og giver den mest nøjagtige forudsigelse for ethvert MHC molekyle.

Mens metoderne for MHC klasse I binding har opnået en meget høj nøjagtighed og anvendes vidt omkring til immunologisk forskning, så er situation med MHC klasse II mindre afklaret. En åben bindingskløft, samt forskelle i polymorfi imellem de MHC-kodende gener, gør forudsigelsen af peptider til MHC klasse II molekyler til et kompliceret problem. Vi adresserede disse problemstillinger med henblik på at udvikle den første pan-specifikke forudsigelsesalgoritme, fælles for alle tre humane klasse II isotyper, HLA-DR, HLA-DP og HLA-DQ. Den anden artikel introducerer *NetMHCIIpan-3.0* forudsigeren baseret på kunstige neurale netværk, som er i stand til at angive bindingsaffinitet til ethvert humant MHC klasse II molekyle.

Kapitel 4 i denne afhandling giver et overblik over de bioinformatiske værktøjer, som er udviklet af gruppen for Immunologisk Bioinformatik på Center for Biologisk Sekvensanalyse. Kapitlet giver detaljerede beskrivelser af, hvordan de forskellige metoder til forskning inden for T celle epi-

topopdagelse anvendes, samt hvordan input skal gives og hvordan output skal fortolkes.

I det sidste kapitel præsenterer jeg resultaterne af en bioinformatisk analyse af epitoper fra gul feber virus. Analysen fandt at der ikke er nogle regioner internt i poly-proteinet, der indeholder en højere forekomst af epitoper. Derudover blev det vist, at tætheden af epitoper imellem forskellige proteiner mest afhang af proteinets længde og aminosyresammensætningen, hvorved vigtigheden af at forudsige peptid–MHC interaktioner blev demonstreret. Ydermere har vi, ved hjælp af gul feber virus epitoper, demonsteret styrken af %Rank scoren sammenlignet med bindingsaffinitetsscoren fra MHC forudsigelsesmetoderne, og konkluderer at denne score bør anvendes frem for bindingsaffinitet til at udvælge potentielle T celle epitoper.

For at opsummere, præsenterer denne afhandling metoder til forudsigelse af peptidbindinger til både MHC klasse I og klasse II molekyler, hvilket er vigtigt for at fremme immunologisk forskning i feltet for T celle epitopbestemmelse, og for generel forståelse af det cellulære respons.

Acknowledgements

It has been a great pleasure and an invaluable experience to work as a PhD student at the Center for Biological Sequence Analysis. I consider myself very lucky because I was surrounded by so many interesting, smart, helpful and friendly people who created a great working environment. I thank the head of CBS Søren Brunak and all the CBSians for making this place so special.

I would like to express my gratitude to my main PhD supervisor Morten Nielsen for the great supervision and support. Thank you for believing in me and sharing with me your enthusiasm and passion for research. Thank you for giving me opportunities to work on very exciting and challenging projects and for your never-ending bright ideas. I am grateful for a very motivating working environment that you created. Even thousands of kilometers away, you can still be the best supervisor.

A big thank you also to my co-supervisor Ole Lund for his valuable contribution to the projects, discussions and for his endless optimism. Also, for being a great leader and creating an extraordinary environment within the Immunological Bioinformatics group at CBS.

I would like to thank our collaborators from University of Copenhagen, Søren Buus, Anette Stryhn Buus, Michael Rasmussen, and Thomas Holberg Blicher, for their contribution to the projects.

Thank you to everyone from the Immunological Bioinformatics groups for sharing lots of valuable moments scientifically and socially. Thank you all for uncountable number of group meetings we had, for all the vaccine days and dinners and for being the best chocolate tasters ever!

I am grateful to the CBS system administrators for their technical support. I thank John Damm Sørensen for being always so helpful with the issues related to the queuing system and always responding very fast. Special thanks goes to Kristoffer Rapacki for his extraordinary ability to encourage and to motivate and help within any aspects of work and life, and for his unforgettable stories. Thanks also goes to the first person I have met from CBS and my first office mate, Peter Wad Sackett. Thank you for sharing your passion for programming and Perl, for trusting me and letting teach your students, and for having lots of (not only) work related talks.

The CBS administration was always very helpful during all these years and I am very grateful for that. Special thanks to Lone Boesen, Dorthe Kjærsgaard, Marlene Beck and Karina Sreseli for helping me with all formalities and organisational issues. Thank you, Lone, for booking many trips for me.

During my PhD I had a fantastic opportunity to go for an external stay to Argentina. I am very grateful to Cristina Marino Buslje from Fundación

Instituto Leloir for a very warm (also literally) welcome to her lab. I also wish to thank all the people that I have met there who introduced me to Argentinian culture and accompanied me through the period of adventures there. The moments I spent in Argentina will stay in my memories forever.

I am also grateful to Sebastian Carrasco Pro and Mirko Zimic from Universidad Peruana Cayetano Heredia for inviting me to give a workshop and organising it. It has been an invaluable experience for me, both teaching and exploring Peru.

Besides work, CBS is a great place to meet very nice people to have a great time with and become friends. I would like to thank all the people that I had a pleasure to share office with, especially Bent, Leon and Massimo with whom we had a lot of nice moments and interesting conversations. I am also grateful for people from our lunch club: Andrea, Anne, Christian, Jens F., Jens K., Mette, and Thomas. Thank you for introducing me to the Danish cuisine and Danish culture and for all the entertaining conversations that we always have. Special thanks goes to Anne for adventurous times in Argentina and for contributing to broadening the circle of my Danish friends.

I would like to acknowledge Andrea for proof-reading and commenting on my thesis. Thank you for your very valuable corrections and comments and for showing so much support during my last days of writing.

A big thank you goes to my friends in Denmark, especially Ola, Adam, Olek and Joana, for their support and interest in my research as well as for the moments we share together. Also to my friends in Lithuania for still keeping in touch and always waiting for me to come visit.

Finally, I would like to thank all my family for believing in me. I am grateful to my parents for giving me an opportunity to reach my goals and for always motivating me. Last but not least, I am deeply grateful to my beloved husband Vaidas for always being extremely supportive and helping to keep my positive spirit. Thank you for always being there for me!

Thank you all!

Papers included in this thesis

- **Paper I:**
  Edita Karosiene, Claus Lundegaard, Ole Lund, and Morten Nielsen
  ***NetMHCcons*: a consensus method for the major histocompatibility complex class I predictions.**
  Immunogenetics 2011, 64: 177–186.

- **Paper II:**
  Edita Karosiene, Michael Rasmussen, Thomas Blicher, Ole Lund, Søren Buus, and Morten 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 2013, (Epub ahead of print).

- **Paper III:**
  Ole Lund, **Edita Karosiene**, Claus Lundegaard, Mette Voldby Larsen, and Morten Nielsen
  Bioinformatics identification of antigenic peptide: predicting the specificity of major MHC class I and II pathway players.
  Methods in Molecular Biology 2013, 960:247–260.

Abbreviations

AN	Actual negative
ANN	Artificial neural network
AP	Actual positive
APC	Antigen presenting cell
ARB	Average relative binding
AUC	Area under the ROC curve
β2m	β2-microglobulin
CTL	Cytotoxic T lymphocyte
ELISPOT	Enzyme-linked immunosorbent spot
ER	Endoplasmic reticulum
FN	False negative
FP	False positive
HLA	Human leukocyte antigen
ICS	Intracellular cytokine staining
IEDB	Immune Epitope Database
LOO	Leave-one-out
MHC	Major histocompatibility complex
OLP	Overlapping peptide
PCC	Pearson's correlation coefficient
PDB	Protein Data Bank
PFR	Peptide flanking residues
PSSM	Position-specific scoring matrix
RMSE	Root-mean-square deviation
ROC	Receiver operating characteristic
SB	Strong binder
SMM	Stabilized matrix method
TAP	Transporter associated with antigen processing
TCR	T cell receptor
TN	True negative
TP	True positive
WB	Weak binder
YF	Yellow fever

Chapter 1

Introduction

Major histocompatibility complex (MHC) molecules play a key role in the adaptive immune responses by presenting peptides to the immune system. In the case of non-self origin, these peptides are known as epitopes and are the potential cause of an immune response. A large part of immunological research is dedicated to the discovery of new epitopes, with the goal to develop new vaccines and immunotherapies. Due to the thousands of MHC variants that exist, it is a highly labour and cost intensive procedure to perform direct experimental studies of peptide–MHC interactions. Therefore, methods predicting peptide binding to MHC act as resource saving tools in epitope discovery.

1.1 The adaptive immune system

The main role of the immune system is to protect organisms from infectious diseases, pathogens, foreign agents and tumour cells. This is achieved by the generation of various cells and molecules capable of differentiating self from non-self components and eliminating them. In jawed vertebrates the immune system consists of two parts: innate immunity and adaptive immunity, which work in collaboration to protect the body. The innate immune system acts as the first line of defence, protecting the organism from invading pathogens in a rapid and non-specific manner. The adaptive immune system, on the other hand, is capable of recognizing foreign agents in a specific manner, eliminating them, and developing an immunological memory so that, if the immune system encounters the same pathogen again, a rapid and highly effective response will be triggered.

The adaptive immune system comprises a large number of cells and molecules that participate in the processes of non-self antigen recognition, pathogen elimination and development of immunological memory. The two arms of the adaptive immune system are the humoral immune system, with the key cells being B lymphocytes, and the cellular immune system, where T lymphocytes play the main role. B lymphocytes, or B cells, secrete antibodies that circulate in blood plasma and lymph, and are able to recognise pathogenic invaders and neutralise them. Another important function of B cells is their ability to differentiate into memory cells that have a long lifespan and trigger a fast immune response in case of the host encountering the same antigen again. As opposed to B cells, T cells are not capable of recognising free antigens, and their interactions with MHC molecules are required for the activation of the immune system. There are two classes of MHC molecules, MHC class I and MHC class II, which are associated with different groups of T cells [1, 2]:

- Cytotoxic T lymphocytes (CTL) recognise peptides from the intercellular environment presented on MHC class I molecules expressed on all nucleated cells. CTLs are effector cells with the ability to recognise and kill infected cells. CTLs are also known as $CD8^+$ cells due to the CD8 glycoprotein receptor they express on the surface [3, 2].

- T helper cells are also called $CD4^+$ cells since they express the CD4 receptor. Through secretion of various cytokines, T helper cells activate other cells of the immune system such as B cells, $CD8^+$ cells, macrophages etc. $CD4^+$ cells recognise antigens from the extracellular environment loaded on MHC class II molecules expressed by so-called antigen presenting cells (APC) including B cells, macrophages and dendritic cells [4, 2].

1.1.1 Major histocompatibility complex molecules

Major histocompatibility complex molecules play a crucial role in the cellular immune system by presenting pathogen-derived peptides on the cell surface for recognition by T lymphocytes. Peptides presented by MHC class I and class II molecules originate from different sources via different antigen presentation pathways. Moreover, MHC class I and class II molecules differ in their protein structure, leading to a different conformation of their binding cleft. This section presents MHC class I and class II molecules from a structural and functional point of view.

1.1.2 MHC class I

Peptides presented by MHC class I molecules derive from cytosolic and nuclear proteins that have been degraded by the proteasome. Such peptides, usually composed of 8–11 amino acids, are transported into the endoplasmic reticulum (ER) by the transporter associated with antigen processing (TAP). The ER is the place where MHC molecules are assembled and loaded with peptides that by their structure and biological properties fit into the binding groove of the MHC molecule [2, 5]. Peptide binding to MHC is a very selective process. It has been estimated that only one out of 200 random natural peptides would bind to a given MHC class I molecule [6]. When a peptide binds to an MHC class I molecule, the stable peptide–MHC complex, with the help of chaperones, is transported to the cell surface through the Golgi apparatus (Figure 1.1a) [2, 5].

MHC class I molecules are membrane glycoproteins composed of a heavy chain (α) and a β2-microglobulin (β2m) chain. The α chain consists of three domains: transmembrane α3 domain and membrane-distal α1 and α2 domains (Figure 1.2a). The α1 and α2 domains form the binding groove [7, 8]. For MHC class I molecules, the binding groove is closed at both ends, restricting the length of binding peptides to fall in the range of 8–11 amino acids (Figure 1.2b).

In humans, MHC molecules are called human leucocyte antigen (HLA) system. Heavy MHC class I chains are encoded on chromosome 6 by three very polymorphic genes: HLA-A, HLA-B, and HLA-C. Thousands of different allelic versions of HLA exist [9] with the most polymorphic residues located within the binding cleft, resulting in a large number of possible MHC molecules with different binding specificities [2]. Such a vast polymorphism is a huge drawback in the process of epitope discovery and creation of new vaccines. Therefore, pan-specific methods capable of giving predictions even for uncharacterized MHC molecules play a very important role. Such pan-specific peptide–MHC binding prediction algorithms are presented later in this chapter followed by the details on specific tools given in Chapter 2 and 4 of this thesis.

1.1.3 MHC class II

As opposed to MHC class I, MHC class II molecules bind peptides derived from proteins taken up from extracellular environment. MHC class II molecules are initially assembled in the endoplasmic reticulum

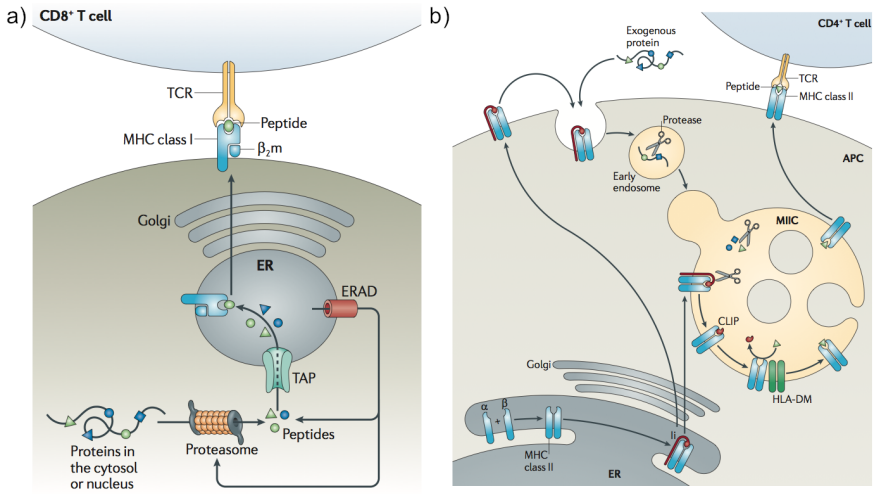

Figure 1.1. MHC class I and class II antigen presentation pathways. a) MHC class I presents peptides from cytosolic and nuclear proteins degraded by the proteasome. These peptides are transported into the ER by the TAP receptor where they are loaded on MHC class I molecules. Stable peptide–MHC complexes are then transported to the cell surface through the Golgi system. **b)** MHC class II molecules bind peptides from exogenous proteins that are taken up into the cell by endocytosis or phagocytosis. MHC class II molecules are assembled within the ER and transported through the Golgi system to MIIC. Eventually endosomes containing degraded proteins fuse with the MIIC compartment, where peptides are loaded on MHC class II molecules. Finally, peptide–MHC complexes are transported to the plasma membrane for presentation to $CD4^+$ cells. Figure source [2].

and transported through the Golgi aparatus to the MHC class II compartment (MIIC). Proteins taken up by the process of endocytosis or phagocytosis are cleaved by proteases and form endosomes that eventually fuse with the MIIC. Finally, after loading the peptides on MHC molecules within MIIC, peptide–MHC complexes are transported to the plasma membrane to be presented to $CD4^+$ T lymphocytes (Figure 1.1b) [2, 10].

Structurally, MHC class II molecules are very different from MHC class I. MHC class II molecules are heterodimers composed of α and β chains, both having transmembrane domains ($\alpha 2$ and $\beta 2$) and membrane-distal domains ($\alpha 1$ and $\beta 1$, Figure 1.2c) [11, 12]. The $\alpha 2$ and $\beta 2$ domains compose the peptide binding pocket which is open at both ends and is able to accommodate peptides of 15–20 residues, or even whole proteins (Figure 1.2d) [13, 14].

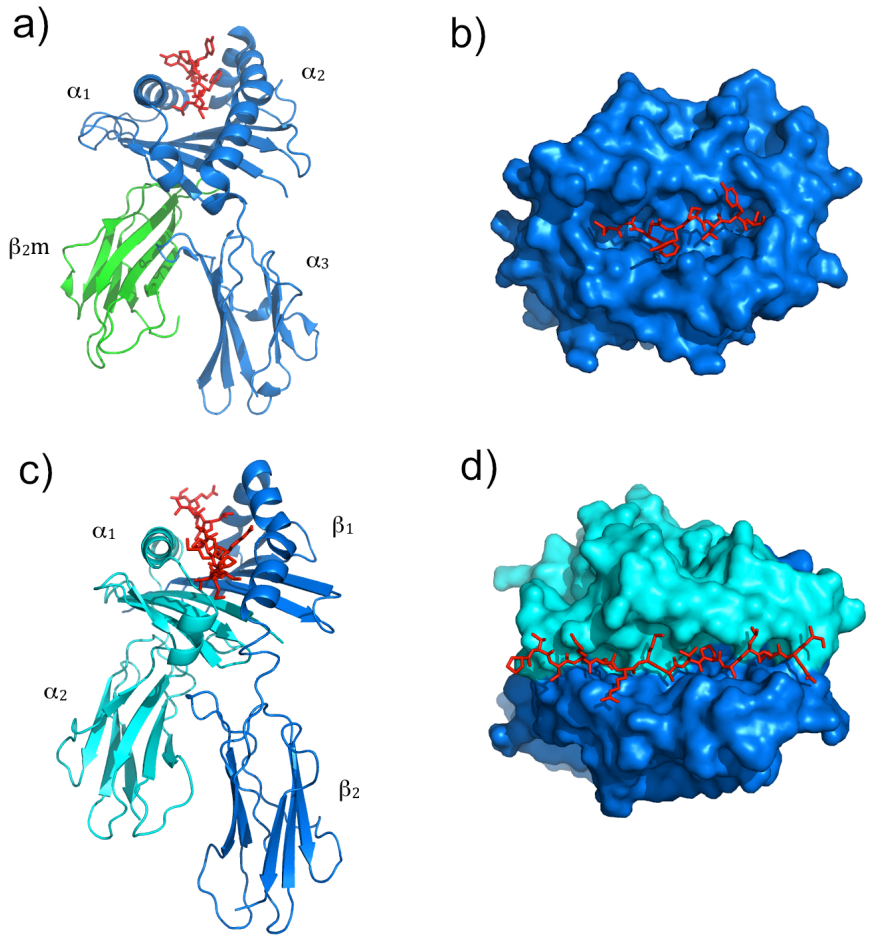

Figure 1.2. Protein structures of MHC class I and MHC class II molecules. **a)** MHC class I structure of HLA-A*0201 with peptide LLFGYPVYV (PDB entry 1DUZ [15]) showing α1, α2, α3 domains (*light blue*), β2-microglobulin (β2m) (*green*) and the peptide (*red*). **b)** Binding pocket of the same peptide–MHC complex demonstrating the closed binding groove of MHC class I molecules. **c)** Structure of MHC class II molecule HLA-DRA*0101-DRB1*0301 with the 15-mer peptide PVSKMRMATPLLMQA (PDB entry 1A6A [16]). α1 and α2 domains are shown in *cyan*, β1 and β2 in *light blue*, and the peptide is shown in *red*. **d)** Top view of the same MHC class II and peptide complex showing that the binding groove is open at both ends and that the peptide extends outside of the pocket. The figure was made using the PyMOL software [17].

In humans, MHC class II encoding genes also lie within chromosome 6. Three different loci encode the α and β chains: HLA-DR, HLA-DP, and HLA-DQ. In the case of HLA-DR, only β chain encoding loci (DRB) have been found to be polymorphic, therefore

the binding specificities of HLA-DR molecules are determined by DRB, and a specific molecule can be denoted by an allele name, e.g. HLA-DRB1*0101. Regarding HLA-DP and HLA-DQ, both α and β chains display polymorphism and a specific molecule can be identified by specifying the names of the alleles encoding both chains, e.g. HLA-DPA1*0201-DPB1*0501 [11, 10, 18].

1.2 Bioinformatics methods

This section presents bioinformatics tools used in the work presented in this thesis, including some details about the machinery underlying our prediction methods, as well as ways to evaluate and benchmark them.

1.2.1 Artificial neural networks

In biology, binding motifs of amino acid or nucleotide sequences in biology can be predicted by different methods. In cases where binding specificity at one position of the motif is independent from the other positions, so-called position-specific scoring matrices (PSSM) can be used [19] . Such matrix-based approaches often act as a good first approximation of the receptor binding motif. However, due to structural constraints, ligand residues must compete for the space in the receptor binding pocket. This is also the case for MHC molecules. Indeed, it was demonstrated by Nielsen et al. that there is a signal of mutual information between the seven non-anchor residue positions (1, 3, 4, 5, 6, 7, and 8) of HLA molecules [20]. To handle such mutual information, higher order correlations need to be captured, and this can be solved by artificial neural networks (ANN) with hidden layers.

Artificial neural networks belong to the group of machine learning techniques that at first are trained to associate different patterns from the large sets of data, and are capable of predicting the outcome of a new example afterwards. Structurally, ANNs are similar to biological neural networks within the brain. Neural networks consist of a large number of interconnected neurons influencing each other by sending information via synapses. The most simple, and most often used in bioinformatics, is a so-called feed-forward multilayer network. The structure of the feed-forward network is similar to the structure described by Rumelhart et al. [21]. Feed-forward multilayer networks were also used for development of the methods presented in this work.

A schematic representation of a feed-forward neural network is shown in Figure 1.3. The network consists of an input layer composed of five neurons, one hidden layer with four neurons and an output layer with a single neuron. The input layer contains input data in some numeric encoding (presented later in this chapter). An artificial neural network may contain more hidden layers, and the output layer may consist of many neurons. Each unit of the network receives signals via synapses with associated weights from the neurons of the previous layer. These weights are real numbers quantifying the influence that one neuron has on the other. A total influence received by one neuron can be calculated by summation of all the weighted inputs. In our example, the input received by the output layer neuron O_j would be expressed as:

$$o_j = \sum_i H_i w_{ij} \tag{1.1}$$

An output of each neuron is then calculated as a function of its input using a non-linear transfer function $f(x)$ as follows:

$$O_j = f(o_j) \tag{1.2}$$

The most commonly used function is a sigmoid function expressed as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1.3}$$

A neuron is activated and sends input to the neurons in the next level if its output value is above a certain threshold.

Training of ANNs

In order for an artificial neural network to give predictions, it first needs to learn patterns in the data from a set of training examples. The training process of an ANN corresponds to the optimization of network parameters that in the feed-forward networks correspond to the weights between neurons. During training, the network receives associated input and output values from the training set. Initially, all weights are assigned to random values and are updated using back-propagation during multiple iterations to minimize the error between the output presented to the network as the target and the output calculated by the network. Back-propagation is conventionally implemented using gradient descent methods [22].

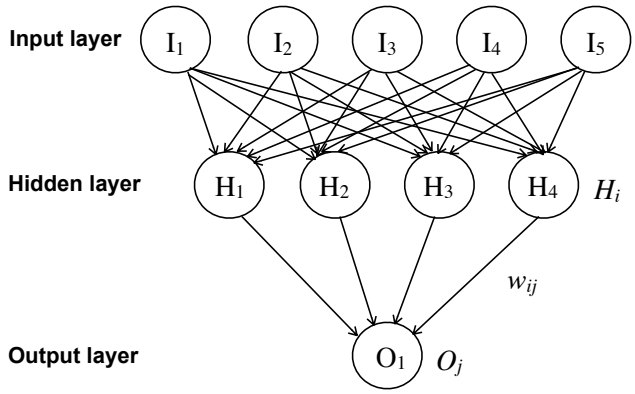

Figure 1.3. Schematic representation of a feed-forward network with one hidden layer. The units of each layer are connected to all the neurons from the next layer. In our example, hidden layer (i) is fully connected to the output layer (j) by the weigths w_{ij}.

Biological sequence encoding

The mathematical model behind the architecture of neural networks implies that the input to the network must come in numerical format. In biology, however, we normally have sequences of peptides or proteins that we want to give as an input to the neural network. To allow this, several encoding schemes for converting amino acid sequences to numerical strings have been developed. Two of the most often used schemes are sparse and BLOSUM encodings. In sparse encoding, each amino acid is represented as a 20-bit vector composed of 19 zeros, and one position, whose index encodes the amino acid, is set to "1". For example, alanine as the first amino acid in the alphabet would be represented as 10000000000000000000.

As opposed to sparse encoding, the BLOSUM encoding scheme is based on biological properties of amino acids and takes their similarities into account [23]. The encoding is based on a 20×20 substitution matrix and the intersection between two amino acids is associated with a substitution value based on their similarity [20]. The most commonly used (and the one used in the work presented here) is the BLOSUM50 matrix.

1.2.2 Pan-specific prediction methods

Prediction methods for peptide–MHC binding can be divided into two main groups: allele-specific and pan-specific methods. Allele-

specific predictors are trained using only peptide sequences and their binding affinities to one specific MHC molecule. Therefore, these methods are able to give predictions only to the molecules that have been part of the training set. On the contrary, pan-specific methods have a power to extrapolate from known MHC molecules to the ones with limited or no experimental peptide binding data. This is achieved by including information about MHC molecules in the training procedure.

For the pan-specific methods presented in this thesis, MHC class I and class II molecules are represented by a so-called pseudo sequence. Pseudo sequence consists of residues from the peptide binding groove of the molecule that are in potential contact with the peptide. These residues are within 4.0 Å of the peptide in one or more MHC class I or class II structures available in the Protein Data Bank (PDB). In addition, only the important residues that are polymorphic across sequenced MHC molecules are included in the pseudo sequence. Figure 1.4 shows pseudo sequence in the structure of HLA-A*0201.

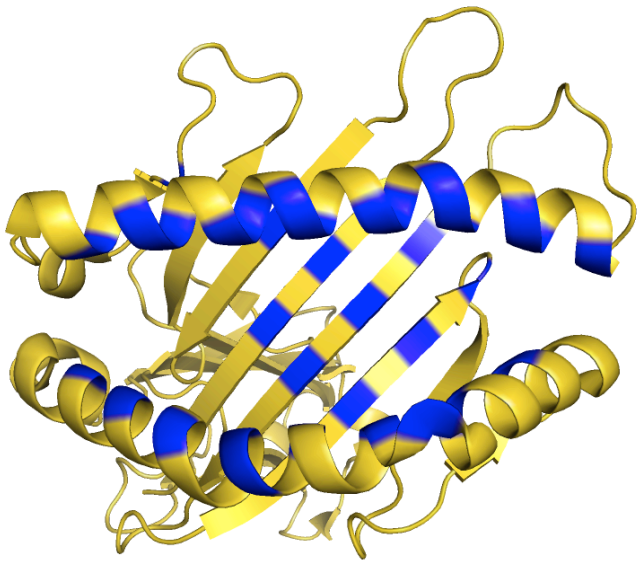

Figure 1.4. MHC pseudo sequence representation. Residues comprising pseudo sequence are highlighted in *blue* in the structure of the HLA-A*0201 molecule (PDB entry 1DUZ [15]).

Pseudo sequence representing MHC molecules is encoded and fed to the neural network together with the peptides and their associated

binding affinity. Binding affinities are also transformed to be suitable for the ANN training by making them fall between 0 and 1 using the relation: $1-\log(\text{IC50nM})/\log(50{,}000)$. An example input for training of a pan-specific method is given in Table 1.1.

Table 1.1. Example input, before encoding, for pan-specific prediction methods presented in this thesis. MHC molecules are represented by pseudo sequences. Binding affinities are log-transformed to fall between 0 and 1.

Peptide	Pseudo sequence	Log-affinity
AEFWDVFLS	YFAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY	0.0847
ADPVDAVIN	YYAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY	0.2890
IRHHVRWAL	YHTEYRNICAKTDVGNLYWTYNFYTWAVLAYEWH	0.4350
YIRRNMINK	YYAMYRNNVAQTDVDTLYIMYRDYTWAVWAYTWY	0.5266
KAGQYVTIW	YDSGYREKYRQADVNKLYLWYDSYTWAEWAYTWY	0.3436
YTAVVPLVS	YTAMYLQNVAQTDANTLYIMYRDYTWAVLAYTWY	0.0014

1.2.3 Cross-validation

In order to assess the predictive performance of the neural networks, we have to evaluate how well the method generalises to an independent data set. This is done by training the ANNs using cross-validation, where a full data set is divided into a number (n) of subsets. The training procedure is repeated n times with the ANNs being trained on $n-1$ subsets. Each time the remaining subset is used as a test set to obtain predictions. At the end, the results are combined and a performance score is calculated. In the work presented here we used 5-fold cross-validation as depicted in Figure 1.5. The networks are using $4/5$ of the data for training and $1/5$ to test the performance. The procedure is repeated five times so that each subset is used as a test set once.

Nested cross-validation

The above mentioned cross-validation procedure is very commonly used with machine learning techniques. However, in some cases, in addition to evaluation of the predictive performance, the test set is used to stop the training in order to avoid overfitting. The stopping procedure can be very important when training neural networks in order to ensure their generalisability. As depicted in Figure 1.6, the prediction error on the training set continuously decreases during training, while the error on the test set reaches a minimum and

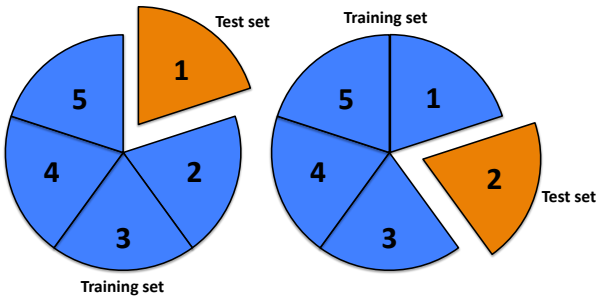

Figure 1.5. Schematic representation of a 5-fold cross-validation setup. In each round, one subset of the data is used as a test set (*orange*) while the other 4/5 of the data set (*blue*) are used to train the neural network. The procedure is repeated five times, giving the role of a test set to a different subset once.

starts to increase. This means that the network became too specific to the training set and is able to learn it better and better. At the same time, the network is loosing its ability to generalize on the test set, which means that the network shows overfitting. Therefore, the test set can be used to terminate training when the prediction error reaches the minimum. However, using cross-validation training in this setup, there is a potential to overestimate the predictive performance due to the fact that the test set is used both to stop the training and to evaluate the predictive performance of the network. To avoid that, a nested cross-validation procedure should be employed as showed in Figure 1.7. In 5-fold nested cross validation, the data is split into five subsets and in each round one subset used as evaluation set is completely taken out from the training data. On the remaining data set, 4-fold cross-validation is then performed as presented previously. When the optimal parameters have been found by the 4-fold cross validation, the evaluation set is used to get predictions using an average of the four networks from the cross-validation. The procedure is repeated five times, each turn changing the evaluation set.

1.2.4 Performance measures

In order to evaluate the predictive power of a prediction method, different performance measures can be used, depending on whether the outcome of a prediction method is quantitative (e.g. giving real binding affinity values) or qualitative (e.g. classifying binders and non-binders). The two measures mainly used in the work presented here

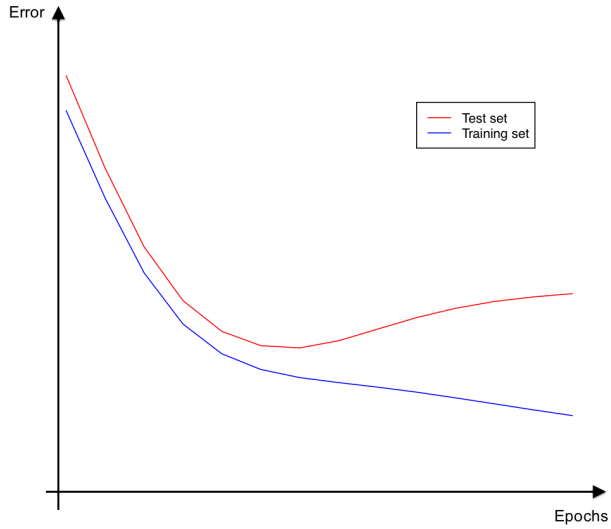

Figure 1.6. Training of a neural network with overfitting. Training set and test set error is shown as a function of training cycles (epochs). After reaching the minimum, the test set error increases while the training set error steadily decreases.

are Pearson's correlation coefficient (PCC) [24] and area under receiver operating characteristic curve (AUC) [25].

Pearson's correlation coefficient

Pearson's correlation coefficient is a widely used measure for evaluating quantitative prediction methods. It is a linear correlation coefficient calculated as

$$PCC = \frac{\sum_i (a_i - \overline{a})(p_i - \overline{p})}{\sqrt{\sum_i (a_i - \overline{a})^2} \sqrt{(\sum_i (p_i - \overline{p})^2}} \tag{1.4}$$

where a is a measured value, p is a corresponding predicted value, and $\overline{a}$ and $\overline{p}$ denote the mean values of a and p, respectively. The values of PCC vary between -1 and 1, with 1 indicating a perfect positive correlation, -1 showing a complete negative correlation, and a value of 0 corresponding to a random prediction.

Receiver operating characteristic curve and AUC

For a prediction method that classifies the output into positives and negatives, one can make a contingency table [25] which in machine

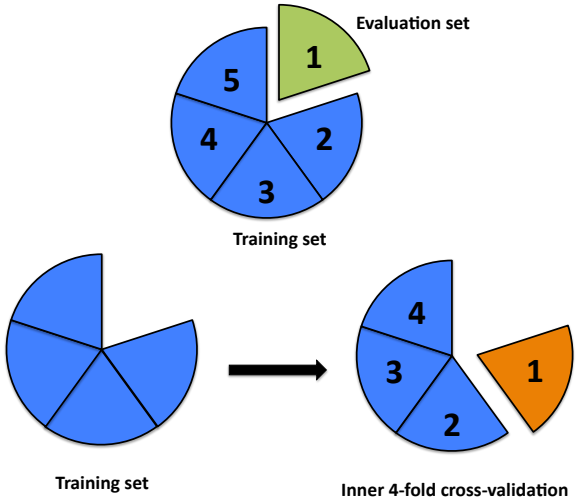

Figure 1.7. Nested cross-validation. In each round, 1/5 of the data is used as evaluation set (*green*) and is not included in the training process. The rest of the data (*blue*) is used for 4-fold inner cross-validation to get optimal parameters of the network. The four optimal networks from each inner cross-validation round are averaged and used to predict the outcome of the evaluation set.

learning is often called a confusion matrix (Figure 1.8). Based on the agreement between an output produced by a prediction method and a target value, the results are divided into four groups: true positives (TP) (actual positives predicted as positives), true negatives (TN) (actual negatives predicted as negatives), false positives (FP) (actual negatives predicted as positives), and false negatives (FN) (actual positives predicted as negatives).

From a confusion matrix, the sensitivity of a method can be calculated as the fraction of actual positives (AP) that are correctly predicted (TP): Sensitivity $= \frac{TP}{AP}$. Another useful measure is specificity which is calculated as the fraction of actual negatives (AN) that are correctly predicted (TN): Specificity $= \frac{TN}{AN}$.

For quantitative data, such as peptide–MHC binding affinities, sensitivity and specificity can be used to obtain a so-called receiver operating characteristic (ROC) curve. Classifying the data into binders and non-binders, the curve is made by plotting for different prediction threshold values the sensitivity against (1− specificity). For the actual values a threshold has to stay fixed dividing all the data into binders and non-binders. In the case of MHC binding, peptides with

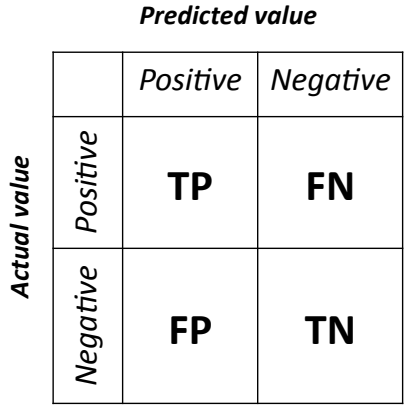

Figure 1.8. Confusion matrix for classification problems. The output of the predictor is compared to the target values, thus dividing prediction results into four groups: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

an affinity stronger than 500 nM are conventionally considered bind MHC molecules [26]. As stated above, AUC refers to the area under the ROC curve and is used as a measure of the predictive performance of a method. An AUC value close to 1 indicates a perfect predictive performance and a high positive correlation between actual and predicted values. A value of 0 corresponds to a negative correlation, and an AUC of 0.5 corresponds to a random prediction.

1.2.5 Evaluation strategies

The performance of peptide–MHC prediction methods is normally evaluated by calculating performance measures for each query MHC molecule. For pan-specific methods, in order to evaluate how well a method performs on the molecules that are not part of the training set, a leave-one-out approach (LOO) is used. In this case, all the binding data associated with a molecule in question is excluded from the training set which is used to train the method. After the training process, the predictive performance of the method for the query allele is evaluated using the excluded data. The query molecule is then changed to another one and the procedure is repeated as many times as there are molecules in the data set.

For the data sets used in this work, a large number of peptides have been measured for their binding to multiple MHC molecules.

Commonly these peptides share very similar binding affinities to several alleles, as presented in the example in Table 1.2 for the peptide AAAATCALV. Several MHC molecules having similar pseudo sequences and close binding affinities to the same peptide, provide an advantage for an artificial neural network to learn their binding specificities. Due to this characteristic, the LOO evaluation approach must in such cases be adjusted in order to ensure unbiased evaluation of the predictive performance of the method. This is achieved by not only removing the binding data for the allele in question, but also removing common peptides between evaluation and training sets. However, using this strict LOO setup might lead to a significant reduction of the training set, leaving the method with too few data points to learn general features. In order to avoid this, only 1/3 of the common peptides are removed from the training set at once. Such a training and evaluation procedure is schematically depicted in Figure 1.9. Firstly, an evaluation set composed of data for one molecule is split into three subsets: E1, E2, E3. Then, we remove peptides common between a training set and each of the evaluation subsets, resulting in three reduced training sets (T1, T2, T3). We note that here, the data for a query molecule is already removed from the initial training set. Each reduced training set is used to train artificial neural networks in a 5-fold cross-validation manner. As a result, we obtain three trained ANNs: ANN1, ANN2 and ANN3. Each network is then used to predict the output of its corresponding evaluation subset. Finally, all the predictions are combined to assess the predictive performance (in terms of PCC or AUC) of the method for the query molecule.

Table 1.2. Binding affinities of peptide AAAATCALV measured to several MHC class I molecules. Binding affinities are log-transformed, as explained in section 1.2.2.

Molecule name	Pseudo sequence	Log-affinity
HLA-A*0202	YFAMYGEKVAHTHVDTLYLRYHYYTWAVWAYTWY	0.7519
HLA-A*0203	YFAMYGEKVAHTHVDTLYVRYHYYTWAEWAYTWY	0.7964
HLA-A*0201	YFAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY	0.6568
HLA-A*0206	YYAMYGEKVAHTHVDTLYVRYHYYTWAVLAYTWY	0.7667

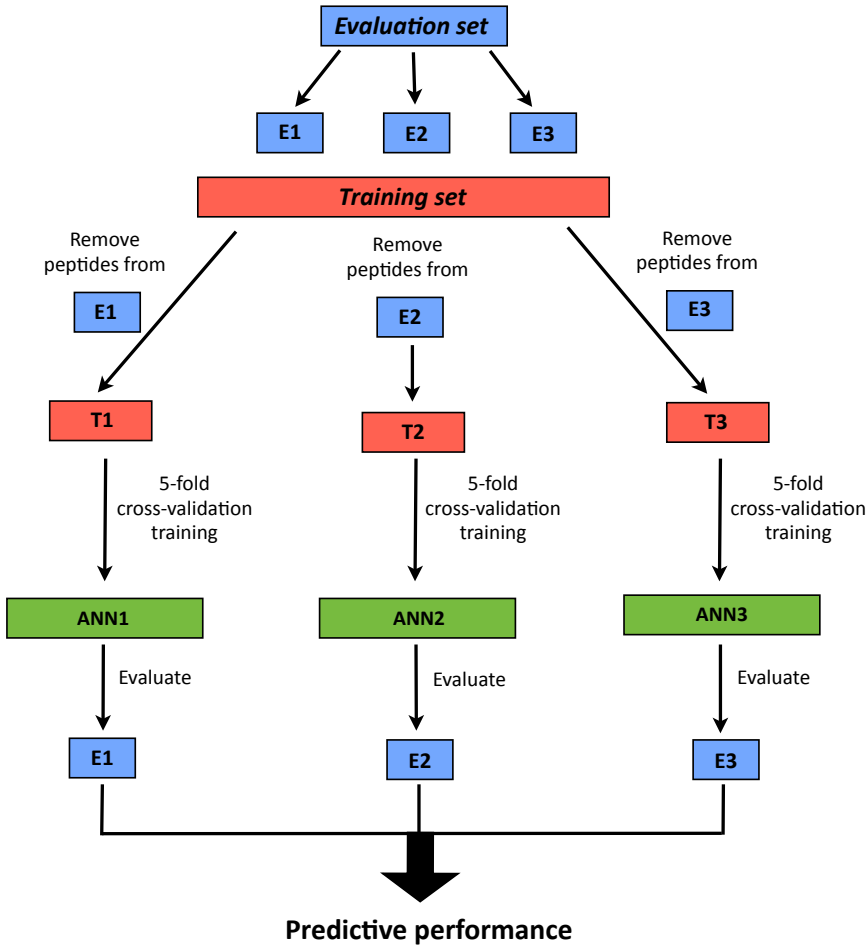

Figure 1.9. Schematic representation of the LOO training and evaluation strategy used in this work. Evaluation set corresponds to the data of a query molecule (*blue*) which is already excluded from the training set (*red*). Three reduced training sets (T1, T2, T3) are obtained by removing common peptides between the training set and three corresponding evaluation subsets: E1, E2, E3. Using 5-fold cross-validation, each reduced training set produces an ANN with optimized parameters: ANN1, ANN2 and ANN3. Each network is evaluated using the corresponding evaluation subset, and the predictive performance is obtained by combining all predictions.

Chapter 2

NetMHCcons: a consensus method for MHC class I predictions

The plethora of MHC class I prediction methods of a high prediction accuracy developed to date introduces difficulties for immunologists to choose the most suitable predictor for their research goals. Several state-of-the-art methods have been developed in our group within the recent years, and all of them differ in their accuracy depending on the conditions defining the query MHC molecule. This chapter addresses the issue of confusion among the large number of prediction tools available for MHC class I and presents a consensus method *NetMHCcons* combining three state-of-the-art methods developed by our group.

2.1 Allele-specific and pan-specific methods

As presented in section 1.2.2, the main division between MHC prediction tools is based on whether the method is able to extrapolate from MHC molecules with defined binding specificities to whose with uncharacterized binding. As indicated by the name, allele-specific methods can only give predictions to MHC molecules (also called alleles in this chapter) which are part of the training data sets. *NetMHC* is a state-of-the-art allele-specific method developed in our group and is included in the analysis presented in this chapter [27, 20]. The method was benchmarked to be one of the best predictors in several independent studies [28, 29]. The *NetMHC* method is based on ANNs and has the power to accurately predict binding for peptides within the range of 8–11 amino acids, despite the fact that most of the training data include measurements for 9-mer peptides [27].

The main properties defining MHC molecules for the allele-specific methods, which influence their prediction accuracy, are number of peptides and number of binders [30, 31, 32]. Here, the number of peptides refers to the amount of peptide–MHC measurements available in the training set for a particular MHC molecule, and the number of binders quantifies actual binding peptides with an affinity stronger than 500 nM.

A second group of MHC predictors consists of methods that are able to learn MHC binding specificities and generalize to molecules with very few or no binding measurements available. These are so-called pan-specific methods that include the MHC molecule in a form of pseudo sequence into the training process as presented in section 1.2.2. In our study, we included the *NetMHCpan* method which was demonstrated to rank at the top among pan-specific predictors [32].

One more method involved in the consensus predictor is the *PickPocket* method, which differs from the other two predictors by the underlying algorithm. The *PickPocket* approach is based on the fact that the peptide binding groove of MHC molecules has several pockets that always interact with a particular part of the peptide. Therefore, by comparing similarities of the pockets from MHC molecules with known specificity and query molecules, it is possible to predict new binding peptides for novel MHC molecules [31]. The *PickPocket* method, when compared to ANN-based predictors, was shown to give high accuracy predictions for molecules with low similarity to the data available for training, allowing the extended use of the method to non-human species [31].

For pan-specific methods, prediction accuracy on novel MHC molecules correlates with their sequence similarity to the molecules from the training set. Here, the two molecules are compared by using their pseudo sequences and calculating a pseudo-distance, as described by Nielsen et al. [33].

2.2 Paper I

The following paper was published in the journal *Immunogenetics* in October 2011. Supplementary material for this paper is given in Appendix A.

NetMHCcons: a consensus method for the major histocompatibility complex class I predictions

Edita Karosiene, Claus Lundegaard, Ole Lund, and Morten Nielsen

Center for Biological Sequence Analysis, Department of Systems Biology,
Technical University of Denmark, DK-2800 Lyngby, Denmark

Abstract

A key role in cell-mediated immunity is dedicated to the major histocompatibility complex (MHC) molecules that bind peptides for presentation on the cell surface. Several *in silico* methods capable of predicting peptide binding to MHC class I have been developed. The accuracy of these methods depends on the data available characterizing the binding specificity of the MHC molecules. It has, moreover, been demonstrated that consensus methods defined as combinations of two or more different methods led to improved prediction accuracy. This plethora of methods makes it very difficult for the non-expert user to choose the most suitable method for predicting binding to a given MHC molecule. In this study, we have therefore made an in-depth analysis of combinations of three state-of-the-art MHC–peptide binding prediction methods (*NetMHC*, *NetMHCpan* and *PickPocket*). We demonstrate that a simple combination of *NetMHC* and *NetMHCpan* gives the highest performance when the allele in question is included in the training and is characterized by at least 50 data points with at least ten binders. Otherwise, *NetMHCpan* is the best predictor. When an allele has not been characterized, the performance depends on the distance to the training data. *NetMHCpan* has the highest performance when close neighbours are present in the training set, while the combination of *NetMHCpan* and *PickPocket* outperforms either of the two methods for alleles with more remote neighbours. The final method, *NetMHCcons*, is publicly available at www.cbs.dtu.dk/services/NetMHCcons, and allows the user in an automatic manner to obtain the most accurate predictions for any given MHC molecule.

Keywords MHC class I · T cell epitope · MHC binding specificity · Peptide–MHC binding · Consensus methods · Artificial neural network

2.2.1 Introduction

Major histocompatibility complex (MHC) molecules play a key role in cell-meditated immunity binding antigenic peptides and presenting them for recognition by the immune system on the cell surface. Through antigen processing, proteins produced within a cell are degraded into short peptides, usually of 8–11 residues in length that may then be loaded on MHC-I molecules and presented on the cell surface. In this way, cytotoxic T lymphocytes are capable of recognizing the infected cells and triggering an immune response. Thousands of different allelic versions of MHC molecules exist [34], making complete experimental characterization of peptide–MHC interactions highly cost-intensive. A number of *in silico* prediction methods for peptide–MHC binding have therefore been successfully developed during the last decade (for a review, see, e.g., [35]). It has been demonstrated that the predictive performance of MHC peptide binding prediction methods depends strongly on both the number of peptides and the number of actual binders available for training [30, 31, 32]. For pan-specific methods, the performance has moreover been demonstrated to dependent strongly on the amino acid sequence distance to the nearest allelic neighbour in the data used to train the method [36, 31]. Moreover have several benchmark studies shown that consensus methods defined as a simple average of two or more different methods can lead to improved prediction accuracy [37, 38, 39, 31, 32]. This means that one method or sets of methods may perform well for one given MHC molecule while performing poorly for others. Even though several benchmarks have been carried out to compare MHC binding methods and rank them based on their prediction accuracy [29, 28, 31, 32], it remains a highly non-trivial task for the end-user to select the best suitable method for a given MHC molecule.

The objective of this study was to address this problem and define a method that for any given MHC molecule in an automatic manner defines an optimal combination of a series of prediction methods, allowing the non-expert end-user to obtain accurate binding predictions. Three state-of-the-art methods *NetMHC*, *NetMHCpan* and *PickPocket* were included in this study. *NetMHC* is an artificial neural network-based (ANN) allele-specific method, capable of predicting binding only to the molecules on which it has been trained [27, 20]. The two other methods are pan-specific meaning that they are able to predict peptide binding also to MHC molecules for which limited or no experimental peptide binding data is available. *NetMHCpan*, is ANN-based [36, 33], and the *PickPocket* method is matrix-based and

relies on receptor-pocket similarities between MHC molecules [31]. The choice of methods to be analyzed was made based on previous benchmark studies. *NetMHC* and *NetMHCpan* methods have in several large-scale benchmark studies been demonstrated to be among the best publically available predictors [35, 28, 32]. Even though the *PickPocket* method has not in any benchmark studies been shown to provide a superior performance, it has been demonstrated that the method for alleles with no close neighbours can improve binding affinity predictions when combined with *NetMHCpan* [31]. All the methods were benchmarked using a large and diverse set of quantitative peptide binding affinity measurements, covering more than 100 MHC class I alleles.

It is apparent that not all methods can be applied to predict binding to any chosen MHC molecule. For example, the *NetMHC* method is available only if the allele in question is also part of the training set used to develop the method. On the other hand, the pan-specific *NetMHCpan* and *PickPocket* methods are capable of predicting binding to any MHC molecule with known protein sequence. The development of the consensus method was guided by simplicity and robustness. This means that the combination of two or more methods was only included into definition of the final method if it demonstrated a significantly improved performance compared to the individual methods within the analysed conditions. In the paper, we first benchmark each method individually and evaluate their performance under different settings. Next, given these results, the consensus method is defined in an allele-specific manner as a combination of one or more prediction methods, and finally is the consensus method, *NetMHCcons*, validated against an independent data set.

2.2.2 Materials and Methods

Data set

The benchmark data set consists of quantitative nonameric peptide–MHC class I binding data with a submission date prior to September 2009 retrieved from the IEDB [40] and an in-house MHC–peptide binding database. In total, it consists of 101,728 unique peptide–MHC class I interactions covering 101 alleles: 34 HLA-A, 35 HLA-B, one HLA-C, one HLA-E, 11 chimpanzee (Patr), 12 rhesus macaque (Mamu), one gorilla (Gogo), and six mouse alleles. Table A.1 contains a detailed description of the benchmark data set. All peptide binding measurements were obtained as IC50/EC50 values

and for this study were log-transformed to fall in the range between 0 and 1 using the relation 1–log(IC50nM)/log(50,000) [20].

Analyzed methods and conditions

For our analysis, we used in-house versions of *NetMHC*, *NetMHCpan* and *PickPocket* trained and evaluated on the MHC class I benchmark data set. Having both allele-specific (*NetMHC*) and pan-specific (*NetMHCpan* and *PickPocket*) methods in the benchmark resulted into two analyzed conditions: (1) when allele in question is part of the training set; (2) when allele in question is not part of the training set.

When an allele for which the binding should be predicted was not part of the training data, the analysis reduced to include only the two pan-specific methods. In all other cases, the analysis included all three methods. In order to obtain a reliable performance when evaluating the methods, we constructed a reduced data set consisting of 78 alleles, for which at least 50 data points were available and at least ten of them were binding peptides (i.e., having an affinity stronger than 500 nM). This reduced data set is presented in Table A.2.

Evaluation strategies

When training ANNs, it is critical to define a strategy to avoid overfitting. Conventionally, this is done using a test set to stop the network training when the performance on the test set is optimal. This is a highly CPU-intensive procedure since the evaluation must be made using nested cross-validation. For 5-fold nested cross-validation for instance, the data is split into five subsets. In each round, one subset is employed as evaluation set and is not included into training process. The remaining four subsets are used in the inner cross-validation loop where four networks are constructed each using three sets to train the network and one set used to stop the training to avoid overfitting. The binding predictions of the peptide in the evaluation set are next calculated as a simple average of the four networks in each cross-validation ensemble. Another faster strategy for cross-validation is to use the test set as evaluation data. In this setting, the test set is used both to stop the network training in order to avoid overfitting, and to evaluate the predictive performance. This cross-validation approach has an inherent potential of overestimating the predictive performance.

To evaluate to what degree the use of the faster training strategy led to an overestimation of the predictive performance, the two different evaluation strategies were compared for the *NetMHC* and

NetMHCpan methods in terms of Pearson's correlation coefficients (PCC). As the *PickPocket* method has no stopping procedure, this method was not included in this comparison.

Our analysis showed that the difference between evaluation on independent data sets compared to evaluation on test sets during cross-validation is significant for the *NetMHC* method when the data set is smaller than 1,000 data points (p=0.02), and not significant for the sets with 1,000 or more data points (p=0.15). For the *NetMHCpan* method, which always has a large evaluation set, no difference in performance was observed between the two evaluation strategies (p=0.19) (see Figure A.1). As a result of this and in order to reduce the computational efforts, we have for the further analysis chosen to use independent set for *NetMHC* evaluations and make the *NetMHCpan* evaluations on the test sets during cross-validation, which is a much faster approach.

In order to evaluate the predictive performance of *NetMHCpan* and *PickPocket* for alleles, which are not part of the training data, a leave-one-out (LOO) approach was used, meaning that the data for the allele in question was excluded from the training set. One important characteristics of the benchmark data set is that many peptides have been tested for binding to multiple MHC molecules. Given this nature of the peptide data set, it is essential to design the LOO training strategy so that not only data for the specific allele in question is removed from the training, but also peptides common between the evaluation and training sets. In doing this, we assure that neither the MHC molecule nor the peptides are present in the training and evaluation sets at the same time. In order to avoid reducing the training set too much in this strict LOO setup, the evaluation set was split into three subsets and the performance for each subset was evaluated. Setting the number of evaluation subset partitions to three was based on a compromise between increase in calculation time and the accuracy of performance estimation. A small number of subset partitions would be computationally fast but would lead to relative large reductions in the size of the training data, and likewise would a large number of subset partitions be computationally costly but lead to only a minor reduction in the size of the training data. A small evaluation of the performance as a function of the number of evaluation subset was carried out for the HLA-A*0201 allele. This is the allele in the data set characterized by the largest number of peptide measurements, and hence is the example where the peptide overlap to other molecules should be the highest. This evaluation demonstrated that only limited gain

in performance was achieved for subset divisions larger than three (data not shown), therefore leading us to use three subset through the benchmark evaluation.

Calculation of pseudo-distance to the nearest neighbour

Pseudo-distance between two alleles was calculated from the pseudo sequences of MHC molecules as described in [33]. The nearest neighbour for a specific allele is defined as the molecule from the neighbour reference which includes MHC molecules with more than 50 data points and more than ten binders [36], with the smallest pseudo-distance to this allele.

Defining the consensus method

A consensus method is defined in terms of combinations of two or more different individual methods. Here, we use a simple average of the raw log-transformed prediction scores from each method to define the consensus method. Combined methods are represented using a plus sign "+" in this study. For each allele, the performance of each prediction method and their possible combinations were given as PCC between the log-transformed predicted and measured binding affinities.

Validation of the consensus method

An independent evaluation set consisting of data from the IEDB [40] and an in-house MHC–peptide binding database with a submission date after September 2009 was constructed. This validation data set had no overlap with the training set. In this way, we ensured that the final consensus method was not trained and evaluated using the same data points. In order to obtain reliable evaluations, the only alleles characterized by at least ten data points and at least two binders were included. The validation data set included 14,923 peptide–MHC binding data and covered 62 alleles (see Table A.3). Part of these alleles (46) were included in the training data set, hence allowing a validation of the consensus method in two conditions: (1) for the alleles in question being part of the training data and (2) for the novel alleles not described in the training data.

Statistical analysis

In this study, the evaluation of significance of the observed differences between the results was performed using one-tailed paired *t*-test with a significance level of 0.05. If a very low p value was obtained during analysis, it is then stated as "$p<0.0001$" and not by exact value.

2.2.3 Results

The objective of this study was to define a strategy that for any given MHC molecule defines an optimal combination of a series of prediction methods, allowing the non-expert end-user in an automated manner to obtain accurate binding predictions for any given MHC molecule. The "Results" section falls into three subsections. First, we illustrate the end-user problem of identifying which method to use for binding prediction for a given MHC molecule, next we analyse in a large-scale benchmark how a simple yet powerful setting can be defined leading to a consensus method that consistently outperforms all single methods included in the benchmark, and finally the consensus method is validated on an independent data set of MHC peptide binding measurements not included in the method development.

Performance variations of different methods

The motivation to perform this study was based on earlier observations that different methods give different prediction results in different conditions. To illustrate this, we compared how two ANN-based methods (*NetMHC* and *NetMHCpan*) that were trained on identical data would handle a given prediction task. A protein sequence was submitted to the *NetMHC* and *NetMHCpan* methods, trained on the benchmark data set, and the methods were asked to predict binding to the HLA-B*3801 molecule, which was defined by 136 peptide–MHC binding measurements of which only three were binders within the training data set. In the left panel of Figure 2.1, the output of this analysis is represented as a scatter plot between the prediction values of *NetMHC* and *NetMHCpan*. It is apparent from the figure that the correlation between the prediction scores obtained by the two methods is low − the PCC is 0.569 − and that the difference is in particular large in the high binding tail of the two methods. In order to investigate the disagreement between these two methods in a more systematic manner, we obtained the predictions of both methods for all

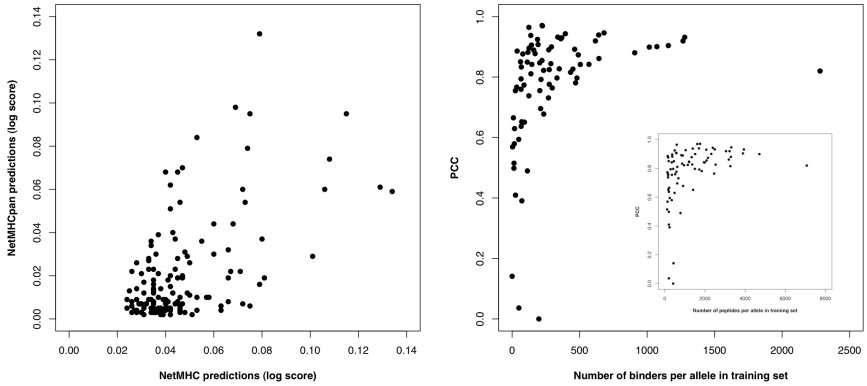

Figure 2.1. Overview of agreement and disagreement between *NetMHC* and *NetMHCpan* methods. The *left panel* shows binding predictions to HLA-B*3801 allele by *NetMHC* and *NetMHCpan* methods for peptides from the same chosen protein. Log-transformed prediction scores by each method are plotted. The right panel demonstrates the dependency of the Pearson's correlation coefficient between the two methods as a function of number of peptides (the inner plot) and number of binders available per allele in the training set.

MHC molecules included in the training data. The right panel of Figure 2.1 demonstrates how the correlation between the predictions by *NetMHC* and *NetMHCpan* methods depends on the size of the training set. For MHC molecules that are defined by few data points and have small number of actual binders, the correlation coefficient between *NetMHC* and *NetMHCpan* methods is very small. The difference between predictions of the two methods is diminished when the number of peptides and binders in the training set is increased.

Defining the consensus method

Allele in question is part of the training data

When an allele is part of the training data, all three methods and their combinations can be used to define the consensus method. Each method was evaluated using cross-validation on the benchmark data set. Figure 2.2 shows accumulative number of instances where each method achieved the highest predictive performance as a function of the number of data points and the number of binding peptides, respectively, characterizing the different MHC molecules. The figure clearly demonstrates that some methods achieve the highest

performance more often than others and, thus, are more important for defining the optimal consensus method. Only one combination, *NetMHC + NetMHCpan*, consistently improved prediction accuracy. This combination has an increased accuracy for alleles characterized by a larger number of peptides and significantly outperforms both the *NetMHCpan* and *NetMHC* methods (p=0.005) for the set of alleles characterized by at least 500 data points. All in all, the *NetMHC + NetMHCpan* combination gives a superior prediction performance for most of the alleles from the benchmark data set. As can be seen in Figure 2.2, *NetMHC + NetMHCpan* has the highest performance 60 times out of 92 (65.2%), excluding ties. The second best method is *NetMHCpan*, which achieves the highest prediction accuracy for the alleles with a little training data set and all in all gives the best scores for 18 alleles (19.6%). A similar tendency is observed when comparing the accumulative number of times any given method is winning as a function of the number of binding peptides within the training set (Figure 2.2, right panel). It is striking to observe that both *NetMHC* and *PickPocket* rarely perform best as single methods. Only when combined with *NetMHCpan* does *NetMHC* contribute to the overall performance and adding *PickPocket* seems to have a direct negative effect on the prediction accuracy.

The results displayed in Figure 2.2 thus suggest *NetMHCpan* as the optimal method for alleles characterized by few peptide measurements, and a consensus method defined by number of peptides (N_p)

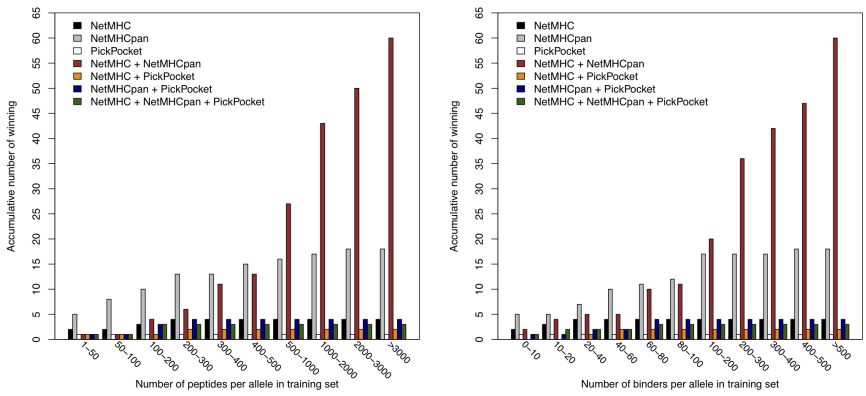

Figure 2.2. Accumulative number of winning for each method included in the analysis depending on the number of peptides (*left*) and number of binders (*right*) per allele in training set.

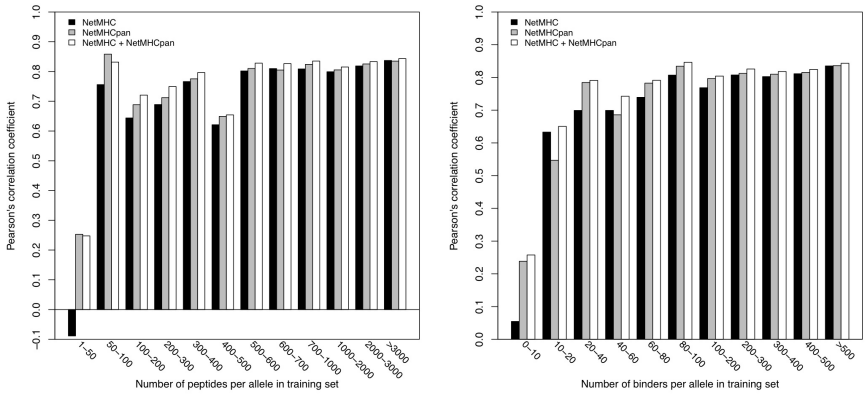

Figure 2.3. The average predictive performance of the alleles in the benchmark data set as a function of number of peptides (*left*) and number of binders (*right*) per allele in the training data.

and number of binders (N_b), respectively, per allele in training set. The result of this analysis is given in Figure 2.3. The analysis demonstrates that for alleles characterized by a small number of data points (N_p<50), the allele-specific *NetMHC* method performs poorly. In this case, the pan-specific *NetMHCpan* method clearly achieves the highest performance and significantly outperforms *NetMHC* (p=0.02). Considering the performance dependency on the number of binders per allele, one can notice that combination of the two methods always outperforms its components. The prediction accuracy of the *NetMHC* method for MHC alleles characterized by few binders (N_b<10) is, however, very low, and the difference between *NetMHCpan* and the consensus method defined as *NetMHC* + *NetMHCpan* is for these alleles statistically insignificant. *NetMHCpan* also achieves the highest performance within the next bin (50≤N_p<100). However, we cannot access the significance of the difference between the different methods in this case as we have only three alleles within the bin. Based on these observations and having in mind that we choose a single method where it is not significantly different from the combined approach, we defined the consensus method for the condition of the allele in question being part of the training data as follows:

$$NetMHCcons = \begin{cases} NetMHCpan & \text{for } N_p < 50 \text{ and } N_b < 10 \\ NetMHC + NetMHCpan & \text{otherwise} \end{cases} \quad (2.1)$$

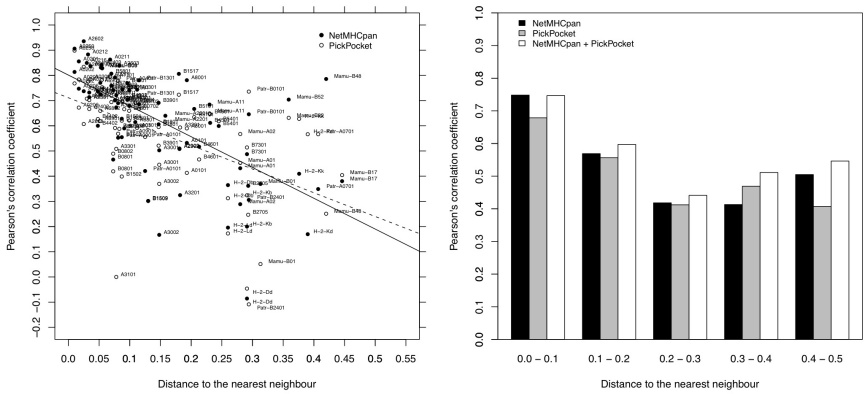

Figure 2.4. Predictive performance of the alleles from the benchmark data set as a function of distance to the nearest neighbour. The *left panel* shows performance for each allele of the *NetMHCpan* and *PickPocket* methods. The *solid line* represents the least square fit for the *NetMHCpan* data, and the *dotted line* gives the least square fit for the *PickPocket* data. A full size of this graph is available in Figure A.2. The *right panel* demonstrates the average performance dependency on the distance to the nearest neighbour. The performance for each allele was calculated using leave-one-out approach as described in "Materials and methods". Distance to the nearest neighbour was calculated using MHC pseudo sequences as described by [33].

Detailed results of the analysis of methods when allele in question is part of the training data are given in Table A.4.

Allele in question is not part of the training data

When the MHC allele for which we wish to predict peptide binding is not part of the training data set, only the pan-specific *NetMHCpan* and *PickPocket* methods can be employed. It has been shown earlier that the predictive performance of pan-specific methods depends on the allelic environment. For example, *NetMHCpan* was demonstrated to perform well for the alleles with well-characterized neighbourhood [36], and *PickPocket* was shown to give a good prediction accuracy for MHC molecules for which the similarity to characterized alleles was low [31]. To investigate the performance of *NetMHCpan*, *PickPocket* and their combination, we conducted an LOO evaluation on the benchmark data set as described in "Materials and methods". The results are illustrated in Figure 2.4 as the performance dependency on the distance to the nearest neighbour as measured in terms of the

MHC pseudo sequence similarity (detailed results of this analysis are presented in Table A.5). The left panel of the figure demonstrates that a large fraction of the alleles from our benchmark data set have close nearest neighbours. Most of these alleles are human HLA-A and HLA-B alleles, whereas chimpanzee (Patr), macaque (Mamu) and mouse alleles tend to have more distant neighbours. It is apparent that the performance of both methods depends strongly on the distance from MHC molecule in question to the nearest molecule in the training set. Regression analysis for each method demonstrated, that the performance is decreased significantly with increasing distance for both methods ($p<0.0001$).

The right panel of the figure gives the average predictive performance of the different methods as a function of the distance to the nearest neighbour within the training data. The figure demonstrates the high performance of *NetMHCpan* in prediction of binding to MHC molecules with close neighbours. This method gives the highest PCC values of all methods when the distance (D) is lower than 0.1 and achieves the highest performance for 23 out of 40 MHC molecules, while *NetMHCpan + PickPocket* wins only 15 times within this bin. The difference between *NetMHCpan* and *NetMHCpan + PickPocket* was not statistically significant. If the distance to the nearest neighbour is larger than 0.1, the combination of the *NetMHCpan* and *PickPocket* methods significantly outperforms both *NetMHCpan* (p=0.019) and *PickPocket* (p=0.003) methods. Based on these observations and our decision to use the simpler method where the significant difference is not observed, we define the optimal method to predict peptide binding to MHC molecules not included in the training set as follows:

$$NetMHCcons = \begin{cases} NetMHCpan & \text{for } D < 0.1 \\ NetMHCpan + PickPocket & \text{for } D \geq 0.1 \end{cases} \tag{2.2}$$

Based on the results obtained above, we can now define the final consensus method. We define a reference set of alleles that are characterized by at least 50 data points and at least ten binders. Based on this reference set, the *NetMHCcons* method can be defined as

$$NetMHCcons = \begin{cases} NetMHC + NetMHCpan & \text{for } D = 0 \\ NetMHCpan & \text{for } 0 < D < 0.1 \\ NetMHCpan + PickPocket & \text{for } D \geq 0.1 \end{cases} \tag{2.3}$$

where D refers to the distance between the query allele and its nearest neighbour in the reference allele set. Note that having the distance equal to 0 (D=0) means that the alleles in question is part of the training set.

Validation of the final consensus method

The consensus method for peptide binding to MHC was next benchmarked on an independent evaluation data set (see "Materials and methods"). In order to compare the results with the methods composing *NetMHCcons*, we obtained predictions of each method separately and compared the results for the subsets of alleles depending on how each method was involved in the final consensus method. This resulted into three different comparisons of the average PCC values: (1) for the alleles that were part of the training set, the results of *NetMHCcons* were compared with the results obtained by *NetMHCpan* and *NetMHC* methods (41 allele); (2) *NetMHCcons* was compared with *NetMHCpan* for all the alleles from the validation set (62 alleles); (3) the comparison of the consensus method with *NetMHCpan* and *PickPocket* was done using the alleles that were not included in the training data set and had a distance of 0.1 or larger to the training reference set (17 alleles).

A summary of the validation results is given in Figure 2.5 (details are given in Table A.6). The performance of *NetMHCcons* on the alleles that were part of the training set was found to be significantly higher than both *NetMHC* ($p<0.0001$) and *NetMHCpan* ($p=0.01$). Comparing *NetMHCcons* and *NetMHCpan* performances using all the alleles, significant difference between performance values was not observed, but the consensus method has the highest performance of the two. The set of alleles that have a distance of 0.1 or more to the training data compose too small set to obtain significant p values; however, the average values show that the consensus method also here has a higher performance than both the *NetMHCpan* and *PickPocket* methods.

The *NetMHCcons* method is implemented as a web server and is available online at: http://www.cbs.dtu.dk/services/NetMHCcons. The method provides affinity predictions for any peptide of length 8–11 amino acids to any given MHC class I molecule of known protein sequence. Two submission types are handled − a list of peptides or a protein in FASTA format. The server provides a possibility for the user to choose MHC molecule in question from a list of alleles or alternatively upload the MHC protein sequence of interest. The method is also implemented as SOAP based Web Service available at: http://www.cbs.dtu.dk/ws/NetMHCcons/.

2.2.4 Discussion

In this study, we performed a detailed analysis of several state-of-the-art methods with a purpose of developing a consensus

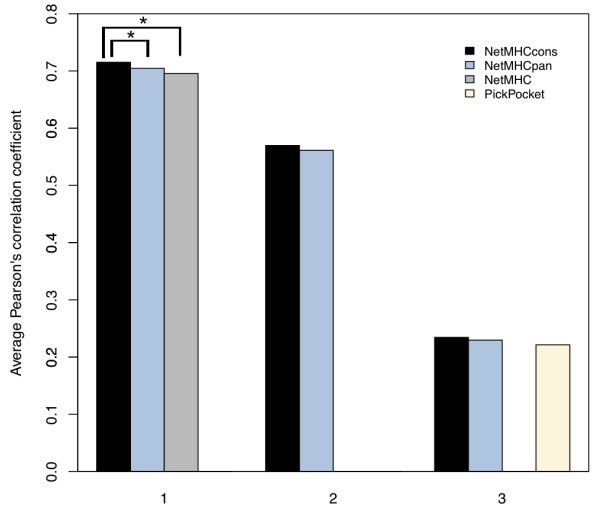

Figure 2.5. Validation results of the *NetMHCcons* method. The plot shows three groups of comparisons, from the left: **(1)** *NetMHCcons*, *NetMHCpan* and *NetMHC* for the alleles common between training and validation sets; **(2)** *NetMHCcons* and *NetMHCpan* for all alleles in the validation set; **(3)** *NetMHCcons*, *NetMHCpan* and *PickPocket* for alleles not included in the training data and having $D \geq 0.1$. Significant difference between any two methods is indicated by stars and was calculated using paired one-tailed *t*-test.

method that consistently provides the most accurate predictions for any given MHC molecule. To the best of our knowledge, this study analyzing and combining several different methods in an allele-specific manner is the first of its kind. Having involved allele-specific (*NetMHC*) and pan-specific (*NetMHCpan* and *PickPocket*) methods, two different conditions were analyzed in our study. First of all, if the given MHC allele had earlier been characterized, then all three methods and their combinations were analyzed. Here, we found that the prediction accuracy of the allele-specific *NetMHC* method depended strongly on the number of data available characterizing the given allele and demonstrated that for MHC molecules that are poorly characterized, the *NetMHCpan* method is the best predictor. On the other hand, increasing number of data points and binders available for the MHC molecule in question, the *NetMHC* method becomes important and the combination of this method with *NetMHCpan*

provides the most accurate predictions. These conclusions are in agreement with an earlier report [32].

The vast majority of MHC molecules remain uncharacterized in terms of their binding specificity. For this reason, several pan-specific methods have been developed [41, 42, 33, 31]. Moreover, several publications have demonstrated the importance of describing the subtle differences in binding specificity between MHC molecules in order to understand cellular immune responses of a given host to an infection [43, 44, 45, 46]. In our analysis, we considered two of the pan-specific methods *NetMHCpan* and *PickPocket*, both being able to produce high accuracy predictions for MHC molecules with limited or no binding data available. These methods were benchmarked under the conditions when the allele in question was not part of the training data set employing a LOO approach. We demonstrated that the performance of both methods reduces with increased distance to the nearest MHC molecule with characterized binding specificity. This is in agreement with previous studies [31]. In our study, we additionally investigated how the performance of both methods and their combination depended on the distance to the closest characterized MHC molecule. At small distances, *NetMHCpan* demonstrated a superior performance, which was not maintained when the distance increased and at larger distances the contribution of the *PickPocket* method was demonstrated to be important when combined with the *NetMHCpan*. This is in accordance with the work by [31]. A consensus method defined as combination of *NetMHCpan* and *PickPocket* was hence shown to perform with the highest accuracy for MHC molecules with a large distance to MHC molecules with characterized binding specificity.

The final *NetMHCcons* method was validated using a diverse independent evaluation set. It was demonstrated that *NetMHCcons* achieved the highest performance compared with each separate method included in this analysis. This is, to our knowledge, the first consensus method defined as combination of three different methods, which involve both allele-specific and pan-specific approaches. Our analysis demonstrated how several methods could be combined into one capable of producing the most accurate predictions for any given allele. Such a method is of high importance to the non-expert user allowing in an automated manner to obtain accurate predictions of binding to any MHC class I molecule of interest and also suggests that a similar approach might be employed to improve the accuracy of MHC class II predictors.

Several other high performing methods are publicly available for MHC class I predictions including the average relative binding (ARB)

matrix method [47] and the stabilized matrix method (SMM) [48], both available as part of the IEDB tools for MHC class I binding prediction. None of these methods have been included in this study. In order to define a consensus method, a large independent evaluation data set is needed for obtaining a reliable performance estimate of the different methods and finding their optimal combination. The fact that the evaluation data must be large and independent makes it troublesome to include publicly available method in the benchmark analysis. If we define a large benchmark data, large parts of the data will most likely have been included in the training of the different methods and the evaluation will be erroneous due to overfitting. If we limit ourselves to recently published data that most likely has not been included in the training of the different method, the evaluation data set become too small to allow for a robust method evaluation. Only by retraining the methods on the large data set can we maintain a large and independent evaluation data set allowing for a robust and unbiased evaluation of the different methods included in the benchmark. Including non-in-house methods would require expert knowledge of each method and hence must be carried out as a collaborative effort between the authors of the different method and is beyond the scope of this paper. We might suggest that such an effort should be carried out in the future along the lines of previous benchmark studies [39, 28].

In conclusion, we have defined a method, *NetMHCcons*, in terms of the *NetMHCpan* method and its combinations with *NetMHC* and *PickPocket* based on conditions defining the MHC molecule in question. The method is implemented as a web server allowing the user in an automatic manner to obtain optimal predictions for any MHC class I molecule of interest.

Acknowledgements This work was supported by two NIH (National Institutes of Health) grants (contract no. HHSN272200900045C, and contract no. HHSNN26600400006C).

Chapter 3

NetMHCIIpan-3.0: a pan-specific method for MHC class II predictions

In the previous chapter we demonstrated the power of MHC class I prediction methods and how they have been expanded to be applied to non-human species as well, and combined to improve their predictive accuracy. However, in comparison the situation for MHC class II is far behind. Structural differences between MHC molecules encoded by different loci have been limiting the development of cross-loci and cross-species pan-specific methods. Additionally, the amount of binding data available for MHC class II molecules, especially HLA-DP and HLA-DQ, is very limited and covers only a small fraction of the thousands of existing molecules. Moreover, as opposed to HLA-DP and HLA-DQ, HLA-DR molecules are polymorphic only in their beta chain. For these reasons, class II pan-specific prediction methods have so far been limited to HLA-DR molecules. This chapter presents a novel pan-specific method capable of predicting peptides binding to all HLA class II molecules with a known protein sequence. The following section presents details about development of the method and gives its benchmarking results.

3.1 Paper II

The following paper was published in the journal *Immunogenetics* in July 2013 (Epub ahead of print). Supplementary material for this paper is included in Appendix B.

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

Edita Karosiene[1], Michael Rasmussen[2], Thomas Blicher[3], Ole Lund[1], Søren Buus[2], and Morten Nielsen[1,4]

[1]Center for Biological Sequence Analysis, Department of Systems Biology, Technical University of Denmark, DK-2800 Lyngby, Denmark

[2]Laboratory of Experimental Immunology, Faculty of Health Sciences, University of Copenhagen, DK-2200 Copenhagen, Denmark

[3]The Novo Nordisk Foundation Center for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen, DK-2200 Copenhagen, Denmark

[4]Instituto de Investigaciones Biotecnológicas, Universidad Nacional de San Martín, San Martín, Buenos Aires, Argentina

Abstract

Major histocompatibility complex class II (MHCII) molecules play an important role in cell-mediated immunity. They present specific peptides derived from endosomal proteins for recognition by T helper cells. The identification of peptides that bind to MHCII molecules is therefore of great importance for understanding the nature of immune responses and identifying T cell epitopes for the design of new vaccines and immunotherapies. Given the large number of MHC variants, and the costly experimental procedures needed to evaluate individual peptide–MHC interactions, computational predictions have become particularly attractive as first-line methods in epitope discovery. However, only a few so-called pan-specific prediction methods capable of predicting binding to any MHC molecule with known protein sequence are currently available, and all of them are limited to HLA-DR. Here, we present the first pan-specific method capable of predicting peptide binding to any HLA class II molecule with a defined protein sequence. The method employs a strategy common for HLA-DR, HLA-DP and HLA-DQ molecules to define the peptide-binding MHC environment in terms of a pseudo sequence. This strategy allows the inclusion of new molecules even from other species. The method was evaluated in several benchmarks and demonstrates a significant improvement over molecule-specific methods as well as the ability to predict peptide binding of previously uncharacterised MHCII molecules. To the best of our knowledge, the *NetMHCIIpan-3.0* method is the first pan-specific predictor covering all HLA class II molecules with known sequences including HLA-DR, HLA-DP, and HLA-DQ. The *NetMHCpan-3.0* method is available at http://www.cbs.dtu.dk/services/NetMHCIIpan-3.0.

Keywords MHC class II · T cell epitope · MHC binding specificity · Peptide–MHC binding · Human leukocyte antigens · Artificial neural networks

3.1.1 Introduction

Major histocompatibility complex (MHC) molecules play a key role in defining the specificity of the cellular immune system by presenting antigens to the immune system cells. In case of MHC class II molecules, these cells are T helper lymphocytes that recognize peptide–MHC complexes on the surface of antigen-presenting cells. Peptides presented by MHC class II molecules are derived from proteins taken up from the extracellular environment. Whereas a large number of peptides can be generated from pathogenic proteins, only a small part of these trigger an immune response. One of the most important events defining which peptides will trigger an immune response is binding to MHCII molecules expressed by the host [14].

The human MHC locus (in humans called HLA for human leukocyte antigens) is extremely polymorphic and encodes thousands of different HLA class II molecules. Characterising the peptide-binding specificities of all the polymorphic MHC class II molecules is a serious experimental challenge. Therefore, during the last decades, large efforts have been put into the development of *in silico* methods for predicting peptide-binding affinities to MHC class II molecules. Using thousands of peptide-binding data points, several predictors have been developed and benchmarked (for review, see [10]). One very important subset of these predictors consists of the so-called pan-specific methods that are capable of obtaining accurate predictions for molecules with limited or no binding data [49, 50, 51, 52]. For MHC class I prediction, it has been demonstrated that a pan-specific approach can benefit from being trained on cross-loci, and even cross-species, data. That is, the predictive performance for HLA-B locus molecules is improved when including HLA-A locus data in the training of the pan-specific MHC class I binding prediction method (and vice versa), and the overall performance of predictions of HLA molecules is improved when including binding data representing non-human MHC molecules [36]. Extending this approach to MHC class II is not a trivial task. Differences in sequence polymorphism and corresponding details in the molecular structures across the different MHC class II loci complicate the development of cross-loci and cross-species training strategies. This, combined with the very limited amount of data available for most MHC class II molecules, has limited the application of pan-specific methods to HLA-DR molecules. The understanding of HLA-DP and HLA-DQ binding specificities is limited to a handful of molecules which have been characterised experimentally, and beyond a few mouse H-2 molecules, to the best of

our knowledge, no general MHC class II prediction method is available for non-human primates and other non-human species.

The number of state-of-the-art pan-specific methods for MHC class II molecules available up to date is very limited. The classical MHC class II predictor, *TEPITOPE* [53], uses position-specific scoring matrices derived from experimental data. The method is, however, limited to 51 HLA-DR molecules only. In addition to this, a *TEPITOPEpan* predictor has been developed [18] by extrapolating from the binding specificities of the molecules characterised by *TEPITOPE*. The method is based on MHC pocket similarities and is capable of providing predictions for any HLA-DR molecule. The same is achieved by the *NetMHCIIpan-2.0* predictor [50], which outperforms the *TEPITOPEpan* method in terms of prediction accuracy [18]. The method is based on artificial neural networks and uses an MHC binding pocket pseudo sequence combined with the peptide sequence as an input. Like the *TEPITOPEpan* method, *NetMHCIIpan-2.0* predicts binding for all HLA-DR molecules with a known primary sequence.

In this paper, we present a novel pan-specific predictor capable of predicting binding affinities to all HLA class II molecules. The method is based on artificial neural networks and has been trained on more than 50,000 quantitative peptide-binding measurements covering HLA-DR, HLA-DP, HLA-DQ as well as two murine molecules. Using a panel of benchmark setups, we seek to investigate to what extent the pan-specific method outperforms allele-specific approaches and whether it can obtain accurate predictions even for HLA molecules, which have not been experimentally characterised. Arriving at a true pan-specific method enabling prediction of the binding specificity for all HLA-II molecules, we end the analysis by conducting the first global analysis covering all prevalent HLA-II molecules, investigating and quantifying the functional diversity of the molecules encoded at the three HLA-II loci.

3.1.2 Materials and Methods

Data sets

Training data used to develop the method consisted of quantitative MHC class II peptide-binding data retrieved from the IEDB database [40]. In total, the training data set comprises 52,062 data points covering 24 HLA-DR, five HLA-DP, six HLA- DQ and two mouse (H-2) molecules. All molecules were covered by more than 50 peptide binding data points measured as IC50/EC50 values which were

log-transformed to fall in the range between 0 and 1 using the relation $1-\log(\text{IC50nM})/\log(50{,}000)$ [20]. The evaluation set was restricted to HLA-DR molecules and contained 9,860 binding affinity measurements covering 13 molecules, four of which were not included in the training set. A summary of the data used to develop the method is presented in Table B.1, and evaluation data set details are given in Table B.2.

Mapping of MHC molecules

For constructing the *NetMHCIIpan* method, all MHC class II molecules need to be mapped to a common reference sequence. This is done by aligning alpha and beta chain sequences of all MHC molecules to the reference sequences, DRA1*0101 and DRB1*0101. For HLA-DR molecules, the mapping on a sequence level is in agreement with the mapping on the structural level. On the other hand, HLA-DP and HLA-DQ molecules demonstrate minor variations from HLA-DR in the peptide-binding domain in both the alpha and beta chains. To evaluate the structural impact of these variations, we employed the analysis described below. The analysis is based on the five available structures solved for HLA-DP and HLA-DQ molecules, which are compared to a representative high-resolution HLA-DR structure selected among the large number of structures available for HLA-DR molecules. The list of available HLA-DP and HLA-DQ structures from the Protein Data Bank (PDB) is given in Table 3.1.

Table 3.1. Structures of all HLA-DP and HLA-DQ molecules available in the PDB database.

PDB ID	Alpha chain	Beta chain
3LQZ	HLA-DPA1*0103	HLA-DPB1*0201
1UVQ	HLA-DQA1*0102	HLA-DQB1*0602
1JK8	HLA-DQA1*0302	HLA-DQB1*0302
1S9V	HLA-DQA1*0501	HLA-DQB1*0201
2NNA	HLA-DQA1*0301	HLA-DQB1*0302

An HLA-DR (PDB ID: 1A6A [16]) structure was chosen as a reference, and the HLA-DQ and HLA-DP structures were aligned to the binding domain of this reference molecule (Figure 3.1). The superimpositions were performed in PyMOL [17] and demonstrate a high

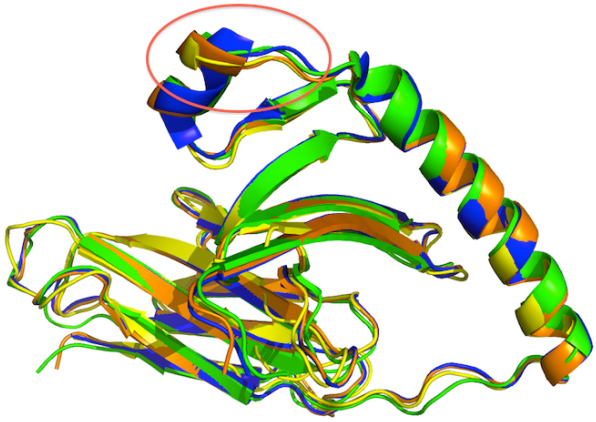

Figure 3.1. Superimposition of HLA-DR, HLA-DP and HLA-DQ alpha chains. HLA-DR alpha chain (PDB ID: 1A6A [16]) is shown in *yellow* and was used as a reference chain. HLA-DP chain (PDB ID: 3LQZ [54]) is shown in *green*, HLA-DQ chain without a gap (PDB ID: 1JK8 [55]) is shown in *orange* and HLA-DQ chain with a gap (PDB ID: 1S9V [56] is shown in *blue*. The area affected by the deletion in DQA sequence is *circled*.

degree of structural conservation among the different loci (RMSD values between 0.7 and 0.8 Å).

During the analysis, an important variation was observed for HLA-DQ molecules only and was investigated in more detail. We observed that sequences belonging to the HLA-DQA1*04, HLA-DQA1*05 and HLA-DQA1*06 serotype groups (e.g. sequences like HLA-DQA1*0401) display a single amino acid deletion, which from a pure sequence point of view, corresponds to position 53 in HLA-DRA [34]. However, this leads to a shift of the preceding residues in the DQA sequences, which now realign with DRA positions 52 and 53. Although the deletion affects the orientation of the short α-helical segment and the loop (residues 45–52) next to the P1 binding pocket (area marked in Figure 3.1), these changes have negligible impact on peptide binding, as the reorientation appears to be a localised change, and very few contacts with the peptide are observed within the area. Due to the minor impact of the area discussed above to the binding of the peptide, an automated sequence alignment approach was chosen to identify the deletion in HLA-DQ sequences. Pair-wise sequence alignments were made and visualized using *ClustalW* [57]. Each HLA-DQ sequence was aligned one by one to the reference sequence of HLA-DR. The results are presented in

Figure 3.2. Figure 3.2a shows the alignments of HLA-DQ alpha chains with the amino acid deletion, while Figure 3.2b demonstrates alignments of HLA-DQ alpha chains with no deletions. The alignments demonstrated that for all the HLA-DQ sequences that have a deletion, the deletion is consistently found in the same place (position 53 in the reference sequence).

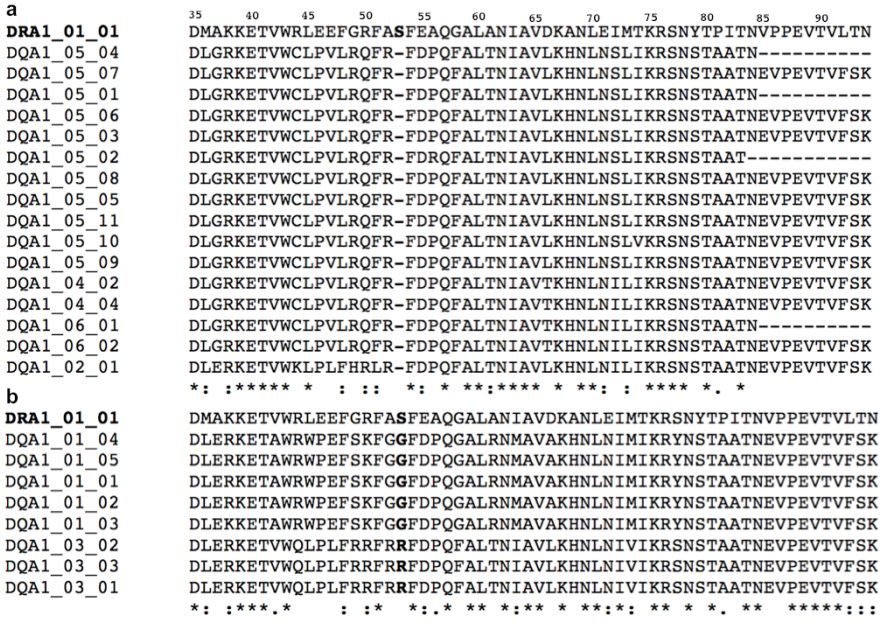

Figure 3.2. Part of sequence alignments of HLA-DQ alpha chains to HLA-DR reference sequence of HLA-DRA1*0101 molecule. a) Sequence alignments of HLA-DQ sequences with gaps, **b)** demonstrates the alignment of other HLA-DQ molecules to the same reference sequence. Reference sequence and the position corresponding to the insertion are marked in *bold*. The alignments were visualized using *ClustalW* [57].

MHC class II pseudo sequence

For constructing the *NetMHCIIpan* method, MHC class II molecules were represented by a pseudo sequence consisting of amino acid residues important for peptide binding. Amino acid residues comprising the pseudo sequence were defined as having their side chains pointing towards the peptide and being within 4.0 Å of the peptide-binding core in one or more of the MHC class II structures (including HLA-DR, HLA-DP and HLA-DQ molecules) available in the

PDB (www.pdb.org; [58]). The MHC molecules were aligned using the PyMOL molecular viewer [17] and interacting residue positions extracted according to the distance criterion. Among the interacting residues, only those found to be polymorphic across the sequences of MHC molecules used for the training of the method were considered. The final pseudo sequence is composed of 15 residues from the alpha chain and 19 residues from the beta chain. The interaction map between the peptide and MHC pseudo sequence is given in Figure 3.3.

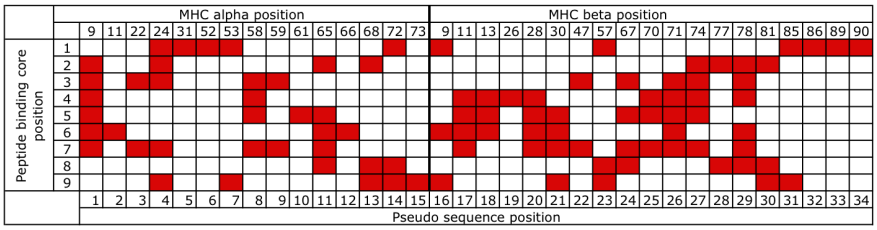

Figure 3.3. Interaction map between the peptide and MHC class II pseudo sequence. The *columns* give the MHC position numbering separately for alpha and beta chains and refer to HLA-DR. The *rows* show peptide binding core positions. *Red squares* marking interaction between a particular position of the peptide and MHC define contacts between corresponding two residues.

Method

The *NetMHCIIpan-3.0* method was implemented as a conventional feed-forward artificial neural network method as described in detail by Nielsen et al. [50]. The networks were trained using 5-fold cross-validation. The data set was split into five groups of peptides based on a common motif clustering as described by Nielsen et al. [59]. The difference in network architecture from the study presented by Nielsen et al. [50] was that network ensembles were trained with 10, 15, 40 and 60 hidden neurons. The BLOSUM50 matrix was used to encode peptide and MHC sequences for the network trainings. Each training was repeated 10 times with different initial configuration values as described in Nielsen et al. [50]. In total, 40 (4 different numbers of hidden neurons times 10 different random seeds) networks were used for each training/test set combination leading to 200 (5 folds times 40 networks) networks for each molecule.

Leave-one-out setup

In order to assess the predictive performance of the method in the situation where a molecule is not part of the training data, a leave-one-out (LOO) approach was applied. Using LOO, the binding data for the molecule in question were excluded from the training data. Since our data set has a large number of peptides that have been measured for binding to multiple molecules, we also removed peptides common between the evaluation and training data sets to ensure unbiased LOO trainings. In order not to reduce the training set too much in this type of LOO trainings, the evaluation set was split into three subsets resulting into three different 5-fold cross-validation trainings for each molecule. The details about such LOO setup are described in Karosiene et al. [60].

Nearest neighbour approach (*NN-finder*)

In order to evaluate the performance of the pan-specific method on the molecules that are not found in the training set, we set up a nearest neighbour prediction approach which in this study we call *NN-finder*. This approach represents the simplest method where the predictions of a query molecule are obtained by first finding its nearest neighbour and using a subsequent allele-specific method to predict the query binding specificity. First of all, for each molecule in question, we found a corresponding nearest neighbour from the training set. The distance between two MHC molecules was calculated from the amino acid similarity between the two pseudo sequences as described by Nielsen et al. [49], and the nearest neighbour to the molecule in question was defined as the molecule in the training set having the shortest distance. The binding data of each nearest neighbour were then used as training data for the corresponding query molecule. We retrained an allele-specific method from those training data using the *NNAlign* method [61] with settings identical to those used for *NetMHCII* [59]. The predictive performance for each query molecule was obtained by using its binding data as an evaluation set. In order for the performance to be directly comparable to the LOO results, the splitting of the evaluation set into three subsets was also used here.

Performance measures and statistical analysis

The predictive performance was measured in terms of Pearson's correlation coefficient (PCC) and area under the ROC curve (AUC). PCC

values vary between 0 and 1, where 1 represents perfect predictions and 0 random predictions. For AUC measures, a performance value of 1 corresponds to a perfect prediction, and a value of 0.5 reflects random predictions. For more details concerning the performance measures, see Nielsen et al. [50]. Throughout this study, PCC and AUC values were compared for different methods and evaluated using binomial tests with a significance level of 0.05.

Generation of HLA-II distance trees

For generation of the HLA-II distance tree, the most prevalent alpha and beta chains in the European population were selected as defined by the allele frequencies database (http://www.allelefrequencies.net) [62]. At a frequency threshold of 1%, we found 21 HLA-DR1, three HLA-DPA1, 12 HLA-DPB1, 12 HAL-DQA1 and 13 HLA-DQB1 alleles. We constructed all HLA-DPA1–HLA-DPB1 and HLA-DQA1–HLA-DQB1 combinations arriving at a total of 21 HLA-DR, 36 HLA-DP and 156 HLA-DQ molecules. Sorting (on a per-loci level) the different molecules on descending population frequencies, we constructed a functional redundancy deduced set containing 72 molecules using the *Hobohm1* algorithm [63] with redundancy defined as two molecules sharing a Pearson's correlation coefficient of 0.99 or above when comparing the predicted binding affinities on a set of 200,000 random natural 15-mer peptides. The set of 72 non-redundant HLA-II molecules is comprised of 21 HLA-DR, 14 HLA-DP and 37 HLA-DQ molecules. Next, we applied the *MHCcluster* method [64] to construct a tree describing the functional similarity between the different molecules. In short, the *MHCcluster* method functions as follows. Binding affinities of a set of 200,000 natural random 15-mer peptides are predicted for each of the HLA molecules using *NetMHCIIpan-3.0*. Next, the functional similarity between any two HLA molecules is defined by correlating the union of the predicted top 10% strongest binding peptides for each molecule. The similarity is 1 if the two HLA molecules are predicted to have a perfectly overlapping peptide repertoire and negative if there is no or very limited overlap. The distance between two molecules is defined as $1-$similarity. By using the unweighted pair group method with arithmetic mean clustering, the distance matrix is converted to a distance tree. Generating 100 distance trees using bootstrap estimates the significance of the distance tree. The trees are next summarized, and a consensus tree is made with branch bootstrap

values. Sequence logos were constructed from the predicted binding core of the top 1% strongest predicted binders using *Seq2Logo* method with default settings [65].

3.1.3 Results

In the following section, we give the results of applying the new pan-specific method: *NetMHCIIpan-3.0* to predict binding for a large set of MHC class II molecules from three human class II loci as well as a small set of mouse H-2 molecules.

NetMHCIIpan-3.0 method's new approach for getting pseudo sequence

In the most recent pan-specific MHC class II prediction method, *NetMHCIIpan-2.0*, the pseudo sequence is composed of 21 amino acids from positions within the HLA-DR beta chain that are in potential contact with a peptide using a 4.0 Å distance cut-off and polymorphic across the set of sequenced MHC class II molecules available at the time of the study [50]. For the *NetMHCIIpan-3.0* method described here, the pseudo sequence contains 19 residues from the beta chain of MHC molecules. The main difference between two pseudo sequence obtaining approaches resulting into different number of pseudo sequence positions is that the *NetMHCIIpan-3.0* considers polymorphism across the sequences of MHC molecules from the training set only. In order to evaluate this new approach for obtaining the pseudo sequence, we performed a 5-fold cross-validation training and compared the results of those reported for *NetMHCIIpan-2.0* [50]. In this comparison, only HLA-DR molecules were considered due to availability of results from both methods. Moreover, for the new method, only the part of the pseudo sequence corresponding to beta chain positions was included as the *NetMHCIIpan-2.0* method only includes beta chain residues in the pseudo sequence. The results are shown in Table 3.2.

The results in Table 3.2 demonstrate that the new approach for obtaining the pseudo sequence leads to a significantly (p values<0.05) improved predictive performance compared to the original approach when the pan-specific training approach is applied to the HLA-DR data set. The average increased from 0.688 to 0.695 and from 0.846 to 0.847 for PCC and AUC values, respectively. *NetMHCIIpan-3.0* achieves the highest performance for most of the molecules (PCC values are higher for 20 out of 24 molecules, and AUC values are

higher for 17 out of 23 molecules, excluding ties). The results demonstrate that the new approach of obtaining pseudo sequences for the neural network trainings improves the predictive performance of the method.

Table 3.2. Fivefold cross-validation performance for HLA-DR molecules of a pan-specific *NetMHCIIpan-2.0* compared with *NetMHCIIpan-3.0*.

Molecule name	#pep	#bind	*NetMHCIIpan-2.0*		*NetMHCIIpan-3.0*	
			PCC	AUC	PCC	AUC
DRB1*0101	7,685	4,382	0.711	0.846	**0.716**	**0.848**
DRB1*0301	2,505	649	0.709	0.864	**0.723**	**0.868**
DRB1*0302	148	44	0.525	0.757	**0.569**	**0.786**
DRB1*0401	3,116	1,039	0.670	**0.848**	**0.671**	0.846
DRB1*0404	577	336	0.630	0.818	**0.656**	**0.829**
DRB1*0405	1,582	627	0.698	0.858	**0.712**	**0.862**
DRB1*0701	1,745	849	**0.740**	**0.864**	0.732	0.862
DRB1*0802	1,520	431	0.526	0.780	**0.542**	**0.784**
DRB1*0806	118	91	**0.796**	0.924	0.792	**0.933**
DRB1*0813	1,370	455	0.746	0.885	**0.751**	**0.888**
DRB1*0819	116	54	0.608	**0.808**	**0.610**	0.803
DRB1*0901	1,520	622	0.634	0.818	**0.647**	**0.828**
DRB1*1101	1,794	778	0.777	0.883	**0.780**	0.883
DRB1*1201	117	81	0.764	0.892	**0.768**	**0.896**
DRB1*1202	117	79	0.769	0.900	**0.778**	**0.910**
DRB1*1302	1,580	493	0.634	**0.825**	**0.636**	0.822
DRB1*1402	118	78	0.694	0.860	**0.724**	**0.879**
DRB1*1404	30	16	**0.613**	**0.737**	0.511	0.629
DRB1*1412	116	63	**0.757**	**0.894**	0.754	0.890
DRB1*1501	1,769	709	0.653	0.819	**0.682**	**0.830**
DRB3*0101	1,501	281	0.690	0.850	**0.700**	**0.858**
DRB3*0301	160	70	0.736	0.853	**0.752**	**0.869**
DRB4*0101	1,521	485	0.675	0.837	**0.699**	**0.847**
DRB5*0101	3,106	1,280	0.765	0.882	**0.769**	**0.885**
Total	**33,931**	**13,992**				
Average			**0.688**	**0.846**	**0.695**	**0.847**
***p* value**			**0.002**	**0.035**		

NetMHCIIpan-2.0 is the method described by Nielsen et al. [50], which uses pseudo sequences composed of polymorphic amino acids that have one or more potential contacts with a peptide (length=21). The performance values for this method are taken from the publication. *NetMHCIIpan-3.0* employs pseudo sequences obtained by finding contacts that side chains of MHC molecules have with the peptide and taking polymorphic positions within the training set (length=19 for beta chain only). The values in *bold* show the higher score for each molecule for corresponding performance measures (PCC or AUC). *p* values were obtained using a binomial test excluding ties.
#pep − number of peptide binding data available for each molecule, #bind − number of peptides that have a binding affinity stronger than 500 nM.

Per-locus training versus cross-loci training

To the best of our knowledge, all pan-specific prediction methods for HLA class II molecules available up to date are limited to HLA-DR. In this study, we introduce an approach for combining residues from the alpha and beta chains into one pseudo sequence. The procedure for the pseudo sequence construction is universal to all MHC class II complexes allowing the pan-specific method to be trained in a cross-loci/cross-species manner (see "Materials and methods") arriving at one common method suitable for all MHC class II molecules.

To evaluate how such a cross-loci/cross-species impacts the predictive performance of the method, we compared per-locus (and per molecule, see below) training with the pan-specific training including cross-loci data. The results are shown in Figure 3.4. Detailed results are given in Table B.3. The figure gives average PCC and AUC values for each locus when the method was trained in a cross-loci manner including all HLA molecules and when trained using binding data restricted to each locus, respectively. As can be seen from the figure, the overall performance of the two training approaches is similar. For HLA-DR, the predictive performance improved when training in a cross-loci manner compared to per-locus training. For HLA-DQ and HLA-DP, the performance on the other hand is slightly reduced. This reduction is, however, only significant for HLA-DQ and only when measuring AUC performance values.

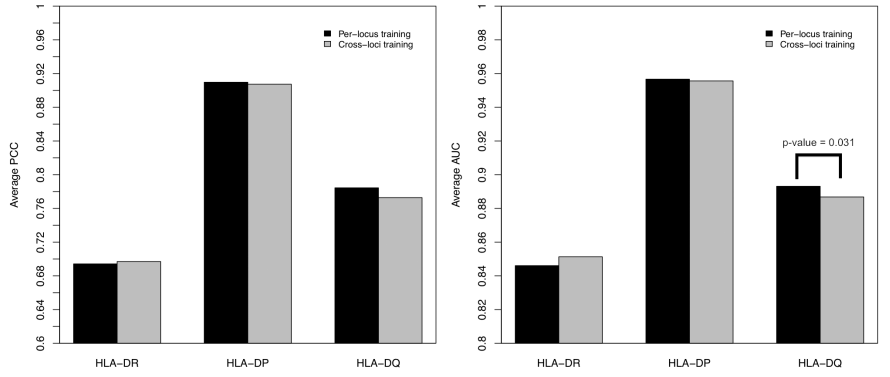

Figure 3.4. Comparison of the method performance when trained on per-locus data and cross-loci data. Average PCC and average AUC values for each locus are demonstrated on the *left* and *right* panel, respectively. Significant p values are given above the *bars* for corresponding loci. The difference in predictive performance between the per-locus and cross-loci training is significant only for HLA-DQ when measuring AUC performance values.

Pan-specific versus allele-specific method

As a pan-specific approach, the method presented in this study benefits from the information even from molecules covered by limited binding data or molecules from different loci/species. To demonstrate this, we present a comparison of the performance values obtained for the *NetMHCIIpan-3.0* method and allele-specific *NN-align* method using 5-fold cross-validation (see Table 3.3). The *NN-align* prediction method was trained as described by Nielsen and Lund [66], using the same data partitioning based on the common motif clustering approach as used for *NetMHCIIpan-3.0*.

The results presented in Table 3.3 demonstrate that the pan-specific *NetMHCIIpan-3.0* predictor significantly outperforms the allele-specific *NN-align* method (p value$<$0.0001 for both PCC and AUC values). These results show that the pan-specific method benefits from the binding data measured to different molecules. It also demonstrates that adding data from other molecules significantly boosts the performance for molecules represented by limited peptide-binding measurements. Out of ten molecules described by less than 400 data points and less than 100 binders, ten and nine are shown to obtain higher performance using pan-specific predictor in terms of PCC and AUC values, respectively. The allele-specific *NN-align* method gives higher PCC and AUC values for three molecules all defined by more than 1,700 peptide-binding data.

Leave-one-out performance

In order to demonstrate how the method performs when predicting binding to novel and uncharacterised molecules, we performed a LOO experiment. In the LOO experiment, a molecule in question was excluded from the training data set, and its binding data acted as an evaluation set. Likewise, all peptides, included in the binding data set of the given molecules, were excluded from the training data in order to avoid biassed overlapping between the evaluation and the training sets. This was done in three rounds by removing one third of the peptides from the evaluation set at a time and performing 5-fold cross-validation training in each round. The performance for the query molecule was obtained by combining the predictions of all three evaluation subsets. For the benchmark, we compared the results with the predictive performance of the simple allele-specific approach based on finding the nearest neighbour, *NN-finder*. For this method, the molecule in question and its binding data were also acting as an

Table 3.3. 5-fold cross-validation performance for the pan-specific *NetMHCIIpan-3.0* method compared with the allele-specific *NN-align* method using our benchmark data set.

Molecule name	#pep	#bind	*NetMHCIIpan-3.0*		*NN-align*	
			PCC	AUC	PCC	AUC
HLA-DPA1*0103-DPB1*0201	1,404	538	**0.922**	**0.957**	0.912	0.952
HLA-DPA1*0103-DPB1*0401	1,337	471	**0.929**	**0.962**	0.914	0.958
HLA-DPA1*0201-DPB1*0101	1,399	597	**0.905**	**0.948**	0.902	0.941
HLA-DPA1*0201-DPB1*0501	1,410	443	**0.868**	**0.954**	0.865	0.950
HLA-DPA1*0301-DPB1*0402	1,407	523	**0.912**	**0.957**	0.905	0.956
HLA-DQA1*0101-DQB1*0501	1,739	522	0.791	0.901	**0.802**	**0.907**
HLA-DQA1*0102-DQB1*0602	1,629	813	**0.698**	**0.872**	0.659	0.855
HLA-DQA1*0301-DQB1*0302	1,719	386	0.723	0.813	**0.729**	**0.833**
HLA-DQA1*0401-DQB1*0402	1,701	559	**0.807**	**0.914**	0.794	0.903
HLA-DQA1*0501-DQB1*0201	1,658	549	0.802	0.902	**0.809**	0.902
HLA-DQA1*0501-DQB1*0301	1,689	863	**0.816**	**0.919**	0.810	0.918
H-2-IAb	660	126	**0.713**	**0.884**	0.664	0.856
H-2-IAd	379	70	**0.577**	0.816	0.420	**0.856**
DRB1*0101	7,685	4,382	**0.717**	**0.849**	0.682	0.831
DRB1*0301	2,505	649	**0.708**	**0.859**	0.671	0.836
DRB1*0302	148	44	**0.601**	**0.800**	0.266	0.627
DRB1*0401	3,116	1,039	**0.659**	**0.841**	0.609	0.817
DRB1*0404	577	336	**0.663**	**0.838**	0.595	0.784
DRB1*0405	1,582	627	**0.711**	**0.862**	0.683	0.843
DRB1*0701	1,745	849	0.729	**0.861**	**0.732**	0.860
DRB1*0802	1,520	431	**0.515**	**0.771**	0.478	0.750
DRB1*0806	118	91	**0.778**	**0.927**	0.707	0.886
DRB1*0813	1,370	455	**0.740**	**0.881**	0.719	0.867
DRB1*0819	116	54	**0.608**	**0.809**	0.334	0.661
DRB1*0901	1,520	621	**0.652**	**0.828**	0.572	0.788
DRB1*1101	1,794	778	**0.770**	**0.879**	0.749	0.868
DRB1*1201	117	81	**0.787**	**0.909**	0.694	0.848
DRB1*1202	117	79	**0.783**	**0.916**	0.682	0.849
DRB1*1302	1,580	493	**0.612**	**0.814**	0.607	0.804
DRB1*1402	118	78	**0.753**	**0.890**	0.546	0.800
DRB1*1404	30	16	**0.611**	**0.728**	0.259	0.603
DRB1*1412	116	63	**0.764**	**0.896**	0.574	0.789
DRB1*1501	1,769	709	**0.677**	**0.831**	0.629	0.803
DRB3*0101	1,501	281	**0.683**	**0.851**	0.613	0.816
DRB3*0301	160	70	**0.754**	**0.864**	0.543	0.773
DRB4*0101	1,521	485	**0.693**	**0.846**	0.687	0.840
DRB5*0101	3,106	1,280	**0.760**	**0.882**	0.740	0.865
Average			**0.735**	**0.871**	**0.664**	**0.838**
***p* value**			**<0.0001**	**<0.0001**		
***p* value**[a]			**0.002**	**0.021**		

NN-align is the method described by Nielsen and Lund [66]; *NetMHCIIpan-3.0* is the method described here. The values in *bold* show the higher score for each molecule for corresponding performance measures (PCC or AUC). The p values for PCC and AUC are given below the first columns of PCC and AUC values respectively.
#pep − number of peptide binding data available for each molecule, #bind − number of peptides that have a binding affinity stronger than 500 nM.
[a]p − values of the 10 molecules characterised by <400 data points and <100 binders.

evaluation set, while the training data were composed of the peptide binding data of the molecule from the training set having the shortest distance to the molecule in question. The results are depicted in Figure 3.5 and presented in detail in Table B.4. It is apparent from the results that *NetMHCIIpan-3.0* outperforms the *NN-finder* approach for all loci in terms of average PCC and AUC. Even though we find general improvement when comparing the pan-specific method to the nearest neighbour approach, a significant difference (due to the small number of molecules for each subset) is observed only for HLA-DR molecules (p value$<$0.0001). The significance for the mouse allelic locus (H-2) was not assessed due to only two molecules being available.

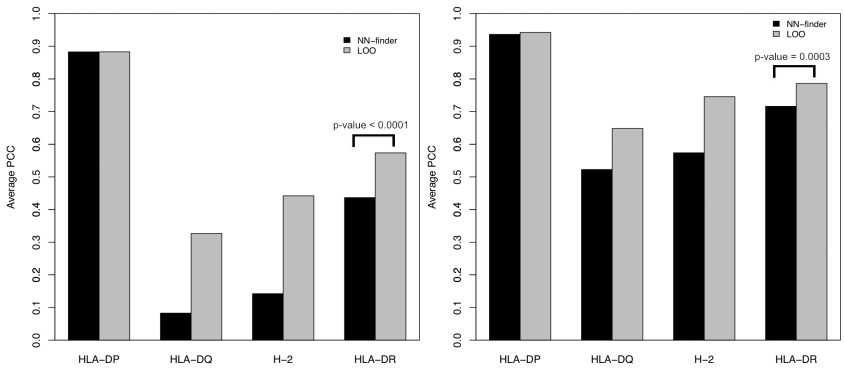

Figure 3.5. Leave-one-out results for the *NetMHCIIpan-3.0* method in comparison with the *NN-finder* approach. Average performance measures in terms of PCC and AUC are given in the *left* and *right* panel, respectively. Significant p values are given above the bars for corresponding loci (not available for H-2 locus).

The method shows decreased predictive performance with the distance to the nearest neighbour from the training set (Figure 3.6). The figure illustrates how the predictive performance of the pan-specific method depends on the distance to the nearest neighbour calculated in terms of pseudo sequence similarities as explained in "Materials and methods". Regression analysis showed that the performance is decreased significantly with the increasing distance (p value=0.031, exact permutation test).

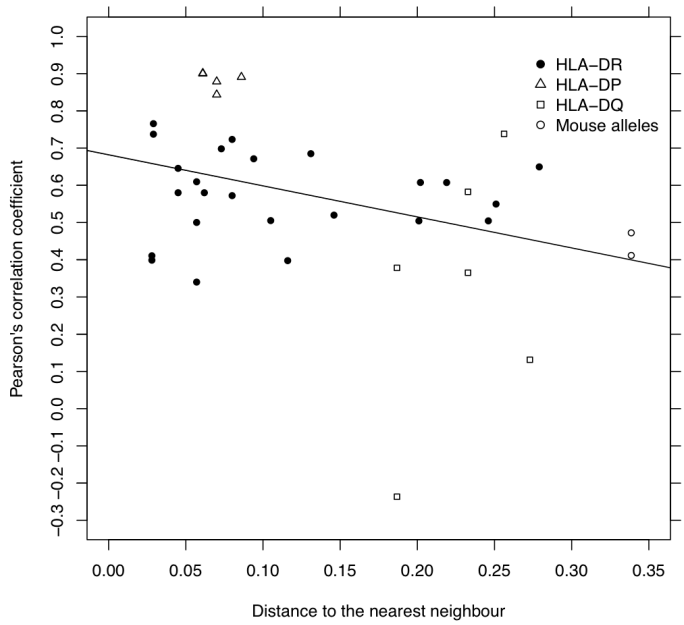

Figure 3.6. Predictive performance of the *NetMHCIIpan-3.0* method for the molecules from our data set as a function of distance to the nearest neighbour. The performance was obtained using LOO setup as explained in the "Materials and methods" section. The distance to the nearest neighbour was calculated as described by Nielsen at al. [49]. The *solid line* represents the least square fit for the data.

Independent evaluation of the final *NetMHCIIpan-3.0* predictor

For the final evaluation of the pan-specific method common for HLA-DR, HLA-DP, HLA-DQ and mouse molecules, the method was trained using all the available data (52,062 data points) and evaluated on an independent HLA-DR evaluation set containing 9,860 data points. The method was compared with the most recent version of the class II pan-specific predictor *NetMHCIIpan-2.0* [50]. From the results given in Table 3.4, it is apparent that *NetMHCIIpan-3.0* outperforms *NetMHCIIpan-2.0* (average PCC is 0.603 compared with 0.586 and average AUC 0.807 compared with 0.802). Although the difference in performances was observed not to be significant, the new pan-specific method shows higher performance for most of the molecules from the evaluation set (*NetMHCIIpan-3.0* wins nine out of 13 and six out of 12 times in terms of PCC and AUC measures, respectively).

The *NetMHCIIpan-3.0* method presented and benchmarked in this paper was implemented as a web server and is available online at http://www.cbs.dtu.dk/services/NetMHCIIpan-3.0.

Table 3.4. Independent evaluation of the *NetMHCIIpan-3.0* method compared with the performance of the *NetMHCIIpan-2.0* predictor.

Molecule name	#pep	#bind	*NetMHCIIpan-3.0*		*NetMHCIIpan-2.0*	
			PCC	AUC	PCC	AUC
DRB1_0101	717	550	**0.820**	0.908	0.817	**0.910**
DRB1_0301	703	408	0.699	0.850	**0.703**	**0.862**
DRB1_0701	682	375	0.754	0.873	**0.771**	**0.882**
DRB1_0801	838	363	**0.738**	**0.875**	0.713	0.861
DRB1_1101	813	426	**0.790**	0.901	0.787	**0.902**
DRB1_1301	803	462	**0.573**	**0.792**	0.488	0.753
DRB1_1302	765	404	**0.392**	**0.713**	0.289	0.668
DRB1_1501	758	218	**0.499**	0.764	0.496	**0.767**
DRB3_0202	726	287	0.490	**0.755**	**0.495**	0.750
DRB3_0301	782	449	0.555	0.776	**0.602**	**0.800**
DRB4_0101	778	235	**0.292**	**0.654**	0.254	0.635
DRB4_0103	764	474	**0.538**	**0.798**	0.505	0.795
DRB5_0101	731	461	**0.699**	0.841	0.697	0.841
Average			**0.603**	**0.808**	**0.586**	**0.802**
p value			**0.267**	**1.000**		

NetMHCIIpan-2.0 method is an updated version of the method proposed by Nielsen et al. [50]. *NetMHCIIpan-3.0* is the method presented in this study. Molecule names in *bold* show molecules that were not part of the training set. The values in *bold* show the higher score for each molecule for corresponding performance measures (PCC or AUC) between the two methods. *p* values were obtained using binomial test for PCC and AUC values.
#pep − number of peptide binding data available for each molecule, #bind − number of peptides that have a binding affinity stronger than 500 nM.

Functional clustering of HLA class II molecules

Given the potential of the *NetMHCpan-3.0* method to predict binding for any MHC class II molecules with known alpha and beta chain protein sequences, we next applied the method to give an overall estimate of the functional diversity of molecules from the HLA-DR, HLA-DP and HLA-DQ loci molecules. The analysis of the most prevalent alpha and beta chains in the European population was done as described in the "Materials and methods", and the result is shown in Figure 3.7. From the figure, it is apparent that the molecules encoded at the three loci display very limited functional overlap. Also, one can notice that the HLA-DP locus molecules display a very limited functional diversity compared to the HLA-DR and HLA-DQ loci

molecules. This is also reflected when measuring the functional diversity of an HLA locus in terms of the mean and standard deviation of the intra-locus distances. Here, we find that the mean intra-distance is significantly shorter ($p<0.001$, Student's t-test) for HLA-DP compared to HLA-DQ and HLA-DR. We can further relate these differences in functional diversity to the degree of polymorphism at a population level of the HLA pseudo sequences of each locus. Estimating polymorphism in terms of the Kullback–Leibler information content (or divergence sum) [67] for the 34 positions in the pseudo sequence for the three loci, we find that this value is significantly higher ($p<0.001$, t-test) for DP compared to DQ and DR, hence demonstrating that DP molecules share a significantly lower degree of polymorphism compared to the molecules at the two other loci.

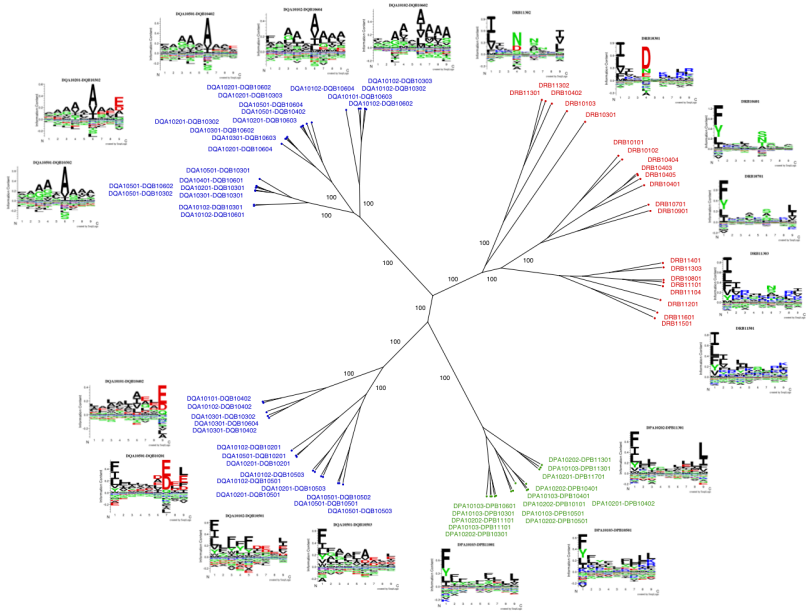

Figure 3.7. Functional clustering of the 72 HLA molecules from the European population. HLA-DR molecules are displayed in *red*, HLA-DP molecules are displayed in *green* and HLA-DQ molecules are shown in *blue*. Sequence logos showing the binding motif are presented for selected molecules representing the different specificity groups.

In terms of the predicted functionality, we recover for HLA-DRB1 the overall clustering proposed earlier [49] with 9 well-defined subgroups (supertypes). For HLA-DQ, the overall functionality seems reduced compared to HLA-DR, with only 5/6 well-defined subgroups,

and as stated above, HLA-DP seems to encode for the least functionally diverse set of molecules with only one specificity group being present. Sequence logos for selected subgroups and subgroup representatives are included in the figure to illustrate the functional difference between the different molecules. In general, the predicted binding motifs are in agreement with the motifs proposed earlier for the limited set of HLA class II molecules experimentally characterised by peptide binding data [68, 61].

3.1.4 Discussion and conclusion

Identification of peptides binding to MHC is a critical step in understanding T cell immune responses. The human MHC genomic region (HLA) is extremely polymorphic comprising several thousands alleles, many encoding a distinct molecule. The potentially unique specificities remain experimentally uncharacterised for the vast majority of HLA molecules.

The sequences of human MHC class II molecules stored in the IMGT database [34] cover over 600 different HLA-DR variants and more than 6,000 different combinations of HLA-DP and HLA-DQ alpha and beta chains. Of these many molecules, less than 30 HLA-DR and only 5 HLA-DP and 6 HLA-DQ molecules have been experimentally characterised with binding data allowing for an accurate estimate of their binding specificity. In order to span this gap, several methods have been developed and benchmarked during the last decade for the prediction of peptide binding to MHC class II molecules (for review, see [10]). Here, pan-specific methods play an important role, as they are capable of giving predictions to those molecules, which have not yet been characterised experimentally. However, until now, MHC class II pan-specific binding prediction approaches have been limited to HLA-DR molecules, leaving a gap in the general understanding of binding specificities for HLA-DP and HLA-DQ molecules [10].

In this paper, we present a pan-specific method, *NetMHCIIpan-3.0*, capable of predicting peptide binding to all HLA molecules. To the best of our knowledge, this is the first predictor common for HLA-DR, HLA-DP and HLA-DQ molecules. The method is based on artificial neural networks and is trained on 52,062 quantitative peptide binding data covering all HLA as well as two mouse molecules.

NetMHCIIpan-3.0 uses a new approach for defining the peptide-binding environment of MHC in terms of pseudo sequence as compared with the most recent *NetMHCIIpan-2.0* method [50]. The main

difference between the two approaches for obtaining pseudo sequence is that for *NetMHCIIpan-3.0*, only polymorphism within the training set is considered whereas the *NetMHCIIpan-2.0* method includes polymorphism across all known MHC class II sequences. Our results demonstrated that the new approach for defining the pseudo sequence leads to a significantly improved predictive performance.

Several large-scale benchmarks were carried out that demonstrated that the *NetMHCIIpan-3.0* method fulfils the requirements for the pan-specific methods. Its performance was found to be significantly better than that of the allele-specific *NN-align* predictor [66]. In particular, the method outperformed *NN-align* for molecules characterised with only a limited number of binding data. These results hence agree with the results obtained when benchmarking the original *NetMHCIIpan* method [50] and underline the unique power of the pan-specific approach in providing accurate predictions also for molecules characterised with limited peptide-binding data as it has also been demonstrated previously for MHC class I predictions [60, 33, 32].

To mimic the situation where the *NetMHCIIpan-3.0* method is applied to predict binding for uncharacterised MHC molecules, we conducted a panel of LOO experiments. In these experiments, binding data for one MHC molecule at a time were removed from the training, and the predictive performance next evaluated on the left-out data. The LOO results demonstrated that the proposed method is capable of predicting binding affinity for the molecules for which no binding data are available in the training process. In addition to this, the method showed decreased predictive performance with the distance to the nearest neighbour from the training set, which is in agreement with previous studies on MHC class I [36, 60, 31].

From the results included here, one can notice that HLA-DP molecules demonstrate higher performance values compared with DR and DQ performances. As this high performance is maintained also for the allele-specific (and pseudo sequence independent) *NN-align* approach, the high performance is not due to the fact that DP molecules are found to be very close to each other in terms of pseudo sequence similarity and function. The reason for this different performance is rather related to the differences within distributions of the binding data available for each locus. The HLA-DP data are very well separated with the majority of the data being either strong or very weak binders. This is in strong contrast to the data for DQ and DR molecules where the majority of the data have intermediate binding

affinity (data not shown). This difference in binding affinity distribution strongly influences the predictive performance, as well-separated data sets (as is the case for DP) in general achieve a higher predictive performance. To prove this further, we have performed the analysis where we, in the evaluation of the prediction methods, mimicked the distribution of the HLA-DP data for the DR loci. The analysis demonstrated that the performance for DR molecules is significantly increased when the data match the distribution observed for the DP molecules compared to the original DR data distributions (data not shown).

The *NetMHCIIpan-3.0* predictor showed higher performance when compared with the *NetMHCIIpan-2.0* method on the external evaluation set. The increased performance of the *NetMHCIIpan-3.0* method demonstrates its promising ability to improve when more data become available for molecules from other loci/species.

We further presented a powerful application of the developed pan-specific predictor. We applied the *NetMHCIIpan-3.0* method to functionally cluster the most prevalent HLA alleles of the European population. For HLA class I, clustering of molecules into supertypes was proposed by the analysis carried out using experimental data [69, 70] and extended by applying pan-specific class I predictor [33]. However, for MHC class II, the amount of experimental data remains too limited to perform such a cluster analysis, which therefore so far has been limited to HLA-DR molecules [49]. The analysis performed here hence is the first study suggesting reduction of polymorphism of HLA class II molecules by definition of clusters based on similarities in predicted functional binding specificities. Such clustering builds a base for facilitating identification of T helper cell epitopes within different ethnic groups having a high value in the design of epitope-based vaccines. As we have discussed earlier, developing a cross-loci method for MHC class II is complicated due to the differences of sequences and structures of different loci [50]. However, as also suggested in this earlier publication, with increasing amounts of binding data covering HLA-DP and HLA-DQ molecules, pan-specific methods may benefit from cross-loci training. In this study, we demonstrated that this is indeed the case. The performance of the proposed *NetMHCIIpan-3.0* method when trained on cross-loci was shown to be comparable with that of a method trained on per-loci data. So far, no significant improvement was found between the per-loci and cross-loci trained method. However, this is most likely due to the very low number of HLA-DQ and HLA-DP molecules included in

the study and is expected to change with the inclusion of more data covering HLA-DP and HLA-DQ molecules. The situation is hence parallel to that observed for MHC class I. Here, at first, only limited data characterising HLA-A and HLA-B molecules were available for development of the original version of *NetMHCpan*, and an optimal performance was obtained when the method was trained in a loci-specific manner [33]. Only when binding data became available covering more HLA molecules as well as MHC molecules from non-human species (including non-human primates) was a cross-loci/cross-species training strategy found to be optimal [36].

The training strategy outlined here for the MHC class II pan-specific prediction method is highly flexible and readily allows inclusion of novel data both in terms of peptides and MHC molecules. This flexibility makes the method a powerful and unique platform for the development of a pan-specific MHC class II predictor covering not only the human class II molecules but also MHC molecules from other species of interest. Lessons learned from MHC class I suggest that such a true pan-specific approach is feasible and that prediction accuracies for both human and non-human MHC molecules can be greatly boosted given the ability of the pan-specific method to leverage information across species and loci [71].

In conclusion, we believe the proposed *NetMHCIIpan-3.0* method is an important step forward in boosting MHC class II binding predictions covering a large number of molecules from different species and therefore reduces experimental costs for the immunologists working within the field of epitope-based vaccine design.

Acknowledgements MN is a researcher at the Argentinean national research council (CONICET). This project has been funded in whole or in part with federal funds from the National Institutes of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under contract nos. HHSN272201200010C and HHSN272200900045C.

Chapter 4

Bioinformatics identification of antigenic peptides

This chapter gives an overview of bioinformatics methods for immunology developed in our group. All the methods are publically available and include prediction tools for peptide binding to MHC class I and class II molecules (*NetMHCcons*, *NetMHCIIpan*), a method for visualizing MHC binding motifs (*MHCMotifViewer*), a patient-specific predictor of HLA restriction elements and optimal epitopes (*HLArestrictor*), a method for predicting proteasomal cleavage sites (*NetChop*), and integrated tools for class I antigen presentation predictions (*NetCTL*, *NetCTLpan*). This chapter relates to the work presented in Chapters 2 and 3 by including detailed guidelines on how to use prediction methods for peptide–MHC binding. Instructions on input submission are provided as well as explanations of the output. Note that the chapter presents the old version of the *NetMHCIIpan* method limited to HLA-DR molecules and not the new version common for all HLA class II molecules as presented in Chapter 3. This is due to the fact that the paper included in this chapter was published prior to the paper presented in Chapter 3.

4.1 Paper III

The following paper was published as a chapter in a book edited by Peter van Endert titled "Antigen Processing: Methods and Protocols" in the series *Methods in Molecular Biology, vol. 960* in January 2013.

Bioinformatics identification of antigenic peptide: predicting the specificity of major MHC class I and II pathway players

Ole Lund, Edita Karosiene, Claus Lundegaard, Mette Voldby Larsen, and Morten Nielsen

Center for Biological Sequence Analysis, Department of Systems Biology, Technical University of Denmark, DK-2800 Lyngby, Denmark

Abstract

Bioinformatics methods for immunology have become increasingly used over the last decade and now form an integrated part of most epitope discovery projects. This wide usage has led to the confusion of defining which of the many methods to use for what problems. In this chapter, an overview is given focusing on the suite of tools developed at the Technical University of Denmark.

Keywords Immune · Epitope · MHC · HLA · Class I · Class II · Antigen processing · Proteasome · TAP · Visualization · Bioinformatics · Prediction · Web server

4.1.1 Introduction

Experimental methods for analyzing antigenic peptide generation, transport, and binding to major histocompatibility complex (MHC) class I molecules are expensive and time consuming. While bioinformatics methods can never replace experiments in the laboratory, they may in a highly cost-effective manner guide the experimental efforts in a direction that increases the likelihood of discovering immunologically important responses. At the Technical University of Denmark, we have over the last decade developed a number of methods for predicting which part of an antigen most likely is presented to the immune system. A complicating factor is that the MHC molecules associated with response to foreign antigens are encoded at several loci. Furthermore, these genes are the most polymorphic in the human genome and thousands of different alleles are known. Many of these alleles encode different variants of MHC molecules having different peptide binding specificities. However, it is possible to cluster alleles with similar specificities into functional groups called supertypes, first described by Sette and Sidney [70]. The pioneering methods for predicting binding to MHC class I molecules such as BIMAS [72] and SYPEITHI [73] helped initiate the field of immunological bioinformatics, but these methods have since been surpassed by newer methods like the ones described in this chapter, and we propose that experimental efforts may be minimized by basing the experiments on these newer methods.

4.1.2 Binding of peptides to MHC

In recent years numerous methods for predicting binding to MHC molecules have been proposed. These methods can broadly be divided into two classes: one being the allele-specific and one being the pan-specific methods. Allele-specific methods are constructed for a given allele, and can interpolate between different ligands and give predictions for peptides for which no binding data are available. An obvious limitation by these methods is that predictions can only be made for alleles for which a number of binding data is already available. This requirement has been circumvented by the so-called pan-specific methods, which can also interpolate between different MHC alleles and thus make predictions for alleles for which no known binders are available. This strongly increases the number of alleles for which predictions can be obtained, from the few hundreds for which

binding data is available to the more than 3,000 for which the protein sequence is known.

The accuracy of methods for MHC peptide binding prediction depends critically on the available data characterizing the binding specificity of the MHC molecules. This makes it very difficult for the non-expert user to choose the most suitable method for predicting binding to a given MHC molecule. To complicate things even further, it has been demonstrated that consensus methods defined as combinations of two or more different methods led to improved prediction accuracy.

4.1.3 Prediction of MHC class I peptide binding

To benefit from the consensus approach and to guide the non-expert user on selecting the most appropriate binding prediction method for a given MHC class I molecule, we have recently developed the *NetMHCcons* method. The method is available at http://www.cbs.dtu.dk/services/NetMHCcons.

The method integrates predictions from three well-established prediction methods (*NetMHC* [27, 20], *NetMHCpan* [36, 33], and *PickPocket* [31]) and allows the user in an automatic manner to obtain the most accurate predictions for any given MHC class I molecule of known protein sequence. The three methods included in *NetMHCcons* are state of the art and have performed well in recent benchmarks [28, 29, 74, 32, 75, 76]. For MHC class I alleles with well-characterized binding specificity, the method is defined as a combination of the *NetMHC* and *NetMHCpan* methods, and for alleles with unknown binding specificity, the method is defined in terms of the *NetMHCpan* method combined with *PickPocket*. For details on the method and its benchmark performance refer to [60].

The submission site of the server can be seen in Figure 4.1.

1. Select method. By default, the consensus method (*NetMHCcons*) is selected but each of the three individual prediction methods can be run separately.

2. Select Allele(s). To aid in navigation, the alleles listed by default are limited to the human supertype representatives, but all alleles from different human/animal loci can be selected under "Select species/loci" (the list of selectable alleles is limited to alleles with well-characterized binding specificity when using the

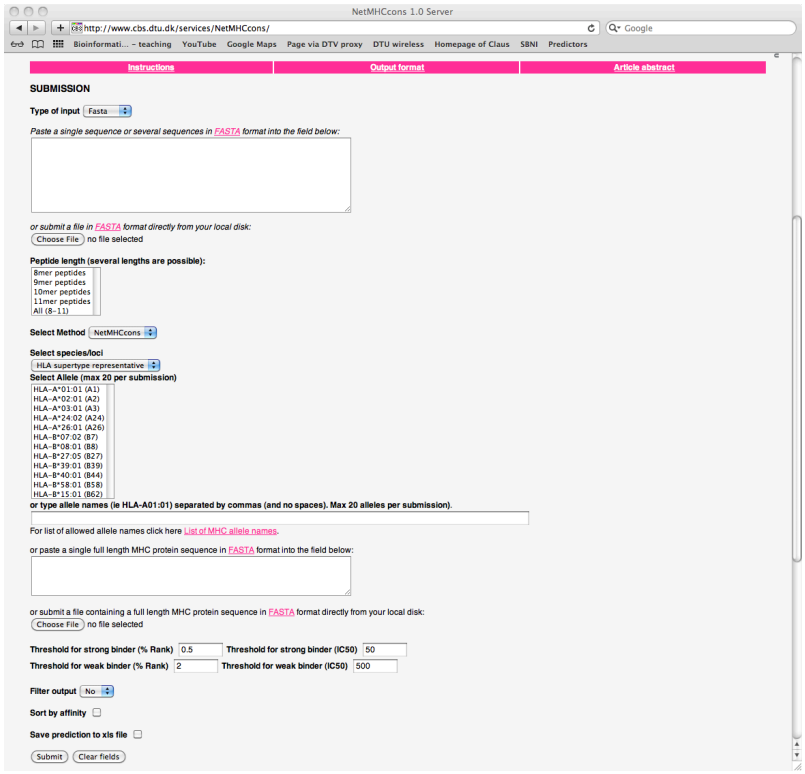

Figure 4.1. Submission site of *NetMHCcons* server. Two submission types are handled – a list of peptides or protein sequence(s). The server provides a possibility for the user to choose MHC molecules in question from a list of alleles or alternatively upload a full-length MHC protein sequence of interest. The user has a choice of setting the threshold for defining strong and weak binders based on predicted affinity (IC50) or %Rank. The output can be sorted based on predicted binding affinity as well as filtered on the user-specified thresholds.

NetMHC method). In the MHC allele selection field, multiple alleles can be selected but the selection is limited to 20 alleles per submission. Multiple alleles can also be inputted as a comma-separated list. For the pan-specific methods (*NetMHCcons*, *NetMHCpan*, and *PickPocket*) the user can upload a file containing the protein sequence of an MHC class I molecule that is not among the available, selectable alleles, and the method will perform peptide binding predictions for this molecule.

3. Provide input sequence. The input can either be in peptide raw text or protein FASTA format. In peptide format, each line is assumed to be a separate peptide. All peptides must be of equal

length. In FASTA format, the sequence of each protein must be preceded by a line beginning with a ">". When FASTA input is used, multiple different epitope lengths from 8 to 11 residues can be selected.

4. Select output formatting. By default the output is sorted by the residue number, but the user can choose to sort the output by the predicted binding affinity. Predictions for all the input peptides given are by default but by setting "Filter output" to "Yes", only the peptides predicted to bind stronger than the defined thresholds are given in the output. The output can optionally be saved to a file readable by spreadsheet applications for further processing by the user.

5. Press submit.

6. Wait for the server to produce output. The output from the server consists of a list of peptides, each associated with three prediction values: 1−log50k(aff), Affinity, and %Rank. The 1−log50k value is the raw score provided by the prediction method, and is related to the predicted binding affinity value as 1−log(aff)/log(50,000). The %Rank score gives % rank of the prediction score to a set of 200,000 random natural 9-mer peptides. Thresholds can be selected for which peptides to report as strong binders (SB) and week binders (WB). The peptides are labeled as a strong binder if the %Rank score or the binding affinity is below the specified thresholds for the strong binders. Likewise, peptides are labeled as weak binders if the %Rank or the binding affinity is above the thresholds of strong binders, but below the specified threshold for the weak binders.

References to other well-performing methods for prediction of MHC class I binding can be found in one of the several reviews that have been written on the subject including a recent one from our group [76].

4.1.3.1 Prediction of MHC class II peptide binding

For class I, alignment-free methods like the ones described earlier can readily be applied, since the binding motif is well characterized and most natural peptides that bind MHC class I are of the same length.

For MHC class II, the situation is quite different due to the great variability in the length of natural MHC-binding peptides. This variation in ligand length makes alignment a crucial and integrated part of estimating the MHC-binding motif and predicting peptide binding. During the last decade, large efforts have been invested in developing data-driven prediction methods for MHC class II peptide binding. For an overview of these refer to one of the many reviews written on the theme including the one written by our group [10].

The binding of a peptide to a given MHC class II molecule is predominantly determined by the amino acids present in the peptide-binding core. However, peptide residues flanking the binding core (the so-called peptide flanking residues, PFR) do also to some degree affect the binding affinity of a peptide [77, 78, 66]. Most published methods for MHC class II binding prediction focus on identifying the peptide-binding core only, ignoring the effects on the binding affinity of PFRs. In the work by [66] it was demonstrated that the additional information provided by the PFR leads to significantly improved predictions.

Two high-performing methods for MHC class II binding prediction developed by our group are *NetMHCII* [66] and *NetMHCIIpan* [49, 50]. The *NetMHCII* method is allele-specific and allows for peptide–MHC binding predictions to a set of 14 HLA-DR, six HLA-DQ, six HLA-DP, and two mouse H2 class II alleles. *NetMHCIIpan* is HLA-DR pan-specific, allowing for prediction of peptide binding to all HLA-DR molecules with a known protein sequence. Several benchmark studies have demonstrated these methods to be high performing and state of the art [29, 38, 79, 80].

1. Select input sequences. Both methods accept input either as individual peptides in raw text format or as protein sequence(s) uploaded in FASTA format (see earlier). If protein sequences are uploaded, the user can specify the peptide length and predictions are made for each overlapping peptide of the specified length. Multiple MHC alleles can be specified.

2. Customize search. The input to (and output from) the *NetMHCIIpan* method is very similar to that of *NetMHCII*. Only does the *NetMHCIIpan* method (as was the case for MHC class I methods described earlier) allow the user to upload a file containing the protein sequence of an HLA-DR molecule that is not among the available, selectable alleles, and the method will perform binding predictions for this molecule. Likewise the user

can define the prediction score threshold values used to classify prediction as strong and weak binders. Also can the output from the *NetMHCIIpan* server be saved to a file readable by most spreadsheet applications for further processing by the user.

3. Select output formatting. By default the output is sorted by the residue number but the output can also be sorted by affinity. Predictions for all peptides are by default given but by setting a "Threshold", only the peptides predicted to bind stronger than the defined threshold (in 1−log50k units) are given in the output.

4. Press Submit.

5. Wait for output. As for the MHC class I prediction server described earlier, the output from the MHC class II prediction servers consists of a list of peptides, each associated with the predicted binding core and three prediction values: 1−log50k(aff), Affinity, and %Rank. The 1−log50k value is the raw score provided by the prediction method, and is related to the predicted binding affinity value as 1−log(aff)/log(50,000). The %Rank score gives % rank of the prediction score to a set of 200,000 random natural peptides. Peptides are labeled as a strong binder if the binding affinity is below 50 nM. Likewise, peptides are labeled as a weak binder if the binding affinity is below 500 nM.

4.1.4 *MHCMotifViewer*: browsing and visualization of MHC class I and class II binding motifs

The number and binding specificity diversity of MHC molecules can be overwhelming for most users. To help get an overview, we have developed the *MHCMotifViewer* server (http://www.cbs.dtu.dk/biotools/MHCMotifViewer/). The homepage is shown in Figure 4.2.

1. Select species/loci. By clicking on "Human alleles" different loci can be selected. For other species the user is taken directly to a list of alleles.

2. Select allele. Clicking on one of the thumbnail pictures will create a larger logo for that allele. This is shown for HLA-A*0103 in the right panel of Figure 4.2. On the x-axis the nine positions in the binding motif are given. The height of the columns of

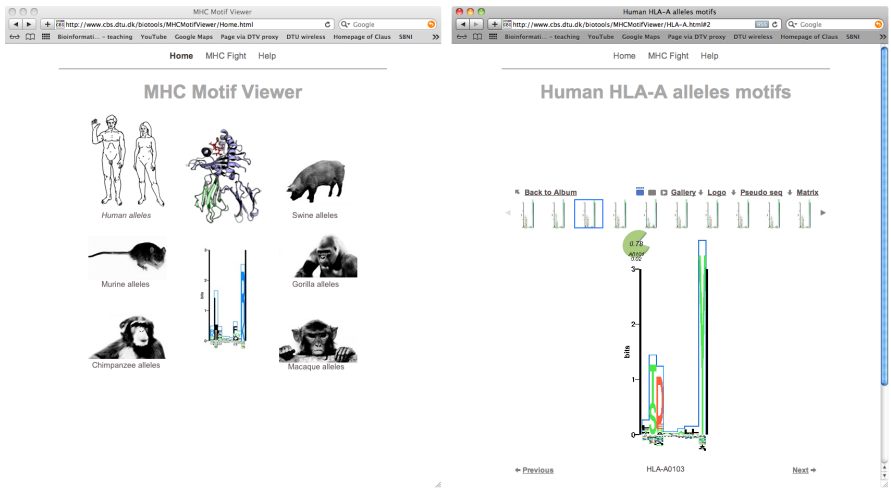

Figure 4.2. The *MHCMotifViewer* server. *Left panel* shows the homepage of the *MHCMotifViewer* server where the organism can be selected. Human, murine, chimpanzee, swine, gorilla, and macaque alleles can be browsed. In the *right panel* an allele from the Human HLA-A loci (HLA-A*0103) is selected and its motif is displayed as the sequence logo representation.

letters at each position corresponds to the predicted contribution to binding on that position calculated according to the formula developed by Kullback–Leibler [67]. The amino acids for which their frequency differs the most from the background frequency for that amino acid in proteins in general are shown with the highest letters. The overrepresented amino acids are shown above the x-axis, and the underrepresented ones below.

The binding motif of up to four different alleles can be shown side by side by clicking on "MHC Fight". By default, all four alleles are the same, but by clicking on the blinking curser, the allele name can be changed by deleting (part of) the name using the backspace key and typing the new name. By holding the curser over the "K" button, the display will shift between showing a Kullback–Leibler (K), and a Sequence frequency (S)-based logo. In a sequence frequency-based logo the relative height of each letter within a column is proportional to the frequency of the corresponding amino acid at that position. A more detailed explanation can be found in [45].

4.1.5 *HLArestrictor*: patient-specific HLA restriction elements and optimal epitopes within peptides

Considering the many different peptides that can be generated, even from a small target protein, and the extensive polymorphism of the presenting MHC molecules, identifying pathogen-specific, HLA-restricted T cell epitopes can be an immense experimental task. To reduce this complexity, one could conveniently exploit a commonly used approach of T cell epitope discovery: testing overlapping peptides (OLP) with a length of 15–18 amino acids in IFNγ release, ELISPOT, or flow cytometric intracellular staining assays. Given a positive peptide it is, however, not a simple task to find the actual stimulatory peptide (minimal epitope) and the presenting HLA restriction element. By way of example, a 15-mer peptide tested positive in a patient with six different HLA class I molecules could potentially be explained by any one of the possible $22 * 6 = 132$ 8–11-mer HLA combinations. To lower this experimental burden, we have developed an immunoinformatics method, *HLArestrictor* (www.cbs.dtu.dk/services/HLArestrictor) [43], which has been tailored to support CTL epitope discovery in individual subjects. As inputs, the method requires the amino acid sequence of the positive peptide(s) and the HLA type of the individual in question (high-resolution HLA typing, e.g., HLA-A*0101, and preferably for all relevant loci, e.g., for HLA-A, -B, -C for HLA class I-restricted CTL responses). Using these inputs, *HLArestrictor* creates all possible 8, 9, 10, and 11-mer peptides from the target peptides(s), predicts their binding to all the HLA molecules in question, and generates an output file consisting of the most likely peptide/HLA combination(s). Peptide/HLA tetramers is one of the most efficient means to validate T cell epitopes, and *HLArestrictor* can also be viewed as a tool for efficient design of specific peptide/HLA tetramers. The vehicle behind the *HLArestrictor* is the *NetMHCpan* method, and the Webpage interface bears a high resemblance to the interfaces for *NetMHCpan*, *NetMHCIIpan*, and *NetMHCcons*.

1. Select input sequences. Multiple peptide sequences can be uploaded in FASTA format.

2. Select HLA alleles. The host HLA allele names can be selected or typed in.

3. Select lengths of epitopes. The lengths of the predicted minimal epitopes can be specified.

4. Select prediction threshold. Threshold values defining how the prediction scores are interpreted can be specified in terms of threshold values for strong and weak binding peptides.

With default settings, the server will scan all possible 8, 9, 10, and 11-mer peptides from the target peptides(s) for binding to all HLA alleles of the host and report peptides with %Rank score less than or equal to 0.5 or affinity stronger than 50 nM as strong binders, and peptides with %Rank score less than or equal to 2 or affinity stronger than 500 nM as weak binders.

4.1.6 Interpreting the output from the prediction servers

All the prediction servers described here provide three prediction scores for each peptide, as well as a label classifying the peptides into groups of strong and weak binders. For the end user, these prediction values are meant to serve as a guide to make rational peptide selections for epitope discovery and/or interpretation of immune responses. This opens for questions on how to define relevant thresholds relating prediction values to likelihoods of a peptide being a T cell epitope. It is becoming apparent that not all MHC molecules present peptides at the same binding threshold [81, 46]. The two distinct prediction values (affinity and %Rank) are included to capture these intrinsic differences between MHC molecules in terms of binding threshold for presentation of peptides. Large benchmark studies have demonstrated that the vast majority of known CTL epitopes are characterized by having a %Rank score less than or equal to 2 or an affinity stronger than 500 nM [43, 44, 82]. These numbers are hence used as default values for the definition of weak binding peptides for all MHC class I prediction methods. For MHC class II the situation is less clear. While it is clear that the prediction values correlate strongly with the measured binding affinity, few studies have investigated the direct correlation between %Rank score, predicted affinity values, and the likelihood of a peptide being immunogenic. The default values for the classification of peptides as weak and strong binders are hence poorly justified for MHC class II, and the relationship to the likelihood of being immunogenic is at the best poorly investigated. However, for both MHC class I and class II it is clear that using the prediction score to rank peptides provides a highly cost-effective tool to guide the experimental efforts in a direction that increases the likelihood of discovering immunologically important responses.

4.1.7 The MHC class I antigen presentation pathway

As part of the protein recycling machinery, proteins in our cells are cut into shorter peptides by the proteasome. These peptides may bind to the transporter associated with antigen processing (TAP) and be transferred to the endoplasmic reticulum (ER). Inside the ER, peptides may be further trimmed, bind the MHC class I molecules, and be transported along with it to the cell surface. If the peptide is of non-self origin, the peptide–MHC complex may bind to a T cell receptor (TCR) on a cytotoxic T cell, which will then initiate an immune response. More detailed descriptions of and references to these processes can be found in other chapters of this book. The three most essential of the above steps (cleavage by the proteasome, transport by TAP, and binding to MHC class I) have been modeled by bioinformatics methods that can predict which peptides from a given protein/organism are most likely to be presented to the immune system.

4.1.8 *NetChop*: proteasomal cleavages (MHC class I ligands)

A method has been developed, which predicts proteasomal cleavage sites. The method is called *NetChop* [83], and a server is available at http://www.cbs.dtu.dk/services/NetChop/.

1. Select prediction method. Two different versions of the method exist: "C term 3.0" and "20S 3.0". They differ by the sets of data they have been trained on. While *NetChop 20S 3.0* has been trained on in vitro constitutive proteasome protein digests, *NetChop C term 3.0* has been trained on natural MHC class I ligands. The rationale for the latter is that the proteasome most likely has generated the ligand s C-terminal ends. *NetChop C term 3.0* predicts the C-terminal end of CTL epitopes with a higher specificity than *NetChop 20S 3.0* (has fewer false positives). The main reason for this is that since it is trained on natural ligands, it predicts a combination of MHC class I binding, TAP transport efficiency, and proteasomal cleavage.

2. Select input sequence. The input to the server is proteins or peptide fragments in FASTA format (see earlier). The method assigns a score in the range 0–1 to each residue in the input sequence. The higher the score, the more likely it is that the proteasome cleaves after this residue. Note that the score refers to cleavage of the peptide bond on the C-terminal side of the residue to which the score is assigned.

3. Select prediction threshold. By default, 0.5 is used as the threshold for predicted proteasomal cleavage. In the output, scores above the threshold are assigned an "S" in the C (cleavage) column, while lower scores are assigned a ".".

4.1.9 *NetCTL* and *NetCTLpan*: integrated class I antigen presentation

Two methods that integrate predictions of proteasomal C-terminal cleavage, TAP transport efficiency, and MHC class I binding for the overall prediction of MHC class I presentation called *NetCTL* and *NetCTLpan* have been developed by our group. The *NetCTL* method [84] is available at http://www.cbs.dtu.dk/services/NetCTL/. For prediction of proteasomal cleavage, it uses *NetChop C term 3.0* (see above). Predictions of TAP transport efficiency are based on the weight matrix-based method described by Peters et al. [85]. For predictions of MHC class I binding, *NetMHC* (see above) is used.

1. Select input sequence. The input to the server is proteins or peptide fragments in FASTA format (see earlier).

2. Select Allele/supertype. The user must specify for which of the 12 MHC class I supertypes the predictions should be performed (A1, A2, A3, A24, A26, B7, B8, B27, B39, B44, B58, or B62; for a definition of supertypes see [70]). *NetCTL* integrates the individual scores from *NetChop*, the TAP matrix, and *NetMHC* into one, overall score. To allow for comparison between different MHC class I supertypes, the rescaled affinity is used (see [84] for details on how the rescaled affinity is calculated).

3. Select weighting of processing steps. As default, the relative weight of C-terminal cleavage is 0.15, while it is 0.05 for TAP transport efficiency. The default weights have been found to result in optimal performance, but can be changed by the user.

4. Select prediction threshold. The user can also specify which threshold to use for defining a CTL epitope. By default it is 0.75.

5. Select sorting of output. Lastly, the user can specify how the 9-mers of the input sequence should be sorted in the output. In the default "no sort" option, the 9-mers are listed according to the order in which they appear in the input sequence. Alternatively,

> they can be sorted according to the combined score, MHC binding, proteasomal cleavage, or TAP. For each 9-mer sub-peptide in the input sequence, the output will list the predicted affinity and the prediction scores of proteasomal cleavage, TAP binding, and finally a combined score. If the combined score is above the selected threshold for defining an epitope, it is marked by an "E".

NetCTLpan is an extended and improved version of *NetCTL*, which is available at http://www.cbs.dtu.dk/services/NetCTLpan/ and described in detail in [46]. The C-terminal proteasomal cleavage and TAP transport efficiency are predicted as for the *NetCTL* method, while MHC class I binding is based on the *NetMHCpan* method. While *NetCTL* only allows for predictions of peptides restricted by one of the 12 MHC class I supertypes, *NetCTLpan* allows for predictions of CTL epitopes binding any MHC class I molecule for which the protein sequence is known. As for the above-described pan prediction methods, it is additionally possible to paste in or upload a file containing the protein sequence of an MHC class I molecule that is not among the available, selectable alleles, and the method will perform CTL epitope predictions for this molecule. *NetCTLpan* furthermore performs predictions for 8–11-mers. The Webpage interface of *NetCTLpan* bears a high resemblance to the interfaces of *NetCTL*. One difference is that it is possible to select a threshold that the combined score must exceed for the predictions to be displayed in the output page. By default, this threshold is -99.9, which results in all predictions being displayed. In the output page, the same values are listed as in the *NetCTL* output. Additionally, the %Rank value is given (see above for definition of the %Rank value).

Chapter 5

Bioinformatics analysis of epitopes from yellow fever virus vaccine strain 17D

During my PhD, I have been involved in a collaborative research project dedicated to discover T cell epitopes from a yellow fever virus. The prediction methods presented in previous chapters composed the largest part of our contribution to the project. The methods were highly used by our collaborators for the selection of potential epitopes. In addition to the prediction tools, we contributed to the project by performing a short study on epitope mapping and distribution analysis. This chapter presents the main findings of this project.

5.1 Introduction

5.1.1 Yellow fever disease and yellow fever virus

Yellow fever (YF) is one of the most infectious diseases found mostly in tropical regions of Africa and South America. It is a mosquito-borne disease caused by a yellow fever virus from the *Flavivirus* genus of the *Flaviviridae* family [86]. YF virus infection in humans may lead to fever, nausea, chills, headache, back and muscle aches etc. In severe cases, infection causes liver damage with jaundice and leads to death [87]. The number of YF infections reported by The World Health Organization reaches 200,000 including about 30,000 deaths every year. YF disease has been re-emerging over the last two decades [88] and since no antiviral drug is available for the treatment, vaccination remains the only and most important solution for the prevention of the disease. The live-attenuated vaccine against YF is considered to be one of the most safe, effective and affordable viral vaccines that has ever been created. The vaccine is based on the 17D virus strain and was developed by Max Theiler in the 1930's. Since then, more than 500 million doses have been used with more than 95% successful protection for at least 10 years [89, 90].

5.1.2 YF virus genome polyprotein

The YF virus vaccine strain 17D (YF-17D) contains a 3,411 amino acid polyprotein (GeneBank ID: AF052437.1 [91]) composed of three structural and several non-structural proteins [92, 93, 94, 95]. The RNA of the YF virus is about 11 kilobases long and has only one open reading frame within which the protein encoding genes have been determined as follows: 5'-C-prM-E-NS1-NS2A-NS2B-NS3-NS4A-NS4B-NS5-3' [96]. The composition of the genome polyprotein is schematically represented in Figure 5.1. The N-terminal end of the polyprotein encodes three structural proteins (capsid protein C; membrane precursor, prM; and envelope protein E) that compose immature virions assembled in the endoplasmic reticulum (ER). The remaining seven proteins are non-structural (NS) proteins with different functions important for the replication and assembly of the virus [97, 98].

5.1.3 Experimental assays

Several immunological methods have been developed in order to measure T cell responses to individual antigens and identify potential epitopes from a chosen pathogenic genome. One group of such methods

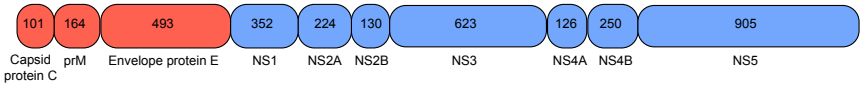

Figure 5.1. Schematic representation of the YF virus 17D vaccine strain polyprotein. Structural and non-structural proteins are represented by the *red* and *blue* units, respectively. The numbers inside the units indicate the length (number of amino acids) of each protein.

is based on detection of various cytokines and includes enzyme-linked immunosorbent spot (ELISPOT) and intracellular cytokine staining (ICS) assays. Such assays are based on measuring a difference between production of cytokines by T cells in the presence and absence of an antigen. An ELISPOT assay is highly sensitive towards quantification of low-frequency T cell responses and is very widely used to detect antibody-producing cells as well as to identify responses of the T cells specific for viral antigens. On the other hand, even with a lower sensitivity, the ICS assay possesses an ability to detect responding cell types. Due to these features both assays are commonly used together, using ELISPOT assays for initial screening, followed by the ICS method to validate detected responses and identify the responding cell types [99, 100]. For further validation of antigen-specific T cells, tetramer staining technique is used [101]. The method is based on tetramers consisting of multiple bound peptide–MHC complexes in order to increase binding avidity for T cells. Tetramers bind only peptide-specific T cells, therefore allowing in vitro identification of T cells specific for infectious agents [101].

5.1.4 Data set

For our analysis, we used a list of 68 unique $CD8^+$ T cell epitopes (see Table C.1) identified by ELISPOT and ICS assays using blood samples from a large cohort of donors vaccinated against yellow fever disease. Phenotypic HLA allelic composition of the donors covered ten HLA-A, 16 HLA-B and one HLA-C allele. Some of the epitopes have been found to be associated with several MHC molecules, resulting in 86 epitope–MHC combinations as presented in Table C.2.

5.2 Bioinformatics analysis

5.2.1 Epitope mapping to YF virus polyprotein

Using the list of epitopes presented in Table C.1, we mapped the repertoire of tested epitopes to the YF-17D polyprotein. The results are depicted in Figure 5.2. Each position of the polyprotein obtained a score equal to the number of epitopes starting at that position. The figure demonstrates that the distribution of epitopes across the full genome is almost flat resulting only in a few gaps and one characteristic peak.

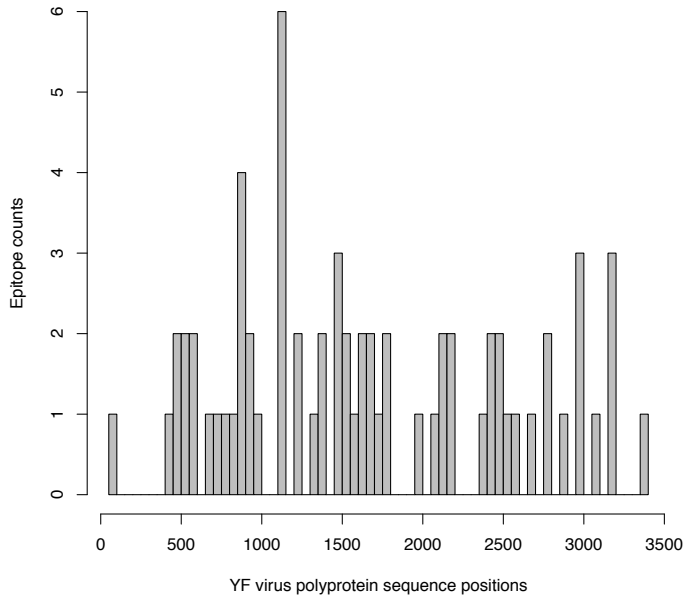

Figure 5.2. Experimentally determined epitopes mapped to YF-17D polyprotein. Epitope count for each position represents a number of epitopes starting at that position.

Zooming into the peak between positions 1,000 and 1,500 of the polyprotein, we find corresponding epitopes mapped to the virus sequence as shown in Figure 5.3. Six epitopes comprising the highest score in Figure 5.2 are found to be closely located on the genome sequence. The epitopes are clustered into two groups of three peptides belonging to either NS1 or NS2A proteins. We note that epitopes starting at position 1,114 have an overlap of ten amino acids and

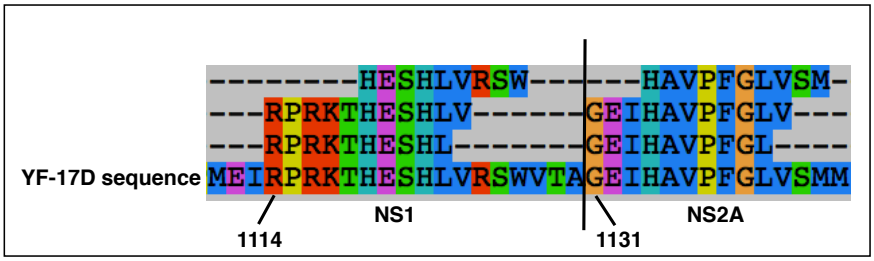

Figure 5.3. Region of the YF-17D polyprotein containing the highest number of epitopes. Six epitopes were identified at the end of protein NS1 and at the beginning of protein NS2A. The epitopes are mapped to the YF virus polyprotein sequence between positions 1114 and 1143. Different *colours* mark different groups of amino acids.

are both found to be restricted to the HLA-B*0702 allele. Similarly, epitopes starting at the first position of NS2A protein (1,131 position of the polyprotein) are both found to bind to the HLA-B*4001 allele. Tetramer staining analysis demonstrated that the RPRKTHESHL and RPRKTHESHLV peptides are recognized by two different T cell populations. On the other hand, the GEIHAVPFGLV and GEIHAVPFGL were recognized by the same T cell population and represent one actual epitope.

5.2.2 Distribution of epitopes within YF virus proteins

In order to investigate if some YF virus proteins contain a higher number of immunogenic signals than others, we calculated the density of epitopes within each protein. The results are presented in Table 5.1 and visualised in Figure 5.4. The density was calculated as the number of epitopes per single position of a protein. For comparison, we defined a background density equal to the density of the whole polyprotein. We used statistical analysis to compare proportions of *#epitopes/#residues* of each protein to the background proportion and assessed Z-scores. For a confidence level of 95%, the Z-scores should satisfy the equations: Z-score≤ -1.96 and Z-score≥ 1.96. Figure 5.4 shows that most of the proteins (seven out of ten) reach the density above the background value of 0.02. However, no statistically significant differences between density of $CD8^+$ epitopes within each protein and whole polyprotein sequence were observed (see Table 5.1).

One can notice from Table 5.1 that in order to maintain uniform density across all proteins, longer proteins should contain a higher

Table 5.1. Epitope density of proteins from the YF-17D virus strain.

Protein name	#epitopes	#residues	Density	Z-score
Capsid protein C	1	102	0,010	-0,727
prM	0	164	0,000	-1,826
Envelope protein E	10	493	0,020	0,052
NS1	11	352	0,031	1,410
NS2A	6	224	0,027	0,703
NS2B	4	130	0,031	0,859
NS3	13	623	0,021	0,152
NS4A	4	126	0,032	0,922
NS4B	5	250	0,020	0,007
NS5	14	905	0,015	-0,875

#epitopes refers to the number of epitopes within each protein, *#residues* refers to the length of each protein in terms of amino acid number, *Z-scores* were calculated comparing proportions of *#epitopes/#residues*.

number of epitopes than shorter proteins. This is confirmed in Figure 5.5 which demonstrates that the observed number of epitopes is a direct function of protein size. Regression analysis confirmed that the number of epitopes increases significantly with increased sequence length of a protein ($p<0.0001$, exact permutation test).

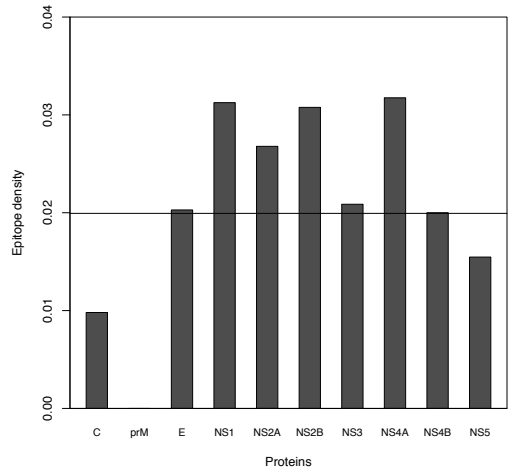

Figure 5.4. Epitope density for each YF virus protein. The density was calculated as the number of epitopes per single protein residue. *Horizontal line* corresponds to the background epitope density of the amino acid sequence of the whole virus genome.

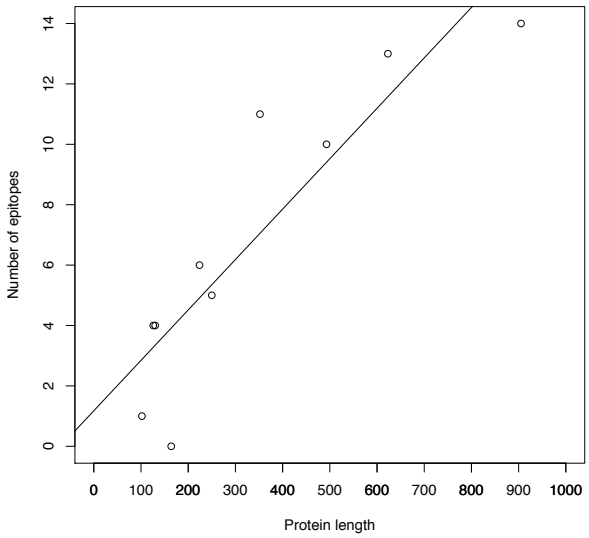

Figure 5.5. Dependency of the number of epitopes on the protein size. The length of the protein is given in amino acid residues. *Solid line* represents the least square fit for the data.

5.2.3 Epitope prediction models

The key step in antigen recognition by $CD8^+$ T cells is presentation of antigen on MHC class I molecules. With this in mind, we analysed the impact of the peptide–MHC predictions for the epitope identification process. Having the epitope distribution profile presented in Figure 5.2, we investigated to what degree epitope distribution can be explained by the profile of predicted binders to MHC molecules.

The full sequence of the YF virus polyprotein was scanned for overlapping 8- to 11-mers, resulting in 13,610 peptides. For this peptide set we predicted binding affinities using the *NetMHCcons* method [60]. From the output we selected only the strong binders with a predicted binding affinity $<$50 nM or a %Rank $\leq$0.5%. A %Rank score is a different way to measure how well a peptide binds to a particular MHC molecule and was previously described in Hoof et al. [44]. The score is calculated by ranking the peptide in question by its predicted affinity along with 200,000 natural random 9-meric peptides for the same HLA molecule. A %Rank of 0.5% means that only 0.5% of random peptides have a predicted binding affinity stronger than that of the query peptide.

The selection of the predicted strong binders from our peptide set was then used to make an HLA binding profile for the YF-17D sequence. For this, we used phenotypic frequencies of all typed donors representing Danish population. For each position, we calculated an HLA frequency score corresponding to the sum of phenotypic HLA frequencies for all HLA alleles from our data set having a strong binder at that particular position. For comparison, the HLA frequency score for the epitopes was obtained by adding up phenotypic frequencies of the alleles binding the epitopes found in each position of the sequence. The plot is shown in Figure 5.6 on the left panel and demonstrates high correlation between the two profiles. The right panel of Figure 5.6 shows corresponding profiles for the score where the HLA frequency was replaced by a binary summation − adding 1 if the position is part of the strong binder or an epitope, and adding 0 if it is not. The correlation between the two curves seems to decrease in this case. Pearson's correlation coefficient calculated for the two curves is equal to 0.406 ($p<0.0001$, exact permutation test) when the real HLA frequencies were used and PCC=0.221 ($p<0.0001$, exact permutation test) when only binding/non-binding strategy was applied. The difference between the PCC of the binary and the frequency-based models was found to be significant ($p<0.001$, using bootstrap). The correlation coefficients and the p values were calculated using fitted curves and not the raw data in order to remove noise occurring from having too few epitopes. As can be seen from the results, epitope profile can to a high degree be explained by the strong predicted binders, and MHC frequency improves the model for epitope predictions.

5.2.4 Selection of potential epitopes using prediction methods

We showed that the key step in identifying YF epitopes is finding strong binders to their restricted MHC molecule. This makes peptide-MHC binding prediction methods very important in the pre-selection process of potential epitopes worth testing in the lab. During the recent years our group has developed several MHC binding prediction methods, as presented in Chapter 4. The methods have been used by different groups of experimentalists to choose potential epitopes against different pathogens, however one issue remains unclear. As explained in Chapter 4, some prediction methods such as *NetMHCpan*, *PickPocket* and *NetMHCcons* provide two kinds of scores − binding affinity and %Rank − for definition of strong binders. Binding affinity in terms of nM is the most common predicted measure

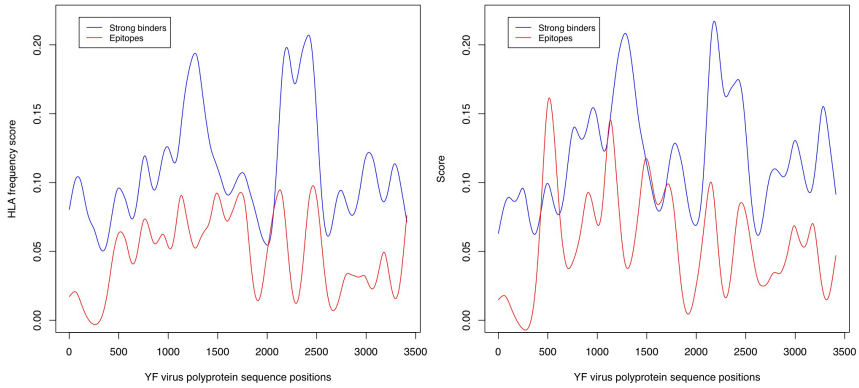

Figure 5.6. Correlation between predicted strong HLA binders and identified YF virus epitopes. The profiles for strong HLA binders and epitopes are calculated by summing phenotypic population frequencies of HLA molecules from the data set (*left*) and using binding/non-binding binary summing approach (*right*). The set of strong binders was created by selecting all 8–11mer peptides from the YF virus sequence and using *NetMHCcons* method to predict binding to HLA molecules. The PCC was calculated using 500 overlapping points from each curve.

used by immunologists, leaving the %Rank as a rarely used measure in the epotipe pre-selection process. We employed yellow fever virus epitope data to investigate the impact of both measures.

Using the *NetMHCcons* prediction method, we investigated what percentage of all 86 epitopes was identified with different cut-offs of log-transformed affinity ($1-\log50{,}000(\text{aff})$) and the %Rank score. The results depicted in Figure 5.7 show that going in the direction from strong to weak binders, the %Rank curve is much steeper than the log-affinity curve. More than 95 % of epitopes can be identified with a %Rank cut-off of 2% while for the binding affinity the log-score has to go down to 0.2 (corresponding to an affinity of 5,800 nM) in order to identify the same amount of epitopes. In addition to this, a ROC curve in Figure 5.7 shows predictive performance of the method when using %Rank and log-score to differentiate binders from non-binders. Here, 86 YF epitopes were acting as true positives and all the other 8–11mer peptides were assigned to be actual negatives. It is apparent in Figure 5.7 that at the same specificity, the rank curve has higher sensitivity and vice versa. For example, at the specificity of 0.980, the log-score has a sensitivity of 0.919 while the rank reaches sensitivity

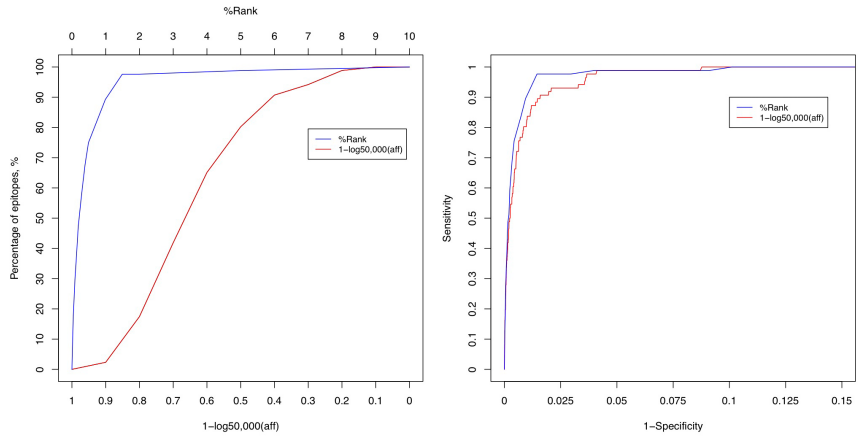

Figure 5.7. Comparison of the %Rank and the log-transformed affinity scores for choosing potential epitopes. Percentage of epitopes identified at different cut-offs for both measures are shown on the *left*. The *right* panel gives ROC curves of predictive performances obtained using both, %Rank and log-affinity measures.

of 0.977. Moreover, at the sensitivity of 0.900, the log-score has specificity of 0.984 (FP=5,860) and for the rank score the specificity is 0.991 (FP=3,450). The AUC values were calculated to be 0.995 and 0.993 for the %Rank and log-score, respectively, and to be significantly different ($p<0.01$, using bootstrap), making the %Rank a better measure to identify potential epitopes.

5.3 Discussion

One can expect that YF virus and other viruses from the *Flaviviridae* family contain a higher number of highly immunogenic epitopes within the structural proteins. Indeed, previous studies on T cell responses against West Nile virus from the same family, suggested that envelope protein E is one of the most immunogenic proteins [102, 103]. In this study, using a set of 86 YF epitopes identified in a large cohort of donors, we investigated whether the epitope distribution depends on the structure and function of the proteins.

Mapping of the validated epitopes on to the YF virus polyprotein demonstrated that there is no distinctive highly immunogenic region within the whole sequence. The epitopes were found to be evenly distributed, with the more dense regions being affected by the bias of the highly overlapping peptides existing in the data pool. For what

concerns epitope density, none of the proteins showed to have significantly higher density of immunogenic peptides than the proteome as a whole. Furthermore, we demonstrated that the number of epitopes depends strongly on the size of the protein. Hence, these findings do not confirm what was found by previosly mentioned studies [102, 103]. On the other hand, the results are in agreement with the study published by our group on the West Nile virus [82].

A strong correlation between the HLA profile of predicted strong HLA binders and actual validated epitopes restricted to a particular HLA molecule, suggested that amino acid composition of the YF virus polyprotein might be the most important property for selection of potential epitopes. Based on such results, the YF epitope discovery process can be mostly guided by the peptide–HLA binding, identification increasing the importance of prediction methods for MHC binding. In order to facilitate pre-selection process of potential candidates, we examined the impact on the selection when using %Rank score and log-transformed affinity score as selecting factors. Even though binding affinity measured in nM is preferred by immunologists working within the field, our analysis demonstrated the power of %Rank score. On the validated YF epitopes we demonstrated high sensitivity and high selective power of the %Rank score. Also, %Rank is a better measure because different HLA molecules show different binding promiscuity. For example, running predictions for 200,000 natural random peptides showed that at an affinity threshold of 500 nM, HLA-A*0201 binds 3.8% of the random peptides, while for HLA-A*0101 this number reaches only 0.4%. Therefore, the use of the %Rank measure should strongly be considered for evaluating the potential epitopes as vaccine candidates.

In summary, having a large data set of validated $CD8^+$ epitopes from a YF virus allowed us to use bioinformatics tools to investigate epitope distribution on the virus proteome, and to expand the knowledge for guiding the epitope discovery process. The preliminary results of our study can facilitate research dedicated to extension of the epitope repertoir not only for yellow fever, but also for other viruses of the *Flaviviridae* family, and contribute to the development of new vaccines.

Chapter 6

Epilogue

Identification of peptides binding to MHC molecules is a crucial step in understanding cellular responses and in facilitating research dedicated to epitope discovery, which is useful for development of new vaccines and immunotherapies. In this thesis, I present my contribution to the development of computational methods for predicting peptide–MHC interactions, which can reduce the experimental effort needed to identify potential epitopes. The work presented here was mostly dedicated to improve and develop pan-specific prediction tools able to predict binding to any MHC molecule with a known protein sequence, and in this way broadening their applicability within immunological research.

First, we addressed the issue that the prediction accuracy of state-of-the-art MHC class I methods depends strongly on the experimental data available that defines the MHC molecule in question. Inspired by the previous works that demonstrated the power of combining two or more methods in order to improve prediction accuracy [37, 38, 31, 32], we developed a consensus method, *NetMHCcons*, presented in Chapter 2. The predictor combines one allele-specific and two pan-specific methods and gives, in an automatic manner, the best predictions for a chosen MHC molecule. Considering prediction accuracy of the *NetMHCcons* and other pan-specific methods, it is apparent that the predictive performance of such methods is mostly defined by the distance to the nearest neighbour in the training set. In order to improve these methods, one should work in a dedicated manner to fill the gaps in the MHC binding specificity space, by identifying novel molecules that are more distant from the ones with a characterized binding specificity.

The *NetMHCcons* method outperforms each of the separate methods and builds a strong base for developing new consensus approaches. A recent study by Harndahl et al. demonstrated the importance of peptide–MHC stability for identification of immunogenic peptides [104]. This led to the development of the *NetMHCstab* method for predicting stability of peptide–MHC complexes [105]. A combination of stability predictions with *NetMHCcons* showed significantly increased performance when identifying potential T cell epitopes [105]. This emphasizes the power of the *NetMHCcons* method to be combined with different prediction tools. In future, it is possible to integrate multiple prediction systems by adding relative weights defining the impact of each separate method. Finally, when the gap of predicting T cell receptor (TCR) interactions with peptide–MHC complexes will be filled with some high accuracy methods, *NetMHCcons* can become one of the tools included in integrative approaches for T cell epitope predictions.

In the second project, I took the challenge of developing a pan-specific predictor combining all HLA class II molecules, presented in Chapter 3. In many aspects, MHC class II molecules are different from MHC class I, complicating the process of developing tools with the same accuracy as available for MHC class I. In order to develop a common method, we had to take into account the structural divergence of the HLA-DQ and HLA-DR/HLA-DP molecules, as well as differences in polymorphism across the different loci. Detailed sequence and structure analysis resulted in a pseudo sequence combing all HLA and two mouse molecules. To our knowledge, the developed *NetMHCIIpan-3.0* method is the first predictor common for all HLA class II loci. The method benefits from cross-loci data, as is characteristic for MHC class I predictors. This implies that in the future, with more experimental binding data becoming available, the method has the power to be extended to be trained on cross-species data. The results presented in Chapter 3 suggest that the trend of developing prediction methods for class II is similar to the one already observed for class I. Therefore, another possible perspective is to follow the path of evolution of MHC class I tools and develop a consensus predictor. For this, the applicability of the receptor-pocket based approach for MHC class II molecules needs to be investigated. Moreover, with stability data for MHC class II molecules being generated, one can combine MHC class II binding affinity and stability predictions to improve accuracy of the methods for identification of potential $CD4^+$ epitopes.

Finally, Chapter 5 demonstrates the importance of peptide–MHC prediction methods, like the ones presented in this thesis, for experimental epitope discovery. We used a list of T cell epitopes from yellow fever virus to show that the location of epitopes across the viral proteome is mostly defined by amino acid composition. Moreover, we demonstrated the power of the %Rank score over binding affinity for pre-selection of potential immunogenic peptides. From the results obtained in this study it is possible to extrapolate to other viruses of the same family, facilitating the design of new vaccines.

Throughout this thesis, I demonstrated the power of pan-specific methods capable of predicting peptides that bind to MHC class I and class II molecules. These methods may serve experimentalists working within the field of epitope-based vaccine discovery and development of new immunotherapies. With space left for improving the methods, I hope that this work will be an inspiration for future studies contributing to immunological research.

Bibliography

[1] Moss PA, Rosenberg WM, Bell JI (1992) The human T cell receptor in health and disease. Annu Rev Immunol 10: 71–96.

[2] Neefjes J, Jongsma ML, Paul P, Bakke O (2011) Towards a systems understanding of MHC class I and MHC class II antigen presentation. Nat Rev Immunol 11: 823–836.

[3] Andersen MH, Schrama D, Thor Straten P, Becker JC (2006) Cytotoxic T cells. J Invest Dermatol 126: 32–41.

[4] Zhu J, Paul WE (2010) Heterogeneity and plasticity of T helper cells. Cell Res 20: 4–12.

[5] Yewdell JW, Reits E, Neefjes J (2003) Making sense of mass destruction: quantitating MHC class I antigen presentation. Nat Rev Immunol 3: 952–961.

[6] Yewdell JW, Bennink JR (1999) Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses. Annu Rev Immunol 17: 51–88.

[7] Adams EJ, Luoma AM (2013) The adaptable major histocompatibility complex (MHC) fold: structure and function of nonclassical and MHC class I-like molecules. Annu Rev Immunol 31: 529–561.

[8] Bjorkman PJ, Saper MA, Samraoui B, Bennett WS, Strominger JL, et al. (1987) Structure of the human class I histocompatibility antigen, HLA-A2. Nature 329: 506–512.

[9] Robinson J, Waller MJ, Fail SC, McWilliam H, Lopez R, et al. (2009) The IMGT/HLA database. Nucleic Acids Research 37: D1013–D1017.

[10] Nielsen M, Lund O, Buus S, Lundegaard C (2010) MHC Class II epitope predictive algorithms. Immunology 130: 319–328.

[11] Jones EY, Fugger L, Strominger JL, Siebold C (2006) MHC class II proteins and disease: a structural perspective. Nature Reviews Immunology 6: 271–282.

[12] Brown JH, Jardetzky TS, Gorga JC, Stern LJ, Urban RG, et al. (1993) Three-dimensional structure of the human class II histocompatibility antigen HLA-DR1. Nature 364: 33–39.

[13] Sette A, Adorini L, Colon SM, Buus S, Grey HM (1989) Capacity of intact proteins to bind to MHC class II molecules. J Immunol 143: 1265–1267.

[14] Castellino F, Zhong G, Germain RN (1997) Antigen presentation by MHC class II molecules: invariant chain function, protein trafficking, and the molecular basis of diverse determinant capture. Hum Immunol 54: 159–169.

[15] Khan AR, Baker BM, Ghosh P, Biddison WE, Wiley DC (2000) The structure and stability of an HLA-A*0201/octameric tax peptide complex with an empty conserved peptide-N-terminal binding site. J Immunol 164: 6398–6405.

[16] Ghosh P, Amaya M, Mellins E, Wiley DC (1995) The structure of an intermediate in class II MHC maturation: CLIP bound to HLA-DR3. Nature 378: 457–462.

[17] Schrödinger, LLC (2010) The PyMOL molecular graphics system, version 1.3r1.

[18] Zhang L, Chen Y, Wong HS, Zhou S, Mamitsuka H, et al. (2012) TEPITOPEpan: Extending TEPITOPE for Peptide Binding Prediction Covering over 700 HLA-DR Molecules. PLoS ONE 7: e30483.

[19] Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25: 3389–3402.

[20] Nielsen M, Lundegaard C, Worning P, Lauemøller SL, Lamberth K, et al. (2003) Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci 12: 1007–1017.

[21] Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323: 533–536.

[22] Lund O, Nielsen M, Lundegaard C, Keşmir C, Brunak S (2005) Immunological Bioinformatics. MIT press .

[23] Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA 89: 10915–10919.

[24] Press W, Flannery B, Teukolsky S, Vetterling W (1992) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press .

[25] Swets JA (1988) Measuring the accuracy of diagnostic systems. Science 240: 1285–1293.

[26] Sette A, Vitiello A, Reherman B, Fowler P, Nayersina R, et al. (1994) The relationship between class I binding affinity and immunogenicity of potential cytotoxic T cell epitopes. J Immunol 153: 5586–5592.

[27] Lundegaard C, Lamberth K, Harndahl M, Buus S, Lund O, et al. (2008) NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8-11. Nucleic Acids Res 36: W509–512.

[28] Peters B, Bui HH, Frankild S, Nielson M, Lundegaard C, et al. (2006) A community resource benchmarking predictions of peptide binding to MHC-I molecules. PLoS Comput Biol 2: e65.

[29] Lin HH, Ray S, Tongchusak S, Reinherz EL, Brusic V (2008) Evaluation of MHC class I peptide binding prediction servers: applications for vaccine research. BMC Immunol 9: 8.

[30] Yu K, Petrovsky N, Schonbach C, Koh JY, Brusic V (2002) Methods for prediction of peptide binding to MHC molecules: a comparative study. Mol Med 8: 137–148.

[31] Zhang H, Lund O, Nielsen M (2009) The PickPocket method for predicting binding specificities for receptors based on receptor pocket similarities: application to MHC-peptide binding. Bioinformatics 25: 1293–1299.

[32] Zhang H, Lundegaard C, Nielsen M (2009) Pan-specific MHC class I predictors: a benchmark of HLA class I pan-specific prediction methods. Bioinformatics 25: 83–89.

[33] Nielsen M, Lundegaard C, Blicher T, Lamberth K, Harndahl M, et al. (2007) NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. PLoS ONE 2: e796.

[34] Robinson J, Waller MJ, Parham P, Bodmer JG, Marsh SG (2001) IMGT/HLA Database − a sequence database for the human major histocompatibility complex. Nucleic Acids Res 29: 210–213.

[35] Lundegaard C, Lund O, Buus S, Nielsen M (2010) Major histocompatibility complex class I binding predictions as a tool in epitope discovery. Immunology 130: 309–318.

[36] Hoof I, Peters B, Sidney J, Pedersen LE, Sette A, et al. (2009) NetMHCpan, a method for MHC class I binding prediction beyond humans. Immunogenetics 61: 1–13.

[37] Moutaftsi M, Peters B, Pasquetto V, Tscharke DC, Sidney J, et al. (2006) A consensus epitope prediction approach identifies the breadth of murine T(CD8+)-cell responses to vaccinia virus. Nat Biotechnol 24: 817–819.

[38] Wang P, Sidney J, Dow C, Mothe B, Sette A, et al. (2008) A systematic assessment of MHC class II peptide binding predictions and evaluation of a consensus approach. PLoS Comput Biol 4: e1000048.

[39] Wang P, Sidney J, Kim Y, Sette A, Lund O, et al. (2010) Peptide binding predictions for HLA DR, DP and DQ molecules. BMC Bioinformatics 11: 568.

[40] Vita R, Zarebski L, Greenbaum JA, Emami H, Hoof I, et al. (2010) The immune epitope database 2.0. Nucleic Acids Res 38: D854–862.

[41] Jacob L, Vert JP (2008) Efficient peptide-MHC-I binding prediction for alleles with few known binders. Bioinformatics 24: 358–366.

[42] Jojic N, Reyes-Gomez M, Heckerman D, Kadie C, Schueler-Furman O (2006) Learning MHC I–peptide binding. Bioinformatics 22: e227–235.

[43] Erup Larsen M, Kloverpris H, Stryhn A, Koofhethile CK, Sims S, et al. (2011) HLArestrictor–a tool for patient-specific predictions of HLA restriction elements and optimal epitopes within peptides. Immunogenetics 63: 43–55.

[44] Hoof I, Perez CL, Buggert M, Gustafsson RK, Nielsen M, et al. (2010) Interdisciplinary analysis of HIV-specific CD8+ T cell responses against variant epitopes reveals restricted TCR promiscuity. J Immunol 184: 5383–5391.

[45] Rapin N, Hoof I, Lund O, Nielsen M (2010) The MHC motif viewer: a visualization tool for MHC binding motifs. Curr Protoc Immunol Chapter 18: Unit 18.17.

[46] Stranzl T, Larsen MV, Lundegaard C, Nielsen M (2010) NetCTLpan: pan-specific MHC class I pathway epitope predictions. Immunogenetics 62: 357–368.

[47] Bui HH, Sidney J, Peters B, Sathiamurthy M, Sinichi A, et al. (2005) Automated generation and evaluation of specific MHC binding predictive tools: ARB matrix applications. Immunogenetics 57: 304–314.

[48] Peters B, Sette A (2005) Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. BMC Bioinformatics 6: 132.

[49] Nielsen M, Lundegaard C, Blicher T, Peters B, Sette A, et al. (2008) Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan. PLoS Comput Biol 4: e1000107.

[50] Nielsen M, Justesen S, Lund O, Lundegaard C, Buus S (2010) NetMHCIIpan-2.0 - Improved pan-specific HLA-DR predictions using a novel concurrent alignment and weight optimization training procedure. Immunome Res 6: 9.

[51] Zaitlen N, Reyes-Gomez M, Heckerman D, Jojic N (2008) Shift-invariant adaptive double threading: learning MHC II-peptide binding. J Comput Biol 15: 927–942.

[52] Zhang GL, Khan AM, Srinivasan KN, August JT, Brusic V (2005) MULTIPRED: a computational system for prediction of promiscuous HLA binding peptides. Nucleic Acids Res 33: W172–179.

[53] Sturniolo T, Bono E, Ding J, Raddrizzani L, Tuereci O, et al. (1999) Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. Nat Biotechnol 17: 555–561.

[54] Dai S, Murphy GA, Crawford F, Mack DG, Falta MT, et al. (2010) Crystal structure of HLA-DP2 and implications for chronic beryllium disease. Proc Natl Acad Sci USA 107: 7425–7430.

[55] Lee KH, Wucherpfennig KW, Wiley DC (2001) Structure of a human insulin peptide-HLA-DQ8 complex and susceptibility to type 1 diabetes. Nat Immunol 2: 501–507.

[56] Kim CY, Quarsten H, Bergseng E, Khosla C, Sollid LM (2004) Structural basis for HLA-DQ2-mediated presentation of gluten epitopes in celiac disease. Proc Natl Acad Sci USA 101: 4175–4179.

[57] Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, et al. (2007) Clustal W and Clustal X version 2.0. Bioinformatics 23: 2947–2948.

[58] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. (2000) The Protein Data Bank. Nucleic Acids Res 28: 235–242.

[59] Nielsen M, Lundegaard C, Lund O (2007) Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. BMC Bioinformatics 8: 238.

[60] Karosiene E, Lundegaard C, Lund O, Nielsen M (2011) NetMHCcons: a consensus method for the major histocompatibility complex class I predictions. Immunogenetics 64: 177–186.

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

[62] Gonzalez-Galarza FF, Christmas S, Middleton D, Jones AR (2011) Allele frequency net: a database and online repository for immune gene frequencies in worldwide populations. Nucleic Acids Res 39: D913–919.

[63] Hobohm U, Scharf M, Schneider R, Sander C (1992) Selection of representative protein data sets. Protein Sci 1: 409–417.

[64] Thomsen M, Lundegaard C, Buus S, Lund O, Nielsen M (2013) MHCcluster, a method for functional clustering of MHC molecules. Immunogenetics 65: 655–665.

[65] Thomsen MC, Nielsen M (2012) 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 40: W281–287.

[66] Nielsen M, Lund O (2009) NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. BMC Bioinformatics 10: 296.

[67] Kullback S, Leibler R (1951) On information and sufficiency. Ann Math Stat 22(1): 142–143.

[68] Andreatta M, Nielsen M (2012) Characterizing the binding motifs of 11 common human HLA-DP and HLA-DQ molecules using NNAlign. Immunology 136: 306–311.

[69] Lund O, Nielsen M, Kesmir C, Petersen AG, Lundegaard C, et al. (2004) Definition of supertypes for HLA molecules using clustering of specificity matrices. Immunogenetics 55: 797–810.

[70] Sette A, Sidney J (1999) Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism. Immunogenetics 50: 201–212.

[71] Nene V, Svitek N, Toye P, Golde WT, Barlow J, et al. (2012) Designing bovine T cell vaccines via reverse immunology. Ticks Tick Borne Dis 3: 188–192.

[72] Parker KC, Bednarek MA, Coligan JE (1994) Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. J Immunol 152: 163–175.

[73] Rammensee H, Bachmann J, Emmerich NP, Bachor OA, Stevanović S (1999) SYFPEITHI: database for MHC ligands and peptide motifs. Immunogenetics 50: 213–219.

[74] Zhang GL, Ansari HR, Bradley P, Cawley GC, Hertz T, et al. (2011) Machine learning competition in immunology - Prediction of HLA class I binding peptides. J Immunol Methods 374: 1–4.

[75] Zhang L, Udaka K, Mamitsuka H, Zhu S (2012) Toward more accurate pan-specific MHC-peptide binding prediction: a review of current methods and tools. Brief Bioinformatics 13: 350–364.

[76] Lundegaard C, Hoof I, Lund O, Nielsen M (2010) State of the art and challenges in sequence based T-cell epitope prediction. Immunome Res 6 Suppl 2: S3.

[77] Godkin AJ, Smith KJ, Willis A, Tejada-Simon MV, Zhang J, et al. (2001) Naturally processed HLA class II peptides reveal highly conserved immunogenic flanking region sequence preferences that reflect antigen processing rather than peptide-MHC interactions. J Immunol 166: 6720–6727.

[78] Lovitch SB, Pu Z, Unanue ER (2006) Amino-terminal flanking residues determine the conformation of a peptide-class II MHC complex. J Immunol 176: 2958–2968.

[79] Bordner AJ, Mittelmann HD (2010) MultiRTA: a simple yet reliable method for predicting peptide binding affinities for multiple class II MHC allotypes. BMC Bioinformatics 11: 482.

[80] Zhang GL, DeLuca DS, Keskin DB, Chitkushev L, Zlateva T, et al. (2011) MULTIPRED2: a computational system for large-scale identification of peptides predicted to bind to HLA supertypes and alleles. J Immunol Methods 374: 53–61.

[81] Rao X, Costa AI, van Baarle D, Kesmir C (2009) A comparative study of HLA binding affinity and ligand diversity: implications for generating immunodominant CD8+ T cell responses. J Immunol 182: 1526–1532.

[82] Larsen MV, Lelic A, Parsons R, Nielsen M, Hoof I, et al. (2010) Identification of CD8+ T Cell Epitopes in the West Nile Virus Polyprotein by Reverse-Immunology Using NetCTL. PLoS ONE 5: e12697.

[83] Nielsen M, Lundegaard C, Lund O, Keşmir C (2005) The role of the proteasome in generating cytotoxic T-cell epitopes: insights obtained from improved predictions of proteasomal cleavage. Immunogenetics 57: 33–41.

[84] Larsen MV, Lundegaard C, Lamberth K, Buus S, Brunak S, et al. (2005) An integrative approach to CTL epitope prediction: a combined algorithm integrating MHC class I binding, TAP transport efficiency, and proteasomal cleavage predictions. Eur J Immunol 35: 2295–2303.

[85] Peters B, Bulik S, Tampe R, Van Endert PM, Holzhutter HG (2003) Identifying MHC class I epitopes by predicting the TAP transport efficiency of epitope precursors. J Immunol 171: 1741–1749.

[86] Chambers TJ, Hahn CS, Galler R, Rice CM (1990) Flavivirus Genome Organization, Expression, and Replication 44: 649–688.

[87] Gould EA, Solomon T (2008) Pathogenic flaviviruses. The Lancet 371: 500–509.

[88] Gardner CL, Ryman KD (2010) Yellow fever: a reemerging threat. Clin Lab Med 30: 237–260.

[89] Monath TP (2005) Yellow fever vaccine. Expert Rev Vaccines 4: 553–574.

[90] Pugachev KV, Guirakhoo F, Monath TP (2005) New developments in flavivirus vaccines with special attention to yellow fever. Curr Opin Infect Dis 18: 387–394.

[91] Xie H, Cass AR, Barrett AD (1998) Yellow fever 17D vaccine virus isolated from healthy vaccinees accumulates very few mutations. Virus Res 55: 93–99.

[92] Bell JR, Kinney RM, Trent DW, Lenches EM, Dalgarno L, et al. (1985) Amino-terminal amino acid sequences of structural proteins of three flaviviruses. Virology 143: 224–229.

[93] Boege U, Heinz FX, Wengler G, Kunz C (1983) Amino acid compositions and amino-terminal sequences of the structural proteins of a flavivirus, European Tick-Borne Encephalitis virus. Virology 126: 651–657.

[94] Castle E, Nowak T, Leidner U, Wengler G, Wengler G (1985) Sequence analysis of the viral core protein and the membrane-associated proteins V1 and NV2 of the flavivirus West Nile virus and of the genome sequence for these proteins. Virology 145: 227–236.

[95] Speight G, Coia G, Parker MD, Westaway EG (1988) Gene mapping and positive identification of the non-structural proteins NS2A, NS2B, NS3, NS4B and NS5 of the flavivirus Kunjin and their cleavage sites. J Gen Virol 69 (Pt 1): 23–34.

[96] Rice CM, Lenches EM, Eddy SR, Shin SJ, Sheets RL, et al. (1985) Nucleotide sequence of yellow fever virus: implications for flavivirus gene expression and evolution. Science 229: 726–733.

[97] Pastorino B, Nougairède A, Wurtz N, Gould E, de Lamballerie X (2010) Role of host cell factors in flavivirus infection: Implications for pathogenesis and development of antiviral drugs. Antiviral Research 87: 281–294.

[98] Bollati M, Alvarez K, Assenberg R, Baronti C, Canard B, et al. (2010) Structure and functionality in flavivirus NS-proteins: Perspectives for drug design. Antiviral Research 87: 125–148.

[99] Meddows-Taylor S, Shalekoff S, Kuhn L, Gray GE, Tiemessen CT (2007) Development of a whole blood intracellular cytokine staining assay for mapping CD4+ and CD8+ T-cell responses across the HIV-1 genome. Journal of virological methods 144: 115–121.

[100] Anthony D (2003) T-cell epitope mapping using the ELISPOT approach. Methods 29: 260–269.

[101] Altman JD, Moss PA, Goulder PJ, Barouch DH, McHeyzer-Williams MG, et al. (2011) Phenotypic analysis of antigen-specific T lymphocytes. Science. 1996. 274: 94-96. J Immunol 187: 7--9.

[102] Lanteri MC, Heitman JW, Owen RE, Busch T, Gefter N, et al. (2008) Comprehensive analysis of west nile virus-specific T cell responses in humans. J Infect Dis 197: 1296–1306.

[103] Parsons R, Lelic A, Hayes L, Carter A, Marshall L, et al. (2008) The memory T cell response to West Nile virus in symptomatic humans following natural infection is not influenced by age and is dominated by a restricted set of CD8+ T cell epitopes. J Immunol 181: 1563–1572.

[104] Harndahl M, Rasmussen M, Roder G, Dalgaard Pedersen I, Sørensen M, et al. (2012) Peptide-MHC class I stability is a better predictor than peptide affinity of CTL immunogenicity. Eur J Immunol 42: 1405–1416.

[105] Jørgensen KW, Rasmussen M, Buus S, Nielsen M (2013) NetMHCstab - predicting stability of peptide:MHC-I complexes; impacts for CTL epitope discovery. Immunology .

Appendix A

Supplementary material for Paper I, Chapter 2

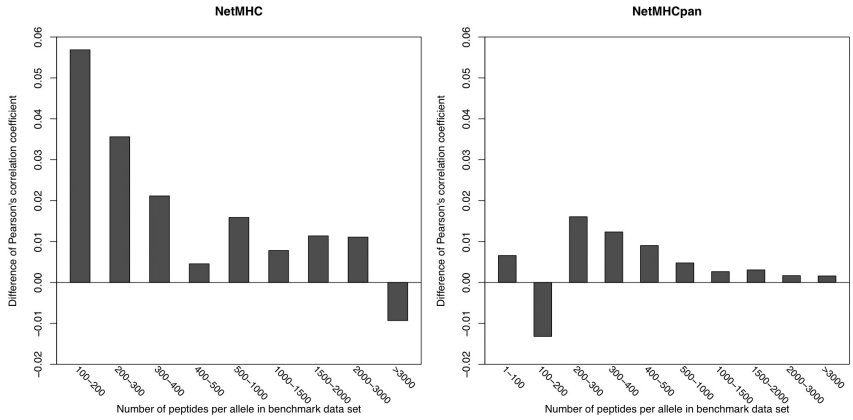

Figure A.1. Difference of performance between two evaluation strategies depending on the number of peptides per allele in training set. The performance difference in terms of Pearson's correlation coefficient was calculated as follows: *Difference = evaluation on test sets − evaluation on independent set*. Positive difference of PCC corresponds to lower performance on independent set, while negative difference refers to higher performance when using independent evaluation set.

Table A.1. List of alleles composing the benchmark data set used for the analysis. # *data points* indicates number of peptide binding measurements available for that allele and # *binders* indicates the number of actual binders from all the peptides.

	Allele	**# data points**	**# binders**
1	Gogo-B*0101	14	5
2	H-2-Db	1,496	480
3	H-2-Dd	201	13
4	H-2-Kb	1,366	349
5	H-2-Kd	343	146
6	H-2-Kk	168	79
7	H-2-Ld	147	34
8	HLA-A*0101	3,263	433
9	HLA-A*0201	7,064	2,281
10	HLA-A*0202	2,314	1,072
11	HLA-A*0203	3,937	1,278
12	HLA-A*0205	36	31
13	HLA-A*0206	3,223	1,266
14	HLA-A*0207	30	7
15	HLA-A*0210	18	0
16	HLA-A*0211	1,038	361
17	HLA-A*0212	1,143	275
18	HLA-A*0216	894	160
19	HLA-A*0219	1,203	204

	Allele	# data points	# binders
20	HLA-A*0250	132	88
21	HLA-A*0301	4,708	1,016
22	HLA-A*0302	2	2
23	HLA-A*1101	3,891	1,157
24	HLA-A*2301	1,513	291
25	HLA-A*2402	2,065	367
26	HLA-A*2403	1,216	287
27	HLA-A*2501	519	66
28	HLA-A*2601	2,457	297
29	HLA-A*2602	202	67
30	HLA-A*2603	205	25
31	HLA-A*2902	1,839	470
32	HLA-A*3001	1,949	569
33	HLA-A*3002	912	234
34	HLA-A*3101	3,309	681
35	HLA-A*3201	575	275
36	HLA-A*3301	1,616	224
37	HLA-A*6601	4	4
38	HLA-A*6801	1,700	641
39	HLA-A*6802	3,188	643
40	HLA-A*6901	2,079	221
41	HLA-A*8001	782	113
42	HLA-B*0702	3,049	617
43	HLA-B*0801	2,151	490
44	HLA-B*0802	486	18
45	HLA-B*0803	217	9
46	HLA-B*1402	3	0
47	HLA-B*1501	3,290	908
48	HLA-B*1502	164	124
49	HLA-B*1503	416	332
50	HLA-B*1509	346	16
51	HLA-B*1517	846	271
52	HLA-B*1801	1,756	222
53	HLA-B*2701	1	1
54	HLA-B*2702	4	0
55	HLA-B*2703	433	0
56	HLA-B*2704	2	0
57	HLA-B*2705	2,389	394
58	HLA-B*3501	1,993	505
59	HLA-B*3503	5	1

	Allele	# data points	# binders
60	HLA-B*3801	136	3
61	HLA-B*3901	957	233
62	HLA-B*4001	2,486	338
63	HLA-B*4002	568	192
64	HLA-B*4201	2	2
65	HLA-B*4402	1,390	138
66	HLA-B*4403	595	124
67	HLA-B*4501	578	143
68	HLA-B*4601	1,411	91
69	HLA-B*4801	861	68
70	HLA-B*5101	1,336	170
71	HLA-B*5301	620	211
72	HLA-B*5401	621	139
73	HLA-B*5701	1,719	214
74	HLA-B*5801	2,529	450
75	HLA-B*5802	31	9
76	HLA-B*7301	115	14
77	HLA-C*0602	6	3
78	HLA-E*0101	3	2
79	Mamu-A1*00101	823	463
80	Mamu-A1*00201	355	205
81	Mamu-A1*00701	33	26
82	Mamu-A1*01101	491	188
83	Mamu-A1*02201	247	49
84	Mamu-B*00101	237	72
85	Mamu-B*00301	372	117
86	Mamu-B*00401	1	1
87	Mamu-B*00801	368	125
88	Mamu-B*01701	678	269
89	Mamu-B*04801	60	40
90	Mamu-B*05201	60	40
91	Patr-A*0101	203	50
92	Patr-A*0301	169	24
93	Patr-A*0401	144	37
94	Patr-A*0602	1	1
95	Patr-A*0701	287	66
96	Patr-A*0901	173	71
97	Patr-B*0101	454	112
98	Patr-B*0901	1	1
99	Patr-B*1301	97	69
100	Patr-B*1701	5	2
101	Patr-B*2401	193	62

Table A.2. List of alleles from the benchmark data set for which at least 50 data points were available and at least 10 of them were binding peptides. *# data points* indicates number of peptide binding measurements available for that allele and *# binders* indicates the number of actual binders from all the peptides.

	Allele	**# data points**	**# binders**
1	H-2-Db	1,496	480
2	H-2-Dd	201	13
3	H-2-Kb	1,366	349
4	H-2-Kd	343	146
5	H-2-Kk	168	79
6	H-2-Ld	147	34
7	HLA-A*0101	3,263	433
8	HLA-A*0201	7,064	2,281
9	HLA-A*0202	2,314	1,072
10	HLA-A*0203	3,937	1,278
11	HLA-A*0206	3,223	1,266
12	HLA-A*0211	1,038	361
13	HLA-A*0212	1,143	275
14	HLA-A*0216	894	160
15	HLA-A*0219	1,203	204
16	HLA-A*0250	132	88
17	HLA-A*0301	4,708	1,016
18	HLA-A*1101	3,891	1,157
19	HLA-A*2301	1,513	291
20	HLA-A*2402	2,065	367
21	HLA-A*2403	1,216	287
22	HLA-A*2501	519	66
23	HLA-A*2601	2,457	297
24	HLA-A*2602	202	67
25	HLA-A*2603	205	25
26	HLA-A*2902	1,839	470
27	HLA-A*3001	1,949	569
28	HLA-A*3002	912	234
29	HLA-A*3101	3,309	681
30	HLA-A*3201	575	275
31	HLA-A*3301	1,616	224
32	HLA-A*6801	1,700	641
33	HLA-A*6802	3,188	643
34	HLA-A*6901	2,079	221
35	HLA-A*8001	782	113
36	HLA-B*0702	3,049	617
37	HLA-B*0801	2,151	490

	Allele	# data points	# binders
38	HLA-B*0802	486	18
39	HLA-B*1501	3,290	908
40	HLA-B*1502	164	124
41	HLA-B*1503	416	332
42	HLA-B*1509	346	16
43	HLA-B*1517	846	271
44	HLA-B*1801	1,756	222
45	HLA-B*2705	2,389	394
46	HLA-B*3501	1,993	505
47	HLA-B*3901	957	233
48	HLA-B*4001	2,486	338
49	HLA-B*4002	568	192
50	HLA-B*4402	1,390	138
51	HLA-B*4403	595	124
52	HLA-B*4501	578	143
53	HLA-B*4601	1,411	91
54	HLA-B*4801	861	68
55	HLA-B*5101	1,336	170
56	HLA-B*5301	620	211
57	HLA-B*5401	621	139
58	HLA-B*5701	1,719	214
59	HLA-B*5801	2,529	450
60	HLA-B*7301	115	14
61	Mamu-A1*00101	823	463
62	Mamu-A1*00201	355	205
63	Mamu-A1*01101	491	188
64	Mamu-A1*02201	247	49
65	Mamu-B*00101	237	72
66	Mamu-B*00301	372	117
67	Mamu-B*00801	368	125
68	Mamu-B*01701	678	269
69	Mamu-B*04801	60	40
70	Mamu-B*05201	60	40
71	Patr-A*0101	203	50
72	Patr-A*0301	169	24
73	Patr-A*0401	144	37
74	Patr-A*0701	287	66
75	Patr-A*0901	173	71
76	Patr-B*0101	454	112
77	Patr-B*1301	97	69
78	Patr-B*2401	193	62

Table A.3. List of alleles used to validate final consensus method. Alleles in *bold* are common between training set and validation set. # *data points* indicates number of peptide binding measurements available for that allele in the validation set and # *binders* indicates the number of actual binders from all the peptides.

	Allele	# data points	# binders
1	BoLA-N*01301	93	88
2	BoLA-N*05201	90	84
3	**HLA-A*0101**	**242**	**44**
4	**HLA-A*0201**	**643**	**210**
5	**HLA-A*0203**	**43**	**32**
6	**HLA-A*0206**	**32**	**30**
7	**HLA-A*0211**	**44**	**40**
8	**HLA-A*0212**	**38**	**31**
9	**HLA-A*0216**	**24**	**18**
10	**HLA-A*0219**	**40**	**27**
11	**HLA-A*0301**	**394**	**157**
12	HLA-A*0319	30	14
13	**HLA-A*1101**	**189**	**18**
14	**HLA-A*2301**	**144**	**30**
15	**HLA-A*2402**	**11**	**2**
16	**HLA-A*2403**	**157**	**43**
17	**HLA-A*2501**	**416**	**5**
18	**HLA-A*2601**	**1,080**	**62**
19	**HLA-A*2602**	**213**	**67**
20	**HLA-A*2603**	**229**	**24**
21	**HLA-A*2902**	**169**	**59**
22	**HLA-A*3001**	**201**	**16**
23	**HLA-A*3002**	**165**	**28**
24	**HLA-A*3101**	**133**	**86**
25	HLA-A*3207	87	78
26	HLA-A*3215	74	59
27	**HLA-A*6601**	**173**	**7**
28	**HLA-A*6802**	**14**	**2**
29	HLA-A*6823	81	76
30	**HLA-A*6901**	**393**	**13**
31	**HLA-A*8001**	**389**	**9**

	Allele	# data points	# binders
32	**HLA-B*0702**	**430**	**229**
33	**HLA-B*0801**	**614**	**82**
34	**HLA-B*0802**	**514**	**18**
35	**HLA-B*1402**	**184**	**16**
36	**HLA-B*1501**	**415**	**57**
37	**HLA-B*1509**	**369**	**16**
38	**HLA-B*1517**	**329**	**12**
39	HLA-B*1542	361	3
40	**HLA-B*1801**	**503**	**15**
41	**HLA-B*2705**	**200**	**26**
42	HLA-B*2720	91	89
43	**HLA-B*3501**	**16**	**10**
44	**HLA-B*3801**	**142**	**3**
45	**HLA-B*3901**	**814**	**68**
46	**HLA-B*4001**	**189**	**32**
47	HLA-B*4013	58	52
48	HLA-B*4506	359	4
49	**HLA-B*4601**	**385**	**2**
50	**HLA-B*5101**	**572**	**4**
51	**HLA-B*5301**	**179**	**5**
52	**HLA-B*5701**	**506**	**12**
53	**HLA-B*5801**	**196**	**31**
54	**HLA-B*7301**	**14**	**3**
55	HLA-B*8301	336	40
56	HLA-C*0401	364	5
57	HLA-C*0501	172	68
58	**HLA-C*0602**	**220**	**88**
59	HLA-C*1402	170	141
60	HLA-C*1502	82	33
61	**HLA-E*0101**	**93**	**12**
62	SLA-1*0401	15	14

Table A.4. Benchmark results for different methods and their combinations when allele in question is part of the training data set. The results are given as Pearson's correlation coefficients (PCC) for 3 analysed methods: *NetMHC* (indicated as *MHC* in this table), *NetMHCpan* (*Pan*), *PickPocket* (*Pick*), and for their possible combinations, expressed as simple averages: *NetMHC+NetMHCpan* (*MHC+Pan*), *NetMHC+PickPocket* (*MHC+Pick*), *NetMHCpan+PickPocket* (*Pan+Pick*) and *NetMHC+NetMHCpan+PickPocket* (*MHC+Pan+Pick*). *#pep* indicates number of peptide binding measurements available for that allele and *#bind* indicates the number of actual binders from all the peptides.

Allele	#pep	#bind	MHC	Pan	Pick	MHC+ Pan	MHC+ Pick	Pan+ Pick	MHC+ Pan+ Pick
Gogo-B*0101	14	5	0.4743	0.1237	-0.0435	0.3841	0.3437	0.0403	0.2957
H-2-Db	1,496	480	0.8676	0.8563	0.7020	0.8700	0.8439	0.8277	0.8567
H-2-Dd	201	13	0.5327	0.2671	0.1408	0.4985	0.3871	0.2117	0.3901
H-2-Kb	1,366	349	0.7146	0.7044	0.5700	0.7232	0.6877	0.6776	0.7064
H-2-Kd	343	146	0.7802	0.7776	0.7243	0.8029	0.7813	0.7888	0.8012
H-2-Kk	168	79	0.5106	0.6617	0.6503	0.6371	0.6040	0.6939	0.6607
H-2-Ld	147	34	0.8725	0.8066	0.7612	0.8639	0.8578	0.8178	0.8578
HLA-A*0101	3,263	433	0.8426	0.8261	0.6138	0.8430	0.8106	0.7827	0.8245
HLA-A*0201	7,064	2,281	0.8766	0.8769	0.7811	0.8808	0.8569	0.8553	0.8696
HLA-A*0202	2,314	1,072	0.8421	0.8496	0.7726	0.8537	0.8338	0.8341	0.8457
HLA-A*0203	3,937	1,278	0.8705	0.8740	0.7837	0.8775	0.8548	0.8531	0.8669
HLA-A*0205	36	31	0.5930	0.9512	0.8274	0.8177	0.7426	0.9215	0.8468
HLA-A*0206	3,223	1,266	0.8143	0.8156	0.7023	0.8241	0.7949	0.7870	0.8092
HLA-A*0207	30	7	0.7131	0.8014	0.7579	0.8210	0.8101	0.8002	0.8251
HLA-A*0210	18	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HLA-A*0211	1,038	361	0.8516	0.8735	0.8056	0.8745	0.8528	0.8639	0.8698
HLA-A*0212	1,143	275	0.8798	0.8918	0.7712	0.8970	0.8637	0.8642	0.8829
HLA-A*0216	894	160	0.7885	0.8582	0.7106	0.8449	0.7896	0.8249	0.8309
HLA-A*0219	1,203	204	0.8411	0.8659	0.7202	0.8688	0.8216	0.8290	0.8498
HLA-A*0250	132	88	0.8590	0.9312	0.8962	0.9195	0.8924	0.9272	0.9210
HLA-A*0301	4,708	1,016	0.8176	0.8158	0.6529	0.8237	0.7962	0.7833	0.8100
HLA-A*0302	2	2	-1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
HLA-A*1101	3,891	1,157	0.8703	0.8702	0.7323	0.8762	0.8546	0.8479	0.8667
HLA-A*2301	1,513	291	0.7343	0.7537	0.7072	0.7569	0.7413	0.7510	0.7556
HLA-A*2402	2,065	367	0.7375	0.7518	0.6726	0.7544	0.7314	0.7372	0.7474
HLA-A*2403	1,216	287	0.9030	0.9025	0.8073	0.9124	0.8909	0.8836	0.9020
HLA-A*2501	519	66	0.7905	0.8187	0.6529	0.8346	0.7786	0.7820	0.8132
HLA-A*2601	2,457	297	0.8083	0.8151	0.6312	0.8241	0.7842	0.7772	0.8067
HLA-A*2602	202	67	0.9120	0.9302	0.8555	0.9379	0.9164	0.9225	0.9337
HLA-A*2603	205	25	0.6932	0.8602	0.6489	0.8399	0.7289	0.8192	0.8201
HLA-A*2902	1,839	470	0.7652	0.7740	0.6264	0.7795	0.7548	0.7544	0.7717
HLA-A*3001	1,949	569	0.8522	0.8569	0.7306	0.8629	0.8420	0.8389	0.8551
HLA-A*3002	912	234	0.7151	0.7219	0.6320	0.7360	0.7196	0.7204	0.7348
HLA-A*3101	3,309	681	0.8389	0.8398	0.6989	0.8465	0.8199	0.8146	0.8352
HLA-A*3201	575	275	0.7703	0.7817	0.7267	0.7957	0.7785	0.7842	0.7949
HLA-A*3301	1,616	224	0.7447	0.7481	0.5803	0.7625	0.7290	0.7194	0.7503
HLA-A*6601	4	4	0.2057	0.5662	0.2458	0.4402	0.2278	0.4467	0.3852
HLA-A*6801	1,700	641	0.8124	0.8223	0.7267	0.8265	0.8087	0.8117	0.8222
HLA-A*6802	3,188	643	0.8152	0.8147	0.6867	0.8246	0.8009	0.7914	0.8140
HLA-A*6901	2,079	221	0.8126	0.8101	0.6161	0.8307	0.7795	0.7616	0.8046
HLA-A*8001	782	113	0.8312	0.8348	0.7043	0.8558	0.8228	0.8100	0.8417
HLA-B*0702	3,049	617	0.8615	0.8576	0.7331	0.8677	0.8398	0.8282	0.8532
HLA-B*0801	2,151	490	0.7254	0.7710	0.5763	0.7655	0.7233	0.7391	0.7553
HLA-B*0802	486	18	0.8244	0.8171	0.5230	0.8568	0.7462	0.7327	0.8010
HLA-B*0803	217	9	0.4973	0.7338	0.5433	0.6866	0.5801	0.6775	0.6694
HLA-B*1402	3	0	-0.9997	0.0992	-0.1010	-0.2592	-0.2040	-0.0562	-0.1380
HLA-B*1501	3,290	908	0.7640	0.7586	0.6748	0.7695	0.7552	0.7454	0.7628
HLA-B*1502	164	124	0.4947	0.6065	0.4188	0.6049	0.5143	0.5768	0.5897
HLA-B*1503	416	332	0.7558	0.7904	0.7377	0.7942	0.7722	0.7870	0.7933
HLA-B*1509	346	16	0.6231	0.6184	0.4758	0.6685	0.6103	0.5862	0.6411
HLA-B*1517	846	271	0.8696	0.8825	0.7852	0.8879	0.8646	0.8647	0.8795

Allele	#pep	#bind	MHC	Pan	Pick	MHC+ Pan	MHC+ Pick	Pan+ Pick	MHC+ Pan+ Pick
HLA-B*1801	1,756	222	0.7798	0.7893	0.6487	0.7981	0.7602	0.7528	0.7802
HLA-B*2701	1	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HLA-B*2702	4	0	-0.8971	-0.0167	0.8745	-0.0567	0.8294	0.3010	0.2673
HLA-B*2703	433	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HLA-B*2704	2	0	-1.0000	1.0000	-1.0000	1.0000	-1.0000	1.0000	1.0000
HLA-B*2705	2,389	394	0.8772	0.8703	0.7663	0.8813	0.8611	0.8515	0.8716
HLA-B*3501	1,993	505	0.8223	0.8173	0.7083	0.8288	0.8032	0.7950	0.8162
HLA-B*3503	5	1	0.2988	0.9701	0.8264	0.9394	0.7875	0.9317	0.9132
HLA-B*3801	136	3	0.3427	0.4183	0.4136	0.4859	0.4388	0.4615	0.4804
HLA-B*3901	957	233	0.8342	0.8290	0.7030	0.8489	0.8223	0.8081	0.8368
HLA-B*4001	2,486	338	0.8789	0.8684	0.7318	0.8814	0.8507	0.8362	0.8643
HLA-B*4002	568	192	0.8095	0.8218	0.7392	0.8363	0.8093	0.8090	0.8273
HLA-B*4201	2	2	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
HLA-B*4402	1,390	138	0.7032	0.7125	0.6266	0.7227	0.6908	0.6898	0.7081
HLA-B*4403	595	124	0.7755	0.8030	0.7464	0.8073	0.7905	0.7973	0.8062
HLA-B*4501	578	143	0.8635	0.8258	0.7756	0.8670	0.8573	0.8219	0.8584
HLA-B*4601	1,411	91	0.7553	0.7373	0.5378	0.7731	0.7115	0.6786	0.7365
HLA-B*4801	861	68	0.8222	0.8439	0.6769	0.8584	0.8021	0.8063	0.8359
HLA-B*5101	1,336	170	0.7016	0.7372	0.6484	0.7346	0.7014	0.7152	0.7246
HLA-B*5301	620	211	0.7750	0.7917	0.7125	0.7984	0.7798	0.7833	0.7952
HLA-B*5401	621	139	0.8320	0.8310	0.7231	0.8497	0.8202	0.8108	0.8373
HLA-B*5701	1,719	214	0.8581	0.8492	0.7299	0.8668	0.8331	0.8221	0.8495
HLA-B*5801	2,529	450	0.8675	0.8668	0.7464	0.8753	0.8486	0.8383	0.8613
HLA-B*5802	31	9	0.4944	0.5033	0.3114	0.5875	0.4966	0.4423	0.5384
HLA-B*7301	115	14	0.5520	0.4858	0.5583	0.5782	0.5984	0.5624	0.6001
HLA-C*0602	6	3	0.0049	-0.6360	-0.0269	-0.2944	-0.0064	-0.3993	-0.3605
HLA-E*0101	3	2	-0.6663	-0.6220	-0.7575	-0.6502	-0.7075	-0.7020	-0.6870
Mamu-A1*00101	823	463	0.7999	0.7981	0.7219	0.8139	0.7925	0.7949	0.8084
Mamu-A1*00201	355	205	0.7727	0.7738	0.7333	0.7978	0.7763	0.7870	0.7963
Mamu-A1*00701	33	26	0.4997	0.3649	0.3619	0.4995	0.5117	0.3893	0.4970
Mamu-A1*01101	491	188	0.7462	0.7813	0.6740	0.7798	0.7407	0.7675	0.7715
Mamu-A1*02201	247	49	0.6508	0.7539	0.6231	0.7390	0.6721	0.7265	0.7261
Mamu-B*00101	237	72	0.9176	0.9074	0.8168	0.9242	0.9015	0.8940	0.9137
Mamu-B*00301	372	117	0.8206	0.8428	0.7575	0.8472	0.8224	0.8347	0.8433
Mamu-B*00401	1	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mamu-B*00801	368	125	0.8346	0.8647	0.7691	0.8657	0.8357	0.8549	0.8609
Mamu-B*01701	678	269	0.8224	0.7922	0.7328	0.8314	0.8169	0.8056	0.8292
Mamu-B*04801	60	40	0.8159	0.9111	0.8306	0.8842	0.8390	0.8874	0.8770
Mamu-B*05201	60	40	0.7853	0.8735	0.7787	0.8509	0.8052	0.8556	0.8459
Patr-A*0101	203	50	0.7478	0.6178	0.6989	0.7461	0.7729	0.6741	0.7518
Patr-A*0301	169	24	0.6360	0.7049	0.6163	0.7665	0.6976	0.7133	0.7559
Patr-A*0401	144	37	0.6981	0.8048	0.7525	0.8059	0.7635	0.8166	0.8133
Patr-A*0602	1	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Patr-A*0701	287	66	0.5604	0.6278	0.5569	0.6248	0.5827	0.6292	0.6231
Patr-A*0901	173	71	0.6173	0.6471	0.6543	0.6852	0.6567	0.6770	0.6915
Patr-B*0101	454	112	0.7790	0.8573	0.7635	0.8401	0.7992	0.8457	0.8378
Patr-B*0901	1	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Patr-B*1301	97	69	0.6667	0.7906	0.7219	0.7598	0.7096	0.7825	0.7609
Patr-B*1701	5	2	0.5076	0.9546	0.9774	0.7222	0.7988	0.9961	0.8757
Patr-B*2401	193	62	0.8563	0.8167	0.4979	0.8600	0.8296	0.7903	0.8473
Average			**0.5940**	**0.6754**	**0.5724**	**0.6864**	**0.6459**	**0.6631**	**0.6809**

Table A.5. Benchmark results for pan-specific methods and their combination, representing the situation when alleles in question are not part of the training data set. The results are given as Pearson's correlation coefficients (PCC) for *NetMHCpan* (indicated as *Pan* in this table), *PickPocket* (*Pick*), and for their combination, expressed as a simple average *NetMHCpan+PickPocket* (*Pan+Pick*). *dist* indicates the distance, as measured in terms of the MHC pseudo sequence similarity, from the query allele to the nearest neighbour from the training set.

Allele	dist	Pan	Pick	Pan+Pick
H-2-Db	0.260	0.3645	0.3125	0.3849
H-2-Dd	0.291	-0.0857	-0.0462	-0.0630
H-2-Kb	0.291	0.2000	0.3240	0.2745
H-2-Kd	0.390	0.1697	0.5665	0.5216
H-2-Kk	0.376	0.4095	0.6276	0.5816
H-2-Ld	0.260	0.1958	0.1727	0.1956
HLA-A*0101	0.193	0.5320	0.4131	0.5074
HLA-A*0201	0.017	0.8555	0.7836	0.8421
HLA-A*0202	0.010	0.8136	0.7681	0.8129
HLA-A*0203	0.036	0.8366	0.7849	0.8290
HLA-A*0206	0.017	0.7471	0.6722	0.7334
HLA-A*0211	0.068	0.8628	0.7953	0.8547
HLA-A*0212	0.032	0.8831	0.7754	0.8592
HLA-A*0216	0.030	0.8493	0.7075	0.8148
HLA-A*0219	0.053	0.8311	0.7233	0.8020
HLA-A*0250	0.010	0.9061	0.8982	0.9137
HLA-A*0301	0.112	0.7433	0.5916	0.7058
HLA-A*1101	0.076	0.7917	0.7007	0.7753
HLA-A*2301	0.034	0.7319	0.7021	0.7392
HLA-A*2402	0.034	0.7136	0.6667	0.7116
HLA-A*2403	0.054	0.8416	0.7809	0.8429
HLA-A*2501	0.099	0.7406	0.6207	0.7192
HLA-A*2601	0.025	0.7372	0.6069	0.7247
HLA-A*2602	0.025	0.9354	0.8338	0.9279
HLA-A*2603	0.083	0.8384	0.5877	0.7850
HLA-A*2902	0.181	0.5078	0.5098	0.5363
HLA-A*3001	0.148	0.5027	0.4454	0.5022
HLA-A*3002	0.148	0.1671	0.3695	0.2626
HLA-A*3101	0.078	0.7208	0.0000	0.7208
HLA-A*3201	0.182	0.3247	0.5933	0.5417
HLA-A*3301	0.078	0.6731	0.5084	0.6422
HLA-A*6801	0.109	0.6147	0.6919	0.7069
HLA-A*6802	0.052	0.7253	0.6593	0.7280
HLA-A*6901	0.052	0.7705	0.6186	0.7350
HLA-A*8001	0.193	0.7809	0.5897	0.7437
HLA-B*0702	0.115	0.6855	0.6671	0.7106
HLA-B*0801	0.073	0.4659	0.4193	0.4676
HLA-B*0802	0.073	0.7417	0.4892	0.6828
HLA-B*1501	0.087	0.6280	0.6202	0.6430

Allele	dist	Pan	Pick	Pan+Pick
HLA-B*1502	0.087	0.5547	0.3987	0.5262
HLA-B*1503	0.091	0.5897	0.6659	0.6506
HLA-B*1509	0.130	0.3017	0.3014	0.3178
HLA-B*1517	0.180	0.8061	0.7233	0.7941
HLA-B*1801	0.147	0.6058	0.5950	0.6167
HLA-B*2705	0.293	0.3616	0.2469	0.3107
HLA-B*3501	0.088	0.7433	0.6978	0.7467
HLA-B*3901	0.147	0.6911	0.5211	0.6721
HLA-B*4001	0.096	0.7694	0.7103	0.7696
HLA-B*4002	0.096	0.7032	0.7111	0.7356
HLA-B*4402	0.048	0.6000	0.6267	0.6480
HLA-B*4403	0.048	0.7361	0.7370	0.7534
HLA-B*4501	0.231	0.6115	0.6462	0.6708
HLA-B*4601	0.213	0.5170	0.4664	0.5173
HLA-B*4801	0.099	0.6791	0.6596	0.6982
HLA-B*5101	0.205	0.6668	0.6343	0.6770
HLA-B*5301	0.088	0.7450	0.6929	0.7478
HLA-B*5401	0.245	0.5991	0.6180	0.6281
HLA-B*5701	0.070	0.7591	0.6940	0.7563
HLA-B*5801	0.070	0.8069	0.7191	0.7957
HLA-B*7301	0.291	0.4874	0.5140	0.5370
Mamu-A1*00101	0.280	0.4315	0.4527	0.4732
Mamu-A1*00201	0.280	0.2889	0.5673	0.4506
Mamu-A1*01101	0.230	0.6842	0.6490	0.6989
Mamu-A1*02201	0.158	0.6397	0.6156	0.6531
Mamu-B*00101	0.313	0.3691	0.0513	0.2320
Mamu-B*00301	0.055	0.8279	0.7529	0.8258
Mamu-B*00801	0.055	0.8296	0.7438	0.8354
Mamu-B*01701	0.446	0.3802	0.4042	0.4416
Mamu-B*04801	0.420	0.7857	0.2510	0.6421
Mamu-B*05201	0.359	0.7039	0.6315	0.7094
Patr-A*0101	0.125	0.4204	0.5559	0.5054
Patr-A*0301	0.076	0.7019	0.5912	0.7013
Patr-A*0401	0.081	0.6897	0.7782	0.7744
Patr-A*0701	0.407	0.3489	0.5665	0.5553
Patr-A*0901	0.081	0.5526	0.5684	0.5899
Patr-B*0101	0.294	0.6457	0.7356	0.7470
Patr-B*1301	0.115	0.7805	0.7191	0.7785
Patr-B*2401	0.294	0.3059	-0.1087	0.1167
Average		**0.6215**	**0.5725**	**0.6374**

Table A.6. Validation results of *NetMHCcons* method. The results are given as Pearson's correlation coefficient (PCC) for final *NetMHCcons* (indicated as *Cons* in this table) and other methods involved in it: *NetMHCpan* (*Pan*), *PickPocket* (*Pick*) and *NetMHC* (*MHC*). Three different averages are given: 1) *Average (all alleles)* indicates the average for all (62) alleles from the validation set 2) *Average (including Pick)* represents the average for the alleles that were not included in the training data set and were within 0.1 or larger distance to the training set (17 alleles) 3) *Average (including MHC)* represents average for the alleles that were part of the training set (41 allele). *#pep* indicates number of peptide binding measurements available for that allele and *#bind* indicates the number of actual binders from all the peptides. *dist* indicates the distance, as measured in terms of the MHC pseudo sequence similarity, from the query allele to the nearest neighbour from the training set.

Allele	#pep	#bind	dist	Cons	Pan	Pick	MHC
HLA-A0101	242	44	0.000	0.8636	0.8290	-	0.8709
HLA-A0201	643	210	0.000	0.8775	0.8859	-	0.8687
HLA-A0203	43	32	0.000	0.7733	0.7734	-	0.7678
HLA-A0206	32	30	0.000	0.7489	0.7299	-	0.7424
HLA-A0211	44	40	0.000	0.6509	0.6850	-	0.6115
HLA-A0212	38	31	0.000	0.6347	0.6822	-	0.5805
HLA-A0216	24	18	0.000	0.8054	0.8274	-	0.7515
HLA-A0219	40	27	0.000	0.8419	0.8217	-	0.7742
HLA-A0301	394	157	0.000	0.7811	0.7736	-	0.7771
HLA-A1101	189	18	0.000	0.7346	0.7211	-	0.7328
HLA-A2301	144	30	0.000	0.6016	0.6098	-	0.5856
HLA-A2402	11	2	0.000	0.5652	0.5207	-	0.5842
HLA-A2403	157	43	0.000	0.8482	0.8469	-	0.8382
HLA-A2501	416	5	0.000	0.5981	0.6236	-	0.5497
HLA-A2601	1,080	62	0.000	0.8140	0.7927	-	0.8145
HLA-A2602	213	67	0.000	0.9440	0.9458	-	0.9271
HLA-A2603	229	24	0.000	0.8682	0.8900	-	0.8257
HLA-A2902	169	59	0.000	0.7945	0.7747	-	0.7961
HLA-A3001	201	16	0.000	0.6644	0.6248	-	0.6692
HLA-A3002	165	28	0.000	0.7415	0.7046	-	0.7421
HLA-A3101	133	86	0.000	0.8404	0.8461	-	0.8293
HLA-A6802	14	2	0.000	0.6404	0.6191	-	0.6489
HLA-A6901	393	13	0.000	0.6205	0.5907	-	0.6097
HLA-A8001	389	9	0.000	0.5601	0.5356	-	0.5499
HLA-B0702	430	229	0.000	0.8563	0.8505	-	0.8504
HLA-B0801	614	82	0.000	0.8889	0.8808	-	0.8860
HLA-B0802	514	18	0.000	0.8234	0.7927	-	0.8027
HLA-B1501	415	57	0.000	0.7714	0.7604	-	0.7646
HLA-B1509	369	16	0.000	0.7005	0.6046	-	0.6554
HLA-B1517	329	12	0.000	0.5214	0.5553	-	0.4894
HLA-B1801	503	15	0.000	0.5576	0.5399	-	0.5518

Allele	#pep	#bind	dist	Cons	Pan	Pick	MHC
HLA-B2705	200	26	0.000	0.6986	0.7149	-	0.6861
HLA-B3501	16	10	0.000	0.7649	0.7705	-	0.6489
HLA-B3901	814	68	0.000	0.7849	0.7365	-	0.7886
HLA-B4001	189	32	0.000	0.9245	0.9072	-	0.9251
HLA-B4601	385	2	0.000	0.4042	0.3701	-	0.3695
HLA-B5101	572	4	0.000	0.4427	0.4787	-	0.3997
HLA-B5301	179	5	0.000	0.4759	0.4320	-	0.4757
HLA-B5701	506	12	0.000	0.5528	0.5234	-	0.5494
HLA-B5801	196	31	0.000	0.8816	0.8624	-	0.8817
HLA-B7301	14	3	0.000	0.4626	0.4553	-	0.3435
HLA-A6823	81	76	0.030	0.4936	0.4936	-	-
HLA-A3207	87	78	0.037	0.4745	0.4745	-	-
HLA-A3215	74	59	0.049	0.5680	0.5680	-	-
HLA-A6601	173	7	0.071	0.4734	0.4734	-	-
HLA-A0319	30	14	0.101	0.5465	0.5968	0.3506	-
HLA-B4013	58	52	0.122	0.3163	0.3297	0.2580	-
HLA-B3801	142	3	0.132	0.5475	0.6119	0.4410	-
HLA-B2720	91	89	0.134	0.6581	0.6704	0.6012	-
HLA-B1402	184	16	0.135	0.3272	0.3608	0.2745	-
HLA-B4506	359	4	0.138	0.2557	0.2652	0.2354	-
HLA-B1542	361	3	0.196	0.3083	0.3050	0.2823	-
HLA-C0501	172	68	0.233	-0.5071	-0.5594	-0.2925	-
HLA-C1502	82	33	0.234	0.3291	0.2715	0.3480	-
HLA-C1402	170	141	0.236	0.4357	0.2140	0.4798	-
HLA-C0602	220	88	0.253	-0.2785	-0.3231	-0.1538	-
HLA-B8301	336	40	0.254	0.7698	0.7015	0.7931	-
BoLA-N01301	93	88	0.309	0.3069	0.3572	0.1771	-
HLA-C0401	364	5	0.315	0.0347	-0.0057	0.0788	-
SLA-10401	15	14	0.352	0.0187	-0.0300	0.0852	-
HLA-E0101	93	12	0.424	-0.0556	0.1127	-0.1332	-
BoLA-N05201	90	84	0.455	-0.0279	0.0255	-0.0612	-
Average (all alleles)				**0.5697**	**0.5613**	**-**	**-**
Average (including Pick)				**0.2344**	**0.2296**	**0.2214**	**-**
Average (including MHC)				**0.7152**	**0.7046**	**-**	**0.6955**

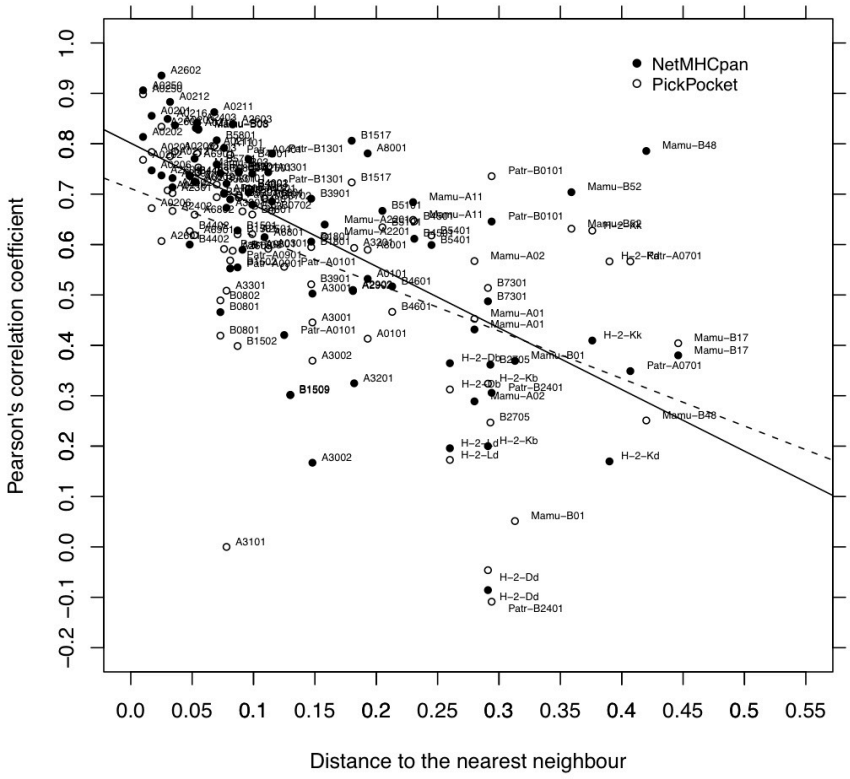

Figure A.2. Predictive performance of the alleles from the benchmark data set as a function of distance to the nearest neighbour. The figure shows performance for each allele of the *NetMHCpan* and *PickPocket* methods. The *solid line* represents the least square fit for the *NetMHCpan* data, and the *dotted line* gives the least square fit for the *PickPocket* data.

Appendix B

Supplementary material for Paper II, Chapter 3

Table B.1. Summary of the quantitative MHC class II peptide-binding data used for the method development.

Molecule name	#pep	#bind
HLA-DR		
DRB1*0101	7,685	4,382
DRB1*0301	2,505	649
DRB1*0302	148	44
DRB1*0401	3,116	1,039
DRB1*0404	577	336
DRB1*0405	1,582	627
DRB1*0701	1,745	849
DRB1*0802	1,520	431
DRB1*0806	118	91
DRB1*0813	1,370	455
DRB1*0819	116	54
DRB1*0901	1,520	621
DRB1*1101	1,794	778
DRB1*1201	117	81
DRB1*1202	117	79
DRB1*1302	1,580	493
DRB1*1402	118	78
DRB1*1404	30	16
DRB1*1412	116	63
DRB1*1501	1,769	709
DRB3*0101	1,501	281
DRB3*0301	160	70
DRB4*0101	1,521	485
DRB5*0101	3,106	1,280
HLA-DP		
HLA-DPA1*0103-DPB1*0201	1,404	538
HLA-DPA1*0103-DPB1*0401	1,337	471
HLA-DPA1*0201-DPB1*0101	1,399	597
HLA-DPA1*0201-DPB1*0501	1,410	443
HLA-DPA1*0301-DPB1*0402	1,407	523
HLA-DQ		
HLA-DQA1*0101-DQB*10501	1,739	522
HLA-DQA1*0102-DQB*10602	1,629	813
HLA-DQA1*0301-DQB*10302	1,719	386
HLA-DQA1*0401-DQB*10402	1,701	559
HLA-DQA1*0501-DQB*10201	1,658	549
HLA-DQA1*0501-DQB*10301	1,689	863
H-2		
H-2-IAb	660	126
H-2-IAd	379	70
Total	**52,062**	**20,451**

The first column gives the names of the molecules, the second column (*#pep*) the number of peptide data available for each molecule, the third column (*#bind*) gives the number of peptide binders. Peptide binders are classified using an IC50 threshold value of 500 nM.

Table B.2. Summary of the evaluation set.

Molecule name	#pep	#bind
DRB1*0101	717	550
DRB1*0301	703	408
DRB1*0701	682	375
DRB1*0801	**838**	**363**
DRB1*1101	813	426
DRB1*1301	**803**	**462**
DRB1*1302	765	404
DRB1*1501	758	218
DRB3*0202	**726**	**287**
DRB3*0301	782	449
DRB4*0101	778	235
DRB4*0103	**764**	**474**
DRB5*0101	731	461

The first column gives the names of the molecules, the second column (#*pep*) the number of peptide data available for each molecule, the third column (#*bind*) gives the number of peptide binders. Peptide binders are classified using an IC50 threshold value of 500 nM. The molecules not present in the training set are marked in *bold*.

Table B.3. Per-locus training comparison with the pan-specific training including cross-loci data.

Molecule name	#pep	#bind	Per-locus training		Cross-loci training	
			PCC	AUC	PCC	AUC
HLA-DP						
HLA-DPA1*0103-DPB1*0201	1,404	538	0.922	0.956	0.922	**0.957**
HLA-DPA1*0103-DPB1*0401	1,337	471	**0.931**	**0.964**	0.929	0.962
HLA-DPA1*0201-DPB1*0101	1,399	597	0.904	0.946	**0.905**	**0.948**
HLA-DPA1*0201-DPB1*0501	1,410	443	**0.873**	**0.956**	0.868	0.954
HLA-DPA1*0301-DPB1*0402	1,407	523	**0.918**	**0.960**	0.912	0.957
Average			**0.910**	**0.957**	**0.907**	**0.956**
***p* value**			**0.625**	**1.000**		
HLA-DQ						
HLA-DQA1*0101-DQB1*0501	1,739	522	**0.817**	**0.919**	0.791	0.901
HLA-DQA1*0102-DQB1*0602	1,629	813	0.692	**0.874**	**0.698**	0.872
HLA-DQA1*0301-DQB1*0302	1,719	386	**0.738**	**0.823**	0.723	0.813
HLA-DQA1*0401-DQB1*0402	1,701	559	**0.815**	**0.916**	0.807	0.914
HLA-DQA1*0501-DQB1*0201	1,658	549	**0.818**	**0.905**	0.802	0.902
HLA-DQA1*0501-DQB1*0301	1,689	863	**0.826**	**0.921**	0.816	0.919
Average			**0.784**	**0.893**	**0.773**	**0.887**
***p* value**			**0.219**	**0.031**		
HLA-DR						
DRB1*0101	7,685	4,382	0.715	0.849	**0.717**	0.849
DRB1*0301	2,505	649	**0.717**	**0.866**	0.708	0.859
DRB1*0302	148	44	0.577	0.769	**0.601**	**0.800**
DRB1*0401	3,116	1,039	**0.663**	**0.843**	0.659	0.841
DRB1*0404	577	336	0.655	0.826	**0.663**	**0.838**
DRB1*0405	1,582	627	0.704	0.860	**0.711**	**0.862**
DRB1*0701	1,745	849	**0.734**	**0.864**	0.729	0.861
DRB1*0802	1,520	431	**0.532**	**0.777**	0.515	0.771
DRB1*0806	118	91	**0.789**	0.925	0.778	**0.927**
DRB1*0813	1,370	455	**0.743**	**0.885**	0.740	0.881
DRB1*0819	116	54	**0.637**	**0.823**	0.608	0.809
DRB1*0901	1,520	621	0.647	**0.829**	**0.652**	0.828
DRB1*1101	1,794	778	**0.775**	**0.880**	0.770	0.879
DRB1*1201	117	81	0.777	0.902	**0.787**	**0.909**
DRB1*1202	117	79	**0.786**	0.915	0.783	**0.916**
DRB1*1302	1,580	493	**0.626**	**0.819**	0.612	0.814
DRB1*1402	118	78	0.737	**0.891**	**0.753**	0.890
DRB1*1404	30	16	0.512	0.607	**0.611**	**0.728**
DRB1*1412	116	63	**0.766**	**0.900**	0.764	0.896
DRB1*1501	1,769	709	0.674	0.828	**0.677**	**0.831**
DRB3*0101	1,501	281	**0.690**	**0.855**	0.683	0.851
DRB3*0301	160	70	0.744	0.858	**0.754**	**0.864**
DRB4*0101	1,521	485	**0.695**	**0.847**	0.693	0.846
DRB5*0101	3,106	1,280	**0.767**	**0.885**	0.760	0.882
Average			**0.694**	**0.846**	**0.697**	**0.851**
***p* value**			**0.541**	**0.405**		

#*pep* is the number of peptide binding data available for each molecule, #*bind* gives the number of peptides that have a binding affinity stronger than 500 nM. The results of the pan-specific approach trained per-locus and on all data are presented in PCC and AUC values. Average performance measures are provided for each locus. *p value* gives *p* values (using binomial test) for PCC and AUC values for each locus .

Table B.4. Leave-one-out results in comparison with *NN-finder* approach.

Molecule name	#pep	#bind	Nearest neighbour	dists	*NN-finder*		*LOO*	
					PCC	AUC	PCC	AUC
HLA-DP								
HLA-DPA1*0103-DPB1*0201	1,404	538	HLA-DPA1*0103-DPB1*0401	0.061	**0.912**	**0.949**	0.900	0.943
HLA-DPA1*0103-DPB1*0401	1,337	471	HLA-DPA1*0103-DPB1*0201	0.061	**0.920**	**0.959**	0.901	0.955
HLA-DPA1*0201-DPB1*0101	1,399	597	HLA-DPA1*0201-DPB1*0501	0.070	0.872	0.913	**0.879**	**0.934**
HLA-DPA1*0201-DPB1*0501	1,410	443	HLA-DPA1*0201-DPB1*0101	0.070	**0.846**	0.939	0.844	0.939
HLA-DPA1*0301-DPB1*0402	1,407	523	HLA-DPA1*0103-DPB1*0201	0.086	0.864	0.922	**0.891**	**0.940**
Average					**0.883**	**0.936**	**0.883**	**0.942**
***p* value**					**1.000**	**1.000**		
HLA-DQ								
HLA-DQA1*0101-DQB1*0501	1,739	522	HLA-DQA1*0102-DQB1*0602	0.233	0.200	0.600	**0.583**	**0.796**
HLA-DQA1*0102-DQB1*0602	1,629	813	HLA-DQA1*0101-DQB1*0501	0.233	-0.079	0.457	**0.365**	**0.672**
HLA-DQA1*0301-DQB1*0302	1,719	386	HLA-DQA1*0501-DQB1*0301	0.187	0.008	0.473	**0.378**	**0.639**
HLA-DQA1*0401-DQB1*0402	1,701	559	HLA-DQA1*0501-DQB1*0301	0.256	0.244	0.602	**0.738**	**0.871**
HLA-DQA1*0501-DQB1*0201	1,658	549	HLA-DQA1*0301-DQB1*0302	0.273	**0.295**	**0.611**	0.131	0.538
HLA-DQA1*0501-DQB1*0301	1,689	863	HLA-DQA1*0301-DQB1*0302	0.187	**-0.172**	**0.392**	-0.236	0.373
Average					**0.083**	**0.523**	**0.326**	**0.648**
***p* value**					**0.688**	**0.688**		
H-2								
H-2-IAb	660	126	H-2-IAd	0.339	0.093	0.538	**0.472**	**0.741**
H-2-IAd	379	70	H-2-IAb	0.339	0.192	0.610	**0.412**	**0.750**
Average					**0.142**	**0.574**	**0.442**	**0.746**

Molecule name	#pep	#bind	Nearest neighbour	dists	*NN-finder*		*LOO*	
					PCC	**AUC**	**PCC**	**AUC**
HLA-DR								
DRB1*0101	7,685	4,382	DRB1*1402	0.219	0.278	0.634	**0.607**	**0.795**
DRB1*0301	2,505	649	DRB1*0302	0.105	0.212	0.599	**0.505**	**0.766**
DRB1*0302	148	44	DRB1*1402	0.080	0.263	0.627	**0.572**	**0.765**
DRB1*0401	3,116	1,039	DRB1*0405	0.045	0.547	0.779	**0.580**	**0.791**
DRB1*0404	577	336	DRB1*0401	0.062	0.564	**0.795**	**0.580**	0.789
DRB1*0405	1,582	627	DRB1*0401	0.045	0.598	0.802	**0.646**	**0.831**
DRB1*0701	1,745	849	DRB1*0819	0.279	0.431	0.719	**0.650**	**0.827**
DRB1*0802	1,520	431	DRB1*0813	0.028	0.361	0.680	**0.399**	**0.699**
DRB1*0806	118	91	DRB1*0802	0.073	0.646	**0.885**	**0.698**	0.882
DRB1*0813	1,370	455	DRB1*0802	0.028	0.395	0.704	**0.411**	**0.707**
DRB1*0819	116	54	DRB1*0813	0.057	0.534	0.748	**0.610**	**0.811**
DRB1*0901	1,520	621	DRB5*0101	0.251	0.411	0.718	**0.550**	**0.768**
DRB1*1101	1,794	778	DRB1*1302	0.057	0.285	0.646	**0.500**	**0.741**
DRB1*1201	117	81	DRB1*1202	0.029	0.685	0.834	**0.737**	**0.874**
DRB1*1202	117	79	DRB1*1201	0.029	0.689	0.859	**0.766**	**0.911**
DRB1*1302	1,580	493	DRB1*1101	0.057	0.264	0.631	**0.340**	**0.663**
DRB1*1402	118	78	DRB1*0302	0.080	0.330	0.676	**0.723**	**0.867**
DRB1*1404	30	16	DRB1*0806	0.131	0.581	0.679	**0.685**	**0.723**
DRB1*1412	116	63	DRB1*0813	0.094	0.549	0.776	**0.671**	**0.897**
DRB1*1501	1,769	709	DRB1*0404	0.201	**0.518**	**0.742**	0.504	0.739
DRB3*0101	1,501	281	DRB1*0302	0.116	0.023	0.515	**0.398**	**0.703**
DRB3*0301	160	70	DRB3*0101	0.146	0.292	0.642	**0.520**	**0.760**
DRB4*0101	1,521	485	DRB1*1404	0.246	0.424	0.693	**0.505**	**0.749**
DRB5*0101	3,106	1,280	DRB1*1101	0.202	0.594	0.798	**0.608**	**0.812**
Average					**0.436**	**0.716**	**0.573**	**0.786**
***p* value**					**<0.0001**	**0.0003**		

#pep is the number of peptide binding data available for each molecule, *#bind* gives the number of peptides that have a binding affinity stronger than 500 nM. *Nearest neighbour* gives the closest molecule from the training set to the molecule in question and *dist* provides the actual distance between two molecules calculated in terms of pseudo sequences similarity. The results of *NN-finder* approach and *NetMHCIIpan-3.0* method in LOO setup are presented in PCC and AUC values. The higher performance value for each molecule is marked in bold. Average performance measures as well as *p values* (where available) are provided for each locus.

Appendix C

Supplementary material for Chapter 5

Table C.1. The list of 68 yellow fever epitopes used for the analysis.

No.	Epitope	No.	Epitope
1	ALYEKKLAL	35	REMHHLVEF
2	AMDTISVFL	36	RIRDGLQYGW
3	DSDDWLNKY	37	RPIDDRFGL
4	DVILPIGTR	38	RPIDDRFGLA
5	EVNPPFGDSY	39	RPIDDRFGLAL
6	FHERGYVKL	40	RPRKTHESHL
7	FLDPASIAA	41	RPRKTHESHLV
8	GEAMDTISV	42	RQWAQDLTL
9	GEIHAVPFGL	43	RRFLPQIL
10	GEIHAVPFGLV	44	RVKLSALTL
11	GLFGGLNWI	45	RVKLSALTLK
12	GLVGVLAGL	46	RVLDTVEKW
13	GLYGNGILV	47	SEMKEAFHGL
14	GMVAPLYGV	48	SMQKTIPLV
15	HAVPFGLVSM	49	SMSMILVGV
16	HESHLVRSW	50	SPKGISRMSM
17	HEVNGTWMI	51	SPRERLVLTL
18	HLKRLWKML	52	SRIRDGLQY
19	HPFALLLVL	53	SVAGRVDGL
20	HTMWHVTRGAF	54	SVKEDLVAY
21	IIMDEAHFL	55	TESWIVDRQW
22	ILNDSGETV	56	TRRFLPQIL
23	IRDGLQYGW	57	VEFEPPHAA
24	IWYMWLGARY	58	VLAGWLFHV
25	IYGIFQSTF	59	VLWDIPTPK
26	KLAQRRVFH	60	VMYNLWKMK
27	KSEYMTSWFY	61	VYMDAVFEY
28	KTWGKNLVF	62	WYMWLGARY
29	KVVNRWLFR	63	YEKKLALYL
30	LLDKRQFELY	64	YMDAVFEYTI
31	LLWNGPMAV	65	YMSPHHKKL
32	MPEAMTIVML	66	YMWLGARY
33	MYMALIAAF	67	YPSGTSGSPI
34	NTDIKTLKF	68	YTDYLTVMDRY

Table C.2. Yellow fever epitopes used for the analysis with their associated HLA class I molecules.

HLA	Epitope
HLA-A*0101	DSDDWLNKY
HLA-A*0101	NTDIKTLKF
HLA-A*0101	YMWLGARY
HLA-A*0101	YTDYLTVMDRY
HLA-A*0101	LLDKRQFELY
HLA-A*0101	KSEYMTSWFY
HLA-A*0201	IIMDEAHFL
HLA-A*0201	GLFGGLNWI
HLA-A*0201	VLAGWLFHV
HLA-A*0201	GLYGNGILV
HLA-A*0201	ALYEKKLAL
HLA-A*0201	LLWNGPMAV
HLA-A*0201	GMVAPLYGV
HLA-A*0201	SMSMILVGV
HLA-A*0201	SMQKTIPLV
HLA-A*0201	YMDAVFEYTI
HLA-A*0201	FLDPASIAA
HLA-A*0201	AMDTISVFL
HLA-A*0201	GLVGVLAGL
HLA-A*0201	ILNDSGETV
HLA-A*0205	IIMDEAHFL
HLA-A*0205	SVAGRVDGL
HLA-A*0205	YMSPHHKKL
HLA-A*0301	VLWDIPTPK
HLA-A*0301	VMYNLWKMK
HLA-A*0301	KVVNRWLFR
HLA-A*0301	RVKLSALTLK
HLA-A*0301	KLAQRRVFH
HLA-A*1101	VLWDIPTPK
HLA-A*2402	IYGIFQSTF
HLA-A*2402	MYMALIAAF
HLA-A*2402	VYMDAVFEY
HLA-A*2601	EVNPPFGDSY
HLA-A*2902	WYMWLGARY
HLA-A*2902	IWYMWLGARY
HLA-A*2902	YMWLGARY
HLA-A*3201	RVKLSALTL
HLA-A*3201	KTWGKNLVF
HLA-A*3201	RIRDGLQYGW
HLA-A*6801	DVILPIGTR
HLA-B*0702	RVKLSALTL
HLA-B*0702	SPRERLVLTL
HLA-B*0702	RPIDDRFGL
HLA-B*0702	RPIDDRFGLAL
HLA-B*0702	RPIDDRFGLA
HLA-B*0702	MPEAMTIVML

HLA	Epitope
HLA-B*0702	SPKGISRMSM
HLA-B*0702	RPRKTHESHL
HLA-B*0702	RPRKTHESHLV
HLA-B*0801	HLKRLWKML
HLA-B*1302	RQWAQDLTL
HLA-B*1501	SVKEDLVAY
HLA-B*1501	HTMWHVTRGAF
HLA-B*2702	SRIRDGLQY
HLA-B*2702	IRDGLQYGW
HLA-B*3501	HAVPFGLVSM
HLA-B*3501	MPEAMTIVML
HLA-B*3501	HPFALLLVL
HLA-B*3501	YPSGTSGSPI
HLA-B*3503	HAVPFGLVSM
HLA-B*3503	RPIDDRFGLAL
HLA-B*3503	MPEAMTIVML
HLA-B*3503	HPFALLLVL
HLA-B*3701	REMHHLVEF
HLA-B*3901	FHERGYVKL
HLA-B*4001	HEVNGTWMI
HLA-B*4001	YEKKLALYL
HLA-B*4001	GEAMDTISV
HLA-B*4001	REMHHLVEF
HLA-B*4001	GEIHAVPFGLV
HLA-B*4001	GEIHAVPFGL
HLA-B*4001	SEMKEAFHGL
HLA-B*4002	RQWAQDLTL
HLA-B*4002	YEKKLALYL
HLA-B*4002	VEFEPPHAA
HLA-B*4402	HESHLVRSW
HLA-B*4402	TESWIVDRQW
HLA-B*4402	SEMKEAFHGL
HLA-B*4403	HESHLVRSW
HLA-B*4403	TESWIVDRQW
HLA-B*5001	VEFEPPHAA
HLA-B*5701	KTWGKNLVF
HLA-B*5701	RVLDTVEKW
HLA-B*5801	KTWGKNLVF
HLA-C*0602	RRFLPQIL
HLA-C*0602	TRRFLPQIL