UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Viktor Mysko

Prediction of MHC class II binding peptides

Master's Thesis (30 ECTS)

Supervisor(s): Ahto Salumets, MSc
Priit Adler, PhD

Tartu 2021

Prediction of MHC class II binding peptides

Abstract:

Major Histocompatibility Complex class II (MHC class II) molecules play an essential role in the immune system. MHC class II molecule's primary function is to load short peptide fragments from pathogens and present them to the T-helper cells to initiate an immune response. The peptide-binding groove of MHC class II molecules is open at both ends. It enables the binding of peptides with different lengths, typically between 13-25 amino acids long. Given the crucial role of MHC-II in selecting peptides for antigen presentation and immune response, significant efforts have been made to develop high-throughput methods for screening for peptide binding to the MHC complex. The main problem is recognizing the peptides that will bind the MHC and initiate the immune response. This thesis shows that neural network models can achieve a good result in binding prediction by using peptide matrices and accumulating outcomes based on other models. It is performed using techniques commonly applied in image recognition problems but is not considered by main research articles. The experiment also showed that binding could be approximately predicted in the same way by a Convolutional Neural Network, Random Forest models, and a stack of Gradient Boosted Models. The predictions are based on the IEDB dataset. The proposed method showed reasonable results compare to other top-ranked models that use complex neural network systems, including different approaches based on MHC biological principles. To sum up, the proposed approach shows that a multi-layer convolutional neural network can be used for binding evaluations and data enlargement techniques like data augmentation.

Keywords:
MHC, peptide, neural network, CNN

CERCS:
B110 Bioinformatics, medical informatics, biomathematics, biometrics

MHC klass II-le seonduvate peptiidide ennustamine

Lühikokkuvõte:

Peamise koesobivuskompleksi (MHC) klass II valgud on immuunsüsteemis väga tähtsal kohal. Selle kompleksi peamiseks ülesandeks on siduda lühikesi patogeenset päritolu peptiide ning esitleda neid teistele immuunrakkudele, milleks on T-abistaja rakud. Kui T-abistajaraku pinnal olev retseptor seondub piisavalt tugevalt peptiidi kandva MHC

II klassi molekuliga, siis see interaktsioon paneb aluse immuunvastusele. MHC II klassi valgu peptiidi seondav “tasku” on mõlemast otsast avatud ning võimaldab siduda väga erineva pikkusega peptiide, valdavalt vahemikus 13-25 aminohapet. Tulenevalt MHC II kompleksi olulisusest immuunvastuse algatamisel on püütud välja selgitada, millised peptiidid võiksid sinna „taskusse“ seonduda. Seda teavet saab kasutada näiteks vaktsiinide väljatöötamisel, näiteks lisades sinna valke, mis sisaldavad selliseid peptiide, mis seonduvad MHC II kompleksile. Käesolevas töös uuritaksegi, kas konvolutsioonilised tehisnärvivõrgud oleksid sobivaks meetodiks ennustamaks, millised peptiidid seonduvad antud kompleksile. Vastamaks sellele küsimusele kasutati antud töös IEDB andmestikku ning töö tulemusena leiti, et konvolutsiooniline tehisnärvivõrk on sobiv vahend antud ülesandeks. Lisaks leiti, et ligilähedasi tulemusi saab ka kasutades juhumetsal või gradiendi võimendusel põhinevaid mudeleid.

Võtmesõnad:
MHC, peptiid, närvivõrk, CNN

CERCS:
B110 – Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika

Contents

1 Terms and Notions

Amino acid - building blocks of proteins.

Antibody - Y-shaped protein produced mainly by B-cells, used to neutralize pathogens.

Antigen - in the thesis presented only as a protein that triggers an immune response.

Epitope - is a region of the antigen that the immune system is recognizing.

Cell - the basic membrane-bound unit that contains the fundamental molecules of life and of which all living things are composed. [bri21]

Pathogen - a bacterium, virus, microorganism that causes disease.

Peptide - a short sequence of amino acids.

Protein - the main building blocks of the organism made out of amino acids.

Receptor - is a protein molecule that receives chemical signals and responds to them (like MHC).

Ligand - is a molecule that forms a complex with other biomolecules and facilitates some biological process.

Innate immunity - refers to non-specific defense mechanisms that come into play immediately after antigen appearance.

Adaptive immunity - is a system of specialized immune cells that enable an antigen-specific immune response.

Major histocompatibility complex (MHC) - regulates the cell-mediated adaptive immune response. The central role is to bind peptides. MHC class I consists of two polypeptide chains, a larger chain encoded on chromosome 6 in the MHC region and a smaller b2 microglobulin encoded on chromosome 15. Peptide length is 8-11 residues. MHC class II has two polypeptide chains as well, but both are encoded on chromosome 6. Peptide length is 15-24 residues. The main difference between MHC classes lies in their peptide-binding clefts, which are more open in MHC-II molecules. [imm14]

2 Introduction

The immune system is a complex immune cell system and proteins that work together to protect the body from invading pathogens. It has two types of responses: innate and adaptive. Innate responses are rapid and reliable but not pathogen-specific and do not have immune memory. On the other hand, the adaptive immune system responds to specific antigens' specific parts, thus being pathogen-specific. Besides, it has a long-lasting memory since not all the cells involved in the immune response die after the infection. The main cells of the immune system are B-cells, and T-cells [df]. The B-cells produce antibodies while T-cells attack pathogens and infected cells.

A key characteristic of T-cells: they are antigen-specific but also cross-reactive. Antigen-specific T-cells are mediator molecules that can function in an antigen-specific manner. Specificity lies in the ability to distinguish between different antigens. It helps the immune system to recognize our cells and invaders (including defective/sick/modified cells). Because the number of potential peptide antigens is much greater than the diversity of T cell receptors, a single T-cell must be able to recognize multiple peptide-MHC complexes - that is called cross-reaction [OvH$^+$14]. This degeneration in T-cell recognition is commonly referred to and associated with both immune protection and disease. T-cell epitope's selection happens mainly due to the delicate balance between the specificity and degeneration of the MHC binding and the T-cell receptor. Although the contribution of MHC to antigen selection has been extensively described and explained, the diversity of the T-cell receptor remains an essential missing link in our understanding of T-cell. Its diversity is generated by the random and imprecise rearrangements of the segments of the α and β genes in the thymus.

Major Histocompatibility Complex class II (MHC class II) molecules play an essential role in vertebrates' cellular immune system. MHC class II molecules' primary function is to load short peptide fragments originated from exogenously derived antigenic proteins and feed them to the cell's antigen-presenting surface, where they can be recognized by T-helper lymphocytes. If the peptide fragment is of foreign origin, T cells can help initiate an appropriate immune response [CZG97]. MHC-II molecules consist of two polypeptide chains, alfa and beta, each consist of two extracellular domains referred to as α1 and α2 and β1 and β2. The chains also consist of a transmembrane segment and a cytoplasmic tail that help to internalize and digest extracellular proteins into fragments [imm14].

Because the peptide-binding groove of MHC class II molecules is open at both ends, there are not that many limitations on the length of the peptide ligand that can protrude at both ends of the pocket. Although typically only about 9 amino acids of the peptide

interact directly with MHC groove residues (binding nucleus). Peptides of up to 30 amino acids can be loaded onto MHC class II molecules [CUG+93]. Human MHC class II molecules (called HLA class II, abbreviated HLA-II) are highly polymorphic, including thousands of different allelic variants in the population. HLA II binding motifs are generally unpredictable - so the peptides capable of binding to multiple alleles have been identified.

Given the crucial role of MHC class II in selecting peptides for antigen presentation and immune response, significant efforts have been made to develop high-throughput methods for screening peptide binding to MHC class II. Indeed, computational methods are an excellent alternative to a valuable laboratory research. To date, there are open databases with quantitative data on the binding of peptide-MHC class II. For example, a widely used Immune epitope database [IED] has binding measurements (IC50 score) for MHC I and MHC II molecules in human and mice cells. The IC50 represents the concentration of an antigen at which 50% of the MHC is inhibited. The lower score represents a higher degree of binding. These data can be used to predict MHC class II binding, which is an attractive alternative to expensive experimental methods.

These crucial MHC research studies inspired this study: DeepMHC [HL17] by using Convolutional Neural Networks for binding predictions for MHC-I, NetMHCIIpan4.1 [RAP+20] by creating the fastest model using peptide flanking regions, NN-align [NL09] also one of the best algorithms that is based on feed-forward neural network, an SMM-align [MN07] - model inspired by ensembles of neural network which is combined with Gibbs sampler method, Consensus method [WSD+08] - an ensemble of the top models at the time of publishment, and a few more.

This thesis aims to build a model that predicts whether a given peptide will bind to MHC II and compare this solution to the existing ones. The model's predictions will help substitute lab experiments partially and unload the amount of work, design better vaccines that will train our immune system to differentiate pathogens and our cells and recognize intruders faster. Additionally, cancer therapies can be designed to recognize cancerous cells by antibodies, et cetera. Moreover, the number of MHC II molecules discovered each year increases dramatically, making it unfeasible to screen serological activity for individual molecules with costly and time-demanding wet-lab experiments [FLW+14]. A great example is a new 2020 pandemic. Multiple research papers are trying to find hints in a problem using in silico methods for binding predictions. One of them investigates how pathogens (their parts) will be presented to the immune cells [KB20]. The research demonstrated MHC I epitope pairs with a pathogen with a high probability for cross-reactivity, but the same pathogen has no pairing with an MHC II. It shows in which direction all the next steps in training the immune system and producing vaccine at the final should be accumulated [KB20].

The proposed method aims to find binding peptides to MHC-II complex for every allele in the IEDB dataset [IED]. The proposed algorithm calculates Convolutional Neural Network (CNN), Logistic Regression (LR), K-nearest Neighbour (KNN), Random Forest (RF), Gradient Boosted Models (GBM), Generalized Linear Models (GLM); model predictions and creating an ensemble of them to select the best performing model or their ensembled combination. The combination of models is unique. Usually, peptide binding models have a lack of data due to the limitation of observed peptides. The proposed method uses a data augmentation technique that multiplies existing data and makes estimations more realistic. This method is not listed in the top research papers but is widely used in image recognition tasks.

The thesis consists of the following parts: Section 3 describes the biological terms of the binding peptides to MHC complexes and basic numerical methods used in the thesis. Section 4 describes related works used for analysis and decision making on how to build the proposed method. Section 5 explains the proposed method and demonstrates results.

3 Background

This chapter provides an overview of the main terms and concepts used in the thesis. Subsection 3.1 describes the background and a specific description of the article's problem from the biological side. Subsection 3.2 describes the main terminology used to define binders and no-binders. Subsection 3.3 describes numerical methods used in the thesis from the informatics side (methods and classification models).

3.1 Immune system and the structure of the MHC-II complex

The immune system is the mechanism of biological components that work together to protect the body from foreign invaders. There are two main parts of the immune system: the innate immune system and the adaptive immune system.

The first defense line is the innate immune system, which includes physical barriers such as skin, different types of white blood cells, and proteins. The chemical properties of the antigen activate the innate immune response. Innate immunity response systems are immediate, while the reactions of the adaptive immune system are slower. If pathogens successfully intrude on the innate immune system, they will face the second line of defense, the adaptive immune system.

However, the responses of the adaptive immune system are more specific (more complex). Adaptive immunity refers to an antigen-specific immune response. At first, the antigen must be processed and recognized. MHC class II molecules loaded with foreign peptides are then transported to the cell membrane to present their cargo to CD4+ T cells. The last step, the process of antigen representation, is based on the interaction between the T-cell receptor and a peptide tied to the MHC molecule. In MHC-I, only the last step is made in processing [vol]. Once the antigen is recognized, the adaptive immune system creates an army of immune cells specially designed to attack this antigen. Adaptive immunity also includes "memory," which makes the future reactions against a particular antigen more effective [oA00].

To sum up, MHC molecules bind the peptide fragments and present them to T-cell receptors for recognition, making them a determining factor in recognizing the host-pathogen interaction regulating many adaptive immune responses. There are two main characteristics of MHC make it complicated for pathogens to evade immune responses: first, MHC is polygenic. Polygenic means that it has several classes, and each person has a set of MHC molecules with a different range of peptide-binding specifics. Second, MHC is exclusively polymorphic. MHC genes show the highest degree of polymorphism

in the human genome. There are several variants of each gene in the population as a whole. The various variants that a person inherits from his parents are known as alleles.

The MHC has three regions: MHC-I, MHC-II, and MHC-III. The human leukocyte antigen encoded in each region includes HLA-A, -B, and -C in the MHC-I region and HLA-DR, -DQ, -DP in the MHC-II region. The MHC-III region encodes an enzyme in steroid metabolism, encodes a chaperone (companion), includes several genes involved in the complement cascade, and includes many other genes of unknown immunological function. When it is referred to as MHC, it usually means MHC-I or MHC-II molecules. A brief genetic map of the MHC regions can be seen in Figure 1. A demonstration of organizational themes within the MHC is demonstrated in the Figure 1. There are more than 200 genes within these regions [imm14].

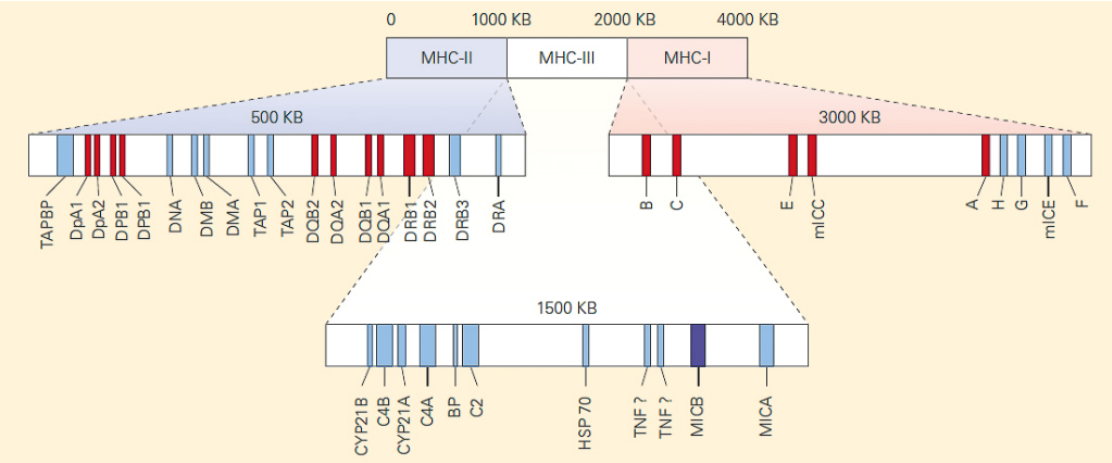

Figure 1. A genetic map of the MHC regions [Bel12].

Both MHC-I and MHC-II molecules consist of two polypeptide chains. The main differences between the two MHC class molecules are in the ends of their peptide-binding clefts, which are more open in MHC-II molecules than MHC-I molecules (Figure 2). MHC class I molecules present peptides derived from intracellular proteins to cytotoxic T-cells, whereas MHC class II molecules stimulate cellular and humoral immunity by presenting extracellular peptides to helper T-cells. The consequence of this difference is that the ends of a peptide bound to an MHC-I molecule are buried within the molecule, whereas peptides bound to MHC-II molecules are not buried. This difference let MHC-II molecules bind peptides of different length and types, which makes it more flexible. Peptides that bind the same MHC-II molecule will share the same middle anchor residues but may vary in other residues' length and sequence. In order for the peptide to stimulate the response of helper T-lymphocytes, it must bind MHC II in endocytic organelles. MHC-I accommodates peptides of 8-11 residues, while MHC-II has 15-24 residues or

even more. MHC-II is more complex, and the number of binding peptides is larger. As shown in Figure 2, MHC-II has a more complex composition of antigen-binding clefts on $\alpha 1$ and $\beta 1$ domains.

A varying length of amino acid chains makes binding prediction a troublesome task. Also, the peptide binding to the MHC molecule is preferably determined by the amino acids present in the peptide-binding core. The binding core usually represents a 9-mer part of the peptide, which definitely will be attached in a case of binding to the MHC-II. However, peptide residues flanking the binding core (so-called peptide flanking residues, PFR), to some extent, affect the affinity of the peptide for binding. There are plenty of articles showing excellent results on predicting binding peptides to MHC-I, but not many for MHC-II. Most of the articles consider binding false positive rate higher than 50% as an excellent result. Moreover, there are many undiscovered genes, which means that any existing research should make good predictions on the unexplored and unknown data. Those challenges explain why MHC II binding prediction remains a problem nowadays.

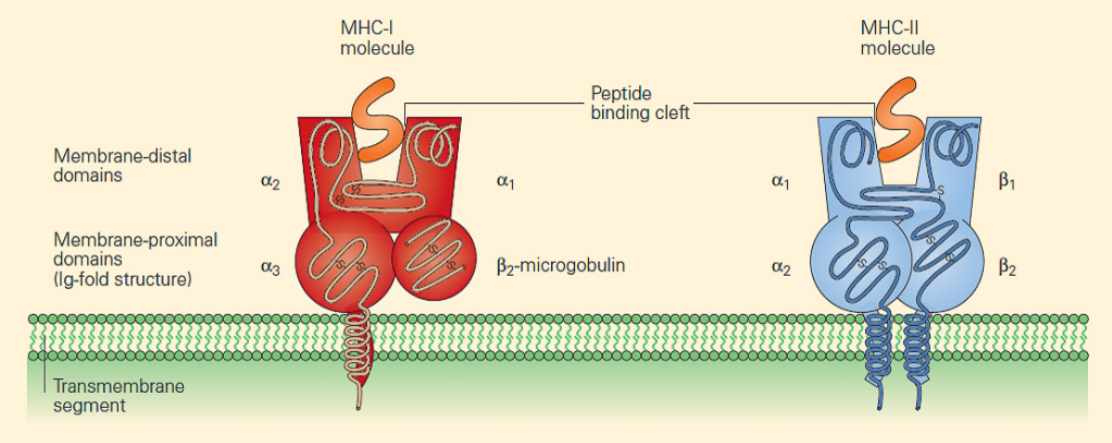

Figure 2. MHC-I and MHC-II molecules in details [Bel12].

There is also such phenomenon as linkage disequilibrium - a genetic phenomenon in which two alleles are occuring together with a higher frequency than is usually expected. It is often used to HLA - human leukocyte antigens, a part of MHC. In HLA class II, linkage disequilibrium exists in the presence of specific HLA-DR alleles that can be used to predict the HLA-DQ allele with a high degree of accuracy before testing. Alleles that are closer together tend to have a higher degree of linkage [imm14]. Specific haplotypes may be advantageous in some immunological sense to have a positive selective advantage. Also, this knowledge can be useful for analysis and peptide binding prediction.

When a peptide tries to bind to the MHC-II complex, there are two main aspects to pay

attention to: does it bind? Furthermore, where does it bind? Those two aspects emphasize the main problems in binding prediction - core binding and binding affinity. Peptide-MHC binding affinity is considerably determined the primary amino acid sequence of the peptide-binding core. However, it shows that the peptide flanking regions (PFRs) on either side of the binding core can affect peptide-MHC binding. Some methods consider a constant number of residues, and some consider a non-stable (flanking) number of binders [CVWV97]. With varying degrees of accuracy, modern methods allow identifying peptides, which are probably binders of MHC class II molecules. However, when it comes to identifying the MHC binding core, most of these methods have limited predictive efficiency [ZUMZ11]. Those problems are illustrated in Figure 3. It is also proved that most of the methods are applicable and similar to methods used in predicting MHC-I-peptide binding.

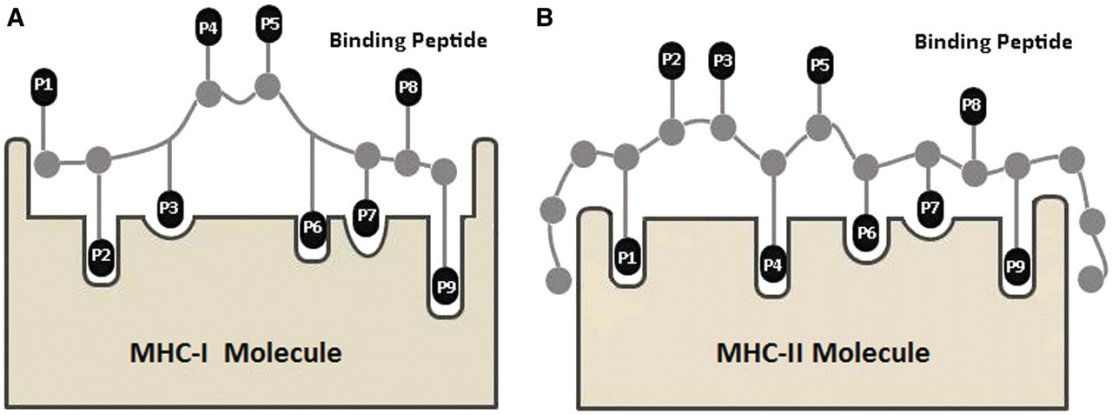

Figure 3. Schematic illustration of peptides binding to (A) MHC-I where a nonamer is fitted into the groove, and (B) MHC-II where a longer peptide is partially fitted into the groove [ZUMZ11].

All listed biological conditions can be used to build a mathematical model that will predict the probability of peptide binding. Different models use the biological properties of MHC to different degrees. Some models consider it necessary to study every detail, and some models are self-learning. There are multiple reviews regarding a given topic in section 4.

3.2 Binders and non-binders

To define binders and no-binders of the MHC-II complex, it is required to numerically represent the peptides and classify them using some affinity measure.

3.2.1 Sequence encoding methods

Peptides are short sequences of amino acids. Amino acids are the building blocks of the protein. There are 20 standard amino acids out of which almost all proteins are made (Table 1). Hence, 20 variables can be represented in multiple ways. There have been different trials of reducing the number of the amino acid alphabet by ignoring some of the acids due to their usage of charge, volume, hydrophobicity, and other chemical descriptors, but it did not improve the final result [RAP+20].

Table 1. 20 standard amino acids

Letter code	Full name	Letter code	Full name
A	Alanine	L	Leucine
R	Arginine	K	Lysine
N	Asparagine	M	Methionine
D	Aspartic acid	F	Phenylalanine
C	Cysteine	P	Proline
Q	Glutamine	S	Serine
E	Glutamic acid	T	Threonine
G	Glycine	W	Tryptophan
H	Histidine	Y	Tyrosine
I	Isoleucine	V	Valine

Using any statistical method to predict antigens, it is a good practice to use different encodings. The most common are:

- **Sparse encoding** - each amino acid is represented as a 20 bit binary string (Figure 4).
- **BLOSUM** (BLOcks SUbstitution Matrix)- is a substitution matrix used for sequence alignment of proteins [HH92] (Figure 5).
  BLOSUM matrices are used to estimate alignments between evolutionarily divergent protein sequences. They scanned the BLOCKS database for very conserved regions of protein families (which have no gaps in sequence alignment) and then calculated the relative frequencies of amino acids and their substitution probabilities. Then, it calculates a log-chance estimate for each of the 210 possible substitution pairs of 20 standard amino acids. All BLOSUM matrices are based on observed alignments. They are not extrapolated from closely related proteins, so some ANNs can have surprisingly bad accuracy on unobserved data. Besides, it is used in most investigations. This sequence encoding method is selected in

the proposed method because BLOSUM matrices are flexible data for deep neural network models to be represented in numerous ways.

- **Reduced amino acid alphabets** - reduces amino acids by using the charge, volume, hydrophobicity, and other chemical descriptors. Top-ranked binding methods are not using the current method. Moreover, developers of the high-ranked NetMHCpan model and Denmark Technical University scientists [RAP+20] described this encoding technique as inefficient. In reduced alphabets, it is usually used 10 or 14 residues. Also, there is a proposed method to use only 5 residues [RSBH+97]. Notably, most of the proposals about reduced amino acids alphabets were made in the 1990s.

Inp Neuron	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
AAcid																				
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4. Sparse encoding example

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2	C
S	-1	4	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3	S
T	-1	1	4	1	-1	1	0	1	0	0	0	-1	-1	-1	-1	-1	-1	-2	-2	-2	T
P	-3	-1	1	7	-1	-2	-2	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4	P
A	0	1	-1	-1	4	0	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-1	-2	-2	-3	A
G	-3	0	0	-2	0	6	-2	-1	-2	-2	-2	-2	-2	-3	-4	-4	-3	-3	-3	-2	G
N	-3	1	0	-2	-1	-2	4	2	0	0	1	0	0	-2	-3	-3	-3	-4	-3	-4	N
D	-3	0	1	-1	-2	-1	2	4	1	0	0	-2	-1	-3	-3	-4	-3	-4	-3	-4	D
E	-4	0	0	-1	-1	-2	0	1	5	2	0	0	1	-3	-3	-3	-3	-3	-2	-3	E
Q	-3	0	0	-1	-1	-2	0	0	2	5	0	1	1	-3	-3	-2	-2	-2	-1	-2	Q
H	-3	-1	-1	-2	-2	-2	1	0	0	0	8	0	0	-2	-3	-3	-3	-1	2	-3	H
R	-3	-1	0	-2	-1	-2	0	0	1	1	0	5	2	-3	-3	-2	-3	-3	-2	-3	R
K	-3	0	0	-1	-1	-1	0	0	1	1	0	2	5	-3	-3	-2	-3	-3	-2	-3	K
M	-1	-1	-1	-2	-1	-3	-2	-3	-3	-2	-2	-3	-3	5	1	2	1	0	-1	-1	M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4	2	3	0	-1	-3	I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4	1	-1	-2	-2	L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4	-1	-1	-3	V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6	3	1	F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	2	Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

Figure 5. BLOSUM example [HH92]

3.2.2 Biding affinity measure

Biding affinity or measure of binding is usually made in one of the frequent measurement units: Kd or IC50.

Kd "dissociation constant" - equilibrium constant, which measures a larger object's tendency to divide (dissociate) into smaller components reversibly. It corresponds to the concentration of antigen (peptide) at which half of the MHC complexes are occupied in equilibrium (i.e. the concentration at which the MHC bound to the antigen is equal to the free MHC).

IC50 or half-maximal inhibitory concentration - is a quantitative measure that indicates the concentration of an antigen (peptide), at which a particular inhibitory substance is needed to inhibit, in vitro, a given component (MHC in our case) by 50%.

Kd and IC50 units are pretty similar. For both measurement units, the number closer to 0 (the smallest) indicates stronger binding. IC50 is used to define binders and non-

biners in preliminary analysis in the thesis. In this thesis, the cut-off value of 1000 means if IC50 < 1000, then the antigen is a binder. Some articles use 200 or 500 as a cut-off value, but in this thesis, it is decided to follow IEDB and DTU (Danish Technical University) approach and includes weak binders that lie in 500-1000 IC50 diapason.

3.3 Numerical methods

As mentioned above, each peptide is a short sequence of amino acids represented numerically as a vector of amino acid residues (in numerous methods). There are multiple ways to work with numerical data presented as vectors/matrices, and it is used in common methods of numerous successful researches. The most common proven methods are artificial neural networks (ANN), hidden Markov models (HMM), kernel methods with support vector machines (SVM), logistic regression with regularization, and k-nearest neighborhood (k-NN). Another popular approach, which is not necessarily a machine learning method, is a position-specific scoring matrix (PSSM), or any statistical method combined with broad biological knowledge for filtering, excluding, or exploring peptides that suit biological terms [ZUMZ11]. Review of best binding methods located in the section 4. The methods listed below are the most common in multiple peptide-MHC binding methods and used in the thesis.

3.3.1 Methods and metrics for evaluating the models

Two main performance measures that are frequently used: AUC and Pearson correlation. **AUC or area under the ROC curve** - is a portion of the area under the unit square's ROC curve. Its value will always be between 0 and 1.0. The AUC has an essential statistical property: the AUC of a classifier is equivalent to the probability that the classifier will give a randomly chosen positive instance a higher score than a randomly chosen negative instance.

The Kappa score - is often used to test the inter-rater reliability. Kappa measures the relationship of an agreement to disagreement in scores. The metric is very dependant on the balance of the True Positives values and True Negatives, meaning that the data scores should be entirely distributed on a scale of the confusion matrix. In simple words, it measures how well the classifier performed as compared to how well it would have performed merely by chance.

Confusion matrix - is a table that is often used to characterize the performance of a classification model (or "classifier") on a set of test data for which the correct values are

known. Using the matrix can be calculated accuracy (AUC), Misclassification Rate, Null Error Rate, Cohen's Kappa, and other metrics that can be important in a classification problem.

Cross-validation is a set of methods for measuring a prediction model's performance on new test data. The basic idea is to divide the data into two sets: a train set and a test set. The train set is used to build the model, and a test set (or validation) is used to test the model by measuring the prediction error. The train set has n-1 folds of the data, while the remaining fold(-s) is used for a test set. Cross-validation involves the selection of the same statistical method several times using different subsets of data. This Thesis uses 5-fold cross-validation (or k-fold). The K-fold method of cross-validation evaluates the model's effectiveness on different subsets of training data and then computes the average prediction error rate.

Overfitting refers to a model that learns noise instead of true signal. This happens when a model learns details and noise in training data to the extent that it negatively affects the model's effectiveness on new data. The opposite term is underfitting, which refers to a model that can neither compose the training data nor generalize to new data.

3.3.2 Classification methods

Deep Learning is becoming a prevalent subfield of machine learning due to its high-performance across many data types. A systematic way to use deep learning to classify images is to build a **convolutional neural network** (CNN). Lots of diverse data can be represented numerically as an image. However, the images are multidimensional matrices with all the needed info about the image's size, color, and pixels. In practice, if any data can be represented as a matrix of numbers, that data can be processed to any neural network. Pixels in images are usually related. For example, a particular pixel group may mean an edge in an image or some other pattern. Convolutions use this to help recognize images. A convolution multiplies a matrix of pixels with a filter matrix called the kernel and sums up the multiplication values [cnn]. After that, the convolution slides over to the next pixel and repeats the same process until all the image pixels have been covered. (Figure 6) The output can be used to train the computational model, which will find patterns that make images recognizable and predictable to perform a set task.

The **K-Nearest Neighbors** KNN [Kha18b] is a supervised machine learning algorithm that relies on labeled input data to learn a function that produces an output when given new unlabeled data. The KNN algorithm assumes that similar things exist nearby. In simple words, similar things are near to each other. The K stands for the number of neighbors that are considered the same class members.

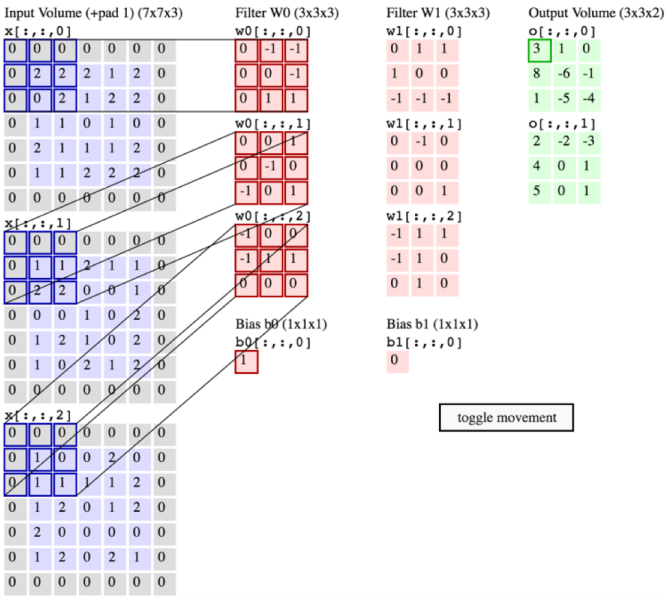

Figure 6. An example of applying a kernel in CNN [Sta]

Logistic regression (LR) [lr19] is an algorithm that is well suited for studying the binary classification between a categorical response variable and one or more categorical or continuous predictor variables. The LR is simple to implement, and it makes no assumptions about distributions of classes in feature space. The number of classes can be extended easily, fast training. The main disadvantage of the LR - it poorly predicts non-linear problems, which are the most common.

Random forest (RF) [Kha18a] is a flexible, easy-to-use machine learning algorithm that gives excellent results most of the time, even without adjusting the hyperparameters. It is one of the most popular algorithms due to its simplicity and variety. The "forest" it builds is an ensemble of decision trees usually taught by the bagging method. It creates several decision trees and combines them to obtain a more accurate and stable forecast. One of the significant advantages of a random forest is that it can be used for classification and regression problems, which constitute most modern machine learning problems. RF usually is precise, but it takes much time to train the model if the number of random trees is too big (which is a good practice in multiclass problems).

Stacking is a method to assemble multiple classification or regression models. There are many techniques to ensemble models, but the widely used approaches are Bagging or **Boosting**. Bagging means that multiple similar models with a high variance are averaged to decrease variance. The boosting builds an ensemble by training weak

learners sequentially so that each added model incrementally increases the model's performance. The main idea is to apply a top layer model to conduct all results from low-level models to perform better predictions. In the proposed method, the boosting stacking ensemble is used. It takes the 4 different models (it is better to use different predictors and select different advantages from each of them) and optimize them. It is good to use different types of predictors in a stacking model because each predictor makes different assumptions and calculates different weights. This info can improve the best model or produce a new model that will outperform others.

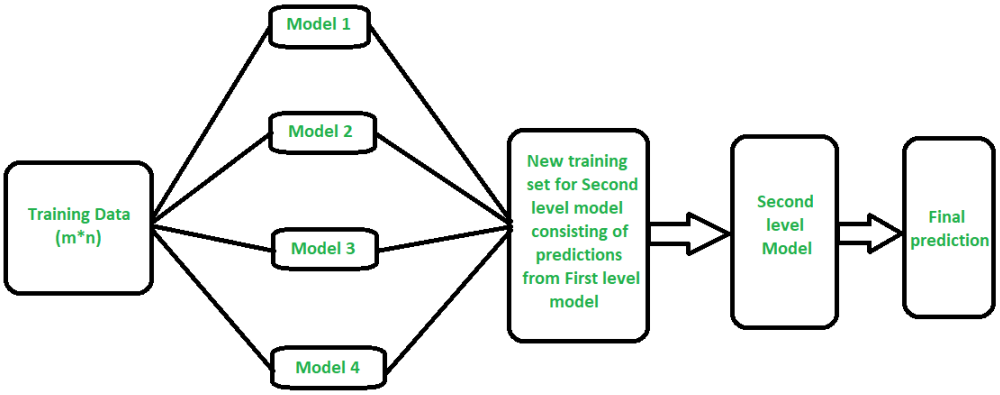

Figure 7. A stacking scheme example, where 4 models are used on the first level, and 1 model is used on the second level before making final predictions. [Sta]

Gradient Boosting (GBM) [Kwa19] is a machine learning enhancer. It depends on the intuition that the best of the possible subsequent models combined with the previous models decreases the overall prediction error. The main idea is to set the target results for the next model in order to minimize errors. The gradient Boosting aims to optimize the loss function and improves the weak learner to predict. It is also an additive model to insert weak learners to minimize the loss function. Gradient Boosting models can be used in stacking ensemble to aggregate knowledge of different models and make suitable predictions by using its nonlinearity.

The **Rectified Linear Unit** (RELU) [rel] is the most popular activation function in deep learning models. The function returns 0 if it receives any negative input, but it returns that value for any positive value x. Surprisingly, such a straightforward function (and one composed of two linear pieces) can allow your model to account for non-linearities and interactions so well. However, the ReLU function works great in most applications, and it is very widely used as a result. RELU as an activation function serves

two primary purposes: to help a model account for interaction effects, and help a model account for non-linear effects.

Dropout [Bud16] is the regularization technique that is used to prevent overfitting in the model. **Regularization methods** are penalizing the model's parameters to prevent overfitting. Dropouts are added to randomly skipping (or dropping) some percentage of neurons of the network. When the neurons are dropped, the next connection to those neurons is also dropped off. This is done to strengthen and intensify the learning of the model.

4 Related work

This section presents an overview of different works related to the topic. Most of them considered high-ranked models that achieve magnificent results. During 2020 and the worldwide pandemic, binding peptide prediction was more discussed than usual in the scientific community. By the end of 2020, there are lots of improvements in popular methods. MHC-II binding peptide prediction accuracy of 60-80% was considered a top result back in 2018-19 (by IEDB stats [IED]). Nowadays, in the second half of 2020, multiple improved high-ranked models promise up to 90% accuracy on some data sets [RAP+20]. However, there is no evidence that those models will perform so well on new HLA alleles and lab experiments. That is why there is no permanent solution for MHC-II binding peptide prediction yet.
Preliminary research included articles that showed outstanding top results in resolving binding core or binding affinity problem. Some other works that are considered to be influential in the field are listed as well.

A stabilization matrix alignment method called **SMM-align** is presented by Morten Nielsen, and his colleagues from the Technical University of Denmark [MN07] in 2007. The method is an extension of the stabilization matrix method that seeks to identify a weight matrix that optimally reproduces the measured IC50 values for each peptide in the training set. SMM-align allows to identify the MHC class II binding motif in terms of a position-specific weight matrix. To sum up, they created a hybrid of existing SMM method and the Gibbs sampler method. The model takes the BLOSUM input, and the ensemble average determines the binding predictions over the predictions obtained from the different encoding schemes. At the time of publishing (2007), the SMM-align method was the best one. It proves the importance of Gibbs sampler weight matrices (a weight matrix describing the binding motif for each allele) and the SMM method that seeks to identify a weight matrix that optimally reproduces the measured IC50 values for each peptide. Currently, the SMM-align groundwork is considered the 3rd best method by the IEDB benchmark.

The method that is often used in IEDB and other benchmarks - **Comblib** [SAM+08] published in 2008. It is not considered the best approach for predicting binding affinity or binding core. This method is extensively used for comparisons with new algorithms. Comblib is considered as an simple method with low computational time. It uses positional scanning of combinatorial libraries and peptide synthesis with MHC purification and detailed bioinformatic analysis. The technique introduces combinatorial libraries and deep bioanalysis potential even though it is based on MHC-I predictions. The average predictions vary by 0-30% ranking score, according to the latest IEDB results.

Given these points, it seems reasonable to create a consensus method based on high-ranked models. Beside this thesis, there is a model that also realizes the consensus idea. It is ranked as the best model by the IEDB benchmark. **The Consensus** approach [WSD+08] was created in 2008 and still valid for predictions. It utilizes SMM-align, ARB (Average Relative Binding matrix), and MHC2PRED (support vector regression method). In theory, a new consensus approach can be developed in the way that outperforms the current, which is based on methods investigated 20 years ago. While scientists try to create the best state-of-the-art algorithm, it is reasonable to accumulate available knowledge that can be competitive for a long time as the current consensus method.

The investigation made by Morten Nielsen and Ole Lund [NL09] in 2009 proposes a new biding prediction method called **NN-align**. They used a big dataset of 14 HLA-DR alleles, each characterized by at least 420 and up to 5166 peptide binding data points. The minimization of peptide overlapping was achieved by partitioning datasets into five parts.
NN-align's main body consists of feed-forward NN that has a two-step procedure that simultaneously estimates the optimal peptide-binding register (core) and network weight configuration. The model is a conventional feed-forward neural network by itself with Gradient descent back-propagation. The effect of data redundancy was limited by the step-size of back-propagation divided by the binding core redundancy of the given peptide. The binding core redundancy calculated using a Hobohm-1 algorithm [HSSS92]. The peptide core was presented to the network using BLOSUM encoding. The networks trained using cross-validation, and ensembles trained with 2, 10, 20, 40, and 60 hidden neurons. For each training/test set configuration, the 10 networks with the highest test-set Pearson correlation coefficient were selected to form the final network ensemble. The binding core of a given peptide is assigned by a majority vote of the ensemble networks. At the final, they used different benchmarks such as Lin, Wang (with 10-fold cross-validation), El-Manzalawy to investigate how the performance of the NN-based method depends on the inherent similarity in the peptide data (with SVM, composition transition distribution (CTD), local alignment kernel (LA), and k-spectrum kernel (5-spectrum)). The final result calculated with AUC rate using IEDB benchmark [IED]. Also, the NN-align proved that it takes into account higher-order sequence correlations. Accordingly to IEDB, NN-align is the second-best method for now and the best single-model method. AUC variates between 55-88% on different datasets.

An interesting method called **TEPITOPEpan** introduced in 2012 without using ANNs [ZCW+12]. A group of scientists extended the existing TEPITOPE model. They achieved the second-best result for predicting binding core at the time of publishing the article. The research is based on HLA-DR molecules because they are the most widely studied. HLA-DR molecules have DRA and DRB, corresponding to alfa and beta domains, respectively. While the DRB allele is diverse, the DRA allele is almost

identical, so the binding specificities of DR molecules are mainly determined by DRB. Thus, the binding specificity of a DRB allele indicates the corresponding HLA-DR group. The decision to update a TEPITOPE method to a pan-specific method was made because pan-specific methods can predict the specificity of peptides binding to MHC molecules of any particular locus. The TEPITOPE generates pseudo sequences of MHC binding pockets, computing the pocket similarity and weights between alleles, and computes PSSM (position-specific scoring matrix). The outcome illustrates that TEPITOPEpan is the best method to predict binding core sequence and among the best for predicting binding affinity (at a time of publishing). Simultaneously, the universality of the method is questionable because TEPITOPEpan is trained and tested in DR gene. It is quite unpredictable how it would act on other genes, and the article does not mention it. Interestingly, the IEDB benchmark rates this method as a 4th best solution with 47-93% accuracy on different datasets.

The analysis about pan-specific method prediction of peptide-MHC class II binding affinity with improved binding core identification proposed another model called **netMHCIIpan** [MAN15] in 2015. Recurrent updates are notable advantage of this method. IEDB servers and benchmarks using netMHCIIpan3.1, while recently in 2020, they presented version 4.1 that promises outstanding results.
The netMHCIIpan3.1 is based on the offset correction method. This method is fully automated and unsupervised. It means that no information about the binding core's actual location is used to define the offset values. Input - BLOSUM encoding. The input layer of the neural network has 906 neurons: a 9-mer core of 20 aminoacids = 180 neurons; 40 additional input neurons were used to encode the composition of the peptide flanking regions (PFRs), calculated as the average BLOSUM scores on a maximum window of 3 amino acids at either end of the binding core; pseudo sequences of 34 residues for alpha and beta chains of MHC resulting additional 34 x 20 = 680 neurons. The ensemble of artificial neural networks was trained using five-fold cross-validation; alternative hidden layers of 10, 15, 40, and 60 hidden neurons; and 10 initial configurations of the network weights. Generated subsets minimized clusters of peptides for cross-validation that share identical stretches of at least nine amino acids. The result calculated using offset values for the 200 networks in the ensemble for all the training set molecules. They used a majority vote scheme to compile a separate list of offset values for each locus. The binding affinity rate achieved a score of 0.8-0.9 of AUC, but binding core results are not listed because they are statistically insignificant.
A new version of **NetMHCIIpan4.1**[RAP+20] that was presented recently in July 2020, promises 90% AUC in most datasets. It is a newly developed method, so there is no data in the IEDB benchmark yet. They deployed NNAlignMA(an updated NN-align method)[ARB+19] to update the NetMHCIIpan3.1 by augmenting their training capabilities and increasing their predictive performance. It was implemented by incorporating NNAlignMA into the new models' core, allowing them to expand their training sets

greatly. Moving further, they performed a full independent epitope evaluation on both models and showed how the updated methods outperform others. The updated method consists of 4+ million data points covering 116 distinct MHC class II molecules. In short, the NNAlignMA framework is a single-allele framework permitting the integration of mixed data types of binding affinity and eluted ligands (EL - data retrieved from mass spectrometry (MS) device experiments). NNAlignMA extends this training framework to allow the incorporation of EL data. It is achieved by iteratively annotating the best single-allele to the multiallelic data during the model training, effectively deconvoluting the multi-allelic binding motifs. The model trained on 1000 human and mice molecules. Also, they added extra configuration. Moreover, the model can define strong and weak binders by the rank score. The rank of a query sequence is computed by comparing its prediction score to the distribution of prediction scores for the MHC in question, estimated from a set of random natural peptides. Lower rank corresponds to a stronger binding.
Both netMHCIIpan methods are considered the best state-of-the-art methods, so their models will be used for further analysis and comparison.

An exploration made by Jianjun Hu, Zhonghao Liu [HL17] in 2017 considered convolution neural networks (CNNs) as one of the best types of neural networks for predicting binding peptides. Their research's final result is a model for predicting binding peptides to the MHC-I complex called **DeepMHC**. It achieves notable results (the avarage AUC is 0.72), but as mentioned above, the peptide-MHC-II binding is a tricky problem. CNNs seem to be a better solution due to their inherent capability to learn the hierarchical features required to achieve high-performance pattern recognition, concluded in the article. Following this success, deep CNN models can be applied to various bioinformatic problems, especially in DNA/RNA motif/sequence modeling and prediction, including well-known algorithms listed above and below. The DeepMHC achieves strongly better performance by a manageable CNN model, which only uses the peptides' raw amino acid sequences as input without tedious, tricky, expert-based feature extraction or encoding. As a note, they used BLOSUM encoding, but they do not consider it as an expert thing. The neural network architecture can be taken into account as well. DeepMHC is considered an easy and successful solution so that it can be interpreted as an MHC-II solution. Their CNN based models are composed of two stacked convolutional layers, one max-pooling layer, and one fully connected layer. In addition, they concluded that the NetMHCpan algorithm (analog of NetMHCIIpan, but for MHC-I) is very time-consuming and sophisticated. It utilized a pan-specific strategy, which trained each allele's samples on a set of artificial neural networks with 22 to 86 hidden neurons with three types of input encodings and then picked the best 15 ANN networks to compose an ensemble ANN prediction model. (In the case of MHC-II, those numbers are more significant and listed above). Out of their exploration, several ideas can be discussed and adopted: the number of convolution filters should be much

larger than those used in DNA-binding predictors due to the significantly larger search space; increasing the number of convolution/max-pooling layers to more than three layers does not necessarily improve the prediction performance (at least in their dataset); multi-channel one-hot encoding works much better than the naive 2D matrix encoding for MHC-I binding prediction.

In the article by Peng Wang and colleagues made in 2010, there is a review of dependencies of HLA DR, DP, and DQ molecules [WSK+10]. It can be concluded that the investigation played a notable role in creating the NetMHCIIpan4.1 model by Morten Nielsen and colleagues. They found those predictions for HLA DR molecules perform equally well for DP or DQ molecules. Also, there is a significant increase in accuracy followed by extra-large dataset. The presence of homologous peptides (peptides with similar structure) between training and testing datasets should be avoided to give real-world estimates of prediction performance metrics. It decreases the absolute AUC values. Also, the article claims that a positive effect of having more training data is more valuable than reducing homologous peptides, which gives no benefits but makes the dataset realistic. Authors recommend that classifiers created for end-user applications should be trained with all available data to gain maximum predictive power for epitope identification. Using the listed study, they created a consensus approach with a combined NN-align and combinatorial peptide library. At the time of posting, it was ranked highly, right after NN-align.
Note, some models are using mice molecules for binding prediction, and it was studied as well. According to Ragnar Lindstedt and colleagues [LLP+95], the MHCII HLA-DM molecule and its murine equivalent H2-M are located intracellularly and are absent from the cell surface. The hypothesis claims that the association between HLA-DM and H2-M is crucial for binding predictions were not verified. The investigation showed that the targeting motif of H2-M appears to be supplementary rather than essential for class II-peptide association. So, mice data was not used in the thesis.

5 Proposed method and results

This section presents a description of the dataset that was used for building the model. Subsection 5.1 is an overview of the dataset. Subsection 5.2 describes the primary analysis. Subsection 5.3 describes the proposed methodology and the results.

5.1 Dataset description

The biggest available dataset from IEDB was used to train a model. It covers 26 MHC alleles and contains over 40,000 binding affinities. It is provided by a large scale dataset of over 17,000 HLA-peptide binding affinities for a set of 11 HLA DP and DQ alleles expanded with HLA DR alleles resulting in a total of 40,000+ MHC class II binding affinities covering 26 allelic variants. The labels of binding peptides are defined as binders and non-binders (BD and NB), where 1 denotes binders (IC50 < 1000) and 0 corresponds to non-binders (IC50 => 1000).. This value was inspired by NetMHCIIpan, and NN-align articles [MAN15, NL09] and server on the DTU website, where they include binding of weak binders. (weak binder is in 500-1000 IC50 range). Most of the data retrieved from mass spectrometry (MS) device experiments. As listed previously, a lower IC50 rate indicates stronger binding.
Available peptides have a different length that varies from 9 amino acids to 24, with some exceptions of 24+ length. For instance, DRB1*01:01 allele has a peptide of 37 amino acids. Usually, extra-large peptides are defined as no-binders.

Table 2. Top rows of the HLA-DPB1*03:01-DPB1*04:01 data

Peptide	Label	IC 50
DITVKNCVLKKSTNG	0	46729.1
APEVKYTVFETALEK	1	8.2202
ATISATPESATPFPH	0	46729.1
FDPYGATIKATPESA	0	13799.5
KFPELGMNPSHCNEM	1	174.635
APQLPDDLMIRVIAQ	0	34403.8

Table 3. An example summary of the HLA-DPB1*03:01-DPB1*04:01 data

Length of peptides	Quantity
10	2
11	1
12	1
13	5
14	20
15	**979**
16	19
17	6

As can be seen, there are multiple challenges regarding dataset. Most of the peptides in every allele have a length of 15 amino acids, so mainly, our predictions should be

evaluated for 15-length peptides. Also, there is a visible lack of data. Multiple studies tried to resolve that problem by combining different datasets or merging allelic data. Also, there are not many models trained on DP and DQ locus data because they have a limited number of peptides. To sum up, challenges are listed below:

1. The massive polymorphism of MHC genes, with several thousand allelic variants in the HLA loci.
2. The unbalanced peptide samples.
3. The variant preference of the peptide lengths for different allele variants.
4. The varying affinity thresholds for different allele variants of MHC molecules. Each model uses a different IC50 value as a threshold.
5. The nonlinear high-order or distal dependencies between different amino acid positions of the binding peptides. But most of the relations are linear [NAPB20].

5.2 Preliminary analysis

Most of the peptides have a length of 15 residues. All the different length peptides will be skipped for simplicity. A predictor for only peptides of 15 amino acids long was done. Shorter or longer peptides will influence the predictor in the wrong direction even though it is comprehensive enough. The BLOSUM encoding represents peptides, where each amino acid particle is represented as a vector of shape [1,20]. Peptides of different lengths will make the task too complicated because BLOSUM matrices will have different shapes for each peptide, and more computational power would be required to normalize the data. Each model can be used for a peptide of any length, but each length group should be evaluated separately. Of course, they can be a positive addition to understanding the MHC behavior, but most research proved that reducing shorter sequences will not affect results dramatically.

Classes were made by thresholding the IC50 at the 1000 (units) level; thus, peptides with IC50 < 1000 were considered binders. Binders can be divided into weak and strong binders, where strong binders can be calculated as the top 2% of binders, such as NetMHCIIpan4 [RAP$^+$20] do that by defining a peptide as a strong binder with IC50 < 100. Each strong binder can be recognized as a weak binder, and it would not influence prediction accuracy because the main task is to find out whether peptides will bind or not. The proposed method does not separate weak and strong binders because, in binary representation, they will be presented as binders.

The primary analysis uses the ggseqlogo package [Wag17] to visualize the sequence motif using a sequence logo. It helps us see which positions are in the peptide and which amino acids are significant for the binding to MHC. Higher letters indicate higher importance and frequency in the sequences — the total height of the position showing the information content measured in bits. A higher number of residues correspond to the higher letters, meaning that better conservation is at that position.

Figure 8 demonstrates that peptides with a shorter sequence have a relatively different position. It can help to define the importance of each amino acid visually, and it is the position. Figure 9 shows that each position is essential for the MHCII allele in most cases while MHCI (that bind 8-11 sequences) has more different important positions for peptides of equal length. It is also visible that amino acids A, K, and G are more frequent in binding for 15-length sequences. This knowledge can reduce the alphabet, but those differences are inefficient [RAP+20].

5.3 Defining models and receiving intermediate results

Models were created so that $f(x) = y$, where f - models, x is the peptide and y is one of two classes: binders and non-binders. Each x is encoded into a 2-dimensional 'image,' which can be visualized using the PepTools package [LEJ] - that helps to derive the PSSM matrix and encode peptide sequences using the BLOSUM62 substitution matrix. Encoded matrices can be represented as in images (Figure 10).

5.3.1 Convolutional Neural Network

The idea of using Convolutional Neural Networks (CNNs) for binding predictions is not novel. CNN uses relatively little pre-processing compared to other image classification algorithms. As shown in Figure 10, BLOSUM matrices can be represented as different pixel images of size 15x20, where 15 is the length of the peptide and 20 - standard amino acids. The first obstacle is in matrix transformation. CNN is an algorithm widely used for image classification, where input data represented as large matrices with multiple color channels. In the case of peptides, they can be represented as relatively small "images" with a 1-color channel and neighboring pixels that are usually not relative as in real-life images. Because of the simplicity of this setup, those conditions will tend to overfit.

The **data augmentation** is used to enlarge the dataset. A CNN can robustly classify objects even if placed in different orientations - a property called invariance. More specifically, a CNN can be invariant to translation, viewpoint, size, or illumination (or a

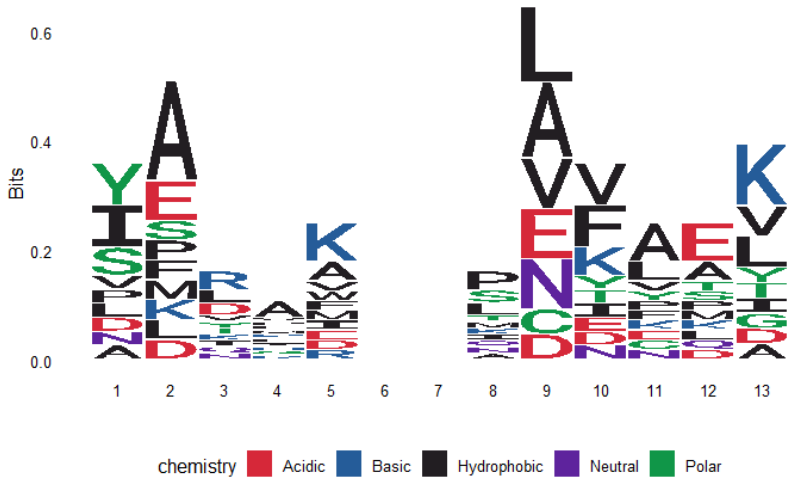

Figure 8. QQSEQLOGO visualisation of the sequence motif of the HLA-DQA1*03:01-DQB1*03:02 (peptide length = 13 amino acids)

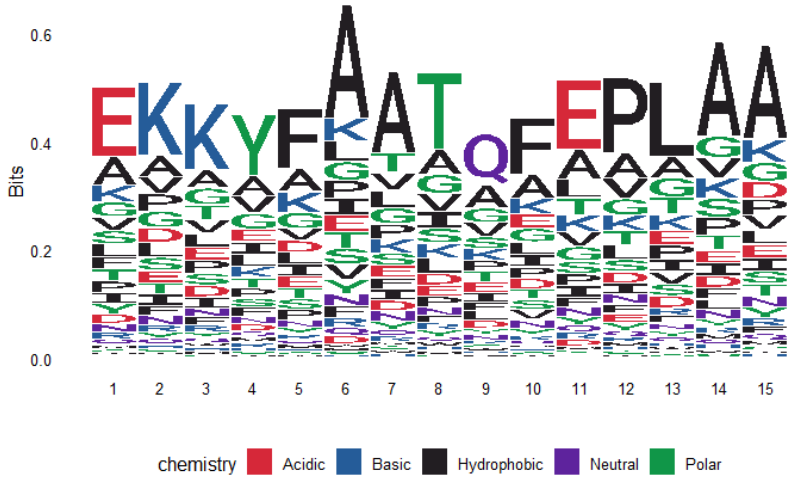

Figure 9. QQSEQLOGO visualisation of the sequence motif of the HLA-DQA1*03:01-DQB1*03:02 (peptide length = 15 amino acids)

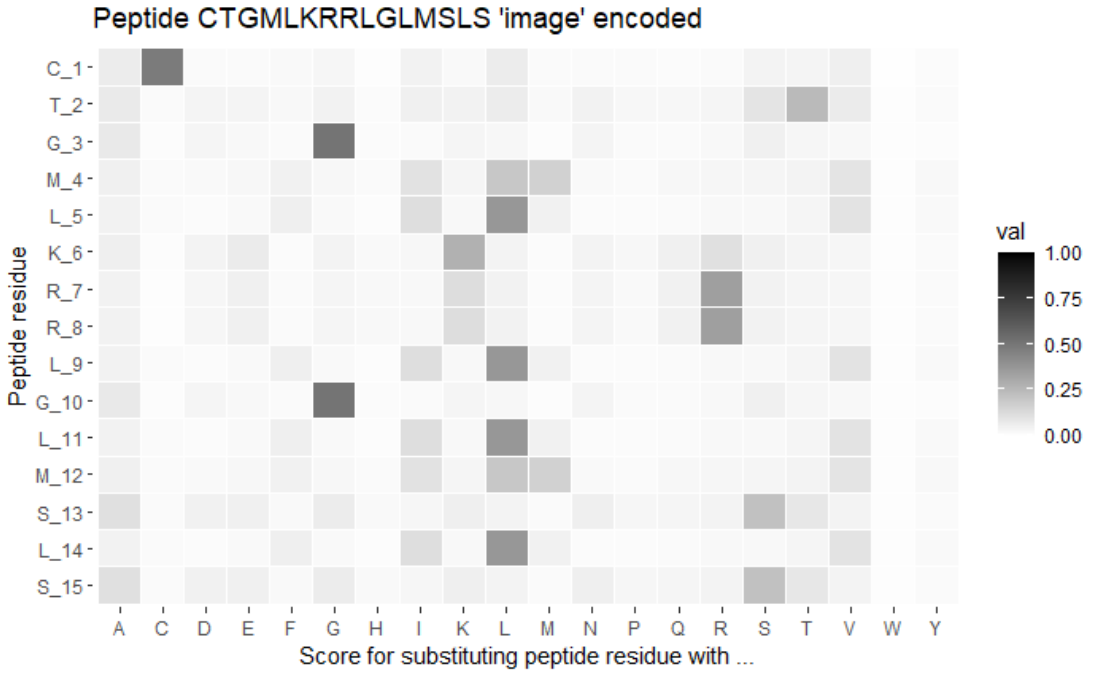

Figure 10. The first peptide of the HLA-DRB1*03:01 allele (peptide length = 15). The Y-axis represents the selected peptide's amino acids (with position included), the X-axis represents all 20 main amino acids. Each position in the peptide become a vector of 20 values, corresponding to the rounded log odds ratio for substituting the amino acid in the peptide with each of the 20 standard amino acids. The final result is a list of peptide 'images' each with values in [0;1]

combination of the above). Data augmentation helps the model "see" the images from different angles, colors, scale. A positive effect of data augmentation makes the model more universal, and the model can make better predictions. A negative side of data augmentation is data enlargement. It duplicates the existing dataset by adding some noise or transforming, resizing images. This negative side can be used as a positive consequence for our dataset. In the MHC-II dataset, it has a lack of data for some alleles and peptides of different lengths. The data augmentation added image rotation was used. The input "images" were rotated by 40 degrees and concatenated with initial "images." It helped to increase the number of peptides twice. This method is not listed in any of the related works but may be used.

Multiple experiments were done for creating a CNN for peptide binding prediction. The final model (Figure 12) has 5 Convolutional 2D layers with 64/32/16/8/4 filters respectfully. Between all layers, RELU was used. Kernel size is 3x3. RELU and Sigmoid

activation functions were used because of binary prediction, but RELU showed better performance because of its ability to change in the y-axis according to change in the x-axis (scale invariance). It was decided to use a relatively large number of filters to find out any hidden intricate patterns. At the last dense layer, a Softmax activation was used. Also, MaxPooling was considered but eventually skipped due to the low-dimensionality of the images.

The model has 3 dropout layers. The dropout rate was gained in empirical experiments by controlling changes in the loss function and AUC score. Dropout is in the proposed method has 25%, 50%, and 30% rates. The first dropout layer stands after convolutional layers, two others - between fully-connected layers. It is not common to set Dropout after convolution, but there is no guarantee that the model would not overfit due to high-order or distal dependencies between different amino acid positions. The fully-connected layers have 180/90/3 units that were taken as a result of empirical experiments as well.

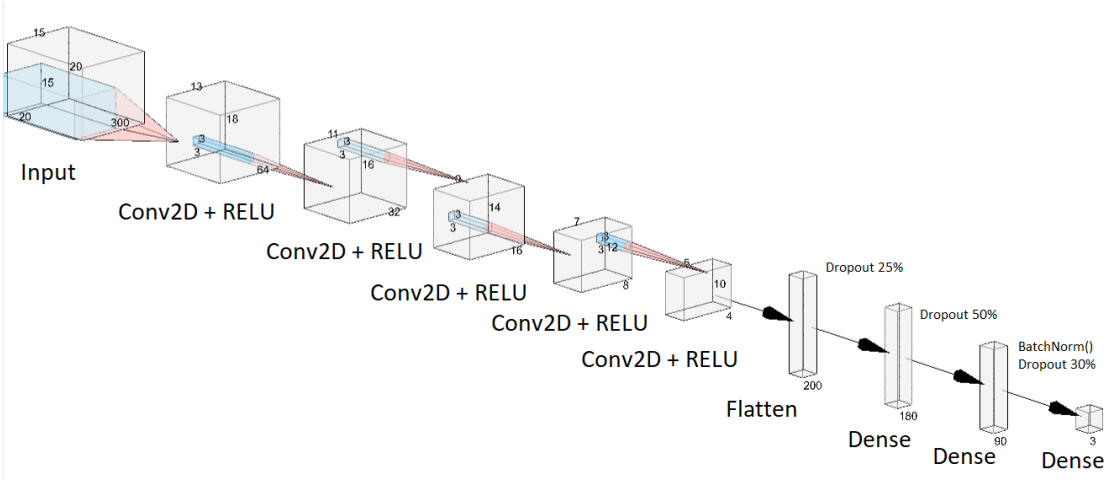

Figure 11. A scheme of the proposed Convolutional Neural Network model

Batch normalization was also used. It is a layer that allows each layer of the network to learn more independently. It is used to normalize the output of the previous layers. Activations scale the input level during normalization. In the proposed model, batch normalization used on ones before the last fully connected layer. It was tried to use several layers of batch normalization in our experiments, but the result was always too penalized. The batch normalization layer is added to the sequential model to standardize the input or output data. Moreover, data inputs to layers deep in the neural network may change after each mini-batch when the weights are updated. It helps by reducing internal covariance shift via mini-batch.

The resulting model is compiled with binary cross-entropy loss function, Rmsprop

optimizer [ZSJ+19]. The evaluations were made by AUC rate. There were discovered alternatives: Hinge Loss [WL07] and Adam optimizer [Zha18] - tends to overfit the model. Adam was increasing the learning rate too much, therefore Rmsprop was taken.

The proposed CNN tends to overfit on DP and DQ locus data. DR allele is the most abundant in the dataset, so the model fitted dominantly on the DR locus. Also, the given model is overtrained for some alleles, leading to lower results on unknown data. Since it has a relatively small number of training examples (app. 1700 peptides for DRB1*03:01, and some MHC complexes have even smaller datasets), overfitting should be the number one concern. Overfitting occurs when the model experiences too few examples or learning on the noise instead of the real signal; thus model would not generalize to new data. The proposed method is generalized, meaning that it does not consider specificities of each allele or locus, it is the same for all peptides of any length, so there can be uncertainties on some particular alleles. The ensemble of models should improve some false predicted values.

To conclude, CNN was trained for 150 epochs. At the beginning of the experiment, it was used 300 epochs, but there were no significant changes after 150 epochs. An example of trained CNN on DQA1*01:01-DQB*1:0501 can be seen in Figure 12, where accuracy score is 79%. The example of a confusion matrix with stats can be seen in Table 4.

5.3.2 Logistic Regression

The logistic regression (LR) was trained as an additional model for the final ensemble. Train control function - 5-fold cross-validation was applied. The number of validations is an assumption made from the multiple related works [NL09, MAN15, RAP+20, HL17]. The tuning parameters were calculated automatically by the caret package via the tune length parameter. The tuneLength parameter set to 3 means that it will evaluate up to 3 kernel parameters. In the case of logistic regression, there are no additional parameters provided. An example of the collective statistics among methods for a specific allele can be seen in Table 4.

5.3.3 Random Forest

The Random Forest (RF) was also trained with 5-fold cross-validation and tuneLength parameter. The number of trees in Random Forest is not limited, so it requires some extra computational power. To fulfill those needs, an additional parallelization was used.

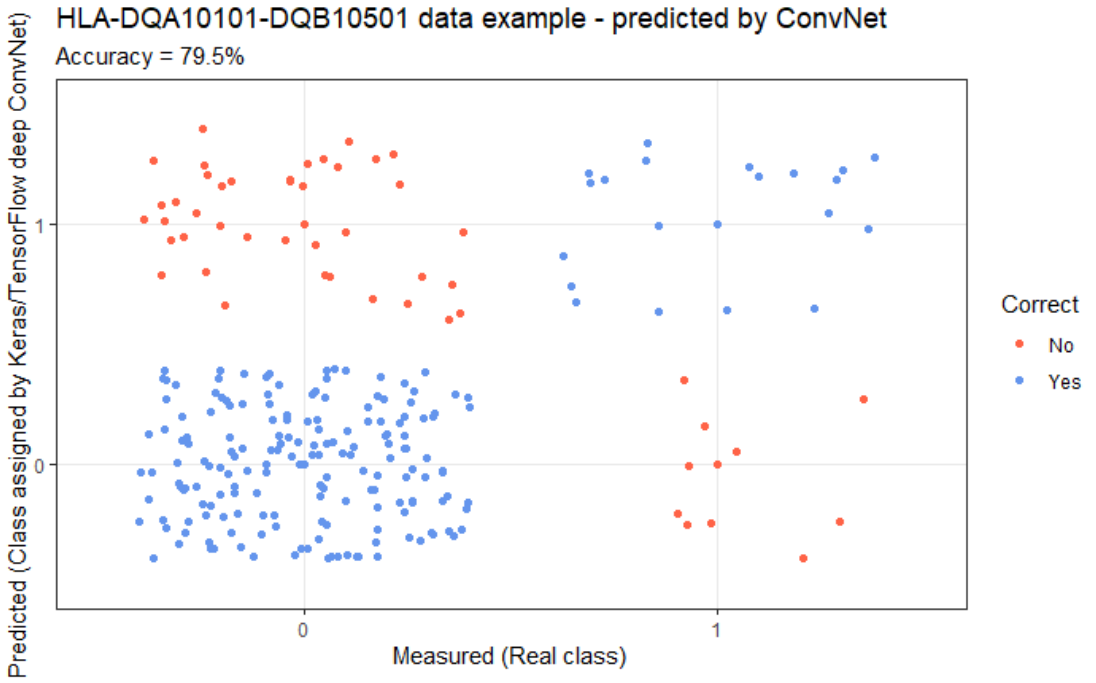

Figure 12. The visualization of the predictions on the test data for DQA1*01:01-DQB*1:0501. Rows and columns represent binders (1) and non-binders (0). Y-axis shows predicted values, X-axis - measured (true) values. Red dots are corresponding to incorrect predictions.

The RF was aligned among all CPU cores via clusters (a cluster library in R). It helped to reduce training time. In most cases, Random Forest showed the best results among all models using only automatically assigned parameters. An example of the collective statistics among methods for a specific allele can be seen in Table 4.

5.3.4 K-nearest neighbour

The K-nearest neighbor (KNN) model was estimated using a 5-fold cross-validation and tuneLength parameter. The number of K clusters defined automatically and equaled to 9. An example of the collective statistics among methods for a specific allele can be seen in Table 4. Both RF and KNN showed less performance accordingly to P-value [accuracy > no information rate] - a metric that describes how accuracy differs with no information rate (NIR). A number closer to 0 represents that the model's accuracy with our parameters is greater than NIR, where NIR is a predicted accuracy of the model

Table 4. An example of a statistic output of the confusion matrices received on HLA-DQA10101-DQB10501 allele.

Statistical metric	**RF**	**KNN**	**LR**	**CNN**
Accuracy	.8497	.8374	.8067	.8037
No Information Rate	.7853	.7975	.7055	.6595
P-value [ACC > NIR]	.00208	.03979	2e-05	6e-09
Kappa	.6126	.5757	.5391	.5511
Mcnemar's Test P-value	6e-05	1e-05	.8011	.1691
Sensitivity	.8477	.8346	.8565	.8791
Specificity	.8571	.8485	.6875	.6577
Positive Predictive Value	.9559	.9559	.8678	.8326
Negative Predictive Value	.6061	.5657	.6667	.7374
Prevalence	.7583	.7975	.7055	.6595
Detection Rate	.6656	.6656	.6043	.5798
Detection Prevalence	.6963	.6963	.6963	.6963
Balanced Accuracy	.8524	.8416	.772	.7684

when no parameters are provided. P-value [accuracy > no information rate] showing how the performance is enhanced with our parameters. For KNN and RF, this metric is still low but not minimal as it can be. Also, Cohen's Kappa rate of more than 0.5 means a good result of the expected accuracy.

5.3.5 Ensemble model

Applying all listed models above, multiple ensemble models were created, and the best one was selected as a final predictor. It consisted of different models like Convolutional neural network, Random Forest, Logistic Regression, and K-nearest neighbor. There is an idea to create a new consensus [WSD$^+$08] approach, a model that collects predictions from other models and evaluates the result using that knowledge.

The first ensembles are basic ensembles. Those are averaging (avg), majority voting (mv), and weighted average (wa). The avg and mv are examples of parallel ensembles.

The average ensemble model suits well for the regression models. Since the predictions are either '0' or '1', averaging does not make much sense for the binary classification. However, it can be performed the averaging of the probabilities of observations. As a next step, probability scores can be converted back to binaries. The scores are validated by 5-fold cross-validation transferred from the original models. The averaging model is

straightforward as in equation 1, where P stands for probability scores and n(models) as a number of models. Then, the avarage score (P(avg)) is converted to binaries as showed in system below, where P(model) corresponds to probabilities predicted by a model with testing set as an input.

$$P(avg) = \begin{cases} 1, & \text{if P(avg) > 0.5} \\ 0, & \text{else} \end{cases}$$

$$P(avg) = (P(KNN) + p(CNN) + P(RF) + P(LR))/n(models) \tag{1}$$

In the majority ensemble, each model has an equal weight. Before the majority voting, all results are converted to binaries in the same way as above. Lately, all the results compared with each other, and a significant point is assigned to the final predictor. For instance, a peptide QSCRRPNAQRFGISN is predicted 1 by KNN, RF, LR, and 0 by CNN. This means that QSCRRPNAQRFGISN will be predicted as 1 - a binder. In a case of a tie, a majority vote would be cast by a random selection.

The second ensemble uses weighted average scores. The main idea is to give a significant "vote" for a better model, so the weights of predictions are higher for more accurate models. The averaging ensemble takes the average score of the probabilities of observations and each model's absolute accuracy. Those final accuracies are summed up. For each model created a weight or importance score (IS) as in equation 2. In this case, the sum of all importance weights equals 1. After that, the weighted average (WA) is calculated as in equation 3, where P(model) corresponds to the set of probabilities predicted by a model with testing set as an input. As a final step, all the probabilities are converted back to binaries.

$$IS = AUC(model)/SUM(AUC(models)) \tag{2}$$

$$WA = SUM(P(model)) * IS \tag{3}$$

If the predictions are highly correlated, using aggregated models might not give better results than individual models. Eventually, it was chosen to compare different models for creating a final ensemble. In most cases, the best scores were achieved by the majority vote ensemble among other ensembles. Listed ensembles are not listed in the final comparison table because these three basic ensembles' best result was similar to the GBM or CNN model. The AUC score was usually varying +/-.04 from the GBM's AUC score. It is decided to list the GBM model in the final table as a delegate from ensemble models.

So far, only basic formulas were used in the top layer. Instead, there are alternatives to that, as a machine learning model of stacking type. Stacking is one of the ways to aggregate the multiple models. In the bottom layer of the stacking, a linear regression was used to map a linear formula for evaluating the predictions. The next step will be to apply logistic regression and Gradient Boosted models as top layer models. The following steps were taken:

- Train the individual base layer models on training data (as was done previously).
- Predict using each base layer model for training data and test data.
- Train the top layer model again on the predictions of the bottom layer models that have been made on the training data.
- Predict using the top layer model with the predictions of bottom layer models that have been made for testing data.

Those steps were made using KNN, RF, and LR models in the stack's lower level. Experiments showed that CNN is hard to interpret for GLM (generalized linear models) and GBM (Gradient Boosted models). At the step, it was additionally trained the out of fold (OOF) prediction probabilities. GLM and GBM were trained with 5-fold cross-validation as well. An example of binding predictions for DQA1*01:01-DQB1*05:01 allele data can be seen in Figure 13. There 3 low-level stack model predictions (KNN, RF, and LR) and 2 top-level ensemble models: GLM and GBM. It was possible to train and compare them because they can interpret predictions as classes and as vectors of predicted probabilities.

The Figure 14 is a boxplot of final predictions made for DQA1*01:01-DQB1*05:01 allele. It shows that the GBM model achieved the best results listed in the plot. Also, it seems convenient to trust this data more compared to basic ensembles models. Figure 15 can be seen as the same plot made for Kappa scores. In DQA1*01:01-DQB1*05:01 allele, Kappa showed a weak general agreement for all the models. It can be caused by the unbalanced distribution of True positives to True negatives. Also, Random Forest and GBM are quite close to the average Kappa score, which is considered as 0.4+. The opposite result with logistic regression. Kappa's score is weak, so the model is not reliable enough.

The results of GBM can be transformed back to peptide data. Using QQSEQPLOT, it can be found out how binders look like in the DQA1*01:01-DQB1*05:01 allele. (Figure 16 shows binders, and Figure 17 shows no binders). After the analysis, it is visible that the binders of our case allele look quite familiar. The peptide EKKYFAATQFEPLAA

```
Call:
summary.resamples(object = rValues)

Models: glm, gbm, rf, knn, lr
Number of resamples: 5

Accuracy
         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
glm 0.7653061 0.7653061 0.7678571 0.7704082 0.7755102 0.7780612    0
gbm 0.7551020 0.7627551 0.7755102 0.7750000 0.7831633 0.7984694    0
rf  0.7551020 0.7729592 0.7729592 0.7714286 0.7755102 0.7806122    0
knn 0.7525510 0.7525510 0.7653061 0.7612245 0.7653061 0.7704082    0
lr  0.6913265 0.7040816 0.7244898 0.7142857 0.7244898 0.7270408    0

Kappa
         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
glm 0.2832918 0.3007600 0.3095420 0.3114256 0.3268810 0.3366529    0
gbm 0.2363636 0.2817401 0.3225452 0.3196257 0.3560108 0.4014689    0
rf  0.2312469 0.3126330 0.3214036 0.3110488 0.3353884 0.3545719    0
knn 0.2459745 0.2650947 0.2917518 0.2873022 0.3007600 0.3329300    0
lr  0.2172421 0.2556312 0.2799020 0.2792188 0.3078091 0.3355098    0
```

Figure 13. The summary of 5 models (including 2 ensemble models) for DQA1*01:01-DQB1*05:01 allele. It is showing two principal measurement scores selected in the thesis: Accuracy (AUC) and Kappa-score. Each mini table is showing minimum, quartile, median, mean, maximum among predicted values.

(there is no peptide with this name in the dataset) may be an absolute binder based on our predictions and averaging results from 69 predicted binders (in case of listed allele).

5.4 Final results

The results were evaluated using the methodology listed in the section 5. Table 5 shows recorded accuracies (AUC scores) and Kappa scores for each allele of the IEDB dataset. For each allele, a model is trained 50 times, and results are averaged. As can be seen, the GBM and RF have the same averaged result among the whole dataset. The Kappa scores are relatively weak. It means that the expected agreement is lower than expected, so the model's performance is better than a guess but considerably weak. It can be explained by an imbalance of the data when there is a huge disbalance in the number of correctly predicted values.

Also, it shows that the expected result in unknown data can be lower. The best-

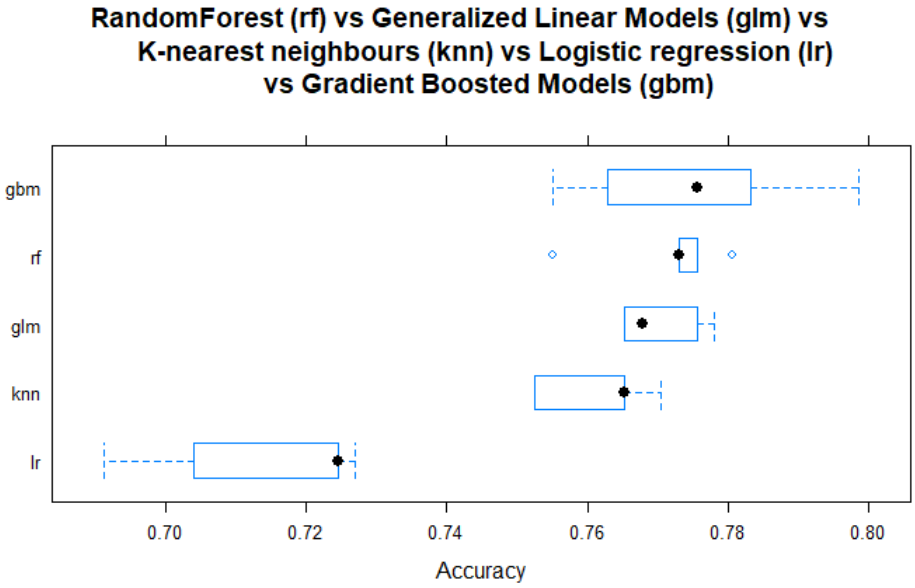

Figure 14. A boxplot with accuracy distribution of 5 models (including 2 ensemble models) for DQA1*01:01-DQB1*05:01 allele. The X-axis represents the accuracy distribution, Y-axis - model's name.

proposed model is CNN. It has the highest AUC-score and Kappa-score. In general, a different model can be considered for different alleles to take the best predictions. The CNN performed exceptionally well on the whole dataset, but multiple alleles were predicted better by another model. For instance, in DRB1*04:04, only one model performed worse than CNN. Simultaneously, the Kappa score is the highest for this allele, so that we can trust CNN predictions more. For some reason, this allele was overfitted for the CNN model, limiting the one-model approach. The approach of using one model for every small dataset has positive and negative sides. The negative side is in the existence of overfitting and underfits along with the dataset. The positive side is the speed/performance ratio and simplicity of the model.

Besides, the experiment showed that straight and manageable predictive models like KNN could perform relatively close to advance models by letting the KNN dynamically select and fit parameters due to the received input. For instance, the standard K neighbors of the KNN in R.Keras library is set to 9. Adding the extra caret package with the tune length parameter allows the model to change the parameters. In every model, the TuneLength was set to 3, and it was used to model that accept extra parameters. If a bigger number is set, it can make a model set random values that will not be beneficial, as it sometimes did during the experiment.

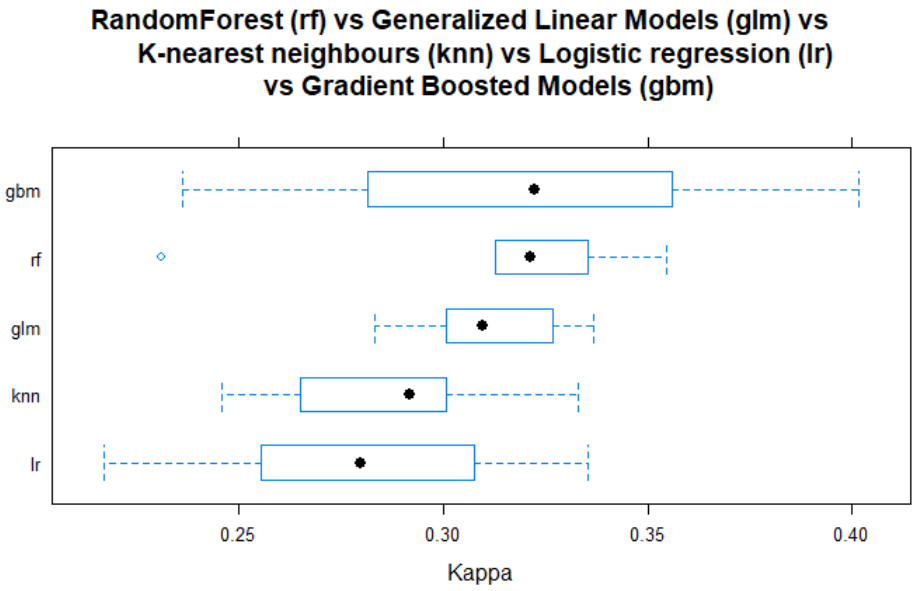

Figure 15. A boxplot with Kappa score distribution of 5 models (including 2 ensemble models) for DQA1*01:01-DQB1*05:01 allele. The x-axis represents the Kappa score distribution, y-axis - model's name.

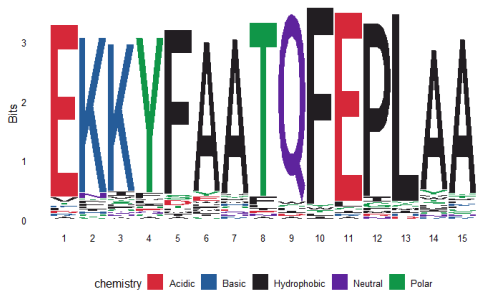

Figure 16. The predicted sequence logos for binders of DQA1*01:01-DQB1*05:01 allele

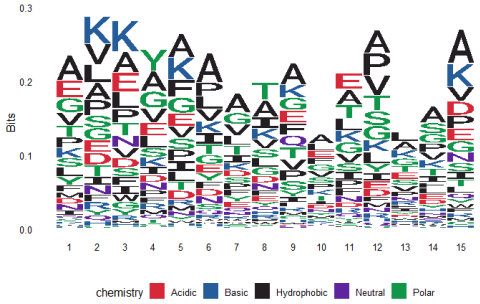

Figure 17. The predicted sequence logos for no-binders of DQA1*01:01-DQB1*05:01 allele

Table 5. The final results of the proposed method

Allelelic variant	**GLM**		**GBM**		**RF**		**KNN**		**LR**		**CNN**	
	AUC	**Kappa**	**AUC**	**Kappa**	**AUC**	**Kappa**	**AUC**	**Kappa**	**AUC**	**Kappa**	**AUC**	**Kappa**
DRB1*01:01	.7684	.3088	.7714	.3093	.7714	.311	.7591	.2822	.7163	.2787	**.8436**	.6226
DRB1*03:01	.7009	.1908	.6989	.1769	.6981	.1459	.6635	.1106	.6607	.1651	**.7585**	.39
DRB1*04:01	.5839	.1416	.5984	.158	.5953	.1591	.5568	.0827	.5752	.1277	**.638**	.2919
DRB1*04:04	**.7055**	.1184	.7044	.123	.6997	.0959	.6892	.0317	.5794	.063	.669	.1511
DRB1*04:05	.6075	.215	.6113	.2265	.6044	.209	.5392	.0789	.5952	.1908	**.6984**	.312
DRB1*07:01	.6416	.2219	.6638	.2429	.6567	.2066	.6066	.1301	.6276	.2014	**.7177**	.4012
DRB1*08:02	.6165	.0735	.6379	.1212	.6352	.1282	.5582	.0615	.5812	.065	**.6952**	.192
DRB1*09:01	5707	.1202	.5983	.1691	.5929	.1569	.5551	.0775	.5631	.1114	**.6684**	.3371
DRB1*11:01	.6312	.2454	.6293	.2403	.6386	.2534	.5799	.1424	.625	.231	**.7294**	.4551
DRB1*13:02	.6106	.1465	.6216	.1623	.6208	.159	.5749	.064	.5996	.1533	**.6897**	.2222
DRB1*15:01	.6108	.2161	.6124	.2155	.6072	.2049	.553	.0966	.6	.194	**.7553**	.2301
DRB3*01:01	.708	.0286	.7176	.1152	.7117	.0308	.6944	.0228	.6753	.1321	**.7452**	.1526
DRB4*01:01	.6181	.1961	.6463	.2499	.6369	.1958	.5733	.1141	.6605	.1504	**.7166**	.3924
DRB5*01:01	.6298	.2172	.646	.2415	.64	.207	.5874	.1066	.6116	.1896	**.7197**	.33
DQA1*01:01-DQB1*05:01	.7704	.3114	.775	.3196	.7714	.311	.7612	.2873	.7142	.2792	**.8466**	.62
DQA1*01:02-DQB1*06:02	.6363	.2147	.6701	.2664	.6754	.2764	.598	.0938	.6367	.2173	**.8016**	.6321
DQA1*03:01-DQB1*03:02	.7191	.2367	.7186	.2266	.7202	.2307	.6849	.1955	.6481	.1642	**.7632**	.4217
DQA1*04:02-DQB1*04:02	.6894	.3418	.6991	.349	.7029	.3485	.667	.2963	.67	.3125	**.8122**	.6142
DQA1*05:01-DQB1*02:01	.7215	.3056	.7257	.3074	**.7324**	.3026	.7061	.2581	.6895	.2922	.7133	.6397
DQA1*05:01-DQB1*03:01	.6864	.2809	.6974	.2772	.7004	.2847	.669	.2265	.6682	.269	**.8015**	.5775
DPA1*01:01-DPB1*04:01	.6958	.3368	6974	.3312	.7065	.3319	.6796	.2931	.6709	.3106	**.8576**	.6947
DPA1*01:03-DPB1*02:01	.6852	.3161	.6759	.2947	.6867	.3416	.6604	.292	.6563	.2967	**.808**	.6135
DPA1*02:01-DPB1*01:01	.6701	.3153	.6618	.3188	.6628	.319	.6218	.2321	.6426	.2823	**.8255**	.5315
DPA1*02:01-DPB1*05:01	.7168	.3314	.7323	.3515	.7344	.3535	.6873	.2964	.6681	.274	**.7788**	.5249
DPA1*03:01-DPB1*04:01	.6824	.3364	.6804	.3229	.6902	.3399	.65	.2883	.6613	.3059	**.8019**	.5979
DPB1*03:01-DPB1*04:01	.7802	.2475	.7781	.239	.7768	.2075	.7428	.1402	.7	.212	**.7951**	.335
Mean	**.6714**	**.2412**	**.6795**	**.2444**	**.6795**	**.2456**	**.6391**	**.1527**	**.6421**	**.165**	**.7557**	**.4339**

In table 6, there is a comparison of predictions done by other methods listed in section 3. Those are the best models available in 2020, and usually, they can be considered benchmarks to check the performance. There are multiple available servers where the antigens (peptides in our case) can be predicted to bind to a certain MHC complex. Moreover, the NN-align model was trained on the same dataset that was used in the thesis. A new updated version of NetMHCIIpan tries to reduce overfitting by accumulating different epitope databases. The train data of NetMHCIIpan includes the data from IEDB that was also used in this thesis. In a column of the proposed method, the highest predicted value of a specific model was used due to our ability to select any submodel of the ensemble with the best performance.

The reviewed model's AUC values are taken from the related articles: "Peptide binding predictions for HLA DR, DP and DQ molecules"[WSK$^+$10], and research about SMM-align[MN07] that was covered in related works.

NetMHCII was used in comparisons instead of a netMHCIIpan. It seems reasonable to compare the first versions of famous models with the proposed method. Also, it seems that our binary predictions perform relatively close to the regression models. On average, the proposed model performed better than PROPRED, netMHCII. The proposed model can also be applied to any allelic variant and has its strong side on predicting the DRB locus data.

The main task of the proposed method is to calculate the binding probability to specific allelic variants. So it is feasible to compare classification and regression models in this case because regression models can be turned into classifiers as well. They define binders and non-binders by IC50 value. If IC50 < 1000 - then the antigen is a binder to the MHC (note: it is common to use lower IC50 values to define binders). The positive side of using regression models lies in finding possible IC50 values for the peptide. In this case, the model can save time and money for finding the IC50 values for unknown newly discovered peptides, and expensive mass spectrometry (MS) devices will become less needed, and they will be used mainly for producing realistic laboratory-made train and test data. Unfortunately, there is no such model that will evaluate as precisely as a real-life experiment. However, the proposed model can be used as a core of any other model to find or classify the binding peptides rapidly. It can save some time for MS devices and filter out some peptides that are useless or no-binders. For example, this will be important when peptides need to be urgently found to produce a vaccine.

Table 6. Comparison of AUC values for the different MHC-II prediction methods

Allelic variant	**Proposed method**	**ARB**	**SMM-align**	**PROPRED**	**Comblib**	**NN-align**	**Consensus**	**netMHCII**
DRB1*01:01	**.8436**	.77	.798	.72	.739	.843	.81	.716
DRB1*03:01	.7585	.753	.852	.693	-	.887	.862	.765
DRB1*04:01	.638	.731	.781	.737	-	.813	.799	.758
DRB1*04:04	.7055	.707	.816	.769	-	.823	.826	.785
DRB1*04:05	.6984	.771	.822	.767	-	.87	.847	.735
DRB1*07:01	.7177	.767	.834	.776	.762	.869	.851	.787
DRB1*08:02	.6952	.702	.741	.647	-	.796	.772	.756
DRB1*09:01	.6684	.747	.765	-	.572	.81	.801	.775
DRB1*11:01	.7294	.8	.864	.804	-	.9	.88	.734
DRB1*13:02	.6897	.727	.797	.6	-	.814	.796	.818
DRB1*15:01	.7553	.763	.796	.743	-	.852	.82	.736
DRB3*01:01	.7452	.709	.819	-	.655	.856	.834	-
DRB4*01:01	.7166	.785	.816	-	-	.886	.848	.736
DRB5*01:01	.7197	.76	.832	.728	.64	-	-	-
DQA1*01:01-DQB1*05:01	.8466	.871	.93	-	.809	.945	.933	.6
DQA1*01:02-DQB1*06:02	.8016	.777	.838	-	.765	.88	.851	.81
DQA1*03:01-DQB1*03:02	.7632	.748	.807	-	.698	.851	.823	.81
DQA1*04:02-DQB1*04:02	.8122	.845	.896	-	.681	.922	.908	.81
DQA1*05:01-DQB1*02:01	.7324	.855	.901	-	.586	.932	.917	.76
DQA1*05:01-DQB1*03:01	.8015	.844	.91	-	.802	.927	.917	.77
DPA1*01:01-DPB1*04:01	.8576	.823	.921	-	.84	.943	.932	-
DPA1*01:03-DPB1*02:01	.808	.847	.93	-	.833	.947	.938	-
DPA1*02:01-DPB1*01:01	.8255	.824	.909	-	.849	.944	.927	-
DPA1*02:01-DPB1*05:01	.7788	.859	.923	-	.867	.956	.942	-
DPA1*03:01-DPB1*04:01	.8019	.821	.932	-	.864	.949	.938	-
DPB1*03:01-DPB1*04:01	.7951	-	-	-	-	-	-	-
Average	**.7579**	**.7842**	**.8492**	**.7258**	**.7476**	**.8568**	**.8655**	**.7539**

6 Discussion

At the beginning of the study, a preliminary analysis was done, which showed that peptide chains could be decoded using the BLOSUM[HH92] algorithm and represented as a matrix. The matrices of binary values are possible to represent as images, where each value of a matrix is a pixel of the image. Regarding the related work, it is common to use neural networks for the given task. For modeling, a convolutional neural network (CNN) was chosen because it can evaluate results without in-depth preliminary biological knowledge, and it is an easily manageable model which allows us to architect a stack of parameters: a number of convolutional layers, the order of additional parameters and layers, et cetera. In the datasets, not all alleles are equally present, and it is a common problem in binding prediction to find more data for the predictive model. This problem was particularly resolved by a standard computer vision technique called data augmentation. It allowed us to double each dataset by simply rotating our matrix images. To process image data, CNN was chosen because of its ability to process different types of data. The final CNN model has 5 convolution layers with a different number of filters. RELU and Softmax activation functions are used to find nonlinearities in the data and 3 fully connected layers with 273 units. Besides, to reduce overfitting, Dropout, and Batch Normalization were used. The additional idea was to implement an ensemble of different models and create a consensus model that will conduct results from multiple models and evaluate them better. It was decided to train KNN, LR, and RF models and feed them to the more complex ensemble model. The models were trained with multiple hyperparameters parameters, and the optimal parameters were selected for each model. There were multiple ensembles: basic - averaging probabilities, major vote, weighted average; stacking models - GBM and GLM.

Each preliminary model's output can be represented both as classification data or assets of probability scores - regression data. The KNN. RF and LR were fed to the stack. Random forest (RF) usually showed similar performance as a CNN, so it is decided to boost our algorithm's speed by skipping CNN in the ensemble. The final results showed that the best predictions were achieved mostly by CNN and sometimes by GBM or RF, but over different alleles, a CNN takes the lead. Furthermore, a CNN showed better average Kappa scores meaning that it is a more reliable model, and there is a moderate-to-good agreement in the scores. Finally, the best-predicted scores were selected either from ensembles or from the separate model (CNN in most of the cases). A comparison with other research showed that the proposed method is competitive to users. It can be applied to an allelic variant data, and the average AUC prediction score along the whole IEDB dataset is 0.7579.

Limitations are based on the massive polymorphism of MHC genes, unbalanced

peptide samples, the nonlinear high-order or distal dependencies between different amino acid positions of the binding peptides, which makes every model overfit predictions. The model was tested only on peptides with 15 residues, and the IC50 threshold was set to 1000, but the desired threshold might be different for a different types of tasks. The model trained on heavily unbalanced data, so it can be biased in some cases (even though cross-validation and oversampling by data augmentation were used). There were used additional dropout layers after the convolutional layer because of high overfitting, but there is no guarantee that the model would not overfit. As a result, the model can produce worse predictions on unknown data.

Future work The model can be developed further by using the output data to estimate the binding core sequence. Figure 16 demonstrates how binder should approximately look like. The future development should emphasize a binding core of each peptide. Using the Pocket algorithm [ZLN09] described in multiple reviewed research, the sequence of amino acids inside the peptide can be calculated with the most significant sum of weights and be assigned as a binding core of the antigen (peptide).

Conclusion. In this thesis, a set of models was created that help to classify peptides into two groups, those that bind to MHCII and those that do not bind. The main idea was to implement a reliable, manageable, and fast algorithm for different antigens that will or will not bind to the MHC-II complex's cleft. As a result, it takes just several milliseconds to estimate how does the set of peptides bind.

Also, the model can be incorporated into another model to boost performance. The performance and speed can be developed further to use the model to help differentiate the data before expensive and time-consuming mass spectrometry lab experiments.

7 Acknowledgement

I want to thank my supervisors Priit Adler and Ahto Salumets, for the multiple ideas that helped finish my work. Their constant work, dedication, and commitment helped me better understand the topic and not be afraid of "scary" biological terms. I am thankful for the time they spent. The information that I received helped me to rise again and again after I gave up on my work. An additional thanks to Ahto for spending extra time on personal lectures dedicated to the MHC-II complex and detailed proofreading.

Moreover, I want to thank my family for their constant support. Also, thanks to my friends: Olha K., Viacheslav K., Mykhailo Y., Kateryna K., Oleksandra P., Diana B., and others for their constant support during hard times.

A big thanks to Anastasia R. for proofreading the thesis and reducing my orthographic mistakes.

Gratitude to the University of Tartu and the Institute of Computer Science for the opportunity to visit Estonia and earn priceless memories, skills, and friends.

References

[ARB+19] Bruno Alvarez, Birkir Reynisson, Carolina Barra, Søren Buus, Nicola Ternette, Tim Connelley, Massimo Andreatta, and Morten Nielsen. Nnalign_ma; mhc peptidome deconvolution for accurate mhc binding motif characterization and improved t-cell epitope predictions. *Molecular & Cellular Proteomics*, 18(12):2459–2477, 2019.

[Bel12] JA Bellanti. Immunology iv: Clinical applications in health and disease, 2012.

[bri21] Cell. *Encyclopedia Britannica*, 2021.

[Bud16] Amar Budhiraja. Dropout in (Deep) Machine learning. 2016. https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5.

[cnn] Image classification. Eurocode standards. `https://www.eurocode.us/theory-of-structures/info-yab-1.html`.

[CUG+93] Roman M Chicz, Robert G Urban, Joan C Gorga, DA Vignali, William S Lane, and Jack L Strominger. Specificity and promiscuity among naturally processed peptides bound to hla-dr alleles. *The Journal of experimental medicine*, 178(1):27–47, 1993.

[CVWV97] Richard T Carson, Kate M Vignali, David L Woodland, and Dario AA Vignali. T cell receptor recognition of mhc class ii–bound peptide flanking residues enhances immunogenicity and results in altered tcr v region usage. *Immunity*, 1997.

[CZG97] Flora Castellino, Guangming Zhong, and Ronald N Germain. Antigen presentation by mhc class ii molecules: Invariant chain function, protein trafficking, and the molecular basis of diverse determinant capture. *Human Immunology*, 54(2):159 – 169, 1997.

[df] Immune deficiency foundation. The immune system and primary immunodeficiency. `https://primaryimmune.org/immune-system-and-primary-immunodeficiency`.

[FLW+14] Ying Fan, Ruoshui Lu, Lusheng Wang, Massimo Andreatta, and Shuai Cheng Li. Quantifying significance of mhc ii residues. 11(1), 2014.

[HH92] S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. 1992.

[HL17] Jianjun Hu and Zhonghao Liu. Deepmhc: Deep convolutional neural networks for high-performance peptide-mhc binding affinity prediction. *bioRxiv*, 2017.

[HSSS92] Uwe Hobohm, Michael Scharf, Reinhard Schneider, and Chris Sander. Selection of representative protein data sets. *Protein Science*, 1(3):409–417, 1992.

[IED] Immune Epitope Database and Analysis Resource. `http://www.iedb.org/`.

[imm14] immunopaedia.org. Immunopaedia. immunology. the basics of the immune system. mhc antigen presentation, 2014. `https://www.immunopaedia.org.za/immunology/basics/4-mhc-antigen-presentation/`.

[KB20] Abhinav Kaushik Franz Cemic Vanessa Heger Harald Renz Kari Nadeau Chrysanthi Skevaki Kathrin Balz, Meng Chen. Homologies between sars-cov-2 and allergen proteins may direct t cell-mediated heterologous immune responses, 2020. `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7553154/`.

[Kha18a] Renu Khandelwal. Decision Tree and Random Forest. 2018. https://medium.com/datadriveninvestor/decision-tree-and-random-forest-e174686dd9eb.

[Kha18b] Renu Khandelwal. K-Nearest Neighbors(KNN). 2018. https://medium.com/datadriveninvestor/k-nearest-neighbors-knn-7b4bd0128da7.

[Kwa19] Sharon Kwak. Very Basic Explanations of Boosting Classifiers. 2019. https://medium.com/datadriveninvestor/boosting-classifiers-e7638c41736a.

[LEJ] Michał Burdukiewicz Leon Eyrich Jessen. PepTools - An Immunoinformatics (Immunological Bioinformatics) R-package for working with peptide data. https://github.com/leonjessen/PepTools.

[LLP+95] Ragnar Lindstedt, Monika Liljedahl, Annick Péléraux, Per A Peterson, and Lars Karlsson. The mhc class ii molecule h2-m is targeted to an endosomal compartment by a tyrosine-based targeting motif. *Immunity*, 3(5):561–572, 1995.

[lr19] Evaluating Logistic Regression Models, 2019. https://insights.blackcoffer.com/evaluating-logistic-regression-models/.

[MAN15] Michael Rasmussen Anette Stryhn Søren Buus Massimo Andreatta, Edita Karosiene and Morten Nielsen. Accurate pan-specific prediction of peptide-MHC class II binding affinity with improved binding core identification. *Immunogenetics*, 2015.

[MN07] Claus Lundegaard Ole Lund Morten Nielsen. Prediction of mhc class ii binding affinity using smm-align, a novel stabilization matrix alignment method. *BMC Bioinformatics*, 07 2007.

[NAPB20] Morten Nielsen, Massimo Andreatta, Bjoern Peters, and Søren Buus. Immunoinformatics: predicting peptide–mhc binding. *Annual Review of Biomedical Data Science*, 3:191–215, 2020.

[NL09] Morten Nielsen and Ole Lund. NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. *BMC Bioinformatics*, 2009.

[oA00] The University of Arizona. The biology project. immunology, 2000. http://www.biology.arizona.edu/immunology/tutorials/immunology/page3.html.

[OvH+14] David O'Sullivan, Gerritje J.W. van der Windt, Stanley Ching-Cheng Huang, Jonathan D. Curtis, Chih-Hao Chang, Michael D. Buck, Jing Qiu, Amber M. Smith, Wing Y. Lam, Lisa M. DiPlato, Fong-Fu Hsu, Morris J. Birnbaum, Edward J. Pearce, and Erika L. Pearce. Memory cd8+ t cells use cell-intrinsic lipolysis to support the metabolic programming necessary for development. *Immunity*, 41(1):75 – 88, 2014.

[RAP+20] Birkir Reynisson, Bruno Alvarez, Sinu Paul, Bjoern Peters, and Morten Nielsen. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Research*, 48(W1):W449–W454, 05 2020.

[rel] What is a relu activation function in keras and why is it used? https://www.dezyre.com/recipes/what-is-relu-activation-function-keras-and-why-is-it-used.

[RSBH+97] David S Riddle, Jed V Santiago, Susan T Bray-Hall, Nikunj Doshi, Viara P Grantcharova, Qian Yi, and David Baker. Functional rapidly folding proteins from simplified amino acid sequences. *Nature structural biology*, 4(10):805–809, 1997.

[SAM+08] John Sidney, Erika Assarsson, Carrie Moore, Sandy Ngo, Clemencia Pinilla, Alessandro Sette, and Bjoern Peters. Quantitative peptide binding motifs for 19 human and mouse mhc class i molecules derived using positional scanning combinatorial peptide libraries. *Immunome research*, 4(1):2, 2008.

[Sta] CS231n: Convolutional Neural Networks for Visual Recognition. `http://cs231n.stanford.edu/`.

[vol] Antigen Processing and Presentation. `https://www.immunology.org/public-information/bitesized-immunology/systems-and-processes/antigen-processing-and-presentation`.

[Wag17] Omar Wagih. ggseqlogo: a versatile R package for drawing sequence logos. *Bioinformatics*, 33(22):3645–3647, 07 2017.

[WL07] Yichao Wu and Yufeng Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007.

[WSD+08] Peng Wang, John Sidney, Courtney Dow, Bianca Mothé, Alessandro Sette, and Bjoern Peters. A systematic assessment of mhc class ii peptide binding predictions and evaluation of a consensus approach. *PLoS Comput Biol*, 4(4):e1000048, 2008.

[WSK+10] Peng Wang, John Sidney, Yohan Kim, Alessandro Sette, Ole Lund, Morten Nielsen, and Bjoern Peters. Peptide binding predictions for hla dr, dp and dq molecules. *BMC bioinformatics*, 11(1):568, 2010.

[ZCW+12] Lianming Zhang, Yiqing Chen, Hau-San Wong, Shuigeng Zhou, Hiroshi Mamitsuka, and Shanfeng Zhu. Tepitopepan: Extending tepitope for peptide binding prediction covering over 700 hla-dr molecules. *PLOS ONE*, 7(2):1–10, 02 2012.

[Zha18] Zijun Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–2. IEEE, 2018.

[ZLN09] Hao Zhang, Ole Lund, and Morten Nielsen. The pickpocket method for predicting binding specificities for receptors based on receptor pocket similarities: application to mhc-peptide binding. *Bioinformatics*, 25(10):1293–1299, 2009.

[ZSJ+19] Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11127–11135, 2019.

[ZUMZ11] Lianming Zhang, Keiko Udaka, Hiroshi Mamitsuka, and Shanfeng Zhu. Toward more accurate pan-specific MHC-peptide binding prediction: a review of current methods and tools. *Briefings in Bioinformatics*, 13(3), 09 2011.

Appendix

I. Code

The source code for the proposed method is located in the following GitHub repository:

https://github.com/myskovik/MasterThesis

Access to the repository can be granted upon sending an email to the address:

2musuk2008@gmail.com

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Viktor Mysko**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Prediction of MHC class II binding peptides
   supervised by Priit Adler and Ahto Salumets

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains these rights specified in p. 1 and 2

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Tartu, 14.01.2021